UNIVERSITY OF THE PHILIPPINES MANILA
COLLEGE OF ARTS AND SCIENCES
DEPARTMENT OF PHYSICAL SCIENCE AND MATHEMATICS

VirHoLex 2.0
(Virus-Host Interaction Lexicon)

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Auradee B. Concepcion
2008-28985

April 2013

ACCEPTANCE SHEET

The Special Problem entitled "VirHoLex 2.0 (Virus-Host Interaction Lexicon)" prepared and submitted by Auradee B. Concepcion in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

_____

Ma. Sheila A. Magboo, M.S.
Adviser

EXAMINERS:

		Approved	**Disapproved**
1.	Gregorio B. Baes, Ph.D. (candidate)	_____	_____
2.	Avegail D. Carpio, M.S.	_____	_____
3.	Richard Bryann L. Chua, Ph.D. (candidate)	_____	_____
4.	Aldrich Colin K. Co, M.S. (candidate)	_____	_____
5.	Perlita E. Gasmen M.S. (candidate)	_____	_____
6.	Vincent Peter C. Magboo, M.D., M.S.	_____	_____
7.	Geoffrey A. Solano,Ph.D. (candidate)	_____	_____
8.	Bernie B. Terrado, M.S. (candidate)	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

_____

Avegail D. Carpio, M.S.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

_____

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

_____

Alex C. Gonzaga, Ph.D.
Dean
College of Arts and Sciences

ABSTRACT

Elepaño et al. has successfully accomplished all features and functional requirements necessary for the development of VirHoLex. However, the system was not thoroughly tested after each module was integrated, generating a number of system bugs as a result. Though Morales has partially refactored VirHoLex, specifically the Virho References and Virho Hotspots modules, the modifications done were not sufficed to make the codes maintainable.

In order to preserve the credibility of the system, Struts framework was incorporated into the program structure of VirHoLex after the system was debugged. Struts is built on MVC design pattern that promotes separation of concerns, that is, separating business logic from the user's view. The whole VirHoLex was refactored a) to remove jumbled and redundant codes; b) to make the codes reusable and modular; and c) to promote programming style uniformity for all modules. This will allow the future developers to easily understand and modify the system.

In addition, the use of Subversion and Mantis Bug Tracking tool is pertinent to the study since it permits the developers to easily track down the history of file changes, and to have a proper documentation of system bugs and new features respectively. More importantly, the second version of VirHoLex was deployed on MaxPlanck server so that virologists can fully utilize the system.

Keywords: *VirHoLex, Struts, Subversion, Mantis Bug Tracking System, MaxPlanck server, Debugging*

TABLE OF CONTENTS

I. INTRODUCTION

A. Background of the Study

Virus is an acellular microorganism which can only grow and reproduce inside the living cells of the organisms – from animals and plants to bacteria and archaea. It consumes the host cells' chemical machinery and metabolism for replication and self-assembly resulting to the death of the cells [1]. The common causes of death include cell lysis, apoptosis, and the alteration to cell's surface membrane [2]. In fact, viruses are causative agents of many infectious diseases such as Influenza, Hepatitis and HIV/AIDS that afflicted millions of people worldwide especially to those living in underdeveloped countries where medicines are not widely available.

With the fast advancements in technology the treatment and prevention for such diseases is much easier to implement to the infected areas. Nevertheless, the propensity of the viruses to develop drug resistance gives no guarantee on the effectiveness of the current medicines. Last August 29, 2011 U.N. Food and Agriculture Organization warned about the new mutant strain of H5N1 avian influenza virus which bypassed the defenses of current vaccines with unpredictable risks to humans [2].

Viruses have continuously imposed an urgent and ongoing threat in public health worldwide. Hence various groups of virologists are conducting numerous researches and studies on viruses, including the nature and the structures, in order to produce cure and preventions for these diseases. With newly emerging and reemerging viral diseases, the demand for web-based application systems is dramatically increasing so that data can be efficiently managed and quickly accessed when needed. A handful of applications were

created to store merely virus information. However, complexity arises when both viruses (virology) and host cell reactions (immunology) are taken into consideration. Since viruses have the innate capability to mutate and adapt to the environment, host cells instead of viruses is the new focus in some of the studies. This alternative method involves changing the host receptor molecules without necessarily altering the cell's normal function. The call to study virus-host interaction dynamics with the systems biology approach has gave rise to the development of **Vir**us-**Ho**st Interaction **Lex**icon, commonly known as VirHoLex [4].

VirHoLex is a web-based application system created by a group of students from two academic institutions in the Philippines (University of the Philippines Manila and De La Salle University) to set up a **C**ommunity **Or**iented **I**nformation (CORI) platform for Systems of Virology community. The idea originated in the context of a joint project between U. Reichl (Director, Max-Planck-Institute for the Dynamics of Complex Technical Systems, Magdeburg, Germany), J. Haas (LMU & Edinburgh University Medical Schools), J. Rädler (LMU Faculty of Physics) and J. Bantang (UPD National Institute of Physics). The project was inspired by EUCLIS (EUCLOCK Information System), another CORI system developed for the global community of chronobiologist [4].

Elepaño *et al.* created VirHoLex on Java Platform using J2SE 5.0. The system implements Java Servlet and JavaServer Pages (JSP) technologies that runs on Apache Tomcat version 6.0.18 web server. The database is managed using MySQL 5.0.45 [4]. VirHoLex is composed of seven modules *(see Chapter 3 for list of functionalities)*:

a. Virho User Interface Module
b. Virho Registered User Services Module
c. Virho Information Services Module

d. Virho Experiments Module

e. Virho Images Module

f. Virho Models Module

g. Virho References Module

The initial version of the system was released last February 19, 2009 which was then deployed on MaxPlanck server in Germany. In early 2011 Jeric Morales, a B.S. Computer Science student in University of the Philippines Manila, was assigned to refactor VirHoLex specifically Reference Module and Virho Hotspots. The sole purpose of refactoring is to make the program structure intuitive and modular, which is essential for easy bug detection and code reusability.

B. Statement of the Problem

Elepaño *et al.* has successfully carried out all the functional requirements needed for the development of VirHoLex. However, the system did not undergo any testing after individual modules were integrated which allowed several components of the system to function improperly. Some of which could not be used at all. More system flaws were encountered when VirHoLex was finally deployed on MaxPlanck server. These resulted to a handful of bugs that hampers the system's capability to produce desired outcomes causing it to become unusable for quite some time. Though VirHoLex was partially fixed and refactored by Morales, some modules particularly Virho Images and Virho Experiments modules contain many bugs and errors. Virho Images module throws errors when clicked, making the module

not functional at all. As for Virho Experiments module, some of its components are not properly performing the CRUD (Create, Read, Update and Delete) operations.

In addition, VirHoLex was created without using a particular architectural model or framework. Hence the program structure looks jumbled and unorganized, that is, the fragment of business logic is written on the page same with the user interface, making the system more difficult to maintain. Also the Java classes are not reusable because the codes were all written on a single class file per functionality. Another flaw of the system is the use of frames as container. Consequently, the page is always redirected back to the homepage when the page is refreshed since the URL at the address bar is fixed for the entire session.

VirHoLex is a large application system comprising of seven different modules. This is an ongoing project which needs to be maintained and improved by future developers. However, the source codes are not stored in a version control system that properly manages file backups. As a result, the developer has the difficulty to track down changes made on files and to recover original files once an error is encountered.

C. Objectives

The purpose of the study is to create a second version of VirHoLex (VirHoLex 2.0) that is free from any show-stopping bugs, and that has a framework to make the source codes modular and organized. VirHoLex 2.0 is redeployed on MaxPlanck server in Germany to replace the old codes on the server. The list of original objectives and functionalities for each module can be seen in the Chapter 3. On top of the system's original objectives and functionalities, this study has the following additional objectives:

1. Use a version control system (Subversion)

a. Create guidelines on proper versioning of the system

b. Create instructions on proper usage of subversion in Eclipse

- Checking out files
- Checking in (committing) files

2. Identify the bugs in the system

a. List all the identified bugs in Mantis Bug Tracking system

b. Fix all the bugs by manually inspecting the output and the code itself with the help of Eclipse Debug tool

3. Add Struts framework in VirHoLex system after the bugs are fixed
4. Deploy and test VirHoLex 2.0 on MaxPlanck server in Germany

D. Significance of the Study

VirHoLex is an essential tool to virologists who conducts research works and experiments concerning viruses, particularly to those communities whose focus in on virus-host interaction. It is a dynamic repository of biological information that supports analyses and modeling of experimental data, as well as storage of bibliographic references and images. It also has image hotspot feature that serves as a virtual encyclopedia about virus stages information.

By fixing bugs or errors in the system, it allows the users to fully utilize the system without receiving any exception errors, invalid results and other show-stopping bugs that hampers the proper flow of the system. Moreover, by incorporating Struts framework in the system, bugs or errors are minimized since Struts follows Model-View-Controller (MVC) design pattern that separates data model from business logic and user interface. MVC clearly

defines the structure and flow of the program, thus making the system easier to understand, develop and maintain *(see Chapter 3.3 for MVC details)*. It also makes the Java class codes reusable, allowing the system to become modular. The use of Struts framework mitigates the arduous work of the developers on tracing files needed to fix bugs or to add new features. This complete VirHoLex code revamp also includes the suppression of frames as the visible parent container of page content so that the user will not be navigated back to the homepage once the page is refreshed.

On the other hand, the implementation of Subversion tool for VirHoLex development allows the future developers to easily maintain the current and historical versions of the files, which is necessary for collaborative development. It becomes easier to revert back the changes once an error is encountered, and it lessens the difficulty of merging conflicting files.

E. Scope and Limitation

The concern of the study is to identify and manually fix the bugs that are still present in VirHoLex system, particularly Virho Images and Virho Experiments modules. Eclipse debugging tool is used to easily search for and resolve the malfunctioning elements or errors in the program code. Only those bugs that were encountered on the initial version of VirHoLex are reported in the Mantis Bug Tracking system. The versioning of system is linear and incremental. The format that to be used is X.Y.Z where X is the version number of previous system, i.e. 1; Y is the bugs or errors (block, crash, major, minor); and Z is the layout or text issue (tweak, text, trivial). After which, the whole VirHoLex is refactored by incorporating Struts framework into the system's program in order to promote separation of concerns – separating business logic from user interface, and making the codes reusable. The

original features and functionalities are retained unless there are certain revisions provided. More importantly, the target of the study is to make VirHoLex 2.0 works properly on MaxPlanck server in Germany. System bugs that were encountered after the deployment are taken into account as well. On the other hand, Morales source codes serve as the initial version of VirHoLex system which is imported in Agila SVN repository.

F. Assumptions

1. The content of VirHoLex 2.0 is under the responsibility of the person who will be using the system for research works and experiments
2. All the experimental results and information uploaded by the contributors will be pertinent to the system since there is no way to check the relevance of the information and files uploaded
3. It is assumed that the files and images uploaded by the contributors and coordinators will not violate copyright laws.

II. REVIEW OF RELATED LITERATURES

Viruses are found in almost every ecosystem on Earth and considered to be the most abundant type of biological entity [5]. They are responsible for the emergence of several infectious diseases worldwide. Given the fact that viruses have the innate capability to adapt to the environment, they tend to develop drug resistance which makes it even difficult to cure viral diseases. Although there can still be millions of types of viruses that are unknown to people, a number of virologists have taken an initiative to conduct researches and experiments to fully described viruses in detail. With the discovery of new viruses and virus-related diseases, the need for virus database is increasing so that the information gathered from the researches and experiments can be properly organize and manage, and can be accessed by a wide range of people worldwide.

ICTVDB was one of the earliest universal virus database formulated and developed by International Committee on the Taxonomy of Viruses (ICTV) in 1991. ICTVDB provided taxonomic classifications and descriptions to all viruses present in different organisms so that it would be easier for the researchers to understand the relationships among viruses. ICTVDB offered precise virus identification, and it linked the agreed taxonomy to sequence database, which in effect, it turned sequences into correctly identified viruses. This information system adopted decimal code convention in naming viruses because of the peculiar nomenclature used in virology [6].

Virus Particle Explorer (VIPER) supplied information about virus structures. VIPER was a structural database where all virus capsid structures (coordinates) were stored in a standard icosahedral orientation in order to have high-throughput analyses of the virus

structures [7]. However, the data stored in the web server were flat files. Hence, a relational database VIPERdb was built to make the data highly accessible to the user, allowing detailed queries at the atomic level. Together with a concomitant expansion to include all known icosahedral virus structures, it provides a significantly improved resource to user [8].

ViralORFeome 1.0 created and managed ORF clones. ViralORFeome was the first open-access database that provided an integrated set of bioinformatics tools to build a collection of viral ORFs clones in a versatile system suitable for reverse proteomic experiments. This was specially designed to handle all possible variants or mutants of a given ORF so that the cloning procedure can be applied to any emerging virus strains. ViralORFeome database allowed users to manage their collection of viral ORF clones, keep track of expression plasmids derived from 'entry' clones, and share plasmids with other laboratories [9].

RNA Virus Database contained genomic data, analytical tools for the 938 known species of RNA virus, and links to other virus Web sites. RNA Virus Database provided multiple whole-genome alignments, gene and whole-genome translations for all RNA virus species; and identification and taxonomic searching facility [10].

PhEVER was a database regarding virus-host relationship that provided accurate evolutionary and phylogenetic information to understand how viruses interact with their host and to analyze the nature of virus-virus and virus-host lateral gene transfer. PhEVER was a data warehouse of homologous gene families containing sequences of all completely sequenced viruses and from fully sequence cellular organisms. The data was not just limited to all known viral groups but to prokaryotes and eukaryotes sequences as well. The database

also offered a clustering of proteins into homologous families containing at least one viral sequence, as well as pre-computed alignments and phylogenies for each of these families [11].

VirusMINT was a viral protein interaction database that aimed to collect and annotate all interactions reported in the scientific literature between viral and host proteins. The curation effort had focused on manuscripts reporting interactions between human proteins and proteins encoded by some of the most medically relevant viruses: papiloma viruses, human immunodeficiency virus 1, Epstein-Barr virus, hepatitis B virus, hepatitis C virus, herpes viruses and Simian virus 40 [12].

Virus-Host Network (VirHostNet) was a public knowledge base specialized in the management and analysis of integrated virus-virus, virus-host and host-host interaction network coupled to their functional annotations. VirHostNet catered a high-confidence resource of manually curated interactions defined for a wide range of viral species. It also provided an automatic screening of specific protein domains or peptides motifs associated to virus-host interactions which might help to delineate at the proteome-wide scale footprints in both viral and host proteins sequences [13].

OpenFlu database (OpenFluDB) provided a convenient and reliable mechanism to collect, manage, store and distribute worldwide influenza data. OpenFlueDB contained genomic and protein sequences, as well as epidemiological data from more than 27, 000 isolates. The links between the database, the SSM and isolate geolocation tool were innovative functionalities to mine Influenza genetic and epidemiological data as well as its evolution. The contents of the system were supplied by direct user submission, as well as a daily automatic procedure importing data from public repositories [14].

BioHealthBase Bioinformatics Resource Center (BRC) was a public bioinformatics database that focused on influenza virus host-pathogen interactions and virulence. BioHealthBase BRC assembled and integrated a variety of different types of data related to influenza virus, including gene and protein structure and function, sequence variation and immunological epitope information. It served as an extensive integrated repository of data imported from public databases, data derived from various computational algorithms and information curated from the scientific literature. Significant emphasis was given to data related to host-pathogen interactions in order to gain a better understanding of the nature of virulence and host range, and the impact of sequence variation on these phenomena [15].

MVirDB was a microbial database that focused on protein toxins, virulence factors and antibiotic resistance genes for bio-defense application. MVirDB served as a data warehouse of DNA and protein sequence information integrated from eight sequence databases that were publicly available: the Tox-Prot, SCORPION, the PRINTS virulence factors, VFDB, TVFac, Islander, ARGO and a subset of VIDA. Blast tool and browser tool were included in the system to allow the user blast against all DNA or protein sequences, and to let him search the databases to retrieve virulence factor description and download sequences of interests respectively [16].

III. THEORETICAL FRAMEWORK

A. VirHoLex

VirHoLex (Virus-Host Interaction Lexicon) is a Java web-based application system that caters biological information on viruses and their interaction with the hosts. It was created by a group of students from two academic institutions in the Philippines (University of the Philippines Manila and De La Salle University) to set up a Community Oriented Information (CORI) platform for Systems of Virology community. The idea originated in the context of a joint project between U. Reichl (Director, Max-Planck-Institute for the Dynamics of Complex Technical Systems, Magdeburg, Germany), J. Haas (LMU & Edinburgh University Medical Schools), J. Rädler (LMU Faculty of Physics) and J. Bantang (UPD National Institute of Physics). The project was inspired by EUCLIS (EUCLOCK Information System), another CORI system developed for the global community of chronobiologists [4].

The features and functionality of the system were subdivided into seven different modules. These are Virho User Interface, Virho Registered User Services, Virho Information Services, Virho Experiments, Virho Images, Virho Models, and Virho References modules. The following describe the functionalities for each of the modules in details.

1. *Virho User Interface Module* is responsible for the overall appearance of VirHoLex system [18].
2. *Virho Registered User Services Module* provides a user role system that manages different access rights and permissions for each module's project, which includes evaluating account requests and user privilege requests, and changing passwords. The

roles assigned to the users are determined by the Administrator. There is a possibility that the registered user can assume multiple roles with different access levels [19].

3. *Virho Information Services Module* handles the overall flow of information internal and external to the system. It facilitates search function for Virho Experiments, Virho Models, Virho Images and Virho References modules collating results from system's database. Searches on external academic databases namely NCBI[1], EBI[2] and KEGG[3] can also be performed. Users can browse through basic information of available viruses with matching images and hotspots, which adds interactivity to the feature [20].
4. *Virho Experiments Module* stores and manages data, metadata, and other relevant information on laboratory experiments. This module also allows the uploading of files such as images taken from experiments [3].
5. *Virho Images Module* serves as repository of images that are relevant to virology such as the images taken from different experimental modules, graphical analysis of experimental data, etc. These images can be made publicly available or restricted to the image set members. In this module, the users may view, upload and download (set of) images [17].
6. *Virho Models Module* stores the summary descriptions of relevant models and modeling studies that are classified according to the taxonomy presented by Roenneberg. Eeach entry in the collection is provided with attributes (scale, scope, size, type, biological aspects, tools, computational properties, external link, and

[1] http://www.ncbi.nlm.nih.gov/
[2] http://www.ebi.ac.uk/
[3] http://www.genome.jp/kegg/

experimental evidence) and portable document format (pdf) file that contains summary description of the model [21].

7. *Virho References Module* stores bibliographic entries or references provided by the users that can be viewed and downloaded in EndNote or BibTex format [22].

The following tables written below summarize the roles defined per module, the capabilities of each user level, and a more detailed description of each user level for each module.

User Levels	ViRUS	Virho Experiments	Virho References	Virho Images	Virho Models	Virho Basic Virus Information and Hotpots Info
5	Administrator					
4		Experiment Lead Investigator	Collection Coordinator	Image Set Coordinator	Model Coordinator	Hotspots Manager
3		Experiment Investigator	Collection Contributor	Image Set Contributor	Model Contributor	
2		Restricted User				
1		Registered User				
0		Unregistered User				

Table 3.1 User Roles Summary

User Level	Capabilities
0	Unregistered users have the capability of viewing general information about each VirHoLex module but they do not have access to any data. If they wish to gain access, they must first request for an account.
1	Registered users have the same capabilities as unregistered users. In addition to this, registered users can browse through the modules and open accessible information. However, they cannot download or upload data to VirHoLex. *See Table 3.1.6 for some exceptions in Images Module*
2	Restricted users have the same capabilities as registered users. Moreover, they can download available materials, and access some information which is not available to the registered users. However, restricted users still do not have any capability on uploading files.
3	Level 3 users are named differently at each module. Besides having the same capabilities as restricted users, level 3 users have the capability of uploading, adding and editing information on their assigned modules.
4	Level 4 users are named also differently at each module, and they have the same capabilities of Level 3 users. In addition, Level 4 users can create, delete and manage sets of data, and they can manage the members to whom the data within a set of their module is visible. Since Virho R US adopted entirely Clocks R US framework, the role can only be granted to the list of pre-approved users only.

5	Administrators have the same capabilities as restricted users. Aside from this, administrators are in charge of granting request for user accounts as well as assigning Level 4 users the capabilities in VirHoLex.

Table 3.2 Capabilities of each user level

	ViRUS Access Level	Privileges	
		Grant Level 4 Access	Grant Access to Users
5	Administrator	✓	✓
4	Level 4 User	✓	×

Table 3.3 ViRUS User privilege table

	Information Services Access Level	Privileges		
		Query External Database	View/Browse/Search Internal Database	Add/Edit/Delete Virus Information
4	Virus Information Annotator	✓	✓	✓
3	Registered User	✓	✓	×
2		✓	✓	×
1		✓	✓	×
0	Unregistered User	✓	×	×

Table 3.4 VirHo Information Services privilege table

	Experiments Access Level	Privileges				
		View/Browse Experiments	Upload Data Files	Edit Experiment Set Details	Assign Membership in Experiment Set	Create/Delete Experiment Set
4	Experiment Lead Investigator	✓	✓	✓	✓	✓
3	Experiment Investigator	✓	✓	✓	×	×
2	Restricted User	✓	×	×	×	×
1	Registered User	✓	×	×	×	×

Table 3.5 VirHo Experiments privilege table

	Images Access Level	Privileges					
		View/Browse Image Details and Image Set Details	Download Image and Image Sets	Use View Manager	Add/Edit/ Delete Images	Add/Edit/ Delete Image Sets	Assign Membership to Image Sets
4	Image Set Coordinator	✓	✓	✓	✓	✓	✓
3	Image Set Contributor	✓	✓	✓	✓	×	×
2	Restricted User	✓	✓	✓	×	×	×
1	Registered	✓/×*	✓/×*	✓	×	×	×

	User						
* Image set contributor/coordinator can make images available to registered users or to image set members only							

Table 3.6 VirHo Images privilege table

	Models Access Level	Privileges			
		View/Browse/Search Model Details	Download Model Descriptions	Add/Edit/Delete Model Entries	Assign Membership
4	Model Coordinator	✓	✓	✓	✓
3	Model Contributor	✓	✓	✓	×
2	Restricted User	✓	✓	×	×
1	Registered User	✓	×	×	×

Table 3.7 VirHo Models privilege table

	References Access Level	Privileges				
		View/Browse Bibliographic Entries	Download Documents /Export Entries to EndNote or XML Files	Add/Edit Bibliographic Entries	Delete Bibliographic Entries	Manage References Collection User
4	Collection Coordinator	✓	✓	✓	✓	✓
3	Collection Contributor	✓	✓	✓	✓	×
2	Restricted User	✓	✓	×	×	×
1	Registered User	✓	×	×	×	×

Table 3.8 VirHo References privilege table

	Hotspots Access Level	Privileges			
		View Hotpots Diagram and Basic Virus Information	Upload New VirHo Hotspot Diagram	Add/Edit/Delete Hotspot and Basic Information	Manage References Collection User
4	Hotspot Manager	✓	✓	✓	✓
0-3		✓	×	×	×

Table 3.9 VirHo Hotspots privilege table

VirHoLex was developed using JavaServer Pages (JSP) and Java Servlet technologies. The source codes were run on Apache Tomcat 6.0.18 using Java Development Kit (JDK) 1.6,

and MySQL 5.0.45 as the database [4]. The initial version of the system was released on February 19, 2009, and was later deployed on MaxPlanck server in Germany.

B. Software Configuration Management

Software Configuration Management (SCM) is the application of standards and procedures for managing an evolving system [23]. It provides foundation for product and project management [24] which cuts down the system cycle cost. SCM is responsible for establishing and maintaining the integrity and traceability of the system so that the development team can easily track the changes and versions. Since systems may exist in different configurations – maybe due to different computers or systems, client-specific functions etc. – it is the duty of configuration managers to ensure that software versions are handled in a controlled way, and to release new versions to the right customer at the right time [23].

Version Numbering

Version numbering is one of the basic techniques used to have an unambiguous way for identifying each component version. It is the most commonly used identification scheme wherein the component is given an explicit, unique version number, e.g. VirHoLex 2.0 (second version of VirHoLex). Different versions of system may have different functionality, enhanced performance or repaired software faults. Some versions may seem to have equivalent functionality but designed for different hardware and software configurations. Version numbering seems to be simple but it actually maintains a lot of extra information to

keep track of the differences between versions and the relationships between system change proposals and versions [23].

Types of Versioning

1. *Linear Versioning* - Successive versions of software are organized in sequence or series. If the first version is called 1.0, subsequent versions are 1.1, 1.2, and so on.

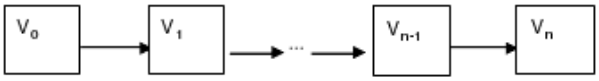


Figure 3.1 Linear Versioning

2. *Branch Versioning* – Multiple versions have branch out from the generic version of the software due to different computers available, different operating systems used, or different client-specific functions

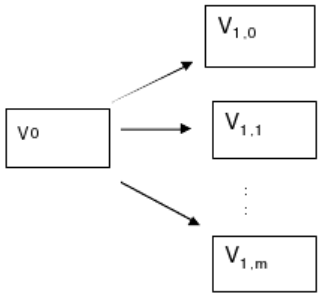


Figure 3.2 Branch Versioning

C. **Subversion**

Subversion (SVN) is an open source *version control system* (VCS) founded in 2000 by CollabNet Inc. It allows developers to easily recover old versions of data and to examine data history by tracking changes made on files. SVN supports collaborative software development,

which permits the developers within a team, working on same files, from overwriting each other's work [25]. Figure 3.3 illustrates the architecture of subversion.

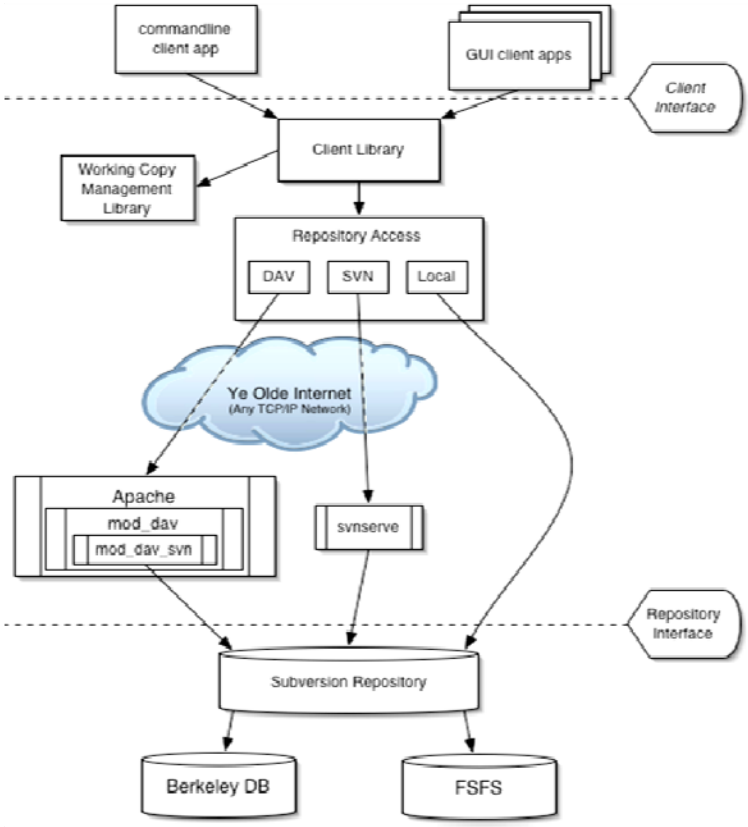


Figure 3.3 Subversion's Architecture

At the top is the subversion client program that manages the local reflections of versioned data called "working copies". At the bottom is the subversion repository that holds all versioned data committed by the clients. Between these two are multiple routes through various Repository Access (RA) layers. Some routes go across computer networks and through network servers which then access the repository. Others bypass network altogether and access the repository directly [25].

Repository

Repository is the core of version control system that acts as the main data warehouse. The information is usually stored in form of a *filesystem tree*—a hierarchy of files and directories. Any number of clients connected to the repository can read and can write to files. By writing data, a client can make the information available to others; whereas by reading data, a client receives information from others [25], as shown on Figure 3.4.

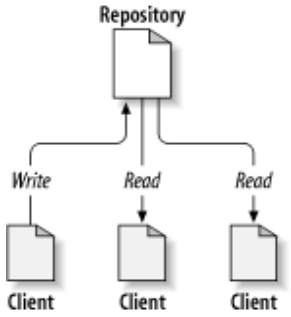


Figure 3.4 A typical client/server system

Revision

The client commits files as a single atomic transaction meaning either all or none of the changes is accepted into the repository. That way, if a crash interrupts the commit, there is no risk that the database will be corrupted. The database will just automatically be restored to its state before the commit began [25].

Each time the client commits files, the repository creates a new state of the filesystem tree, called *revision*. A unique natural number is assigned to each revision, which is applied not on individual files but to the entire trees. Newly created repository contains an empty directory with revision number 0 [25].

D. Model-View-Controller

Model-View-Controller (MVC) is a web architectural model that separates representation of information from user's interaction with it. MVC takes business logic out of the servlet, and puts it in a *model class* – a reusable plain old Java class. The *model* is a combination of business data and methods that operate on that data. The *controller* mediates input, converting it to commands for model or view [26]. The central idea behind MVC is code reusability and separation of concerns. It separates responsibilities into three layers of functionality, as shown on Figure 3.5.

a. *Model* holds real business logic and state, i.e. all rules for getting and updating the state. It has means to interact with persistent storage to retrieve, add, and update data.
b. *View* is responsible for displaying data from Model to user. This layer also sends user data to Controller. This means that both request and response are in domain of View.
c. *Controller* handles all requests from the user and selects the view to return. When receiving a request, the Controller forwards the request to the appropriate handler that interprets what action to take based on the request. Controller calls on Model to perform the desired function, then it selects the View to send back to the user based on the state of the Model's data.

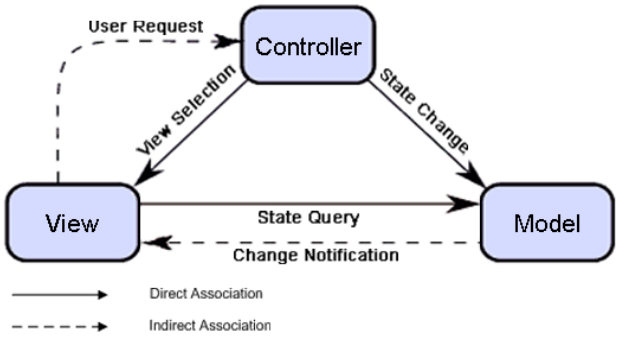


Figure 3.5 A typical collaboration of MVC components

E. Struts

Struts structures all components of Java-based web application into a unified whole to make the application programming more systematic and consistent [27]. Struts is built around the Model-View-Controller (MVC) design pattern that promotes code reusability and separation of concerns in order to make the source codes maintainable. Figure 3.6 illustrates how MVC maps into Struts. In addition, Struts provides much of the infrastructure the programmer needs to create webapps. This makes Struts relatively easy to pick up and use effectively [28].

In Struts, *Model* code consists of plain old Java objects (POJOs), meaning it has no restrictions on how the programmer creates the Model portion of webapp [28]. Nevertheles, the best practice is to encapsulate data access/persistence into Model classes. The data may reside in the same class or in different classes, depending on the application [26]. *View* code consists of JSPs and a set of custom tags supplied by Struts. These custom tags enable the View to be separated from Controller. *Controller* code falls into three broad categories [28]:

a. *Simple validations* are performed by subclasses of Struts base class called `ActionForm`. This does not require complex processing or business logic, e.g. checking of password length and email address format.
b. *Complex validations and business logic* are done in subclasses of Struts base class called `Action`. This is dependent on processing of business logic, e.g. checking for duplicate user ID (complex validation), and calculating the total amount of due after a purchase in a shopping cart (business logic).
c. *Flow control* is also decided by the `Action` subclasses, restricted to paths declared in Struts configuration file called *struts-config.xml*.

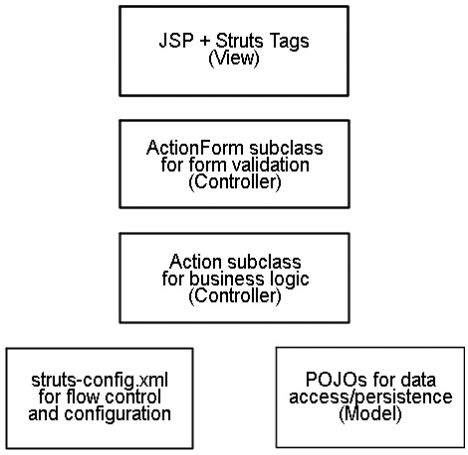


Figure 3.6 How MVC maps into Struts

Since Struts is built on servlet technology, all submissions of form data to a Struts application are intercepted by a single "master" servlet, an instance of `ActionServlet`. This master servlet is responsible for delegating the actual work of the application to subclasses of `ActionForm` and `Action` [28]. The incoming form data is processed in stages:

a. *Data transfer*: Form data is transferred to `ActionForm` subclass.
b. *Simple validation*: Form data is passed through simple validation. If simple validation fails, then error messages detailing the errors will be displayed.
c. *Further processing*: Form data is then sent to `Action` subclass for complex validation and the processing of business logic. The `Action` subclass also specifies the "next" page to be displayed.

Simple Validation

The central class for performing simple validation is

`org.apache.struts.action.ActionForm`

Struts requires each page's form to be associated with subclass of `ActionForm` that performs two things [28]:

a. *It holds all form data*: There must have getters and setters corresponding to each property of the form.

b. *It performs simple validation*: The subclass must override the `validate()` function when performing any simple check.

Listing 3.1 shows the implementation of `ActionForm` subclass corresponding to the Registration webapp form. Note that `ActionErrors` is like a `HashMap` that store error messages, indexed by error keys. The error message stored in `ActionErrors` is not a `String`, but an instance of `ActionMessage` [28].

Listing 3.1 RegistrationForm.java

```
import javax.servlet.http.*;
import org.apache.struts.action.*;

public final class RegistrationForm extends ActionForm{

   private String userID = null;
   private String password = null;
   private String password2 = null;
   /* getXXX and setXXX functions corresponding to form properties */
   public String getUserID(){
      return userID;
   }
   public void setUserID(String userID){
      this.userID = userID;
   }
   public String getPassword(){
      return password;
   }
   public void setPassword(String password){
      this.password = password;
   }
   public String getPassword2(){
      return password2;
   }
   public void setPassword2(String password2){
      this.password2 = password2;
   }
```

```
   /* Validate the user input. Called automatically by Struts framework. */
   public ActionErrors validate(ActionMapping mapping, HttpServletRequest
request){
       //create a blank ActionErrors
       ActionErrors errors = new ActionErrors();
       //check for a blank user ID
       if( null == userID ){
          errors.add("userID",new ActionMessage("error.userid.missing"));
       }
       //check password 1 == password 2
       if( !password.equals(password2)){
           errors.add("password",new ActionMessage("error.password.mismatch"));
       }
       return errors;
   }
}
```

The constructor for `ActionMessage` might seem a little cryptic

`new ActionMessage(“error.password.mismatch”)`

because a *key* is passed instead to an error message. This key is stored in a *properties file* accessible to Struts. Properties files are just text files, which contain key/value pairs. These files are essential for easy internationalization or localization of the application [28]. For example, if the client wants a French version of Registration webapp, the programmer will just create a translated version of *registration.properties*, save it as *registration_fr.properties*, and bundle it with the webapp. Listing 3.2 shows the possible properties file for the Registration webapp.

Listing 3.2 registration.properties

```
# Error messages:
error.userid.missing=The user id is missing.
error.userid.exists = The user id exists. Choose another.
error.userid.bad = Use only alphanumerics for the userid.
error.password.mismatch = The passwords you keyed in don't match!
error.password.long = The password is too long!
error.password.short = The password is too short!

# Prompts:
prompt.userid=User ID
prompt.password=Password
prompt.password.confirmation=Password Confirmation
```

The key under which the `ActionMessage` is stored on the `ActionErrors` instance corresponds to the name of the form property associated with the error for convention. Struts uses `<html:errors>` tag with an attribute called `property` that matches with the error key. This tag is placed on the JSP containing the form. Figure 3.11 summarizes the relationship between `ActionErrors`, `ActionMessage`, properties files, error keys, and error tag.

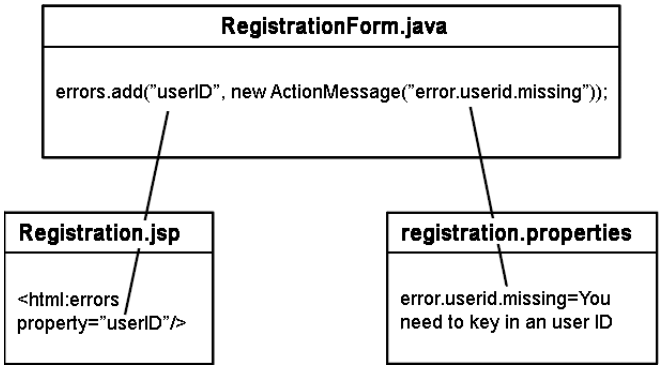


Figure 3.11 How `ActionErrors` ties error messages to the View component

Processing Business Logic

The processing of business logic is done by the subclass of

`org.apache.struts.action.Action`

The `Action` subclass is the Controller because it communicates with both Model and View components of the webapp. There is one method that needs to be overridden in `Action` subclass, and that is `execute()` function [28].

Listing 3.3 shows the `Action` subclass called *RegistrationAction.java*. The way the code is written on Listing 3.3 is the pattern for every `Action` subclass the programmer will implement. Recall that `ActionForm` passed into `execute()` is really a `RegistrationForm` instance (see Listing 3.1), and it contains form data that has passed

simple validation. The cast is done so that the `userID` and `password` properties of the form can be read.

Listing 3.3 RegistrationAction.java

```
import javax.servlet.http.*;
import org.apache.struts.action.*;
import org.registration.User;

public class RegistrationAction extends Action{

   public ActionForward execute(ActionMapping mapping,
                                ActionForm form,
                                HttpServletRequest request,
                                   HttpServletResponse response){
      //get userID and password
      RegistrationForm rForm = (RegistrationForm) form;
      String userID = rForm.getUserID();
      String password = rForm.getPassword();
      //complex validation: check if userID exists

      if(User.exists(userID)){
         ActionMessages errors = new ActionMessages();
         errors.add("userID", new ActionMessage("error.userid.exists"));
         saveErrors(request, errors);
         //navigation: redisplay the user form
         return mapping.getInputForward();
      }else{
         //data transformation: save the userID and password to database:
         User user = new User();
         user.setUserID(userID);
         user.setPassword(password);
         user.save();
         //navigation: display "you're registered" page
         return mapping.findForward("success");
      }
   }
}
```

The following are the three broad tasks every `Action` subclass performs: complex validation, data transformation, and navigation [28].

a. Complex Validation

The relevant section performing complex validation is shown on Listing 3.4. The `User.exists()` function, that checks if the given user ID exists, is the typical “well-

defined" interfaces between Controller (`RegistrationAction`) and Model (`User` class). Having model class handles data access makes the code cleaner, much more maintainable, and more future-proof.

Listing 3.4 Complex validation in RegistrationAction.java

```
if(User.exists(userID)){

   ActionMessages errors = new ActionMessages();
   errors.add("userID", new ActionMessage("error.userid.exists"));
   saveErrors(request, errors);

   //navigation: redisplay the user form
   return mapping.getInputForward();

}else{ ...
```

b. Data Transformation

The only data transformation involved in Listing 3.3 is saving user ID and password to the database, as Listing 3.5 illustrates. Model classes are not part of Struts because this is a task best left to dedicated persistence frameworks like Hibernate or Torque.

Listing 3.5 Data Transformation in RegistrationAction.java

```
User user = new User();
user.setUserID(userID);
user.setPassword(password);
user.save();
```

c. Navigation

Navigation refers to programming logic deciding what page to display next to the user. There are two possible "next" pages in Listing 3.3: the page containing the submitted form (with error messages) and the page indicating successful registration. `ActionForward` is a reference to "next" page. Listing 3.6 shows the navigation logic in *RegistrationAction.java*.

Listing 3.6 Navigation in RegistrationAction.java

```
if(User.exists(userID)){
   ...
   //navigation: redisplay the user form
   return mapping.getInputForward();
}else{
   ...
   //navigation: display "you're registered" page
   return mapping.findForward("success");
}
```

There are two important ways of instantiating `ActionForward`. First is the use of `getInputForward()` function on `ActionMapping` instance to get reference to the input page. The other one is to instantiate `ActionForward` pointing to a named "next" page. The

`mapping.findForward("success")`

is the indirect way of referring to a page (see Listing 3.11). There are two reasons why using indirect reference is encouraged:

a. It makes the webapp more maintainable. The named pages are specified in *struts-config.xml*. Once the page is changed, this one file is the only one need to be amended.
b. It promotes code reuse. This is the case when `Action` is reused in different contexts.

Struts Configuration

The *struts-config.xml* file contains several sections, each of which handles configuration for specific portions on Struts. Listing 3.7 shows the *struts-config.xml* for Registration webapp.

Listing 3.7 struts-config.xml for the Registration webapp

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE struts-config PUBLIC
   "-//Apache Software Foundation//DTD Struts Configuration 1.1//EN"
   "http://jakarta.apache.org/struts/dtds/struts-config_1_1.dtd">

<struts-config>
```

```
   <form-beans>
      <form-bean
         name="RegistrationForm"
         type="org.registration.RegistrationForm"/>
   </form-beans>

   <global-exceptions>
      <exception key="reg.error.io-unknown"
         type="java.io.IOException"
         handler="org.registration.ErrorHandler"/>
      <exception key="reg.error.unknown"
         type="java.lang.Exception"
         path="/errors.jsp" />
   </global-exceptions>

   <global-forwards>
      <forward name="ioError" path="/errors.jsp"/>
   </global-forwards>

   <action-mappings>
      <action
         path="/Registration"
         type="org.registration.RegistrationAction"
         name="RegistrationForm"
         scope="request"
         validate="true"
         input="/Registration.jsp">
         <forward name="success" path="/RegistrationSuccess.jsp"/>
      </action>
   </action-mappings>

   <message-resources parameter="registration"/>

</struts-config>
```

Following are the seven sections that are commonly configured in Struts. Form bean, form handler (action mapping), and message resources are the most important sections that the programmer needs to master before implementing Struts [28].

a. Form Bean Declarations

This is the section where `ActionForm` subclass is mapped to a name. This name acts as an alias use for both *struts-config.xml file* and on JSP pages. The declaration consists of a single enclosing `<form-beans>` tag, and one or more `<form-bean>` tags, as shown on Listing 3.8.

Listing 3.8 The Form Beans Section

```
<form-beans>
   <form-bean
      name="RegistrationForm"
      type="org.registration.RegistrationForm"/>
</form-beans>
```

b. Global Exceptions

Global Exceptions defines handlers for exceptions thrown during processing. This catches the uncaught runtime exceptions that occur in `Action` subclasses, displaying the errors with a custom error message. Listing 3.9 shows two ways to define an exception handler. The `<global-exceptions>` tag contains the global exception handlers, each represented by an `<exception>` tag.

Listing 3.9 Declaring Global Exception Handlers

```
<global-exceptions>
   <exception key="reg.error.io-unknown"
      type="java.io.IOException"
      handler="org.registration.IOErrorHandler"/>
   <exception key="reg.error.unknown"
      type="java.lang.Exception"
      path="/errors.jsp"/>
</global-exceptions>
```

c. Global Forwards

Global forward is used to define forwarding paths accessible to all `Action`s or `ExceptionsHandler`s. It is declared within an enclosing `<global-forwards>` tag, within which one or more `<forward>` tags can be placed, as shown on Listing 3.10.

Listing 3.10 Declaring a Global Forward

```
<global-forwards>
   <forward name="ioError" path="/errors.jsp"/>
</global-forwards>
```

d. Form Handlers

Form handlers are defined within a single `<action-mappings>` enclosing tag, as shown on Listing 3.11. The `<action-mappings>` tag acts as a container for each form handler, described by an `<action>` tag. Each `<action>` tag can contain zero or more `<forward>` tags. Unlike the global forward, this forward is visible only to the enclosing form handler. `<forward>`s represent the possible "next" pages for that form handler.

Listing 3.11 The Form Handler Declaration

```
<action-mappings>
   <action
      path="/Registration"
      type="org.registration.RegistrationAction"
      name="RegistrationForm"
      scope="request"
      validate="true"
      input="/Registration.jsp">
      <forward name="success" path="/RegistrationSuccess.jsp"/>
   </action>
</action-mappings>
```

e. Controller Declarations

This section is used to manually override some default Struts settings. Some of which are listed below:

- `maxFileSize`: Specifies the upper size limit on uploaded files. K, M, or G is used after the number to indicate the size in kilobytes, megabytes, or gigabytes respectively.
- `nocache`: Tells Struts whether it should cache content. Setting `nocache="true"` disables content caching.
- `contentType`: Specifies the default content type of pages.

f. Message Resources

Message resources tells Struts the location of the properties files that contain prompts and error messages. The `<message-resources>` tag does not have an enclosing tag

```
<message-resources parameter="registration"/>
```

Its main attribute is the `parameter` attribute, which gives the location of properties file relative to *\WEB-INF\classes* directory of the webapp. Note that *.properties* extension is implied. If *registration.properties* file is placed in *\WEB-INF\classes\org\registration\resources* the parameter attribute value would be

```
org.registration.resources.registration
```

g. Plug-in Declarations

This is section tells Struts what plug-ins to initialize and what data they require (usually paths to various configuration files required by the plug-in). *Tiles* and *Validator* are the two important plug-ins used in webapp. Listing 3.12 shows a typical plug-in declaration.

Listing 3.12 A Possible Plug-in Declaration for Tiles

```
<plug-in className="org.apache.struts.tiles.TilesPlugin" >
    <set-property property="definitions-config"
       value="/WEB-INF/tiles-defs.xml"/>
</plug-in>
```

Each `<plug-in>` tag declares a single plug-in. There is no limit to how many plug-ins needs to include in the *struts-config.xml* file. The `className` attribute is required, and it points to the plug-in class that Struts calls. This class is unique to each plug-in. In addition, each `<plug-in>` tag may contain zero or more `<set-property>` tags to set the various properties needed by the plug-in.

Basic Struts Tags

Struts implements the View component of MVC design pattern entirely through the use of custom tags. These tags are applied to the JSPs that constitute the View component of the webapp. The Struts tags are bundled into five tag libraries [28]:

a. *HTML*: The custom tags in the HTML library are essentially in one-to-one-relationship to the ordinary HTML `<form>` tag and its associated input tags. The purpose of this tag library is to connect the View component to the Controller components.
b. *Bean*: This library has custom tags primarily for writing text. There are two reasons why Bean tags are used instead of hard-coding the text into the JSPs. First is to enable internationalization, that is, to display View component in multiple languages. Second reason is to avoid using scriptlets to display objects stored on request or session objects.
c. *Logic*: This library provides tags for conditional processing and looping, in lieu of using scriptlets.
d. *Nested*: This library has tags for displaying "nested" properties of a form or object.
e. *Tiles*: This library contains tags that allow the programmer to create layouts.

Custom tags have a *prefix* defined on the JSP page and a *tag name*, which is fixed in tag's **TLD** (**T**ag **L**ibrary **D**escriptor) file. Before they can be used on JSP, a tag has to be declared at the start of page as shown on Listing 3.14. TLD files and the Java handlers are all bundled in the Struts distribution. Several tags may be collected together into a TLD file [28]:

- The TLD files should be placed in \WEB-INF\.

- Struts comes with *web.xml* file that contains the appropriate `<taglib>` sections, as shown on Listing 3.13.
- The Java handler classes for these tags are in various JAR files that comprise the Struts binaries, which should be placed in the webapp's \lib\ directory.

Listing 3.13 A Taglib section in web.xml

```
<taglib>
   <taglib-uri>/tags/struts-bean</taglib-uri>
   <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
```

Listing 3.14 shows the View component of Registration webapp consisting of a single page *Registration.jsp*.

Listing 3.14 Registration.jsp

```
<%@ page contentType="text/html;charset=UTF-8" %>
<%@ taglib uri="/tags/struts-bean" prefix="bean" %>
<%@ taglib uri="/tags/struts-html" prefix="html" %>
<html:html>
   <head>
      <title><bean:message key="registration.title"/></title>
   </head>
   <body>
      <h1><bean:message key="registration.heading"/></h1>
      <html:form action="Registration.do" focus="userID">
         <p>
            <bean:message key="registration.prompt.userid"/>
            <html:text property="userID" size="20" />
            <html:errors property="userID" />
         </p>
         <p>
            <bean:message key="registration.prompt.password"/>
            <html:password property="password" size="20" />
            <html:errors property="password" />
         </p>
         <html:submit>
            <bean:message key="registration.prompt.submit"/>
         </html:submit>
         <html:reset>
            <bean:message key="registration.prompt.reset"/>
         </html:reset>
      </html:form>
   </body>
</html:html>
```

The `<bean:message>` tags are used to display static text. Just like `ActionMessage`, the key attribute on this tag points to a key-value pair on the *properties files*. On the other hand, forms are declared within an `<html:form>` tag, which has the `action` attribute that describes the handlers that processes the form data. A handler is a particular combination of `ActionForm` subclass that performs simple validation and `Action` subclass that does business logic, as shown on Listing 3.11. Traditionally, handlers end with `.do` prefix that tells the servlet container that the `action` is to be handled by Struts. This mapping is done in *web.xml* as shown on Listing 3.15. This also shows the main Struts servlet class declaration and the reference to the Struts configuration file (*struts-config.xml*).

Listing 3.15 Struts' standard servlet and servlet mapping declarations

```
<!-- Standard Action Servlet Configuration -->
<servlet>
   <servlet-name>action</servlet-name>
   <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
   <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
   </init-param>
   <load-on-startup>2</load-on-startup>
</servlet>

<!-- Standard Action Servlet Mapping -->
<servlet-mapping>
   <servlet-name>action</servlet-name>
   <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

Struts tags that accept data input must be placed within the `<html:form>` tag. The textual field tags, e.g. `<html:text>` and `<html:password>` all have a property attribute that ties them to the field on the form. Each field must correspond to a `getXXX()` and `setXXX()` pair on the `ActionForm` subclass associated with the form handler.

IV. DESIGN AND IMPLEMENTATION

A. Debugging Plan

Debugging, the process of locating and fixing or bypassing bugs or errors in computer program code, plays an important part of the study. Struts framework cannot be easily implemented in VirHoLex unless the bugs or errors are handled and fixed. The system is debugged by manually checking the output and/or the source codes with the help of Eclipse debugging tool. For efficiency, each module is debugged separately at first, followed by the program as a whole.

For proper documentation of identified bugs, the format shown on Table 4.1 is employed. It follows the default format on Mantis Bug Tracking system including a new field Version Number. The Bug Tracking system contains bugs that are encountered only on the initial version of the system, i.e. before adding the framework.

Mantis ID Automatically generated by Mantis. This ID is used to easily track the files and/or directories committed in Agila SVN
Module (Category) The module or sub-module where the bug (error) is found • Experiments • Images • Models • References • Hotspots • External Search • Internal Search • Registered User Services • User Interface
Severity Nature of the problem • block - prevents further work/progress from being made

• crash - crashes the application or OS • major - major bug • minor - minor bug • tweak - needs tweaking • text - error in the text • trivial - being nitpicky • feature - requesting new feature
Problem (Summary) Title or summary of the problem
Description Detailed description of the problem
Version Number Version of the system after the bug is fixed. The versioning format is as follow **Version Number X.Y.Z** where • **X** – version number of the previous system, i.e. 1 • **Y** – bugs or errors (block, crash, major, minor) • **Z** – layout or text issue (tweak, text, trivial)
Attached Files Screenshots of the bug and its fix

Table 4.1 Bug Documentation Format

B. Subversion Plan

There is a one-to-many relationship between the bug written on Mantis Bug Tracking system and the batch of files (containing fix for the bug) committed on Agila SVN, that is, the programmer can commit more than once using same Mantis ID. A single or a batch of newly added, modified or deleted files can be committed into the repository. Table 4.2 shows the details that are included when committing files. This format is followed only if the issue was reported on Mantis Bug Tracking system; otherwise a brief description of the problem is sufficed so that future VirHoLex developers can easily know the modifications done on those files.

On the other hand, Eclipse SVN Repository plug-in is used as a tool to check in and checkout files to and from the SVN so that the working copy can be easily synchronized with the copy in the repository. As stated before, Morales source codes are imported into Agila repository and serve as the initial version of VirHoLex.

Mantis ID Automatically generated by Mantis. This ID is used to easily track the files committed in SVN.
Version Number Version of the system after the bug is fixed (See *How to create version number* on Table 4.1.1 under Version Number).
Problem (Summary) Title or summary of the problem
Description The detailed description of the problem
Files/Directories List of files or directories that are committed on Agila SVN

Table 4.2 Details included when committing files on SVN

C. Deployment Plan

The deployment diagram of VirHoLex 2.0 on localhost server is shown on Figure 4.1. The Operating System (OS) may vary depending on the computer machine available. For this case, the system is operated on Windows 7 Professional (32-bit) machine. Tomcat 6.0.35 and MySQL 5.5.16 are used to run the source codes. After debugging and refactoring the whole system, VirHoLex 2.0 is deployed on Agila server. The system is then tested if it runs properly on the said server without any show-stopping bugs. Subsequently, VirHoLex 2.0 is deployed on MaxPlanck server ang it tested as well. Figure 4.2 and Figure 4.3 show the deployment diagrams for Agila and MaxPlanck servers respectively.

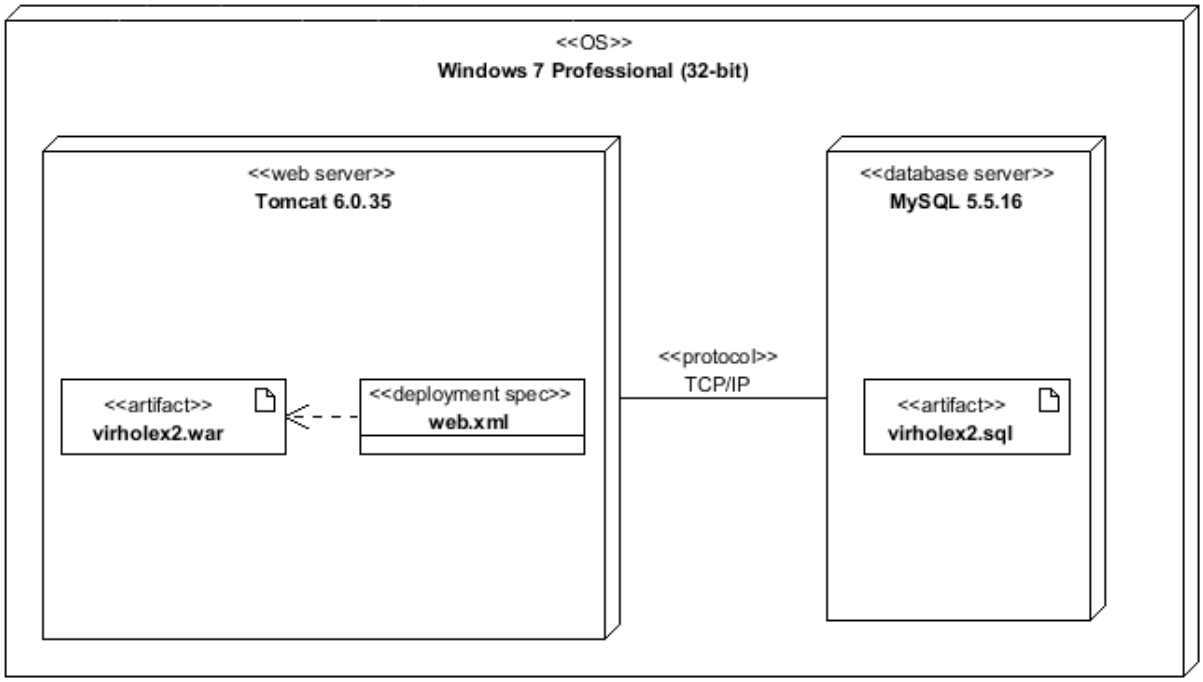


Figure 4.1 Deployment Diagram of VirHoLex 2.0 on localhost server

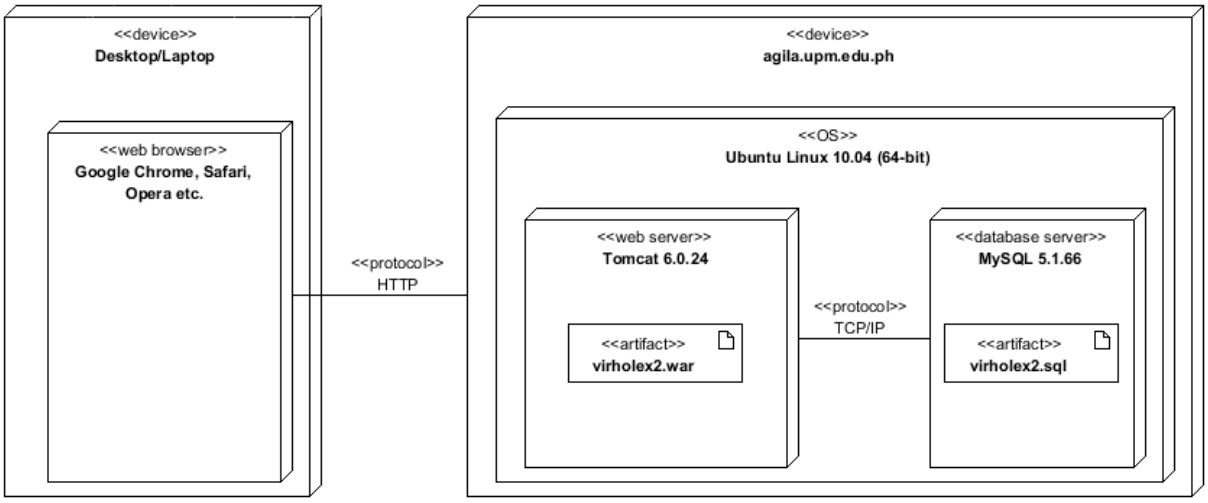


Figure 4.2 Deployment Diagram of VirHoLex 2.0 on Agila server

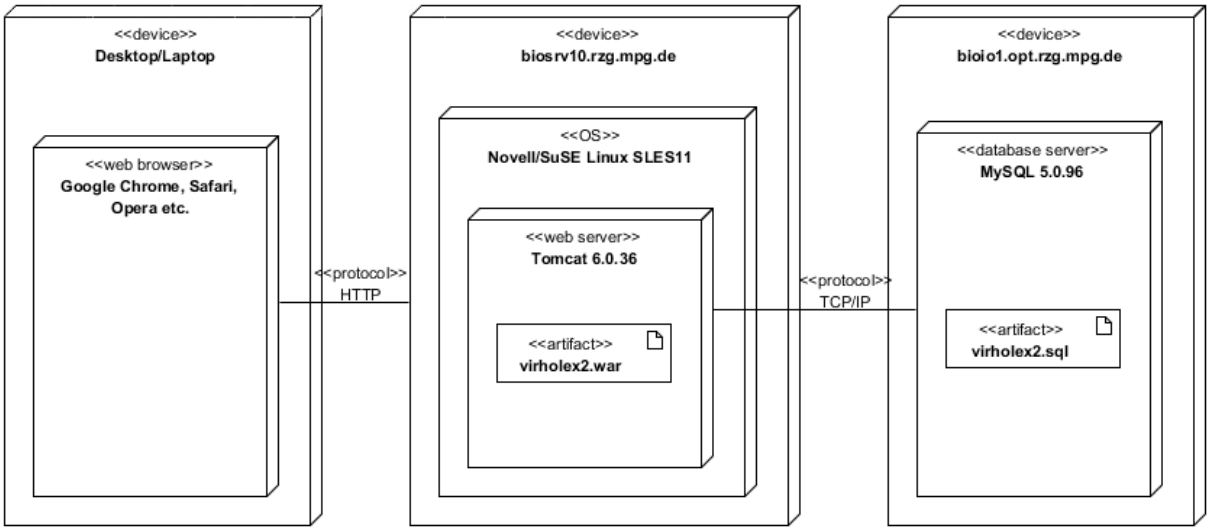


Figure 4.3 Deployment Diagram of VirHoLex 2.0 on MaxPlanck server

D. Sequence Diagram

Figure 4.4 shows the sequence diagram for Login application which employs Struts framework. This gives the overview of how Struts works in the system. The following is the message description for the given sequence diagram.

- 1 The client submits the *loginAccount.jsp*'s form, `doPost` method is called to allow `HtppServlet ActionServlet` to handle the POST request
- 1.1 The Struts Controller `ActionServlet` delegates to `RequestProcessor` the process of the request
- `RequestProcessor` does the following actions :
    - 1.1.1 retrieve and return the `LoginAccountForm` bean associated with the mapping, creating one if necessary
    - 1.1.2 populates `LoginAccountForm` bean with input fields of the *loginAccount.jsp*'s form

- 1.1.3 validates input field values and creates error messages if validation errors
- 1.1.4 acquires an `LoginAccountAction` instance to process the request, calling the overwritten execute method of `LoginAccountAction`
  - 1.1.4.1 retrieves data from the `LoginAccountForm` bean by `getProperties` methods
  - 1.1.4.2 calls persistent storage through the `LoginAccountDAO`
  - 1.1.4.3 forwards to the specified destination in *struts-config.xml*

- 3 The forwarded page retrieves data from the `ActionForm` bean

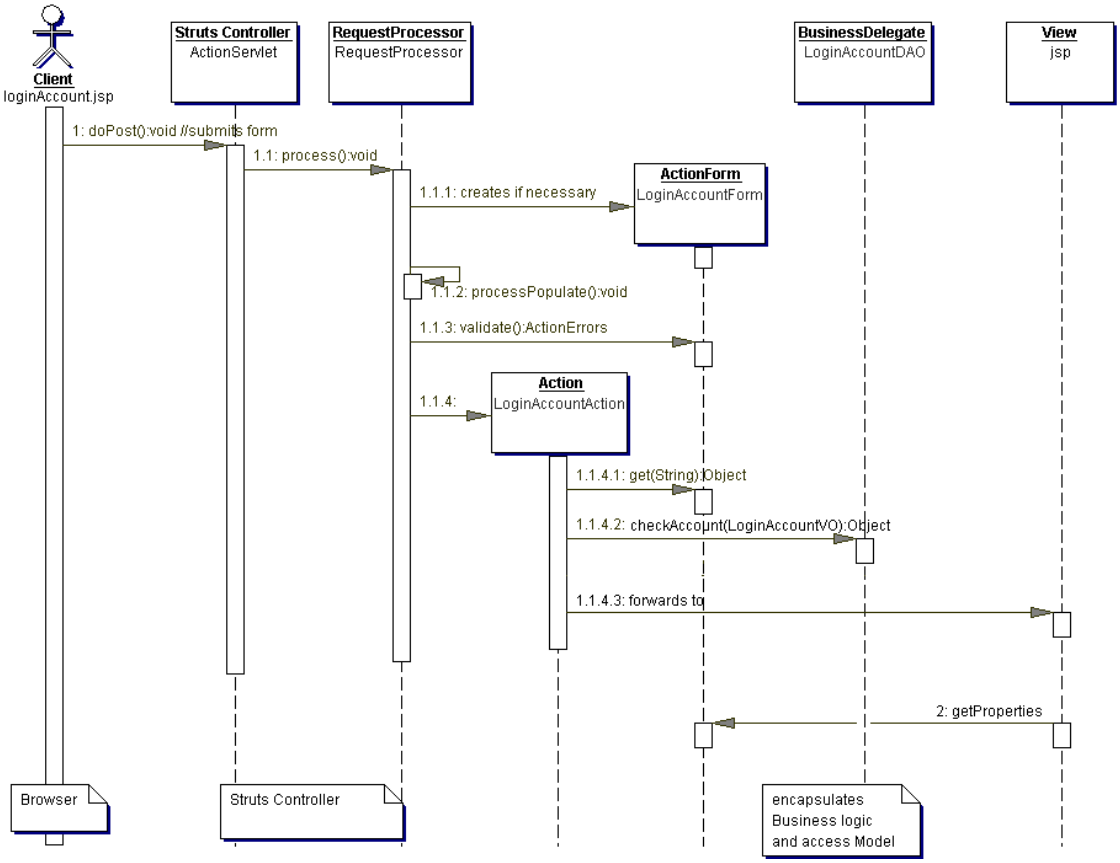


Figure 4.4 Sequence Diagram for Login application

E. Deployment Directory

Figure 4.5 shows the deployment directory of VirHoLex 2.0

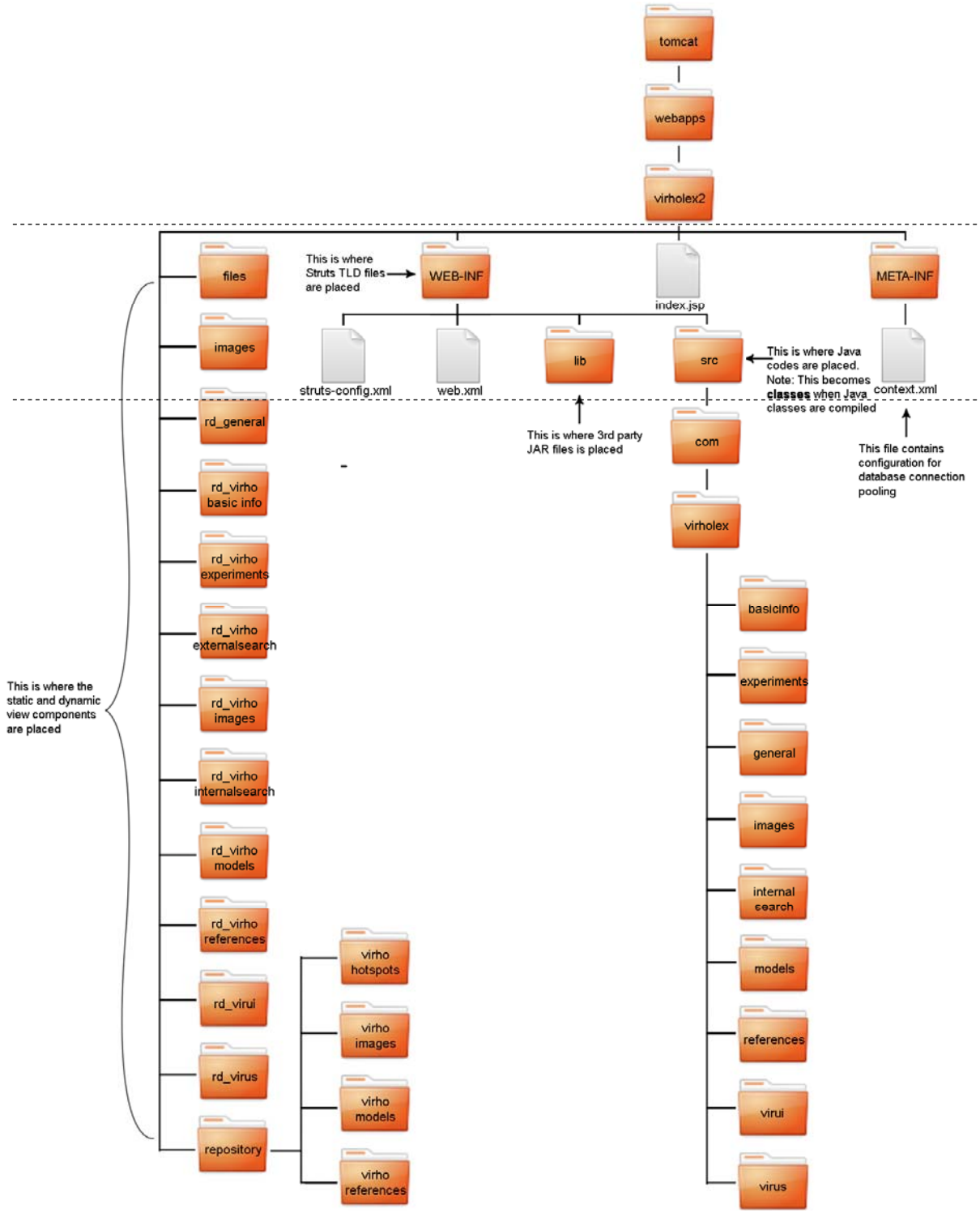


Figure 4.5 VirHoLex 2.0 directory structure

V. RESULTS

VirHoLex source codes were first imported into the Agila SVN repository via Tortoise SVN 1.7.4. As mentioned before, Morales codes serves as the initial version of VirHoLex and became the basis for the succeeding refactoring of the system.

After importing the codes into the repository, the Agila SVN url for VirHoLex[4] was added on Eclipse SVN Repositories view in order to check out the codes that served as the working copy of the developer, as shown on Figure 5.1. Figure 5.2 shows how to check out codes from the Agila SVN using the url repository on Eclipse SVN Repositories view. The working copy was run on apache-tomcat 6.0.35 server using MySQL 5.5.16 as its backend.

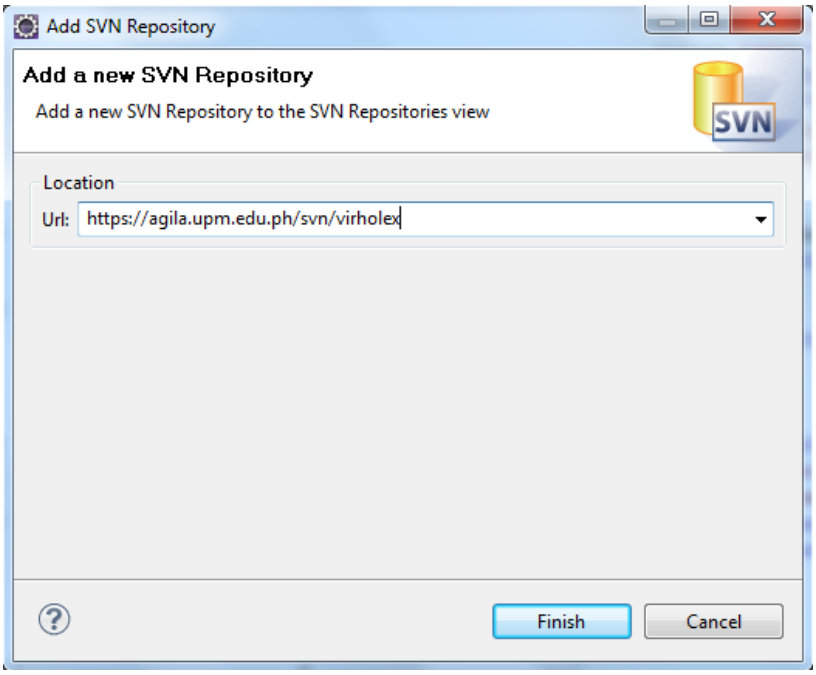


Figure 5.1 Adding VirHoLex SVN url on Eclipse SVN Repository

[4] https://agila.upm.edu.ph/svn/VirHoLex

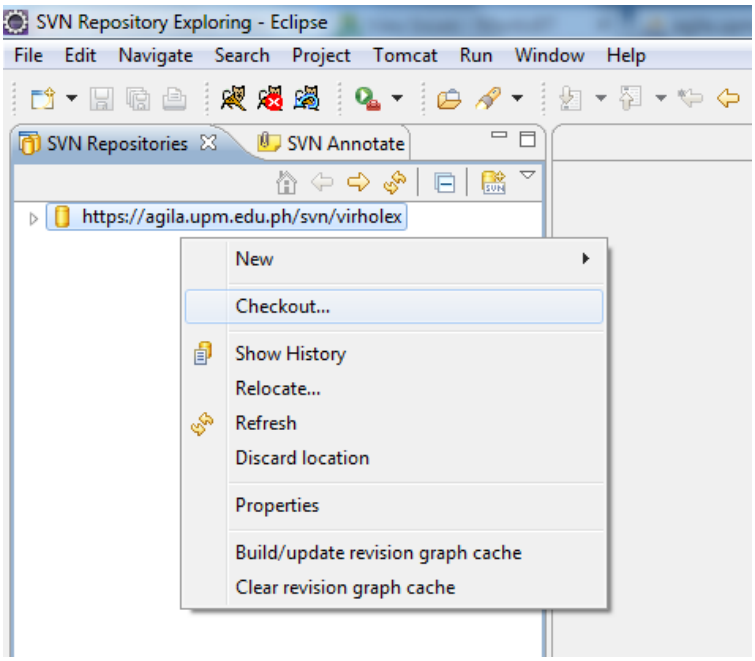
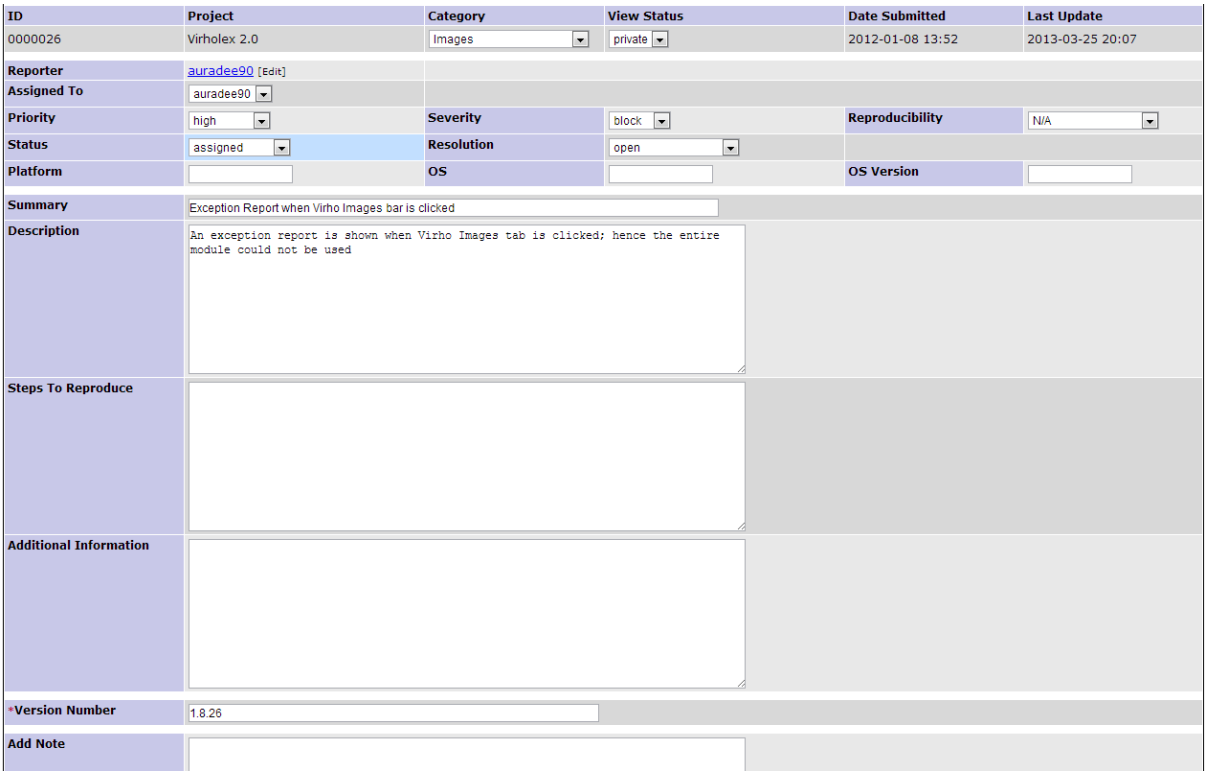


Figure 5.2 Checking out VirHoLex codes from Agila SVN repository via Eclipse SVN tool

Each component of the system, particularly Virho Images and Virho Experiments modules, was thoroughly investigated and tested to check for bugs (or errors). These system bugs were then reported on Mantis Bug Tracking system containing the necessary details and screenshots (if needed) of the problem, as illustrated on Figure 5.3. The status of the bug in Mantis Bug Tracking system is changed to 'Resolved' and the version number is updated once the issue was fixed. Figure 5.4 shows the complete list of bugs encountered from the initial version of the system. The bugs were classified according to the severity of the problem and the module where the bug was found.



ID	Project	Category	View Status	Date Submitted	Last Update
0000026	Virholex 2.0	Images	private	2012-01-08 13:52	2013-03-25 20:07
Reporter	auradee90 [Edit]				
Assigned To	auradee90				
Priority	high	**Severity**	block	**Reproducibility**	N/A
Status	assigned	**Resolution**	open		
Platform		**OS**		**OS Version**	
Summary	Exception Report when Virho Images bar is clicked				
Description	An exception report is shown when Virho Images tab is clicked; hence the entire module could not be used				
Steps To Reproduce					
Additional Information					
***Version Number**	1.8.26				
Add Note					

Figure 5.3 Bug reported on Mantis Bug Tracking system

Viewing Issues (1 - 58 / 58) [Print Reports] [CSV Export] [Excel Export]

P	ID	#	@	Category	Severity	Status	Updated	Summary
–	0000001	1	2	User Interface	feature	resolved (auradee90)	2012-08-08	Modify the GUI of the system
–	0000002	1	4	External Search	minor	resolved (auradee90)	2012-08-08	Modify the External Search Form GUI
–	0000003	1	2	Registered User Services	feature	resolved (auradee90)	2012-08-08	Modify Register Account and Why Should I Register?
⌄	0000004	1	2	Registered User Services	minor	resolved (auradee90)	2012-08-08	Modify Forgot Password
⌄	0000005	1	2	Registered User Services	minor	resolved (auradee90)	2012-08-08	Modify Login link
–	0000006	1	5	Registered User Services	**major**	resolved (auradee90)	2012-08-08	Error in changing the email address and/or password
⌄	0000007	1	2	Registered User Services	feature	resolved (auradee90)	2012-08-08	Modify Unsubsribe
⌄	0000008	1		Registered User Services	feature	resolved (auradee90)	2012-08-08	Modify the Pending Request GUI
–	0000009	1	3	Registered User Services	feature	resolved (auradee90)	2012-08-08	Modify User List
⌄	0000010	2	2	Models	feature	resolved (auradee90)	2012-08-08	Modify the Models Classification, Project with its corresponding Model
⌄	0000011	1	2	Hotspots	feature	resolved (auradee90)	2012-08-08	Modify Virus List Legend
⌄	0000012	2	5	Models	minor	resolved (auradee90)	2012-08-08	Fixed the connection between My Project and Virho Models' Add Users
⌄	0000013	1		Internal Search	minor	resolved (auradee90)	2012-08-08	Modify Internal Search Advanced Button
⌄	0000014	1		Internal Search	feature	resolved (auradee90)	2012-08-08	Modify Internal Search
–	0000015	1	1	References	minor	resolved (auradee90)	2012-08-08	Missing buttons and link when the Collection's name at View Users page is clicked
–	0000016	1	1	References	**major**	resolved (auradee90)	2012-08-08	Missing resource when URL of the Link is clicked
^	0000017	1	2	References	minor	resolved (auradee90)	2012-08-08	Errors on 'Export to Bibtext'
^^	0000018	1	1	References	trivial	resolved (auradee90)	2012-08-08	Modify Add Reference

		–	0000019	1	1	Models	minor	resolved (auradee90)	2012-08-08	Modify Add Model to project
		⌄	0000020	1		Models	feature	resolved (auradee90)	2012-08-08	Modify the Reference linked to Add Model
		⌄	0000021	1		Models	feature	resolved (auradee90)	2012-08-08	Modify the Edit Model
		⌄	0000022	1	1	Hotspots	trivial	resolved (auradee90)	2012-08-08	Modify Add Virus Information
		–	0000023	1	3	Hotspots	**major**	resolved (auradee90)	2012-08-08	Modify the View Virus Information
		^	0000024	1	5	Hotspots	**major**	resolved (auradee90)	2012-08-08	Modify the Add and Edit Virus Information
		⌄	0000025	1		Registered User Services	minor	resolved (auradee90)	2012-08-08	Fix the wrong links in Privilege Request (Virho Models, Virho Hotspots, Virho References)
		^	0000026	1	2	Images	**block**	resolved (auradee90)	**2013-03-25**	Exception Report when Virho Images bar is clicked
		^	0000027	2	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error after clicking the Submit button in Add Image Set
		⌄	0000028	1	1	Images	feature	resolved (auradee90)	2012-08-08	Use images in Add Photo, Edit and Delete
		^	0000029	2	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error after clicking the Submit button in Edit Image Set
		⌄	0000030	1		Images	feature	resolved (auradee90)	2012-08-08	Modify the Upload Images
		^	0000031	1	4	Images	**major**	resolved (auradee90)	2012-08-08	Exception error after clicking the Submit button in View Manager
		^	0000032	1	3	Images	**major**	resolved (auradee90)	2012-08-08	Exception error after clicking the Download Selected or Remove Selected from View in Saved View
		^	0000033	1	4	Images	**major**	resolved (auradee90)	2012-08-08	Exception report after clicking the Save button in the Edit View
		^	0000034	1	3	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when one of the images in the Images Set is clicked
		^	0000035	1	1	Images	**major**	resolved (auradee90)	2012-08-08	Exception report when Slideshow is clicked
		^	0000036	1	3	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when deleting an images set
		^	0000037	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when deleting a particular Saved View
		–	0000038	1	2	Images	minor	resolved (auradee90)	2012-08-08	"Remove Selected from View" is not functioning properly
		^	0000039	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when deleting an image in Single View
		^	0000040	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when the Submit button is clicked in Edit Metadata
		^	0000041	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when the View Members is clicked
		^	0000042	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when the Manage Members link is clicked
		^	0000043	1	2	Images	**major**	resolved (auradee90)	2012-08-08	Exception error when the "Save Changes" button is clicked
		–	0000044	1	1	Images	minor	resolved (auradee90)	2012-08-08	No images are shown on the "Most Recent Images" pane
		^	0000045	1	2	Internal Search	**major**	resolved (auradee90)	2012-08-08	Exception errors when the value of the dropdown box under virho models or virho images module is changed
		–	0000046	1		References	**major**	resolved (auradee90)	2012-08-08	References with no files included are not shown in the exported XML file
		^	0000047	1	3	References	**major**	resolved (auradee90)	2012-08-08	Reference cannot be deleted
		–	0000048	1	3	References	minor	resolved (auradee90)	2012-08-08	Fix some problems in Reference
		–	0000049	1	4	Experiments	**major**	resolved (auradee90)	2012-08-08	Modify the Virho Experiments GUI; Cannot edit project
		^	0000050	1	1	Experiments	**major**	resolved (auradee90)	2012-08-08	Problem when adding experiment
		–	0000051	2	1	Experiments	**major**	resolved (auradee90)	2012-08-08	Delete Project and Experiment
		–	0000052	2		Experiments	**major**	resolved (auradee90)	2012-08-08	No Edit Experiment functionality
		^	0000053	1	4	Experiments	**major**	resolved (auradee90)	2012-08-08	Every registered user can add, modify and delete the project and/or experiment
		⌄	0000054	1	2	Experiments	**major**	resolved (auradee90)	2012-08-08	Experiment Photos (or Image Set)
		–	0000055	1	2	Internal Search	minor	resolved (auradee90)	2012-08-08	Include Experiments in the Internal Search
		^	0000058	1		Experiments	**block**	resolved (auradee90)	2013-03-24	Status report error for Conditions, Variables, and Chemical Activators links
		^^	0000059	1		Experiments	**block**	resolved (auradee90)	2013-03-24	Modifying an experiment
		^^	0000060	1		Experiments	**block**	resolved (auradee90)	2013-03-24	Deleting an experiment

Figure 5.4 List of bugs found on VirHoLex 1.0

Eclipse debugging tool was utilized hand in hand with manual investigation to ease the difficulty of locating and fixing the bugs. After resolving each reported issue, the batch of files (that were added, modified and deleted) that were part of the fix was committed to the

Agila repository. Figure 5.6 shows the commit dialogue box with multiple files being committed. A brief description or comment was necessary to easily understand the modifications made on those files.

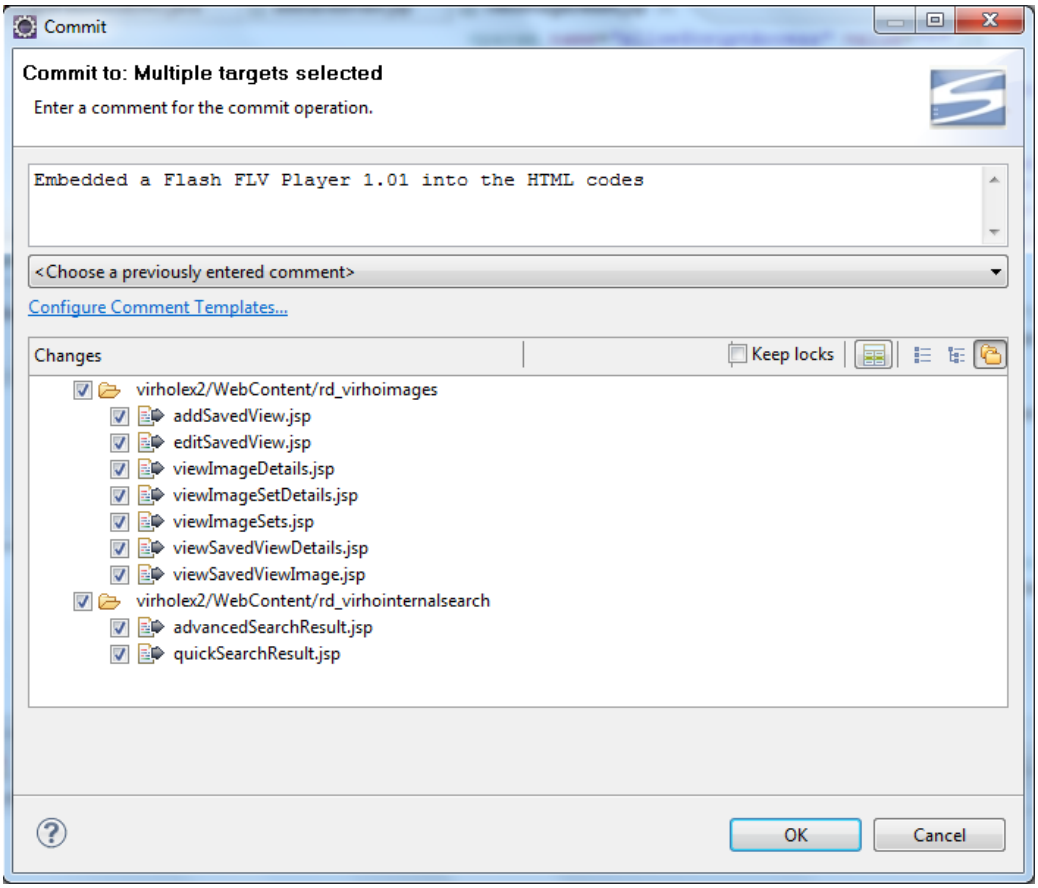


Figure 5.6 Multiple files committed to Agila SVN repository

Struts framework was applied into the VirHoLex source codes once all the bugs listed on the Mantis Bug Tracking system were fixed. A complete code revamp was done to all modules of the system promoting uniformity in the way the codes were written. New set of files was created during the system refactoring, for both .jsp pages and .java files, to avoid confusion with the initial set of codes (see Figure 4.4. for the new directory structure of

VirHoLex). Figure 5.7 illustrates the final architectural diagram of VirHoLex. Virho External Information Services, Virho Basic Virus Info and Virho Hotspot Info are under the Virho Information Services module. On the other hand, Figure 5.8 to Figure 5.12 depict the six modules of the system with their new and uniformed layout on which Struts was applied.

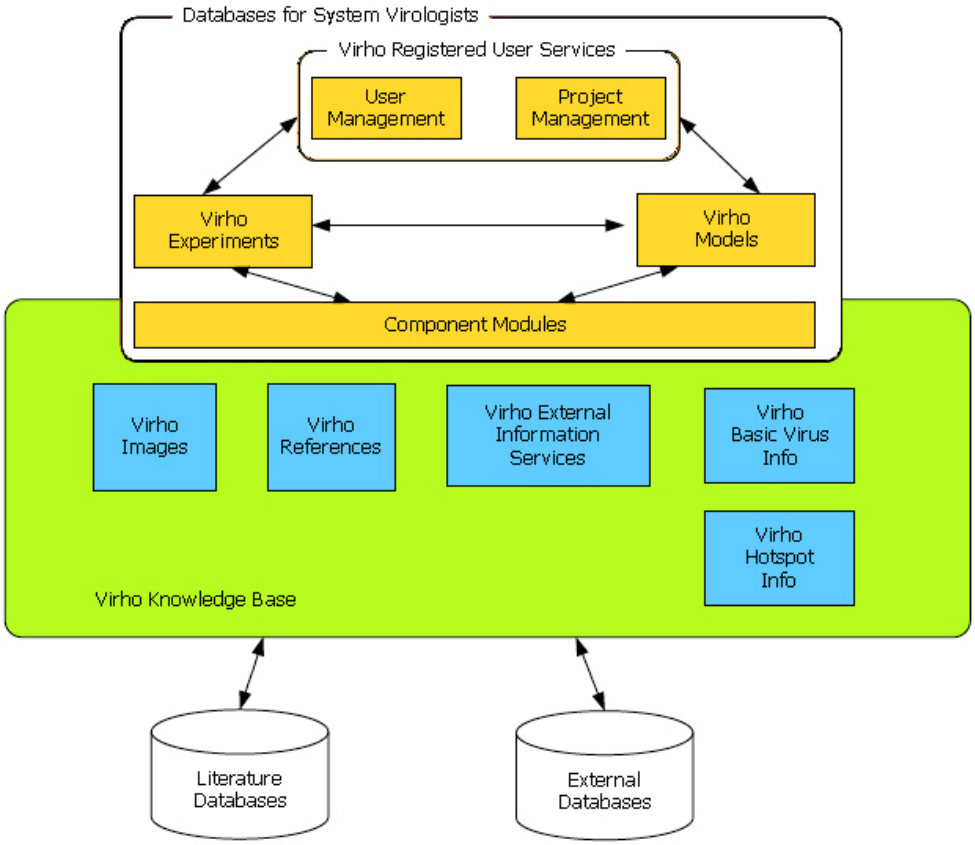


Figure 5.7 VirHoLex Architectural Diagram

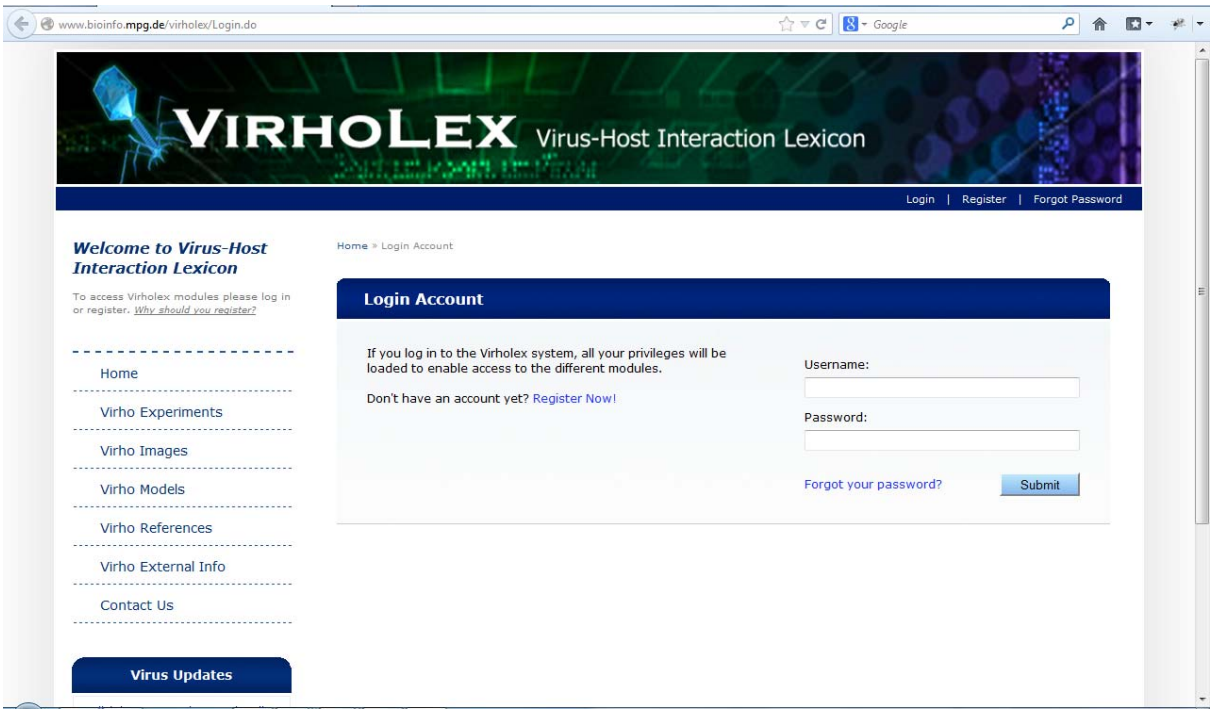


Figure 5.8 Virho Registered User Services - Login page

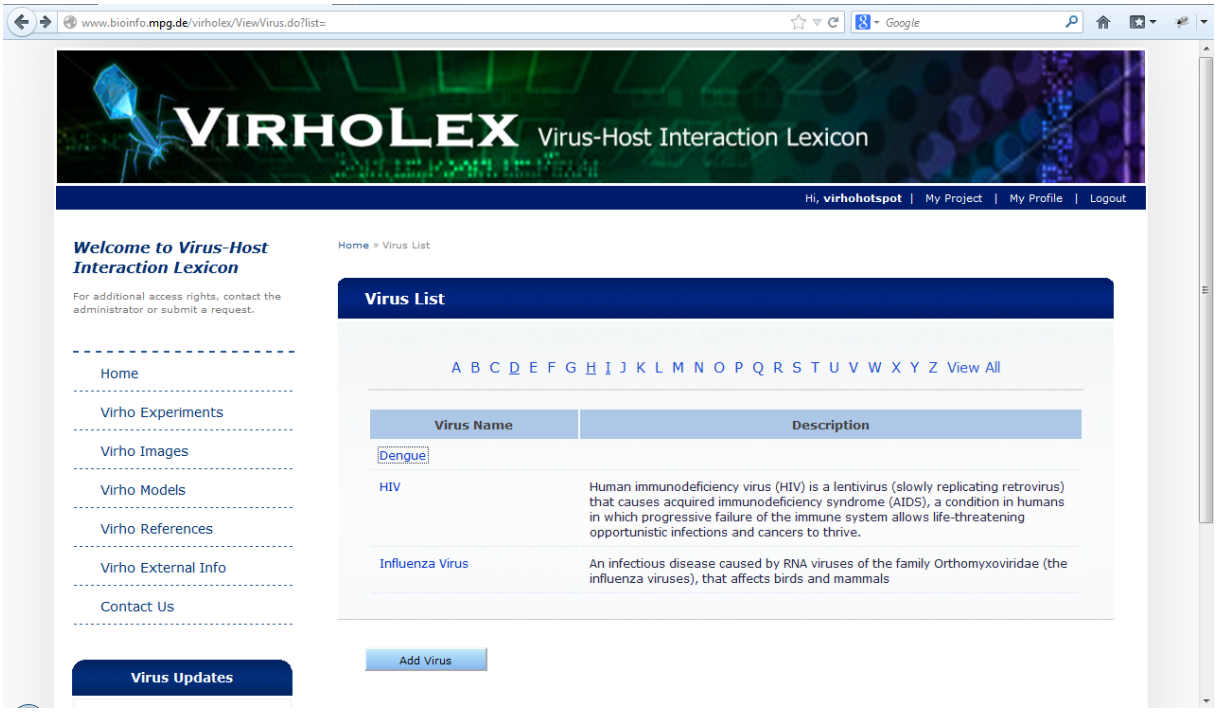


Figure 5.9 Virho Information Services – Virus List page

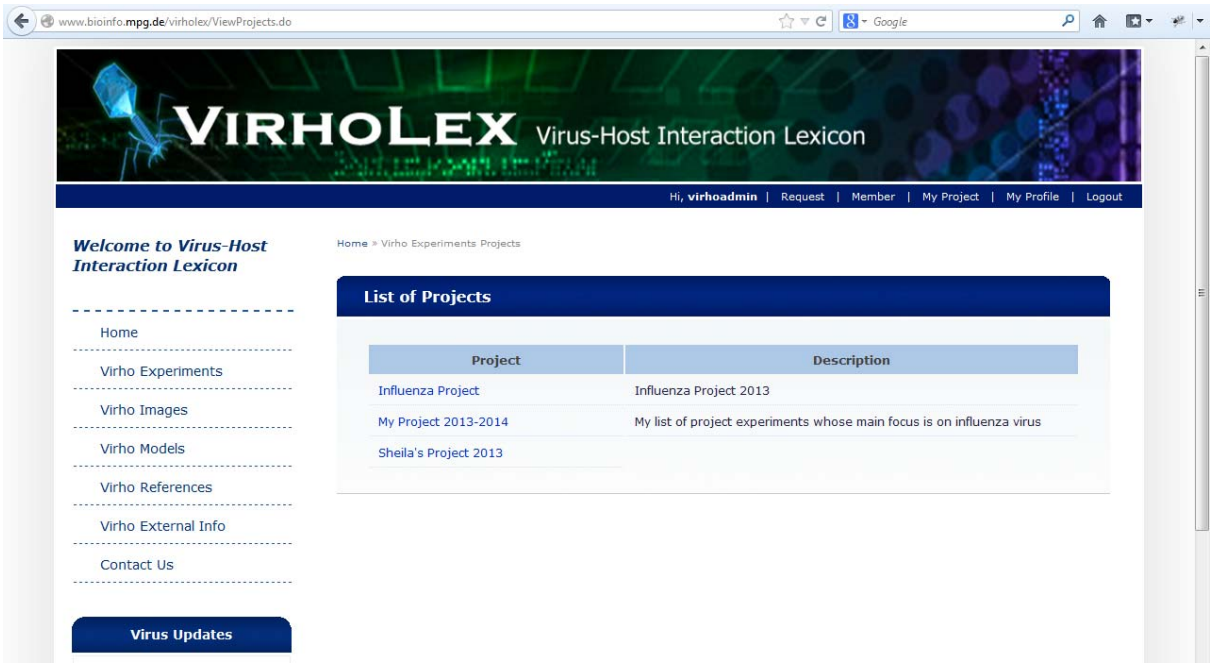


Figure 5.10 Virho Experiments – List of Projects page

Figure 5.11 Virho Images – List of Image Sets page

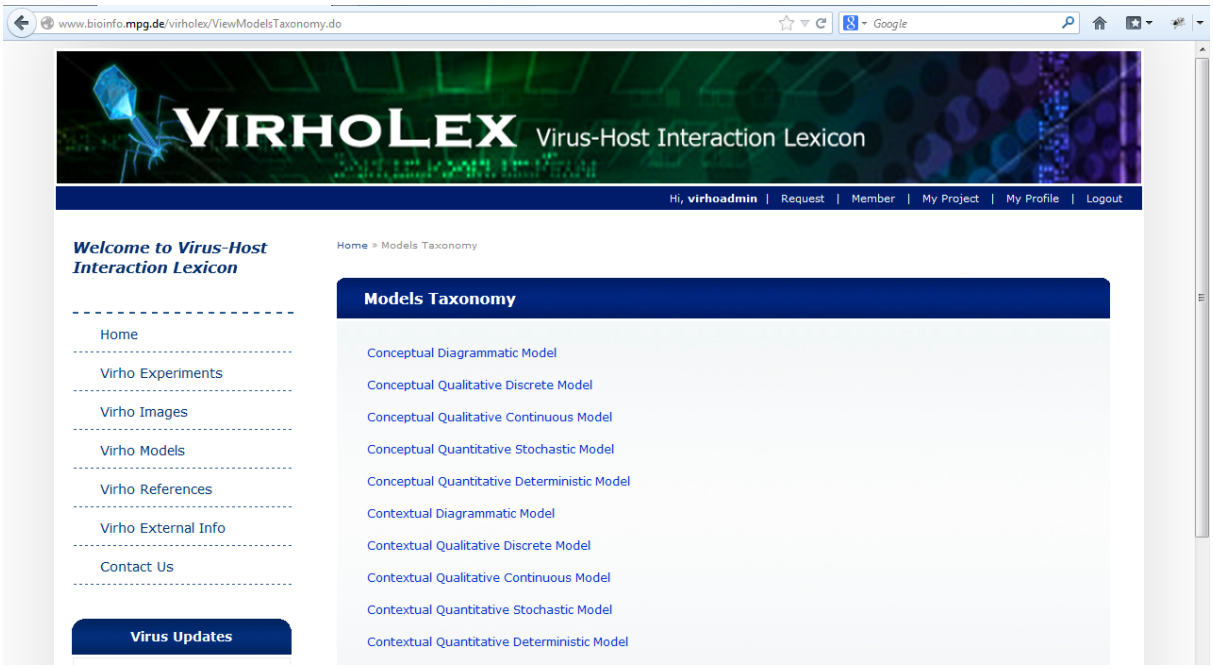


Figure 5.12 Virho Models – Models Taxonomy page

Figure 5.13 Virho References – List of References Collection page

To get start with Struts, necessary jar files were added into the *WEB-INF/lib* directory, as depicted on Figure 5.14.

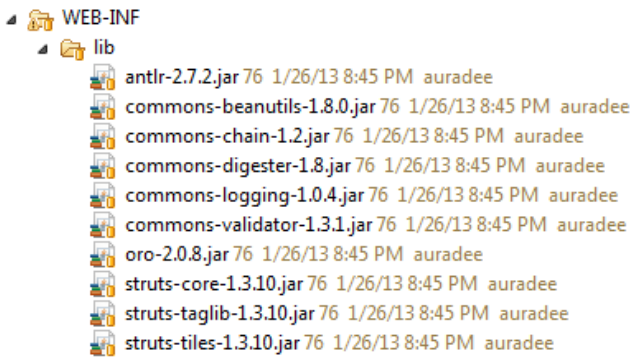


Figure 5.14 Struts jar files

Subsequently, the deployment descriptor (*web.xml*) was configured overwriting the previous set of servlets with a single servlet called `ActionServlet`. This servlet was mapped with a URL pattern `*.do` for it to be invoked when making HTTP requests. The *struts-config.xml*, which handles all Struts configuration, was also declared in the deployment descriptor. Figure 5.15 illustrates the Struts' standard servlet and servlet mapping declarations in the *web.xml*.

```
    <!-- Standard Action Servlet Configuration -->
    <servlet>
        <servlet-name>action</servlet-name>
        <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
        <init-param>
            <param-name>config</param-name>
            <param-value>/WEB-INF/struts-config.xml</param-value>
        </init-param>
        <init-param>
            <param-name>debug</param-name>
            <param-value>2</param-value>
        </init-param>
        <init-param>
            <param-name>detail</param-name>
            <param-value>2</param-value>
        </init-param>
        <load-on-startup>2</load-on-startup>
    </servlet>
    <!-- Standard Action Servlet Mapping -->
    <servlet-mapping>
        <servlet-name>action</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>
```

Figure 5.15 Struts' standard servlet and servlet mapping declarations in *web.xml*

To clearly understand how Struts was incorporated and worked on the VirHoLex system, Login page under Virho Registered User Services module will be used as a sample application to be discussed in details. Login application will be broken into parts to illustrate that the system was built around the MVC design pattern. Figure 5.16 shows the new login form of VirHoLex 2.0 after applying Struts framework.

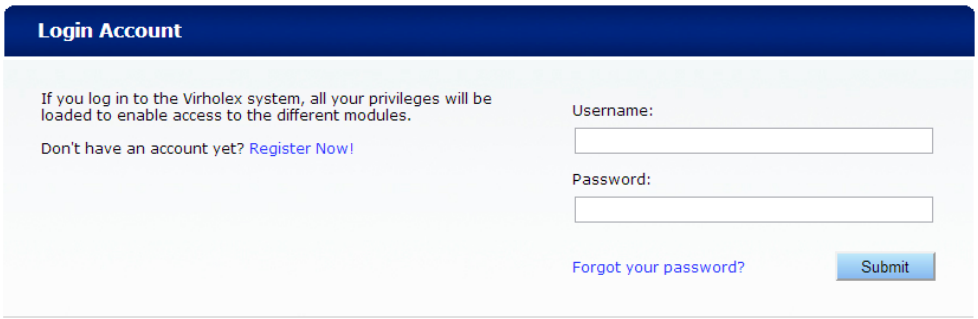


Figure 5.16 Login application of VirHoLex 2.0

Figures 5.17 and 5.18 show the *LoginAccountForm.java* file extending the `ActionForm` class for Login application. The *LoginAccountForm.java* contains getter and setter methods corresponding to each field on the input page's form, the *loginAccount.jsp* (see Figure 5.25). As shown in line 22-27 of Figure 5.16, the username property has `getUsername()` and `setUsername(String username)` methods, likewise with the other form properties. Figure 5.18 shows that the *LoginAccountForm.java* contains a simple validation for username and password. Inside the validate method username and/or password is checked whether the value is not empty. When both have a value, the method validates the username if it is an existing account and/or the password if it matches with the username (complex validation). The corresponding error message is displayed to the user if the input does not satisfy the given conditions.

Figure 5.20 illustrates a sample error message on the login form wherein the user entered an invalid username or password. Either line 69 or 86 of Figure 5.18 was triggered during simple validation; hence "Invalid username or password" was displayed.

```
12 public class LoginAccountForm extends ActionForm {
13
14     private String username;
15     private String password;
16     private String dbUsername;
17     private String dbPassword;
18     private String status;
19     private String errorMessage;
20     private String command;
21
22     public void setUsername(String username) {
23         this.username = username;
24     }
25     public String getUsername() {
26         return username;
27     }
28     public void setPassword(String password) {
29         this.password = password;
30     }
31     public String getPassword() {
32         return password;
33     }
34     public void setDbUsername(String dbUsername) {
35         this.dbUsername = dbUsername;
36     }
37     public String getDbUsername() {
38         return dbUsername;
39     }
40     public void setDbPassword(String dbPassword) {
41         this.dbPassword = dbPassword;
42     }
43     public String getDbPassword() {
44         return dbPassword;
45     }
46     public void setStatus(String status) {
47         this.status = status;
48     }
49     public String getStatus() {
50         return status;
51     }
52     public void setCommand(String command) {
53         this.command = command;
54     }
55     public String getCommand() {
56         return command;
57     }
58     public void setErrorMessage(String errorMessage) {
59         this.errorMessage = errorMessage;
60     }
61     public String getErrorMessage() {
62         return errorMessage;
63     }
```

Figure 5.17 Getter and setter methods corresponding to each property of the form

```
64
65    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
66        ActionErrors errors = new ActionErrors();
67
68        if (Utilities.isEmpty(this.getUsername()) && Utilities.isEmpty(this.getPassword())) {
69            errors.add("errorMessage", new ActionMessage("error.invalid.login"));
70            errors.add("username", new ActionMessage(""));
71            errors.add("password", new ActionMessage(""));
72        }
73        else {
74
75            if (Utilities.isEmpty(this.getUsername())) {
76                errors.add("errorMessage", new ActionMessage("error.empty", "Username"));
77                errors.add("username", new ActionMessage(""));
78            } else {
79                if(Utilities.isEmpty(this.getDbUsername()) && !Utilities.isEmpty(this.getPassword())){
80                    errors.add("errorMessage", new ActionMessage("error.available.username", this.getUsername()));
81                } else {
82                    if (Utilities.isEmpty(this.getPassword())) {
83                        errors.add("errorMessage", new ActionMessage("error.empty", "Password"));
84                        errors.add("password", new ActionMessage(""));
85                    } else if(this.getUsername().equals(this.getDbUsername()) && !this.getPassword().equals(this.getDbPassword())) {
86                        errors.add("errorMessage", new ActionMessage("error.invalid.login"));
87                        errors.add("username", new ActionMessage(""));
88                        errors.add("password", new ActionMessage(""));
89                    }
90                }
91            }
92        }
93
94        return errors;
95    }
96
```

Figure 5.18 Simple Validation of Login application

The key passed to the error message was configured on *ApplicationResource.properties* file, which was located in */src/com/VirHoLex/general/* directory. For example, the key “error.invalid.login” was used in the `ActionMessage` to add new error when the username and/or password entered was invalid. When the instance of `ActionErrors` was returned, the value of “error.invalid.login” would be displayed on the login form using the `<html:errors>` tag (see Figure 5.25), as illustrated on Figure 5.20. The complete list of key/value pairs for error messages is shown on Figure 5.19.

```
error.required.all = All fields are required
error.required.asterisk = All fields with asterisk are required
error.minlength = {0} cannot be less than 6 characters
error.required = {0} is required
error.notEqual.password = Passwords entered do not match
error.invalid = {0} is invalid
error.invalid.login = Invalid username or password<br/><br/>
error.invalid.account = <br/>Invalid username or email address<br/><br/>
error.available.username = {0} is still available<br/><br/>
error.empty = <br/>Please enter your {0}<br/><br/>
error.empty2 = Please enter your {0}
error.select.default = Please select value for {0}
error.notAvailable = This {0} is not available
error.default = Please select {0}
error.file = {0} is not a PDF file
error.invalid.image = Invalid image file
error.image = You must upload an image file with one of the following extensions: jpg, jpeg, gif, png, bmp
```

Figure 5.19 `ActionMessage` key/value pairs in *ApplicationResource.properties*

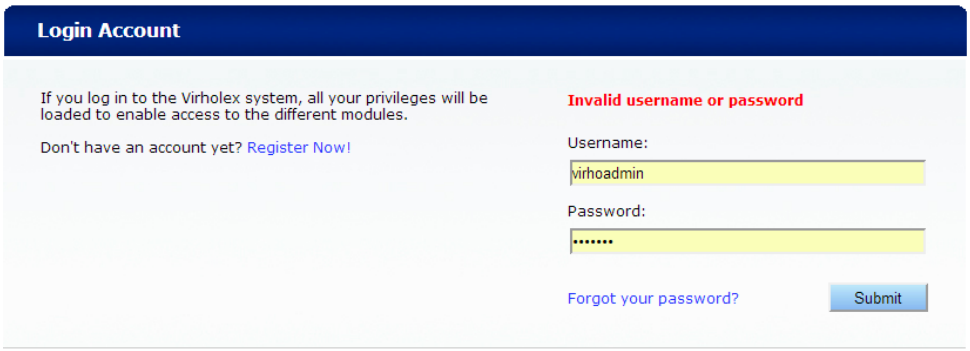


Figure 5.20 Error message on login form when invalid username or password is entered

The *LoginAccountForm.java* was then mapped to a name `LoginForm` that served as an alias when used for both *struts-config.xml* and *loginAccount.jsp*. The declaration was done on the *struts-config.xml* under `<form-beans>` tag, as shown in Figure 5.21.

```
10        <form-bean name="LoginForm" type="com.virholex.virus.LoginAccountForm"/>
```

Figure 5.21 Form bean declaration for *LoginAccountForm.java*

Figure 5.22 shows the *LoginAccountAction.java* extending the Struts' `Action` class, and the business logic of Login application that was contained in the execute method. Initially, the `ActionForm` was typecast to `LoginAccountForm` so that the form properties, such as the username and password, could be accessed via the getter and setter methods. In this `Action` subclass, the complex validation had coincided with the simple validation as seen in lines 34-38 of Figure 5.22 and the validate method on Figure 5.18. When the username entered by the user is not empty, the application will check if the username is valid. Method in Figure 5.23 will return `null` if the given account does not exist; otherwise the returned values will be stored in dbUsername and dbPassword respectively before calling the validate method of *LoginActionForm.java*.

The *LoginAccountDAO.java* on Figure 5.23 is a sample model class. The persistent storage was separated from the *LoginAccountAction.java* to make the codes reusable and easy-to-read. Instead `checkAccount()` method was called to retrieve the necessary data from the database.

Figure 5.22 also shows the Login application's indirect reference to the four possible "next" pages. The default "next" page is the "VirHoLex.virus.login" or the input page. The list of the possible "next" pages for *LoginAccountAction.java* was specified in the *struts-config.xml*, as shown on Figure 5.24.

```java
public class LoginAccountAction extends Action {

@Override
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse
    LoginAccountForm loginForm = (LoginAccountForm) form;
    LoginAccountVO loginVO = new LoginAccountVO();
    LoginAccountVO vo = new LoginAccountVO();
    VirhoVIRUSWorker virusWorker = new VirhoVIRUSWorker();
    ActionErrors errors = new ActionErrors();
    HttpSession session = request.getSession(true);

    String forward = "virholex.virus.login";
    String status = "", username = "";

    PropertyUtils.copyProperties(loginVO, loginForm);

    if(loginForm.getCommand() != null && loginForm.getCommand().equals("Submit")){

        if(loginVO.getUsername()!=null) {
            vo = (LoginAccountVO)virusWorker.checkAccount(loginVO);
            loginForm.setDbUsername(vo.getDbUsername());
            loginForm.setDbPassword(vo.getDbPassword());
        }

        errors = loginForm.validate(mapping, request);

        if(errors.isEmpty()) {

            username = loginVO.getUsername();
            status = virusWorker.getUserStatus(username);

            if(status.equals(VirholexConstants.STATUS_ACTIVATED) || status.equals(VirholexConstants.STATUS_ADMIN) ||
                    status.equals(VirholexConstants.STATUS_SUPERADMIN)) {
                session.setAttribute("userLogin", username);
                session.setAttribute("statusLogin", status);
                forward = "virholex.virui.home";
            } else {
                request.setAttribute("status", status);

                if(status.equals(VirholexConstants.STATUS_SUSPENDED)) {
                    session.setAttribute("username", username);
                    forward = "virholex.virus.suspendedAccount";
                } else if(status.equals(VirholexConstants.STATUS_APPROVED)) {
                    session.setAttribute("userLogin", username);
                    session.setAttribute("statusLogin", VirholexConstants.STATUS_ACTIVATED);
                    request.setAttribute("username", username);
                    virusWorker.changeStatus(username);
                    forward = "virholex.virus.statusCheck";
                } else {
                    request.setAttribute("username", username);
                    forward = "virholex.virus.statusCheck";
                }
            }
        } else {
            saveErrors(request, errors);
        }

    } else if(loginForm.getCommand() != null && loginForm.getCommand().equals("Yes")) {
        username = (String)session.getAttribute("username");
        session.removeAttribute("username");
        virusWorker.evaluateAccount(username, VirholexConstants.STATUS_SUSPENDED);
        request.setAttribute("status", VirholexConstants.STATUS_SUSPENDED);
        forward = "virholex.virus.statusCheck";
    } else if(loginForm.getCommand() != null && loginForm.getCommand().equals("No")) {
        forward = "virholex.virui.home";
    }

    return mapping.findForward(forward);
}
}
```

Figure 5.22 Processing of business logic for Login application

```
40
41  public static LoginAccountVO checkAccount(LoginAccountVO vo) throws SQLException, NamingException {
42
43      Connection conn = null;
44      PreparedStatement ps = null;
45      ResultSet rs = null;
46
47      try {
48          String sql = "";
49          conn = DBConnect.getConnection();
50
51          sql = "SELECT username FROM users WHERE username=?";
52          ps = conn.prepareStatement(sql);
53          ps.setString(1, vo.getUsername());
54          rs = ps.executeQuery();
55
56          while(rs.next()) {
57              vo.setDbUsername(rs.getString("username"));
58          }
59
60          rs.close();
61          ps.close();
62
63          if(vo.getUsername()!=null) {
64              sql = "SELECT AES_DECRYPT(password,?) AS password FROM users WHERE username=?";
65              ps = conn.prepareStatement(sql);
66              ps.setString(1, vo.getUsername());
67              ps.setString(2, vo.getUsername());
68              rs = ps.executeQuery();
69
70              if(rs.next()) {
71                  vo.setDbPassword(rs.getString("password"));
72              }
73
74              rs.close();
75              ps.close();
76          }
77
78      } finally {
79          if(rs != null) rs.close();
80          if(ps != null) ps.close();
81          if(conn != null) conn.close();
82      }
83
84      return vo;
85  }
```

Figure 5.23 The `checkAccount()` method in LoginAccountDAO.java

In addition, Figure 5.24 illustrates the declaration of form handler for Login application named `Login.do`, which was also defined in the Struts' configuration file (*struts-config.xml*). This handler was used by `action` attribute of `<html:form>`, as shown in line 11 of Figure 25.

```
97        <action path="/Login"
98                type="com.virholex.virus.LoginAccountAction"
99                name="LoginForm"
100               scope="request"
101               validate="false">
102           <forward name="virholex.virui.home" path="/Home.do"/>
103           <forward name="virholex.virus.login" path="/rd_virus/loginAccount.jsp"/>
104           <forward name="virholex.virus.statusCheck" path="/rd_virus/statusAccountMessage.jsp"/>
105           <forward name="virholex.virus.suspendedAccount" path="/rd_virus/reactivateAccount.jsp"/>
106       </action>
```

Figure 5.24 Form handler declaration for Login application

Figure 5.25 shows the View component of the Login application (*loginAccount.jsp*). Three form properties that were declared in *LoginAccountForm.java* were present on the jsp page's form namely username, password and command, as shown in lines 45-71. On the other hand, the error messages encountered during simple validation are displayed on the `<html:errors>` tag depending on which property key has an error.

Two custom tags were present on *loginAccount.jsp* page: the HTML and Bean tags. In order to use the tags, TLD files corresponding to those tags were first placed in the */WEB-INF/* directory. The files were then added on the deployment descriptor (*web.xml*) under the `<taglib>` sections, as shown Figure 5.26.

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Virus-Host Interaction Lexicon</title>
<link rel="shortcut icon" href="./images/favicon.ico" >
<link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
</head>
<body>
<html:form action="/Login" method="POST">
    <table class="body" align="center">
        <jsp:include page="/rd_virui/header.jsp" />
        <tr>
            <td> </td>
        </tr>
        <tr valign="top">
            <td class="sidebar">
                <jsp:include page="/rd_virui/sidebar.jsp" />
            </td>
            <td>
            <table class="content">
                <tr>
                    <td class="breadcrumb">
                        <html:link page="/Home.do"><bean:message key="button.virho.home" /></html:link> &#187;
                        <span class="breadcrumb-selected">Login Account</span>
                    </td>
                </tr>
                <tr>
                    <td> </td>
                </tr>
                <tr>
                    <td class="content-title-form" colspan="2">Login Account</td>
                </tr>
                <tr>
                    <td>
                    <table class="content-desc-form">
                        <tr valign="top">
                            <td rowspan="6" width="470px">
                                If you log in to the Virholex system, all your privileges will be<br/>
                                loaded to enable access to the different modules.<br><br>
                                Don't have an account yet? <html:link page="/Register.do" >Register Now!</html:link>
                            </td>
                        </tr>
                        <tr>
                            <td colspan="2">
                                <span class="error-text-big"><html:errors property="errorMessage" /></span>
                                <br/>
                                <span style="position:relative; top:-6px;">Username:</span>
                                <br/>
                                <html:text property="username" size="45" style="remove-error-field"
                                        errorStyleClass="error-field" errorKey="org.apache.struts.action.ERROR" />
                                <span class="error-text"><html:errors property="username" /></span>
                            </td>
                        </tr>
                        <tr>
                            <td colspan="2">
                                <span style="position:relative; top:-6px;">Password: </span>
                                <br/>
                                <html:password property="password" size="45" style="remove-error-field"
                                        errorStyleClass="error-field" errorKey="org.apache.struts.action.ERROR" />
                                <span class="error-text"><html:errors property="password" /></span>
                            </td>
                        </tr>
                        <tr>
                            <td height="50px">
                                <html:link page="/ForgotPassword.do" >Forgot your password?</html:link>
                            </td>
                                <td align="right">
                                <html:submit property="command" value="Submit" styleClass="button" />
                            </td>
                        </tr>
                    </table>
                    </td>
                </tr>
            </table>
            </td>
        </tr>
        <tr class="footer">
            <td colspan="2">
            &copy; <bean:message key="virho.copyright" />
            </td>
        </tr>
    </table>
</html:form>
</body>
</html>
```

Figure 5.25 View component of Login application

```
<jsp-config>
      <taglib>
          <taglib-uri>/WEB-INF/c.tld</taglib-uri>
          <taglib-location>/WEB-INF/c.tld</taglib-location>
      </taglib>
      <taglib>
          <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
          <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
      </taglib>
      <taglib>
          <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
          <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
      </taglib>
      <taglib>
          <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
          <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
      </taglib>
      <taglib>
          <taglib-uri>/WEB-INF/struts-nested.tld</taglib-uri>
          <taglib-location>/WEB-INF/struts-nested.tld</taglib-location>
      </taglib>
      <taglib>
          <taglib-uri>/WEB-INF/struts-tiles.tld</taglib-uri>
          <taglib-location>/WEB-INF/struts-tiles.tld</taglib-location>
      </taglib>
  </jsp-config>
```

Figure 5.26 Taglib section in web.xml

In order to show that Struts were also applied to other modules of VirHoLex, one feature was taken per module. This feature was then broken into parts just what was done on Login page application. The following shows the Controller (`ActionForm` subclass, `Action` subclass, and its Form bean and Form handler declarations); the Model (sample model class, i.e. a DAO file) and the View (jsp file) components for each feature of the other modules.

a. The View Virus List feature under Virho Information Services module that displays the complete list of viruses available in the system.

```
12
13  public class VirusForm extends ActionForm {
14
15      private static final long serialVersionUID = -4735625968524520 21L;
16
17      private String virusName;
18      private String commonName;
19      private String scientificName;
20      private String serotype;
21      private String host;
22      private String virusArchitecture;
23      private String infectivity;
24      private String mannerOfInfection;
25      private String vector;
26      private String antimicrobialTreatment;
27      private String receptors;
28      private String description;
29      private String dateAdded;
30      private String dateModified;
31      private String status;
32      private int virusId;
33      private String command;
34      private String author;
35      private boolean hasVirus;
36      private String commonNameDB;
37      private String descriptionDB;
38      private FormFile file;
39      private String image;
40      private String imageDB;
41      private boolean hasImage;
42
43      public void setVirusName(String virusName) {
44          this.virusName = virusName;
45      }
46      public String getVirusName() {
47          return virusName;
48      }
49      public void setCommonName(String commonName) {
50          this.commonName = commonName;
51      }
52      public String getCommonName() {
53          return commonName;
54      }
55      public void setScientificName(String scientificName) {
56          this.scientificName = scientificName;
57      }
58      public String getScientificName() {
59          return scientificName;
60      }
61      public void setSerotype(String serotype) {
62          this.serotype = serotype;
63      }
64      public String getSerotype() {
65          return serotype;
66      }
67      public void setHost(String host) {
68          this.host = host;
69      }
70      public String getHost() {
71          return host;
72      }
73      public void setVirusArchitecture(String virusArchitecture) {
74          this.virusArchitecture = virusArchitecture;
75      }
76      public String getVirusArchitecture() {
77          return virusArchitecture;
78      }
79      public void setInfectivity(String infectivity) {
80          this.infectivity = infectivity;
81      }
82      public String getInfectivity() {
83          return infectivity;
84      }
85      public void setMannerOfInfection(String mannerOfInfection) {
86          this.mannerOfInfection = mannerOfInfection;
87      }
88      public String getMannerOfInfection() {
89          return mannerOfInfection;
90      }
91      public void setVector(String vector) {
92          this.vector = vector;
93      }
94      public String getVector() {
95          return vector;
96      }
97      public void setAntimicrobialTreatment(String antimicrobialTreatment) {
98          this.antimicrobialTreatment = antimicrobialTreatment;
99      }
100     public String getAntimicrobialTreatment() {
101         return antimicrobialTreatment;
102     }
103     public void setReceptors(String receptors) {
104         this.receptors = receptors;
105     }
106     public String getReceptors() {
107         return receptors;
108     }
109     public void setDescription(String description) {
110         this.description = description;
111     }
112     public String getDescription() {
113         return description;
114     }
115     public void setDateAdded(String dateAdded) {
116         this.dateAdded = dateAdded;
117     }
118     public String getDateAdded() {
119         return dateAdded;
120     }
```

```
121    public void setDateModified(String dateModified) {
122        this.dateModified = dateModified;
123    }
124    public String getDateModified() {
125        return dateModified;
126    }
127    public void setStatus(String status) {
128        this.status = status;
129    }
130    public String getStatus() {
131        return status;
132    }
133    public void setVirusId(int virusId) {
134        this.virusId = virusId;
135    }
136    public int getVirusId() {
137        return virusId;
138    }
139    public void setAuthor(String author) {
140        this.author = author;
141    }
142    public String getAuthor() {
143        return author;
144    }
145    public void setCommand(String command) {
146        this.command = command;
147    }
148    public String getCommand() {
149        return command;
150    }
151    public void setHasVirus(boolean hasVirus) {
152        this.hasVirus = hasVirus;
153    }
154    public boolean isHasVirus() {
155        return hasVirus;
156    }
157
158    public void setCommonNameDB(String commonNameDB) {
159        this.commonNameDB = commonNameDB;
160    }
161    public String getCommonNameDB() {
162        return commonNameDB;
163    }
164    public void setDescriptionDB(String descriptionDB) {
165        this.descriptionDB = descriptionDB;
166    }
167    public String getDescriptionDB() {
168        return descriptionDB;
169    }
170    public void setFile(FormFile file) {
171        this.file = file;
172    }
173    public FormFile getFile() {
174        return file;
175    }
176    public void setImage(String image) {
177        this.image = image;
178    }
179    public String getImage() {
180        return image;
181    }
182    public void setImageDB(String imageDB) {
183        this.imageDB = imageDB;
184    }
185    public String getImageDB() {
186        return imageDB;
187    }
188
189    public void setHasImage(boolean hasImage) {
190        this.hasImage = hasImage;
191    }
192    public boolean isHasImage() {
193        return hasImage;
194    }
195    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
196        ActionErrors errors = new ActionErrors();
197
198        if (Utilities.isEmpty(this.getCommonName())) {
199            errors.add("errorMessage", new ActionMessage("error.required.asterisk"));
200            errors.add("commonName", new ActionMessage(""));
201        } else {
202            if(this.isHasVirus()) {
203                errors.add("commonName", new ActionMessage("error.notAvailable", " virus"));
204            }
205        }
206
207        return errors;
208    }
209    public ActionErrors validateUploadSVG(ActionMapping mapping, HttpServletRequest request) {
210        ActionErrors errors = new ActionErrors();
211
212        if (!Utilities.isEmpty(this.getFile().toString())) {
213            if(!(Utilities.getFileExtension(this.getFile().toString()).equals("jpg") || Utilities.getFileExtension
214                    Utilities.getFileExtension(this.getFile().toString()).equals("jpeg") || Utilities.getFileExten
215                    Utilities.getFileExtension(this.getFile().toString()).equals("gif") || Utilities.getFileExtens
216                    Utilities.getFileExtension(this.getFile().toString()).equals("png") || Utilities.getFileExtens
217                    Utilities.getFileExtension(this.getFile().toString()).equals("bmp") || Utilities.getFileExtens
218                errors.add("file", new ActionMessage("error.image"));
219            }
220        }
221
222        return errors;
223    }
```

Figure 5.27 *ViewVirusForm.java* (`ActionForm` subclass)

```
13
14 public class ViewVirusAction extends Action {
15
16 @Override
17 public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
18     VirusForm virusForm = (VirusForm) form;
19     VirusVO virusVO = new VirusVO();
20     VirhoBasicInfoWorker basicInfoWorker = new VirhoBasicInfoWorker();
21     ArrayList<String> letterList = new ArrayList<String>();
22     ArrayList<VirusVO> virusList = new ArrayList<VirusVO>();
23     HttpSession session = request.getSession(true);
24
25     String username = (String)session.getAttribute("userLogin");
26     String forward = "virholex.basicinfo.viewVirus";
27     String letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
28     String list = request.getParameter("list");
29     boolean isManager = false;
30
31     if(virusForm.getCommand()!=null && virusForm.getCommand().equals("Add Virus")) {
32         forward = "virholex.basicinfo.addVirus";
33     } else {
34
35         for(int i=0; i < letters.length() ; i++){
36             if(basicInfoWorker.checkVirus(letters.charAt(i)).size() > 0)
37                 letterList.add(Character.toString(letters.charAt(i)));
38             else
39                 letterList.add("-");
40         }
41         // Get list of viruses
42         if(list != null) {
43             virusList = basicInfoWorker.getVirus(virusVO, list);
44         }
45         // Check if the user is a Hotspot Manager
46         isManager = basicInfoWorker.isManager(username);
47
48         request.setAttribute("letters", letterList);
49         request.setAttribute("virus", virusList);
50         request.setAttribute("privilege", isManager);
51
52     }
53
54     return mapping.findForward(forward);
55 }
56
57 }
```

Figure 5.28 *ViewVirusAction.java* (`Action` subclass)

```
28         <form-bean name="VirusForm" type="com.virholex.basicinfo.VirusForm"/>
```

Figure 5.29 Form bean declaration for *VirusForm.java*

```
590         <action path="/ViewVirus"
591                 type="com.virholex.basicinfo.ViewVirusAction"
592                 name="VirusForm"
593                 scope="request"
594                 validate="false">
595             <forward name="virholex.basicinfo.viewVirus" path="/rd_virhobasicinfo/viewVirusList.jsp" />
596             <forward name="virholex.basicinfo.addVirus" path="/AddVirus.do" />
597         </action>
```

Figure 5.30 Form handler declaration for View Virus List

```java
public static ArrayList<VirusVO> getVirus(VirusVO vo, String letter) throws SQLException, NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    VirusVO virusVO = new VirusVO();
    ArrayList<VirusVO> virusList = new ArrayList<VirusVO>();

    try {
        String sql = "SELECT * FROM virus WHERE virus LIKE '" + letter + "%' ORDER BY virus";
        conn = DBConnect.getConnection();
        s = conn.createStatement();
        s.execute(sql);
        rs = s.getResultSet();

        while(rs.next()) {
            virusVO.setVirusName(rs.getString("virus"));
            virusVO.setDescription(rs.getString("description"));

            virusList.add(virusVO);
            virusVO = new VirusVO();
        }

    } finally {
        if(rs != null) rs.close();
        if(s != null) s.close();
        if(conn != null) conn.close();
    }

    return virusList;
}
```

Figure 5.31 The `getVirus()` method in *ViewVirusDAO.java* (Model class)

```jsp
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Virus-Host Interaction Lexicon</title>
<link rel="shortcut icon" href="./images/favicon.ico" >
<link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
</head>
<body>
<html:form action="/ViewVirus" method="POST">
    <table class="body" align="center">
        <jsp:include page="/rd_virui/header.jsp" />
        <tr>
            <td> </td>
        </tr>
        <tr class="content" valign="top">
            <td class="sidebar">
                <jsp:include page="/rd_virui/sidebar.jsp" />
            </td>
            <td>
            <table class="content">
                <tr>
                    <td class="breadcrumb">
                        <html:link page="/Home.do"><bean:message key="button.virho.home" /></html:link> &#187;
                        <span class="breadcrumb-selected">Virus List</span>
                    </td>
                </tr>
                <tr>
                    <td> </td>
                </tr>
                <tr>
                    <td class="content-title-form" colspan="2">Virus List</td>
                </tr>
                <tr>
                    <td>
                    <table class="content-desc-form">
                        <tr>
                            <td>
                            <table class="letterList">
                                <tr>
                                    <td>
                                    <c:forEach var="letter" items="A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,
                                        <c:choose>
                                            <c:when test="${letters[loop.index] eq letter}">
                                                <html:link page="/ViewVirus.do?list=${letter}"><span class="font-lett
                                            </c:when>
                                            <c:otherwise>
                                                ${letter} 
                                            </c:otherwise>
                                        </c:choose>
                                    </c:forEach>
                                    <html:link page="/ViewVirus.do?list=">View All</html:link>
                                    </td>
                                </tr>
                            </table>
                            </td>
                        </tr>
```

```
61                      <tr>
62                          <td>
63                          <table class="table-content">
64                              <tr>
65                                  <th>Virus Name</th>
66                                  <th>Description</th>
67                              </tr>
68                              <c:choose>
69                                  <c:when test="${!empty virus}">
70                                      <c:forEach var="detail" items="${virus}">
71                                      <tr>
72                                          <td width="200px" valign="top">
73                                              <html:link page="/ViewVirusDetails.do?virus=${detail.virusName}"><spa
74                                          </td>
75                                          <td>${detail.description} </td>
76                                      </tr>
77                                      </c:forEach>
78                                  </c:when>
79                                  <c:otherwise>
80                                      <tr>
81                                          <td class="font-italic" colspan="2">No Virus Yet</td>
82                                      </tr>
83                                  </c:otherwise>
84                              </c:choose>
85                          </table>
86                          </td>
87                      </tr>
88                  </table>
89                  </td>
90              </tr>
91              <tr>
92                  <td> </td>
93              </tr>
94              <tr>
95                  <td>    
96                  <c:if test="${(userLogin ne null) and (privilege eq true)}" >
97                      <html:submit property="command" value="Add Virus" styleClass="button-long-blue" />
98                  </c:if>
99                  </td>
100             </tr>
101         </table>
102         </td>
103     </tr>
104     <tr class="footer">
105         <td colspan="2">
106         &copy; <bean:message key="virho.copyright" />
107         </td>
108     </tr>
109 </table>
110 </html:form>
111 </body>
112 </html>
```

Figure 5.32 *viewVirusList.jsp* (view component)

b. The View Project feature under Virho Experiments module that lists down all the projects created by the Experiment Lead Coordinators.

```
15
16 public class ViewProjectDetailsForm extends ActionForm {
17
18     private static final long serialVersionUID = 6191348032739415272L;
19     private String command;
20     private int privilege;
21     private String projectName;
22     private String projectOldName;
23     private String description;
24     private String status;
25     private String author;
26     private ArrayList<ViewExperimentDetailsVO> experiments;
27     private ArrayList<ViewModelDetailsVO> models;
28     private int projectId;
29     private String dateAdded;
30     private boolean hasProject;
31
32     public void setCommand(String command) {
33         this.command = command;
34     }
35
36     public String getCommand() {
37         return command;
38     }
```

```
39
40      public void setPrivilege(int privilege) {
41          this.privilege = privilege;
42      }
43
44      public int getPrivilege() {
45          return privilege;
46      }
47
48      public void setProjectName(String projectName) {
49          this.projectName = projectName;
50      }
51
52      public String getProjectName() {
53          return projectName;
54      }
55
56      public void setDescription(String description) {
57          this.description = description;
58      }
59
60      public String getDescription() {
61          return description;
62      }
63
64      public void setStatus(String status) {
65          this.status = status;
66      }
67
68      public String getStatus() {
69          return status;
70      }
71
72      public void setAuthor(String author) {
73          this.author = author;
74      }
75
76      public String getAuthor() {
77          return author;
78      }
79
80      public void setProjectOldName(String projectOldName) {
81          this.projectOldName = projectOldName;
82      }
83
84      public String getProjectOldName() {
85          return projectOldName;
86      }
87
88      public void setExperiments(ArrayList<ViewExperimentDetailsVO> experiments) {
89          this.experiments = experiments;
90      }
91
92      public ArrayList<ViewExperimentDetailsVO> getExperiments() {
93          return experiments;
94      }
95
96      public void setModels(ArrayList<ViewModelDetailsVO> models) {
97          this.models = models;
98      }
99
100     public ArrayList<ViewModelDetailsVO> getModels() {
101         return models;
102     }
103
104     public void setProjectId(int projectId) {
105         this.projectId = projectId;
106     }
107
108     public int getProjectId() {
109         return projectId;
110     }
111
112     public void setDateAdded(String dateAdded) {
113         this.dateAdded = dateAdded;
114     }
115
116     public String getDateAdded() {
117         return dateAdded;
118     }
119
120     public void setHasProject(boolean hasProject) {
121         this.hasProject = hasProject;
122     }
123
124     public boolean isHasProject() {
125         return hasProject;
126     }
127
128     public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
129         ActionErrors errors = new ActionErrors();
130
131         if(Utilities.isEmpty(this.getProjectName()) || this.getStatus().equals(VirholexConstants.DEFAULT_VALUE)) {
132             errors.add("errorMessage", new ActionMessage("error.required.asterisk"));
133
134             if(Utilities.isEmpty(this.getProjectName())) errors.add("projectName", new ActionMessage(""));
135             if(this.getStatus().equals(VirholexConstants.DEFAULT_VALUE)) errors.add("status", new ActionMessage(""));
136         }
137
138         if(this.isHasProject()) {
139             errors.add("projectName", new ActionMessage("error.notAvailable", "project name"));
140         }
141
142         return errors;
143     }
```

Figure 5.33 *ViewProjectDetailsForm.java* (`ActionForm` subclass)

```java
public class ViewProjectsAction extends Action {

@Override
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
    ViewProjectDetailsForm projectForm = (ViewProjectDetailsForm) form;
    ViewProjectDetailsVO projectVO = new ViewProjectDetailsVO();
    VirhoExperimentsWorker experimentWorker = new VirhoExperimentsWorker();
    HttpSession session = request.getSession(true);
    ArrayList<ViewProjectDetailsVO> projectList = new ArrayList<ViewProjectDetailsVO>();

    String forward = "virholex.experiments.viewProjects";
    String username = (String)session.getAttribute("userLogin");
    String roleName = "";
    int projectId = 0;
    boolean isLead = false;

    if(projectForm.getCommand() != null && projectForm.getCommand().equals("Add Project")) {
        forward = "virholex.experiments.addProject";
    } else {

        projectList = experimentWorker.getProjects(projectVO);
        isLead = experimentWorker.isLeadExperimentInvestigator(username);

        for(int i=0; i<projectList.size(); i++) {
            projectId = ((ViewProjectDetailsVO)projectList.get(i)).getProjectId();
            roleName = experimentWorker.getProjectRole(username, projectId);

            if(roleName.equals(VirholexConstants.EXPERIMENT_LEAD_INVESTIGATOR_DESC))
                ((ViewProjectDetailsVO)projectList.get(i)).setPrivilege(VirholexConstants.EXPERIMENT_LEAD_INVESTIGATOR);
            else if(roleName.equals(VirholexConstants.EXPERIMENT_INVESTIGATOR_DESC))
                ((ViewProjectDetailsVO)projectList.get(i)).setPrivilege(VirholexConstants.EXPERIMENT_INVESTIGATOR);
            else if(roleName.equals(VirholexConstants.RESTRICTED_USER_DESC))
                ((ViewProjectDetailsVO)projectList.get(i)).setPrivilege(VirholexConstants.RESTRICTED_USER);
            else
                ((ViewProjectDetailsVO)projectList.get(i)).setPrivilege(VirholexConstants.REGISTERED_USER);
        }

        request.setAttribute("isLead", isLead);
    }

    request.setAttribute("projects", projectList);
    session.setAttribute("module", VirholexConstants.VIRHO_EXPERIMENTS);

    return mapping.findForward(forward);
}

}
```

Figure 5.34 *ViewProjectsAction.java* (`Action` subclass)

```
<form-bean name="ProjectDetailsForm" type="com.virholex.experiments.ViewProjectDetailsForm"/>
```

Figure 5.35 Form bean declaration for ViewProjectDetailsForm.java

```
<action path="/ViewProjects"
        type="com.virholex.experiments.ViewProjectsAction"
        name="ProjectDetailsForm"
        scope="request"
        validate="false">
    <forward name="virholex.experiments.viewProjects" path="/rd_virhoexperiments/viewProjects.jsp"/>
    <forward name="virholex.experiments.addProject" path="/rd_virhoexperiments/addProject.jsp"/>
</action>
```

Figure 5.36 Form handler declaration for View Project List

```
16    public static ArrayList<ViewProjectDetailsVO> getProjects(ViewProjectDetailsVO vo) throws SQLException, NamingException {
17
18        Connection conn = null;
19        Statement s = null;
20        ResultSet rs = null;
21        ArrayList<ViewProjectDetailsVO> projectList = new ArrayList<ViewProjectDetailsVO>();
22
23        try {
24            String sql = "SELECT project_id, project_name, description FROM project WHERE project_name NOT LIKE '%VirhoHotspot%' AND " +
25                    "project_name NOT LIKE '%VirhoReference%' ORDER BY project_name";
26            conn = DBConnect.getConnection();
27            s = conn.createStatement();
28            s.executeQuery(sql);
29            rs = s.getResultSet();
30
31            while(rs.next()) {
32                vo.setProjectId(rs.getInt("project_id"));
33                vo.setProjectName(rs.getString("project_name"));
34                vo.setDescription(rs.getString("description"));
35
36                projectList.add(vo);
37                vo = new ViewProjectDetailsVO();
38            }
39
40        } finally {
41            if(rs != null) rs.close();
42            if(s != null) s.close();
43            if(conn != null) conn.close();
44        }
45
46        return projectList;
47    }
```

Figure 5.37 The `getProjects()` method in *ViewProjectDAO.java* (Model class)

```
1  <%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
2  <%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
3  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
4  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8  <title>Virus-Host Interaction Lexicon</title>
9  <link rel="shortcut icon" href="./images/favicon.ico" >
10 <link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
11 <script src="./files/jquery-1.8.0.js"></script>
12 <script>
13     function loadPage(project, id){
14         if(id == 'editProject')
15             window.open ('EditProject.do?project='+project, '_self', false);
16         else if(id == 'addExperiment')
17             window.open ('AddExperiment.do?project='+project, '_self', false);
18     }
19 </script>
20 </head>
21 <body>
22 <html:form action="/ViewProjects" method="POST">
23     <table class="body" align="center">
24         <jsp:include page="/rd_virui/header.jsp" />
25         <tr>
26             <td> </td>
27         </tr>
28         <tr valign="top">
29             <td class="sidebar">
30                 <jsp:include page="/rd_virui/sidebar.jsp" />
31             </td>
32             <td>
33             <c:if test="${(userLogin ne null) and ((statusLogin eq 'activated') or (statusLogin eq 'admin') or (statusLogin eq 'superadmin'))}">
34             <table class="content">
35                 <tr>
36                     <td class="breadcrumb">
37                         <html:link page="/Home.do"><bean:message key="button.virho.home" /></html:link> &#187;
38                         <span class="breadcrumb-selected">Virho Experiments Projects</span>
39                     </td>
40                 </tr>
41                 <tr>
42                     <td> </td>
43                 </tr>
44                 <tr>
45                     <td class="content-title-form" colspan="2">List of Projects</td>
46                 </tr>
47                 <tr>
48                     <td>
49                     <table class="content-desc-form">
50                         <tr>
51                             <td>
52                             <table class="table-content">
53                                 <tr>
54                                     <th>Project</th>
55                                     <th>Description</th>
56                                 </tr>
57                                 <c:choose>
58                                     <c:when test="${fn:length(projects) gt 0}">
59                                         <c:forEach var="detail" items="${projects}">
```

```
                                    <tr>
                                        <td width="250px">
                                            <html:link page="/ViewProjectExperimentDetails.do?project=${detail.projectName}">
                                            <span class="font-blue"><c:out value="${detail.projectName}" /></span></html:link><br/>
                                            <c:if test="${detail.privilege ge 3}" >
                                                <html:submit property="command" value="Add Experiment" styleClass="button-remove" onclick="lo
                                                <html:submit property="command" value="Edit Project" styleClass="button-remove" onclick="loac
                                                <c:if test="${detail.privilege eq 4}" >
                                                    <c:set var="privilege" value="${detail.privilege}" />
                                                </c:if>
                                            </c:if>
                                        </td>
                                        <td><c:out value="${detail.description}" /></td>
                                    </tr>
                                    </c:forEach>
                                </c:when>
                                <c:otherwise>
                                    <tr>
                                        <td class="font-italic" colspan="2">No Projects Yet</td>
                                    </tr>
                                </c:otherwise>
                            </c:choose>
                        </table>
                        </td>
                    </tr>
                </table>
                </td>
            </tr>
            <tr>
                <td> </td>
            </tr>
            <tr>
                <td>
                    
                <c:if test="${(privilege eq 4) or (requestScope.isLead eq true)}" >
                    <html:submit property="command" value="Add Project" styleClass="button-long-blue" />
                </c:if>
                </td>
            </tr>
        </table>
        </c:if>
        </td>
    </tr>
    <tr>
        <td> </td>
    </tr>
    <tr class="footer">
        <td colspan="2">
        &copy; <bean:message key="virho.copyright" />
        </td>
    </tr>
</table>
</html:form>
</body>
</html>
```

Figure 5.38 *viewProjects.jsp* (view component)

c. The View Image Sets feature under Virho Images module that shows the list of images sets created by the Image Set Coordinators.

```
public class ViewImageSetDetailsForm extends ActionForm {

    private static final long serialVersionUID = -120093072935720016L;
    private int experimentId;
    private int projectSetId;
    private String projectSetName;
    private String projectSetNameDB;
    private String description;
    private String addedBy;
    private String dateAdded;
    private String dateModified;
    private String experimentName;
    private String command;
    private int privilege;
    private ArrayList<String> images;
    private ArrayList<String> imagesDB;
    private ArrayList<ViewExperimentDetailsVO> experiments;

    public void setExperimentId(int experimentId) {
        this.experimentId = experimentId;
    }
```

```
    public int getExperimentId() {
        return experimentId;
    }
    public void setProjectSetId(int projectSetId) {
        this.projectSetId = projectSetId;
    }
    public int getProjectSetId() {
        return projectSetId;
    }
    public void setProjectSetName(String projectSetName) {
        this.projectSetName = projectSetName;
    }
    public String getProjectSetName() {
        return projectSetName;
    }
    public void setProjectSetNameDB(String projectSetNameDB) {
        this.projectSetNameDB = projectSetNameDB;
    }
    public String getProjectSetNameDB() {
        return projectSetNameDB;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public String getDescription() {
        return description;
    }
    public void setAddedBy(String addedBy) {
        this.addedBy = addedBy;
    }
    public String getAddedBy() {
        return addedBy;
    }
    public void setDateAdded(String dateAdded) {
        this.dateAdded = dateAdded;
    }
    public String getDateAdded() {
        return dateAdded;
    }
    public void setDateModified(String dateModified) {
        this.dateModified = dateModified;
    }
    public String getDateModified() {
        return dateModified;
    }
    public void setExperimentName(String experimentName) {
        this.experimentName = experimentName;
    }
    public String getExperimentName() {
        return experimentName;
    }
    public void setCommand(String command) {
        this.command = command;
    }
    public String getCommand() {
        return command;
    }
    public void setPrivilege(int privilege) {
        this.privilege = privilege;
    }
    public int getPrivilege() {
        return privilege;
    }
    public void setImages(ArrayList<String> images) {
        this.images = images;
    }
    public ArrayList<String> getImages() {
        return images;
    }
    public void setImagesDB(ArrayList<String> imagesDB) {
        this.imagesDB = imagesDB;
    }
    public ArrayList<String> getImagesDB() {
        return imagesDB;
    }
    public void setExperiments(ArrayList<ViewExperimentDetailsVO> experiments) {
        this.experiments = experiments;
    }
    public ArrayList<ViewExperimentDetailsVO> getExperiments() {
        return experiments;
    }

    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();

        if (Utilities.isEmpty(this.getProjectSetName())) {
            errors.add("errorMessage", new ActionMessage("error.required.asterisk"));
            errors.add("projectSetName", new ActionMessage(""));
        }

        return errors;
    }

}
```

Figure 5.39 *ViewImageSetDetailsForm.java* (`ActionForm` subclass)

```
16
17 public class ViewImageSetsAction extends Action {
18
19 @Override
20 public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
21     ViewImageSetDetailsForm imageSetForm = (ViewImageSetDetailsForm) form;
22     ViewImageSetDetailsVO imageSetVO = new ViewImageSetDetailsVO();
23     VirhoImagesWorker imageWorker = new VirhoImagesWorker();
24     HttpSession session = request.getSession(true);
25     ArrayList<ViewImageSetDetailsVO> imageSetList = new ArrayList<ViewImageSetDetailsVO>();
26     ArrayList<ViewImageDetailsVO> recentImagesList = new ArrayList<ViewImageDetailsVO>();
27
28     String username = (String)session.getAttribute("userLogin");
29     String forward = "virholex.images.viewImageSets";
30     String roleName = "", project = "";
31     boolean isLead = false;
32
33     if(imageSetForm.getCommand()!=null && imageSetForm.getCommand().equals("Add Image Set")) {
34         forward = "virholex.images.addImageSet";
35     } else if(imageSetForm.getCommand()!=null && imageSetForm.getCommand().equals("My Saved Views")) {
36         forward = "virholex.images.viewSavedViews";
37     } else {
38
39         imageSetList = imageWorker.getImageSets(imageSetVO);
40         recentImagesList = imageWorker.getRecentImages();
41         isLead = imageWorker.isImageSetCoordinator(username);
42
43         for(int i=0; i<imageSetList.size(); i++) {
44             project = ((ViewImageSetDetailsVO)imageSetList.get(i)).getProjectSetName();
45             roleName = imageWorker.getProjectRole(username, project);
46
47             if(roleName.equals(VirholexConstants.IMAGE_SET_COORDINATOR_DESC))
48                 ((ViewImageSetDetailsVO)imageSetList.get(i)).setPrivilege(VirholexConstants.IMAGE_SET_COORDINATOR);
49             else if(roleName.equals(VirholexConstants.IMAGE_SET_CONTRIBUTOR_DESC))
50                 ((ViewImageSetDetailsVO)imageSetList.get(i)).setPrivilege(VirholexConstants.IMAGE_SET_CONTRIBUTOR);
51             else if(roleName.equals(VirholexConstants.RESTRICTED_USER_DESC))
52                 ((ViewImageSetDetailsVO)imageSetList.get(i)).setPrivilege(VirholexConstants.RESTRICTED_USER);
53             else
54                 ((ViewImageSetDetailsVO)imageSetList.get(i)).setPrivilege(VirholexConstants.REGISTERED_USER);
55         }
56
57         request.setAttribute("imageSets", imageSetList);
58         request.setAttribute("recentImages", recentImagesList);
59         request.setAttribute("isLead", isLead);
60
61     }
62
63     PropertyUtils.copyProperties(imageSetForm, imageSetVO);
64
65     session.setAttribute("module", VirholexConstants.VIRHO_IMAGES);
66
67     return mapping.findForward(forward);
68 }
69 }
70
```

Figure 5.40 *ViewImageSetsAction.java* (`Action` subclass)

```
23         <form-bean name="ImageSetsForm" type="com.virholex.images.ViewImageSetDetailsForm"/>
```

Figure 5.41 Form bean declaration for *ViewImageDetailsForm.java*

```
319        <action path="/ViewImageSets"
320                type="com.virholex.images.ViewImageSetsAction"
321                name="ImageSetsForm"
322                scope="request"
323                validate="false">
324            <forward name="virholex.images.viewImageSets" path="/rd_virhoimages/viewImageSets.jsp"/>
325            <forward name="virholex.images.addImageSet" path="/AddImageSet.do"/>
326            <forward name="virholex.images.viewSavedViews" path="/ViewSavedViews.do"/>
327        </action>
```

Figure 5.42 Form handler declaration for View Image Set List

```java

    public static ArrayList<ViewImageSetDetailsVO> getImageSets(int experimentId) throws SQLException, NamingException {

        Connection conn = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        ViewImageSetDetailsVO vo = new ViewImageSetDetailsVO();
        ArrayList<ViewImageSetDetailsVO> imageSetsList = new ArrayList<ViewImageSetDetailsVO>();

        try {
            String sql = "SELECT project_set_name, project_set_id, image_sets.experiment_id FROM image_sets, project_set WHERE " +
                         "image_sets.experiment_id=? AND project_set_id=project_id";
            conn = DBConnect.getConnection();
            ps = conn.prepareStatement(sql);
            ps.setInt(1, experimentId);
            rs = ps.executeQuery();

            while(rs.next()) {
                vo.setProjectSetName(rs.getString(1));
                vo.setProjectSetId(rs.getInt(2));
                vo.setExperimentId(rs.getInt(3));

                imageSetsList.add(vo);
                vo = new ViewImageSetDetailsVO();
            }

        } finally {
            if(rs != null) rs.close();
            if(ps != null) ps.close();
            if(conn != null) conn.close();
        }

        return imageSetsList;
    }
```

Figure 5.43 The `getImageSets()` method in *ImageSetsDAO.java* (Model class)

```jsp
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Virus-Host Interaction Lexicon</title>
<link rel="shortcut icon" href="./images/favicon.ico" >
<link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
<script>
    function loadPage(projectSet, id){
        if(id == 'editImageSet')
            window.open ('EditImageSet.do?projectSet='+projectSet, '_self', false);
        else if(id == 'addImages')
            window.open ('AddImage.do?projectSet='+projectSet, '_self', false);
    }
</script>
</head>
<body>
<html:form action="/ViewImageSets" method="POST" >
    <table class="body" align="center">
        <jsp:include page="/rd_virui/header.jsp" />
        <tr>
            <td> </td>
        </tr>
        <tr valign="top">
            <td class="sidebar">
                <jsp:include page="/rd_virui/sidebar.jsp" />
            </td>
            <td>
            <c:if test="${(userLogin ne null) and ((statusLogin eq 'activated') or (statusLogin eq 'admin') or (statusLogin eq 'superadmin'))}">
            <table class="content">
                <tr>
                    <td class="breadcrumb">
                        <html:link page="/Home.do"><bean:message key="button.virho.home" /></html:link> &#187;
                        <span class="breadcrumb-selected">Virho Image Sets</span>
                    </td>
                </tr>
                <tr>
                    <td> </td>
                </tr>
                <tr>
                    <td class="content-title-form" colspan="2">List of Image Sets</td>
                </tr>
                <tr>
                    <td>
                    <table class="content-desc-form">
                        <tr>
                            <td>
                            <table class="table-content">
                                <tr>
                                    <th>Title</th>
                                    <th>Description</th>
                                </tr>
                            <c:choose>
                                <c:when test="${fn:length(imageSets) gt 0}">
                                    <c:forEach var="detail" items="${imageSets}">
                                    <tr>
                                        <td width="250px">
                                            <html:link page="/ViewImageSetDetails.do?projectSet=${detail.projectSetName}">
```

```
                                    <span class="font-blue">${detail.projectSetName}</span>
                                    </html:link><br/>
                                    <c:if test="${detail.privilege ge 3}" >
                                        <html:submit property="command" value="Add Images" styleClass="button-remove" onclick="loadPage('
                                        <html:submit property="command" value="Edit Image Set" styleClass="button-remove" onclick="loadPa
                                        <c:if test="${detail.privilege eq 4}" >
                                            <c:set var="privilege" value="${detail.privilege}" />
                                        </c:if>
                                    </c:if>
                                </td>
                                <td>${detail.description}</td>
                            </tr>
                            <c:if test="${detail.privilege ge 2}">
                                <c:set var="hasPrivilege" value="true"/>
                            </c:if>
                            </c:forEach>
                        </c:when>
                        <c:otherwise>
                            <tr>
                                <td class="font-italic" colspan="2">No Image Sets Yet</td>
                            </tr>
                        </c:otherwise>
                    </c:choose>
                    </table>
                    </td>
                </tr>
            </table>
            </td>
        </tr>
        <tr>
            <td> </td>
        </tr>
        <tr>
            <td>
            <table class="table-content">
                <tr>
                    <th colspan="5">More Recent Images</th>
                </tr>
                <c:choose>
                    <c:when test="${fn:length(recentImages) gt 0}">
                        <c:forEach var="image" items="${recentImages}" varStatus="loop">
                        <c:if test="${loop.index mod 5 eq 0}">
                            <tr valign="top">
                        </c:if>
                        <td align="center" width="220px" valign="middle">
                            <c:choose>
                                <c:when test="${(hasPrivilege eq true) or (image.accessibility eq 'Public')}">
                                    <html:link page="/ViewImage.do?projectSet=${image.imageSet}&image=${image.imagePathDB}">
                                    <c:choose>
                                        <c:when test="${fn:endsWith(image.imagePathDB, 'exe') or fn:endsWith(image.imagePathDB, 'avi') or
                                            <img src="images/defaultThumb.gif" class="thumbnail" />
                                        </c:when>
                                        <c:when test="${fn:endsWith(image.imagePathDB, 'flv')}">
                                            <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
                                                    codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versior
                                                    width="100" height="75" id="VideoPlayer" align="middle">
                                                <param name="allowScriptAccess" value="*" />
                                                <param name="allowFullScreen" value="true" />
                                                <param name="movie" value="http://www.platipus.nl/flvplayer/download/1.0/FLVPlayer.swf?vi
                                                <param name"quality" value="high" />
                                                <param name="bgcolor" value="#ffffff" />
                                                <embed src="http://www.platipus.nl/flvplayer/download/1.0/FLVPlayer.swf?video=http://www.
                                                    quality="high" bgcolor="#000000" width="100" height="75" name="VideoPlayer" align="mi
                                                    allowScriptAccess="*" allowFullScreen="true" type="application/x-shockwave-flash"
                                                    pluginspage="http://www.macromedia.com/go/getflashplayer" />
                                            </object>
                                        </c:when>
                                        <c:otherwise>
                                            <img src="repository/virhoimages/${image.imagePathDB}" alt="${image.imagePath}" title="${imag
                                        </c:otherwise>
                                    </c:choose>
                                    </html:link>
                                </c:when>
                                <c:otherwise>
                                    <img src="images/restricted.png" class="thumbnail" alt="Restricted Image" title="Restricted Image"/>
                                </c:otherwise>
                            </c:choose>
                        </td>
                        <c:if test="${(loop.index mod 5 eq 4) or (fn:length(recentImages) eq loop.index+1)}">
                            <c:if test="${loop.index mod 5 ne 4}">
                                <c:forEach var="counter" begin="1" end="${4-(loop.index mod 5)}">
                                    <td width="220px"> </td>
                                </c:forEach>
                            </c:if>
                            </tr>
                        </c:if>
                        </c:forEach>
                    </c:when>
                    <c:otherwise>
                        <tr>
                            <td class="font-italic" colspan="2">No Images Yet</td>
                        </tr>
                    </c:otherwise>
                </c:choose>
                <tr>
                    <td class="note" colspan="5">
                        <c:if test="${hasPrivilege ne true}" >
                        Note: You need to be a Restricted User, Image Contributor or Image Coordinator to gain access to the restricted i
                        </c:if>
                    </td>
                </tr>
            </table>
            </td>
        </tr>
        <tr>
            <td> </td>
        </tr>
        <tr>
            <td>    
            <c:if test="${(privilege eq 4) or (isLead eq true)}" >
                <html:submit property="command" value="Add Image Set" styleClass="button-long-blue" />
```

```
                    </c:if>
                    <html:submit property="command" value="My Saved Views" styleClass="button-long-blue" />
                    </td>
                </tr>
            </table>
            </c:if>
            </td>
        </tr>
        <tr>
            <td> </td>
        </tr>
        <tr class="footer">
            <td colspan="2">
            &copy; <bean:message key="virho.copyright" />
            </td>
        </tr>
    </table>
</html:form>
</body>
</html>
```

Figure 5.44 *viewImageSets.jsp* (view component)

d. The Models Taxonomy feature under Virho Models module that contains the list of models taxonomy used to categorize the project models created by Model Coordinators.

```
package com.virholex.models;

import javax.servlet.http.HttpServletRequest;

public class ViewTaxonomyAction extends Action {

@Override
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {

    HttpSession session = request.getSession(true);
    String forward = "virholex.models.viewTaxonomy";

    session.setAttribute("module", VirholexConstants.VIRHO_MODELS);

    return mapping.findForward(forward);
}

}
```

Figure 5.45 *ViewTaxonomyAction.java* (Action subclass)

```
        <action path="/ViewModelsTaxonomy"
                type="com.virholex.models.ViewTaxonomyAction"
                scope="request"
                validate="false">
            <forward name="virholex.models.viewTaxonomy" path="/rd_virhomodels/viewTaxonomy.jsp"/>
        </action>
```

Figure 5.46 Form handler declaration for View Models Taxonomy

Note that `ActionForm` subclass is not required if no form is present on the view component, as shown on Figure 5.47

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Virus-Host Interaction Lexicon</title>
<link rel="shortcut icon" href="./images/favicon.ico" >
<link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
</head>
<body>
    <table class="body" align="center">
        <jsp:include page="/rd_virui/header.jsp" />
        <tr>
            <td> </td>
        </tr>
        <tr valign="top">
            <td class="sidebar">
                <jsp:include page="/rd_virui/sidebar.jsp" />
            </td>
            <td>
            <c:if test="${(userLogin ne null) and ((statusLogin eq 'activated') or (statusLogin eq 'admin') or (statusLogin eq 'superadmin'))}">
            <table class="content">
                <tr>
                    <td class="breadcrumb">
                        <html:link page="/Home.do"><bean:message key="button.virho.home" /></html:link> &#187;
                        <span class="breadcrumb-selected">Models Taxonomy</span>
                    </td>
                </tr>
                <tr>
                    <td> </td>
                </tr>
                <tr>
                    <td class="content-title-form" colspan="2">Models Taxonomy</td>
                </tr>
                <tr>
                    <td>
                    <table class="content-desc-form">
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=cd"><span class="font-blue">Conceptual Diagrammatic Model</span></h
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=cqd"><span class="font-blue">Conceptual Qualitative Discrete Model<
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=cqc"><span class="font-blue">Conceptual Qualitative Continuous Mode
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=cqns"><span class="font-blue">Conceptual Quantitative Stochastic Mc
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=cqnd"><span class="font-blue">Conceptual Quantitative Deterministic
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=ctd"><span class="font-blue">Contextual Diagrammatic Model</span></
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=ctqd"><span class="font-blue">Contextual Qualitative Discrete Model
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=ctqc"><span class="font-blue">Contextual Qualitative Continuous Mod
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=ctqns"><span class="font-blue">Contextual Quantitative Stochastic M
                        </tr>
                        <tr>
                            <td><html:link page="/ViewModels.do?classification=ctqnd"><span class="font-blue">Contextual Quantitative Deterministi
                        </tr>
                    </table>
                    </td>
                </tr>
            </table>
            </c:if>
            </td>
        </tr>
        <tr>
            <td> </td>
        </tr>
        <tr class="footer">
            <td colspan="2">
            &copy; <bean:message key="virho.copyright" />
            </td>
        </tr>
    </table>
</body>
</html>
```

Figure 5.48 *viewTaxonomy.jsp* (view component)

e. The View References Collection feature under Virho References module that displays all collection of references created by the Reference Coordinators.

```
14 public class ViewCollectionDetailsForm extends ActionForm {
15
16     private static final long serialVersionUID = 4577093286852466949L;
17     private String collectionName;
18     private String oldCollectionName;
19     private String description;
20     private String type;
21     private String author;
22     private String downloadable;
23     private String command;
24     private ArrayList<String> letters;
25     private ArrayList<ViewReferenceDetailsVO> references;
26     private int privilege;
27     private boolean hasCollection;
28     private String dateAdded;
29
30     public void setCollectionName(String collectionName) {
31         this.collectionName = collectionName;
32     }
33     public String getCollectionName() {
34         return collectionName;
35     }
36     public void setDescription(String description) {
37         this.description = description;
38     }
39     public String getDescription() {
40         return description;
41     }
42     public void setCommand(String command) {
43         this.command = command;
44     }
45     public String getCommand() {
46         return command;
47     }
48     public void setLetters(ArrayList<String> letters) {
49         this.letters = letters;
50     }
51     public ArrayList<String> getLetters() {
52         return letters;
53     }
54     public void setReferences(ArrayList<ViewReferenceDetailsVO> references) {
55         this.references = references;
56     }
57     public ArrayList<ViewReferenceDetailsVO> getReferences() {
58         return references;
59     }
60     public void setPrivilege(int privilege) {
61         this.privilege = privilege;
62     }
63     public int getPrivilege() {
64         return privilege;
65     }
66     public void setType(String type) {
67         this.type = type;
68     }
69     public String getType() {
70         return type;
71     }
72     public void setAuthor(String author) {
73         this.author = author;
74     }
75     public String getAuthor() {
76         return author;
77     }
78     public void setHasCollection(boolean hasCollection) {
79         this.hasCollection = hasCollection;
80     }
81     public boolean isHasCollection() {
82         return hasCollection;
83     }
84     public void setDownloadable(String downloadable) {
85         this.downloadable = downloadable;
86     }
87     public String getDownloadable() {
88         return downloadable;
89     }
90     public void setOldCollectionName(String oldCollectionName) {
91         this.oldCollectionName = oldCollectionName;
92     }
93     public String getOldCollectionName() {
94         return oldCollectionName;
95     }
96
97     public void setDateAdded(String dateAdded) {
98         this.dateAdded = dateAdded;
99     }
100    public String getDateAdded() {
101        return dateAdded;
102    }
```

```
    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();

        if (Utilities.isEmpty(this.getCollectionName()) || Utilities.isEmpty(this.getDownloadable())) {
            errors.add("errorMessage", new ActionMessage("error.required.asterisk"));

            if(Utilities.isEmpty(this.getCollectionName())) errors.add("collectionName", new ActionMessage(""));
            if(Utilities.isEmpty(this.getDownloadable())) errors.add("downloadable", new ActionMessage(""));

        }

        if(this.isHasCollection()) {
            errors.add("collectionName", new ActionMessage("error.notAvailable", " name"));
        }

        return errors;
    }

}

```

Figure 5.49 *ViewCollectionDetailsForm.java* (`ActionForm` subclass)

```
public class ViewCollectionsAction extends Action {

@Override
public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {
    ViewCollectionDetailsForm collectionForm = (ViewCollectionDetailsForm) form;
    ViewCollectionDetailsVO collectionVO = new ViewCollectionDetailsVO();
    VirhoReferencesWorker referenceWorker = new VirhoReferencesWorker();
    HttpSession session = request.getSession(true);
    ArrayList<String> letterList = new ArrayList<String>();
    ArrayList<ViewCollectionDetailsVO> collectionList = new ArrayList<ViewCollectionDetailsVO>();

    String username = (String)session.getAttribute("userLogin");
    String forward = "virholex.references.viewCollections";
    String letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    String list = "", roleName = "", project = "";
    boolean isLead = false;

    if(collectionForm.getCommand() != null && collectionForm.getCommand().equals("Add Collection")) {
        forward = "virholex.references.addCollection";
    } else {
        for(int i=0; i < letters.length() ; i++){
            if(referenceWorker.getCollections(collectionVO, Character.toString(letters.charAt(i))).size() > 0)
                letterList.add(Character.toString(letters.charAt(i)));
            else
                letterList.add("-");
        }

        collectionVO.setLetters(letterList);

        if(request.getParameter("list") != null)
            list = request.getParameter("list");

        collectionList = referenceWorker.getCollections(collectionVO, list);
        isLead = referenceWorker.isCollectionCoordinator(username);

        for(int i=0; i< collectionList.size(); i++) {
            project = ((ViewCollectionDetailsVO)collectionList.get(i)).getCollectionName() + " " + VirholexConstants.VIRHO_REFERENCE;
            project = project.substring(0, project.lastIndexOf(VirholexConstants.VIRHO_REFERENCE)-1);
            roleName = referenceWorker.getProjectRole(username, project);

            if(roleName.equals(VirholexConstants.COLLECTION_COORDINATOR_DESC))
                ((ViewCollectionDetailsVO)collectionList.get(i)).setPrivilege(VirholexConstants.COLLECTION_COORDINATOR);
            else if(roleName.equals(VirholexConstants.COLLECTION_CONTRIBUTOR_DESC))
                ((ViewCollectionDetailsVO)collectionList.get(i)).setPrivilege(VirholexConstants.COLLECTION_CONTRIBUTOR);
            else if(roleName.equals(VirholexConstants.RESTRICTED_USER_DESC))
                ((ViewCollectionDetailsVO)collectionList.get(i)).setPrivilege(VirholexConstants.RESTRICTED_USER);
            else
                ((ViewCollectionDetailsVO)collectionList.get(i)).setPrivilege(VirholexConstants.REGISTERED_USER);
        }

        request.setAttribute("isLead", isLead);

    }

    PropertyUtils.copyProperties(collectionForm, collectionVO);

    session.setAttribute("module", VirholexConstants.VIRHO_REFERENCES);
    request.setAttribute("collections", collectionList);

    return mapping.findForward(forward);
}

}
```

Figure 5.50 *ViewCollectionsAction.java* (`Action` subclass)

```
26        <form-bean name="CollectionsForm" type="com.virholex.references.ViewCollectionDetailsForm"/>
```

Figure 5.51 Form bean declaration for *ViewCollectionDetailsForm.java*

```
496        <action path="/ViewCollections"
497                type="com.virholex.references.ViewCollectionsAction"
498                name="CollectionsForm"
499                input="/rd_virui/sidebar.jsp"
500                scope="request"
501                validate="false">
502            <forward name="virholex.references.viewCollections" path="/rd_virhoreferences/viewCollections.jsp"/>
503            <forward name="virholex.references.addCollection" path="/rd_virhoreferences/addCollection.jsp"/>
504        </action>
```

Figure 5.52 Form handler declaration for View Reference Collection List

```
46
47  public static ArrayList<ViewCollectionDetailsVO> getCollections(ViewCollectionDetailsVO vo, String letter) throws SQLException, NamingException {
48
49      Connection conn = null;
50      PreparedStatement ps = null;
51      ResultSet rs = null;
52      ArrayList<ViewCollectionDetailsVO> collectionList = new ArrayList<ViewCollectionDetailsVO>();
53
54      try {
55          String sql = "SELECT title, description, downloadable, date_added FROM collections WHERE title LIKE ? ORDER BY title, description";
56          conn = DBConnect.getConnection();
57          ps = conn.prepareStatement(sql);
58          ps.setString(1, letter+"%");
59          rs = ps.executeQuery();
60
61          while(rs.next()) {
62              vo.setCollectionName(rs.getString(1));
63              vo.setDescription(rs.getString(2));
64              vo.setDownloadable(rs.getString(3));
65              vo.setDateAdded(rs.getString(4));
66              collectionList.add(vo);
67              vo = new ViewCollectionDetailsVO();
68          }
69
70      } finally {
71          if(rs != null) rs.close();
72          if(ps != null) ps.close();
73          if(conn != null) conn.close();
74      }
75
76      return collectionList;
77
78  }
```

Figure 5.53 The `getCollections()` method in *ViewCollectionsDAO.java* (Model class)

```
1  <%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
2  <%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
3  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
4  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
5  <html>
6  <head>
7  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
8  <title>Virus-Host Interaction Lexicon</title>
9  <link rel="shortcut icon" href="./images/favicon.ico" >
10 <link rel="stylesheet" type="text/css" href="./files/custom.css" media="screen"/>
11 <script>
12     function loadPage(collection, id){
13         if(id == 'editCollection')
14             window.open ('EditCollection.do?collection='+collection, '_self', false);
15         else if(id == 'addReference')
16             window.open ('AddReference.do?collection='+collection, '_self', false);
17     }
18 </script>
19 </head>
20 <body>
21 <html:form action="/ViewCollections" method="POST" >
22     <table class="body" align="center">
23         <jsp:include page="/rd_virui/header.jsp" />
24         <tr>
25             <td> </td>
26         </tr>
```

Figure 5.55 *viewCollections.jsp* (view component)

VI. DISCUSSION

VirHoLex is an ongoing project that was built on a Java-based platform. Each module of the system was individually created by a group of students from University of the Philippines Manila and De La Salle University. However, the lack of version control system made the merging of the codes difficult since the merging was manually done. Hence, a number of system bugs were encountered during the system run through, prohibiting the users to fully utilize the VirHoLex system. Simple refactoring of each module is not sufficed to address the issues on maintainability and reusability of the source codes. A framework is necessary to preserve the credibility of the system.

Struts framework was incorporated into the second version of VirHoLex. Struts follows the MVC design pattern that promotes separation of concerns, i.e. MVC separates responsibilities into three layers of functionality: a) Model that holds the business logic and state; b) View that displays data from Model to user; and c) Controller that handles all requests from the user and selects the view to return. Hence, application of Struts into VirHoLex made the codes reusable and the system easier to maintain. Also it is much easier to make form validations since Struts caters a method that does simple checking, and a tag that displays error messages. Unlike the initial system, there are no complications in tracking of files that are used for certain functionality since all files are listed in a single Struts' configuration file (*struts-config.xml*). The drawback of employing Struts is that much more files are needed to create and maintain compared to the basic servlets. Subclasses for `ActionForm` and `Action` are required per functionality, aside from the jsp pages, properties files and other model classes, making the Struts a little bit confusing at first. Since VirHoLex

is quite a big system, there was a difficulty in code migration which took ample of time to be completed. Nonetheless, with the help of Struts VirHoLex has greatly improved code-wise and functionality-wise. Future developers can add a new functionality into the system with less trouble since a certain design pattern is being followed.

Moreover, VirHoLex is now using SVN repository in Agila that allows the developers working on the system to easily manage and track the history of modifications committed to the files. They simply need to check out the project from the repository to have their working copy, and then synchronize their working copy with the repository to check for differences, mitigating the task of manually inspecting for conflicting line of codes.

Deployment of VirHoLex to MaxPlanck server has a critical importance to the study since the system will be of no use if it cannot be run on the client's server. The system was first test on the localhost then on Agila server to check if the system is working before deploying it on MaxPlanck server. Knowledge on the usage of terminal emulator, SCP and VPN client is of great advantage so that the developer will not be lost on how these programs work and can make the process of deployment run smoothly.

VII. CONCLUSION

VirHoLex is a dynamic repository of biological information on viruses that supports analyses and modeling of experimental data, as well as storage of bibliographic references and images. It consists of seven modules working together to produce desired results necessary for the researches. To make the system fully functional, it underwent debugging process in order to remove and fix all showstoppers. Bugs that were identified during debugging were properly documented in the Mantis Bug Tracking system. Subversion was also used to easily track the changes made on the files. Struts framework was then applied to the system in order to make the codes maintainable and reusable so that a new set of functionality can easily be integrated into the system. With the help of Struts, VirHoLex has greatly improved code-wise and functionality-wise though there is still a room of the improvement for the system.

VIII. RECOMMENDATION

Although the program structure of VirHoLex has greatly improved, it is advisable to further maximize the benefits the Struts is providing to ensure the credibility of the system. Struts caters *properties files* that allow internationalization or localization of the webapp, that is, the View component can be displayed in multiple languages. This is essential to those people who barely understand English and prefer to use their native languages. Struts plug-ins like Tiles and Validator can also be implemented to enhance the system's functionality. Tiles can easily provide a uniform layout for the webapp, and it is flexible enough to display dynamic content – content that can change depending on user interaction or on external data sources. On the other hand, Validator can provide a set of generic validations that can be used to run a variety of simple validations. Instead of implementing `validate()` function in `ActionForm` subclass, validations can be declared in an XML file. This can lessen the code duplication and can make the validations easier to maintain.

In addition, VirHoLex can be optimized by minimizing the use of session scopes in the source codes. Using many session scopes can affect the system's performance especially if there are many clients using the webapp. Also the server can only handle a limited number of session scopes at a time. Cache can be used to optimize the system. It stores the data so that future request for that data can be served faster. The use of cache can incrementally increase system's speed since this will also reduce the number of times database connection is called.

IX. BIBLIOGRAPHY

[1] Definition of Virus. (2011). Retrieve from http://www.medterms.com/script/main/art.asp?articlekey=5997. Last accessed August 11, 2011.

[2] Virus. Retrieve from http://en.wikipedia.org/wiki/Virus. Last accessed August 11, 2011.

[3] Hughes, G. (2011). *U.N. warns on mutant strain of bird flu virus*. CNN. Retrieve from http://www.cnn.com/2011/HEALTH/08/30/un.bird.flu/. Last accessed September 01, 2011.

[4] VirHoLex Team. (2009). VirHoLex Functional Specifications for Release 1.0.

[5] Breitbart, M. & Rohwer, F. (2005). *Here a virus, there a virus, everywhere the same virus?*. Trends Microbiol: pp 278–84

[6] Büchen-Osmond, C. (2003). *The Universal Virus Database ICTVDB*. Computing in Science and Engineering: pp. 16-25.

[7] Reddy, V., Nataraja, P., Okerberg, O. *et al.* (2001). *Virus Particle Explorer (VIPER), a Website for Virus Capsid Structures and Their Computational Analyses*. Journal of Virology: pp 11943-11947.

[8] Shepherd, C., Borelli, I., Lander, G. *et al.* (2006). *VIPERdb: a relational database for structural virology*. Nucleic Acids Research Vol. 34: D386-D389 (Database Issue).

[9] Pellet, J., Tafforeau, L., Lucas-Hourani, M. *et al.* (2010). *ViralORFeome: an integrated database to generate a versatile collection of viral ORFs*. Nucleic Acids Research Vol. 38: D371-D378 (Database Issue).

[10] Belshaw, R., de Oliveira, T., Markowitz, S. & Rambaut, A. (2009). *The RNA Virus Database*. Nucleic Acids Vol. 37: D431-D435 (Database Issue).

[11] Palmeira, L., Penel, S., Lotteau, V. *et al.* (2010). *PhEVER: a database for the global exploration of virus-host evolutionary relationships*. Nucleic Acids Research Vol. 39: D569-D575 (Database Issue).

[12] Chatr-aryamontri, A., Ceol, A., Peluso, D. *et al.* (2009). *VirusMINT: a viral protein interaction database*. Nucleic Acids Research Vol. 37: D669-D673 (Database Issue).

[13] Navratil, V., de Chassey, B., Meynie, L. *et al.* (2009). *VirHostNet: a knowledge based for the management and the analysis of proteome-wide virus-host interaction networks*. Nucleic Acids Research Vol. 37:D661-D668 (Database Issue).

[14] Liechti, R., Gleizes, A., Kuznetsov, D. *et al.* (2010). *OpenFluDB: a database for human and animal influenza virus*". (Database Issue).

[15] Squires, B., Macken, C., Garcia-Sastre, A. *et al.* (2008). *BioHealthBase: informatics support in the elucidation of influenza virus host–pathogen interactions and virulence*. Nucleic Acids Research Vol. 36: D497-D503 (Database Issue).

[16] Zhou, C. E., Smith, J., Lam, M. *et al.* (2007). *MvirDB: a microbial database of protein toxins, virulence factors and antibiotic resistance genes for bio-defense application.* Nucleic Acids Research Vol. 35: D391-D394 (Database Issue).

[17] Dionisio, A. (2009). *Virho Image Module* (Thesis). University of the Philippines Manila.

[18] Yao, S. (2009).*Virus-Host Interaction Lexicon User Interface* (Thesis). University of the Philippines Manila.

[19] Elepaño, R. A. (2009). *Virho Registered User Services Module (ViRUS) and Virho Hotspots* (Thesis). University of the Philippines Manila.

[20] Cariño, M. L. (2009). *Virho Information Services Module* (Thesis). University of the Philippines Manila.

[21] Antonio, A. (2009). *Virho Models Module* (Thesis). University of the Philippines Manila.

[22] Morales, J. J. (2009). *Refactoring VirHoLex: References and Image Hotspots Modules* (Thesis). University of the Philippines Manila.

[23] Sommerville, I. (2007). *Configuration Management*. Software Engineering 8th Edition, Addison-Wesley.

[24] *IEEE Standard for Software Configuration Management Plans*. IEEE Std 828™ - 2005.

[25] Collins-Sussman, B., Fitzpatrick, B., Pilato, C. M. (2011). *Version Control with Subversion—for Subversion 1.7*. Retrieved from http://svnbook.redbean.com/nightly/en/svn-book.html#svn.intro.whatis. Last accessed September 12, 2011.

[26] Basham, B., Sierra, K., Bates, B. (2008). *Head First Servlet & JSP 2nd Edition*. O'Reilly Media, Inc.

[27] Doray, A. (2006). *Beginning Apache Struts: From Novice to Professional*. Apress The Expert's Voice.

[28] Robinson, M., Finkelstein, E. (2004). *Jakarta Struts for Dummies*. Wiley Publishing Inc.

X. APPENDIX

A. xml and properties files

ApplicationResource.properties

```
label.common.username = <b>Username</b>
label.common.password = <b>Password</b>
label.common.repassword = <b>Retype
Password</b>
label.common.oldPassword = <b>Old Password</b>
label.common.newPassword = <b>New Password</b>
label.common.renewPassword = <b>Retype New
Password</b>
label.common.reNewPassword = <b>Retype New
Password</b>
label.common.email = <b>Email Address</b>
label.common.firstName = <b>First Name</b>
label.common.name = <b>Name</b>
label.common.lastName = <b>Last Name</b>
label.common.affiliation = <b>Affiliation</b>
label.common.jobTitle = <b>Job Title</b>

label.common.project = <b>Project Name</b>
label.common.description = <b>Description</b>
label.common.status = <b>Status</b>
label.common.author = <b>Author</b>

link.common.login = Login
link.common.register = Register
link.common.forgotPassword = Forgot Password
link.common.request = Request
link.common.member = Member
link.common.project = My Project
link.common.profile = My Profile
link.common.logout = Logout

button.virho.home = Home
button.virho.contactUs = Contact Us
button.virho.experiments = Virho Experiments
button.virho.images = Virho Images
button.virho.models = Virho Models
button.virho.references = Virho References
button.virho.externalInfo = Virho External
Info
virho.externalSearch = Virho External Search
virho.externalAltSearch = Virho Alternative
External Search
virho.copyright = Copyright Virus-Host
Interaction Lexicon 2012

error.required.all = All fields are required
error.required.asterisk = All fields with
asterisk are required
error.minlength = {0} cannot be less than 6
characters
error.required = {0} is required
error.notEqual.password = Passwords entered do
not match
error.invalid = {0} is invalid
error.invalid.login = Invalid username or
password<br/><br/>
error.invalid.account = <br/>Invalid username
or email address<br/><br/>
error.available.username = {0} is still
available<br/><br/>
error.empty = <br/>Please enter your
{0}<br/><br/>
error.empty2 = Please enter your {0}
error.select.default = Please select value for
{0}
error.notAvailable = This {0} is not available
error.default = Please select {0}
error.file = {0} is not a PDF file
error.invalid.image = Invalid image file
error.image = You must upload an image file
with one of the following extensions: jpg,
jpeg, gif, png, bmp

link.search.ncbi =
http://www.ncbi.nlm.nih.gov/sites/entrez?
link.search.ncbi.alt1 =
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?
cmd=search
link.search.ncbi.alt2 =
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?
cmd=retrieve
link.search.ncbi.alt3 =
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?
cmd=link
link.search.kegg = http://www.genome.jp/dbget-
bin/www_bfind_sub?mode=bfind&max_hit=1000&serv
=kegg&dbkey=
link.search.ebi =
http://www.ebi.ac.uk/ebisearch/search.ebi?db=
```

context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/DataSource">
  <Resource
    auth="Container"
    driverClassName="com.mysql.jdbc.Driver"
    name="jdbc/virholexPool"
    type="javax.sql.DataSource"
    url="jdbc:mysql://bioio1.opt.rzg.mpg.de/Vir
    holex2"
    username="Virholex2"
    password="d7AmyfpXHQMjAyVb"
  />
</Context>
```

struts-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE struts-config PUBLIC
"-//Apache Software Foundation//DTD Struts
Configuration 1.2//EN"
"http://jakarta.apache.org/struts/dtds/struts-
config_1_2.dtd">

<struts-config>

  <form-beans>
    <form-bean name="RegistrationForm"
    type="com.virholex.virus.RegisterAccountFor
    m"/>
    <form-bean name="LoginForm"
    type="com.virholex.virus.LoginAccountForm"/
    >
    <form-bean name="ForgotPasswordForm"
    type="com.virholex.virus.ForgotPasswordForm
    "/>
    <form-bean name="AccountRequestForm"
    type="com.virholex.virus.AccountRequestForm
    "/>
    <form-bean name="UserProfileForm"
    type="com.virholex.virus.UserProfileForm"/>
    <form-bean name="EditUserProfileForm"
    type="com.virholex.virus.EditUserProfileFor
    m"/>
    <form-bean name="CancelAccountForm"
    type="com.virholex.virus.CancelAccountForm"
    />
```

```
  <form-bean name="ViewUsersForm"
  type="com.virholex.virus.ViewUsersForm"/>
  <form-bean name="RequestPrivilegeForm"
  type="com.virholex.virus.RequestPrivilegeFo
  rm"/>
  <form-bean name="ModelDetailsForm"
  type="com.virholex.models.ViewModelDetailsF
  orm"/>
  <form-bean name="EditUserPrivilegeForm"
  type="com.virholex.models.EditUserPrivilege
  Form"/>
  <form-bean name="ProjectDetailsForm"
  type="com.virholex.experiments.ViewProjectD
  etailsForm"/>
  <form-bean name="ExperimentDetailsForm"
  type="com.virholex.experiments.ViewExperime
  ntDetailsForm"/>
  <form-bean name="ExperimentComponentsForm"
  type="com.virholex.experiments.ExperimentCo
  mponentsForm"/>
  <form-bean name="ImageSetsForm"
  type="com.virholex.images.ViewImageSetDetai
  lsForm"/>
  <form-bean name="ImageForm"
  type="com.virholex.images.ViewImageDetailsF
  orm"/>
  <form-bean name="SavedViewForm"
  type="com.virholex.images.ViewSavedViewDeta
  ilsForm"/>
  <form-bean name="CollectionsForm"
  type="com.virholex.references.ViewCollectio
  nDetailsForm"/>
  <form-bean name="ReferenceDetailsForm"
  type="com.virholex.references.ViewReference
  DetailsForm"/>
  <form-bean name="VirusForm"
  type="com.virholex.basicinfo.VirusForm"/>
  <form-bean name="SearchForm"
  type="com.virholex.internalsearch.SearchFor
  m"/>
</form-beans>

<global-forwards>
  <forward name="virholex.virui.home"
  path="/Home.do" />
</global-forwards>

<action-mappings>

<!-- VIRUI -->

<action path="/Home"
type="org.apache.struts.actions.ForwardActio
n" parameter="/rd_virui/index.jsp"
scope="request" validate="false">
</action>

<action path="/VirhoExperiments"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virui/virhoExperimentsDescrip
tion.jsp" input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/VirhoImages"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virui/virhoImagesDescription.
jsp" input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/VirhoModels"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virui/virhoModelsDescription.
jsp" input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/VirhoReferences"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virui/virhoReferencesDescript
ion.jsp" input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/ContactUs"
type="org.apache.struts.actions.ForwardActio
n" parameter="/rd_virui/contactUs.jsp"
input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<!-- VIRUS -->

<action path="/VirholexPrivileges"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virus/WhyShouldIRegister.jsp"
input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/Register"
type="com.virholex.virus.RegisterAccountActi
on" name="RegistrationForm" scope="request"
validate="false">
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
  <forward name="virholex.virus.registration"
  path="/rd_virus/registerAccount.jsp"/>
  <forward name="virholex.virui.home"
  path="/Home.do"/>
</action>

<action path="/Login"
type="com.virholex.virus.LoginAccountAction"
name="LoginForm" scope="request"
validate="false">
  <forward name="virholex.virui.home"
  path="/Home.do"/>
  <forward name="virholex.virus.login"
  path="/rd_virus/loginAccount.jsp"/>
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
  <forward
  name="virholex.virus.suspendedAccount"
  path="/rd_virus/reactivateAccount.jsp"/>
</action>

<action path="/ForgotPassword"
type="com.virholex.virus.ForgotPasswordActio
n" name="ForgotPasswordForm" scope="request"
validate="false">
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
  <forward
  name="virholex.virus.forgotPassword"
  path="/rd_virus/forgotPassword.jsp"/>
</action>

<action path="/ResetPassword"
type="com.virholex.virus.ResetPasswordAction
" name="ForgotPasswordForm" scope="request"
validate="false">
  <forward
  name="virholex.virus.resetPassword"
  path="/rd_virus/resetPassword.jsp"/>
```

```
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
</action>

<action path="/AccountRequest"
type="com.virholex.virus.AccountRequestActio
n" name="AccountRequestForm" scope="request"
validate="false">
  <forward
  name="virholex.virus.accountRequest"
  path="/rd_virus/accountRequest.jsp"/>
</action>

<action path="/AccountRequestEvaluate"
type="com.virholex.virus.AccountRequestActio
n" name="AccountRequestForm" scope="request"
validate="false">
  <forward
  name="virholex.virus.accountRequestEvaluate
  "
  path="/rd_virus/accountRequestEvaluate.jsp"
  />
  <forward
  name="virholex.virus.accountRequestConfig"
  path="/rd_virus/accountRequestConfig.jsp"/>
  <forward
  name="virholex.virus.accountRequest"
  path="/AccountRequest.do"/>
  <forward
  name="virholex.virus.accountRequestFinal"
  path="/rd_virus/accountRequestFinal.jsp"/>
</action>

<action path="/AccountRequestConfig"
type="com.virholex.virus.AccountRequestActio
n" name="AccountRequestForm" scope="request"
validate="false">
  <forward
  name="virholex.virus.accountRequestConfig"
  path="/rd_virus/accountRequestConfig.jsp"/>
  <forward
  name="virholex.virus.accountRequestFinal"
  path="/rd_virus/accountRequestFinal.jsp"/>
  </action>
  <action path="/ReactivateAccount"
  type="com.virholex.virus.LoginAccountAction
  " name="LoginForm" scope="request"
  validate="false">
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
  <forward name="virholex.virui.home"
  path="/rd_virui/index.jsp"/>
</action>

<action path="/MyProjects"
type="com.virholex.virus.UserProjectsAction"
input="/rd_virui/header.jsp" scope="request"
validate="false">
  <forward name="virholex.virus.userProjects"
  path="/rd_virus/userProjects.jsp"/>
</action>

<action path="/ProjectRequest"
type="com.virholex.virus.ViewProjectRequestA
ction" name="RequestPrivilegeForm"
scope="request" validate="false">
  <forward
  name="virholex.virus.projectRequest"
  path="/rd_virus/viewProjectRequest.jsp"/>
  <forward
  name="virholex.virus.projectRequestEvaluate
  "
  path="/rd_virus/projectRequestEvaluate.jsp"
  />
  <forward name="virholex.virus.userProjects"
  path="/MyProjects.do" redirect="true"/>
</action>

<action path="/VirholexUsers"
type="com.virholex.virus.ViewUsersAction"
name="ViewUsersForm"
input="/rd_virui/header.jsp" scope="request"
validate="false">
  <forward name="virholex.virus.viewUsers"
  path="/rd_virus/viewUsers.jsp"/>
</action>

<action path="/UserProfileDetails"
type="com.virholex.virus.ViewUserProfileDeta
ilsAction" name="UserProfileForm"
input="/rd_virus/viewUsers.jsp"
scope="request" validate="false">
  <forward
  name="virholex.virus.viewUserDetail"
  path="/rd_virus/viewUserProfileDetails.jsp"
  />
  <forward name="virholex.virus.viewUsers"
  path="/VirholexUsers.do?list="/>
</action>

<action path="/MyProfile"
type="com.virholex.virus.UserProfileAction"
name="UserProfileForm"
input="/rd_virui/header.jsp" scope="request"
validate="false">
  <forward name="virholex.virus.userProfile"
  path="/rd_virus/userProfile.jsp"/>
  <forward name="virholex.virus.editProfile"
  path="/EditProfile.do"/>
  <forward
  name="virholex.virus.cancelAccount"
  path="/UnsubscribeAccount.do"/>
</action>

<action path="/EditProfile"
type="com.virholex.virus.EditUserProfileActi
on" name="EditUserProfileForm"
scope="request" validate="false">
  <forward name="virholex.virus.editProfile"
  path="/rd_virus/editUserProfile.jsp"/>
  <forward name="virholex.virus.userProfile"
  path="/MyProfile.do"/>
</action>

<action path="/UnsubscribeAccount"
type="com.virholex.virus.CancelAccountAction
" name="CancelAccountForm" scope="request"
validate="false">
  <forward
  name="virholex.virus.cancelAccount"
  path="/rd_virus/cancelAccount.jsp"/>
  <forward name="virholex.virus.statusCheck"
  path="/rd_virus/statusAccountMessage.jsp"/>
</action>

<action path="/Logout"
type="com.virholex.virus.LogoutAccountAction
" scope="request" validate="false">
  <forward name="virholex.virus.login"
  path="/Login.do"/>
</action>

<action path="/PrivilegeRequest"
type="com.virholex.virus.RequestPrivilegeAct
ion" name="RequestPrivilegeForm"
scope="request" validate="false">
  <forward
  name="virholex.virus.requestPrivilege"
  path="/rd_virus/requestPrivilege.jsp"/>
  <forward
  name="virholex.virus.requestPrivilegeFinal"
```

```
  path="/rd_virus/requestPrivilegeFinal.jsp"/
  >
  <forward
  name="virholex.experiments.viewProject"
  path="/ViewProjectExperimentDetails.do"
  redirect="true"/>
  <forward name="virholex.images.viewProject"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
  <forward name="virholex.models.viewProject"
  path="/ViewProjectDetails.do"
  redirect="true"/>
  <forward
  name="virholex.references.viewProject"
  path="/ViewReferences.do" redirect="true"/>
  <forward
  name="virholex.basicinfo.viewProject"
  path="/ViewVirusDetails.do"
  redirect="true"/>
</action>

<!-- VIRHO EXPERIMENTS -->

<action path="/ViewProjects"
type="com.virholex.experiments.ViewProjectsA
ction" name="ProjectDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.viewProjects"
  path="/rd_virhoexperiments/viewProjects.jsp
  "/>
  <forward
  name="virholex.experiments.addProject"
  path="/rd_virhoexperiments/addProject.jsp"/
  >
</action>

<action path="/ViewProjectExperimentDetails"
type="com.virholex.experiments.ViewProjectDe
tailsAction" name="ProjectDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.viewProjectDetai
  ls"
  path="/rd_virhoexperiments/viewProjectDetai
  ls.jsp"/>
  <forward
  name="virholex.experiments.editProject"
  path="/EditProject.do"/>
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward
  name="virholex.experiments.viewUsers"
  path="/VirhoExperimentsUsers.do?list="
  redirect="true"/>
  <forward
  name="virholex.experiments.addExperiment"
  path="/AddExperiment.do"/>
  <forward
  name="virholex.experiments.viewProjects"
  path="/ViewProjects.do" redirect="true"/>
</action>

<action path="/ViewExperimentDetails"
type="com.virholex.experiments.ViewExperimen
tDetailsAction" name="ExperimentDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.viewExperimentDe
  tails"
  path="/rd_virhoexperiments/viewExperimentDe
  tails.jsp"/>
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward
  name="virholex.experiments.viewProjectDetai
  ls" path="/ViewProjectExperimentDetails.do"
  redirect="true"/>
  <forward
  name="virholex.experiments.editExperiment"
  path="/EditExperiment.do"/>
  <forward
  name="virholex.experiments.addExperimentCom
  ponent" path="/AddExperimentComponent.do"
  redirect="true"/>
</action>

<action path="/AddProject"
type="com.virholex.experiments.AddProjectAct
ion" name="ProjectDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.viewProjects"
  path="/ViewProjects.do"/>
  <forward
  name="virholex.experiments.addProject"
  path="/rd_virhoexperiments/addProject.jsp"/
  >
  <forward
  name="virholex.experiments.viewProjectDetai
  ls" path="/ViewProjectExperimentDetails.do"
  redirect="true"/>
</action>

<action path="/EditProject"
type="com.virholex.experiments.EditProjectAc
tion" name="ProjectDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.viewProjects"
  path="/ViewProjects.do"/>
  <forward
  name="virholex.experiments.editProject"
  path="/rd_virhoexperiments/editProject.jsp"
  />
  <forward
  name="virholex.experiments.viewProjectDetai
  ls" path="/ViewProjectExperimentDetails.do"
  redirect="true"/>
</action>

<action path="/AddExperiment"
type="com.virholex.experiments.AddExperiment
Action" name="ExperimentDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.addExperiment"
  path="/rd_virhoexperiments/addExperiment.js
  p"/>
  <forward
  name="virholex.experiments.viewProjectDetai
  ls" path="/ViewProjectExperimentDetails.do"
  redirect="true"/>
  <forward
  name="virholex.experiments.viewExperimentDe
  tails" path="/ViewExperimentDetails.do"/>
</action>

<action path="/EditExperiment"
type="com.virholex.experiments.EditExperimen
tAction" name="ExperimentDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.editExperiment"
  path="/rd_virhoexperiments/editExperiment.j
  sp"/>
  <forward
  name="virholex.experiments.viewExperimentDe
  tails" path="/ViewExperimentDetails.do"
  redirect="true"/>
```

```
</action>

<action path="/AddExperimentComponent"
type="com.virholex.experiments.AddExperiment
ComponentAction"
name="ExperimentComponentsForm"
scope="request" validate="false">
  <forward
  name="virholex.experiments.addExperimentCom
  ponent"
  path="/rd_virhoexperiments/addExperimentCom
  ponent.jsp"/>
  <forward
  name="virholex.experiments.viewExperimentDe
  tails" path="/ViewExperimentDetails.do"
  redirect="true"/>
</action>

<action path="/VirhoExperimentsUsers"
type="com.virholex.experiments.ViewUsersActi
on" name="ViewUsersForm" scope="request"
validate="false">
  <forward
  name="virholex.experiments.viewUsers"
  path="/rd_virhoexperiments/viewUsers.jsp"/>
</action>

<!-- VIRHO IMAGES -->

<action path="/ViewImageSets"
type="com.virholex.images.ViewImageSetsActio
n" name="ImageSetsForm" scope="request"
validate="false">
  <forward
  name="virholex.images.viewImageSets"
  path="/rd_virhoimages/viewImageSets.jsp"/>
  <forward name="virholex.images.addImageSet"
  path="/AddImageSet.do"/>
  <forward
  name="virholex.images.viewSavedViews"
  path="/ViewSavedViews.do"/>
</action>

<action path="/ViewImageSetDetails"
type="com.virholex.images.ViewImageSetDetail
sAction" name="ImageSetsForm"
scope="request"  validate="false">
  <forward
  name="virholex.images.viewImageSet"
  path="/rd_virhoimages/viewImageSetDetails.j
  sp"/>
  <forward
  name="virholex.images.viewSavedViews"
  path="/ViewSavedViews.do"/>
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward name="virholex.images.viewUsers"
  path="/VirhoImagesUsers.do?list="/>
  <forward
  name="virholex.images.addSavedView"
  path="/AddSavedView.do"/>
  <forward name="virholex.images.addImage"
  path="/AddImage.do"/>
  <forward
  name="virholex.images.viewImageSets"
  path="/ViewImageSets.do"/>
  <forward
  name="virholex.images.editImageSet"
  path="/EditImageSet.do"/>
</action>

<action path="/ViewSavedViews"
type="com.virholex.images.ViewSavedViewsActi
on" name="SavedViewForm" scope="request"
validate="false">
  <forward
  name="virholex.images.viewSavedViews"
  path="/rd_virhoimages/viewSavedViews.jsp"/>
</action>

<action path="/AddImageSet"
type="com.virholex.images.AddImageSetAction"
name="ImageSetsForm" scope="request"
validate="false">
  <forward name="virholex.images.addImageSet"
  path="/rd_virhoimages/addImageSet.jsp"/>
  <forward
  name="virholex.images.viewImageSets"
  path="/ViewImageSets.do"/>
  <forward
  name="virholex.images.viewImageSetDetais"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
</action>

<action path="/EditImageSet"
type="com.virholex.images.EditImageSetAction
" name="ImageSetsForm" scope="request"
validate="false">
  <forward
  name="virholex.images.editImageSet"
  path="/rd_virhoimages/editImageSet.jsp"/>
  <forward
  name="virholex.images.viewImageSetDetais"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
</action>

<action path="/ViewImage"
type="com.virholex.images.ViewImageDetailsAc
tion" name="ImageForm" scope="request"
validate="false">
  <forward
  name="virholex.images.viewImageDetails"
  path="/rd_virhoimages/viewImageDetails.jsp"
  />
  <forward
  name="virholex.images.viewSlideshow"
  path="/rd_virhoimages/viewImagesSlideshow.j
  sp"/>
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward name="virholex.images.editImage"
  path="/EditImage.do"/>
  <forward
  name="virholex.images.viewImageSetDetais"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
</action>

<action path="/AddImage"
type="com.virholex.images.AddImageAction"
name="ImageForm" scope="request"
validate="false">
  <forward name="virholex.images.addImage"
  path="/rd_virhoimages/addImage.jsp"/>
  <forward
  name="virholex.images.viewImageSetDetais"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
</action>

<action path="/EditImage"
type="com.virholex.images.EditImageAction"
name="ImageForm" scope="request"
validate="false">
  <forward name="virholex.images.editImage"
  path="/rd_virhoimages/editImage.jsp"/>
```

```
  <forward
  name="virholex.images.viewImageDetails"
  path="/ViewImage.do" redirect="true"/>
</action>

<action path="/SavedViewDetails"
type="com.virholex.images.ViewSavedViewDetai
lsAction" name="SavedViewForm"
scope="request" validate="false">
  <forward
  name="virholex.images.savedViewDetails"
  path="/rd_virhoimages/viewSavedViewDetails.
  jsp"/>
  <forward
  name="virholex.images.editSavedView"
  path="/EditSavedView.do"/>
  <forward
  name="virholex.images.viewSavedViews"
  path="/ViewSavedViews.do"/>
</action>

<action path="/AddSavedView"
type="com.virholex.images.AddSavedViewAction
" name="SavedViewForm" scope="request"
validate="false">
  <forward
  name="virholex.images.addSavedView"
  path="/rd_virhoimages/addSavedView.jsp"/>
  <forward
  name="virholex.images.savedViewDetails"
  path="/SavedViewDetails.do"
  redirect="true"/>
  <forward
  name="virholex.images.viewImageSet"
  path="/ViewImageSetDetails.do"
  redirect="true"/>
</action>

<action path="/EditSavedView"
type="com.virholex.images.EditSavedViewActio
n" name="SavedViewForm" scope="request"
validate="false">
  <forward
  name="virholex.images.editSavedView"
  path="/rd_virhoimages/editSavedView.jsp"/>
  <forward
  name="virholex.images.savedViewDetails"
  path="/SavedViewDetails.do"
  redirect="true"/>
</action>

<action path="/ViewSavedViewImage"
type="com.virholex.images.ViewSavedViewImage
Action" name="SavedViewForm" scope="request"
validate="false">
  <forward
  name="virholex.images.viewSavedViewImage"
  path="/rd_virhoimages/viewSavedViewImage.js
  p"/>
</action>

<action path="/VirhoImagesUsers"
type="com.virholex.images.ViewUsersAction"
name="ViewUsersForm" scope="request"
validate="false">
  <forward name="virholex.images.viewUsers"
  path="/rd_virhoimages/viewUsers.jsp"/>
</action>

<!-- VIRHO MODELS -->

<action path="/ViewModelsTaxonomy"
type="com.virholex.models.ViewTaxonomyAction
" scope="request" validate="false">
  <forward
  name="virholex.models.viewTaxonomy"
  path="/rd_virhomodels/viewTaxonomy.jsp"/>
</action>

<action path="/ViewModels"
type="com.virholex.models.ViewModelsAction"
name="ProjectDetailsForm"
input="/rd_virhomodels/viewTaxonomy.jsp"
scope="request" validate="false">
  <forward name="virholex.models.viewModels"
  path="/rd_virhomodels/viewModels.jsp"/>
</action>

<action path="/ViewProjectDetails"
type="com.virholex.models.ViewProjectDetails
Action" name="ProjectDetailsForm"
input="/rd_virhomodels/viewModels.jsp"
scope="request" validate="false">
  <forward name="virholex.models.viewProject"
  path="/rd_virhomodels/viewProjectDetails.js
  p"/>
  <forward name="virholex.models.addModel"
  path="/AddModelDetails.do"/>
  <forward name="virholex.models.viewUsers"
  path="/ViewProjectUsers.do"/>
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward name="virholex.models.viewUsers"
  path="/VirhoModelsUsers.do?list="/>
</action>

<action path="/ViewModelDetails"
type="com.virholex.models.ViewModelDetailsAc
tion" name="ModelDetailsForm"
input="/rd_virhomodels/viewModels.jsp"
scope="request" validate="false">
  <forward name="virholex.models.viewModel"
  path="/rd_virhomodels/viewModelDetails.jsp"
  />
  <forward name="virholex.models.viewProject"
  path="/ViewProjectDetails.do"
  redirect="true"/>
</action>

<action path="/AddModelDetails"
type="com.virholex.models.AddModelDetailsAct
ion" name="ModelDetailsForm" scope="request"
validate="false">
  <forward name="virholex.models.addModel"
  path="/rd_virhomodels/addModelDetails.jsp"/
  >
  <forward name="virholex.models.viewProject"
  path="/ViewProjectDetails.do"
  redirect="true"/>
  <forward name="virholex.models.viewModel"
  path="/ViewModelDetails.do"
  redirect="true"/>
</action>

<action path="/EditModelDetails"
type="com.virholex.models.EditModelDetailsAc
tion" name="ModelDetailsForm"
scope="request" validate="false">
  <forward name="virholex.models.editModel"
  path="/rd_virhomodels/editModelDetails.jsp"
  />
  <forward name="virholex.models.viewModel"
  path="/ViewModelDetails.do"
  redirect="true"/>
</action>

<action path="/VirhoModelsUsers"
type="com.virholex.models.ViewUsersAction"
```

```
name="ViewUsersForm" scope="request"
validate="false">
  <forward name="virholex.models.viewUsers"
  path="/rd_virhomodels/viewUsers.jsp"/>
</action>

<!-- VIRHO REFERENCES -->

<action path="/ViewCollections"
type="com.virholex.references.ViewCollection
sAction" name="CollectionsForm"
input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
  <forward
  name="virholex.references.viewCollections"
  path="/rd_virhoreferences/viewCollections.j
  sp"/>
  <forward
  name="virholex.references.addCollection"
  path="/rd_virhoreferences/addCollection.jsp
  "/>
</action>

<action path="/ViewReferences"
type="com.virholex.references.ViewReferences
Action" name="CollectionsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.viewReferences"
  path="/rd_virhoreferences/viewReferences.js
  p"/>
  <forward
  name="virholex.virus.privilegeRequest"
  path="/PrivilegeRequest.do"/>
  <forward
  name="virholex.references.editCollection"
  path="/EditCollection.do"/>
  <forward
  name="virholex.references.addReference"
  path="/rd_virhoreferences/addReference.jsp"
  />
  <forward
  name="virholex.references.exportReferences"
  path="/ExportReferences.do"
  redirect="true"/>
  <forward
  name="virholex.references.viewUsers"
  path="/VirhoReferencesUsers.do?list="/>
  <forward
  name="virholex.references.viewCollections"
  path="/ViewCollections.do"/>
</action>

<action path="/ViewReferenceDetails"
type="com.virholex.references.ViewReferenceD
etailsAction" name="ReferenceDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.viewReferenceDeta
  ils"
  path="/rd_virhoreferences/viewReferenceDeta
  ils.jsp"/>
  <forward
  name="virholex.virus.privilegeRequest"
  path="/PrivilegeRequest.do"/>
  <forward
  name="virholex.references.addReference"
  path="/rd_virhoreferences/addReference.jsp"
  />
  <forward
  name="virholex.references.editReference"
  path="/EditReference.do"/>
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
</action>

<action path="/AddCollection"
type="com.virholex.references.AddCollectionA
ction" name="CollectionsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.addCollection"
  path="/rd_virhoreferences/addCollection.jsp
  "/>
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
  <forward
  name="virholex.references.viewCollections"
  path="/ViewCollections.do"/>
</action>

<action path="/EditCollection"
type="com.virholex.references.EditCollection
Action" name="CollectionsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.editCollection"
  path="/rd_virhoreferences/editCollection.js
  p"/>
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
</action>

<action path="/AddReference"
type="com.virholex.references.AddReferenceAc
tion" name="ReferenceDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.addReference"
  path="/rd_virhoreferences/addReference.jsp"
  />
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
  <forward
  name="virholex.references.viewReferenceDeta
  ils" path="/ViewReferenceDetails.do"
  redirect="true"/>
</action>

<action path="/EditReference"
type="com.virholex.references.EditReferenceA
ction" name="ReferenceDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.editReference"
  path="/rd_virhoreferences/editReference.jsp
  "/>
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
  <forward
  name="virholex.references.viewReferenceDeta
  ils" path="/ViewReferenceDetails.do"
  redirect="true"/>
</action>

<action path="/ExportReferences"
type="com.virholex.references.ExportReferenc
esAction" name="CollectionsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.exportReferences"
  path="/rd_virhoreferences/exportReferences.
  jsp"/>
  <forward
  name="virholex.references.viewReferences"
  path="/ViewReferences.do" redirect="true"/>
</action>
```

```
<action path="/DisplayReferences"
type="com.virholex.references.DisplayReferen
cesAction" name="ReferenceDetailsForm"
scope="request" validate="false">
  <forward
  name="virholex.references.displayReferences
  "
  path="/rd_virhoreferences/displayReferences
  .jsp"/>
</action>

<action path="/VirhoReferencesUsers"
type="com.virholex.references.ViewUsersActio
n" name="ViewUsersForm" scope="request"
validate="false">
  <forward
  name="virholex.references.viewUsers"
  path="/rd_virhoreferences/viewUsers.jsp"/>
</action>

<!-- VIRHO BASIC INFO -->

<action path="/ViewVirus"
type="com.virholex.basicinfo.ViewVirusAction
" name="VirusForm" scope="request"
validate="false">
  <forward
  name="virholex.basicinfo.viewVirus"
  path="/rd_virhobasicinfo/viewVirusList.jsp"
  />
  <forward name="virholex.basicinfo.addVirus"
  path="/AddVirus.do" />
</action>

<action path="/ViewVirusDetails"
type="com.virholex.basicinfo.ViewVirusDetail
sAction" name="VirusForm" scope="request"
validate="false">
  <forward
  name="virholex.basicinfo.viewVirusDetails"
  path="/rd_virhobasicinfo/viewVirusDetails.j
  sp" />
  <forward
  name="virholex.basicinfo.viewVirus"
  path="/ViewVirus.do" />
  <forward
  name="virholex.basicinfo.editVirus"
  path="/EditVirus.do" />
  <forward
  name="virholex.virus.requestPrivilege"
  path="/PrivilegeRequest.do"/>
  <forward
  name="virholex.basicinfo.viewUsers"
  path="/VirhoBasicInfoUsers.do?list="/>
</action>

<action path="/AddVirus"
type="com.virholex.basicinfo.AddVirusAction"
name="VirusForm" scope="request"
validate="false">
  <forward name="virholex.basicinfo.addVirus"
  path="/rd_virhobasicinfo/addVirus.jsp"/>
  <forward
  name="virholex.basicinfo.viewVirusDetails"
  path="/ViewVirusDetails.do"
  redirect="true"/>
  <forward
  name="virholex.basicinfo.viewVirus"
  path="/ViewVirus.do?list="/>
</action>

<action path="/EditVirus"
type="com.virholex.basicinfo.EditVirusAction
" name="VirusForm" scope="request"
validate="false">
  <forward
  name="virholex.basicinfo.editVirus"
  path="/rd_virhobasicinfo/editVirus.jsp"/>
  <forward
  name="virholex.basicinfo.viewVirusDetails"
  path="/ViewVirusDetails.do"
  redirect="true"/>
</action>

<action path="/VirhoBasicInfoUsers"
type="com.virholex.basicinfo.ViewUsersAction
" name="ViewUsersForm" scope="request"
validate="false">
  <forward
  name="virholex.basicinfo.viewUsers"
  path="/rd_virhobasicinfo/viewUsers.jsp"/>
</action>

<!-- EXTERNAL SEARCH -->

<action path="/VirhoExternalSearch"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virhoexternalsearch/virhoExte
rnalSearch.jsp"
input="/rd_virui/sidebar.jsp"
scope="request" validate="false">
</action>

<action path="/AlternativeExternalSearch"
type="org.apache.struts.actions.ForwardActio
n"
parameter="/rd_virhoexternalsearch/alternati
veExternalSearch.jsp"
input="/rd_virhoexternalsearch/virhoExternal
Search.jsp" scope="request"
validate="false">
</action>

<!-- INTERNAL SEARCH -->

<action path="/QuickSearch"
type="com.virholex.internalsearch.QuickSearc
hAction" name="SearchForm" scope="request"
validate="false">
  <forward
  name="virholex.internalsearch.searchResult"
  path="/rd_virhointernalsearch/quickSearchRe
  sult.jsp"/>
  </action>
  <action path="/AdvancedSearch"
  type="com.virholex.internalsearch.AdvancedS
  earchAction"
  name="SearchForm"
  scope="request"
  validate="false">
  <forward
  name="virholex.internalsearch.advancedSearc
  h"
  path="/rd_virhointernalsearch/advancedSearc
  h.jsp"/>
  <forward
  name="virholex.internalsearch.advancedSearc
  hResult"
  path="/rd_virhointernalsearch/advancedSearc
  hResult.jsp"/>
</action>

</action-mappings>

<!-- APPLICATION RESOURCES -->
<message-resources
parameter="com/virholex/general/ApplicationR
esource"/>

</struts-config>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/w
eb-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns
/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" id="WebApp_ID" version="2.5">

  <display-name>virholex</display-name>
  <!-- Action Servlet Configuration -->
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-
    class>org.apache.struts.action.ActionServle
    t</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-
      config.xml</param-value>
    </init-param>
    <init-param>
      <param-name>debug</param-name>
      <param-value>2</param-value>
    </init-param>
    <init-param>
      <param-name>detail</param-name>
      <param-value>2</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
  </servlet>
  <!-- Standard Action Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>FetchHotspots</servlet-name>
    <servlet-
    class>com.virholex.hotspots.FetchHotspotsSe
    rvlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>FetchHotspots</servlet-name>
    <url-pattern>/FetchHotspots</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>AddHotspot</servlet-name>
    <servlet-
    class>com.virholex.hotspots.AddHotspotServl
    et</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddHotspot</servlet-name>
    <url-pattern>/AddHotspot</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>DeleteHotspot</servlet-name>
    <servlet-
    class>com.virholex.hotspots.DeleteHotspotSe
    rvlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>DeleteHotspot</servlet-name>
    <url-pattern>/DeleteHotspot</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>FetchHotspotInfo</servlet-
    name>
    <servlet-
    class>com.virholex.hotspots.FetchHotspotInf
    oServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>FetchHotspotInfo</servlet-
    name>
    <url-pattern>/FetchHotspotInfo</url-
    pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>60</session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

  <!-- ERROR PAGE -->
  <error-page>
    <exception-
    type>java.lang.Throwable</exception-type>
    <location>/rd_general/errorPage.jsp</locati
    on>
  </error-page>

  <!-- TAG LIBRARIES -->
  <jsp-config>
    <taglib>
      <taglib-uri>/WEB-INF/c.tld</taglib-uri>
      <taglib-location>/WEB-INF/c.tld</taglib-
      location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-
      bean.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-
      bean.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-
      html.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-
      html.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-
      logic.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-
      logic.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-
      nested.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-
      nested.tld</taglib-location>
    </taglib>
    <taglib>
      <taglib-uri>/WEB-INF/struts-
      tiles.tld</taglib-uri>
      <taglib-location>/WEB-INF/struts-
      tiles.tld</taglib-location>
    </taglib>
  </jsp-config>
</web-app>
```

B. Java files

VIRHO USER INTERFACE

UpdatesAction.java

```
package com.virholex.virui;

import java.util.ArrayList;
import java.sql.Timestamp;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.basicinfo.VirusVO;
import com.virholex.basicinfo.VirusForm;

public class UpdatesAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirusForm virusForm = (VirusForm) form;
  VirusVO virusVO = new VirusVO();
  UpdatesDAO updatesDAO = new UpdatesDAO();
  ArrayList<VirusVO> updates = new
  ArrayList<VirusVO>();

  String forward = "virholex.virui.home";
  updates = updatesDAO.readVirusUpdates();

  if(!updates.isEmpty()) {

    for(int i=0; i<updates.size(); i++) {

      String conv1 =
      ((VirusVO)updates.get(i)).getDateAdded();
      String conv2 =
      ((VirusVO)updates.get(i)).getDateModified
      ();
      Timestamp ts1 = Timestamp.valueOf(conv1);
      Timestamp ts2 = Timestamp.valueOf(conv2);
      long time1 = ts1.getTime();
      long time2 = ts2.getTime();
      @SuppressWarnings("deprecation")
      long date1 = ts1.getDate();
      @SuppressWarnings("deprecation")
      long date2 = ts2.getDate();

      if(date1 == date2) {
        if(time1 == time2)
          ((VirusVO)updates.get(i)).setStatus("a
          dded");
        else if(time2 != time1)
          ((VirusVO)updates.get(i)).setStatus("m
          odified");
      }
      else if(date2 != date1) {
        ((VirusVO)updates.get(i)).setStatus("mo
        dified");
      }
    }
  }
```

```
  PropertyUtils.copyProperties(virusForm,
  virusVO);
  request.getSession().setAttribute("updates",
  updates);

  return mapping.findForward(forward);
}
}
```

UpdatesDAO.java

```
package com.virholex.virui;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.basicinfo.VirusVO;

public class UpdatesDAO {

  public ArrayList<VirusVO> readVirusUpdates()
  throws SQLException, NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    VirusVO vo = new VirusVO();
    ArrayList<VirusVO> updates = new
    ArrayList<VirusVO>();

    try {
      String sql = "SELECT virus, date_added,
      date_modified FROM virus ORDER BY
      date_modified DESC LIMIT 10";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.execute(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        vo.setVirusName(rs.getString(1));
        vo.setDateAdded(rs.getString(2));
        vo.setDateModified(rs.getString(3));
        updates.add(vo);
        vo = new VirusVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return updates;
  }
}
```

VIRHO REGISTERED USER SERVICES

AccountRequestAction.java

```
package com.virholex.virus;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;

public class AccountRequestAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {

  AccountRequestForm requestForm =
  (AccountRequestForm) form;
  UserProfileVO usersVO = new UserProfileVO();
  UserProfileVO detailsVO = new
  UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();

  String forward =
  "virholex.virus.accountRequest";
  String requestType =
  request.getParameter("type");

  if(requestForm.getCommand() != null &&
  requestForm.getCommand().equals("Evaluate"))
  {

    String[] users =
    request.getParameterValues("actionUser");
    ArrayList<UserProfileVO> approvedUsers =
    new ArrayList<UserProfileVO>();
    boolean hasApprovedRequest = false;

    if(users.length > 0) {
      for(int i=0; i<users.length; i++) {
        if(getAction(request,
        users[i]).equals("Approve")) {

          detailsVO.setUsername(users[i]);
          detailsVO = (UserProfileVO);
          virusWorker.getName(detailsVO);
          approvedUsers.add(detailsVO);
          detailsVO = new UserProfileVO();
          hasApprovedRequest = true;
          virusWorker.evaluateAccount(users[i],
          VirholexConstants.STATUS_APPROVED);

        } else if(getAction(request,
        users[i]).equals("Deny")) {

          virusWorker.evaluateAccount(users[i],
          VirholexConstants.STATUS_DENIED);

        } else if(getAction(request,
        users[i]).equals("Reactivate")) {

          virusWorker.evaluateAccount(users[i],
          VirholexConstants.STATUS_REACTIVATE);

        }

      }

      if(hasApprovedRequest)
        forward =
        "virholex.virus.accountRequestConfig";
      else
        forward =
        "virholex.virus.accountRequestFinal";
    }

    usersVO.setUsers(approvedUsers);

  } else if(requestForm.getCommand() != null
  && requestForm.getCommand().equals("Add
  Privileges")) {

    String[] users =
    request.getParameterValues("actionUser");

    if(users.length > 0) {
      for(int i=0; i<users.length; i++) {

        if(getExperiment(request, users[i])!=
        null)
          virusWorker.grantPrivilege(users[i],
          VirholexConstants.EXPERIMENT_LEAD_INVE
          STIGATOR_DESC,
          VirholexConstants.PROJECT);
        if(getImage(request, users[i])!= null)
          virusWorker.grantPrivilege(users[i],
          VirholexConstants.IMAGE_SET_COORDINATO
          R_DESC, VirholexConstants.PROJECTSET);
        if(getCollection(request, users[i])!=
        null)
          virusWorker.grantPrivilege(users[i],
          VirholexConstants.COLLECTION_COORDINAT
          OR_DESC, VirholexConstants.PROJECT);
        if(getHotspot(request, users[i])!=
        null)
          virusWorker.grantPrivilege(users[i],
          VirholexConstants.HOTSPOT_MANAGER_DESC
          , VirholexConstants.PROJECT);
      }
    }

    forward =
    "virholex.virus.accountRequestFinal";

  } else {

    if(requestType != null)
      requestForm.setRequestType(requestType);
    if(requestForm.getRequestType() != null) {

      if(requestType.equals("PendingAccount"))
        usersVO.setUsers(virusWorker.getPending
        Account(usersVO));
      else
      if(requestType.equals("ReactivateAccount"
      ))
        usersVO.setUsers(virusWorker.getReactiv
        ateAccount(usersVO));

      PropertyUtils.copyProperties(requestForm,
      usersVO);
      forward =
      "virholex.virus.accountRequestEvaluate";

    } else {
      requestForm.setPendingSize(virusWorker.ge
      tPendingAccount(usersVO).size());
      requestForm.setReactivateSize(virusWorker
      .getReactivateAccount(usersVO).size());
    }
  }

  PropertyUtils.copyProperties(requestForm,
  usersVO);

  return mapping.findForward(forward);
}
```

```
public String getAction (HttpServletRequest
request, String user) {
  return request.getParameter("action" +
  user);
}

public String getExperiment
(HttpServletRequest request, String user) {
  return request.getParameter("experiment" +
  user);
}

public String getImage (HttpServletRequest
request, String user) {
  return request.getParameter("image" + user);
}

public String getCollection
(HttpServletRequest request, String user) {
  return request.getParameter("collection" +
  user);
}

public String getHotspot (HttpServletRequest
request, String user) {
  return request.getParameter("hotspot" +
  user);
}
```

AccountRequestForm.java

```
package com.virholex.virus;

import java.util.ArrayList;

import org.apache.struts.action.ActionForm;

public class AccountRequestForm extends
ActionForm {

  private static final long serialVersionUID =
  4589595437438235867L;
  private int pendingSize;
  private int reactivateSize;
  private String command;
  private String requestType;
  private String status;
  private ArrayList<UserProfileVO> users;

  public void setPendingSize(int pendingSize)
  {
    this.pendingSize = pendingSize;
  }
  public int getPendingSize() {
    return pendingSize;
  }
  public void setReactivateSize(int
  reactivateSize) {
    this.reactivateSize = reactivateSize;
  }
  public int getReactivateSize() {
     return reactivateSize;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setRequestType(String
  requestType) {
    this.requestType = requestType;
  }
  public String getRequestType() {
    return requestType;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }
  public void
  setUsers(ArrayList<UserProfileVO> users) {
    this.users = users;
  }
  public ArrayList<UserProfileVO> getUsers() {
    return users;
  }

}
```

AdminPageDAO.java

```
package com.virholex.virus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.SendEmail;

public class AdminPageDAO {

  public static void evaluateAccount(String
  username, String status) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String emailHeader = "";
    String message = "";
    String emailAddress = "";
    String password = "";
    String name = "";

    try {

      String sql = "";
      conn = DBConnect.getConnection();

      sql = "SELECT AES_DECRYPT(password,?) AS
      password, first_name, last_name, email
      FROM users WHERE username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.setString(2, username);
      rs = ps.executeQuery();

      if(rs.next()) {
        password = rs.getString(1);
        name = rs.getString(2) + " " +
        rs.getString(3);
        emailAddress = rs.getString(4);
      }

      rs.close();
      ps.close();

      if(status.equals("approved")) {
```

```
    sql = "UPDATE users SET status=?,
    approval_date=NOW() WHERE username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, status);
    ps.setString(2, username);
    ps.executeUpdate();
    ps.close();

    emailHeader = "Welcome to Virholex";
    message = "Welcome to Virus-Host
    Interaction Lexicon (Virholex).\n\nYou
    can log in to Virholex using your
    username and password
    below.\n\n\tUsername : " + username +
    "\n\tPassword : " + password;

  } else if(status.equals("reactivate")) {

    sql = "UPDATE users SET
    status='activated', date_modified=NOW()
    WHERE username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.executeUpdate();
    ps.close();

    emailHeader = "Virholex Account
    Reactivated";
    message = "Your reactivation request
    was approved by the Virholex
    Team.\n\nYou can now log in to Virholex
    using your username and password
    below.\n\n\tUsername : " + username +
    "\n\tPassword : " + password;

  } else if(status.equals("suspended")) {

    sql = "UPDATE users SET
    status='reactivate',
    date_modified=NOW() WHERE username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.executeUpdate();
    ps.close();

    emailHeader = "Virholex Request
    Reactivation";
    message = "Your reactivation request
    will be evaluated within 24 hours.";

  } else {

    sql = "UPDATE users SET status=?,
    date_modified=NOW() WHERE username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, status);
    ps.setString(2, username);
    ps.executeUpdate();
    ps.close();

    emailHeader = "Virholex Account was not
    approved";
    message = "We Regret to inform you that
    your account request was denied.";
  }

  SendEmail.sendEmail(emailAddress,
  emailHeader, message, name);

} catch(Exception e) {
  e.printStackTrace();
} finally {
  if(rs != null) rs.close();
  if(ps != null) ps.close();
  if(conn != null) conn.close();
}
  }

  public static boolean isAdmin(UserProfileVO
  vo) throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean isAdmin = false;

    try {

      String sql = "SELECT * FROM users WHERE
      username=? AND status='admin'";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()) {
        isAdmin = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return isAdmin;
  }

public static int getTotalAdmin() throws
SQLException, NamingException {

  Connection conn = null;
  Statement s = null;
  ResultSet rs = null;
  int counter = 0;

  try {

     String sql = "SELECT COUNT(*) AS
     admin_count FROM users WHERE
     status='admin'";
     conn = DBConnect.getConnection();
     s = conn.createStatement();
     s.executeQuery(sql);
     rs = s.getResultSet();

     while(rs.next()) {
       counter = rs.getInt("admin_count");
     }

    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return counter;
  }

  public static boolean
  isProjectLeader(UserProfileVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String involvement = "";
    String role = "";
    int projectCount = 0;
    boolean isProjectLeader = false;

    try {
```

```
    String sql = "SELECT
    user_prev.involvement AS involvement,
    user_prev.role_name AS role_name FROM
    user_prev, user_roles WHERE
    user_prev.username=? AND
    user_prev.role_name=user_roles.role_name
    AND user_roles.level=4";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    rs = ps.executeQuery();

    while(rs.next()) {
      involvement =
      rs.getString("user_prev.involvement");
      role =
      rs.getString("user_prev.role_name");
      projectCount =
      getTotalProjectLeader(involvement,
      role);

      if(projectCount > 1) {
        isProjectLeader = true;
        break;
      }
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isProjectLeader;
}

public static int
getTotalProjectLeader(String involvement,
String role_name) throws SQLException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  int counter = 0;

  try {

    String sql = "SELECT COUNT(*) AS
    leader_count FROM user_prev WHERE
    involvement=? AND role_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, involvement);
    ps.setString(2, role_name);
    rs = ps.executeQuery();

    while(rs.next()) {
      counter = rs.getInt("leader_count");
    }

  } catch(Exception e) {
    e.printStackTrace();
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return counter;
}

public static void
changeApprovedStatus(String username) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    String sql = "UPDATE users SET
    status='activated', date_modified=NOW()
    WHERE username=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.executeUpdate();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static ArrayList<UserProfileVO>
getPendingAccount(UserProfileVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  Statement s = null;
  ResultSet rs = null;
  UserProfileVO detailsVO = new
  UserProfileVO();
  ArrayList<UserProfileVO> usersList = new
  ArrayList<UserProfileVO>();

  try {

    String sql = "SELECT first_name,
    last_name, username, email, affiliation
    from users where status='pending'";
    conn = DBConnect.getConnection();
    s = conn.createStatement();
    s.executeQuery(sql);
    rs = s.getResultSet();

    while(rs.next()) {
      detailsVO.setFirstName(rs.getString("fi
      rst_name"));
      detailsVO.setLastName(rs.getString("las
      t_name"));
      detailsVO.setUsername(rs.getString("use
      rname"));
      detailsVO.setEmailAddress(rs.getString(
      "email"));
      detailsVO.setAffiliation(rs.getString("
      affiliation"));

      usersList.add(detailsVO);
      detailsVO = new UserProfileVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(s != null) s.close();
    if(conn != null) conn.close();
  }

  return usersList;
}

public static ArrayList<UserProfileVO>
getReactivateAccount(UserProfileVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  Statement s = null;
  ResultSet rs = null;
  UserProfileVO detailsVO = new
  UserProfileVO();
```

```
    ArrayList<UserProfileVO> usersList = new
    ArrayList<UserProfileVO>();

    try {

      String sql = "SELECT first_name,
      last_name, username, email, affiliation
      from users where status='reactivate'";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.executeQuery(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        detailsVO.setFirstName(rs.getString("fi
        rst_name"));
        detailsVO.setLastName(rs.getString("las
        t_name"));
        detailsVO.setUsername(rs.getString("use
        rname"));
        detailsVO.setEmailAddress(rs.getString(
        "email"));
        detailsVO.setAffiliation(rs.getString("
        affiliation"));

        usersList.add(detailsVO);
        detailsVO = new UserProfileVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return usersList;
  }

  public static void changeAdminStatus(String
  username, String status) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      String sql = "UPDATE users SET status=?,
      date_modified=NOW() WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, status);
      ps.setString(2, username);
      ps.executeUpdate();
    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

CancelAccountAction.java

```
package com.virholex.virus;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class CancelAccountAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  CancelAccountForm cancelAccountForm =
  (CancelAccountForm) form;
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();
  ArrayList<String> userAccount = new
  ArrayList<String>();

  String forward =
  "virholex.virus.cancelAccount";
  String username =
  (String)session.getAttribute("userLogin");
  String password =
  cancelAccountForm.getPassword();

  if(cancelAccountForm.getCancelAccount()!=nul
  l &&
  cancelAccountForm.getCancelAccount().equals(
  "Cancel Account")) {

    if(cancelAccountForm.getPassword() != null)
    {
      userAccount =
      virusWorker.checkPassword(username,
      password);
      cancelAccountForm.setDbPassword(userAccou
      nt.get(0));
      cancelAccountForm.setDbEmailAddress(userA
      ccount.get(1));
    }

    errors =
    cancelAccountForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      virusWorker.cancelAccount(username,
      cancelAccountForm.getDbEmailAddress());
      session.invalidate();
      request.setAttribute("status",
      "unsubscribe");
      forward = "virholex.virus.statusCheck";
    } else {
      saveErrors(request, errors);
    }
  }

  return mapping.findForward(forward);
}
}
```

CancelAccountDAO.java

```
package com.virholex.virus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.naming.NamingException;
```

```
import com.virholex.general.DBConnect;
import com.virholex.general.SendEmail;

public class CancelAccountDAO {

  public static ArrayList<String>
  verifyPassword(String username, String
  password) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<String> userAccount = new
    ArrayList<String>();

    try {
      String sql = "SELECT
      AES_DECRYPT(password,?) AS password,
      email FROM users WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.setString(2, username);
      rs = ps.executeQuery();

      while(rs.next()) {
        userAccount.add(rs.getString(1));
        userAccount.add(rs.getString(2));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return userAccount;
  }

  public static void cancelAccount(String
  username, String email) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    String message = "You have successfully
    unsubscribed your account from Virholex.";
    String emailHeader = "Virholex
    Unscubscription";
    String name = "";

    try {
      String sql = "UPDATE users SET
      status='suspended', date_modified=NOW()
      WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.executeUpdate();
      ps.close();

      sql = "SELECT first_name, last_name FROM
      users WHERE username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      rs = ps.executeQuery();

      if(rs.next()){
        name = rs.getString(1) + " " +
        rs.getString(2);
      }

      SendEmail.sendEmail(email, emailHeader,
      message, name);

    } catch(Exception e) {
      e.printStackTrace();
    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

CancelAccountForm.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

import com.virholex.general.Utilities;

public class CancelAccountForm extends
ActionForm {

  private static final long serialVersionUID =
  -9107352106055401156L;
  private String password;
  private String dbPassword;
  private String dbEmailAddress;
  private String cancelAccount;

  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setDbPassword(String dbPassword)
  {
    this.dbPassword = dbPassword;
  }
  public String getDbPassword() {
    return dbPassword;
  }
  public void setDbEmailAddress(String
  dbEmailAddress) {
    this.dbEmailAddress = dbEmailAddress;
  }
  public String getDbEmailAddress() {
    return dbEmailAddress;
  }
  public void setCancelAccount(String
  cancelAccount) {
    this.cancelAccount = cancelAccount;
  }
  public String getCancelAccount() {
    return cancelAccount;
  }

  public ActionErrors validate(ActionMapping
  mapping, HttpServletRequest request) {

    ActionErrors errors = new ActionErrors();

    if (Utilities.isEmpty(this.getPassword()))
    {
```

```
      errors.add("password", new
      ActionMessage("error.empty2",
      "Password"));
    } else {
      if(!Utilities.isEmpty(this.getDbPassword(
      )) &&
      !this.getPassword().equals(this.getDbPass
      word())) {
        errors.add("password", new
        ActionMessage("error.invalid",
        "Password"));
      }
    }

    return errors;
  }

}
```

EditUserProfileAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class EditUserProfileAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  EditUserProfileForm userForm =
  (EditUserProfileForm) form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.virus.editProfile";
  String username =
  (String)session.getAttribute("userLogin");

  PropertyUtils.copyProperties(userVO,
  userForm);
  userVO.setUsername(username);

  if(userForm.getCommand()!=null &&
  userForm.getCommand().equals("Submit")) {
    // validate password and/or email entered
    if(userVO.isChangePassword())
      userForm.setDbPassword(virusWorker.checkP
      assword(userVO));
    if(userVO.getEmailAddress() != null)
      userForm.setHasEmail(virusWorker.checkEma
      il(userVO));

    errors = userForm.validate(mapping,
    request);

    if(errors.isEmpty()){
      virusWorker.updateUser(userVO);
      forward = "virholex.virus.userProfile";
    }
    else {
      saveErrors(request, errors);
    }

  } else if(userForm.getCommand()!=null &&
  userForm.getCommand().equals("Cancel")) {
    forward = "virholex.virus.userProfile";
  } else {
    userVO = (UserProfileVO)
    virusWorker.read(userVO);
    userVO.setDbPassword(userVO.getPassword());
  }

  // set the value of Edit checkbox
  userForm.setEditPassword(String.valueOf(user
  VO.isChangePassword()));

  // set the user profile details
  PropertyUtils.copyProperties(userForm,
  userVO);

  return mapping.findForward(forward);
}
}
```

EditUserProfileDAO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class EditUserProfileDAO {

  public static boolean
  checkEmail(UserProfileVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean isEmail = false;

    try {
      String sql = "SELECT email FROM users
      WHERE email=? AND username<>?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getEmailAddress());
      ps.setString(2, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()) {
        isEmail = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return isEmail;
  }

  public static String
  checkPassword(UserProfileVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
```

```
    ResultSet rs = null;

    try {
      String sql = "SELECT username,
      AES_DECRYPT(password,?) AS password FROM
      users WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      ps.setString(2, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()) {
        vo.setDbPassword(rs.getString("password
        "));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo.getDbPassword();
  }

  public static void updateUser(UserProfileVO
  vo) throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    String sql = "";

    try {
      if(vo.isChangePassword())
        sql = "UPDATE users SET last_name=?,
        first_name=?, email=?, affiliation=?,
        job_title=?, password=AES_ENCRYPT(?,?),
        date_modified=NOW() WHERE username=?";
      else
        sql = "UPDATE users SET last_name=?,
        first_name=?, email=?, affiliation=?,
        job_title=?, date_modified=NOW() WHERE
        username=?";

        conn = DBConnect.getConnection();
        ps = conn.prepareStatement(sql);
        ps.setString(1, vo.getLastName());
        ps.setString(2, vo.getFirstName());
        ps.setString(3, vo.getEmailAddress());
        ps.setString(4, vo.getAffiliation());
        ps.setString(5, vo.getJobTitle());

      if(vo.isChangePassword()) {
        ps.setString(6, vo.getNewPassword());
        ps.setString(7, vo.getUsername());
        ps.setString(8, vo.getUsername());
      } else {
        ps.setString(6, vo.getUsername());
      }

      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

  }

}
```

EditUserProfileForm.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;

import com.virholex.general.Utilities;

public class EditUserProfileForm extends
ActionForm {

  private static final long serialVersionUID =
  4496916054957869681L;
  private String oldPassword;
  private String newPassword;
  private String reNewPassword;
  private boolean changePassword;
  private String editPassword;
  private String emailAddress;
  private boolean hasEmail;
  private String firstName;
  private String lastName;
  private String affiliation;
  private String jobTitle;
  private String command;
  private String errorMessage;
  private String dbPassword;

  public EditUserProfileForm() {
    super();
  }
  public void setOldPassword(String
  oldPassword) {
    this.oldPassword = oldPassword;
  }
  public String getOldPassword() {
    return oldPassword;
  }
  public void setNewPassword(String
  newPassword) {
    this.newPassword = newPassword;
  }
  public String getNewPassword() {
    return newPassword;
  }
  public void setReNewPassword(String
  reNewPassword) {
    this.reNewPassword = reNewPassword;
  }
  public String getReNewPassword() {
    return reNewPassword;
  }
  public void setChangePassword(boolean
  changePassword) {
    this.changePassword = changePassword;
  }
  public boolean isChangePassword() {
    return changePassword;
  }
  public void setEditPassword(String
  editPassword) {
    this.editPassword = editPassword;
  }
  public String getEditPassword() {
    return editPassword;
  }
  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
```

```
public void setFirstName(String firstName) {
  this.firstName = firstName;
}
public String getFirstName() {
  return firstName;
}
public void setLastName(String lastName) {
  this.lastName = lastName;
}
public String getLastName() {
  return lastName;
}
public void setAffiliation(String
affiliation) {
  this.affiliation = affiliation;
}
public String getAffiliation() {
  return affiliation;
}
public void setJobTitle(String jobTitle) {
  this.jobTitle = jobTitle;
}
public String getJobTitle() {
  return jobTitle;
}
public void setHasEmail(boolean hasEmail) {
  this.hasEmail = hasEmail;
}
public boolean isHasEmail() {
  return hasEmail;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setErrorMessage(String
errorMessage) {
  this.errorMessage = errorMessage;
}
public String getErrorMessage() {
  return errorMessage;
}
public void setDbPassword(String dbPassword)
{
  this.dbPassword = dbPassword;
}
public String getDbPassword() {
  return dbPassword;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if
  (Utilities.isEmpty(this.getEmailAddress())
  || Utilities.isEmpty(this.getFirstName())
  || Utilities.isEmpty(this.getLastName()) ||
  Utilities.isEmpty(this.getAffiliation()) ||
  Utilities.isEmpty(this.getJobTitle())) {
    errors.add("errorMessage", new
    ActionMessage("error.required.all"));

    if(Utilities.isEmpty(this.getEmailAddress
    ()))
      errors.add("emailAddress", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getFirstName())
    )
      errors.add("firstName", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getLastName()))
      errors.add("lastName", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getAffiliation(
    )))
      errors.add("affiliation", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getJobTitle()))
      errors.add("jobTitle", new
      ActionMessage(""));

  }

  if(this.isChangePassword()) {
    if(!(Utilities.isEmpty(this.getOldPasswor
    d()) &&
    Utilities.isEmpty(this.getNewPassword())
    &&
    Utilities.isEmpty(this.getReNewPassword()
    ))) {
      if(Utilities.isEmpty(this.getOldPasswor
      d()))
        errors.add("oldPassword", new
        ActionMessage("error.empty2", "old
        password"));
      if (this.getNewPassword().length() < 6)
        errors.add("newPassword", new
        ActionMessage("error.minlength",
        "Password"));
      else
      if(!Utilities.validInput(this.getNewPas
      sword(), 2))
        errors.add("password", new
        ActionMessage("error.invalid",
        "Password"));
      if
      (!this.getNewPassword().toString().equa
      ls(this.getReNewPassword().toString()))
        errors.add("reNewPassword", new
        ActionMessage("error.notEqual.password
        "));
      if(!Utilities.isEmpty(this.getOldPasswo
      rd()) &&
      !Utilities.isEmpty(this.getDbPassword()
      )){
        if(!(this.getOldPassword().equals(this
        .getDbPassword()))){
          errors.add("oldPassword", new
          ActionMessage("error.invalid", "Old
          Password"));
        }
      }
    } else {
      this.setChangePassword(false);
    }
  }

  if(!Utilities.isEmpty(this.getEmailAddress(
  ))) {
    if(!Utilities.validInput(this.getEmailAdd
    ress(), 3)) {
      errors.add("emailAddress", new
      ActionMessage("error.invalid", "Email
      Address"));
    } else if(this.isHasEmail()){
      errors.add("emailAddress", new
      ActionMessage("error.notAvailable",
      "email address"));
    }
  }

  return errors;
 }

}
```

FetchPrivilegesDAO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class FetchPrivilegesDAO {

  public static int getPrivilegeLevel(String
  role) throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    int level = 0;

    try {
      String sql = "SELECT level FROM
      user_roles WHERE role_name=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, role);
      rs = ps.executeQuery();

      if(rs.next()) {
        level = rs.getInt("level");
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return level;
  }

  public static void grantPrivilege(String
  username, String involvement, String type)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    final String project =
    VirholexConstants.DEFAULT_PROJECT;

    try {
      String sql = "INSERT INTO user_prev
      (username, role_name, involvement,
      date_added, kind, project_id) VALUES
      (?,?,?,NOW(),?,?)";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.setString(2, involvement);
      ps.setString(3, project);
      ps.setString(4, type);
      ps.setInt(5, 0);
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

  }

}
```

ForgotPasswordAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class ForgotPasswordAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ForgotPasswordForm forgotPasswordForm =
  (ForgotPasswordForm) form;
  ForgotPasswordVO forgotPasswordVO = new
  ForgotPasswordVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();

  String forward =
  "virholex.virus.forgotPassword";

  PropertyUtils.copyProperties(forgotPasswordV
  O, forgotPasswordForm);

  if(forgotPasswordForm.getCommand()!= null &&
  forgotPasswordForm.getCommand().equals("Rese
  t Password")){

    if(forgotPasswordVO.getUsername()!=null &&
    forgotPasswordVO.getEmailAddress()!=null) {
      if(!virusWorker.checkAccount(forgotPasswo
      rdVO)) {
        forgotPasswordForm.setNotValidAccount(t
        rue);
      }
    }

    errors =
    forgotPasswordForm.validate(mapping,
    request);

    if(errors.isEmpty()){
      virusWorker.resetPassword(forgotPasswordV
      O);
      forward = "virholex.virus.statusCheck";
    } else {
      saveErrors(request, errors);
    }
  }

  return mapping.findForward(forward);
}
}
```

ForgotPasswordDAO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class ForgotPasswordDAO {

  public static boolean
  checkAccount(ForgotPasswordVO vo) throws
  SQLException, NamingException {
```

```
  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean validAccount = false;

  try {
    String sql = "SELECT username, email FROM
    users WHERE username=? AND email=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    ps.setString(2, vo.getEmailAddress());
    rs = ps.executeQuery();

    while(rs.next()) {
      validAccount = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return validAccount;
}

public static boolean checkUsername(String
username, String key) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean hasUsername = false;

  try {
    String sql = "SELECT users.username FROM
    users, password_reset WHERE
    users.username=password_reset.username
    AND password_reset.username=? AND
    password_reset.activation_key=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, key);
    rs = ps.executeQuery();

    if(rs.next()) {
      hasUsername = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return hasUsername;

}

public static boolean
checkActivationKey(String key) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean hasActivationKey = false;

  try {
    String sql = "SELECT username FROM
    password_reset WHERE activation_key=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, key);
    rs = ps.executeQuery();

    if(rs.next()) {
      hasActivationKey = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return hasActivationKey;

}

public static void
resetPassword(ForgotPasswordVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  String emailHeader = "Virholex Reset
  Password";
  String email = vo.getEmailAddress();
  String message = "", name = "";
  String state = "active";
  //sequence of characters for random string
  String characterSequence =
  "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ123456
  7890abcdefghijklmnopqrstubwxyz";
  //random sequence
  String randomSequence = "";
  //dummy key
  String key = "";
  //dummy crack
  String crack = "000";
  Random random = new Random();
  int ind = 0;

  for(int i=0; i < 30; i++) {

    ind =
    random.nextInt(characterSequence.length()
    );
    randomSequence +=
    characterSequence.substring(ind, ind+1);

    if(i < 15) {
      ind =
      random.nextInt(characterSequence.length
      ());
      key += characterSequence.substring(ind,
      ind+1);
    }

    if(i<7) {
      ind =
      random.nextInt(characterSequence.length
      ());
      crack +=
      characterSequence.substring(ind,
      ind+1);
    }
  }

  try {
    String sql = "INSERT INTO password_reset
    (username,activation_key) VALUES (?,?)";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    ps.setString(2, randomSequence);
```

```
      ps.executeUpdate();

      //construct outgoing message
      message = "Click on this link to reset
      your
      password:\nhttp://www.bioinfo.mpg.de/virh
      olex/ResetPassword.do?state=" + state +
      "&key=" + key + "&rand=" + randomSequence
      + "&crack=" + crack;
      sql = "SELECT first_name, last_name FROM
      users WHERE username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()){
        name = rs.getString(1) + " " +
        rs.getString(2);
      }

      rs.close();
      ps.close();

      //send message
      SendEmail.sendEmail(email, emailHeader,
      message, name);

    } catch(Exception e) {
      e.printStackTrace();
    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

  }

  public static void resetPassword(String
  username, String password) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {

      conn = DBConnect.getConnection();

      String sql = "UPDATE users SET
      password=AES_ENCRYPT(?,?) WHERE
      username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, password);
      ps.setString(2, username);
      ps.setString(3, username);
      ps.executeUpdate();

      sql = "DELETE FROM password_reset WHERE
      username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

  }

}
```

ForgotPasswordForm.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;

import com.virholex.general.Utilities;

public class ForgotPasswordForm extends
ActionForm {

  private static final long serialVersionUID =
  -5895525374841630718L;
  private String username;
  private String emailAddress;
  private String dbUsername;
  private String dbEmailAddress;
  private String newPassword;
  private String renewPassword;
  private String errorMessage;
  private String command;
  private boolean notValidAccount;
  private boolean invalidUsername;

  public ForgotPasswordForm() {
    super();
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setDbUsername(String dbUsername)
  {
    this.dbUsername = dbUsername;
  }
  public String getDbUsername() {
    return dbUsername;
  }
  public void setDbEmailAddress(String
  dbEmailAddress) {
    this.dbEmailAddress = dbEmailAddress;
  }
  public String getDbEmailAddress() {
    return dbEmailAddress;
  }
  public void setNewPassword(String
  newPassword) {
    this.newPassword = newPassword;
  }
  public String getNewPassword() {
    return newPassword;
  }
  public void setRenewPassword(String
  renewPassword) {
    this.renewPassword = renewPassword;
  }
  public String getRenewPassword() {
    return renewPassword;
  }
  public void setNotValidAccount(boolean
  notValidAccount) {
    this.notValidAccount = notValidAccount;
  }
  public boolean isNotValidAccount() {
```

```
  return notValidAccount;
}
public void setInvalidUsername(boolean
invalidUsername) {
  this.invalidUsername = invalidUsername;
}
public boolean isInvalidUsername() {
  return invalidUsername;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setErrorMessage(String
errorMessage) {
  this.errorMessage = errorMessage;
}
public String getErrorMessage() {
  return errorMessage;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if (Utilities.isEmpty(this.getUsername())
  &&
  Utilities.isEmpty(this.getEmailAddress()))
  {
    errors.add("errorMessage", new
    ActionMessage("error.invalid.account"));
    errors.add("username", new
    ActionMessage(""));
    errors.add("emailAddress", new
    ActionMessage(""));
  } else {
    if
    (Utilities.isEmpty(this.getUsername())) {
      errors.add("errorMessage", new
      ActionMessage("error.empty",
      "Username"));
      errors.add("username", new
      ActionMessage(""));
    } else if
    (Utilities.isEmpty(this.getEmailAddress()
    )) {
      errors.add("errorMessage", new
      ActionMessage("error.empty", "Email
      Address"));
      errors.add("emailAddress", new
      ActionMessage(""));
    } else {
      if(isNotValidAccount()) {
        errors.add("errorMessage", new
        ActionMessage("error.invalid.account")
        );
        errors.add("username", new
        ActionMessage(""));
        errors.add("emailAddress", new
        ActionMessage(""));
      }
    }
  }

  return errors;
}

public ActionErrors
validateAccount(ActionMapping mapping,
HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

    if(Utilities.isEmpty(this.getUsername()) ||
    Utilities.isEmpty(this.getNewPassword()) ||
    Utilities.isEmpty(this.getRenewPassword()))
    {
      errors.add("errorMessage", new
      ActionMessage("error.required.all"));

      if(Utilities.isEmpty(this.getUsername()))
        errors.add("username", new
        ActionMessage(""));
      if(Utilities.isEmpty(this.getNewPassword(
      )))
        errors.add("newPassword", new
        ActionMessage(""));
      if(Utilities.isEmpty(this.getRenewPasswor
      d()))
        errors.add("renewPassword", new
        ActionMessage(""));
    }

    if(this.isInvalidUsername()) {
      errors.add("username", new
      ActionMessage("error.invalid",
      "Username"));
    }

    if(!Utilities.isEmpty(this.getNewPassword()
    )) {
      if (this.getNewPassword().length() < 6) {
        errors.add("newPassword", new
        ActionMessage("error.minlength", "New
        Password"));
      } else
      if(!Utilities.validInput(this.getNewPassw
      ord(), 2)) {
        errors.add("newPassword", new
        ActionMessage("error.invalid", "New
        Password"));
      }
    }

    if
    (!this.getNewPassword().equals(this.getRene
    wPassword()) &&
    !(Utilities.isEmpty(this.getNewPassword())
    ||
    Utilities.isEmpty(this.getRenewPassword()))
    ) {
      errors.add("renewPassword", new
      ActionMessage("error.notEqual.password"))
      ;
    }

    return errors;
  }

}
```

ForgotPasswordVO.java

```
package com.virholex.virus;

import com.virholex.general.VO;

public class ForgotPasswordVO implements VO {

  private static final long serialVersionUID =
  -4820794245116160708L;
  private String username;
  private String emailAddress;
  private String dbUsername;
  private String dbEmailAddress;

  public void setUsername(String username) {
```

```
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setDbUsername(String dbUsername)
  {
    this.dbUsername = dbUsername;
  }
  public String getDbUsername() {
    return dbUsername;
  }
  public void setDbEmailAddress(String
  dbEmailAddress) {
    this.dbEmailAddress = dbEmailAddress;
  }
  public String getDbEmailAddress() {
    return dbEmailAddress;
  }

}
```

LoginAccountAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

public class LoginAccountAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  LoginAccountForm loginForm =
  (LoginAccountForm) form;
  LoginAccountVO loginVO = new
  LoginAccountVO();
  LoginAccountVO vo = new LoginAccountVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward = "virholex.virus.login";
  String status = "", username = "";

  PropertyUtils.copyProperties(loginVO,
  loginForm);

  if(loginForm.getCommand() != null &&
  loginForm.getCommand().equals("Submit")){

    if(loginVO.getUsername()!=null) {
      vo =
      (LoginAccountVO)virusWorker.checkAccount(
      loginVO);
      loginForm.setDbUsername(vo.getDbUsername(
      ));
      loginForm.setDbPassword(vo.getDbPassword(
      ));
    }
    errors = loginForm.validate(mapping,
    request);

    if(errors.isEmpty()) {

      username = loginVO.getUsername();
      status =
      virusWorker.getUserStatus(username);

      if(status.equals(VirholexConstants.STATUS
      _ACTIVATED) ||
      status.equals(VirholexConstants.STATUS_AD
      MIN) ||
      status.equals(VirholexConstants.STATUS_SU
      PERADMIN)) {
        session.setAttribute("userLogin",
        username);
        session.setAttribute("statusLogin",
        status);
        forward = "virholex.virui.home";
      } else {
        request.setAttribute("status", status);

        if(status.equals(VirholexConstants.STAT
        US_SUSPENDED)) {
          session.setAttribute("username",
          username);
          forward =
          "virholex.virus.suspendedAccount";
        } else
        if(status.equals(VirholexConstants.STAT
        US_APPROVED)) {
          session.setAttribute("userLogin",
          username);
          session.setAttribute("statusLogin",
          VirholexConstants.STATUS_ACTIVATED);
          request.setAttribute("username",
          username);
          virusWorker.changeStatus(username);
          forward =
          "virholex.virus.statusCheck";
        } else {
          request.setAttribute("username",
          username);
          forward =
          "virholex.virus.statusCheck";
        }
      }
    } else {
      saveErrors(request, errors);
    }

  } else if(loginForm.getCommand() != null &&
  loginForm.getCommand().equals("Yes")) {
    username =

    (String)session.getAttribute("username");
    session.removeAttribute("username");
    virusWorker.evaluateAccount(username,
    VirholexConstants.STATUS_SUSPENDED);
    request.setAttribute("status",
    VirholexConstants.STATUS_SUSPENDED);
    forward = "virholex.virus.statusCheck";

  } else if(loginForm.getCommand() != null &&
  loginForm.getCommand().equals("No")) {
    forward = "virholex.virui.home";
  }

  return mapping.findForward(forward);
}
}
```

ForgotPasswordVO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class LoginAccountDAO {

  public static String getUserStatus(String
  username) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String status = "";

    try {
      String sql = "SELECT status FROM users
      WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      rs = ps.executeQuery();

      while(rs.next()) {
        status = rs.getString("status");
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return status;
  }

  public static LoginAccountVO
  checkAccount(LoginAccountVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "";
      conn = DBConnect.getConnection();

      sql = "SELECT username FROM users WHERE
      username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setDbUsername(rs.getString("username
        "));
      }

      rs.close();
      ps.close();

      if(vo.getUsername()!=null) {
        sql = "SELECT AES_DECRYPT(password,?)
        AS password FROM users WHERE
        username=?";
        ps = conn.prepareStatement(sql);
        ps.setString(1, vo.getUsername());
        ps.setString(2, vo.getUsername());
        rs = ps.executeQuery();

        if(rs.next()) {
          vo.setDbPassword(rs.getString("passwor
          d"));
        }
        rs.close();
        ps.close();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

}
```

LoginAccountForm.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

public class LoginAccountForm extends
ActionForm {

  private static final long serialVersionUID =
  -5895525374841630718L;
  private String username;
  private String password;
  private String dbUsername;
  private String dbPassword;
  private String status;
  private String errorMessage;
  private String command;

  public LoginAccountForm() {
    super();
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setDbUsername(String dbUsername)
  {
    this.dbUsername = dbUsername;
  }
  public String getDbUsername() {
    return dbUsername;
  }
  public void setDbPassword(String dbPassword)
  {
    this.dbPassword = dbPassword;
  }
  public String getDbPassword() {
    return dbPassword;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
```

```
    return command;
  }
  public void setErrorMessage(String
  errorMessage) {
    this.errorMessage = errorMessage;
  }

  public String getErrorMessage() {
    return errorMessage;
  }

  public ActionErrors validate(ActionMapping
  mapping, HttpServletRequest request) {

    ActionErrors errors = new ActionErrors();

    if (Utilities.isEmpty(this.getUsername())
    && Utilities.isEmpty(this.getPassword())) {
      errors.add("errorMessage", new
      ActionMessage("error.invalid.login"));
      errors.add("username", new
      ActionMessage(""));
      errors.add("password", new
      ActionMessage(""));
    }
    else {

      if
      (Utilities.isEmpty(this.getUsername())) {
        errors.add("errorMessage", new
        ActionMessage("error.empty",
        "Username"));
        errors.add("username", new
        ActionMessage(""));
      } else {
        if(Utilities.isEmpty(this.getDbUsername
        ()) &&
        !Utilities.isEmpty(this.getPassword()))
        {
          errors.add("errorMessage", new
          ActionMessage("error.available.usernam
          e", this.getUsername()));
        } else {
          if
          (Utilities.isEmpty(this.getPassword())
          ) {
            errors.add("errorMessage", new
            ActionMessage("error.empty",
            "Password"));
            errors.add("password", new
            ActionMessage(""));
          } else
          if(this.getUsername().equals(this.getD
          bUsername()) &&
          !this.getPassword().equals(this.getDbP
          assword())) {
            errors.add("errorMessage", new
            ActionMessage("error.invalid.login")
            );
            errors.add("username", new
            ActionMessage(""));
            errors.add("password", new
            ActionMessage(""));
          }
        }
      }
    }

    return errors;
  }

}
```

LoginAccountForm.java

```
package com.virholex.virus;

import com.virholex.general.VO;

public class LoginAccountVO implements VO {

  private static final long serialVersionUID =
  2410980012065030220L;
  private String username;
  private String password;
  private String dbUsername;
  private String dbPassword;
  private String status;

  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setDbUsername(String dbUsername)
  {
    this.dbUsername = dbUsername;
  }
  public String getDbUsername() {
    return dbUsername;
  }
  public void setDbPassword(String dbPassword)
  {
    this.dbPassword = dbPassword;
  }
  public String getDbPassword() {
    return dbPassword;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }

}
```

LogoutAccountAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class LogoutAccountAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {

  HttpSession session =
  request.getSession(true);
```

```
  String forward = "virholex.virus.login";

  session.invalidate();

  return mapping.findForward(forward);
}
}
```

RegisterAccountAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class RegisterAccountAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  RegisterAccountForm registerForm =
  (RegisterAccountForm) form;
  RegisterAccountVO registerVO = new
  RegisterAccountVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  String forward =
  "virholex.virus.registration";

  PropertyUtils.copyProperties(registerVO,
  registerForm);

  if(registerForm.getCommand()!=null &&
  registerForm.getCommand().equals("Submit")){

    if(registerVO.getUsername() != null)
      registerForm.setDbUsername(virusWorker.ch
      eckUsername(registerVO));
    if(registerVO.getEmailAddress() != null)
      registerForm.setDbEmailAddress(virusWorke
      r.checkEmail(registerVO));

    errors = registerForm.validate(mapping,
    request);

    if(errors.isEmpty()){
      virusWorker.insertUser(registerVO);
      request.setAttribute("status",
      "registered");
      forward = "virholex.virus.statusCheck";
    } else {
      saveErrors(request, errors);
    }

  } else if(registerForm.getCommand()!=null &&
  registerForm.getCommand().equals("Cancel")){
    forward = "virholex.virui.home";
  }

  return mapping.findForward(forward);
}
}
```

RegisterAccountDAO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class RegisterAccountDAO {

  public static String
  checkUsername(RegisterAccountVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "SELECT username FROM users
      WHERE username=?";

      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()) {
        vo.setDbUsername(rs.getString(1));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo.getDbUsername();

  }

  public static RegisterAccountVO
  insertUser(RegisterAccountVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    String emailHeader = "Greetings from
    Virholex";
    String message = "Thank you for
    registering. Your request will be evaluated
    within 24 hours.";
    String email = vo.getEmailAddress();
    String name = "";

    try {
      String sql = "INSERT INTO users
      (username, password, email, last_name,
      first_name, affiliation, job_title,
      request_date) VALUES
      (?,AES_ENCRYPT(?,?),?,?,?,?,?,NOW())";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      ps.setString(2, vo.getPassword());
      ps.setString(3, vo.getUsername());
      ps.setString(4, vo.getEmailAddress());
      ps.setString(5, vo.getLastName());
      ps.setString(6, vo.getFirstName());
      ps.setString(7, vo.getAffiliation());
      ps.setString(8, vo.getJobTitle());
      ps.executeUpdate();
      ps.close();
```

```
      sql = "SELECT first_name, last_name FROM
      users WHERE username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      if(rs.next()){
        name = rs.getString(1) + " " +
        rs.getString(2);
      }

      rs.close();
      ps.close();

      SendEmail.sendEmail(email, emailHeader,
      message, name);

    } catch(Exception e) {
      e.printStackTrace();
    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

  public static String
  checkEmail(RegisterAccountVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "SELECT email FROM users
      WHERE email=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getEmailAddress());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setDbEmailAddress(rs.getString("emai
        l"));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo.getDbEmailAddress();
  }

}
```

RegisterAccountForm.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;

import com.virholex.general.Utilities;

public class RegisterAccountForm extends
ActionForm {

  private static final long serialVersionUID =
  -5895525374841630718L;
  private String username;
  private String password;
  private String repassword;
  private String emailAddress;
  private String firstName;
  private String lastName;
  private String affiliation;
  private String jobTitle;
  private String command;
  private String errorMessage;
  private String dbUsername;
  private String dbEmailAddress;

  public RegisterAccountForm() {
    super();
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setRepassword(String repassword)
  {
    this.repassword = repassword;
  }
  public String getRepassword() {
    return repassword;
  }
  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public String getFirstName() {
    return firstName;
  }
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setAffiliation(String
  affiliation) {
    this.affiliation = affiliation;
  }
  public String getAffiliation() {
    return affiliation;
  }
  public void setJobTitle(String jobTitle) {
    this.jobTitle = jobTitle;
  }
  public String getJobTitle() {
    return jobTitle;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
```

```
  return command;
}
public void setErrorMessage(String
errorMessage) {
  this.errorMessage = errorMessage;
}
public String getErrorMessage() {
  return errorMessage;
}
public void setDbUsername(String dbUsername)
{
  this.dbUsername = dbUsername;
}
public String getDbUsername() {
  return dbUsername;
}
public void setDbEmailAddress(String
dbEmailAddress) {
  this.dbEmailAddress = dbEmailAddress;
}
public String getDbEmailAddress() {
  return dbEmailAddress;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if (Utilities.isEmpty(this.getUsername())
  || Utilities.isEmpty(this.getPassword()) ||
  Utilities.isEmpty(this.getRepassword()) ||
  Utilities.isEmpty(this.getEmailAddress())
  || Utilities.isEmpty(this.getFirstName())
  || Utilities.isEmpty(this.getLastName()) ||
  Utilities.isEmpty(this.getAffiliation()) ||
  Utilities.isEmpty(this.getJobTitle())) {
    errors.add("errorMessage", new
    ActionMessage("error.required.all"));

    if(Utilities.isEmpty(this.getUsername()))
      errors.add("username", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getPassword()))
      errors.add("password", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getRepassword()
    ))
      errors.add("repassword", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getEmailAddress
    ()))
      errors.add("emailAddress", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getFirstName())
    )
      errors.add("firstName", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getLastName()))
      errors.add("lastName", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getAffiliation(
    )))
      errors.add("affiliation", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getJobTitle()))
      errors.add("jobTitle", new
      ActionMessage(""));

  }

  if(!Utilities.isEmpty(this.getUsername()))
  {
    if (this.getUsername().length() < 6) {
      errors.add("username", new
      ActionMessage("error.minlength",
      "Username"));
    } else
    if(!Utilities.validInput(this.getUsername
    (), 1)) {
      errors.add("username", new
      ActionMessage("error.invalid",
      "Username"));
    }
  }

  if(!Utilities.isEmpty(this.getPassword()))
  {
    if (this.getPassword().length() < 6) {
      errors.add("password", new
      ActionMessage("error.minlength",
      "Password"));
    } else
    if(!Utilities.validInput(this.getPassword
    (), 2)) {
      errors.add("password", new
      ActionMessage("error.invalid",
      "Password"));
    }
  }

  if
  (!this.getPassword().equals(this.getRepassw
  ord()) &&
  !(Utilities.isEmpty(this.getPassword()) ||
  Utilities.isEmpty(this.getRepassword()))) {
    errors.add("repassword", new
    ActionMessage("error.notEqual.password"))
    ;
  }

  if(!Utilities.isEmpty(this.getEmailAddress(
  ))) {
    if(!Utilities.validInput(this.getEmailAdd
    ress(), 3)) {
      errors.add("emailAddress", new
      ActionMessage("error.invalid", "Email
      Address"));
    }
  }

  if(!Utilities.isEmpty(this.getUsername())
  &&
  !Utilities.isEmpty(this.getDbUsername())) {
    if(this.getUsername().equals(this.getDbUs
    ername())){
      errors.add("username", new
      ActionMessage("error.notAvailable",
      "username"));
    }
  }

  if(!Utilities.isEmpty(this.getEmailAddress(
  )) &&
  !Utilities.isEmpty(this.getDbEmailAddress()
  )) {
    if(this.getEmailAddress().equals(this.get
    DbEmailAddress())){
      errors.add("emailAddress", new
      ActionMessage("error.notAvailable",
      "email address"));
    }
  }

  return errors;
 }

}
```

RegisterAccountVO.java

```
package com.virholex.virus;

import com.virholex.general.VO;

public class RegisterAccountVO implements VO {

  private static final long serialVersionUID =
  8313000477776082807L;
  private String username;
  private String password;
  private String repassword;
  private String emailAddress;
  private String firstName;
  private String lastName;
  private String affiliation;
  private String jobTitle;
  private String dbUsername;
  private String dbEmailAddress;

  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setRepassword(String repassword)
  {
    this.repassword = repassword;
  }
  public String getRepassword() {
    return repassword;
  }
  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public String getFirstName() {
    return firstName;
  }
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setAffiliation(String
  affiliation) {
    this.affiliation = affiliation;
  }
  public String getAffiliation() {
    return affiliation;
  }
  public void setJobTitle(String jobTitle) {
    this.jobTitle = jobTitle;
  }
  public String getJobTitle() {
    return jobTitle;
  }
  public void setDbUsername(String dbUsername)
  {
    this.dbUsername = dbUsername;
  }
  public String getDbUsername() {
    return dbUsername;
  }
  public void setDbEmailAddress(String
  dbEmailAddress) {
    this.dbEmailAddress = dbEmailAddress;
  }
  public String getDbEmailAddress() {
    return dbEmailAddress;
  }

}
```

RequestPrivilegeAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.VirholexConstants;

public class RequestPrivilegeAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  RequestPrivilegeForm requestForm =
  (RequestPrivilegeForm) form;
  RequestPrivilegeVO requestVO = new
  RequestPrivilegeVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.virus.requestPrivilege";
  String username =
  (String)session.getAttribute("userLogin");
  String module =
  (String)session.getAttribute("module");
  String project =
  (String)session.getAttribute("project");
  String project2 = "";

  PropertyUtils.copyProperties(requestVO,
  requestForm);
  requestVO.setModule(module);
  requestVO.setProject(project);

  if(module.equals(VirholexConstants.VIRHO_IMA
  GES))
    requestVO.setType(VirholexConstants.PROJECT
    SET);
  else
    requestVO.setType(VirholexConstants.PROJECT
    );
```

```
if(module.equals(VirholexConstants.VIRHO_REF
ERENCES))
  project2 = project + " " +
  VirholexConstants.VIRHO_REFERENCE;
else
if(module.equals(VirholexConstants.VIRHO_HOT
SPOTS))
  project2 = project + " " +
  VirholexConstants.VIRHO_HOTSPOT;
else
if(module.equals(VirholexConstants.VIRHO_MOD
ELS) ||
module.equals(VirholexConstants.VIRHO_EXPERI
MENTS) ||
module.equals(VirholexConstants.VIRHO_IMAGES
))
  project2 = project;

if(virusWorker.checkProject(project2)) {
  if(virusWorker.checkRequest(username,
  project)) {
    request.setAttribute("hasRequest",
    String.valueOf(true));
    forward =
    "virholex.virus.requestPrivilegeFinal";
  } else {
    if(requestForm.getCommand() != null &&
    requestForm.getCommand().equals("Submit
    Request")) {
      virusWorker.sendPrivilege(requestVO,
      username);
      request.setAttribute("hasRequest",
      String.valueOf(false));
      forward =
      "virholex.virus.requestPrivilegeFinal";
    } else if(requestForm.getCommand() !=
    null &&
    requestForm.getCommand().equals("Cancel")
    ) {
      if(module.equals(VirholexConstants.VIRH
      O_EXPERIMENTS)) {
        forward =
        "virholex.experiments.viewProject";
        ActionRedirect redirect = new
        ActionRedirect(mapping.findForward(for
        ward));
        redirect.addParameter("project",
        project);
        return redirect;
      } else
      if(module.equals(VirholexConstants.VIRH
      O_IMAGES)) {
        forward =
        "virholex.images.viewProject";
        ActionRedirect redirect = new
        ActionRedirect(mapping.findForward(for
        ward));
        redirect.addParameter("projectSet",
        project);
        return redirect;
      } else
      if(module.equals(VirholexConstants.VIRH
      O_MODELS)) {
        forward =
        "virholex.models.viewProject";
        ActionRedirect redirect = new
        ActionRedirect(mapping.findForward(for
        ward));
        redirect.addParameter("project",
        project);
        return redirect;
      } else
      if(module.equals(VirholexConstants.VIRH
      O_REFERENCES)) {
        forward =
        "virholex.references.viewProject";
        ActionRedirect redirect = new
        ActionRedirect(mapping.findForward(for
        ward));
        redirect.addParameter("collection",
        project);
        return redirect;
      } else
      if(module.equals(VirholexConstants.VIRH
      O_HOTSPOTS)) {
        forward =
        "virholex.basicinfo.viewProject";
        ActionRedirect redirect = new
        ActionRedirect(mapping.findForward(for
        ward));
        redirect.addParameter("virus",
        project);
        return redirect;
      }
    }
  }
}

  PropertyUtils.copyProperties(requestForm,
  requestVO);

  return mapping.findForward(forward);
}
}
```

RequestPrivilegeDAO.java

```
package com.virholex.virus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.SendEmail;
import com.virholex.general.VirholexConstants;

public class RequestPrivilegeDAO {

  public static boolean
  checkRequestPrivilege(String username,
  String involvement) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean hasRequest = false;

    try {
      String sql = "SELECT * FROM requests
      WHERE involvement=? AND username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, involvement);
      ps.setString(2, username);
      rs = ps.executeQuery();

      if(rs.next()) {
        hasRequest = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
```

```
  }

  return hasRequest;
}

public static void
sendRequestPrivilege(RequestPrivilegeVO vo,
String username) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    String sql = "INSERT INTO requests
    (username, involvement, module, request,
    type, date_added) VALUES
    (?,?,?,?,?,NOW())";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, vo.getProject());
    ps.setString(3, vo.getModule());
    ps.setString(4, vo.getRequestMessage());
    ps.setString(5, vo.getType());
    ps.executeUpdate ();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static boolean checkProject(String
involvement) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean hasProject = false;

  try {
    String sql = "SELECT * FROM project WHERE
    project_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, involvement);
    rs = ps.executeQuery();

    if(rs.next()) {
      hasProject = true;
    } else {
      sql = "SELECT * FROM project_set WHERE
      project_set_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, involvement);
      rs = ps.executeQuery();

      if(rs.next()) {
        hasProject = true;
      }
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return hasProject;
}

public static ArrayList<String>
getUserRoles(String module) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> userRoles = new
  ArrayList<String>();
  String roleName = "";

  try {
    String sql = "SELECT role_name FROM
    user_roles where module=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, module);
    rs = ps.executeQuery();

    while(rs.next()) {
      roleName = rs.getString(1);

      if(roleName.equals(VirholexConstants.RE
      STRICTED_USER_DESC))
        userRoles.add(0, roleName);
      else
        userRoles.add(roleName);
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return userRoles;
}

public static ArrayList<UserProfileVO>
getProjectRequestUsers(String username,
String project, String module, String
roleName) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<UserProfileVO> usersList= new
  ArrayList<UserProfileVO>();
  UserProfileVO vo = new UserProfileVO();

  try {
    String sql = "SELECT users.username,
    users.first_name, users.last_name,
    users.affiliation FROM requests, users,
    user_prev WHERE
    requests.username=users.username AND
    requests.involvement=? AND module=? AND
    (status='admin' OR status='activated' OR
    status='approved') AND
    user_prev.username=? AND
    user_prev.involvement LIKE ? AND
    user_prev.role_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, project);
    ps.setString(2, module);
    ps.setString(3, username);
    ps.setString(4, "%"+project+"%");
    ps.setString(5, roleName);
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setUsername(rs.getString(1));
      vo.setFirstName(rs.getString(2));
      vo.setLastName(rs.getString(3));
```

```
    vo.setAffiliation(rs.getString(4));

    usersList.add(vo);
    vo = new UserProfileVO();
   }

  } finally {
   if(rs != null) rs.close();
   if(ps != null) ps.close();
   if(conn != null) conn.close();
  }

  return usersList;
}

public static RequestPrivilegeVO
readProjectRequest(RequestPrivilegeVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {
   String sql = "SELECT requests.username,
   involvement, module, request, date_added,
   first_name, last_name FROM requests,
   users WHERE
   requests.username=users.username AND
   requests.username=? AND involvement=? AND
   module=?";
   conn = DBConnect.getConnection();
   ps = conn.prepareStatement(sql);
   ps.setString(1, vo.getUsername());
   ps.setString(2, vo.getProject());
   ps.setString(3, vo.getModule());
   rs = ps.executeQuery();

   while(rs.next()) {
    vo.setUsername(rs.getString(1));
    vo.setProject(rs.getString(2));
    vo.setModule(rs.getString(3));
    vo.setRequestMessage(rs.getString(4));
    vo.setDateAdded(rs.getString(5));
    vo.setFirstName(rs.getString(6));
    vo.setLastName(rs.getString(7));
   }

  } finally {
   if(rs != null) rs.close();
   if(ps != null) ps.close();
   if(conn != null) conn.close();
  }

  return vo;
}

public static void
approveUserPrivilege(RequestPrivilegeVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String name = "", message = "",
  emailAddress = "";
  String emailHeader = "Privilege Request
  Approved";
  int projectId = 0;

  try {
   conn = DBConnect.getConnection();
   String sql = "";

   if(vo.getModule().equals(VirholexConstant
   s.VIRHO_EXPERIMENTS) ||
vo.getModule().equals(VirholexConstants.V
IRHO_MODELS)) {
  sql = "SELECT project_id FROM project
  WHERE project_name=?";
  ps = conn.prepareStatement(sql);
  ps.setString(1, vo.getProject());
  rs = ps.executeQuery();
  rs.next();
  projectId = rs.getInt(1);

  if(rs.next()){
   emailAddress = rs.getString(1);
  }

  rs.close();
  ps.close();
}

sql = "INSERT into user_prev (username,
role_name, involvement, date_added,
date_modified, kind, project_id) VALUES
(?,?,?,NOW(),NOW(),?,?)";
ps = conn.prepareStatement(sql);
ps.setString(1, vo.getUsername());
ps.setString(2, vo.getPrivilege());

if(vo.getModule().equals(VirholexConstant
s.VIRHO_REFERENCES))
  ps.setString(3, vo.getProject()+"
  "+VirholexConstants.VIRHO_REFERENCE);
else
if(vo.getModule().equals(VirholexConstant
s.VIRHO_HOTSPOTS))
  ps.setString(3, vo.getProject()+"
  "+VirholexConstants.VIRHO_HOTSPOT);
else
  ps.setString(3, vo.getProject());

ps.setString(4, vo.getKind());
ps.setInt(5, projectId);
ps.executeUpdate();
ps.close();

sql = "DELETE FROM requests WHERE
username=? AND involvement=? AND
module=?";
ps = conn.prepareStatement(sql);
ps.setString(1, vo.getUsername());
ps.setString(2, vo.getProject());
ps.setString(3, vo.getModule());
ps.executeUpdate();
ps.close();

sql = "SELECT first_name, last_name,
email FROM users WHERE username=?";
ps = conn.prepareStatement(sql);
ps.setString(1, vo.getUsername());
rs = ps.executeQuery();

if(rs.next()){
  name = rs.getString(1) + " " +
  rs.getString(2);
  emailAddress = rs.getString(3);
}

rs.close();
ps.close();

message = "Congratulations!\n\n You are
now a " + vo.getPrivilege() + " for the
project " + vo.getProject() + ".";

// send message
SendEmail.sendEmail(emailAddress,
emailHeader, message, name);
```

```
  } catch(Exception e) {
    e.printStackTrace();
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static void
denyUserPrivilege(RequestPrivilegeVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String name = "", message = "",
  emailAddress = "";
  String emailHeader = "Privilege Request
  Denied";

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE FROM requests WHERE
    username=? AND involvement=? AND
    module=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    ps.setString(2, vo.getProject());
    ps.setString(3, vo.getModule());
    ps.executeUpdate();
    ps.close();

    sql = "SELECT first_name, last_name,
    email FROM users WHERE username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    rs = ps.executeQuery();

    if(rs.next()){
      name = rs.getString(1) + " " +
      rs.getString(2);
      emailAddress = rs.getString(3);
    }

    rs.close();
    ps.close();

    message = "We regret to inform you that
    we can't give you any privilege for the
    project " + vo.getProject() + ".";

    // send message
    SendEmail.sendEmail(emailAddress,
    emailHeader, message, name);

  } catch(Exception e) {
    e.printStackTrace();
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static void
updateProjectPrivilege(ArrayList<String>
users, ArrayList<String> roles, String
project, String module) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String name = "", message = "", emailHeader
  = "", emailAddress = "", project2 = "";
  int prevId = 0;

  try {
    conn = DBConnect.getConnection();
    String sql = "";

    if(module.equals(VirholexConstants.VIRHO_
    REFERENCES)) {
      project2 = project + " " +
      VirholexConstants.VIRHO_REFERENCE;
    } else
    if(module.equals(VirholexConstants.VIRHO_
    HOTSPOTS)) {
      project2 = project + " " +
      VirholexConstants.VIRHO_HOTSPOT;
    } else {
      project2 = project;
    }

    for(int i=0; i<users.size(); i++) {

      if(module.equals(VirholexConstants.VIRH
      O_IMAGES)) {
        sql = "SELECT prev_id FROM user_prev
        WHERE username=? AND involvement=? AND
        (role_name LIKE '%Image%' OR
        role_name='Restricted User')";
      } else
      if(module.equals(VirholexConstants.VIRH
      O_EXPERIMENTS)) {
        sql = "SELECT prev_id FROM user_prev
        WHERE username=? AND involvement=? AND
        role_name LIKE '%Experiment%'";
      } else
      if(module.equals(VirholexConstants.VIRH
      O_MODELS)) {
        sql = "SELECT prev_id FROM user_prev
        WHERE username=? AND involvement=? AND
        role_name LIKE '%Model%'";
      } else
      if(module.equals(VirholexConstants.VIRH
      O_REFERENCES)) {
        sql = "SELECT prev_id FROM user_prev
        WHERE username=? AND involvement=? AND
        (role_name LIKE '%Collection%' OR
        role_name='Restricted User')";
      } else
      if(module.equals(VirholexConstants.VIRH
      O_HOTSPOTS)) {
        sql = "SELECT prev_id FROM user_prev
        WHERE username=? AND involvement=? AND
        role_name LIKE '%Hotspot%'";
      }

      ps = conn.prepareStatement(sql);
      ps.setString(1, users.get(i));
      ps.setString(2, project2);
      rs = ps.executeQuery();

      if(rs.next()) {
      prevId = rs.getInt(1);

      }
      rs.close();
      ps.close();

      if(roles.get(i).equals(VirholexConstant
      s.REMOVE_PRIVILEGE)) {
        sql = "DELETE FROM user_prev WHERE
        prev_id=?";
        ps = conn.prepareStatement(sql);
        ps.setInt(1, prevId);
        ps.executeUpdate();
```

```
        ps.close();

        message = "We regret to inform you
        that your privilege for the project "
        + project + " was removed.";
        emailHeader = "Privilege was changed";

      } else {
        sql = "UPDATE user_prev SET
        role_name=? WHERE prev_id=?";
        ps = conn.prepareStatement(sql);
        ps.setString(1, roles.get(i));
        ps.setInt(2, prevId);
        ps.executeUpdate();
        ps.close();

        message = "We would like to inform you
        that your privilege for the project "
        + project + " was changed to " +
        roles.get(i) + ".";
        emailHeader = "Privilege was changed";

      }

      sql = "SELECT first_name, last_name,
      email FROM users WHERE username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, users.get(i));
      rs = ps.executeQuery();

      if(rs.next()){
        name = rs.getString(1) + " " +
        rs.getString(2);
        emailAddress = rs.getString(3);
      }

      rs.close();
      ps.close();
      // send message
      SendEmail.sendEmail(emailAddress,
      emailHeader, message, name);
    }

   } catch (Exception e) {
     e.printStackTrace();
   } finally {
     if(rs != null) rs.close();
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }

 }

}
```

RequestPrivilegeForm.java

```
package com.virholex.virus;

import java.util.ArrayList;

public class RequestPrivilegeForm extends
ActionForm {

private static final long serialVersionUID = -
5055845311125236664L;
private String username;
private String firstName;
private String lastName;
private String requestMessage;
private String project;
private String module;
private String command;
private String type;
private String dateAdded;
private String privilege;
private String kind;
private ArrayList<UserProfileVO> users;
private ArrayList<String> roles;

public void setUsername(String username) {
  this.username = username;
}
public String getUsername() {
  return username;
}
public void setFirstName(String firstName) {
  this.firstName = firstName;
}
public String getFirstName() {
  return firstName;
}
public void setLastName(String lastName) {
  this.lastName = lastName;
}
public String getLastName() {
  return lastName;
}
public void setRequestMessage(String
requestMessage) {
  this.requestMessage = requestMessage;
}
public String getRequestMessage() {
  return requestMessage;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setProject(String project) {
  this.project = project;
}
public String getProject() {
  return project;
}
public void setModule(String module) {
  this.module = module;
}
public String getModule() {
  return module;
}
public void setType(String type) {
  this.type = type;
}
public String getType() {
  return type;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
  return dateAdded;
}
public void setPrivilege(String privilege) {
  this.privilege = privilege;
}
public String getPrivilege() {
  return privilege;
}
public void setKind(String kind) {
  this.kind = kind;
}
public String getKind() {
  return kind;
}
public void setUsers(ArrayList<UserProfileVO>
users) {
  this.users = users;
```

```
}
public ArrayList<UserProfileVO> getUsers() {
  return users;
}
public void setRoles(ArrayList<String> roles)
{
  this.roles = roles;
}
public ArrayList<String> getRoles() {
  return roles;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if
  (this.getPrivilege().equals(VirholexConstant
  s.DEFAULT_VALUE)) {
    errors.add("privilege", new
    ActionMessage("error.select.default",
    "privilege"));
  }

  return errors;
}

}
```

RequestPrivilegeForm.java

```
package com.virholex.virus;

import java.util.ArrayList;

import com.virholex.general.VO;

public class RequestPrivilegeVO implements VO
{

  private static final long serialVersionUID =
  4748200742953986979L;
  private String username;
  private String firstName;
  private String lastName;
  private String requestMessage;
  private String project;
  private String module;
  private String type;
  private String dateAdded;
  private String privilege;
  private String kind;
  private ArrayList<UserProfileVO> users;
  private ArrayList<String> roles;

  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public String getFirstName() {
    return firstName;
  }
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setRequestMessage(String
  requestMessage) {
    this.requestMessage = requestMessage;
  }
  public String getRequestMessage() {
    return requestMessage;
  }
  public void setProject(String project) {
    this.project = project;
  }
  public String getProject() {
    return project;
  }
  public void setModule(String module) {
    this.module = module;
  }
  public String getModule() {
    return module;
  }
  public void setType(String type) {
    this.type = type;
  }
  public String getType() {
    return type;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }
  public void setPrivilege(String privilege) {
    this.privilege = privilege;
  }
  public String getPrivilege() {
    return privilege;
  }
  public void setKind(String kind) {
    this.kind = kind;
  }
  public String getKind() {
    return kind;
  }
  public void
  setUsers(ArrayList<UserProfileVO> users) {
    this.users = users;
  }
  public ArrayList<UserProfileVO> getUsers() {
    return users;
  }
  public void setRoles(ArrayList<String>
  roles) {
    this.roles = roles;
  }
  public ArrayList<String> getRoles() {
    return roles;
  }

}
```

ResetPasswordAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;

public class ResetPasswordAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
```

```
  ForgotPasswordForm passwordForm =
  (ForgotPasswordForm) form;
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();

  String forward =
  "virholex.virus.resetPassword";
  String state =
  request.getParameter("state");
  String key = request.getParameter("key");
  String randomSequence =
  request.getParameter("rand");
  String crack =
  request.getParameter("crack");
  String username =
  passwordForm.getUsername();

  if(passwordForm.getCommand()!=null &&
  passwordForm.getCommand().equals("Change
  Password")) {

    if(!Utilities.isEmpty(username)) {
      if(!virusWorker.checkUsername(username,
      randomSequence)) {
        passwordForm.setInvalidUsername(true);
      }
    }

    errors =
    passwordForm.validateAccount(mapping,
    request);

    if(errors.isEmpty()){
      virusWorker.resetPassword(passwordForm.ge
      tUsername(),
      passwordForm.getNewPassword());
      request.setAttribute("status", "reset
      password");
      forward = "virholex.virus.statusCheck";
    } else {
      saveErrors(request, errors);
      }
    } else {

      if(!virusWorker.checkActivationKey(random
      Sequence)) {
        request.setAttribute("status", "invalid
        key");
        forward = "virholex.virus.statusCheck";
      } else {
        request.setAttribute("state", state);
        request.setAttribute("key", key);
        request.setAttribute("randomSequence",
        randomSequence);
        request.setAttribute("crack", crack);
      }
  }

  return mapping.findForward(forward);
}
}
```

UserProfileAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class UserProfileAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  UserProfileForm userForm = (UserProfileForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.virus.userProfile";
  String username =
  (String)session.getAttribute("userLogin");

  if(userForm.getCommand()!=null &&
  userForm.getCommand().equals("Account
  Manager")) {
    forward = "virholex.virus.editProfile";
  } else if(userForm.getCommand()!=null &&
  userForm.getCommand().equals("Unsubscribe"))
  {
    forward = "virholex.virus.cancelAccount";
  } else {
    userVO.setUsername(username);
    userVO = (UserProfileVO)
    virusWorker.read(userVO);

    if(virusWorker.checkProjectLeader(userVO){
      if(virusWorker.checkAdmin(userVO)) {
        if(virusWorker.getAdminCount()==1) {
          userForm.setUnsubscribe("disabled");
        }
      }
    }

    PropertyUtils.copyProperties(userForm,
    userVO);
  }

  return mapping.findForward(forward);
}
}
```

UserProfileDAO.java

```
package com.virholex.virus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;

public class UserProfileDAO {

  public static UserProfileVO
  getUserDetails(UserProfileVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
```

```
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {

      String sql = "SELECT username,
      first_name, last_name, email,
      affiliation, job_title,
      AES_DECRYPT(password,?) AS password,
      status FROM users WHERE username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      ps.setString(2, vo.getUsername());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setUsername(rs.getString(1));
        vo.setFirstName(rs.getString(2));
        vo.setLastName(rs.getString(3));
        vo.setEmailAddress(rs.getString(4));
        vo.setAffiliation(rs.getString(5));
        vo.setJobTitle(rs.getString(6));
        vo.setPassword(rs.getString(7));
        vo.setStatus(rs.getString(8));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

  public static UserProfileVO
  getUserName(UserProfileVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {

      String sql = "SELECT username,
      first_name, last_name FROM users WHERE
      username=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setUsername(rs.getString("username")
        );
        vo.setFirstName(rs.getString("first_nam
        e"));
        vo.setLastName(rs.getString("last_name"
        ));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

}
```

UserProfileForm.java

```
package com.virholex.virus;

import java.util.ArrayList;

import org.apache.struts.action.ActionForm;

public class UserProfileForm extends
ActionForm {

  private static final long serialVersionUID =
  4496916054957869681L;
  private String username;
  private String emailAddress;
  private String firstName;
  private String lastName;
  private String affiliation;
  private String jobTitle;
  private String accountManager;
  private String status;
  private String command;
  private String password;
  private String unsubscribe;
  private String projectRole;
  private ArrayList<UserProfileVO> users;

  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public String getFirstName() {
    return firstName;
  }
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setAffiliation(String
  affiliation) {
    this.affiliation = affiliation;
  }
  public String getAffiliation() {
    return affiliation;
  }
  public void setJobTitle(String jobTitle) {
    this.jobTitle = jobTitle;
  }
  public String getJobTitle() {
    return jobTitle;
  }
  public void setAccountManager(String
  accountManager) {
    this.accountManager = accountManager;
  }
  public String getAccountManager() {
    return accountManager;
  }
  public void setUnsubscribe(String
  unsubscribe) {
    this.unsubscribe = unsubscribe;
  }
  public String getUnsubscribe() {
    return unsubscribe;
  }
  public void setUsername(String username) {
    this.username = username;
```

```
  }
  public String getUsername() {
    return username;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setProjectRole(String
  projectRole) {
    this.projectRole = projectRole;
  }
  public String getProjectRole() {
    return projectRole;
  }
  public void
  setUsers(ArrayList<UserProfileVO> users) {
    this.users = users;
  }
  public ArrayList<UserProfileVO> getUsers() {
    return users;
  }

}
```

UserProfileVO.java

```
package com.virholex.virus;

import java.util.ArrayList;

import com.virholex.general.VO;

public class UserProfileVO implements VO {

  private static final long serialVersionUID =
  3517773054769370180L;
  private String username;
  private String emailAddress;
  private String firstName;
  private String lastName;
  private String affiliation;
  private String jobTitle;
  private String approvalDate;
  private String status;
  private String password;
  private String newPassword;
  private String oldPassword;
  private String dbPassword;
  private boolean isChangePassword;
  private boolean hasEmail;
  private String projectRole;
  private ArrayList<UserProfileVO> users;

  public void setEmailAddress(String
  emailAddress) {
    this.emailAddress = emailAddress;
  }
  public String getEmailAddress() {
    return emailAddress;
  }
  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }
  public String getFirstName() {
    return firstName;
  }
  public void setLastName(String lastName) {
    this.lastName = lastName;
  }
  public String getLastName() {
    return lastName;
  }
  public void setAffiliation(String
  affiliation) {
    this.affiliation = affiliation;
  }
  public String getAffiliation() {
    return affiliation;
  }
  public void setJobTitle(String jobTitle) {
    this.jobTitle = jobTitle;
  }
  public String getJobTitle() {
    return jobTitle;
  }
  public void setApprovalDate(String
  approvalDate) {
    this.approvalDate = approvalDate;
  }
  public String getApprovalDate() {
    return approvalDate;
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }
  public void setPassword(String password) {
    this.password = password;
  }
  public String getPassword() {
    return password;
  }
  public void setNewPassword(String
  newPassword) {
    this.newPassword = newPassword;
  }
  public String getNewPassword() {
    return newPassword;
  }
  public void setOldPassword(String
  oldPassword) {
    this.oldPassword = oldPassword;
  }
  public String getOldPassword() {
    return oldPassword;
  }
  public void setChangePassword(boolean
  isChangePassword) {
    this.isChangePassword = isChangePassword;
  }
  public boolean isChangePassword() {
    return isChangePassword;
  }
  public void setHasEmail(boolean hasEmail) {
    this.hasEmail = hasEmail;
  }
  public boolean isHasEmail() {
```

```
    return hasEmail;
  }
  public void setDbPassword(String dbPassword)
  {
    this.dbPassword = dbPassword;
  }
  public String getDbPassword() {
    return dbPassword;
  }
  public void setProjectRole(String
  projectRole) {
    this.projectRole = projectRole;
  }
  public String getProjectRole() {
    return projectRole;
  }
  public void
  setUsers(ArrayList<UserProfileVO> users) {
    this.users = users;
  }
  public ArrayList<UserProfileVO> getUsers() {
    return users;
  }

}
```

UserProjectsAction.java

```
package com.virholex.virus;

import java.util.ArrayList;

public class UserProjectsAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ProjectComponent> projectsList =
  new ArrayList<ProjectComponent>();

  String forward =
  "virholex.virus.userProjects";
  String username =
  (String)session.getAttribute("userLogin");
  String involvement = "", role = "", module =
  "";

  if(request.getParameter("evaluate")!=null &&
  request.getParameter("evaluate").equals("Eva
  luate")) {

  } else {

    projectsList =
    virusWorker.fetchProjects(username);

    for(int i=0; i<projectsList.size(); i++) {
      involvement =
      ((ProjectComponent)projectsList.get(i)).g
      etProjectName();
      role =
      ((ProjectComponent)projectsList.get(i)).g
      etRole();
      module = virusWorker.getModule(role);
      ((ProjectComponent)projectsList.get(i)).s
      etModule(module);
      if(module.equals(VirholexConstants.VIRHO_
      REFERENCES)) {
        ((ProjectComponent)projectsList.get(i))
        .setProjectName(involvement.substring(0
        ,
        involvement.indexOf(VirholexConstants.V
        IRHO_REFERENCE)-1));
        involvement =
        ((ProjectComponent)projectsList.get(i))
        .getProjectName();
      }
      if(module.equals(VirholexConstants.VIRHO_
      HOTSPOTS)) {
        ((ProjectComponent)projectsList.get(i))
        .setProjectName(involvement.substring(0
        ,
        involvement.indexOf(VirholexConstants.V
        IRHO_HOTSPOT)-1));
        involvement =
        ((ProjectComponent)projectsList.get(i))
        .getProjectName();
      }
      if(virusWorker.getPrivilege(role) > 2) {
        ((ProjectComponent)projectsList.get(i))
        .setPrivilege(virusWorker.getPrivilege(
        role));
        ((ProjectComponent)projectsList.get(i))
        .setRequestCount(virusWorker.countReque
        st(involvement, module));
      }
    }

    request.setAttribute("projects",
    projectsList);

  }

  return mapping.findForward(forward);
}
}
```

UserProjectsAction.java

```
package com.virholex.virus;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.VirholexConstants;

public class UserProjectsDAO {

  public static ArrayList<ProjectComponent>
  fetchUserProjects(String username) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ProjectComponent project = new
    ProjectComponent();
    ArrayList<ProjectComponent> projectList =
    new ArrayList<ProjectComponent>();

    try {
      String sql = "SELECT involvement,
      role_name FROM user_prev WHERE username=?
```

```
    AND NOT involvement='DEFAULT_PROJECT'
    ORDER BY involvement";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    rs = ps.executeQuery();

    while(rs.next()) {
      project.setProjectName(rs.getString("in
      volvement"));
      project.setRole(rs.getString("role_name
      "));
      projectList.add(project);
      project = new ProjectComponent();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return projectList;
}

public static String getProjectModule(String
role) throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String module = "";

  try {
    String sql = "SELECT module FROM
    user_roles WHERE role_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, role);
    rs = ps.executeQuery();

    if(rs.next()) {
      module = rs.getString("module");
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return module;
}

public static String getProjectVirus(String
involvement, String type) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String virus = "";

  try {
    String sql = "";
    if(type.equals(VirholexConstants.PROJECT)
    )
      sql = "SELECT virus FROM project WHERE
      project_name=?";
    else
    if(type.equals(VirholexConstants.PROJECTS
    ET))
      sql = "SELECT virus FROM project_set
      WHERE project_set_name=?";

    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, involvement);
    rs = ps.executeQuery();

    if(rs.next()) {
      virus = rs.getString("virus");
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return virus;
}

public static int countProjectRequest(String
involvement, String module) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  int counter = 0;

  try {
    String sql = "SELECT request FROM
    requests WHERE involvement=? AND
    module=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, involvement);
    ps.setString(2, module);
    rs = ps.executeQuery();

    while(rs.next()) {
      counter++;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return counter;
 }
}
```

ViewProjectRequestAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class ViewProjectRequestAction extends
Action {

@Override
```

```
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  RequestPrivilegeForm requestForm =
  (RequestPrivilegeForm) form;
  RequestPrivilegeVO requestVO = new
  RequestPrivilegeVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session = request.getSession();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.virus.projectRequest";
  String userEval =
  request.getParameter("userEval");
  String project =
  (String)session.getAttribute("project");
  String module =
  (String)session.getAttribute("module");
  String roleName = "", dateAdded = "";

  PropertyUtils.copyProperties(requestVO,
  requestForm);

  if(request.getParameter("module")!=null)
    module = request.getParameter("module");
  if(request.getParameter("project")!=null)
    project = request.getParameter("project");

  requestVO.setModule(module);
  requestVO.setProject(project);

  if(requestForm.getCommand()!=null &&
  requestForm.getCommand().equals("Cancel")) {
    forward = "virholex.virus.userProjects";
  } else if(!Utilities.isEmpty(userEval)) {

    requestVO.setUsername(userEval);
    requestVO.setModule(module);
    requestVO.setProject(project);

    if(requestForm.getCommand()!=null) {

      if(requestForm.getCommand().equals("Appro
      ve")) {

        errors = requestForm.validate(mapping,
        request);

        if(errors.isEmpty()) {
          if(module.equals(VirholexConstants.VIR
          HO_IMAGES))
            requestVO.setKind(VirholexConstants.
            PROJECTSET);
          else
            requestVO.setKind(VirholexConstants.
            PROJECT);

          virusWorker.approveUserPrivilege(reque
          stVO);
        } else {
          saveErrors(request, errors);

          requestVO.setRoles(virusWorker.getUser
          Roles(module));
          requestVO =
          (RequestPrivilegeVO)virusWorker.readPr
          ojectRequest(requestVO);
          dateAdded = requestVO.getDateAdded();

          if(dateAdded.contains(".0")) {
            dateAdded = dateAdded.substring(0,
            dateAdded.lastIndexOf(".0"));
          }

          requestVO.setDateAdded(Utilities.dateF
          ormat(dateAdded));

          forward =
          "virholex.virus.projectRequestEvaluate
          ";
        }

      } else if(requestForm.getCommand()!=null
      &&
      requestForm.getCommand().equals("Deny"))
      {
        virusWorker.denyUserPrivilege(requestVO
        );
      } else if(requestForm.getCommand()!=null
      && requestForm.getCommand().equals("No
      Change")) {
        // do nothing
      }

      if(errors.isEmpty()) {
        roleName = getRoleName(module);
        requestVO.setUsers(virusWorker.getProje
        ctRequestUsers(username, project,
        module, roleName));
      }

    }else {

      requestVO.setRoles(virusWorker.getUserRol
      es(module));
      requestVO =
      (RequestPrivilegeVO)virusWorker.readProje
      ctRequest(requestVO);
      dateAdded = requestVO.getDateAdded();

      if(dateAdded.contains(".0")) {
        dateAdded = dateAdded.substring(0,
        dateAdded.lastIndexOf(".0"));
      }

      requestVO.setDateAdded(Utilities.dateForm
      at(dateAdded));

      forward =
      "virholex.virus.projectRequestEvaluate";
    }

  } else {
    roleName = getRoleName(module);
    requestVO.setUsers(virusWorker.getProjectRe
    questUsers(username, project, module,
    roleName));
    session.setAttribute("module", module);
    session.setAttribute("project", project);
  }

  PropertyUtils.copyProperties(requestForm,
  requestVO);

  return mapping.findForward(forward);
}

public String getRoleName(String module) {

  if(module.equals(VirholexConstants.VIRHO_IMA
  GES))
    return
    VirholexConstants.IMAGE_SET_COORDINATOR_DES
    C;
```

```
  else
  if(module.equals(VirholexConstants.VIRHO_EXP
  ERIMENTS))
    return
    VirholexConstants.EXPERIMENT_LEAD_INVESTIGA
    TOR_DESC;
  else
  if(module.equals(VirholexConstants.VIRHO_REF
  ERENCES))
    return
    VirholexConstants.COLLECTION_COORDINATOR_DE
    SC;
  else
  if(module.equals(VirholexConstants.VIRHO_MOD
  ELS))
    return
    VirholexConstants.MODEL_COORDINATOR_DESC;
  else
  if(module.equals(VirholexConstants.VIRHO_HOT
  SPOTS))
    return
    VirholexConstants.HOTSPOT_MANAGER_DESC;

  return "";
}
}
```

ViewUserProfileDetailsAction.java

```
package com.virholex.virus;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;

public class ViewUserProfileDetailsAction
extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  UserProfileForm userForm = (UserProfileForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();

  String forward =
  "virholex.virus.viewUserDetail";
  String username = "";

  if(request.getParameter("user") != null)
    username = request.getParameter("user");

  userVO.setUsername(username);
  userVO = (UserProfileVO)
  virusWorker.read(userVO);

  if(userForm.getCommand() != null &&
  userForm.getCommand().equals("Promote to
  Admin")) {
    virusWorker.updateStatus(userVO.getUsername
    (), VirholexConstants.STATUS_ADMIN);
    userVO.setStatus(VirholexConstants.STATUS_A
    DMIN);
  } else if(userForm.getCommand() != null &&
  userForm.getCommand().equals("Remove from
  Admin")) {
    virusWorker.updateStatus(userVO.getUsername
    (), VirholexConstants.STATUS_ACTIVATED);
    userVO.setStatus(VirholexConstants.STATUS_A
    CTIVATED);
  } else if(userForm.getCommand() != null &&
  userForm.getCommand().equals("View Virholex
  Users")) {
    forward = "virholex.virus.viewUsers";
  }

  PropertyUtils.copyProperties(userForm,
  userVO);

  return mapping.findForward(forward);
}
}
```

ViewUsersAction.java

```
package com.virholex.virus;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward = "virholex.virus.viewUsers";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", field = "", key = "";

  for(int i=0; i < letters.length() ; i++){
    if(virusWorker.getUsers(userVO,
    Character.toString(letters.charAt(i))).size
    () > 0)
      letterList.add(Character.toString(letters
      .charAt(i)));
    else
      letterList.add("-");
  }

  usersForm.setLetters(letterList);
```

```
  if(usersForm.getSearch() != null &&
  usersForm.getSearch().equals("Search
  Members")) {

    if(usersForm.getSearchTerm() != null) {
      key = usersForm.getSearchTerm();
      usersForm.setKey(key);
    }
    if(usersForm.getSearchOption() != null) {
      field = usersForm.getSearchOption();
      usersForm.setField(field);
    }

    userVO.setUsers(virusWorker.searchUsers(use
    rVO, username, field, key));

  } else {
    if(request.getParameter("list") != null)
      list = request.getParameter("list");
      userVO.setUsers(virusWorker.getUsers(user
      VO, list));
  }

  PropertyUtils.copyProperties(usersForm,
  userVO);

  return mapping.findForward(forward);
}
}
```

ViewUsersDAO.java

```
package com.virholex.virus;

import java.sql.Connection;

public class ViewUsersDAO {

  public static ArrayList<UserProfileVO>
  getUsers(UserProfileVO vo, String letter)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<UserProfileVO> userDetails = new
    ArrayList<UserProfileVO>();
    UserProfileVO detailsVO = new
    UserProfileVO();

    try {
      String sql = "SELECT username,
      first_name, last_name, email,
      approval_date FROM users WHERE last_name
      LIKE ? AND (status='approved' OR
      status='activated' OR status='admin')
      ORDER BY first_name, last_name,
      username";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      rs = ps.executeQuery();

      while(rs.next()) {
        detailsVO.setUsername(rs.getString(1));
        detailsVO.setFirstName(rs.getString(2))
        ;
        detailsVO.setLastName(rs.getString(3));
        detailsVO.setEmailAddress(rs.getString(
        4));
        detailsVO.setApprovalDate(rs.getString(
        5));
        userDetails.add(detailsVO);
        detailsVO = new UserProfileVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return userDetails;
  }

  public static ArrayList<UserProfileVO>
  searchUser(UserProfileVO vo, String
  username, String field, String key) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<UserProfileVO> userDetails = new
    ArrayList<UserProfileVO>();
    UserProfileVO detailsVO = new
    UserProfileVO();

    try {
      String sql = "";
      conn = DBConnect.getConnection();

      if(!field.equals("name")) {
        sql = "SELECT username, first_name,
        last_name, email, approval_date FROM
        users WHERE " + field + " LIKE ? AND
        (status='approved' OR
        status='activated' OR status='admin')
        AND NOT username=? ORDER BY last_name,
        first_name, username";
        ps = conn.prepareStatement(sql);
        ps.setString(1, "%"+key+"%");
        ps.setString(2, username);
      } else {
        sql = "SELECT username, first_name,
        last_name, email, approval_date FROM
        users WHERE (first_name LIKE ? OR
        last_name LIKE ?) AND
        (status='approved' OR
        status='activated' OR status='admin')
        AND NOT username=? ORDER BY last_name,
        first_name, username";
        ps = conn.prepareStatement(sql);
        ps.setString(1, "%"+key+"%");
        ps.setString(2, "%"+key+"%");
        ps.setString(3, username);
      }

      rs = ps.executeQuery();

      while(rs.next()) {
        detailsVO.setUsername(rs.getString(1));
        detailsVO.setFirstName(rs.getString(2))
        ;
        detailsVO.setLastName(rs.getString(3));
        detailsVO.setEmailAddress(rs.getString(
        4));
        detailsVO.setApprovalDate(rs.getString(
        5));
        userDetails.add(detailsVO);
        detailsVO = new UserProfileVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
```

```
  return userDetails;
}

public static ArrayList<UserProfileVO>
getUsers(String project, String letter,
String module) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<UserProfileVO> userDetails = new
  ArrayList<UserProfileVO>();
  UserProfileVO vo = new UserProfileVO();

  try {
    String sql = "";
    conn = DBConnect.getConnection();

    if(module.equals(VirholexConstants.VIRHO_
    IMAGES)) {
      sql = "SELECT users.username,
      first_name, last_name, affiliation,
      role_name FROM users, user_prev WHERE
      last_name LIKE ? AND
      user_prev.username=users.username AND
      user_prev.involvement=? AND
      kind='projectset' ORDER BY last_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      ps.setString(2, project);
    } else
    if(module.equals(VirholexConstants.VIRHO_
    MODELS)) {
      sql = "SELECT users.username,
      first_name, last_name, affiliation,
      role_name FROM users, user_prev WHERE
      last_name LIKE ? AND
      user_prev.username=users.username AND
      user_prev.involvement=? AND
      kind='project' AND role_name LIKE
      '%Model%' ORDER BY last_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      ps.setString(2, project);
    } else
    if(module.equals(VirholexConstants.VIRHO_
    EXPERIMENTS)) {
      sql = "SELECT users.username,
      first_name, last_name, affiliation,
      role_name FROM users, user_prev WHERE
      last_name LIKE ? AND
      user_prev.username=users.username AND
      user_prev.involvement=? AND
      kind='project' AND role_name LIKE
      '%Experiment%' ORDER BY last_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      ps.setString(2, project);
    } else
    if(module.equals(VirholexConstants.VIRHO_
    REFERENCES)) {
      sql = "SELECT users.username,
      first_name, last_name, affiliation,
      role_name FROM users, user_prev WHERE
      last_name LIKE ? AND
      user_prev.username=users.username AND
      user_prev.involvement=? AND
      kind='project' ORDER BY last_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      ps.setString(2, project+"
      "+VirholexConstants.VIRHO_REFERENCE);
    } else
    if(module.equals(VirholexConstants.VIRHO_
    HOTSPOTS)) {
      sql = "SELECT users.username,
      first_name, last_name, affiliation,
      role_name FROM users, user_prev WHERE
      last_name LIKE ? AND
      user_prev.username=users.username AND
      user_prev.involvement=? AND role_name
      LIKE '%Hotspot%' ORDER BY last_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      ps.setString(2, project+"
      "+VirholexConstants.VIRHO_HOTSPOT);
    }

    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setUsername(rs.getString(1));
      vo.setFirstName(rs.getString(2));
      vo.setLastName(rs.getString(3));
      vo.setAffiliation(rs.getString(4));
      vo.setProjectRole(rs.getString(5));
      userDetails.add(vo);
      vo = new UserProfileVO();
    }

   } finally {
     if(rs != null) rs.close();
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }

   return userDetails;
  }

}
```

ViewUsersForm.java

```
package com.virholex.virus;

import java.util.ArrayList;

import org.apache.struts.action.ActionForm;

public class ViewUsersForm extends ActionForm
{

  private static final long serialVersionUID =
  1507782025215752070L;
  private String searchTerm;
  private String searchOption;
  private String search;
  private ArrayList<String> letters;
  private String key;
  private String field;
  private ArrayList<UserProfileVO> users;
  private String command;

  public void setSearchTerm(String searchTerm)
  {
    this.searchTerm = searchTerm;
  }
  public String getSearchTerm() {
    return searchTerm;
  }
  public void setSearchOption(String
  searchOption) {
    this.searchOption = searchOption;
  }
  public String getSearchOption() {
    return searchOption;
```

```
  }
  public void setSearch(String search) {
    this.search = search;
  }
  public String getSearch() {
    return search;
  }
  public void setLetters(ArrayList<String>
  letters) {
    this.letters = letters;
  }
  public ArrayList<String> getLetters() {
    return letters;
  }
  public void setKey(String key) {
    this.key = key;
  }
  public String getKey() {
    return key;
  }
  public void setField(String field) {
    this.field = field;
  }
  public String getField() {
    return field;
  }
  public void
  setUsers(ArrayList<UserProfileVO> users) {
    this.users = users;
  }
  public ArrayList<UserProfileVO> getUsers() {
    return users;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }

}
```

VirhoVIRUSWorker.java

```
package com.virholex.virus;

import java.util.ArrayList;

import com.virholex.general.VO;
import com.virholex.general.Worker;

public class VirhoVIRUSWorker implements
Worker {

  public RegisterAccountVO
  insertUser(RegisterAccountVO vo) throws
  Exception {
    return (RegisterAccountVO)
    RegisterAccountDAO.insertUser(vo);
  }

  public void updateUser(UserProfileVO vo)
  throws Exception {
    EditUserProfileDAO.updateUser(vo);
  }

  public UserProfileVO read(UserProfileVO vo)
  throws Exception {
    return (UserProfileVO)
    UserProfileDAO.getUserDetails(vo);
  }

  public ArrayList<UserProfileVO>
  searchUsers(UserProfileVO vo, String
username, String field, String key) throws
Exception {
  return ViewUsersDAO.searchUser(vo,
  username, field, key);
}

public String
checkUsername(RegisterAccountVO vo) throws
Exception {
  return
  RegisterAccountDAO.checkUsername(vo);
}

public boolean checkUsername(String
username, String key) throws Exception {
  return
  ForgotPasswordDAO.checkUsername(username,
  key);
}

public boolean checkActivationKey(String
key) throws Exception {
  return
  ForgotPasswordDAO.checkActivationKey(key);
}

public String checkEmail(RegisterAccountVO
vo) throws Exception {
  return RegisterAccountDAO.checkEmail(vo);
}

public boolean checkEmail(UserProfileVO vo)
throws Exception {
  return EditUserProfileDAO.checkEmail(vo);
}

public String checkPassword(UserProfileVO
vo) throws Exception {
  return
  EditUserProfileDAO.checkPassword(vo);
}

public boolean
checkProjectLeader(UserProfileVO vo) throws
Exception {
  return AdminPageDAO.isProjectLeader(vo);
}

public boolean checkProject(String project)
throws Exception {
  return
  RequestPrivilegeDAO.checkProject(project);
}

public boolean checkRequest(String username,
String project) throws Exception {
  return
  RequestPrivilegeDAO.checkRequestPrivilege(u
  sername, project);
}

public boolean checkAdmin(UserProfileVO vo)
throws Exception {
  return AdminPageDAO.isAdmin(vo);
}

public ArrayList<UserProfileVO>
getUsers(UserProfileVO vo, String letter)
throws Exception {
  return ViewUsersDAO.getUsers(vo, letter);
}

public ArrayList<UserProfileVO>
getUsers(String project, String letter,
String module) throws Exception {
```

```
  return ViewUsersDAO.getUsers(project,
  letter, module);
}

public UserProfileVO getName(UserProfileVO
vo) throws Exception {
  return
  (UserProfileVO)UserProfileDAO.getUserName(v
  o);
}

public UserProfileVO getName(String
username) throws Exception {
  UserProfileVO vo = new UserProfileVO();
  vo.setUsername(username);
  return
  (UserProfileVO)UserProfileDAO.getUserName(v
  o);
}

public int getAdminCount() throws Exception
{
  return AdminPageDAO.getTotalAdmin();
}

public ArrayList<UserProfileVO>
getPendingAccount(UserProfileVO vo) throws
Exception {
  return AdminPageDAO.getPendingAccount(vo);
}

public ArrayList<UserProfileVO>
getReactivateAccount(UserProfileVO vo)
throws Exception {
  return
  AdminPageDAO.getReactivateAccount(vo);
}

public void updateStatus(String username,
String status) throws Exception {
  AdminPageDAO.changeAdminStatus(username,
  status);
}

public void evaluateAccount(String username,
String status) throws Exception {
  AdminPageDAO.evaluateAccount(username,
  status);
}

public void grantPrivilege(String username,
String involvement, String type) throws
Exception {
  FetchPrivilegesDAO.grantPrivilege(username,
  involvement, type);
}

public void sendPrivilege(RequestPrivilegeVO
vo, String username) throws Exception {
  RequestPrivilegeDAO.sendRequestPrivilege(vo
  , username);
}

public boolean checkAccount(ForgotPasswordVO
vo) throws Exception {
  return ForgotPasswordDAO.checkAccount(vo);
}

public void resetPassword(ForgotPasswordVO
vo) throws Exception {
  ForgotPasswordDAO.resetPassword(vo);
}

public LoginAccountVO
checkAccount(LoginAccountVO vo) throws
Exception {
  return
  (LoginAccountVO)LoginAccountDAO.checkAccoun
  t(vo);
}

public String getUserStatus(String username)
throws Exception {
  return
  LoginAccountDAO.getUserStatus(username);
}

public void changeStatus(String username)
throws Exception {
  AdminPageDAO.changeApprovedStatus(username)
  ;
}

public ArrayList<String>
checkPassword(String username, String
password) throws Exception {
  return
  CancelAccountDAO.verifyPassword(username,
  password);
}

public void cancelAccount(String username,
String email) throws Exception {
  CancelAccountDAO.cancelAccount(username,
  email);
}

public ArrayList<ProjectComponent>
fetchProjects(String username) throws
Exception {
  return
  UserProjectsDAO.fetchUserProjects(username)
  ;
}

public String getModule(String role) throws
Exception {
  return
  UserProjectsDAO.getProjectModule(role);
}

public int countRequest(String involvement,
String module) throws Exception {
  return
  UserProjectsDAO.countProjectRequest(involve
  ment, module);
}

public int getPrivilege(String role) throws
Exception {
  return
  FetchPrivilegesDAO.getPrivilegeLevel(role);
}

public ArrayList<String> getUserRoles(String
module) throws Exception {
  return
  RequestPrivilegeDAO.getUserRoles(module);
}

public ArrayList<UserProfileVO>
getProjectRequestUsers(String username,
String project, String module, String
roleName) throws Exception {
  return
  RequestPrivilegeDAO.getProjectRequestUsers(
  username, project, module, roleName);
}

public RequestPrivilegeVO
readProjectRequest(RequestPrivilegeVO vo)
throws Exception {
```

```
    return
    RequestPrivilegeDAO.readProjectRequest(vo);
  }

  public void
  approveUserPrivilege(RequestPrivilegeVO vo)
  throws Exception {
    RequestPrivilegeDAO.approveUserPrivilege(vo
    );
  }

  public void
  denyUserPrivilege(RequestPrivilegeVO vo)
  throws Exception {
    RequestPrivilegeDAO.denyUserPrivilege(vo);
  }

  public void
  updateProjectPrivilege(ArrayList<String>
  users, ArrayList<String> roles, String
  project, String module) throws Exception {
    RequestPrivilegeDAO.updateProjectPrivilege(
    users, roles, project, module);
  }

  public void resetPassword(String username,
  String password) throws Exception {
    ForgotPasswordDAO.resetPassword(username,
    password);
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VIRHO EXPERIMENTS

AddExperimentAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.Utilities;
import com.virholex.images.VirhoImagesWorker;
import
com.virholex.images.ViewImageSetDetailsVO;

public class AddExperimentAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewExperimentDetailsForm experimentForm =
  (ViewExperimentDetailsForm) form;
  ViewExperimentDetailsVO experimentVO = new
  ViewExperimentDetailsVO();
  ViewImageSetDetailsVO imageSetVO = new
  ViewImageSetDetailsVO();
  ExperimentComponentsVO expComponentVO = new
  ExperimentComponentsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewImageSetDetailsVO> imagesList
  = new ArrayList<ViewImageSetDetailsVO>();
  ArrayList<ExperimentComponentsVO>
  componentList = new
  ArrayList<ExperimentComponentsVO>();

  String forward =
  "virholex.experiments.addExperiment";
  String username =
  (String)session.getAttribute("userLogin");
  String project =
  (String)session.getAttribute("project");
  String experiment = "", timePoints = "";
  int projectId = 0, experimentId = 0;

  PropertyUtils.copyProperties(experimentVO,
  experimentForm);

  if(experimentForm.getCommand() != null &&
  experimentForm.getCommand().equals("Submit")
  ) {

    errors = experimentForm.validate(mapping,
    request);

    if(errors.isEmpty()){

      // Get experiment components
      String[] methodName =
      request.getParameterValues("methodName");
      String[] methodDesc =
      request.getParameterValues("methodDesc");
      String[] condition =
      request.getParameterValues("condition");
      String[] varType =
      request.getParameterValues("varType");
      String[] varValue =
      request.getParameterValues("varValue");
      String[] chemActivator =
      request.getParameterValues("chemActivator
      ");

      experimentVO.setAddedBy(username);
      experimentVO.setProjectName(project);
      experimentVO.setExperimentDate(Utilities.
      formatDate(experimentVO.getExperimentDate
      ()) + " 00:00:00");
      timePoints =
      Utilities.formatDate(experimentVO.getTime
      Points()) + " " +
      Utilities.formatTime(String.valueOf(exper
      imentForm.getTpHours()),
      String.valueOf(experimentForm.getTpMinute
      s()),
      String.valueOf(experimentForm.getTpSecond
      s()));
```

```
    experimentVO.setTimePoints(timePoints);
    experimentWorker.insertExperiment(experim
    entVO);

    projectId = experimentVO.getProjectId();
    experimentId =
    experimentVO.getExperimentId();

    for(int i=0; i<methodName.length; i++) {
      if(!Utilities.isEmpty(methodName[i]) ||
      !Utilities.isEmpty(methodDesc[i]) ||
      !Utilities.isEmpty(condition[i]) ||
      !Utilities.isEmpty(varType[i]) ||
      !Utilities.isEmpty(varValue[i]) ||
      !Utilities.isEmpty(chemActivator[i])) {
        expComponentVO.setProjectId(projectId)
        ;
        expComponentVO.setExperimentId(experim
        entId);
        expComponentVO.setConditionValue(condi
        tion[i]);
        expComponentVO.setMethodName(methodNam
        e[i]);
        expComponentVO.setMethodDescription(me
        thodDesc[i]);
        expComponentVO.setVariableType(varType
        [i]);
        expComponentVO.setVariableValue(varVal
        ue[i]);
        expComponentVO.setChemActivatorName(ch
        emActivator[i]);
        componentList.add(expComponentVO);
        expComponentVO = new
        ExperimentComponentsVO();
      }
    }

    experimentWorker.insertComponents(compone
    ntList);

    experiment =
    experimentVO.getExperimentNameDB();
    session.setAttribute("experiment",
    experiment);

    forward =
    "virholex.experiments.viewExperimentDetai
    ls";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("projectId",
    experimentVO.getProjectId());
    redirect.addParameter("experiment",
    experiment);
    return redirect;
  } else {
    imagesList =
    imageWorker.getImageSets(imageSetVO);
    saveErrors(request, errors);
  }

} else if(experimentForm.getCommand() !=
null &&
experimentForm.getCommand().equals("Cancel")
) {
  forward =
  "virholex.experiments.viewProjectDetails";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("project", project);
  return redirect;
} else {
    imagesList =
    imageWorker.getImageSets(imageSetVO);
  }

  request.setAttribute("imagesList",
  imagesList);

  return mapping.findForward(forward);
}
}
```

AddExperimentComponentAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;

public class AddExperimentComponentAction
extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {

  ExperimentComponentsForm expComponentForm =
  (ExperimentComponentsForm) form;
  ExperimentComponentsVO expComponentVO = new
  ExperimentComponentsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ExperimentComponentsVO>
  componentList = new
  ArrayList<ExperimentComponentsVO>();

  String forward =
  "virholex.experiments.addExperimentComponent
  ";
  String experiment =
  (String)session.getAttribute("experiment");
  int projectId = 0, experimentId = 0;

  PropertyUtils.copyProperties(expComponentVO,
  expComponentForm);

  try {
    if(request.getParameter("projectId")!=null)
    {
      projectId =
      Integer.parseInt(request.getParameter("pr
      ojectId"));
      expComponentVO.setProjectId(projectId);
    }
    if(request.getParameter("experimentId")!=nu
    ll) {
      experimentId =
      Integer.parseInt(request.getParameter("ex
      perimentId"));
      expComponentVO.setExperimentId(experiment
      Id);
    }
  } catch(NumberFormatException e) {
    e.printStackTrace();
  }

  if(expComponentForm.getCommand()!=null &&
  expComponentForm.getCommand().equals("Submit
  ")) {

    projectId = expComponentVO.getProjectId();
    experimentId =
    expComponentVO.getExperimentId();
```

```
  String[] methodName =
  request.getParameterValues("methodName");
  String[] methodDesc =
  request.getParameterValues("methodDesc");
  String[] condition =
  request.getParameterValues("condition");
  String[] varType =
  request.getParameterValues("varType");
  String[] varValue =
  request.getParameterValues("varValue");
  String[] chemActivator =
  request.getParameterValues("chemActivator")
  ;

  for(int i=0; i<methodName.length; i++) {
    if(!Utilities.isEmpty(methodName[i]) ||
    !Utilities.isEmpty(methodDesc[i]) ||
    !Utilities.isEmpty(condition[i]) ||
    !Utilities.isEmpty(varType[i]) ||
    !Utilities.isEmpty(varValue[i]) ||
    !Utilities.isEmpty(chemActivator[i])) {
      expComponentVO.setProjectId(projectId);
      expComponentVO.setExperimentId(experime
      ntId);
      expComponentVO.setConditionValue(condit
      ion[i]);
      expComponentVO.setMethodName(methodName
      [i]);
      expComponentVO.setMethodDescription(met
      hodDesc[i]);
      expComponentVO.setVariableType(varType[
      i]);
      expComponentVO.setVariableValue(varValu
      e[i]);
      expComponentVO.setChemActivatorName(che
      mActivator[i]);
      componentList.add(expComponentVO);
      expComponentVO = new
      ExperimentComponentsVO();
    }
  }

  experimentWorker.insertComponents(component
  List);

  forward =
  "virholex.experiments.viewExperimentDetails
  ";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectId",
  projectId);
  redirect.addParameter("experiment",
  experiment);
  return redirect;

} else
if(expComponentForm.getCommand()!=null &&
expComponentForm.getCommand().equals("Cancel
")) {
  forward =
  "virholex.experiments.viewExperimentDetails
  ";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectId",
  projectId);
  redirect.addParameter("experiment",
  experiment);
  return redirect;
}

PropertyUtils.copyProperties(expComponentFor
m, expComponentVO);
  return mapping.findForward(forward);
}
}
```

AddProjectAction.java

```
package com.virholex.experiments;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.virus.VirhoVIRUSWorker;
import com.virholex.virus.UserProfileVO;

public class AddProjectAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsForm projectForm =
  (ViewProjectDetailsForm) form;
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  UserProfileVO userVO = new UserProfileVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.experiments.addProject";
  String username =
  (String)session.getAttribute("userLogin");
  String project = "", author = "";

  PropertyUtils.copyProperties(projectVO,
  projectForm);

  if(projectForm.getCommand() != null &&
  projectForm.getCommand().equals("Submit")) {

    if(projectVO.getProjectName() != null) {
      project = projectVO.getProjectName();
      if(experimentWorker.checkProject(project)
      ) {
        projectForm.setHasProject(true);
      }
    }

    errors = projectForm.validate(mapping,
    request);

    if(errors.isEmpty()){
      // Get name of the author
      userVO = virusWorker.getName(username);
      author = userVO.getFirstName() + " " +
      userVO.getLastName();
```

```
    projectVO.setAuthor(author);
    // Insert date in database
    experimentWorker.insertProject(projectVO,
    username);

    forward =
    "virholex.experiments.viewProjectDetails"
    ;
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("project",
    project);
    return redirect;
  } else {
    saveErrors(request, errors);
  }

 } else if(projectForm.getCommand() != null
 &&
 projectForm.getCommand().equals("Cancel")) {
   forward =
   "virholex.experiments.viewProjects";
 }

 return mapping.findForward(forward);
}
}
```

AddProjectDAO.java

```
package com.virholex.experiments;

import java.sql.Connection;

public class AddProjectDAO {

 public static boolean checkProject(String
 project) throws SQLException,
 NamingException {

   Connection conn = null;
   PreparedStatement ps = null;
   ResultSet rs = null;
   boolean sameProjectName = false;

   try {
     String sql = "SELECT project_name FROM
     project WHERE project_name=?";
     conn = DBConnect.getConnection();
     ps = conn.prepareStatement(sql);
     ps.setString(1, project);
     rs = ps.executeQuery();

     if(rs.next()) {
       sameProjectName = true;
     }

   } finally {
     if(rs != null) rs.close();
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }

   return sameProjectName;
 }

 public static void
 addProject(ViewProjectDetailsVO vo, String
 username) throws SQLException,
 NamingException {

   Connection conn = null;
   PreparedStatement ps = null;
   ResultSet rs = null;

   try {
     conn = DBConnect.getConnection();

     String sql = "INSERT INTO project
     (project_name, description, status,
     author, date_added, date_modified) VALUES
     (?,?,?,?,NOW(),NOW())";
     ps = conn.prepareStatement(sql);
     ps.setString(1, vo.getProjectName());
     ps.setString(2, vo.getDescription());
     ps.setString(3, vo.getStatus());
     ps.setString(4, vo.getAuthor());
     ps.executeUpdate();
     ps.close();

     sql = "SELECT LAST_INSERT_ID() AS id";
     ps = conn.prepareStatement(sql);
     rs = ps.executeQuery();
     rs.next();
     vo.setProjectId(rs.getInt("id"));
     rs.close();
     ps.close();

     sql = "INSERT INTO user_prev (username,
     role_name, involvement, date_added,
     date_modified, kind, project_id)
     VALUES(?,?,?,NOW(),NOW(),?,?)";
     ps = conn.prepareStatement(sql);
     ps.setString(1, username);
     ps.setString(2,
     VirholexConstants.EXPERIMENT_LEAD_INVESTI
     GATOR_DESC);
     ps.setString(3, vo.getProjectName());
     ps.setString(4,
     VirholexConstants.PROJECT);
     ps.setInt(5, vo.getProjectId());
     ps.executeUpdate();
     ps.close();

     sql = "INSERT INTO user_prev (username,
     role_name, involvement, date_added,
     date_modified, kind, project_id)
     VALUES(?,?,?,NOW(),NOW(),?,?)";
     ps = conn.prepareStatement(sql);
     ps.setString(1, username);
     ps.setString(2,
     VirholexConstants.MODEL_COORDINATOR_DESC)
     ;
     ps.setString(3, vo.getProjectName());
     ps.setString(4,
     VirholexConstants.PROJECT);
     ps.setInt(5, vo.getProjectId());
     ps.executeUpdate();

   } finally {
     if(rs != null) rs.close();
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }

 }

}
```

EditExperimentAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;

public class EditExperimentAction extends
Action {
```

```
@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewExperimentDetailsForm experimentForm =
  (ViewExperimentDetailsForm) form;
  ViewExperimentDetailsVO experimentVO = new
  ViewExperimentDetailsVO();
  ViewImageSetDetailsVO imageSetVO = new
  ViewImageSetDetailsVO();
  ExperimentComponentsVO expComponentVO = new
  ExperimentComponentsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewImageSetDetailsVO> imagesList
  = new ArrayList<ViewImageSetDetailsVO>();
  ArrayList<ExperimentComponentsVO>
  componentList = new
  ArrayList<ExperimentComponentsVO>();
  ArrayList<Integer> deleteComponentList = new
  ArrayList<Integer>();

  String forward =
  "virholex.experiments.editExperiment";
  String project =
  (String)session.getAttribute("project");
  String experiment =
  (String)session.getAttribute("experiment");
  String timePoints = "";
  int projectId = 0, experimentId = 0;

  PropertyUtils.copyProperties(experimentVO,
  experimentForm);

  if(request.getParameter("experiment")!=null)
  {
    experiment =
    request.getParameter("experiment");
  }
  if(request.getParameter("projectId")!=null)
  {
    projectId =
    Integer.parseInt(request.getParameter("proj
    ectId"));
    project =
    experimentWorker.getProjectName(projectId);
    session.setAttribute("project", project);
  }

  if(experimentForm.getCommand() != null &&
  experimentForm.getCommand().equals("Submit")
  ) {

    errors = experimentForm.validate(mapping,
    request);

    if(errors.isEmpty()){

      String[] myImageSets =
      request.getParameterValues("myImageSets")
      ;
      String[] methodName =
      request.getParameterValues("methodName");
      String[] methodDesc =
      request.getParameterValues("methodDesc");
      String[] condition =
      request.getParameterValues("condition");
      String[] varType =
      request.getParameterValues("varType");
      String[] varValue =
      request.getParameterValues("varValue");
      String[] chemActivator =
      request.getParameterValues("chemActivator
      ");
      String[] componentId =
      request.getParameterValues("componentId")
      ;
      String[] deleteComponent =
      request.getParameterValues("deleteCompone
      nt");
      ArrayList<String> deleteComponentIdList =
      new ArrayList<String>();

      // Values contained in imagesetList are
      experiments the user wants to be removed
      if(myImageSets!=null) {
        for(int i=0; i<myImageSets.length; i++)
        {
          try {
            imageSetVO.setProjectSetId(Integer.p
            arseInt(myImageSets[i]));
            imagesList.add(imageSetVO);
            imageSetVO = new
            ViewImageSetDetailsVO();
          } catch(NumberFormatException e) {
            e.printStackTrace();
          }
        }
      }

      try{

        if(deleteComponent != null) {
          for(int i=0; i<deleteComponent.length;
          i++) {
            deleteComponentIdList.add(deleteComp
            onent[i]);
          }
        }

        for(int i=0; i<methodName.length; i++)
        {
          if(!Utilities.isEmpty(methodName[i])
          || !Utilities.isEmpty(methodDesc[i])
          || !Utilities.isEmpty(condition[i]) ||
          !Utilities.isEmpty(varType[i]) ||
          !Utilities.isEmpty(varValue[i]) ||
          !Utilities.isEmpty(chemActivator[i]))
          {

            if(!deleteComponentIdList.contains(c
            omponentId[i])) {
              expComponentVO.setExperimentCompone
              ntId(Integer.parseInt(componentId[i
              ]));
              expComponentVO.setConditionValue(co
              ndition[i]);
              expComponentVO.setMethodName(method
              Name[i]);
              expComponentVO.setMethodDescription
              (methodDesc[i]);
              expComponentVO.setVariableType(varT
              ype[i]);
              expComponentVO.setVariableValue(var
              Value[i]);
              expComponentVO.setChemActivatorName
              (chemActivator[i]);
              componentList.add(expComponentVO);
              expComponentVO = new
              ExperimentComponentsVO();
            } else {
              deleteComponentList.add(Integer.par
              seInt(componentId[i]));
            }
          }
```

```
    }
   } catch(NumberFormatException e) {
    e.printStackTrace();
   }

   experimentWorker.editComponents(component
   List);
   experimentWorker.deleteComponents(deleteC
   omponentList);

   experimentVO.setImages(imagesList);
   experimentVO.setExperimentNameDB(experime
   nt);
   experimentVO.setProjectName(project);
   experimentVO.setExperimentDate(Utilities.
   formatDate(experimentVO.getExperimentDate
   ()) + " 00:00:00");
   timePoints =
   Utilities.formatDate(experimentVO.getTime
   Points()) + " " +
   Utilities.formatTime(String.valueOf(exper
   imentForm.getTpHours()),
   String.valueOf(experimentForm.getTpMinute
   s()),
   String.valueOf(experimentForm.getTpSecond
   s()));
   experimentVO.setTimePoints(timePoints);
   experimentWorker.updateExperiment(experim
   entVO);
   experiment =
   experimentVO.getExperimentName();

   forward =
   "virholex.experiments.viewExperimentDetai
   ls";
   ActionRedirect redirect = new
   ActionRedirect(mapping.findForward(forwar
   d));
   redirect.addParameter("projectId",
   experimentVO.getProjectId());
   redirect.addParameter("experiment",
   experiment);
   return redirect;

  } else {
   saveErrors(request, errors);

   experimentId =
   experimentVO.getExperimentId();
   experimentVO.setImages(imageWorker.getIma
   geSets(experimentId));
   imagesList =
   imageWorker.getImageSets(imageSetVO);
   imagesList = getImageSets(imagesList,
   experimentVO.getImages());
   experimentVO.setExperimentComponents(expe
   rimentWorker.getExperimentComponents(expe
   rimentId));
  }

} else if(experimentForm.getCommand() !=
null &&
experimentForm.getCommand().equals("Cancel")
) {
  forward =
  "virholex.experiments.viewExperimentDetails
  ";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectId",
  experimentVO.getProjectId());
  redirect.addParameter("experiment",
  experiment);
  return redirect;
 } else {

  experimentVO.setProjectName(project);
  experimentVO.setExperimentName(experiment);
  experimentVO =
  (ViewExperimentDetailsVO)experimentWorker.r
  eadExperiment(experimentVO);
  experimentVO.setExperimentDate(Utilities.da
  teCalendarFormat(experimentVO.getExperiment
  Date()));

  String tp = experimentVO.getTimePoints();
  int firstIndex = tp.indexOf(":");
  int lastIndex = tp.lastIndexOf(":");

  try {
   experimentVO.setTpHours(Integer.parseInt(
   tp.substring(firstIndex-2, firstIndex)));
   experimentVO.setTpMinutes(Integer.parseIn
   t(tp.substring(firstIndex+1,
   lastIndex)));
   experimentVO.setTpSeconds(Integer.parseIn
   t(tp.substring(lastIndex+1)));
   experimentVO.setTimePoints(Utilities.date
   CalendarFormat(tp));
  } catch(NumberFormatException e) {
   e.printStackTrace();
  }

  experimentId =
  experimentVO.getExperimentId();
  experimentVO.setExperimentId(experimentId);
  experimentVO.setImages(imageWorker.getImage
  Sets(experimentId));
  imagesList =
  imageWorker.getImageSets(imageSetVO);
  imagesList = getImageSets(imagesList,
  experimentVO.getImages());
  experimentVO.setExperimentComponents(experi
  mentWorker.getExperimentComponents(experime
  ntId));
 }

 PropertyUtils.copyProperties(experimentForm,
 experimentVO);

 session.setAttribute("experiment",
 experiment);
 request.setAttribute("imagesList",
 imagesList);

 return mapping.findForward(forward);
}

public ArrayList<ViewImageSetDetailsVO>
getImageSets(ArrayList<ViewImageSetDetailsVO>
imageSetsList,
ArrayList<ViewImageSetDetailsVO>
oldImageSetsList) {

 for(int i=0; i<imageSetsList.size(); i++) {
  for(int j=0; j<oldImageSetsList.size();
  j++) {
   if(((ViewImageSetDetailsVO)imageSetsList.
   get(i)).getProjectSetName().equals(((View
   ImageSetDetailsVO)oldImageSetsList.get(j)
   ).getProjectSetName())) {
    imageSetsList.remove(i);
    i--;
    break;
   }
  }
 }

 return imageSetsList;
}
```

```
}
```

EditProjectAction.java

```
package com.virholex.experiments;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.virus.UserProfileVO;
import com.virholex.virus.VirhoVIRUSWorker;

public class EditProjectAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsForm projectForm =
  (ViewProjectDetailsForm) form;
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  UserProfileVO userVO = new UserProfileVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.experiments.editProject";
  String username =
  (String)session.getAttribute("userLogin");
  String project = "", author = "";

  if(request.getParameter("project")!=null)
    project = request.getParameter("project");
  else
    project =
    (String)session.getAttribute("project");

  PropertyUtils.copyProperties(projectVO,
  projectForm);

  if(projectForm.getCommand() != null &&
  projectForm.getCommand().equals("Submit")) {
    if(projectVO.getProjectName()!=null &&
    projectVO.getProjectOldName()!=null) {
      if(!projectVO.getProjectName().equals(pro
      jectVO.getProjectOldName())) {
        project = projectVO.getProjectName();

        if(experimentWorker.checkProject(projec
        t)) {
          projectForm.setHasProject(true);
        }
      }
    }
    errors = projectForm.validate(mapping,
    request);

    if(errors.isEmpty()){
      // Get name of author
      userVO = virusWorker.getName(username);
      author = userVO.getFirstName() + " " +
      userVO.getLastName();
      projectVO.setAuthor(author);
      // Update data in database
      experimentWorker.updateProject(projectVO)
      ;

      forward =
      "virholex.experiments.viewProjectDetails"
      ;
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("project",
      project);
      return redirect;
    } else {
      saveErrors(request, errors);
    }
  } else if(projectForm.getCommand() != null
  &&
  projectForm.getCommand().equals("Cancel")) {
    forward =
    "virholex.experiments.viewProjects";
  } else {
    projectVO.setProjectName(project);
    projectVO =
    (ViewProjectDetailsVO)experimentWorker.read
    Project(projectVO);
    projectVO.setProjectOldName(projectVO.getPr
    ojectName());
  }

  PropertyUtils.copyProperties(projectForm,
  projectVO);

  session.setAttribute("project", project);

  return mapping.findForward(forward);
}
}
```

EditProjectDAO.java

```
package com.virholex.experiments;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;

public class EditProjectDAO {

  public static void
  updateProject(ViewProjectDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();
```

```
      String sql = "UPDATE project SET
      project_name=?, description=?, author=?,
      status=?, date_modified=NOW() WHERE
      project_id=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getProjectName());
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getAuthor());
      ps.setString(4, vo.getStatus());
      ps.setInt(5, vo.getProjectId());
      ps.executeUpdate();
      ps.close();

      sql = "UPDATE user_prev SET
      involvement=?, date_modified=NOW() WHERE
      project_id=? AND role_name LIKE
      '%Experiment%'";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getProjectName());
      ps.setInt(2, vo.getProjectId());
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

ExperimentComponentsForm.java

```
package com.virholex.experiments;

import org.apache.struts.action.ActionForm;

public class ExperimentComponentsForm extends
ActionForm {

  private static final long serialVersionUID =
  4272590318956752441L;
  private int chemActivatorId;
  private int conditionId;
  private int methodId;
  private int variableId;
  private int experimentComponentId;
  private int projectId;
  private int experimentId;
  private String chemActivatorName;
  private String conditionValue;
  private String methodName;
  private String methodDescription;
  private int sequenceNumber;
  private String variableType;
  private String variableValue;
  private String command;

  public void setChemActivatorId(int
  chemActivatorId) {
    this.chemActivatorId = chemActivatorId;
  }
  public int getChemActivatorId() {
    return chemActivatorId;
  }
  public void setConditionId(int conditionId)
  {
    this.conditionId = conditionId;
  }
  public int getConditionId() {
    return conditionId;
  }
  public void setMethodId(int methodId) {
    this.methodId = methodId;
  }
  public int getMethodId() {
    return methodId;
  }
  public void setVariableId(int variableId) {
    this.variableId = variableId;
  }
  public int getVariableId() {
    return variableId;
  }
  public void setExperimentComponentId(int
  experimentComponentId) {
    this.experimentComponentId =
    experimentComponentId;
  }
  public int getExperimentComponentId() {
    return experimentComponentId;
  }
  public void setProjectId(int projectId) {
    this.projectId = projectId;
  }
  public int getProjectId() {
    return projectId;
  }
  public void setExperimentId(int
  experimentId) {
    this.experimentId = experimentId;
  }
  public int getExperimentId() {
    return experimentId;
  }
  public void setChemActivatorName(String
  chemActivatorName) {
    this.chemActivatorName = chemActivatorName;
  }
  public String getChemActivatorName() {
    return chemActivatorName;
  }
  public void setConditionValue(String
  conditionValue) {
    this.conditionValue = conditionValue;
  }
  public String getConditionValue() {
    return conditionValue;
  }
  public void setMethodName(String methodName)
  {
    this.methodName = methodName;
  }
  public String getMethodName() {
    return methodName;
  }
  public void setMethodDescription(String
  methodDescription) {
    this.methodDescription = methodDescription;
  }
  public String getMethodDescription() {
    return methodDescription;
  }
  public void setSequenceNumber(int
  sequenceNumber) {
    this.sequenceNumber = sequenceNumber;
  }
  public int getSequenceNumber() {
    return sequenceNumber;
  }
  public void setVariableType(String
  variableType) {
    this.variableType = variableType;
  }
  public String getVariableType() {
    return variableType;
  }
  public void setVariableValue(String
  variableValue) {
    this.variableValue = variableValue;
  }
```

```
  public String getVariableValue() {
    return variableValue;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }

}
```

ExperimentComponentsVO.java

```
package com.virholex.experiments;

import com.virholex.general.VO;

public class ExperimentComponentsVO implements
VO {

  private static final long serialVersionUID =
  1380228423052906364L;
  private int chemActivatorId;
  private int conditionId;
  private int methodId;
  private int variableId;
  private int experimentComponentId;
  private int projectId;
  private int experimentId;
  private String chemActivatorName;
  private String conditionValue;
  private String methodName;
  private String methodDescription;
  private int sequenceNumber;
  private String variableType;
  private String variableValue;
  private String dateAdded;
  private String dateModified;

  public void setChemActivatorId(int
  chemActivatorId) {
    this.chemActivatorId = chemActivatorId;
  }
  public int getChemActivatorId() {
    return chemActivatorId;
  }
  public void setConditionId(int conditionId)
  {
    this.conditionId = conditionId;
  }
  public int getConditionId() {
    return conditionId;
  }
  public void setMethodId(int methodId) {
    this.methodId = methodId;
  }
  public int getMethodId() {
    return methodId;
  }
  public void setVariableId(int variableId) {
    this.variableId = variableId;
  }
  public int getVariableId() {
    return variableId;
  }
  public void setExperimentComponentId(int
  experimentComponentId) {
    this.experimentComponentId =
    experimentComponentId;
  }
  public int getExperimentComponentId() {
    return experimentComponentId;
  }
  public void setProjectId(int projectId) {
    this.projectId = projectId;
  }
  public int getProjectId() {
    return projectId;
  }
  public void setExperimentId(int
  experimentId) {
    this.experimentId = experimentId;
  }
  public int getExperimentId() {
    return experimentId;
  }
  public void setChemActivatorName(String
  chemActivatorName) {
    this.chemActivatorName = chemActivatorName;
  }
  public String getChemActivatorName() {
    return chemActivatorName;
  }
  public void setConditionValue(String
  conditionValue) {
    this.conditionValue = conditionValue;
  }
  public String getConditionValue() {
    return conditionValue;
  }
  public void setMethodName(String methodName)
  {
    this.methodName = methodName;
  }
  public String getMethodName() {
    return methodName;
  }
  public void setMethodDescription(String
  methodDescription) {
    this.methodDescription = methodDescription;
  }
  public String getMethodDescription() {
    return methodDescription;
  }
  public void setSequenceNumber(int
  sequenceNumber) {
    this.sequenceNumber = sequenceNumber;
  }
  public int getSequenceNumber() {
    return sequenceNumber;
  }
  public void setVariableType(String
  variableType) {
    this.variableType = variableType;
  }
  public String getVariableType() {
    return variableType;
  }
  public void setVariableValue(String
  variableValue) {
    this.variableValue = variableValue;
  }
  public String getVariableValue() {
    return variableValue;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }
  public void setDateModified(String
  dateModified) {
    this.dateModified = dateModified;
  }
  public String getDateModified() {
    return dateModified;
  }
```

```
}
```

ViewExperimentDetailsAction.java

```
package com.virholex.experiments;

import javax.servlet.http.HttpServletRequest;

public class ViewExperimentDetailsAction
extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewExperimentDetailsForm experimentForm =
  (ViewExperimentDetailsForm) form;
  ViewExperimentDetailsVO experimentVO = new
  ViewExperimentDetailsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.experiments.viewExperimentDetails"
  ;
  String username =
  (String)session.getAttribute("userLogin");
  String project =
  (String)session.getAttribute("project");
  String experiment =
  (String)session.getAttribute("experiment");
  String roleName = "";
  int projectId = 0, experimentId = 0;

  PropertyUtils.copyProperties(experimentVO,
  experimentForm);

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_EXPERIMENTS);
  if(request.getParameter("experiment")!=null)
    experiment =
    request.getParameter("experiment");
  if(request.getParameter("projectId")!=null)
  {
    projectId =
    Integer.parseInt(request.getParameter("proj
    ectId"));
    project =
    experimentWorker.getProjectName(projectId);
    session.setAttribute("project", project);
  }

  if(experimentForm.getCommand() != null &&
  experimentForm.getCommand().equals("Send
  Membership Request")) {
    forward =
    "virholex.virus.requestPrivilege";
  } else if(experimentForm.getCommand() !=
  null &&
  experimentForm.getCommand().equals("Edit
  Experiment")) {
    forward =
    "virholex.experiments.editExperiment";
  } else if(experimentForm.getCommand() !=
  null &&
  experimentForm.getCommand().equals("Add
  Experiment Component")) {
    forward =
    "virholex.experiments.addExperimentComponen
    t";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("projectId",
    experimentVO.getProjectId());
    redirect.addParameter("experimentId",
    experimentVO.getExperimentId());
    return redirect;
  } else if(experimentForm.getCommand() !=
  null &&
  experimentForm.getCommand().equals("Delete
  Experiment")) {
    experimentWorker.deleteExperiment(experimen
    tVO);
    forward =
    "virholex.experiments.viewProjectDetails";
  } else {
    experimentVO.setProjectName(project);
    experimentVO.setExperimentName(experiment);
    experimentVO =
    (ViewExperimentDetailsVO)experimentWorker.r
    eadExperiment(experimentVO);
    experimentVO.setExperimentDate(Utilities.da
    teFormat(experimentVO.getExperimentDate()))
    ;

    String tp = experimentVO.getTimePoints();
    int firstIndex = tp.indexOf(":");
    int lastIndex = tp.lastIndexOf(":");

    try {
      experimentVO.setTpHours(Integer.parseInt(
      tp.substring(firstIndex-2, firstIndex)));
      experimentVO.setTpMinutes(Integer.parseIn
      t(tp.substring(firstIndex+1,
      lastIndex)));
      experimentVO.setTpSeconds(Integer.parseIn
      t(tp.substring(lastIndex+1)));
      experimentVO.setTimePoints(Utilities.date
      Format(tp));
    } catch(NumberFormatException e) {
      e.printStackTrace();
    }

    experimentId =
    experimentVO.getExperimentId();
    experimentVO.setImages(imageWorker.getImage
    Sets(experimentId));
    experimentVO.setExperimentComponents(experi
    mentWorker.getExperimentComponents(experime
    ntId));

    roleName =
    experimentWorker.getProjectRole(username,
    projectId);

    if(roleName.equals(VirholexConstants.EXPERI
    MENT_LEAD_INVESTIGATOR_DESC))
      experimentVO.setPrivilege(VirholexConstan
      ts.EXPERIMENT_LEAD_INVESTIGATOR);
    else
    if(roleName.equals(VirholexConstants.EXPERI
    MENT_INVESTIGATOR_DESC))
      experimentVO.setPrivilege(VirholexConstan
      ts.EXPERIMENT_INVESTIGATOR);
    else
    if(roleName.equals(VirholexConstants.RESTRI
    CTED_USER_DESC))
      experimentVO.setPrivilege(VirholexConstan
      ts.RESTRICTED_USER);
    else
      experimentVO.setPrivilege(VirholexConstan
      ts.REGISTERED_USER);
```

```
  }

  PropertyUtils.copyProperties(experimentForm,
  experimentVO);

  session.setAttribute("experiment",
  experiment);

  return mapping.findForward(forward);
}
}
```

ViewExperimentDetailsDAO.java

```
package com.virholex.experiments;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.Utilities;
import
com.virholex.images.ViewImageSetDetailsVO;

public class ViewExperimentDetailsDAO {

  public static ViewExperimentDetailsVO
  readExperiment(ViewExperimentDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {

      String sql = "SELECT experiment_name,
      experiments.author AS author,
      experiments.description AS description,
      experiments.date_added AS date_added,
      experiments.date_modified AS
      date_modified, experiment_date,
      virus_used, measurement_interval,
      time_points, biological_replicates,
      technical_replicates, added_by,
      experiments.experiment_id AS
      experiment_id, experiments.project_id AS
      project_id FROM project, experiments
      WHERE
      project.project_id=experiments.project_id
      AND project.project_name=? AND
      experiments.experiment_name=? ORDER BY
      experiments.experiment_name";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getProjectName());
      ps.setString(2, vo.getExperimentName());
      rs = ps.executeQuery();

      while (rs.next()) {
        vo.setExperimentName(rs.getString("expe
        riment_name"));
        vo.setAuthor(rs.getString("author"));
        vo.setDescription(rs.getString("descrip
        tion"));
        vo.setVirus(rs.getString("virus_used"))
        ;
        vo.setExperimentDate(rs.getString("expe
        riment_date"));
        vo.setMeasurementInterval(rs.getInt("me
        asurement_interval"));
        vo.setTimePoints(rs.getString("time_poi
        nts"));
        vo.setBiologicalReplicates(rs.getInt("b
        iological_replicates"));
        vo.setTechnicalReplicates(rs.getInt("te
        chnical_replicates"));
        vo.setAddedBy(rs.getString("added_by"))
        ;
        vo.setDateAdded(rs.getString("date_adde
        d"));
        vo.setDateModified(rs.getString("date_m
        odified"));
        vo.setExperimentId(rs.getInt("experimen
        t_id"));
        vo.setProjectId(rs.getInt("project_id")
        );
      }

    } finally {
      if(rs != null) rs.close();
      if (ps != null) ps.close();
      if (conn != null) conn.close();
    }

    return vo;
}

public static
ArrayList<ExperimentComponentsVO>
getExperimentMethods(
    int experimentId) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ExperimentComponentsVO componentVO = new
  ExperimentComponentsVO();
  ArrayList<ExperimentComponentsVO>
  methodsList = new
  ArrayList<ExperimentComponentsVO>();

  try {

    String sql = "SELECT * FROM methods WHERE
    experiment_id=? ORDER BY method_id";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, experimentId);
    rs = ps.executeQuery();

    while (rs.next()) {
      componentVO.setMethodId(rs.getInt("meth
      od_id"));
      componentVO.setProjectId(rs.getInt("pro
      ject_id"));
      componentVO.setExperimentId(rs.getInt("
      experiment_id"));
      componentVO.setMethodName(rs.getString(
      "method_name"));
      componentVO.setMethodDescription(rs.get
      String("description"));

      methodsList.add(componentVO);
      componentVO = new
      ExperimentComponentsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
```

```
  return methodsList;
}

public static
ArrayList<ExperimentComponentsVO>
getExperimentConditions(int experimentId)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ExperimentComponentsVO componentVO = new
  ExperimentComponentsVO();
  ArrayList<ExperimentComponentsVO>
  conditionsList = new
  ArrayList<ExperimentComponentsVO>();

  try {

    String sql = "SELECT * FROM conditions
    WHERE experiment_id=? ORDER BY
    condition_id";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, experimentId);
    rs = ps.executeQuery();

    while (rs.next()) {
      componentVO.setMethodId(rs.getInt("cond
      ition_id"));
      componentVO.setProjectId(rs.getInt("pro
      ject_id"));
      componentVO.setExperimentId(rs.getInt("
      experiment_id"));
      componentVO.setConditionValue(rs.getStr
      ing("condition_value"));

      conditionsList.add(componentVO);
      componentVO = new
      ExperimentComponentsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return conditionsList;
}

public static
ArrayList<ExperimentComponentsVO>
getExperimentVariables(int experimentId)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ExperimentComponentsVO componentVO = new
  ExperimentComponentsVO();
  ArrayList<ExperimentComponentsVO>
  variablesList = new
  ArrayList<ExperimentComponentsVO>();

  try {

    String sql = "SELECT * FROM variables
    WHERE experiment_id=? ORDER BY
    variable_id";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, experimentId);
    rs = ps.executeQuery();

    while (rs.next()) {
      componentVO.setMethodId(rs.getInt("vari
      able_id"));
      componentVO.setProjectId(rs.getInt("pro
      ject_id"));
      componentVO.setExperimentId(rs.getInt("
      experiment_id"));
      componentVO.setVariableType(rs.getStrin
      g("variable_type"));
      componentVO.setVariableValue(rs.getStri
      ng("variable_value"));

      variablesList.add(componentVO);
      componentVO = new
      ExperimentComponentsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return variablesList;
}

public static
ArrayList<ExperimentComponentsVO>
getExperimentChemActivator(int experimentId)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ExperimentComponentsVO componentVO = new
  ExperimentComponentsVO();
  ArrayList<ExperimentComponentsVO>
  chemActivatorList = new
  ArrayList<ExperimentComponentsVO>();

  try {

    String sql = "SELECT * FROM
    chemical_activators WHERE experiment_id=?
    ORDER BY chem_activator_id";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, experimentId);
    rs = ps.executeQuery();

    while (rs.next()) {
      componentVO.setMethodId(rs.getInt("chem
      _activator_id"));
      componentVO.setProjectId(rs.getInt("pro
      ject_id"));
      componentVO.setExperimentId(rs.getInt("
      experiment_id"));
      componentVO.setChemActivatorName(rs.get
      String("chem_activator_name"));

      chemActivatorList.add(componentVO);
      componentVO = new
      ExperimentComponentsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return chemActivatorList;
}

public static
ArrayList<ExperimentComponentsVO>
```

```
getExperimentComponents(int experimentId)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ExperimentComponentsVO componentVO = new
  ExperimentComponentsVO();
  ArrayList<ExperimentComponentsVO>
  chemActivatorList = new
  ArrayList<ExperimentComponentsVO>();

  try {

    String sql = "SELECT * FROM
    experiment_components WHERE
    experiment_id=? ORDER BY date_added,
    experiment_component_id ASC";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, experimentId);
    rs = ps.executeQuery();

    while (rs.next()) {
      componentVO.setExperimentComponentId(rs
      .getInt("experiment_component_id"));
      componentVO.setProjectId(rs.getInt("pro
      ject_id"));
      componentVO.setExperimentId(rs.getInt("
      experiment_id"));
      componentVO.setMethodName(rs.getString(
      "method_name"));
      componentVO.setMethodDescription(rs.get
      String("description"));
      componentVO.setVariableType(rs.getStrin
      g("variable_type"));
      componentVO.setVariableValue(rs.getStri
      ng("variable_value"));
      componentVO.setConditionValue(rs.getStr
      ing("condition_value"));
      componentVO.setChemActivatorName(rs.get
      String("chem_activator_name"));
      componentVO.setDateAdded(rs.getString("
      date_added"));

      chemActivatorList.add(componentVO);
      componentVO = new
      ExperimentComponentsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return chemActivatorList;
}

public static
ArrayList<ViewExperimentDetailsVO>
getExperiments(int projectSetId) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewExperimentDetailsVO vo = new
  ViewExperimentDetailsVO();
  ArrayList<ViewExperimentDetailsVO>
  experimentList = new
  ArrayList<ViewExperimentDetailsVO>();

  try {

    String sql = "SELECT
    experiments.experiment_id,
    experiments.project_id, experiment_name
    FROM image_sets, experiments WHERE "
        + "image_sets.project_id=? AND
    experiments.experiment_id=image_sets.expe
    riment_id";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, projectSetId);
    rs = ps.executeQuery();

    while (rs.next()) {
      vo.setExperimentId(rs.getInt(1));
      vo.setProjectId(rs.getInt(2));
      vo.setExperimentName(rs.getString(3));

      experimentList.add(vo);
      vo = new ViewExperimentDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return experimentList;
}

public static
ArrayList<ViewExperimentDetailsVO>
getExperiments() throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewExperimentDetailsVO vo = new
  ViewExperimentDetailsVO();
  ArrayList<ViewExperimentDetailsVO>
  experimentList = new
  ArrayList<ViewExperimentDetailsVO>();

  try {

    String sql = "SELECT DISTINCT
    experiment_id, experiments.project_id,
    experiment_name, project_name FROM
    project, experiments WHERE
    experiments.project_id=project.project_id
    ";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();

    while (rs.next()) {
      vo.setExperimentId(rs.getInt(1));
      vo.setProjectId(rs.getInt(2));
      vo.setExperimentName(rs.getString(3));
      vo.setProjectName(rs.getString(4));

      experimentList.add(vo);
      vo = new ViewExperimentDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return experimentList;
}
```

```
public static void
insertExperiment(ViewExperimentDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> titleDBList = new
  ArrayList<String>();

  try {

    conn = DBConnect.getConnection();

    String sql = "SELECT experiment_name FROM
    experiments, project WHERE
    (experiment_name=? OR experiment_name
    LIKE ?) AND project_name=? AND
    experiments.project_id=project.project_id
    ORDER BY experiment_name";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getExperimentName());
    ps.setString(2, vo.getExperimentName() +
    " (%)");
    ps.setString(3, vo.getProjectName());
    rs = ps.executeQuery();

    while (rs.next()) {
      titleDBList.add(rs.getString(1));
    }

    rs.close();
    ps.close();
    // generate new title for duplicates
    vo.setExperimentNameDB(Utilities.generate
    Title(titleDBList,
    vo.getExperimentName()));

    sql = "SELECT project_id FROM project
    WHERE project_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectName());
    rs = ps.executeQuery();

    if (rs.next()) {
      vo.setProjectId(rs.getInt(1));
    }

    rs.close();
    ps.close();

    sql = "INSERT INTO experiments
    (project_id, experiment_name,
    description, author, experiment_date,
    virus_used, measurement_interval,
    time_points, biological_replicates,
    technical_replicates, added_by,
    date_added, date_modified) VALUES
    (?,?,?,?,?,?,?,?,?,?,?,NOW(),NOW())";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.setString(2,
    vo.getExperimentNameDB());
    ps.setString(3, vo.getDescription());
    ps.setString(4, vo.getAuthor());
    ps.setString(5, vo.getExperimentDate());
    ps.setString(6, vo.getVirus());
    ps.setDouble(7,
    vo.getMeasurementInterval());
    ps.setString(8, vo.getTimePoints());
    ps.setInt(9,
    vo.getBiologicalReplicates());
    ps.setInt(10,
    vo.getTechnicalReplicates());
    ps.setString(11, vo.getAddedBy());
    ps.executeUpdate();
    ps.close();

    sql = "SELECT LAST_INSERT_ID() AS id";
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();
    rs.next();
    vo.setExperimentId(rs.getInt("id"));
    rs.close();
    ps.close();

    if(vo.getImageSetId() > 0) {
      sql = "SELECT project_set_name FROM
      project_set WHERE project_set_id=?";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getImageSetId());
      rs = ps.executeQuery();
      rs.next();
      vo.setImageSetName(rs.getString(1));
      rs.close();
      ps.close();

      sql = "INSERT INTO image_sets
      (project_id, experiment_id,
      imageset_name) VALUES (?,?,?)";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getImageSetId());
      ps.setInt(2, vo.getExperimentId());
      ps.setString(3, vo.getImageSetName());
      ps.executeUpdate();
      ps.close();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
updateExperiment(ViewExperimentDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> titleDBList = new
  ArrayList<String>();

  try {

    conn = DBConnect.getConnection();
    String sql = "";

    if(!vo.getExperimentName().equals(vo.getE
    xperimentNameDB())) {

      sql = "SELECT experiment_name FROM
      experiments, project WHERE
      (experiment_name=? OR experiment_name
      LIKE ?) AND project_name=? AND
      experiments.project_id=project.project_
      id ORDER BY experiment_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      vo.getExperimentName());
      ps.setString(2, vo.getExperimentName()
      + " (%)");
      ps.setString(3, vo.getProjectName());
      rs = ps.executeQuery();

      while (rs.next()) {
        titleDBList.add(rs.getString(1));
      }

      rs.close();
```

```
  ps.close();
  // generate new title for duplicates
  vo.setExperimentName(Utilities.generate
  Title(titleDBList,
  vo.getExperimentName()));
}

sql = "SELECT project_id FROM project
WHERE project_name=?";
ps = conn.prepareStatement(sql);
ps.setString(1, vo.getProjectName());
rs = ps.executeQuery();

if (rs.next()) {
  vo.setProjectId(rs.getInt(1));
}

rs.close();
ps.close();

sql = "UPDATE experiments SET
experiment_name=?, description=?,
author=?, experiment_date=?,
virus_used=?, measurement_interval=?,
time_points=?, biological_replicates=?,
technical_replicates=?,
date_modified=NOW() WHERE project_id=?
AND experiment_name=?";
ps = conn.prepareStatement(sql);
ps.setString(1, vo.getExperimentName());
ps.setString(2, vo.getDescription());
ps.setString(3, vo.getAuthor());
ps.setString(4, vo.getExperimentDate());
ps.setString(5, vo.getVirus());
ps.setDouble(6,
vo.getMeasurementInterval());
ps.setString(7, vo.getTimePoints());
ps.setInt(8,
vo.getBiologicalReplicates());
ps.setInt(9,
vo.getTechnicalReplicates());
ps.setInt(10, vo.getProjectId());
ps.setString(11,
vo.getExperimentNameDB());
ps.executeUpdate();
ps.close();

if(!vo.getImages().isEmpty()) {
  sql = "DELETE FROM image_sets WHERE
  project_id=? AND experiment_id=?";

  for(int i=0; i<vo.getImages().size();
  i++) {
    ps = conn.prepareStatement(sql);
    ps.setInt(1,
    ((ViewImageSetDetailsVO)vo.getImages()
    .get(i)).getProjectSetId());
    ps.setInt(2, vo.getExperimentId());
    ps.executeUpdate();
    ps.close();
  }
}

if(vo.getImageSetId() > 0) {
  sql = "SELECT project_set_name FROM
  project_set WHERE project_set_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getImageSetId());
  rs = ps.executeQuery();
  rs.next();
  vo.setImageSetName(rs.getString(1));
  rs.close();
  ps.close();

    sql = "INSERT INTO image_sets
    (project_id, experiment_id,
    imageset_name) VALUES (?,?,?)";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getImageSetId());
    ps.setInt(2, vo.getExperimentId());
    ps.setString(3, vo.getImageSetName());
    ps.executeUpdate();
    ps.close();
  }

 } finally {
  if(rs != null) rs.close();
  if(ps != null) ps.close();
  if(conn != null) conn.close();
 }
}

public static void
deleteExperiment(ViewExperimentDetailsVO vo)
throws SQLException, NamingException {

 Connection conn = null;
 PreparedStatement ps = null;

 try {

  conn = DBConnect.getConnection();

  String sql = "DELETE FROM experiments
  WHERE project_id=? AND experiment_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getProjectId());
  ps.setInt(2, vo.getExperimentId());
  ps.executeUpdate();
  ps.close();

  sql = "DELETE FROM chemical_activators
  WHERE project_id=? AND experiment_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getProjectId());
  ps.setInt(2, vo.getExperimentId());
  ps.executeUpdate();
  ps.close();

  sql = "DELETE FROM conditions WHERE
  project_id=? AND experiment_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getProjectId());
  ps.setInt(2, vo.getExperimentId());
  ps.executeUpdate();
  ps.close();

  sql = "DELETE FROM methods WHERE
  project_id=? AND experiment_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getProjectId());
  ps.setInt(2, vo.getExperimentId());
  ps.executeUpdate();
  ps.close();

  sql = "DELETE FROM variables WHERE
  project_id=? AND experiment_id=?";
  ps = conn.prepareStatement(sql);
  ps.setInt(1, vo.getProjectId());
  ps.setInt(2, vo.getExperimentId());
  ps.executeUpdate();
  ps.close();

 } finally {
  if(ps != null) ps.close();
  if(conn != null) conn.close();
 }
}
```

```
public static void
insertComponents(ArrayList<ExperimentCompone
ntsVO> componentList) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    conn = DBConnect.getConnection();
    String sql = "";

    for(int i=0; i<componentList.size(); i++)
    {
      sql = "INSERT INTO
      experiment_components (project_id,
      experiment_id, method_name,
      description, variable_type,
      variable_value, condition_value,
      chem_activator_name, date_added,
      date_modified) VALUES
      (?,?,?,?,?,?,?,?,NOW(),NOW())";
      ps = conn.prepareStatement(sql);
      ps.setInt(1,
      ((ExperimentComponentsVO)componentList.
      get(i)).getProjectId());
      ps.setInt(2,
      ((ExperimentComponentsVO)componentList.
      get(i)).getExperimentId());
      ps.setString(3,
      ((ExperimentComponentsVO)componentList.
      get(i)).getMethodName());
      ps.setString(4,
      ((ExperimentComponentsVO)componentList.
      get(i)).getMethodDescription());
      ps.setString(5,
      ((ExperimentComponentsVO)componentList.
      get(i)).getVariableType());
      ps.setString(6,
      ((ExperimentComponentsVO)componentList.
      get(i)).getVariableValue());
      ps.setString(7,
      ((ExperimentComponentsVO)componentList.
      get(i)).getConditionValue());
      ps.setString(8,
      ((ExperimentComponentsVO)componentList.
      get(i)).getChemActivatorName());
      ps.executeUpdate();
    }

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
editComponents(ArrayList<ExperimentComponent
sVO> componentList) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    conn = DBConnect.getConnection();
    String sql = "";

    for(int i=0; i<componentList.size(); i++)
    {
      sql = "UPDATE experiment_components SET
      method_name=?, description=?,
      variable_type=?, variable_value=?,
      condition_value=?,
      chem_activator_name=?,
      date_modified=NOW()WHERE
      experiment_component_id=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      ((ExperimentComponentsVO)componentList.
      get(i)).getMethodName());
      ps.setString(2,
      ((ExperimentComponentsVO)componentList.
      get(i)).getMethodDescription());
      ps.setString(3,
      ((ExperimentComponentsVO)componentList.
      get(i)).getVariableType());
      ps.setString(4,
      ((ExperimentComponentsVO)componentList.
      get(i)).getVariableValue());
      ps.setString(5,
      ((ExperimentComponentsVO)componentList.
      get(i)).getConditionValue());
      ps.setString(6,
      ((ExperimentComponentsVO)componentList.
      get(i)).getChemActivatorName());
      ps.setInt(7,
      ((ExperimentComponentsVO)componentList.
      get(i)).getExperimentComponentId());
      ps.executeUpdate();
    }

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static void
deleteComponents(ArrayList<Integer>
componentList) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    conn = DBConnect.getConnection();
    String sql = "";

    for(int i=0; i<componentList.size(); i++)
    {
      sql = "DELETE FROM
      experiment_components WHERE
      experiment_component_id=?";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, componentList.get(i));
      ps.executeUpdate();
    }

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
 }

}
```

ViewExperimentDetailsForm.java

```
package com.virholex.experiments;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
```

```
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

import com.virholex.general.Utilities;
import
com.virholex.images.ViewImageSetDetailsVO;

public class ViewExperimentDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  2044588614863580469L;
  private int experimentId;
  private int projectId;
  private int imageSetId;
  private String imageSetName;
  private String experimentName;
  private String experimentNameDB;
  private String projectName;
  private String description;
  private String author;
  private String experimentDate;
  private String virus;
  private double measurementInterval;
  private String timePoints;
  private int tpHours;
  private int tpMinutes;
  private int tpSeconds;
  private int biologicalReplicates;
  private int technicalReplicates;
  private String addedBy;
  private String dateAdded;
  private String dateModified;
  private String roleName;
  private boolean hasExperiment;
  private ArrayList<ViewImageSetDetailsVO>
  images;
  private ArrayList<ExperimentComponentsVO>
  methods;
  private ArrayList<ExperimentComponentsVO>
  chemActivators;
  private ArrayList<ExperimentComponentsVO>
  variables;
  private ArrayList<ExperimentComponentsVO>
  conditions;
  private ArrayList<ExperimentComponentsVO>
  experimentComponents;
  private String command;
  private int privilege;
  private int userId;

  public void setExperimentId(int
  experimentId) {
    this.experimentId = experimentId;
  }
  public int getExperimentId() {
    return experimentId;
  }
  public void setProjectId(int projectId) {
    this.projectId = projectId;
  }
  public int getProjectId() {
    return projectId;
  }
  public void setImageSetId(int imageSetId) {
    this.imageSetId = imageSetId;
  }
  public int getImageSetId() {
    return imageSetId;
  }
  public void setImageSetName(String
  imageSetName) {
    this.imageSetName = imageSetName;
  }
  public String getImageSetName() {
    return imageSetName;
  }
  public void setExperimentName(String
  experimentName) {
    this.experimentName = experimentName;
  }
  public String getExperimentName() {
    return experimentName;
  }
  public void setExperimentNameDB(String
  experimentNameDB) {
    this.experimentNameDB = experimentNameDB;
  }
  public String getExperimentNameDB() {
    return experimentNameDB;
  }
  public void setProjectName(String
  projectName) {
    this.projectName = projectName;
  }
  public String getProjectName() {
    return projectName;
  }
  public void setDescription(String
  description) {
    this.description = description;
  }
  public String getDescription() {
    return description;
  }
  public void setAuthor(String author) {
    this.author = author;
  }
  public String getAuthor() {
    return author;
  }
  public void setExperimentDate(String
  experimentDate) {
    this.experimentDate = experimentDate;
  }
  public String getExperimentDate() {
    return experimentDate;
  }
  public void setVirus(String virus) {
    this.virus = virus;
  }
  public String getVirus() {
    return virus;
  }
  public void setMeasurementInterval(double
  measurementInterval) {
    this.measurementInterval =
    measurementInterval;
  }
  public double getMeasurementInterval() {
    return measurementInterval;
  }
  public void setTimePoints(String timePoints)
  {
    this.timePoints = timePoints;
  }
  public String getTimePoints() {
    return timePoints;
  }
  public void setTpHours(int tpHours) {
    this.tpHours = tpHours;
  }
  public int getTpHours() {
    return tpHours;
  }
  public void setTpMinutes(int tpMinutes) {
    this.tpMinutes = tpMinutes;
  }
  public int getTpMinutes() {
    return tpMinutes;
  }
```

```
public void setTpSeconds(int tpSeconds) {
  this.tpSeconds = tpSeconds;
}
public int getTpSeconds() {
  return tpSeconds;
}
public void setBiologicalReplicates(int
biologicalReplicates) {
  this.biologicalReplicates =
  biologicalReplicates;
}
public int getBiologicalReplicates() {
  return biologicalReplicates;
}
public void setTechnicalReplicates(int
technicalReplicates) {
  this.technicalReplicates =
  technicalReplicates;
}
public int getTechnicalReplicates() {
  return technicalReplicates;
}
public void setAddedBy(String addedBy) {
  this.addedBy = addedBy;
}
public String getAddedBy() {
  return addedBy;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
  return dateAdded;
}
public void setDateModified(String
dateModified) {
  this.dateModified = dateModified;
}
public String getDateModified() {
  return dateModified;
}
public void setRoleName(String roleName) {
  this.roleName = roleName;
}
public String getRoleName() {
  return roleName;
}
public void setHasExperiment(boolean
hasExperiment) {
  this.hasExperiment = hasExperiment;
}
public boolean isHasExperiment() {
  return hasExperiment;
}
public void
setImages(ArrayList<ViewImageSetDetailsVO>
images) {
  this.images = images;
}
public ArrayList<ViewImageSetDetailsVO>
getImages() {
  return images;
}
public void
setMethods(ArrayList<ExperimentComponentsVO>
methods) {
  this.methods = methods;
}
public ArrayList<ExperimentComponentsVO>
getMethods() {
  return methods;
}
public void
setChemActivators(ArrayList<ExperimentCompon
entsVO> chemActivators) {
  this.chemActivators = chemActivators;
}
public ArrayList<ExperimentComponentsVO>
getChemActivators() {
  return chemActivators;
}
public void
setVariables(ArrayList<ExperimentComponentsV
O> variables) {
  this.variables = variables;
}
public ArrayList<ExperimentComponentsVO>
getVariables() {
  return variables;
}
public void
setConditions(ArrayList<ExperimentComponents
VO> conditions) {
  this.conditions = conditions;
}
public ArrayList<ExperimentComponentsVO>
getConditions() {
  return conditions;
}
public void
setExperimentComponents(ArrayList<Experiment
ComponentsVO> experimentComponents) {
  this.experimentComponents =
  experimentComponents;
}
public ArrayList<ExperimentComponentsVO>
getExperimentComponents() {
  return experimentComponents;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setPrivilege(int privilege) {
  this.privilege = privilege;
}
public int getPrivilege() {
  return privilege;
}
public void setUserId(int userId) {
  this.userId = userId;
}
public int getUserId() {
  return userId;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if
  (Utilities.isEmpty(this.getExperimentName()
  ) || Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getExperimentDate())
  || Utilities.isEmpty(this.getVirus())) {
    errors.add("errorMessage", new
    ActionMessage("error.required.asterisk"))
    ;

    if(Utilities.isEmpty(this.getExperimentNa
    me()))
      errors.add("experimentName", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getAuthor()))
      errors.add("author", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getExperimentDa
    te()))
```

```
      errors.add("experimentDate", new
      ActionMessage(""));
    if(Utilities.isEmpty(this.getVirus()))
      errors.add("virus", new
      ActionMessage(""));
  }

  if(!Utilities.validInput(String.valueOf(thi
  s.getMeasurementInterval()), 5)) {
    errors.add("measurementInterval", new
    ActionMessage("error.invalid",
    "Measurement Interval"));
  }

  if(!Utilities.validInput(String.valueOf(thi
  s.getTpHours()), 4) ||
  !Utilities.validInput(String.valueOf(this.g
  etTpMinutes()), 4) ||
  !Utilities.validInput(String.valueOf(this.g
  etTpSeconds()), 4) || this.getTpHours() >
  24 || this.getTpMinutes() > 60 ||
  this.getTpSeconds() > 60) {
    errors.add("timePoints", new
    ActionMessage("error.invalid", "Time
    Point"));

    if(!Utilities.validInput(String.valueOf(t
    his.getTpHours()), 4) ||
    this.getTpHours() > 24)
      errors.add("tpHours", new
      ActionMessage(""));
    if(!Utilities.validInput(String.valueOf(t
    his.getTpMinutes()), 4) ||
    this.getTpMinutes() > 60)
      errors.add("tpMinutes", new
      ActionMessage(""));
    if(!Utilities.validInput(String.valueOf(t
    his.getTpSeconds()), 4) ||
    this.getTpSeconds() > 60)
      errors.add("tpSeconds", new
      ActionMessage(""));

  } else
  if(Utilities.isEmpty(this.getTimePoints())
  && (this.getTpHours() > 0 ||
  this.getTpMinutes() > 0 ||
  this.getTpSeconds() > 0)) {
    errors.add("timePoints", new
    ActionMessage("error.empty2", "time point
    date"));
    errors.add("timePoints", new
    ActionMessage(""));
  }

  if(!Utilities.validInput(String.valueOf(thi
  s.getBiologicalReplicates()), 4)) {
    errors.add("biologicalReplicates", new
    ActionMessage("error.invalid",
    "Biological Replicates"));
  }

  if(!Utilities.validInput(String.valueOf(thi
  s.getTechnicalReplicates()), 4)) {
    errors.add("technicalReplicates", new
    ActionMessage("error.invalid", "Technical
    Replicates"));
  }

  return errors;
 }

}
```

ViewExperimentDetailsVO.java

```
package com.virholex.experiments;

import java.util.ArrayList;

import com.virholex.general.VO;
import
com.virholex.images.ViewImageSetDetailsVO;

public class ViewExperimentDetailsVO
implements VO {

 private static final long serialVersionUID =
 5072215973144693783L;
 private int experimentId;
 private int projectId;
 private int imageSetId;
 private String imageSetName;
 private String experimentName;
 private String experimentNameDB;
 private String projectName;
 private String description;
 private String author;
 private String experimentDate;
 private String virus;
 private double measurementInterval;
 private String timePoints;
 private int tpHours;
 private int tpMinutes;
 private int tpSeconds;
 private int biologicalReplicates;
 private int technicalReplicates;
 private String addedBy;
 private String dateAdded;
 private String dateModified;
 private String roleName;
 private boolean hasExperiment;
 private ArrayList<ViewImageSetDetailsVO>
 images;
 private ArrayList<ExperimentComponentsVO>
 methods;
 private ArrayList<ExperimentComponentsVO>
 chemActivators;
 private ArrayList<ExperimentComponentsVO>
 variables;
 private ArrayList<ExperimentComponentsVO>
 conditions;
 private ArrayList<ExperimentComponentsVO>
 experimentComponents;
 private int privilege;
 private int userId;

 public void setExperimentId(int
 experimentId) {
   this.experimentId = experimentId;
 }
 public int getExperimentId() {
   return experimentId;
 }
 public void setProjectId(int projectId) {
   this.projectId = projectId;
 }
 public int getProjectId() {
   return projectId;
 }
 public void setImageSetId(int imageSetId) {
   this.imageSetId = imageSetId;
 }
 public int getImageSetId() {
   return imageSetId;
 }
 public void setImageSetName(String
 imageSetName) {
   this.imageSetName = imageSetName;
 }
 public String getImageSetName() {
```

```
  return imageSetName;
}
public void setExperimentName(String
experimentName) {
  this.experimentName = experimentName;
}
public String getExperimentName() {
  return experimentName;
}
public void setExperimentNameDB(String
experimentNameDB) {
  this.experimentNameDB = experimentNameDB;
}
public String getExperimentNameDB() {
  return experimentNameDB;
}
public void setProjectName(String
projectName) {
  this.projectName = projectName;
}
public String getProjectName() {
  return projectName;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setExperimentDate(String
experimentDate) {
  this.experimentDate = experimentDate;
}
public String getExperimentDate() {
  return experimentDate;
}
public void setVirus(String virus) {
  this.virus = virus;
}
public String getVirus() {
  return virus;
}
public void setMeasurementInterval(double
measurementInterval) {
  this.measurementInterval =
  measurementInterval;
}
public double getMeasurementInterval() {
  return measurementInterval;
}
public void setTimePoints(String timePoints)
{
  this.timePoints = timePoints;
}
public String getTimePoints() {
  return timePoints;
}
public void setTpSeconds(int tpSeconds) {
  this.tpSeconds = tpSeconds;
}
public int getTpSeconds() {
  return tpSeconds;
}
public void setTpMinutes(int tpMinutes) {
  this.tpMinutes = tpMinutes;
}
public int getTpMinutes() {
  return tpMinutes;
}
public void setTpHours(int tpHours) {
  this.tpHours = tpHours;
}
public int getTpHours() {
  return tpHours;
}
public void setBiologicalReplicates(int
biologicalReplicates) {
  this.biologicalReplicates =
  biologicalReplicates;
}
public int getBiologicalReplicates() {
  return biologicalReplicates;
}
public void setTechnicalReplicates(int
technicalReplicates) {
  this.technicalReplicates =
  technicalReplicates;
}
public int getTechnicalReplicates() {
  return technicalReplicates;
}
public void setAddedBy(String addedBy) {
  this.addedBy = addedBy;
}
public String getAddedBy() {
  return addedBy;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
return dateAdded;
  }
public void setDateModified(String
dateModified) {
  this.dateModified = dateModified;
}
public String getDateModified() {
  return dateModified;
}
public void setRoleName(String roleName) {
  this.roleName = roleName;
}
public String getRoleName() {
  return roleName;
}
public void setHasExperiment(boolean
hasExperiment) {
  this.hasExperiment = hasExperiment;
}
public boolean isHasExperiment() {
  return hasExperiment;
}
public void
setImages(ArrayList<ViewImageSetDetailsVO>
images) {
  this.images = images;
}
public ArrayList<ViewImageSetDetailsVO>
getImages() {
  return images;
}
public void
setMethods(ArrayList<ExperimentComponentsVO>
methods) {
  this.methods = methods;
}
public ArrayList<ExperimentComponentsVO>
getMethods() {
  return methods;
}
public void
setChemActivators(ArrayList<ExperimentCompon
entsVO> chemActivators) {
  this.chemActivators = chemActivators;
```

```
  }
  public ArrayList<ExperimentComponentsVO>
  getChemActivators() {
    return chemActivators;
  }
  public void
  setVariables(ArrayList<ExperimentComponentsV
  O> variables) {
    this.variables = variables;
  }
  public ArrayList<ExperimentComponentsVO>
  getVariables() {
    return variables;
  }
  public void
  setConditions(ArrayList<ExperimentComponents
  VO> conditions) {
    this.conditions = conditions;
  }
  public ArrayList<ExperimentComponentsVO>
  getConditions() {
    return conditions;
  }
  public void
  setExperimentComponents(ArrayList<Experiment
  ComponentsVO> experimentComponents) {
    this.experimentComponents =
    experimentComponents;
  }
  public ArrayList<ExperimentComponentsVO>
  getExperimentComponents() {
    return experimentComponents;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setUserId(int userId) {
    this.userId = userId;
  }
  public int getUserId() {
    return userId;
  }

}
```

ViewProjectDetailsAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.VirholexConstants;
import com.virholex.general.Utilities;
import com.virholex.models.VirhoModelsWorker;

public class ViewProjectDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsForm projectForm =
  (ViewProjectDetailsForm) form;
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  VirhoModelsWorker modelWorker = new
  VirhoModelsWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> modelList = new
  ArrayList<String>();

  String forward =
  "virholex.experiments.viewProjectDetails";
  String username =
  (String)session.getAttribute("userLogin");
  String project = "", roleName = "";
  int projectId = 0;

  PropertyUtils.copyProperties(projectVO,
  projectForm);

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_EXPERIMENTS)
  if(request.getParameter("project")!=null)
    project = request.getParameter("project");
  else
    project =
    (String)session.getAttribute("project");

  if(projectForm.getCommand() != null &&
  projectForm.getCommand().equals("Add
  Experiment")) {
    forward =
    "virholex.experiments.addExperiment";
  } else if(projectForm.getCommand() != null
  && projectForm.getCommand().equals("Edit
  Project")) {
    forward =
    "virholex.experiments.editProject";
  } else if(projectForm.getCommand() != null
  && projectForm.getCommand().equals("Delete
  Project")) {
    modelList = modelWorker.getModels(project);

    for(int i=0; i<modelList.size(); i++) {
      Utilities.deleteFile(request,
      VirholexConstants.MODELS_REPOSITORY,
      modelList.get(i));
    }

    experimentWorker.deleteProject(projectVO);

    forward =
    "virholex.experiments.viewProjects";
  } else if(projectForm.getCommand() != null
  && projectForm.getCommand().equals("Send
  Membership Request")) {
    forward =
    "virholex.virus.requestPrivilege";
  } else {
    projectVO.setProjectName(project);
    projectVO =
    (ViewProjectDetailsVO)experimentWorker.read
    Project(projectVO);
    projectVO.setDateAdded(Utilities.dateFormat
    (projectVO.getDateAdded()));
    projectVO.setExperiments(experimentWorker.g
    etExperiments(project));
```

```
    projectId = projectVO.getProjectId();

    if(projectForm.getCommand() != null &&
    projectForm.getCommand().equals("View
    Members")) {
      forward =
      "virholex.experiments.viewUsers";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("projectId",
      projectId);
      return redirect;
    } else {
      roleName =
      experimentWorker.getProjectRole(username,
      projectId);

      if(roleName.equals(VirholexConstants.EXPE
      RIMENT_LEAD_INVESTIGATOR_DESC))
        projectVO.setPrivilege(VirholexConstant
        s.EXPERIMENT_LEAD_INVESTIGATOR);
      else
      if(roleName.equals(VirholexConstants.EXPE
      RIMENT_INVESTIGATOR_DESC))
        projectVO.setPrivilege(VirholexConstant
        s.EXPERIMENT_INVESTIGATOR);
      else
      if(roleName.equals(VirholexConstants.REST
      RICTED_USER_DESC))
        projectVO.setPrivilege(VirholexConstant
        s.RESTRICTED_USER);
      else
        projectVO.setPrivilege(VirholexConstant
        s.REGISTERED_USER);
    }
  }

  PropertyUtils.copyProperties(projectForm,
  projectVO);

  session.setAttribute("project", project);

  return mapping.findForward(forward);
}
}
```

ViewProjectDetailsDAO.java

```
package com.virholex.experiments;

import java.sql.Connection;

public class ViewProjectDetailsDAO {

  public static ViewProjectDetailsVO
  readProject(ViewProjectDetailsVO vo) throws
  SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {
    String sql = "SELECT project_id,
    project_name, description, author, status,
    date_added FROM project WHERE
    project_name=? ORDER BY project_name";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectName());
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setProjectId(rs.getInt(1));
      vo.setProjectName(rs.getString(2));
      vo.setDescription(rs.getString(3));
      vo.setAuthor(rs.getString(4));
      vo.setStatus(rs.getString(5));
      vo.setDateAdded(rs.getString(6));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return vo;
}

public static
ArrayList<ViewExperimentDetailsVO>
getExperiments(String project) throws
SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ViewExperimentDetailsVO experimentVO =
    new ViewExperimentDetailsVO();
    ArrayList<ViewExperimentDetailsVO>
    experimentsList = new
    ArrayList<ViewExperimentDetailsVO>();

    try {
      String sql = "SELECT
      experiments.experiment_id AS
      experiment_id,
      experiments.experiment_name AS
      experiment_name,
      experiments.description AS description
      FROM project, experiments WHERE
      project.project_id=experiments.project_
      id AND project.project_name=? ORDER BY
      experiments.experiment_name";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, project);
      rs = ps.executeQuery();

      while(rs.next()) {
        experimentVO.setExperimentId(rs.getInt
        ("experiment_id"));
        experimentVO.setExperimentName(rs.getS
        tring("experiment_name"));
        experimentVO.setDescription(rs.getStri
        ng("description"));
        experimentsList.add(experimentVO);
        experimentVO = new
        ViewExperimentDetailsVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return experimentsList;
  }

  public static String getProjectName(int
  projectId) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String projectName = "";
```

```
  try {
    String sql = "SELECT project_name FROM
    project WHERE project_id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, projectId);
    rs = ps.executeQuery();

    if(rs.next()) {
      projectName = rs.getString(1);
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return projectName;
}

public static void
deleteProject(ViewProjectDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE FROM project WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM user_prev WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM experiments WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM chemical_activators
    WHERE project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM conditions WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM methods WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM variables WHERE
    project_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getProjectId());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM models WHERE
    project_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectName());
    ps.executeUpdate();
    ps.close();

  } finally {
    if (ps != null) ps.close();
    if (conn != null) conn.close();
  }
 }

}
```

ViewProjectDetailsForm.java

```
package com.virholex.experiments;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.models.ViewModelDetailsVO;

public class ViewProjectDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  6191348032739415272L;
  private String command;
  private int privilege;
  private String projectName;
  private String projectOldName;
  private String description;
  private String status;
  private String author;
  private ArrayList<ViewExperimentDetailsVO>
  experiments;
  private ArrayList<ViewModelDetailsVO>
  models;
  private int projectId;
  private String dateAdded;
  private boolean hasProject;

  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setProjectName(String
  projectName) {
    this.projectName = projectName;
  }
  public String getProjectName() {
    return projectName;
```

```
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setStatus(String status) {
  this.status = status;
}
public String getStatus() {
  return status;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setProjectOldName(String
projectOldName) {
  this.projectOldName = projectOldName;
}
public String getProjectOldName() {
  return projectOldName;
}
public void
setExperiments(ArrayList<ViewExperimentDetai
lsVO> experiments) {
  this.experiments = experiments;
}
public ArrayList<ViewExperimentDetailsVO>
getExperiments() {
  return experiments;
}
public void
setModels(ArrayList<ViewModelDetailsVO>
models) {
  this.models = models;
}
public ArrayList<ViewModelDetailsVO>
getModels() {
  return models;
}
public void setProjectId(int projectId) {
  this.projectId = projectId;
}
public int getProjectId() {
  return projectId;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
  return dateAdded;
}
public void setHasProject(boolean
hasProject) {
  this.hasProject = hasProject;
}
public boolean isHasProject() {
  return hasProject;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if(Utilities.isEmpty(this.getProjectName())
  ||
  this.getStatus().equals(VirholexConstants.D
  EFAULT_VALUE)) {
    errors.add("errorMessage", new
    ActionMessage("error.required.asterisk"))
    ;

    if(Utilities.isEmpty(this.getProjectName(
    )))
      errors.add("projectName", new
      ActionMessage(""));
    if(this.getStatus().equals(VirholexConsta
    nts.DEFAULT_VALUE))
      errors.add("status", new
      ActionMessage(""));
  }

  if(this.isHasProject()) {
    errors.add("projectName", new
    ActionMessage("error.notAvailable",
    "project name"));
  }

  return errors;
 }

}
```

ViewProjectDetailsVO.java

```
package com.virholex.experiments;

import java.util.ArrayList;

import com.virholex.general.VO;
import com.virholex.models.ViewModelDetailsVO;

public class ViewProjectDetailsVO implements
VO {

 private static final long serialVersionUID =
 4428502759384751672L;
 private String projectName;
 private String projectOldName;
 private String description;
 private String author;
 private String status;
 private int projectId;
 private ArrayList<ViewExperimentDetailsVO>
 experiments;
 private ArrayList<ViewModelDetailsVO>
 models;
 private int privilege;
 private String dateAdded;
 private boolean hasProject;

 public void setProjectName(String
 projectName) {
  this.projectName = projectName;
 }
 public String getProjectName() {
  return projectName;
 }
 public void setProjectOldName(String
 projectOldName) {
  this.projectOldName = projectOldName;
 }
 public String getProjectOldName() {
  return projectOldName;
 }
 public void setDescription(String
 description) {
  this.description = description;
 }
 public String getDescription() {
  return description;
 }
```

```
  public void setAuthor(String author) {
    this.author = author;
  }
  public String getAuthor() {
    return author;
  }
  public void setStatus(String status) {
    this.status = status;
  }
  public String getStatus() {
    return status;
  }
  public void setProjectId(int projectId) {
    this.projectId = projectId;
  }
  public int getProjectId() {
    return projectId;
  }
  public void
  setExperiments(ArrayList<ViewExperimentDetai
  lsVO> experiments) {
    this.experiments = experiments;
  }
  public ArrayList<ViewExperimentDetailsVO>
  getExperiments() {
    return experiments;
  }
  public void
  setModels(ArrayList<ViewModelDetailsVO>
  models) {
    this.models = models;
  }
  public ArrayList<ViewModelDetailsVO>
  getModels() {
    return models;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }
  public void setHasProject(boolean
  hasProject) {
    this.hasProject = hasProject;
  }
  public boolean isHasProject() {
    return hasProject;
  }

}
```

ViewProjectsAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;

public class ViewProjectsAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsForm projectForm =
  (ViewProjectDetailsForm) form;
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewProjectDetailsVO> projectList
  = new ArrayList<ViewProjectDetailsVO>();

  String forward =
  "virholex.experiments.viewProjects";
  String username =
  (String)session.getAttribute("userLogin");
  String roleName = "";
  int projectId = 0;
  boolean isLead = false;

  if(projectForm.getCommand() != null &&
  projectForm.getCommand().equals("Add
  Project")) {
    forward =
    "virholex.experiments.addProject";
  } else {
    projectList =
    experimentWorker.getProjects(projectVO);
    isLead =
    experimentWorker.isLeadExperimentInvestigat
    or(username);

    for(int i=0; i<projectList.size(); i++) {
      projectId =
      ((ViewProjectDetailsVO)projectList.get(i)
      ).getProjectId();
      roleName =
      experimentWorker.getProjectRole(username,
      projectId);

      if(roleName.equals(VirholexConstants.EXPE
      RIMENT_LEAD_INVESTIGATOR_DESC))
        ((ViewProjectDetailsVO)projectList.get(
        i)).setPrivilege(VirholexConstants.EXPE
        RIMENT_LEAD_INVESTIGATOR);
      else
      if(roleName.equals(VirholexConstants.EXPE
      RIMENT_INVESTIGATOR_DESC))
        ((ViewProjectDetailsVO)projectList.get(
        i)).setPrivilege(VirholexConstants.EXPE
        RIMENT_INVESTIGATOR);
      else
      if(roleName.equals(VirholexConstants.REST
      RICTED_USER_DESC))
        ((ViewProjectDetailsVO)projectList.get(
        i)).setPrivilege(VirholexConstants.REST
        RICTED_USER);
      else
        ((ViewProjectDetailsVO)projectList.get(
        i)).setPrivilege(VirholexConstants.REGI
        STERED_USER);

    }

    request.setAttribute("isLead", isLead);
  }

  request.setAttribute("projects",
  projectList);
```

```
  session.setAttribute("module",
  VirholexConstants.VIRHO_EXPERIMENTS);

  return mapping.findForward(forward);
}
}
```

ViewProjectsDAO.java

```
package com.virholex.experiments;

import java.sql.Connection;

public class ViewProjectsDAO {

  public static
  ArrayList<ViewProjectDetailsVO>
  getProjects(ViewProjectDetailsVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    ArrayList<ViewProjectDetailsVO> projectList
    = new ArrayList<ViewProjectDetailsVO>();

    try {
      String sql = "SELECT project_id,
      project_name, description FROM project
      WHERE project_name NOT LIKE
      '%VirhoHotspot%' AND project_name NOT
      LIKE '%VirhoReference%' ORDER BY
      project_name";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.executeQuery(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        vo.setProjectId(rs.getInt("project_id")
        );
        vo.setProjectName(rs.getString("project
        _name"));
        vo.setDescription(rs.getString("descrip
        tion"));

        projectList.add(vo);
        vo = new ViewProjectDetailsVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return projectList;
  }

  public static String getProjectRole(String
  username, int projectId) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String role = "";

    try {
      String sql = "SELECT role_name FROM
      user_prev WHERE username=? AND
      project_id=? AND role_name LIKE
      '%Experiment%'";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.setInt(2, projectId);
      rs = ps.executeQuery();

      if(rs.next()) {
        role = rs.getString("role_name");
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return role;
  }

  public static boolean
  isLeadExperimentInvestigator(String
  username) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean isLead = false;

    try {
      String sql = "SELECT role_name FROM
      user_prev WHERE username=? AND
      role_name='Experiment Lead Investigator'
      AND involvement='DEFAULT_PROJECT'";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      rs = ps.executeQuery();

      if(rs.next()) {
        isLead = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return isLead;
  }

}
```

ViewUsersAction.java

```
package com.virholex.experiments;

import java.util.ArrayList;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
```

```
HttpSession session =
request.getSession(true);
ArrayList<String> letterList = new
ArrayList<String>();
ArrayList<String> rolesList = new
ArrayList<String>();
ArrayList<String> usersList = new
ArrayList<String>();

String username =
(String)session.getAttribute("userLogin");
String project =
(String)session.getAttribute("project");
String projectId =
request.getParameter("projectId");
String forward =
"virholex.experiments.viewUsers";
String letters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";
String list = "", roleName = "";
int privilege = 0;

if(!Utilities.isEmpty(request.getParameter("
list")))
  list= request.getParameter("list");

for(int i=0; i < letters.length() ; i++){
  if(virusWorker.getUsers(project,
  Character.toString(letters.charAt(i)),
  VirholexConstants.VIRHO_EXPERIMENTS).size()
  > 0)
    letterList.add(Character.toString(letters
    .charAt(i)));
  else
    letterList.add("-");
}

usersForm.setLetters(letterList);

if(usersForm.getCommand()!=null &&
usersForm.getCommand().equals("Change
Privilege Role")) {
  String[] roles =
  request.getParameterValues("role");
  String[] oldRoles =
  request.getParameterValues("oldRole");
  String[] users =
  request.getParameterValues("userName");

  if(roles!=null && oldRoles!=null &&
  users!=null) {
    for(int i=0; i<oldRoles.length; i++) {
      if(!roles[i].equals(oldRoles[i])) {
        rolesList.add(roles[i]);
        usersList.add(users[i]);
      }
    }

    if(!rolesList.isEmpty() &&
    !usersList.isEmpty()) {
      virusWorker.updateProjectPrivilege(user
      sList, rolesList, project,
      VirholexConstants.VIRHO_EXPERIMENTS);
      rolesList = new ArrayList<String>();
    }
  }
}

userVO.setUsers(virusWorker.getUsers(project
, list,
VirholexConstants.VIRHO_EXPERIMENTS));
rolesList =
virusWorker.getUserRoles(VirholexConstants.V
IRHO_EXPERIMENTS);

try {
  roleName =
  experimentWorker.getProjectRole(username,
  Integer.parseInt(projectId));
 } catch(NumberFormatException e) {
  e.printStackTrace();
 }

 if(roleName.equals(VirholexConstants.EXPERIM
 ENT_LEAD_INVESTIGATOR_DESC))
  privilege =
  VirholexConstants.EXPERIMENT_LEAD_INVESTIGA
  TOR;
 else
 if(roleName.equals(VirholexConstants.EXPERIM
 ENT_INVESTIGATOR_DESC))
  privilege =
  VirholexConstants.EXPERIMENT_INVESTIGATOR;
 else
  privilege =
  VirholexConstants.REGISTERED_USER;

 PropertyUtils.copyProperties(usersForm,
 userVO);

 request.setAttribute("privilege",
 privilege);
 request.setAttribute("projectId",
 projectId);
 request.setAttribute("roles", rolesList);

 return mapping.findForward(forward);
}
}
```

VirhoExperimentsWorker.java

```
package com.virholex.experiments;

import java.util.ArrayList;

import com.virholex.general.VO;
import com.virholex.general.Worker;

public class VirhoExperimentsWorker implements
Worker {

 public ArrayList<ViewProjectDetailsVO>
 getProjects(ViewProjectDetailsVO vo) throws
 Exception {
  return ViewProjectsDAO.getProjects(vo);
 }

 public String getProjectRole(String
 username, int projectId) throws Exception {
  return
  ViewProjectsDAO.getProjectRole(username,
  projectId);
 }

 public boolean
 isLeadExperimentInvestigator(String
 username) throws Exception {
  return
  ViewProjectsDAO.isLeadExperimentInvestigato
  r(username);
 }

 public ViewProjectDetailsVO
 readProject(ViewProjectDetailsVO vo) throws
 Exception {
  return (ViewProjectDetailsVO)
  ViewProjectDetailsDAO.readProject(vo);
 }
```

```
public String getProjectName(int projectId)
throws Exception {
  return
  ViewProjectDetailsDAO.getProjectName(projec
  tId);
}

public ArrayList<ViewExperimentDetailsVO>
getExperiments(String project) throws
Exception {
  return
  ViewProjectDetailsDAO.getExperiments(projec
  t);
}

public ArrayList<ViewExperimentDetailsVO>
getExperiments(int projectSetId) throws
Exception {
  return
  ViewExperimentDetailsDAO.getExperiments(pro
  jectSetId);
}

public ArrayList<ViewExperimentDetailsVO>
getExperiments() throws Exception {
  return
  ViewExperimentDetailsDAO.getExperiments();
}

public ViewExperimentDetailsVO
readExperiment(ViewExperimentDetailsVO vo)
throws Exception {
  return
  (ViewExperimentDetailsVO)ViewExperimentDeta
  ilsDAO.readExperiment(vo);
}

public ArrayList<ExperimentComponentsVO>
getMethods(int experimentId) throws
Exception {
  return
  ViewExperimentDetailsDAO.getExperimentMetho
  ds(experimentId);
}

public ArrayList<ExperimentComponentsVO>
getVariables(int experimentId) throws
Exception {
  return
  ViewExperimentDetailsDAO.getExperimentVaria
  bles(experimentId);
}

public ArrayList<ExperimentComponentsVO>
getConditions(int experimentId) throws
Exception {
  return
  ViewExperimentDetailsDAO.getExperimentCondi
  tions(experimentId);
}

public ArrayList<ExperimentComponentsVO>
getChemicalActivators(int experimentId)
throws Exception {
  return
  ViewExperimentDetailsDAO.getExperimentChemA
  ctivator(experimentId);
}

public ArrayList<ExperimentComponentsVO>
getExperimentComponents(int experimentId)
throws Exception {
  return
  ViewExperimentDetailsDAO.getExperimentCompo
  nents(experimentId);
}
```

```
public boolean checkProject(String project)
throws Exception{
  return AddProjectDAO.checkProject(project);
}

public void
insertProject(ViewProjectDetailsVO vo,
String username) throws Exception{
  AddProjectDAO.addProject(vo, username);
}

public void
updateProject(ViewProjectDetailsVO vo)
throws Exception {
  EditProjectDAO.updateProject(vo);
}

public void
insertExperiment(ViewExperimentDetailsVO vo)
throws Exception {
  ViewExperimentDetailsDAO.insertExperiment(v
  o);
}

public void
updateExperiment(ViewExperimentDetailsVO vo)
throws Exception {
  ViewExperimentDetailsDAO.updateExperiment(v
  o);
}

public void
deleteExperiment(ViewExperimentDetailsVO vo)
throws Exception {
  ViewExperimentDetailsDAO.deleteExperiment(v
  o);
}

public void
deleteProject(ViewProjectDetailsVO vo)
throws Exception {
  ViewProjectDetailsDAO.deleteProject(vo);
}

public void
insertComponents(ArrayList<ExperimentCompone
ntsVO> componentList) throws Exception {
  if(!componentList.isEmpty()) {
    ViewExperimentDetailsDAO.insertComponents
    (componentList);
  }
}

public void
editComponents(ArrayList<ExperimentComponent
sVO> componentList) throws Exception {
  if(!componentList.isEmpty()) {
    ViewExperimentDetailsDAO.editComponents(c
    omponentList);
  }
}

public void
deleteComponents(ArrayList<Integer>
componentList) throws Exception {
  if(!componentList.isEmpty()) {
    ViewExperimentDetailsDAO.deleteComponents
    (componentList);
  }
}

public VO read(VO vo) throws Exception {
  // TODO Auto-generated method stub
  return null;
}
```

```
  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VIRHO IMAGES

AddImageAction.java

```
package com.virholex.images;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class AddImageAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageDetailsForm imageForm =
  (ViewImageDetailsForm) form;
  ViewImageDetailsVO imageVO = new
  ViewImageDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  FileOutputStream outputStream = null;
  FormFile formFile = null;
  ArrayList<ViewImageDetailsVO> imageVOList =
  new ArrayList<ViewImageDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward = "virholex.images.addImage";
  String projectSet = "";
  int imgIndex = 0;
  boolean isEdit = false;

  PropertyUtils.copyProperties(imageVO,
  imageForm);

  if(request.getParameter("projectSet")!=null)
  {
    projectSet =
    request.getParameter("projectSet");
    session.setAttribute("projectSet",
    projectSet);
  } else {
    projectSet =
    (String)session.getAttribute("projectSet");
  }

  if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Add to
  Queue")) {

    formFile = imageVO.getFile();
    imageForm.setImagePath(formFile.toString())
    ;

    errors = imageForm.validate(mapping,
    request);

    if(errors.isEmpty()) {

      if(session.getAttribute("imageVOs")!=null
      ) {
        imageVOList =
        (ArrayList<ViewImageDetailsVO>)session.
        getAttribute("imageVOs");
      }

      imageVO.setImagePath(formFile.toString())
      ;
      imageVO.setImagePathDB(Utilities.formatFi
      leName(formFile));
      imageVO.setUploadedBy(username);
      imageVO.setImageSet(projectSet);
      imageVOList.add(imageVO);
      session.setAttribute("imageVOs",
      imageVOList);

      imageVO = new ViewImageDetailsVO();
      imageVO.setPhenolRed("No");

    } else {
      saveErrors(request, errors);
    }

  } else if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Edit
  Metadata")) {

    errors = imageForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      try {
        imageVOList =
        (ArrayList<ViewImageDetailsVO>)session.
        getAttribute("imageVOs");
        if(!imageVOList.isEmpty()) {
          imgIndex =
          Integer.parseInt(request.getParameter(
          "imgIndex"));
          imageVO.setFile(((ViewImageDetailsVO)i
          mageVOList.get(imgIndex)).getFile());
          imageVO.setUploadedBy(username);
          imageVO.setImageSet(projectSet);
          imageVOList.set(imgIndex, imageVO);
          session.setAttribute("imageVOs",
          imageVOList);

          imageVO = new ViewImageDetailsVO();
          imageVO.setPhenolRed("No");
        }
      } catch(NumberFormatException e) {
        e.printStackTrace();
      }
    } else {
      isEdit = true;
```

```
    saveErrors(request, errors);
  }

} else if(imageForm.getCommand()!=null &&
imageForm.getCommand().equals("Upload
Images")) {
  if(session.getAttribute("imageVOs")!=null)
  {
    imageVOList =
    (ArrayList<ViewImageDetailsVO>)session.ge
    tAttribute("imageVOs");
    session.removeAttribute("imageVOs");
    // Save images
    for(int i=0; i<imageVOList.size(); i++) {
      try {
        formFile =
        ((ViewImageDetailsVO)imageVOList.get(i
        )).getFile();
        String filename =
        ((ViewImageDetailsVO)imageVOList.get(i
        )).getImagePathDB();
        String filePath =
        getServlet().getServletContext().getRe
        alPath(VirholexConstants.IMAGES_REPOSI
        TORY) + File.separator + filename;
        outputStream = new
        FileOutputStream(new File(filePath));
        outputStream.write(formFile.getFileDat
        a());
      } catch(FileNotFoundException e) {
        e.printStackTrace();
      } finally {
        outputStream.close();
      }
    }

    imageWorker.insertImages(imageVOList);
  }

  forward =
  "virholex.images.viewImageSetDetais";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectSet",
  projectSet);
  return redirect;

} else if(imageForm.getCommand()!=null &&
imageForm.getCommand().equals("Cancel")) {
  session.removeAttribute("imageVOs");
  forward =
  "virholex.images.viewImageSetDetais";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectSet",
  projectSet);
  return redirect;
} else if(imageForm.getCommand()!=null &&
imageForm.getCommand().equals("Cancel
Edit")) {
  imageVO = new ViewImageDetailsVO();
} else {

  imageVO = new ViewImageDetailsVO();
  imageVO.setPhenolRed("No");

  if(request.getParameter("delete")!=null &&
  session.getAttribute("imageVOs")!=null) {
    try {
      imageVOList =
      (ArrayList<ViewImageDetailsVO>)session.
      getAttribute("imageVOs");

      if(!imageVOList.isEmpty()) {
        imgIndex =
        Integer.parseInt(request.getParameter(
        "delete"));
        imageVOList.remove(imgIndex);
      }
    } catch(NumberFormatException e) {
      e.printStackTrace();
    }
  } else
  if(request.getParameter("edit")!=null &&
  session.getAttribute("imageVOs")!=null) {

    try {
      imageVOList =
      (ArrayList<ViewImageDetailsVO>)session.
      getAttribute("imageVOs");

      if(!imageVOList.isEmpty()) {
        imgIndex =
        Integer.parseInt(request.getParameter(
        "edit"));
        imageVO = new ViewImageDetailsVO();
        PropertyUtils.copyProperties(imageVO,
        (ViewImageDetailsVO)imageVOList.get(im
        gIndex));
        isEdit = true;
        request.setAttribute("imgIndex",
        imgIndex);
      }
    } catch(NumberFormatException e) {
      e.printStackTrace();
    }
  }
}

  PropertyUtils.copyProperties(imageForm,
  imageVO);

  request.setAttribute("isEdit", isEdit);

  return mapping.findForward(forward);
}
}
```

AddImageSetAction.java

```
package com.virholex.images;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import
com.virholex.experiments.ViewExperimentDetails
VO;
import
com.virholex.experiments.VirhoExperimentsWorke
r;

public class AddImageSetAction extends Action
{
```

```
@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageSetDetailsForm imageSetForm =
  (ViewImageSetDetailsForm) form;
  ViewImageSetDetailsVO imageSetVO = new
  ViewImageSetDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewExperimentDetailsVO>
  experimentsList = new
  ArrayList<ViewExperimentDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.addImageSet";

  PropertyUtils.copyProperties(imageSetVO,
  imageSetForm);

  if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("Submit"))
  {

    errors = imageSetForm.validate(mapping,
    request);

    if(errors.isEmpty()) {

      imageSetVO.setAddedBy(username);
      imageWorker.insertImageSet(imageSetVO);

      forward =
      "virholex.images.viewImageSetDetais";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("projectSet",
      imageSetVO.getProjectSetNameDB());
      return redirect;

    } else {
      experimentsList =
      experimentWorker.getExperiments();
      saveErrors(request, errors);
    }

  } else if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("Cancel"))
  {
    forward = "virholex.images.viewImageSets";
  } else {
    experimentsList =
    experimentWorker.getExperiments();
  }

  PropertyUtils.copyProperties(imageSetForm,
  imageSetVO);

  request.setAttribute("experiments",
  experimentsList);

  return mapping.findForward(forward);
}
}
```

AddSavedViewAction.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class AddSavedViewAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewSavedViewDetailsForm savedViewForm =
  (ViewSavedViewDetailsForm) form;
  ViewSavedViewDetailsVO savedViewVO = new
  ViewSavedViewDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();
  ArrayList<ViewSavedViewDetailsVO>
  savedViewsList = new
  ArrayList<ViewSavedViewDetailsVO>();
  ArrayList<String> includedImages = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.addSavedView";
  String projectSet =
  (String)session.getAttribute("projectSet");
  String savedView = "";
  boolean disableMerge = false;
  int viewId = 0;

  PropertyUtils.copyProperties(savedViewVO,
  savedViewForm);

  // Instantiate the value for the view name
  option
  if(Utilities.isEmpty(savedViewForm.getOption
  View())) {
    savedViewForm.setOptionView("New View");
  }

  if(request.getParameterValues("imagesList")!
  =null) {
    String[] images =
    request.getParameterValues("imagesList");
    String checkbox = "";
    for(int i=0; i<images.length; i++) {
      checkbox = getImage(request, images[i]);
      if(checkbox!=null &&
      checkbox.equals(VirholexConstants.TRUE))
      {
```

```
      includedImages.add(images[i]);
    }
  }
} else
if(request.getParameterValues("myImages")!=n
ull) {
  String[] images =
  request.getParameterValues("myImages");
  if(savedViewForm.getCommand()!=null &&
  savedViewForm.getCommand().equals("Delete")
  ) {
    for(int i = 0; i < images.length; i++) {
      if(getImage(request, images[i])==null)
      {
        includedImages.add(images[i]);
      }
    }
  } else {
    for(int i=0; i<images.length; i++) {
      includedImages.add(images[i]);
    }
  }
}

savedViewVO.setImagesDB(includedImages);

if(savedViewForm.getCommand()!=null &&
savedViewForm.getCommand().equals("Submit"))
{
  errors = savedViewForm.validate(mapping,
  request);

  if(errors.isEmpty()) {

    if(savedViewForm.getOptionView().equals("
    New View")) {
      savedViewVO.setUsername(username);
      imageWorker.insertView(savedViewVO);
    } else
    if(savedViewForm.getOptionView().equals("
    Old View")) {
      try {
        if(request.getParameter("view"+savedVi
        ewVO.getViewNameDB())!=null) {
          viewId =
          Integer.parseInt(request.getParamete
          r("view"+savedViewVO.getViewNameDB()
          ));
          savedViewVO.setViewName(savedViewVO.
          getViewNameDB());
          savedViewVO.setViewId(viewId);
          imageWorker.insertToExistingView(sav
          edViewVO);
        }
      } catch(NumberFormatException e) {
        e.printStackTrace();
      }
    }

    savedView = savedViewVO.getViewNameDB();

    forward =
    "virholex.images.savedViewDetails";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("savedView",
    savedView);
    return redirect;

  } else {
    saveErrors(request, errors);
  }
} else if(savedViewForm.getCommand()!=null
&&
  savedViewForm.getCommand().equals("Cancel"))
  {
    forward = "virholex.images.viewImageSet";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("projectSet",
    projectSet);
    return redirect;
  }

  includedImages = new ArrayList<String>();

  for(int i=0;
  i<savedViewVO.getImagesDB().size(); i++) {
    String image =
    savedViewVO.getImagesDB().get(i);
    image =
    Utilities.removeFileExtension(image,
    "."+Utilities.getFileExtension(image));
    includedImages.add(image);
  }

  savedViewsList =
  imageWorker.getSavedViews(username);

  if(savedViewsList.isEmpty()) {
    disableMerge = true;
  }

  savedViewVO.setImages(includedImages);

  PropertyUtils.copyProperties(savedViewForm,
  savedViewVO);

  request.setAttribute("views",
  savedViewsList);
  request.setAttribute("disabled",
  disableMerge);

  return mapping.findForward(forward);
}

public String getImage(HttpServletRequest
request, String image) {
  return request.getParameter("img-" + image);
}

}
```

EditImageAction.java

```
package com.virholex.images;

import javax.servlet.http.HttpServletRequest;

public class EditImageAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageDetailsForm imageForm =
  (ViewImageDetailsForm) form;
  ViewImageDetailsVO imageVO = new
  ViewImageDetailsVO();
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  VirhoReferencesWorker referencesWorker = new
  VirhoReferencesWorker();
  ActionErrors errors = new ActionErrors();
```

```
  HttpSession session =
  request.getSession(true);

  String imageDB =
  (String)session.getAttribute("image");
  String forward =
  "virholex.images.editImage";

  PropertyUtils.copyProperties(imageVO,
  imageForm);

  if(request.getParameter("image")!=null)
    imageDB = request.getParameter("image");

  imageVO.setImagePathDB(imageDB);

  if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Submit")) {

    errors = imageForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      imageWorker.updateImage(imageVO);
    } else {
      saveErrors(request, errors);
    }

    forward =
    "virholex.images.viewImageDetails";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("image", imageDB);
    return redirect;

  } else if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Cancel")) {
    forward =
    "virholex.images.viewImageDetails";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("image", imageDB);
    return redirect;
  } else {
    imageVO =
    (ViewImageDetailsVO)imageWorker.readImageMe
    tadata(imageVO);
    imageVO.setImagePath(Utilities.removeFileEx
    tension(imageVO.getImagePathDB(),
    "."+Utilities.getFileExtension(imageVO.getI
    magePathDB())));
    // Get reference value
    if(!Utilities.isEmpty(String.valueOf(imageV
    O.getReferenceId()))) {
      referenceVO.setReferenceId(imageVO.getRef
      erenceId());
      referenceVO =
      (ViewReferenceDetailsVO)referencesWorker.
      readReference(referenceVO);
      imageVO.setReference(referenceVO.getTitle
      ());
      request.setAttribute("referenceDetails",
      referenceVO);
    }
  }

  PropertyUtils.copyProperties(imageForm,
  imageVO);

  return mapping.findForward(forward);
}
}
```

EditImageSetAction.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import
com.virholex.experiments.ViewExperimentDetails
VO;
import
com.virholex.experiments.VirhoExperimentsWorke
r;

public class EditImageSetAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageSetDetailsForm imageSetForm =
  (ViewImageSetDetailsForm) form;
  ViewImageSetDetailsVO imageSetVO = new
  ViewImageSetDetailsVO();
  ViewExperimentDetailsVO experimentVO = new
  ViewExperimentDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  VirhoExperimentsWorker experimentWorker =
  new VirhoExperimentsWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewExperimentDetailsVO>
  experimentsList = new
  ArrayList<ViewExperimentDetailsVO>();
  ArrayList<ViewExperimentDetailsVO>
  oldExperimentsList = new
  ArrayList<ViewExperimentDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.editImageSet";
  String projectSet =
  (String)session.getAttribute("projectSet");

  PropertyUtils.copyProperties(imageSetVO,
  imageSetForm);

  if(request.getParameter("projectSet")!=null)
  {
    projectSet =
    request.getParameter("projectSet");
    session.setAttribute("projectSet",
    projectSet);
  }

  imageSetVO.setProjectSetNameDB(projectSet);
```

```
if(imageSetForm.getCommand()!=null &&
imageSetForm.getCommand().equals("Submit"))
{
  if(!imageSetVO.getProjectSetNameDB().equals
  (imageSetVO.getProjectSetName())) {
    projectSet =
    imageSetVO.getProjectSetName();
    errors = imageSetForm.validate(mapping,
    request);
  }

  if(errors.isEmpty()) {

    String[] myExperiments =
    request.getParameterValues("myExperiments
    ");
    // Values contained in experimentList are
    experiments the user wants to be removed
    if(myExperiments!=null) {
      for(int i=0; i<myExperiments.length;
      i++) {
        try {
          experimentVO.setExperimentId(Integer
          .parseInt(myExperiments[i]));
          experimentsList.add(experimentVO);
          experimentVO = new
          ViewExperimentDetailsVO();
        } catch(NumberFormatException e) {
          e.printStackTrace();
        }
      }
    }

    imageSetVO.setExperiments(experimentsList
    );
    imageSetVO.setAddedBy(username);
    imageWorker.updateImageSet(imageSetVO);

    forward =
    "virholex.images.viewImageSetDetais";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("projectSet",
    projectSet);
    return redirect;

  } else {
    saveErrors(request, errors);
    experimentsList =
    experimentWorker.getExperiments();
    oldExperimentsList =
    experimentWorker.getExperiments(imageSetV
    O.getProjectSetId());
    experimentsList =
    getExperiments(experimentsList,
    oldExperimentsList);
    imageSetVO.setExperiments(oldExperimentsL
    ist);

  }

} else if(imageSetForm.getCommand()!=null &&
imageSetForm.getCommand().equals("Cancel"))
{
  forward =
  "virholex.images.viewImageSetDetais";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("projectSet",
  projectSet);
  return redirect;
} else {
  imageSetVO.setProjectSetName(projectSet);
    imageSetVO =
    (ViewImageSetDetailsVO)imageWorker.readImag
    eSet(imageSetVO);
    experimentsList =
    experimentWorker.getExperiments();
    oldExperimentsList =
    experimentWorker.getExperiments(imageSetVO.
    getProjectSetId());
    experimentsList =
    getExperiments(experimentsList,
    oldExperimentsList);

    imageSetVO.setExperiments(oldExperimentsLis
    t);
  }

  PropertyUtils.copyProperties(imageSetForm,
  imageSetVO);

  request.setAttribute("experiments",
  experimentsList);

  return mapping.findForward(forward);
}

public ArrayList<ViewExperimentDetailsVO>
getExperiments(ArrayList<ViewExperimentDetails
VO> experimentsList,
ArrayList<ViewExperimentDetailsVO>
oldExperimentsList) {

  for(int i=0; i<experimentsList.size(); i++)
  {
    for(int j=0; j<oldExperimentsList.size();
    j++) {
      if(((ViewExperimentDetailsVO)experimentsL
      ist.get(i)).getExperimentName().equals(((
      ViewExperimentDetailsVO)oldExperimentsLis
      t.get(j)).getExperimentName()) &&
      ((ViewExperimentDetailsVO)experimentsList
      .get(i)).getProjectId()==((ViewExperiment
      DetailsVO)oldExperimentsList.get(j)).getP
      rojectId()) {
        experimentsList.remove(i);
        i--;
        break;
      }
    }
  }

  return experimentsList;
}

}
```

EditSavedViewAction.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;
```

```
import com.virholex.general.Utilities;

public class EditSavedViewAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewSavedViewDetailsForm savedViewForm =
  (ViewSavedViewDetailsForm) form;
  ViewSavedViewDetailsVO savedViewVO = new
  ViewSavedViewDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();
  ArrayList<ViewSavedViewDetailsVO>
  savedViewsList = new
  ArrayList<ViewSavedViewDetailsVO>();
  ArrayList<String> imagesList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String savedView =
  (String)session.getAttribute("savedView");
  String forward =
  "virholex.images.editSavedView";
  boolean disableMerge = false;
  int viewId = 0;

  PropertyUtils.copyProperties(savedViewVO,
  savedViewForm);

  if(request.getParameter("savedView")!=null)
  {
    savedView =
    request.getParameter("savedView");
    session.setAttribute("savedView",
    savedView);
  }
  savedViewVO.setOldViewName(savedView);

  if(savedViewForm.getCommand()!=null &&
  savedViewForm.getCommand().equals("Submit"))
  {
    errors = savedViewForm.validate(mapping,
    request);

    if(errors.isEmpty()) {

      savedViewVO.setUsername(username);

      if(savedViewForm.getOptionView().equals("
      New View") &&
      !savedViewVO.getViewName().equals(savedVi
      ewVO.getOldViewName())) {
        imageWorker.updateView(savedViewVO);
        savedView =
        savedViewVO.getViewNameDB();
      } else
      if(savedViewForm.getOptionView().equals("
      Old View")) {
        try {
          if(request.getParameter("view"+savedVi
          ewVO.getViewNameDB())!=null) {
            viewId =
            Integer.parseInt(request.getParamete
            r("view"+savedViewVO.getViewNameDB()
            ));
            savedViewVO.setViewId(viewId);
            imageWorker.mergeToExistingView(save
            dViewVO);
            savedView =
            savedViewVO.getViewNameDB();
          }
        } catch(NumberFormatException e) {
          e.printStackTrace();
        }
      }

      forward =
      "virholex.images.savedViewDetails";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("savedView",
      savedView);
      return redirect;

    } else {
      saveErrors(request, errors);
    }
  } else if(savedViewForm.getCommand()!=null
  &&
  savedViewForm.getCommand().equals("Delete"))
  {
    if(request.getParameterValues("myImages")!=
    null) {
      String[] images =
      request.getParameterValues("myImages");

      for(int i = 0; i < images.length; i++) {
        if(getImage(request, images[i])!=null)
        {
          imagesList.add(images[i]);
        }
      }

      if(!imagesList.isEmpty()) {
        imageWorker.deleteIncludedImages(savedV
        iew, imagesList);
      }
    }

  } else if(savedViewForm.getCommand()!=null
  &&
  savedViewForm.getCommand().equals("Cancel"))
  {
    forward =
    "virholex.images.savedViewDetails";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("savedView",
    savedView);
    return redirect;
  } else {
    // Instantiate
    if(Utilities.isEmpty(savedViewForm.getOptio
    nView()))
      savedViewForm.setOptionView("New View");
    if(Utilities.isEmpty(savedViewVO.getViewNam
    e()))
      savedViewVO.setViewName(savedView);
  }

  imagesList =
  imageWorker.getIncludedImages(username,
  savedView);
  savedViewVO.setImagesDB(imagesList);
  imagesList = new ArrayList<String>();

  for(int i=0;
  i<savedViewVO.getImagesDB().size(); i++) {
    String image =
    savedViewVO.getImagesDB().get(i);
```

```
    image =
    Utilities.removeFileExtension(image,
    "."+Utilities.getFileExtension(image));
    imagesList.add(image);
  }

  savedViewVO.setImages(imagesList);
  savedViewsList =
  imageWorker.getSavedViews(username);

  if(savedViewsList.isEmpty()) {
    disableMerge = true;
  }

  PropertyUtils.copyProperties(savedViewForm,
  savedViewVO);

  request.setAttribute("views",
  savedViewsList);
  request.setAttribute("disabled",
  disableMerge);

  return mapping.findForward(forward);
}

public String getImage(HttpServletRequest
request, String image) {
  return request.getParameter("img-" + image);
}

}
```

ImageDAO.java

```
package com.virholex.images;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.Utilities;

public class ImageDAO {

  public static ViewImageDetailsVO
  readImageMetadata(ViewImageDetailsVO vo)
  throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {

    String sql = "SELECT category,
    accessibility, description, date_uploaded,
    uploaded_by, start_date, end_date,
    cell_type, virus_type, mult_of_infection,
    medium, temperature, co2_level, owner,
    container, phenol_red, magnification,
    objective, illumination_type,
    illumination_level, filter, version,
    reference_id, image_set_id, image_id,
    image_path FROM image WHERE image_path=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getImagePathDB());
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setCategory(rs.getString(1));
      vo.setAccessibility(rs.getString(2));
      vo.setDescription(rs.getString(3));
      vo.setDateUploaded(rs.getString(4));
      vo.setUploadedBy(rs.getString(5));
      vo.setStartDate(rs.getString(6));
      vo.setEndDate(rs.getString(7));
      vo.setCellType(rs.getString(8));
      vo.setVirusType(rs.getString(9));
      vo.setMultOfInfection(rs.getString(10));
      vo.setMedium(rs.getString(11));
      vo.setTemperature(rs.getString(12));
      vo.setCO2Level(rs.getString(13));
      vo.setOwner(rs.getString(14));
      vo.setContainer(rs.getString(15));
      vo.setPhenolRed(rs.getString(16));
      vo.setMagnification(rs.getString(17));
      vo.setObjective(rs.getString(18));
      vo.setIlluminationType(rs.getString(19));
      vo.setIlluminationLevel(rs.getString(20))
      ;
      vo.setFilter(rs.getString(21));
      vo.setVersion(rs.getString(22));
      vo.setReferenceId(rs.getInt(23));
      vo.setImageSetId(rs.getInt(24));
      vo.setImageId(rs.getInt(25));
      vo.setImagePathDB(rs.getString(26));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return vo;
  }

  public static ArrayList<ViewImageDetailsVO>
  getRecentImages() throws SQLException,
  NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewImageDetailsVO vo = new
  ViewImageDetailsVO();
  ArrayList<ViewImageDetailsVO>
  recentImagesList = new
  ArrayList<ViewImageDetailsVO>();

  try {

    String sql = "SELECT image_id, image_path,
    accessibility, project_set_name FROM image,
    project_set WHERE
    image_set_id=project_set_id ORDER BY
    date_uploaded DESC LIMIT 10";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setImageId(rs.getInt(1));
      vo.setImagePathDB(rs.getString(2));
      vo.setAccessibility(rs.getString(3));
      vo.setImageSet(rs.getString(4));
      vo.setImagePath(Utilities.removeFileExten
      sion(vo.getImagePathDB(),
      "."+Utilities.getFileExtension(vo.getImag
      ePathDB())));
      recentImagesList.add(vo);
      vo = new ViewImageDetailsVO();
    }
```

```
} finally {
  if(rs != null) rs.close();
  if(ps != null) ps.close();
  if(conn != null) conn.close();
}

return recentImagesList;
}

public static void
insertImages(ArrayList<ViewImageDetailsVO>
list) throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {
    conn = DBConnect.getConnection();
    String sql = "";

    for(int i=0; i<list.size(); i++) {

      sql = "SELECT project_set_id FROM
      project_set WHERE project_set_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      ((ViewImageDetailsVO)list.get(i)).getIm
      ageSet());
      rs = ps.executeQuery();

      if(rs.next()) {
        ((ViewImageDetailsVO)list.get(i)).setI
        mageSetId(rs.getInt(1));
      }

      rs.close();
      ps.close();

      sql = "INSERT INTO image (image_path,
      image_set_id, uploaded_by,
      accessibility, category, description,
      start_date, end_date, cell_type,
      virus_type, mult_of_infection, medium,
      temperature, co2_level, owner,
      container, phenol_red, magnification,
      objective, illumination_type,
      illumination_level, filter, version,
      reference_id, date_uploaded)
      VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
      ?,?,?,?,?,?,?,?,NOW())";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      ((ViewImageDetailsVO)list.get(i)).getIm
      agePathDB());
      ps.setInt(2,
      ((ViewImageDetailsVO)list.get(i)).getIm
      ageSetId());
      ps.setString(3,
      ((ViewImageDetailsVO)list.get(i)).getUp
      loadedBy());
      ps.setString(4,
      ((ViewImageDetailsVO)list.get(i)).getAc
      cessibility());
      ps.setString(5,
      ((ViewImageDetailsVO)list.get(i)).getCa
      tegory());
      ps.setString(6,
      ((ViewImageDetailsVO)list.get(i)).getDe
      scription());
      ps.setString(7,
      ((ViewImageDetailsVO)list.get(i)).getSt
      artDate());
      ps.setString(8,
      ((ViewImageDetailsVO)list.get(i)).getEn
      dDate());
      ps.setString(9,
      ((ViewImageDetailsVO)list.get(i)).getCe
      llType());
      ps.setString(10,
      ((ViewImageDetailsVO)list.get(i)).getVi
      rusType());
      ps.setString(11,
      ((ViewImageDetailsVO)list.get(i)).getMu
      ltOfInfection());
      ps.setString(12,
      ((ViewImageDetailsVO)list.get(i)).getMe
      dium());
      ps.setString(13,
      ((ViewImageDetailsVO)list.get(i)).getTe
      mperature());
      ps.setString(14,
      ((ViewImageDetailsVO)list.get(i)).getCO
      2Level());
      ps.setString(15,
      ((ViewImageDetailsVO)list.get(i)).getOw
      ner());
      ps.setString(16,
      ((ViewImageDetailsVO)list.get(i)).getCo
      ntainer());
      ps.setString(17,
      ((ViewImageDetailsVO)list.get(i)).getPh
      enolRed());
      ps.setString(18,
      ((ViewImageDetailsVO)list.get(i)).getMa
      gnification());
      ps.setString(19,
      ((ViewImageDetailsVO)list.get(i)).getOb
      jective());
      ps.setString(20,
      ((ViewImageDetailsVO)list.get(i)).getIl
      luminationType());
      ps.setString(21,
      ((ViewImageDetailsVO)list.get(i)).getIl
      luminationLevel());
      ps.setString(22,
      ((ViewImageDetailsVO)list.get(i)).getFi
      lter());
      ps.setString(23,
      ((ViewImageDetailsVO)list.get(i)).getVe
      rsion());
      ps.setInt(24,
      ((ViewImageDetailsVO)list.get(i)).getRe
      ferenceId());
      ps.executeUpdate();
      ps.close();

    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static void
updateImage(ViewImageDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    String sql = "UPDATE image SET
    accessibility=?, category=?,
    description=?, start_date=?, end_date=?,
    cell_type=?, virus_type=?,
    mult_of_infection=?, medium=?,
    temperature=?, co2_level=?, owner=?,
```

```
    container=?, phenol_red=?,
    magnification=?, objective=?,
    illumination_type=?,
    illumination_level=?, filter=?,
    version=?, reference_id=? where
    image_path=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getAccessibility());
    ps.setString(2, vo.getCategory());
    ps.setString(3, vo.getDescription());
    ps.setString(4, vo.getStartDate());
    ps.setString(5, vo.getEndDate());
    ps.setString(6, vo.getCellType());
    ps.setString(7, vo.getVirusType());
    ps.setString(8, vo.getMultOfInfection());
    ps.setString(9, vo.getMedium());
    ps.setString(10, vo.getTemperature());
    ps.setString(11, vo.getCO2Level());
    ps.setString(12, vo.getOwner());
    ps.setString(13, vo.getContainer());
    ps.setString(14, vo.getPhenolRed());
    ps.setString(15, vo.getMagnification());
    ps.setString(16, vo.getObjective());
    ps.setString(17,
    vo.getIlluminationType());
    ps.setString(18,
    vo.getIlluminationLevel());
    ps.setString(19, vo.getFilter());
    ps.setString(20, vo.getVersion());
    ps.setInt(21, vo.getReferenceId());
    ps.setString(22, vo.getImagePathDB());
    ps.executeUpdate();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

}

public static void deleteImage(String image)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE saved_view_image
    FROM saved_view_image, image WHERE
    saved_view_image.image_id=image.image_id
    AND image_path=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, image);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM image WHERE
    image_path=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, image);
    ps.executeUpdate();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
deleteImages(ArrayList<String> images) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    conn = DBConnect.getConnection();
    String sql = "";

    for(int i=0; i<images.size(); i++) {
      sql = "DELETE saved_view_image FROM
      saved_view_image, image WHERE
      saved_view_image.image_id=image.image_id
      AND image_path=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, images.get(i));
      ps.executeUpdate();
      ps.close();

      sql = "DELETE FROM image WHERE
      image_path=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, images.get(i));
      ps.executeUpdate();
      ps.close();
      }

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

  public static void
  deleteIncludedImages(String title,
  ArrayList<String> images) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();
      String sql = "";

      for(int i=0; i<images.size(); i++) {
        sql = "DELETE saved_view_image FROM
        image, saved_view, saved_view_image
        WHERE image.image_path=? AND
        saved_view.view_name=? AND
        image.image_id=saved_view_image.image_i
        d AND
        saved_view.view_id=saved_view_image.vie
        w_id";
        ps = conn.prepareStatement(sql);
        ps.setString(1, images.get(i));
        ps.setString(2, title);
        ps.executeUpdate();
        ps.close();
      }

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

ImageSetsDAO.java

```
package com.virholex.images;

import java.sql.Connection;

public class ImageSetsDAO {
```

```
public static
ArrayList<ViewImageSetDetailsVO>
getImageSets(ViewImageSetDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  Statement s = null;
  ResultSet rs = null;
  ArrayList<ViewImageSetDetailsVO>
  imageSetsList = new
  ArrayList<ViewImageSetDetailsVO>();

  try {
    String sql = "SELECT project_set_name,
    project_set_desc, project_set_id FROM
    project_set ORDER BY project_set_name";
    conn = DBConnect.getConnection();
    s = conn.createStatement();
    s.executeQuery(sql);
    rs = s.getResultSet();

    while(rs.next()) {
      vo.setProjectSetName(rs.getString(1));
      vo.setDescription(rs.getString(2));
      vo.setProjectSetId(rs.getInt(3));

      imageSetsList.add(vo);
      vo = new ViewImageSetDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(s != null) s.close();
    if(conn != null) conn.close();
  }

  return imageSetsList;
}

public static
ArrayList<ViewImageSetDetailsVO>
getImageSets(int experimentId) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewImageSetDetailsVO vo = new
  ViewImageSetDetailsVO();
  ArrayList<ViewImageSetDetailsVO>
  imageSetsList = new
  ArrayList<ViewImageSetDetailsVO>();

  try {
  String sql = "SELECT project_set_name,
  project_set_id, image_sets.experiment_id
  FROM image_sets, project_set WHERE " +
         "image_sets.experiment_id=? AND
  project_set_id=project_id";
  conn = DBConnect.getConnection();
  ps = conn.prepareStatement(sql);
  ps.setInt(1, experimentId);
  rs = ps.executeQuery();

  while(rs.next()) {
  vo.setProjectSetName(rs.getString(1));
  vo.setProjectSetId(rs.getInt(2));
  vo.setExperimentId(rs.getInt(3));

  imageSetsList.add(vo);
  vo = new ViewImageSetDetailsVO();
  }

  } finally {
  if(rs != null) rs.close();
  if(ps != null) ps.close();
  if(conn != null) conn.close();
  }

  return imageSetsList;
  }

  public static String
  getProjectSetRole(String username, String
  project) throws SQLException,
  NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String role = "";

  try {
    String sql = "SELECT role_name FROM
    user_prev WHERE username=? AND
    involvement=? AND (role_name LIKE
    '%Image%' OR role_name='Restricted
    User')";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, project);
    rs = ps.executeQuery();

    if(rs.next()) {
      role = rs.getString(1);
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return role;
}

public static boolean
isImageSetCoordinator(String username)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean isLead = false;

  try {
    String sql = "SELECT role_name FROM
    user_prev WHERE username=? AND
    involvement='DEFAULT_PROJECT' AND
    role_name='Image Set Coordinator'";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    rs = ps.executeQuery();

    if(rs.next()) {
      isLead = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isLead;
}
```

```
public static ViewImageSetDetailsVO
readImageSet(ViewImageSetDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {
    String sql = "SELECT project_set_name,
    project_set_desc, date_added,
    date_modified, added_by, project_set_id
    FROM project_set WHERE
    project_set_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectSetName());
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setProjectSetName(rs.getString(1));
      vo.setDescription(rs.getString(2));
      vo.setDateAdded(rs.getString(3));
      vo.setDateModified(rs.getString(4));
      vo.setAddedBy(rs.getString(5));
      vo.setProjectSetId(rs.getInt(6));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return vo;
}

public static ArrayList<String>
getPublicImages(String projectSet) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> imagePathList = new
  ArrayList<String>();

  try {
    String sql = "SELECT image_path FROM
    image, project_set WHERE
    project_set.project_set_name=? AND
    image.image_set_id =
    project_set.project_set_id AND
    image.accessibility='public' ORDER BY
    date_uploaded DESC";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, projectSet);
    rs = ps.executeQuery();

    while(rs.next()) {
      imagePathList.add(rs.getString(1));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return imagePathList;
}

public static ArrayList<String>
getPrivateImages(String projectSet) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> imagePathList = new
  ArrayList<String>();

  try {
    String sql = "SELECT image_path FROM
    image, project_set WHERE
    project_set.project_set_name=? AND
    image.image_set_id=project_set.project_se
    t_id ORDER BY date_uploaded DESC";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, projectSet);
    rs = ps.executeQuery();

    while(rs.next()) {
      imagePathList.add(rs.getString(1));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return imagePathList;
}

public static void
insertImageSet(ViewImageSetDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> titleDBList = new
  ArrayList<String>();

  try {
    conn = DBConnect.getConnection();

    String sql = "SELECT project_set_name
    FROM project_set WHERE project_set_name=?
    OR project_set_name LIKE ? ORDER BY
    project_set_name";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectSetName());
    ps.setString(2, vo.getProjectSetName()+"
    (%)");
    rs = ps.executeQuery();

    while(rs.next()) {
    titleDBList.add(rs.getString(1));
    }

    rs.close();
    ps.close();
    // generate new title for duplicates
    vo.setProjectSetNameDB(Utilities.generate
    Title(titleDBList,
    vo.getProjectSetName()));

    sql = "INSERT INTO project_set
    (project_set_name, project_set_desc,
    date_added, date_modified, added_by)
    VALUES(?,?,NOW(),NOW(),?)";
    ps = conn.prepareStatement(sql);
    ps.setString(1,
    vo.getProjectSetNameDB());
    ps.setString(2, vo.getDescription());
    ps.setString(3, vo.getAddedBy());
    ps.executeUpdate();
    ps.close();
```

```
    sql = "SELECT LAST_INSERT_ID() AS id";
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();
    rs.next();
    vo.setProjectSetId(rs.getInt(1));
    rs.close();
    ps.close();

    if(vo.getExperimentId()!=0) {
      sql = "INSERT INTO image_sets
      (project_id, experiment_id,
      imageset_name) VALUES(?,?,?)";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getProjectSetId());
      ps.setInt(2, vo.getExperimentId());
      ps.setString(3,
      vo.getProjectSetNameDB());
      ps.executeUpdate();
      ps.close();
    }

    sql = "INSERT INTO user_prev (username,
    role_name, involvement, date_added,
    date_modified, kind)
    VALUES(?,?,?,NOW(),NOW(),?)";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getAddedBy());
    ps.setString(2,
    VirholexConstants.IMAGE_SET_COORDINATOR_D
    ESC);
    ps.setString(3,
    vo.getProjectSetNameDB());
    ps.setString(4,
    VirholexConstants.PROJECTSET);
    ps.executeUpdate();

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
updateImageSet(ViewImageSetDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> titleDBList = new
  ArrayList<String>();

  try {
    conn = DBConnect.getConnection();
    String sql = "";

    if(!vo.getProjectSetName().equals(vo.getP
    rojectSetNameDB())) {

      sql = "SELECT project_set_name FROM
      project_set WHERE project_set_name=? OR
      project_set_name LIKE ? ORDER BY
      project_set_name";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      vo.getProjectSetName());
      ps.setString(2,
      vo.getProjectSetName()+" (%)");
      rs = ps.executeQuery();

      while(rs.next()) {
        titleDBList.add(rs.getString(1));
      }
      rs.close();
      ps.close();
      // generate new title for duplicates
      vo.setProjectSetName(Utilities.generate
      Title(titleDBList,
      vo.getProjectSetName()));
    }

    sql = "UPDATE project_set SET
    project_set_name=?, project_set_desc=?,
    date_modified=NOW() WHERE
    project_set_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectSetName());
    ps.setString(2, vo.getDescription());
    ps.setString(3,
    vo.getProjectSetNameDB());
    ps.executeUpdate();
    ps.close();

    sql = "SELECT project_set_id FROM
    project_set WHERE project_set_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectSetName());
    rs = ps.executeQuery();
    rs.next();
    vo.setProjectSetId(rs.getInt(1));
    rs.close();
    ps.close();

    if(!vo.getExperiments().isEmpty()) {
      sql = "DELETE FROM image_sets WHERE
      project_id=? AND experiment_id=?";

      for(int i=0;
      i<vo.getExperiments().size(); i++) {
        ps = conn.prepareStatement(sql);
        ps.setInt(1, vo.getProjectSetId());
        ps.setInt(2,
        ((ViewExperimentDetailsVO)vo.getExperi
        ments().get(i)).getExperimentId());
        ps.executeUpdate();
        ps.close();
      }
    }

    if(vo.getExperimentId() > 0) {
      sql = "INSERT INTO image_sets
      (project_id, experiment_id,
      imageset_name) VALUES(?,?,?)";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getProjectSetId());
      ps.setInt(2, vo.getExperimentId());
      ps.setString(3,
      vo.getProjectSetName());
      ps.executeUpdate();
      ps.close();
    }

    sql = "UPDATE user_prev SET
    involvement=?, date_modified=NOW() WHERE
    involvement=? AND role_name LIKE
    '%Image%'";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getProjectSetName());
    ps.setString(2,
    vo.getProjectSetNameDB());
    ps.executeUpdate();

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}
```

```
  public static void deleteImageSet(String
  projectSetName) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    int projectSetId = 0;

    try {
      conn = DBConnect.getConnection();

      String sql = "SELECT project_set_id FROM
      project_set WHERE project_set_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, projectSetName);
      rs = ps.executeQuery();

      if(rs.next()) {
        projectSetId = rs.getInt(1);
      }

      rs.close();
      ps.close();

      sql = "DELETE FROM image_sets WHERE
      imageset_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, projectSetName);
      ps.executeUpdate();
      ps.close();

      sql = "DELETE FROM user_prev WHERE
      involvement=? AND kind='projectset'";
      ps = conn.prepareStatement(sql);
      ps.setString(1, projectSetName);
      ps.executeUpdate();
      ps.close();

      sql = "DELETE FROM project_set WHERE
      project_set_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, projectSetName);
      ps.executeUpdate();
      ps.close();

      sql = "DELETE saved_view_image FROM
      saved_view_image, image WHERE
      saved_view_image.image_id=image.image_id
      AND image.image_set_id=?";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, projectSetId);
      ps.executeUpdate();
      ps.close();

      sql = "DELETE FROM image WHERE
      image_set_id=?";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, projectSetId);
      ps.executeUpdate();
      ps.close();

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

SavedViewsDAO.java

```
package com.virholex.images;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.Utilities;

public class SavedViewsDAO {

  public static
  ArrayList<ViewSavedViewDetailsVO>
  getSavedViews(String username) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ViewSavedViewDetailsVO vo = new
    ViewSavedViewDetailsVO();
    ArrayList<ViewSavedViewDetailsVO>
    savedViewsList = new
    ArrayList<ViewSavedViewDetailsVO>();

    try {

      String sql = "SELECT view_id, view_name,
      date_added, date_modified FROM saved_view
      WHERE username=? ORDER BY view_name";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setViewId(rs.getInt(1));
        vo.setViewName(rs.getString(2));
        vo.setDateAdded(Utilities.dateFormat(rs
        .getString(3)));
        vo.setDateModified(Utilities.dateFormat
        (rs.getString(4)));

        savedViewsList.add(vo);
        vo = new ViewSavedViewDetailsVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return savedViewsList;
  }

  public static ArrayList<String>
  getIncludedImages(String username, String
  title) throws SQLException, NamingException
  {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<String> imagesList = new
    ArrayList<String>();

    try {

      String sql = "SELECT image.image_path
      FROM image, saved_view_image, saved_view
      WHERE saved_view.username=? AND
      saved_view.view_name=? AND
      saved_view.view_id=saved_view_image.view_
```

```
    id AND
    saved_view_image.image_id=image.image_id"
    ;
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, title);
    rs = ps.executeQuery();

    while(rs.next()) {
      imagesList.add(rs.getString(1));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return imagesList;

}

public static void
insertView(ViewSavedViewDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> viewNameDBList = new
  ArrayList<String>();

  try {

    conn = DBConnect.getConnection();
    int imageId = 0;

    String sql = "SELECT view_name FROM
    saved_view WHERE username=? AND
    (view_name=? OR view_name LIKE ?)";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    ps.setString(2, vo.getViewName());
    ps.setString(3, vo.getViewName()+" (%)");
    rs = ps.executeQuery();

    while(rs.next()) {
      viewNameDBList.add(rs.getString(1));
    }

    ps.close();
    // generate new title for duplicates
    vo.setViewNameDB(Utilities.generateTitle(
    viewNameDBList, vo.getViewName()));

    sql = "INSERT INTO saved_view (username,
    view_name, date_added, date_modified)
    VALUES(?,?,NOW(),NOW())";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getUsername());
    ps.setString(2, vo.getViewNameDB());
    ps.executeUpdate();
    ps.close();

    sql = "SELECT LAST_INSERT_ID() AS id";
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();
    rs.next();
    vo.setViewId(rs.getInt("id"));
    rs.close();
    ps.close();

    for(int i=0; i<vo.getImagesDB().size();
    i++) {
      sql = "SELECT image_id FROM image WHERE
      image_path=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      vo.getImagesDB().get(i));
      rs = ps.executeQuery();
      rs.next();
      imageId = rs.getInt(1);
      rs.close();
      ps.close();

      sql = "INSERT INTO saved_view_image
      (view_id, image_id) VALUES(?,?)";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getViewId());
      ps.setInt(2, imageId);
      ps.executeUpdate();
      ps.close();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
insertToExistingView(ViewSavedViewDetailsVO
vo) throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {

    conn = DBConnect.getConnection();
    String sql = "";
    int imageId = 0;

    sql = "UPDATE saved_view SET
    date_modified=NOW() WHERE view_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getViewId());
    ps.executeUpdate();
    ps.close();

    for(int i=0; i<vo.getImagesDB().size();
    i++) {
      sql = "SELECT image_id FROM image WHERE
      image_path=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      vo.getImagesDB().get(i));
      rs = ps.executeQuery();
      rs.next();
      imageId = rs.getInt(1);
      rs.close();
      ps.close();

      sql = "REPLACE INTO saved_view_image
      (view_id, image_id) VALUES(?,?)";
      ps = conn.prepareStatement(sql);
      ps.setInt(1, vo.getViewId());
      ps.setInt(2, imageId);
      ps.executeUpdate();
      ps.close();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
```

```
}

public static void
updateView(ViewSavedViewDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ArrayList<String> viewNameDBList = new
  ArrayList<String>();

  try {
    conn = DBConnect.getConnection();
    String sql = "";

    if(!vo.getViewName().equals(vo.getOldView
    Name())) {
      sql = "SELECT view_name FROM saved_view
      WHERE username=? AND (view_name=? OR
      view_name LIKE ?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getUsername());
      ps.setString(2, vo.getViewName());
      ps.setString(3, vo.getViewName()+"
      (%)");
      rs = ps.executeQuery();

      while(rs.next()) {
        viewNameDBList.add(rs.getString(1));
      }

      rs.close();
      ps.close();
      // generate new title for duplicates
      vo.setViewNameDB(Utilities.generateTitl
      e(viewNameDBList, vo.getViewName()));
    }

    sql = "UPDATE saved_view SET view_name=?,
    date_modified=NOW() WHERE view_name=? AND
    username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getViewNameDB());
    ps.setString(2, vo.getOldViewName());
    ps.setString(3, vo.getUsername());
    ps.executeUpdate();

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
mergeToExistingView(ViewSavedViewDetailsVO
vo) throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    conn = DBConnect.getConnection();

    String sql = "UPDATE IGNORE saved_view,
    saved_view_image SET
    saved_view_image.view_id=?,
    date_modified=NOW() WHERE
    saved_view.view_name=? AND
    saved_view.username=? AND
    saved_view.view_id=saved_view_image.view_
    id";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getViewId());
    ps.setString(2, vo.getOldViewName());
    ps.setString(3, vo.getUsername());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE saved_view_image FROM
    saved_view_image, saved_view WHERE
    saved_view.view_id=saved_view_image.view_
    id AND saved_view.view_name=? AND
    saved_view.username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getOldViewName());
    ps.setString(2, vo.getUsername());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE from saved_view where
    view_name=? AND username=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getOldViewName());
    ps.setString(2, vo.getUsername());
    ps.executeUpdate();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void deleteView(String
username, String viewName) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  int viewId = 0;

  try {
    conn = DBConnect.getConnection();

    String sql = "SELECT view_id FROM
    saved_view WHERE username=? AND
    view_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, viewName);
    rs = ps.executeQuery();
    rs.next();
    viewId = rs.getInt(1);
    rs.close();
    ps.close();

    sql = "DELETE FROM saved_view WHERE
    view_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, viewId);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM saved_view_image WHERE
    view_id=?";
    ps = conn.prepareStatement(sql);
    ps.setInt(1, viewId);
    ps.executeUpdate();

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
 }

}
```

ViewImageDetailsAction.java

```
package com.virholex.images;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import
com.virholex.references.ViewReferenceDetailsVO
;
import
com.virholex.references.VirhoReferencesWorker;

public class ViewImageDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageDetailsForm imageForm =
  (ViewImageDetailsForm) form;
  ViewImageDetailsVO imageVO = new
  ViewImageDetailsVO();
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  VirhoReferencesWorker referencesWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> imagesList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String projectSet =
  (String)session.getAttribute("projectSet");
  String imageDB =
  (String)session.getAttribute("image");
  String forward =
  "virholex.images.viewImageDetails";
  String roleName = "";

  PropertyUtils.copyProperties(imageVO,
  imageForm);

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_IMAGES);
  if(request.getParameter("projectSet")!=null)
  {
    projectSet =
    request.getParameter("projectSet");
    session.setAttribute("projectSet",
    projectSet);
  }
  if(request.getParameter("image")!=null)
    imageDB = request.getParameter("image");

  if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Edit
  Metadata")) {
    forward = "virholex.images.editImage";
    } else if(imageForm.getCommand()!=null &&
    imageForm.getCommand().equals("Delete
    Image")) {
    // delete the image from the repository
    Utilities.deleteFile(request,
    VirholexConstants.IMAGES_REPOSITORY,
    imageDB);
    // delete the image
    imageWorker.deleteImage(imageDB);

    forward =
    "virholex.images.viewImageSetDetais";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("projectSet",
    projectSet);
    return redirect;

  } else if(imageForm.getCommand()!=null &&
  imageForm.getCommand().equals("Send
  Membership Request")) {
    session.setAttribute("project",
    projectSet);
    forward =
    "virholex.virus.requestPrivilege";
  } else {
    imageVO.setImagePathDB(imageDB);
    imageVO =
    (ViewImageDetailsVO)imageWorker.readImageMe
    tadata(imageVO);
    imageVO.setImagePath(Utilities.removeFileEx
    tension(imageVO.getImagePathDB(),
    "."+Utilities.getFileExtension(imageVO.getI
    magePathDB())));
    imageVO.setDateUploaded(Utilities.dateForma
    t(imageVO.getDateUploaded()));
    // Get reference value

    if(!Utilities.isEmpty(String.valueOf(imageV
    O.getReferenceId()))) {
      referenceVO.setReferenceId(imageVO.getRef
      erenceId());
      referenceVO =
      (ViewReferenceDetailsVO)referencesWorker.
      readReference(referenceVO);
      imageVO.setReference(referenceVO.getTitle
      ());
      request.setAttribute("referenceDetails",
      referenceVO);
    }

    roleName =
    imageWorker.getProjectRole(username,
    projectSet);

    if(roleName.equals(VirholexConstants.IMAGE_
    SET_COORDINATOR_DESC))
      imageVO.setPrivilege(VirholexConstants.IM
      AGE_SET_COORDINATOR);
    else
    if(roleName.equals(VirholexConstants.IMAGE_
    SET_CONTRIBUTOR_DESC))
```

```
    imageVO.setPrivilege(VirholexConstants.IM
    AGE_SET_CONTRIBUTOR);
  else
  if(roleName.equals(VirholexConstants.RESTRI
  CTED_USER_DESC))
    imageVO.setPrivilege(VirholexConstants.RE
    STRICTED_USER);
  else
    imageVO.setPrivilege(VirholexConstants.RE
    GISTERED_USER);

  if(imagesList.isEmpty()) {
    if(imageVO.getPrivilege() > 1)
      imagesList =
      imageWorker.getPrivateImages(projectSet
      );
    else
      imagesList =
      imageWorker.getPublicImages(projectSet)
      ;
    }

    if(imageForm.getCommand()!=null &&
    imageForm.getCommand().equals("Download
    Image")) {

      File file = null;
      InputStream in = null;
      ServletOutputStream out = null;
      String pathname = "";

      try {
        String filename =
        imageVO.getImagePathDB();

        response.setContentType("image/jpg");
        response.setHeader("Content-
        Disposition", "attachment;filename=" +
        filename);

        Utilities.copyFile(request, filename,
        VirholexConstants.IMAGES_REPOSITORY,
        VirholexConstants.IMAGES_TEMP_REPOSITO
        RY);
        pathname =
        request.getSession().getServletContext
        ().getRealPath(VirholexConstants.IMAGE
        S_TEMP_REPOSITORY) + File.separator +
        filename;
        file = new File(pathname);
        in = new FileInputStream(file);
        out = response.getOutputStream();

        int bit = in.read();
        while((bit) >= 0) {
          out.write(bit);
          bit = in.read();
        }

    } catch (FileNotFoundException e) {
      e.printStackTrace();
    } catch (IOException e) {
      e.printStackTrace();
    } catch (NullPointerException e) {
      e.printStackTrace();
      System.out.println("File " + pathname +
      " is not is the images repository.");
    } finally {
      out.flush();
      out.close();
      in.close();
      file.delete();
    }
  }
}
```

```
  PropertyUtils.copyProperties(imageForm,
  imageVO);

  session.setAttribute("image", imageDB);
  request.setAttribute("images", imagesList);

  return mapping.findForward(forward);
}
}
```

ViewImageDetailsForm.java

```
package com.virholex.images;

import javax.servlet.http.HttpServletRequest;

public class ViewImageDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  -1080831782144114658L;
  private int imageId;
  private int imageSetId;
  private FormFile file;
  private String imageSet;
  private String imagePath;
  private String imagePathDB;
  private String accessibility;
  private String category;
  private String description;
  private String cellType;
  private String virusType;
  private String multOfInfection;
  private String medium;
  private String temperature;
  private String CO2Level;
  private String owner;
  private String container;
  private String phenolRed;
  private String magnification;
  private String objective;
  private String illuminationType;
  private String illuminationLevel;
  private String filter;
  private String uploadedBy;
  private String dateUploaded;
  private String startDate;
  private String endDate;
  private String version;
  private String reference;
  private int referenceId;
  private int privilege;
  private String command;

  public void setImageId(int imageId) {
    this.imageId = imageId;
  }
  public int getImageId() {
    return imageId;
  }
  public void setImageSetId(int imageSetId) {
    this.imageSetId = imageSetId;
  }
  public int getImageSetId() {
    return imageSetId;
  }
  public void setFile(FormFile file) {
    this.file = file;
  }
  public FormFile getFile() {
    return file;
  }
  public void setImageSet(String imageSet) {
    this.imageSet = imageSet;
```

```
}
public String getImageSet() {
  return imageSet;
}
public void setImagePath(String imagePath) {
  this.imagePath = imagePath;
}
public String getImagePath() {
  return imagePath;
}
public void setImagePathDB(String
imagePathDB) {
  this.imagePathDB = imagePathDB;
}
public String getImagePathDB() {
  return imagePathDB;
}
public void setAccessibility(String
accessibility) {
  this.accessibility = accessibility;
}
public String getAccessibility() {
  return accessibility;
}
public void setCategory(String category) {
  this.category = category;
}
public String getCategory() {
  return category;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setCellType(String cellType) {
  this.cellType = cellType;
}
public String getCellType() {
  return cellType;
}
public void setVirusType(String virusType) {
  this.virusType = virusType;
}
public String getVirusType() {
  return virusType;
}
public void setMultOfInfection(String
multOfInfection) {
  this.multOfInfection = multOfInfection;
}
public String getMultOfInfection() {
  return multOfInfection;
}
public void setMedium(String medium) {
  this.medium = medium;
}
public String getMedium() {
  return medium;
}
public void setTemperature(String
temperature) {
  this.temperature = temperature;
}
public String getTemperature() {
  return temperature;
}
public void setCO2Level(String cO2Level) {
  CO2Level = cO2Level;
}
public String getCO2Level() {
  return CO2Level;
}
public void setOwner(String owner) {
  this.owner = owner;
}
public String getOwner() {
  return owner;
}
public void setContainer(String container) {
  this.container = container;
}
public String getContainer() {
  return container;
}
public void setPhenolRed(String phenolRed) {
  this.phenolRed = phenolRed;
}
public String getPhenolRed() {
  return phenolRed;
}
public void setMagnification(String
magnification) {
  this.magnification = magnification;
}
public String getMagnification() {
  return magnification;
}
public void setObjective(String objective) {
  this.objective = objective;
}
public String getObjective() {
  return objective;
}
public void setIlluminationType(String
illuminationType) {
  this.illuminationType = illuminationType;
}
public String getIlluminationType() {
  return illuminationType;
}
public void setIlluminationLevel(String
illuminationLevel) {
  this.illuminationLevel = illuminationLevel;
}
public String getIlluminationLevel() {
  return illuminationLevel;
}
public void setFilter(String filter) {
  this.filter = filter;
}
public String getFilter() {
  return filter;
}
public void setUploadedBy(String uploadedBy)
{
  this.uploadedBy = uploadedBy;
}
public String getUploadedBy() {
  return uploadedBy;
}
public void setDateUploaded(String
dateUploaded) {
  this.dateUploaded = dateUploaded;
}
public String getDateUploaded() {
  return dateUploaded;
}
public void setStartDate(String startDate) {
  this.startDate = startDate;
}
public String getStartDate() {
  return startDate;
}
public void setEndDate(String endDate) {
  this.endDate = endDate;
}
public String getEndDate() {
  return endDate;
}
```

```
public void setVersion(String version) {
  this.version = version;
}
public String getVersion() {
  return version;
}
public void setReference(String reference) {
  this.reference = reference;
}
public String getReference() {
  return reference;
}
public void setReferenceId(int referenceId)
{
  this.referenceId = referenceId;
}
public int getReferenceId() {
  return referenceId;
}
public void setPrivilege(int privilege) {
  this.privilege = privilege;
}
public int getPrivilege() {
  return privilege;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if (Utilities.isEmpty(this.getImagePath())
  ||
  this.getAccessibility().equals(VirholexCons
  tants.DEFAULT_VALUE) ||
  this.getCategory().equals(VirholexConstants
  .DEFAULT_VALUE)) {
    errors.add("errorMessage", new
    ActionMessage("error.required.asterisk"))
    ;

    if(Utilities.isEmpty(this.getImagePath())
    )
      errors.add("file", new
      ActionMessage(""));
    if(this.getAccessibility().equals(Virhole
    xConstants.DEFAULT_VALUE))
      errors.add("accessibility", new
      ActionMessage(""));
    if(this.getCategory().equals(VirholexCons
    tants.DEFAULT_VALUE))
      errors.add("category", new
      ActionMessage(""));
  }

  if(!Utilities.isEmpty(this.getImagePath()))
  {
    if(!(Utilities.getFileExtension(this.getI
    magePath()).equalsIgnoreCase("jpg") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("jpeg") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("gif") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("png") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("bmp") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("tiff") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("svg") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("flv") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("avi") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("mp4") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("wmv") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("exe") ||
    Utilities.getFileExtension(this.getImageP
    ath()).equalsIgnoreCase("jar"))) {
      errors.add("file", new
      ActionMessage("error.invalid.image"));
    }
  }

  return errors;
 }

}
```

ViewImageDetailsVO.java

```
package com.virholex.images;

import org.apache.struts.upload.FormFile;

import com.virholex.general.VO;

public class ViewImageDetailsVO implements VO
{

 private static final long serialVersionUID =
 -6784807879154690922L;
 private int imageId;
 private int imageSetId;
 private FormFile file;
 private String imageSet;
 private String imagePath;
 private String imagePathDB;
 private String accessibility;
 private String category;
 private String description;
 private String cellType;
 private String virusType;
 private String multOfInfection;
 private String medium;
 private String temperature;
 private String CO2Level;
 private String owner;
 private String container;
 private String phenolRed;
 private String magnification;
 private String objective;
 private String illuminationType;
 private String illuminationLevel;
 private String filter;
 private String uploadedBy;
 private String dateUploaded;
 private String startDate;
 private String endDate;
 private String version;
 private String reference;
 private int referenceId;
 private int privilege;

 public void setImageId(int imageId) {
   this.imageId = imageId;
 }
 public int getImageId() {
   return imageId;
 }
```

```
public void setImageSetId(int imageSetId) {
  this.imageSetId = imageSetId;
}
public int getImageSetId() {
  return imageSetId;
}
public void setFile(FormFile file) {
  this.file = file;
}
public FormFile getFile() {
  return file;
}
public void setImageSet(String imageSet) {
  this.imageSet = imageSet;
}
public String getImageSet() {
  return imageSet;
}
public void setImagePath(String imagePath) {
  this.imagePath = imagePath;
}
public String getImagePath() {
  return imagePath;
}
public void setImagePathDB(String
imagePathDB) {
  this.imagePathDB = imagePathDB;
}
public String getImagePathDB() {
  return imagePathDB;
}
public void setAccessibility(String
accessibility) {
  this.accessibility = accessibility;
}
public String getAccessibility() {
  return accessibility;
}
public void setCategory(String category) {
  this.category = category;
}
public String getCategory() {
  return category;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setCellType(String cellType) {
  this.cellType = cellType;
}
public String getCellType() {
  return cellType;
}
public void setVirusType(String virusType) {
  this.virusType = virusType;
}
public String getVirusType() {
  return virusType;
}
public void setMultOfInfection(String
multOfInfection) {
  this.multOfInfection = multOfInfection;
}
public String getMultOfInfection() {
  return multOfInfection;
}
public void setMedium(String medium) {
  this.medium = medium;
}
public String getMedium() {
  return medium;
}
public void setTemperature(String
temperature) {
  this.temperature = temperature;
}
public String getTemperature() {
  return temperature;
}
public void setCO2Level(String cO2Level) {
  CO2Level = cO2Level;
}
public String getCO2Level() {
  return CO2Level;
}
public void setOwner(String owner) {
  this.owner = owner;
}
public String getOwner() {
  return owner;
}
public void setContainer(String container) {
  this.container = container;
}
public String getContainer() {
  return container;
}
public void setPhenolRed(String phenolRed) {
  this.phenolRed = phenolRed;
}
public String getPhenolRed() {
  return phenolRed;
}
public void setMagnification(String
magnification) {
  this.magnification = magnification;
}
public String getMagnification() {
  return magnification;
}
public void setObjective(String objective) {
  this.objective = objective;
}
public String getObjective() {
  return objective;
}
public void setIlluminationType(String
illuminationType) {
  this.illuminationType = illuminationType;
}
public String getIlluminationType() {
  return illuminationType;
}
public void setIlluminationLevel(String
illuminationLevel) {
  this.illuminationLevel = illuminationLevel;
}
public String getIlluminationLevel() {
  return illuminationLevel;
}
public void setFilter(String filter) {
  this.filter = filter;
}
public String getFilter() {
  return filter;
}
public void setUploadedBy(String uploadedBy)
{
  this.uploadedBy = uploadedBy;
}
public String getUploadedBy() {
  return uploadedBy;
}
public void setDateUploaded(String
dateUploaded) {
  this.dateUploaded = dateUploaded;
}
public String getDateUploaded() {
```

```
  return dateUploaded;
 }
 public void setStartDate(String startDate) {
  this.startDate = startDate;
 }
 public String getStartDate() {
  return startDate;
 }
 public void setEndDate(String endDate) {
  this.endDate = endDate;
 }
 public String getEndDate() {
  return endDate;
 }
 public void setVersion(String version) {
  this.version = version;
 }
 public String getVersion() {
  return version;
 }
 public void setReference(String reference) {
  this.reference = reference;
 }
 public String getReference() {
  return reference;
 }
 public void setReferenceId(int referenceId)
 {
  this.referenceId = referenceId;
 }
 public int getReferenceId() {
  return referenceId;
 }
 public void setPrivilege(int privilege) {
  this.privilege = privilege;
 }
 public int getPrivilege() {
  return privilege;
 }

}
```

ViewImageSetDetailsAction.java

```
package com.virholex.images;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;

import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.virus.UserProfileVO;
import
com.virholex.experiments.VirhoExperimentsWorke
r;
import com.virholex.virus.VirhoVIRUSWorker;

public class ViewImageSetDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
 ViewImageSetDetailsForm imageSetForm =
 (ViewImageSetDetailsForm) form;
 ViewImageSetDetailsVO imageSetVO = new
 ViewImageSetDetailsVO();
 UserProfileVO userVO = new UserProfileVO();
 VirhoImagesWorker imageWorker = new
 VirhoImagesWorker();
 VirhoExperimentsWorker experimentWorker =
 new VirhoExperimentsWorker();
 VirhoVIRUSWorker virusWorker = new
 VirhoVIRUSWorker();
 HttpSession session =
 request.getSession(true);
 ArrayList<String> imagesList = new
 ArrayList<String>();

 String username =
 (String)session.getAttribute("userLogin");
 String forward =
 "virholex.images.viewImageSet";
 String roleName = "", projectSet = "";

 if(request.getParameter("module")!=null)
  session.setAttribute("module",
  VirholexConstants.VIRHO_IMAGES);
 if(request.getParameter("projectSet")!=null)
 {
  projectSet =
  request.getParameter("projectSet");
  session.setAttribute("projectSet",
  projectSet);
 } else {
  projectSet =
  (String)session.getAttribute("projectSet");
 }

 if(imageSetForm.getCommand()!=null &&
 imageSetForm.getCommand().equals("Add
 Images")) {
  forward = "virholex.images.addImage";
 } else if(imageSetForm.getCommand()!=null &&
 imageSetForm.getCommand().equals("Edit Image
 Set")) {
  forward = "virholex.images.editImageSet";
 } else if(imageSetForm.getCommand()!=null &&
 imageSetForm.getCommand().equals("Delete
 Image Set")) {
  imagesList =
  imageWorker.getPrivateImages(projectSet);

  for(int i=0; i<imagesList.size(); i++) {
   Utilities.deleteFile(request,
   VirholexConstants.IMAGES_REPOSITORY,
   imagesList.get(i));
  }

  imageWorker.deleteImageSet(projectSet);

  forward = "virholex.images.viewImageSets";

 } else if(imageSetForm.getCommand()!=null &&
 imageSetForm.getCommand().equals("View
 Members")) {
  forward = "virholex.images.viewUsers";
 } else if(imageSetForm.getCommand()!=null &&
 imageSetForm.getCommand().equals("Include in
 View")) {
  forward = "virholex.images.addSavedView";
```

```
} else if(imageSetForm.getCommand()!=null &&
imageSetForm.getCommand().equals("My Saved
View")) {
  forward = "virholex.images.viewSavedViews";
} else if(imageSetForm.getCommand()!=null &&
imageSetForm.getCommand().equals("Send
Membership Request")) {
  session.setAttribute("project",
  projectSet);
  forward =
  "virholex.virus.requestPrivilege";
} else {
  imageSetVO.setProjectSetName(projectSet);
  imageSetVO =
  (ViewImageSetDetailsVO)imageWorker.readImag
  eSet(imageSetVO);
  userVO =
  (UserProfileVO)virusWorker.getName(imageSet
  VO.getAddedBy());
  imageSetVO.setAddedBy(userVO.getFirstName()
  + " " + userVO.getLastName());
  imageSetVO.setDateAdded(Utilities.dateForma
  t(imageSetVO.getDateAdded()));
  imageSetVO.setDateModified(Utilities.dateFo
  rmat(imageSetVO.getDateModified()));

  roleName =
  imageWorker.getProjectRole(username,
  projectSet);

  if(roleName.equals(VirholexConstants.IMAGE_
  SET_COORDINATOR_DESC))
    imageSetVO.setPrivilege(VirholexConstants
    .IMAGE_SET_COORDINATOR);
  else
  if(roleName.equals(VirholexConstants.IMAGE_
  SET_CONTRIBUTOR_DESC))
    imageSetVO.setPrivilege(VirholexConstants
    .IMAGE_SET_CONTRIBUTOR);
  else
  if(roleName.equals(VirholexConstants.RESTRI
  CTED_USER_DESC))
    imageSetVO.setPrivilege(VirholexConstants
    .RESTRICTED_USER);
  else
    imageSetVO.setPrivilege(VirholexConstants
    .REGISTERED_USER);

  if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("Delete")
  &&
  request.getParameterValues("imagesList")!=n
  ull) {
    String[] images =
    request.getParameterValues("imagesList");
    ArrayList<String> files = new
    ArrayList<String>();
    String checkbox = "";

    for(int i=0; i<images.length; i++) {
      checkbox = getImage(request,
      images[i]);
      if(checkbox!=null &&
      checkbox.equals(VirholexConstants.TRUE)
      ) {
        files.add(images[i]);
      }
    }

    if(!files.isEmpty()) {
      for(int i =0; i<files.size(); i++) {
        Utilities.deleteFile(request,
        VirholexConstants.IMAGES_REPOSITORY,
        files.get(i));
      }
      imageWorker.deleteImages(files);
    }
  }

  // get images
  if(imageSetVO.getPrivilege() > 1)
    imageSetVO.setImagesDB(imageWorker.getPri
    vateImages(projectSet));
  else
    imageSetVO.setImagesDB(imageWorker.getPub
    licImages(projectSet));

  for(int i=0;
  i<imageSetVO.getImagesDB().size(); i++) {
    String image =
    imageSetVO.getImagesDB().get(i);
    image =
    Utilities.removeFileExtension(image,
    "."+Utilities.getFileExtension(image));
    imagesList.add(image);
  }

  imageSetVO.setImages(imagesList);
  imageSetVO.setExperiments(experimentWorker.
  getExperiments(imageSetVO.getProjectSetId()
  ));

  if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("Download"
  ) &&
  request.getParameterValues("imagesList")!=n
  ull) {
    String[] images =
    request.getParameterValues("imagesList");
    ArrayList<String> files = new
    ArrayList<String>();
    String filename = "", zippedFile = "",
    checkbox = "", pathname = "";

    for(int i=0; i<images.length; i++) {
      checkbox = getImage(request,
      images[i]);

      if(checkbox!=null &&
      checkbox.equals(VirholexConstants.TRUE)
      ) {
        Utilities.copyFile(request, images[i],
        VirholexConstants.IMAGES_REPOSITORY,
        VirholexConstants.IMAGES_TEMP_REPOSITO
        RY);
        files.add(images[i]);
      }
    }

    if(!files.isEmpty()) {

      if(files.size()>1) {
        filename =
        imageSetVO.getProjectSetName();
        // Zip the file
        zippedFile =
        Utilities.zipFile(request, files,
        filename,
        VirholexConstants.IMAGES_TEMP_REPOSITO
        RY);
        response.setContentType("application/z
        ip");
        response.setHeader("Content-
        Disposition", "attachment;filename=" +
        zippedFile);
        pathname =
        request.getSession().getServletContext
        ().getRealPath(VirholexConstants.IMAGE
        S_TEMP_REPOSITORY) + File.separator +
        zippedFile;
      } else {
```

```
        response.setContentType("image/jpg");
        response.setHeader("Content-
        Disposition", "attachment;filename=" +
        files.get(0));
        pathname =
        request.getSession().getServletContext
        ().getRealPath(VirholexConstants.IMAGE
        S_TEMP_REPOSITORY) + File.separator +
        files.get(0);
      }

      File file = new File(pathname);
      InputStream in = new
      FileInputStream(file);
      ServletOutputStream out =
      response.getOutputStream();

      try {
        int bit = in.read();
        while((bit) >= 0) {
          out.write(bit);
          bit = in.read();
        }
      } catch (IOException e) {
        e.printStackTrace();
      } finally {
        out.flush();
        out.close();
        in.close();
        file.delete();
      }
    }
  }
 }

 PropertyUtils.copyProperties(imageSetForm,
 imageSetVO);

 session.setAttribute("projectSet",
 projectSet);

 return mapping.findForward(forward);
}

public String getImage(HttpServletRequest
request, String image) {
 return request.getParameter("img-" + image);
}

}
```

ViewImageSetDetailsForm.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;

import com.virholex.general.Utilities;
import
com.virholex.experiments.ViewExperimentDetails
VO;

public class ViewImageSetDetailsForm extends
ActionForm {

 private static final long serialVersionUID =
 -120093072935720016L;
 private int experimentId;
 private int projectSetId;
 private String projectSetName;
 private String projectSetNameDB;
 private String description;
 private String addedBy;
 private String dateAdded;
 private String dateModified;
 private String experimentName;
 private String command;
 private int privilege;
 private ArrayList<String> images;
 private ArrayList<String> imagesDB;
 private ArrayList<ViewExperimentDetailsVO>
 experiments;

 public void setExperimentId(int
 experimentId) {
  this.experimentId = experimentId;
 }
 public int getExperimentId() {
  return experimentId;
 }
 public void setProjectSetId(int
 projectSetId) {
  this.projectSetId = projectSetId;
 }
 public int getProjectSetId() {
  return projectSetId;
 }
 public void setProjectSetName(String
 projectSetName) {
  this.projectSetName = projectSetName;
 }
 public String getProjectSetName() {
  return projectSetName;
 }
 public void setProjectSetNameDB(String
 projectSetNameDB) {
  this.projectSetNameDB = projectSetNameDB;
 }
 public String getProjectSetNameDB() {
  return projectSetNameDB;
 }
 public void setDescription(String
 description) {
  this.description = description;
 }
 public String getDescription() {
  return description;
 }
 public void setAddedBy(String addedBy) {
  this.addedBy = addedBy;
 }
 public String getAddedBy() {
  return addedBy;
 }
 public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
 }
 public String getDateAdded() {
  return dateAdded;
 }
 public void setDateModified(String
 dateModified) {
  this.dateModified = dateModified;
 }
 public String getDateModified() {
  return dateModified;
 }
 public void setExperimentName(String
 experimentName) {
  this.experimentName = experimentName;
 }
 public String getExperimentName() {
  return experimentName;
 }
```

```
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setImages(ArrayList<String>
  images) {
    this.images = images;
  }
  public ArrayList<String> getImages() {
    return images;
  }
  public void setImagesDB(ArrayList<String>
  imagesDB) {
    this.imagesDB = imagesDB;
  }
  public ArrayList<String> getImagesDB() {
    return imagesDB;
  }
  public void
  setExperiments(ArrayList<ViewExperimentDetai
  lsVO> experiments) {
    this.experiments = experiments;
  }
  public ArrayList<ViewExperimentDetailsVO>
  getExperiments() {
    return experiments;
  }

  public ActionErrors validate(ActionMapping
  mapping, HttpServletRequest request) {

    ActionErrors errors = new ActionErrors();

    if
    (Utilities.isEmpty(this.getProjectSetName()
    )) {
      errors.add("errorMessage", new
      ActionMessage("error.required.asterisk"))
      ;
      errors.add("projectSetName", new
      ActionMessage(""));
    }

    return errors;
  }

}
```

ViewImageSetDetailsVO.java

```
package com.virholex.images;

import java.util.ArrayList;

public class ViewImageSetDetailsVO implements
VO {

  private static final long serialVersionUID =
  -8946517689592974365L;
  private int experimentId;
  private int projectSetId;
  private String projectSetName;
  private String projectSetNameDB;
  private String description;
  private String addedBy;
  private String dateAdded;
  private String dateModified;
  private int privilege;
  private ArrayList<String> images;
  private ArrayList<String> imagesDB;
  private ArrayList<ViewExperimentDetailsVO>
  experiments;

  public void setExperimentId(int
  experimentId) {
    this.experimentId = experimentId;
  }
  public int getExperimentId() {
    return experimentId;
  }
  public void setProjectSetId(int
  projectSetId) {
    this.projectSetId = projectSetId;
  }
  public int getProjectSetId() {
    return projectSetId;
  }
  public void setProjectSetName(String
  projectSetName) {
    this.projectSetName = projectSetName;
  }
  public String getProjectSetName() {
    return projectSetName;
  }
  public void setProjectSetNameDB(String
  projectSetNameDB) {
    this.projectSetNameDB = projectSetNameDB;
  }
  public String getProjectSetNameDB() {
    return projectSetNameDB;
  }
  public void setDescription(String
  description) {
    this.description = description;
  }
  public String getDescription() {
    return description;
  }
  public void setAddedBy(String addedBy) {
    this.addedBy = addedBy;
  }
  public String getAddedBy() {
    return addedBy;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }
  public void setDateModified(String
  dateModified) {
    this.dateModified = dateModified;
  }
  public String getDateModified() {
    return dateModified;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setImages(ArrayList<String>
  images) {
    this.images = images;
  }
  public ArrayList<String> getImages() {
    return images;
  }
  public void setImagesDB(ArrayList<String>
  imagesDB) {
```

```
    this.imagesDB = imagesDB;
  }
  public ArrayList<String> getImagesDB() {
    return imagesDB;
  }
  public void
  setExperiments(ArrayList<ViewExperimentDetai
  lsVO> experiments) {
    this.experiments = experiments;
  }
  public ArrayList<ViewExperimentDetailsVO>
  getExperiments() {
    return experiments;
  }

}
```

ViewImageSetsAction.java

```
package com.virholex.images;

import java.util.ArrayList;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;

public class ViewImageSetsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewImageSetDetailsForm imageSetForm =
  (ViewImageSetDetailsForm) form;
  ViewImageSetDetailsVO imageSetVO = new
  ViewImageSetDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewImageSetDetailsVO>
  imageSetList = new
  ArrayList<ViewImageSetDetailsVO>();
  ArrayList<ViewImageDetailsVO>
  recentImagesList = new
  ArrayList<ViewImageDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.viewImageSets";
  String roleName = "", project = "";
  boolean isLead = false;

  if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("Add Image
  Set")) {
    forward = "virholex.images.addImageSet";
  } else if(imageSetForm.getCommand()!=null &&
  imageSetForm.getCommand().equals("My Saved
  Views")) {
    forward = "virholex.images.viewSavedViews";
  } else {
    imageSetList =
    imageWorker.getImageSets(imageSetVO);
    recentImagesList =
    imageWorker.getRecentImages();
    isLead =
    imageWorker.isImageSetCoordinator(username)
    ;

    for(int i=0; i<imageSetList.size(); i++) {
      project =
      ((ViewImageSetDetailsVO)imageSetList.get(
      i)).getProjectSetName();
      roleName =
      imageWorker.getProjectRole(username,
      project);

      if(roleName.equals(VirholexConstants.IMAG
      E_SET_COORDINATOR_DESC))
        ((ViewImageSetDetailsVO)imageSetList.ge
        t(i)).setPrivilege(VirholexConstants.IM
        AGE_SET_COORDINATOR);
      else
      if(roleName.equals(VirholexConstants.IMAG
      E_SET_CONTRIBUTOR_DESC))
        ((ViewImageSetDetailsVO)imageSetList.ge
        t(i)).setPrivilege(VirholexConstants.IM
        AGE_SET_CONTRIBUTOR);
      else
      if(roleName.equals(VirholexConstants.REST
      RICTED_USER_DESC))
        ((ViewImageSetDetailsVO)imageSetList.ge
        t(i)).setPrivilege(VirholexConstants.RE
        STRICTED_USER);
      else
        ((ViewImageSetDetailsVO)imageSetList.ge
        t(i)).setPrivilege(VirholexConstants.RE
        GISTERED_USER);
    }

    request.setAttribute("imageSets",
    imageSetList);
    request.setAttribute("recentImages",
    recentImagesList);
    request.setAttribute("isLead", isLead);

  }

  PropertyUtils.copyProperties(imageSetForm,
  imageSetVO);

  session.setAttribute("module",
  VirholexConstants.VIRHO_IMAGES);

  return mapping.findForward(forward);
}
}
```

ViewSavedViewDetailsAction.java

```
package com.virholex.images;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
```

```
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class ViewSavedViewDetailsAction
extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewSavedViewDetailsForm savedViewForm =
  (ViewSavedViewDetailsForm) form;
  ViewSavedViewDetailsVO savedViewVO = new
  ViewSavedViewDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> imagesList = new
  ArrayList<String>();

  String savedView =
  (String)session.getAttribute("savedView");
  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.savedViewDetails";

  if(request.getParameter("savedView")!=null)
  {
    savedView =
    request.getParameter("savedView");
    session.setAttribute("savedView",
    savedView);
  }

  if(savedViewForm.getCommand()!=null &&
  savedViewForm.getCommand().equals("Edit
  View")) {
    forward = "virholex.images.editSavedView";
  } else if(savedViewForm.getCommand()!=null
  && savedViewForm.getCommand().equals("Delete
  View")) {
    imageWorker.deleteView(username,
    savedView);
    forward = "virholex.images.viewSavedViews";
  }else {
    if(savedViewForm.getCommand()!=null &&
    savedViewForm.getCommand().equals("Delete")
    &&
    request.getParameterValues("imagesList")!=n
    ull) {
      String[] images =
      request.getParameterValues("imagesList");
      ArrayList<String> files = new
      ArrayList<String>();
      String checkbox = "";

      for(int i=0; i<images.length; i++) {
        checkbox = getImage(request,
        images[i]);
        if(checkbox!=null &&
        checkbox.equals(VirholexConstants.TRUE)
        ) {
          files.add(images[i]);
        }
      }

      if(!files.isEmpty()) {
        imageWorker.deleteIncludedImages(savedV
        iew, files);
      }
    }

    imagesList =
    imageWorker.getIncludedImages(username,
    savedView);
    savedViewVO.setViewName(savedView);
    savedViewVO.setImagesDB(imagesList);
    imagesList = new ArrayList<String>();

    for(int i=0;
    i<savedViewVO.getImagesDB().size(); i++) {
      String image =
      savedViewVO.getImagesDB().get(i);
      image =
      Utilities.removeFileExtension(image,
      "."+Utilities.getFileExtension(image));
      imagesList.add(image);
    }

    savedViewVO.setImages(imagesList);
  }

  if(savedViewForm.getCommand()!=null &&
  savedViewForm.getCommand().equals("Download"
  ) &&
  request.getParameterValues("imagesList")!=nu
  ll) {
    String[] images =
    request.getParameterValues("imagesList");
    ArrayList<String> files = new
    ArrayList<String>();
    String filename = "", zippedFile = "",
    checkbox = "", pathname = "";

    for(int i=0; i<images.length; i++) {
      checkbox = getImage(request, images[i]);
      if(checkbox!=null &&
      checkbox.equals(VirholexConstants.TRUE))
      {
        Utilities.copyFile(request, images[i],
        VirholexConstants.IMAGES_REPOSITORY,
        VirholexConstants.IMAGES_TEMP_REPOSITOR
        Y);
        files.add(images[i]);
      }
    }

    if(!files.isEmpty()) {

      if(files.size()>1) {
        filename = savedViewVO.getViewName();
        // Zip the file
        zippedFile = Utilities.zipFile(request,
        files, filename,
        VirholexConstants.IMAGES_TEMP_REPOSITOR
        Y);
        response.setContentType("application/zi
        p");
        response.setHeader("Content-
        Disposition", "attachment;filename=" +
        zippedFile);
        pathname =
        request.getSession().getServletContext(
        ).getRealPath(VirholexConstants.IMAGES_
        TEMP_REPOSITORY) + File.separator +
        zippedFile;
      } else {
        response.setContentType("image/jpg");
        response.setHeader("Content-
        Disposition", "attachment;filename=" +
        files.get(0));
        pathname =
        request.getSession().getServletContext(
```

```
      ).getRealPath(VirholexConstants.IMAGES_
      TEMP_REPOSITORY) + File.separator +
      files.get(0);
    }

    File file = new File(pathname);
    InputStream in = new
    FileInputStream(file);
    ServletOutputStream out =
    response.getOutputStream();

    try {
      int bit = in.read();
      while((bit) >= 0) {
        out.write(bit);
        bit = in.read();
      }
    } catch (IOException e) {
      e.printStackTrace();
    } finally {
      out.flush();
      out.close();
      in.close();
      file.delete();
    }
  }
 }

 PropertyUtils.copyProperties(savedViewForm,
 savedViewVO);

 return mapping.findForward(forward);
}

public String getImage(HttpServletRequest
request, String image) {
 return request.getParameter("img-" + image);
}

}
```

ViewSavedViewDetailsForm.java

```
package com.virholex.images;

import java.util.ArrayList;

public class ViewSavedViewDetailsForm extends
ActionForm {

 private static final long serialVersionUID =
 1052103936843406329L;
 private int viewId;
 private String username;
 private String viewName;
 private String viewNameDB;
 private String oldViewName;
 private String dateAdded;
 private String dateModified;
 private String command;
 private String optionView;
 private ArrayList<String> images = new
 ArrayList<String>();
 private ArrayList<String> imagesDB = new
 ArrayList<String>();

 public void setViewId(int viewId) {
   this.viewId = viewId;
 }
 public int getViewId() {
   return viewId;
 }
 public void setUsername(String username) {
   this.username = username;
}
public String getUsername() {
 return username;
}
public void setViewName(String viewName) {
 this.viewName = viewName;
}
public String getViewName() {
 return viewName;
}
public void setViewNameDB(String viewNameDB)
{
 this.viewNameDB = viewNameDB;
}
public String getViewNameDB() {
 return viewNameDB;
}
public void setOldViewName(String
oldViewName) {
 this.oldViewName = oldViewName;
}
public String getOldViewName() {
 return oldViewName;
}
public void setDateAdded(String dateAdded) {
 this.dateAdded = dateAdded;
}
public String getDateAdded() {
 return dateAdded;
}
public void setDateModified(String
dateModified) {
 this.dateModified = dateModified;
}
public String getDateModified() {
 return dateModified;
}
public void setCommand(String command) {
 this.command = command;
}
public String getCommand() {
 return command;
}
public void setOptionView(String optionView)
{
 this.optionView = optionView;
}
public String getOptionView() {
 return optionView;
}
public void setImages(ArrayList<String>
images) {
 this.images = images;
}
public ArrayList<String> getImages() {
 return images;
}
public void setImagesDB(ArrayList<String>
imagesDB) {
 this.imagesDB = imagesDB;
}
public ArrayList<String> getImagesDB() {
 return imagesDB;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

 ActionErrors errors = new ActionErrors();

 if (Utilities.isEmpty(this.getViewName())
 && this.getOptionView().equals("New View"))
 {
   errors.add("errorMessage", new
   ActionMessage("error.required", "View
   Name"));
```

```
      errors.add("viewName", new
      ActionMessage(""));
    } else if (this.getOptionView().equals("Old
    View") &&
    this.getViewNameDB().equals(VirholexConstan
    ts.DEFAULT_VALUE)) {
      errors.add("errorMessage", new
      ActionMessage("error.select.default",
      "Old View"));
      errors.add("viewNameDB", new
      ActionMessage(""));
    }

  return errors;
  }

}
```

ViewSavedViewDetailsVO.java

```
package com.virholex.images;

import java.util.ArrayList;

import com.virholex.general.VO;

public class ViewSavedViewDetailsVO implements
VO {

  private static final long serialVersionUID =
  -3044599668035819374L;
  private int viewId;
  private String username;
  private String viewName;
  private String viewNameDB;
  private String oldViewName;
  private String dateAdded;
  private String dateModified;
  private ArrayList<String> images = new
  ArrayList<String>();
  private ArrayList<String> imagesDB = new
  ArrayList<String>();

  public void setViewId(int viewId) {
    this.viewId = viewId;
  }
  public int getViewId() {
    return viewId;
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }
  public void setViewName(String viewName) {
    this.viewName = viewName;
  }
  public String getViewName() {
    return viewName;
  }
  public void setViewNameDB(String viewNameDB)
  {
    this.viewNameDB = viewNameDB;
  }
  public String getViewNameDB() {
    return viewNameDB;
  }
  public void setOldViewName(String
  oldViewName) {
    this.oldViewName = oldViewName;
  }
  public String getOldViewName() {
    return oldViewName;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }
  public void setDateModified(String
  dateModified) {
    this.dateModified = dateModified;
  }
  public String getDateModified() {
    return dateModified;
  }
  public void setImages(ArrayList<String>
  images) {
    this.images = images;
  }
  public ArrayList<String> getImages() {
    return images;
  }
  public void setImagesDB(ArrayList<String>
  imagesDB) {
    this.imagesDB = imagesDB;
  }
  public ArrayList<String> getImagesDB() {
    return imagesDB;
  }

}
```

ViewSavedViewImageAction.java

```
package com.virholex.images;

import java.util.ArrayList;

public class ViewSavedViewImageAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewSavedViewDetailsForm savedViewForm =
  (ViewSavedViewDetailsForm) form;
  ViewSavedViewDetailsVO savedViewVO = new
  ViewSavedViewDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> imagesList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.viewSavedViewImage";
  String savedView =
  (String)session.getAttribute("savedView");
  String imageDB =
  request.getParameter("image");
  String image =
  Utilities.removeFileExtension(imageDB,
  "."+Utilities.getFileExtension(imageDB));
  String imageTemp = "";

  PropertyUtils.copyProperties(savedViewVO,
  savedViewForm);
```

```
  imagesList =
  imageWorker.getIncludedImages(username,
  savedView);
  savedViewVO.setViewName(savedView);
  savedViewVO.setImagesDB(imagesList);
  imagesList = new ArrayList<String>();

  for(int i=0;
  i<savedViewVO.getImagesDB().size(); i++) {
    imageTemp =
    savedViewVO.getImagesDB().get(i);
    imageTemp =
    Utilities.removeFileExtension(imageTemp,
    "."+Utilities.getFileExtension(imageTemp));
    imagesList.add(imageTemp);
  }

  savedViewVO.setImages(imagesList);
  PropertyUtils.copyProperties(savedViewForm,
  savedViewVO);

  request.setAttribute("imageDB", imageDB);
  request.setAttribute("image", image);

  return mapping.findForward(forward);
}
}
```

ViewSavedViewsAction.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

public class ViewSavedViewsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewSavedViewDetailsForm savedViewForm =
  (ViewSavedViewDetailsForm) form;
  ViewSavedViewDetailsVO savedViewVO = new
  ViewSavedViewDetailsVO();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewSavedViewDetailsVO>
  savedViewsList = new
  ArrayList<ViewSavedViewDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.images.viewSavedViews";

  PropertyUtils.copyProperties(savedViewVO,
  savedViewForm);

  savedViewsList =
  imageWorker.getSavedViews(username);

  PropertyUtils.copyProperties(savedViewForm,
  savedViewVO);

  session.setAttribute("savedViews",
  savedViewsList);

  return mapping.findForward(forward);
}
}
```

ViewUsersAction.java

```
package com.virholex.images;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.virus.*;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<String> rolesList = new
  ArrayList<String>();
  ArrayList<String> usersList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String projectSet =
  (String)session.getAttribute("projectSet");
  String forward =
  "virholex.images.viewUsers";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", roleName = "";
  int privilege = 0;

  if(!Utilities.isEmpty(request.getParameter("
  list")))
    list= request.getParameter("list");

  for(int i=0; i < letters.length() ; i++){
    if(virusWorker.getUsers(projectSet,
    Character.toString(letters.charAt(i)),
    VirholexConstants.VIRHO_IMAGES).size() > 0)
```

```
    letterList.add(Character.toString(letters
    .charAt(i)));
  else
    letterList.add("-");
}

usersForm.setLetters(letterList);

if(usersForm.getCommand()!=null &&
usersForm.getCommand().equals("Change
Privilege Role")) {
  String[] roles =
  request.getParameterValues("role");
  String[] oldRoles =
  request.getParameterValues("oldRole");
  String[] users =
  request.getParameterValues("userName");

  if(roles!=null && oldRoles!=null &&
  users!=null) {
    for(int i=0; i<oldRoles.length; i++) {
      if(!roles[i].equals(oldRoles[i])) {
        rolesList.add(roles[i]);
        usersList.add(users[i]);
      }
    }

    if(!rolesList.isEmpty() &&
    !usersList.isEmpty()) {
      virusWorker.updateProjectPrivilege(user
      sList, rolesList, projectSet,
      VirholexConstants.VIRHO_IMAGES);
      rolesList = new ArrayList<String>();
    }
  }
}

userVO.setUsers(virusWorker.getUsers(project
Set, list, VirholexConstants.VIRHO_IMAGES));

rolesList =
virusWorker.getUserRoles(VirholexConstants.V
IRHO_IMAGES);
roleName =
imageWorker.getProjectRole(username,
projectSet);

if(roleName.equals(VirholexConstants.IMAGE_S
ET_COORDINATOR_DESC))
  privilege =
  VirholexConstants.IMAGE_SET_COORDINATOR;
else
if(roleName.equals(VirholexConstants.IMAGE_S
ET_CONTRIBUTOR_DESC))
  privilege =
  VirholexConstants.IMAGE_SET_CONTRIBUTOR;
else
if(roleName.equals(VirholexConstants.RESTRIC
TED_USER_DESC))
  privilege =
  VirholexConstants.RESTRICTED_USER;
else
  privilege =
  VirholexConstants.REGISTERED_USER;

PropertyUtils.copyProperties(usersForm,
userVO);

request.setAttribute("privilege",
privilege);
request.setAttribute("roles", rolesList);

return mapping.findForward(forward);
}
}
```

VirhoImagesWorker.java

```
package com.virholex.images;

import java.util.ArrayList;

import com.virholex.general.VO;
import com.virholex.general.Worker;

public class VirhoImagesWorker implements
Worker {

  public ArrayList<ViewImageSetDetailsVO>
  getImageSets(ViewImageSetDetailsVO vo)
  throws Exception {
    return ImageSetsDAO.getImageSets(vo);
  }

  public ArrayList<ViewImageSetDetailsVO>
  getImageSets(int experimentId) throws
  Exception {
    return
    ImageSetsDAO.getImageSets(experimentId);
  }

  public String getProjectRole(String
  username, String project) throws Exception {
    return
    ImageSetsDAO.getProjectSetRole(username,
    project);
  }

  public boolean isImageSetCoordinator(String
  username) throws Exception {
    return
    ImageSetsDAO.isImageSetCoordinator(username
    );
  }

  public ViewImageSetDetailsVO
  readImageSet(ViewImageSetDetailsVO vo)
  throws Exception {
    return ImageSetsDAO.readImageSet(vo);
  }

  public ViewImageDetailsVO
  readImageMetadata(ViewImageDetailsVO vo)
  throws Exception {
    return
    (ViewImageDetailsVO)ImageDAO.readImageMetad
    ata(vo);
  }

  public ArrayList<String>
  getPublicImages(String projectSet) throws
  Exception {
    return
    ImageSetsDAO.getPublicImages(projectSet);
  }

  public ArrayList<String>
  getPrivateImages(String projectSet) throws
  Exception {
    return
    ImageSetsDAO.getPrivateImages(projectSet);
  }

  public ArrayList<ViewSavedViewDetailsVO>
  getSavedViews(String username) throws
  Exception {
    return
    SavedViewsDAO.getSavedViews(username);
  }
```

```
public ArrayList<ViewImageDetailsVO>
getRecentImages() throws Exception {
  return ImageDAO.getRecentImages();
}

public ArrayList<String>
getIncludedImages(String username, String
title) throws Exception {
  return
  SavedViewsDAO.getIncludedImages(username,
  title);
}

public void
insertImageSet(ViewImageSetDetailsVO vo)
throws Exception {
  ImageSetsDAO.insertImageSet(vo);
}

public void
updateImageSet(ViewImageSetDetailsVO vo)
throws Exception {
  ImageSetsDAO.updateImageSet(vo);
}

public void
insertView(ViewSavedViewDetailsVO vo) throws
Exception {
  SavedViewsDAO.insertView(vo);
}

public void
insertToExistingView(ViewSavedViewDetailsVO
vo) throws Exception {
  SavedViewsDAO.insertToExistingView(vo);
}

public void
updateView(ViewSavedViewDetailsVO vo) throws
Exception {
  SavedViewsDAO.updateView(vo);
}

public void
mergeToExistingView(ViewSavedViewDetailsVO
vo) throws Exception {
  SavedViewsDAO.mergeToExistingView(vo);
}

public void
insertImages(ArrayList<ViewImageDetailsVO>
list) throws Exception {
  ImageDAO.insertImages(list);
}

public void updateImage(ViewImageDetailsVO
vo) throws Exception {
  ImageDAO.updateImage(vo);
}

public void deleteImage(String image) throws
Exception {
  ImageDAO.deleteImage(image);
}

public void deleteImages(ArrayList<String>
images) throws Exception {
  ImageDAO.deleteImages(images);
}

public void deleteIncludedImages(String
title, ArrayList<String> images) throws
Exception {
  ImageDAO.deleteIncludedImages(title,
  images);
}

  public void deleteView(String username,
  String viewName) throws Exception {
    SavedViewsDAO.deleteView(username,
    viewName);
  }

  public void deleteImageSet(String
  projectSetName) throws Exception {
    ImageSetsDAO.deleteImageSet(projectSetName)
    ;
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VIRHO MODELS

AddModelDetailsAction.java

```
package com.virholex.models;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.virus.UserProfileVO;
import com.virholex.virus.VirhoVIRUSWorker;

public class AddModelDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewModelDetailsForm modelForm =
  (ViewModelDetailsForm) form;
  ViewModelDetailsVO modelVO = new
  ViewModelDetailsVO();
  UserProfileVO userVO = new UserProfileVO();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
```

```
HttpSession session =
request.getSession(true);
FileOutputStream outputStream = null;
FormFile formFile = null;

String forward = "virholex.models.addModel";
String username =
(String)session.getAttribute("userLogin");
String project =
(String)session.getAttribute("project");
String classification =
(String)session.getAttribute("classification
");
String author = "";

PropertyUtils.copyProperties(modelVO,
modelForm);

if(modelVO.getClassification()==null)
  modelVO.setClassification(classification);

if(modelForm.getCommand() != null &&
modelForm.getCommand().equals("Submit")) {

  if(modelVO.getModelName()!= null) {
    classification =
    modelVO.getClassification();
    if(modelsWorker.checkModel(modelVO)) {
      modelForm.setHasModel(true);
    }
  }

  errors = modelForm.validate(mapping,
  request);

  if(errors.isEmpty()) {
    try {
      // Save file
      formFile = modelVO.getFile();
      String filename =
      Utilities.formatFileName(formFile);
      String filePath =
      getServlet().getServletContext().getRea
      lPath(VirholexConstants.MODELS_REPOSITO
      RY) + File.separator + filename;
      outputStream = new FileOutputStream(new
      File(filePath));
      outputStream.write(formFile.getFileData
      ());
      modelVO.setFileDescriptionDB(filename);

    } catch(FileNotFoundException e) {
      e.printStackTrace();
    } finally {
      outputStream.close();
    }
    // Get the name of author
    userVO = (UserProfileVO)
    virusWorker.getName(username);
    author = userVO.getFirstName() + " " +
    userVO.getLastName();
    modelVO.setAuthor(author);
    // Insert model details in the database
    modelsWorker.addModel(modelVO);
    // set new session for classification
    session.setAttribute("classification",
    classification);

    forward = "virholex.models.viewModel";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("project",
    project);
    redirect.addParameter("model",
    modelForm.getModelName());
      return redirect;
    } else {
      saveErrors(request, errors);
    }
  } else if(modelForm.getCommand() != null &&
  modelForm.getCommand().equals("Cancel")) {
    forward = "virholex.models.viewProject";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("project", project);
    return redirect;
  }

  PropertyUtils.copyProperties(modelForm,
  modelVO);

  return mapping.findForward(forward);
}
}
```

EditModelDetailsAction.java

```
package com.virholex.models;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.virus.UserProfileVO;
import com.virholex.virus.VirhoVIRUSWorker;

public class AddModelDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewModelDetailsForm modelForm =
  (ViewModelDetailsForm) form;
  ViewModelDetailsVO modelVO = new
  ViewModelDetailsVO();
  UserProfileVO userVO = new UserProfileVO();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  FileOutputStream outputStream = null;
  FormFile formFile = null;

  String forward = "virholex.models.addModel";
  String username =
  (String)session.getAttribute("userLogin");
```

```
String project =
(String)session.getAttribute("project");
String classification =
(String)session.getAttribute("classification
");
String author = "";

PropertyUtils.copyProperties(modelVO,
modelForm);

if(modelVO.getClassification()==null)
  modelVO.setClassification(classification);

if(modelForm.getCommand() != null &&
modelForm.getCommand().equals("Submit")) {

  if(modelVO.getModelName()!= null) {
    classification =
    modelVO.getClassification();
    if(modelsWorker.checkModel(modelVO)) {
      modelForm.setHasModel(true);
    }
  }

  errors = modelForm.validate(mapping,
  request);

  if(errors.isEmpty()) {
    try {
      // Save file
      formFile = modelVO.getFile();
      String filename =
      Utilities.formatFileName(formFile);
      String filePath =
      getServlet().getServletContext().getRea
      lPath(VirholexConstants.MODELS_REPOSITO
      RY) + File.separator + filename;
      outputStream = new FileOutputStream(new
      File(filePath));
      outputStream.write(formFile.getFileData
      ());
      modelVO.setFileDescriptionDB(filename);

    } catch(FileNotFoundException e) {
      e.printStackTrace();
    } finally {
      outputStream.close();
    }
    // Get the name of author
    userVO = (UserProfileVO)
    virusWorker.getName(username);
    author = userVO.getFirstName() + " " +
    userVO.getLastName();
    modelVO.setAuthor(author);
    // Insert model details in the database
    modelsWorker.addModel(modelVO);
    // set new session for classification
    session.setAttribute("classification",
    classification);

    forward = "virholex.models.viewModel";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("project",
    project);
    redirect.addParameter("model",
    modelForm.getModelName());
    return redirect;

  } else {
    saveErrors(request, errors);
  }
} else if(modelForm.getCommand() != null &&
modelForm.getCommand().equals("Cancel")) {
  forward = "virholex.models.viewProject";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("project", project);
    return redirect;
  }

  PropertyUtils.copyProperties(modelForm,
  modelVO);

  return mapping.findForward(forward);
}
}
```

EditUserPrivilegeForm.java

```
package com.virholex.models;

import org.apache.struts.action.ActionForm;

public class EditUserPrivilegeForm extends
ActionForm {

  private static final long serialVersionUID =
  -5272721091337087384L;
  private String command;
  private String privilege;
  private String username;

  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setPrivilege(String privilege) {
    this.privilege = privilege;
  }
  public String getPrivilege() {
    return privilege;
  }
  public void setUsername(String username) {
    this.username = username;
  }
  public String getUsername() {
    return username;
  }

}
```

ViewModelDetailsAction.java

```
package com.virholex.models;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.VirholexConstants;
import com.virholex.general.Utilities;
import
com.virholex.references.ViewReferenceDetailsVO
;
```

```
import
com.virholex.references.VirhoReferencesWorker;

public class ViewModelDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewModelDetailsForm modelForm =
  (ViewModelDetailsForm) form;
  ViewModelDetailsVO modelVO = new
  ViewModelDetailsVO();
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  VirhoReferencesWorker referencesWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.models.viewModel";
  String username =
  (String)session.getAttribute("userLogin");
  String classification =
  (String)session.getAttribute("classification
  ");
  String project =
  (String)session.getAttribute("project");
  String model = "", roleName = "";
  int role = 1;

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_MODELS);
  if(request.getParameter("classification") !=
  null) {
    classification =
    request.getParameter("classification");
    session.setAttribute("classification",
    classification);
  }
  if(request.getParameter("project") != null)
  {
    project = request.getParameter("project");
    session.setAttribute("project", project);
  }
  if(request.getParameter("model") != null)
    model = request.getParameter("model");
  else
    model =
    (String)session.getAttribute("model");

  modelVO.setClassification(classification);
  modelVO.setProjectName(project);
  modelVO.setModelName(model);
  modelVO = (ViewModelDetailsVO)
  modelsWorker.getModelDetails(modelVO);

  if(modelForm.getCommand() != null &&
  modelForm.getCommand().equals("Delete
  Model")) {
    int modelId = modelVO.getModelId();

  if(modelsWorker.checkModel(modelId)) {
    Utilities.deleteFile(request,
    VirholexConstants.MODELS_REPOSITORY,
    modelVO.getFileDescriptionDB());
    modelsWorker.deleteModel(modelId);
  }

  forward = "virholex.models.viewProject";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward))
    ;
    redirect.addParameter("project", project);
    return redirect;

  } else {
    // Remove application protocol
    modelVO.setLink(Utilities.removeProtocol(mo
    delVO.getLink()));
    // Remove underscore and set of random
    numbers from PDF file
    modelVO.setFileDescription(Utilities.remove
    FileExtension(modelVO.getFileDescriptionDB(
    ), ".pdf"));
    // Get reference value
    if(!Utilities.isEmpty(String.valueOf(modelV
    O.getReferenceId()))) {
      referenceVO.setReferenceId(modelVO.getRef
      erenceId());
      referenceVO =
      (ViewReferenceDetailsVO)referencesWorker.
      readReference(referenceVO);
      modelVO.setEvidence(referenceVO.getTitle(
      ));
      request.setAttribute("reference",
      referenceVO);
    }

    roleName =
    modelsWorker.getProjectRole(username,
    project);

    if(roleName.equals(VirholexConstants.MODEL_
    CONTRIBUTOR_DESC))
      role =
      VirholexConstants.MODEL_CONTRIBUTOR;
    else
    if(roleName.equals(VirholexConstants.MODEL_
    COORDINATOR_DESC))
      role =
      VirholexConstants.MODEL_COORDINATOR;

      modelVO.setPrivilege(role);
  }

  PropertyUtils.copyProperties(modelForm,
  modelVO);
  session.setAttribute("project", project);
  session.setAttribute("model", model);

  return mapping.findForward(forward);
}
}
```

ViewModelDetailsForm.java

```
package com.virholex.models;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.action.ActionForm;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;

public class ViewModelDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  1723977392190566913L;
```

```
private String modelName;
private String classification;
private String projectName;
private FormFile file;
private String fileDescription;
private String fileDescriptionDB;
private String author;
private String scale;
private String scope;
private String size;
private String type;
private String aspects;
private String tools;
private String properties;
private String evidence;
private String link;
private String command;
private int referenceId;
private int privilege;
private int modelId;
private boolean hasModel;

public void setModelName(String modelName) {
  this.modelName = modelName;
}
public String getModelName() {
  return modelName;
}
public void setClassification(String
classification) {
  this.classification = classification;
}
public String getClassification() {
  return classification;
}
public void setProjectName(String
projectName) {
  this.projectName = projectName;
}
public String getProjectName() {
  return projectName;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setScale(String scale) {
  this.scale = scale;
}
public String getScale() {
  return scale;
}
public void setScope(String scope) {
  this.scope = scope;
}
public String getScope() {
  return scope;
}
public void setSize(String size) {
  this.size = size;
}
public String getSize() {
  return size;
}
public void setType(String type) {
  this.type = type;
}
public String getType() {
  return type;
}
public void setAspects(String aspects) {
  this.aspects = aspects;
}
public String getAspects() {
  return aspects;
}
public void setTools(String tools) {
  this.tools = tools;
}
public String getTools() {
  return tools;
}
public void setProperties(String properties)
{
  this.properties = properties;
}
public String getProperties() {
  return properties;
}
public void setEvidence(String evidence) {
  this.evidence = evidence;
}
public String getEvidence() {
  return evidence;
}
public void setLink(String link) {
  this.link = link;
}
public String getLink() {
  return link;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setReferenceId(int referenceId)
{
  this.referenceId = referenceId;
}
public int getReferenceId() {
return referenceId;
  }
public void setFileDescription(String
fileDescription) {
  this.fileDescription = fileDescription;
}
public String getFileDescription() {
  return fileDescription;
}
public void setFileDescriptionDB(String
fileDescriptionDB) {
  this.fileDescriptionDB = fileDescriptionDB;
}
public String getFileDescriptionDB() {
  return fileDescriptionDB;
}
public void setPrivilege(int privilege) {
  this.privilege = privilege;
}
public int getPrivilege() {
  return privilege;
}
public void setFile(FormFile file) {
  this.file = file;
}
public FormFile getFile() {
  return file;
}
public void setModelId(int modelId) {
  this.modelId = modelId;
}
public int getModelId() {
  return modelId;
}
public void setHasModel(boolean hasModel) {
  this.hasModel = hasModel;
}
public boolean isHasModel() {
```

```
    return hasModel;
  }

  public ActionErrors validate(ActionMapping
  mapping, HttpServletRequest request) {

    ActionErrors errors = new ActionErrors();

    if (Utilities.isEmpty(this.getModelName())
    ||
    (Utilities.isEmpty(this.getFile().toString(
    )) &&
    Utilities.isEmpty(this.getFileDescriptionDB
    ()))) {
      errors.add("errorMessage", new
      ActionMessage("error.required.asterisk"))
      ;

      if(Utilities.isEmpty(this.getModelName())
      )
        errors.add("modelName", new
        ActionMessage(""));
      if(Utilities.isEmpty(this.getFile().toStr
      ing()) &&
      Utilities.isEmpty(this.getFileDescription
      DB()))
        errors.add("file", new
        ActionMessage(""));
    }

    if(!Utilities.isEmpty(this.getModelName()))
    {
      if(this.isHasModel()) {
        errors.add("modelName", new
        ActionMessage("error.notAvailable",
        "model name"));
      }
    }

    if(!Utilities.isEmpty(this.getFile().toStri
    ng())) {
      if(!(Utilities.getFileExtension(this.getF
      ile().toString()).equals("pdf") ||
      Utilities.getFileExtension(this.getFile()
      .toString()).equals("PDF"))) {
        errors.add("file", new
        ActionMessage("error.file",
        this.getFile()));
      }
    }

    return errors;
  }

}
```

ViewModelDetailsVO.java

```
package com.virholex.models;

import org.apache.struts.upload.FormFile;

import com.virholex.general.VO;

public class ViewModelDetailsVO implements VO
{

  private static final long serialVersionUID =
  5322539607265680847L;
  private String modelName;
  private String classification;
  private String projectName;
  private FormFile file;
  private String fileDescription;
  private String fileDescriptionDB;
  private String author;
  private String scale;
  private String scope;
  private String size;
  private String type;
  private String aspects;
  private String tools;
  private String properties;
  private String evidence;
  private String link;
  private int referenceId;
  private int modelId;
  private int privilege;

  public void setModelName(String modelName) {
    this.modelName = modelName;
  }
  public String getModelName() {
    return modelName;
  }
  public void setClassification(String
  classification) {
    this.classification = classification;
  }
  public String getClassification() {
    return classification;
  }
  public void setProjectName(String
  projectName) {
    this.projectName = projectName;
  }
  public String getProjectName() {
    return projectName;
  }
  public void setAuthor(String author) {
    this.author = author;
  }
  public String getAuthor() {
    return author;
  }
  public void setScale(String scale) {
    this.scale = scale;
  }
  public String getScale() {
    return scale;
  }
  public void setScope(String scope) {
    this.scope = scope;
  }
  public String getScope() {
    return scope;
  }
  public void setSize(String size) {
    this.size = size;
  }
  public String getSize() {
    return size;
  }
  public void setType(String type) {
    this.type = type;
  }
  public String getType() {
    return type;
  }
  public void setAspects(String aspects) {
    this.aspects = aspects;
  }
  public String getAspects() {
    return aspects;
  }
  public void setTools(String tools) {
    this.tools = tools;
  }
  public String getTools() {
    return tools;
```

```
  }
  public void setProperties(String properties)
  {
    this.properties = properties;
  }
  public String getProperties() {
    return properties;
  }
  public void setEvidence(String evidence) {
    this.evidence = evidence;
  }
  public String getEvidence() {
    return evidence;
  }
  public void setLink(String link) {
    this.link = link;
  }
  public String getLink() {
    return link;
  }
  public void setReferenceId(int referenceId)
  {
    this.referenceId = referenceId;
  }
  public int getReferenceId() {
    return referenceId;
  }
  public void setFileDescription(String
  fileDescription) {
    this.fileDescription = fileDescription;
  }
  public String getFileDescription() {
    return fileDescription;
  }
  public void setFileDescriptionDB(String
  fileDescriptionDB) {
    this.fileDescriptionDB = fileDescriptionDB;
  }
  public String getFileDescriptionDB() {
    return fileDescriptionDB;
  }
  public void setModelId(int modelId) {
    this.modelId = modelId;
  }
  public int getModelId() {
    return modelId;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setFile(FormFile file) {
    this.file = file;
  }
  public FormFile getFile() {
    return file;
  }

}
```

ViewModelsAction.java

```
package com.virholex.models;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import
com.virholex.experiments.ViewProjectDetailsVO;

public class ViewModelsAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  ViewModelDetailsVO modelVO = new
  ViewModelDetailsVO();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> projectList = new
  ArrayList<String>();
  ArrayList<ViewProjectDetailsVO>
  projectListTemp = new
  ArrayList<ViewProjectDetailsVO>();

  String forward =
  "virholex.models.viewModels";
  String classificationCode = "",
  classificationName = "", project = "";

  if(request.getParameter("classification") !=
  null) {
    classificationCode =
    request.getParameter("classification");
    classificationName =
    getClassification(classificationCode);
    session.setAttribute("classification",
    classificationName);

    projectList = modelsWorker.getProjects();

    for(int i=0; i<projectList.size(); i++) {
      modelVO.setClassification(classificationN
      ame);
      project = projectList.get(i).toString();
      modelVO.setProjectName(project);
      projectVO.setProjectName(project);
      projectVO.setModels(modelsWorker.getModel
      s(modelVO));

      projectListTemp.add(projectVO);
      modelVO = new ViewModelDetailsVO();
      projectVO = new ViewProjectDetailsVO();
    }
  }

  request.setAttribute("projects",
  projectListTemp);

  return mapping.findForward(forward);
}

public String getClassification(String code) {

  String classification = "";

  if (code.equals("cd") ||
  code.equals("Conceptual Diagrammatic"))
    classification = "Conceptual Diagrammatic";
  else if (code.equals("cqd") ||
  code.equals("Conceptual Qualitative
  Discrete"))
    classification = "Conceptual Qualitative
    Discrete";
```

```
  else if (code.equals("cqc") ||
  code.equals("Conceptual Qualitative
  Continuous"))
    classification = "Conceptual Qualitative
    Continuous";
  else if (code.equals("cqns") ||
  code.equals("Conceptual Quantitative
  Stochastic"))
    classification = "Conceptual Quantitative
    Stochastic";
  else if (code.equals("cqnd") ||
  code.equals("Conceptual Quantitative
  Deterministic"))
    classification = "Conceptual Quantitative
    Deterministic";
  else if (code.equals("ctd") ||
  code.equals("Contextual Diagrammatic"))
    classification = "Contextual Diagrammatic";
  else if (code.equals("ctqd") ||
  code.equals("Contextual Qualitative
  Discrete"))
    classification = "Contextual Qualitative
    Discrete";
  else if (code.equals("ctqc") ||
  code.equals("Contextual Qualitative
  Continuous"))
    classification = "Contextual Qualitative
    Continuous";
  else if (code.equals("ctqns") ||
  code.equals("Contextual Quantitative
  Stochastic"))
    classification = "Contextual Quantitative
    Stochastic";
  else if (code.equals("ctqnd") ||
  code.equals("Contextual Quantitative
  Deterministic"))
    classification = "Contextual Quantitative
    Deterministic";

  return classification;
}

}
```

ViewModelsDAO.java

```
package com.virholex.models;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;

public class ViewModelsDAO {

  public static ArrayList<ViewModelDetailsVO>
  getModels(ViewModelDetailsVO mvo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<ViewModelDetailsVO> modelList =
    new ArrayList<ViewModelDetailsVO>();
    ViewModelDetailsVO modelVO = new
    ViewModelDetailsVO();

    try {
      String sql = "SELECT name FROM models
      WHERE project_name=? AND
      classification=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, mvo.getProjectName());
      ps.setString(2, mvo.getClassification());
      rs = ps.executeQuery();

      while(rs.next()) {
        modelVO.setModelName(rs.getString("name
        "));
        modelList.add(modelVO);
        modelVO = new ViewModelDetailsVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return modelList;
  }

  public static ArrayList<String>
  getModels(String projectName) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<String> modelList = new
    ArrayList<String>();

    try {
      String sql = "SELECT description FROM
      models WHERE project_name=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, projectName);
      rs = ps.executeQuery();

      while(rs.next()) {
        modelList.add(rs.getString(1));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return modelList;
  }

  public static ViewModelDetailsVO
  getModelDetails(ViewModelDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "SELECT * FROM models WHERE
      name=? AND project_name=? AND
      classification=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getModelName());
      ps.setString(2, vo.getProjectName());
      ps.setString(3, vo.getClassification());
      rs = ps.executeQuery();

      while(rs.next()) {
```

```
    vo.setModelName(rs.getString("name"));
    vo.setAuthor(rs.getString("author"));
    vo.setFileDescriptionDB(rs.getString("d
    escription"));
    vo.setClassification(rs.getString("clas
    sification"));
    vo.setProjectName(rs.getString("project
    _name"));
    vo.setScale(rs.getString("scale"));
    vo.setScope(rs.getString("scope"));
    vo.setSize(rs.getString("size"));
    vo.setType(rs.getString("type"));
    vo.setAspects(rs.getString("aspects"));
    vo.setTools(rs.getString("tools"));
    vo.setProperties(rs.getString("properti
    es"));
    vo.setReferenceId(rs.getInt("evidence")
    );
    vo.setLink(rs.getString("link"));
    vo.setModelId(rs.getInt("model_id"));
  }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return vo;
}

public static boolean checkModel(int
modelId) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean isExisting = false;

  try {
    String sql = "SELECT * FROM models WHERE
    model_id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, modelId);
    rs = ps.executeQuery();

    if(rs.next()) {
      isExisting = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isExisting;
}

public static boolean
isModelExisted(ViewModelDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean isExisting = false;

  try {
    String sql = "SELECT * FROM models WHERE
    classification=? AND project_name=? AND
    name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getClassification());
    ps.setString(2, vo.getProjectName());
    ps.setString(3, vo.getModelName());
    rs = ps.executeQuery();

    if(rs.next()) {
      isExisting = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isExisting;

}

public static void
addModel(ViewModelDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    String sql = "INSERT INTO models (name,
    classification, description,
    project_name, scale, scope, size, type,
    aspects, tools, properties, evidence,
    link, author, date_added) VALUES
    (?,?,?,?,?,?,?,?,?,?,?,?,?,?,NOW())";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getModelName());
    ps.setString(2, vo.getClassification());
    ps.setString(3,
    vo.getFileDescriptionDB());
    ps.setString(4, vo.getProjectName());
    ps.setString(5, vo.getScale());
    ps.setString(6, vo.getScope());
    ps.setString(7, vo.getSize());
    ps.setString(8, vo.getType());
    ps.setString(9, vo.getAspects());
    ps.setString(10, vo.getTools());
    ps.setString(11, vo.getProperties());
    ps.setInt(12, vo.getReferenceId());
    ps.setString(13, vo.getLink());
    ps.setString(14, vo.getAuthor());
    ps.executeUpdate();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void
editModel(ViewModelDetailsVO vo) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {

    String sql = "UPDATE models SET name=?,
    classification=?, description=?,
    project_name=?, scale=?, scope=?, size=?,
    type=?, aspects=?, tools=?, properties=?,
    evidence=?, link=?, author=?,
    date_last_modified=NOW() WHERE
    model_id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
```

```
      ps.setString(1, vo.getModelName());
      ps.setString(2, vo.getClassification());
      ps.setString(3,
      vo.getFileDescriptionDB());
      ps.setString(4, vo.getProjectName());
      ps.setString(5, vo.getScale());
      ps.setString(6, vo.getScope());
      ps.setString(7, vo.getSize());
      ps.setString(8, vo.getType());
      ps.setString(9, vo.getAspects());
      ps.setString(10, vo.getTools());
      ps.setString(11, vo.getProperties());
      ps.setInt(12, vo.getReferenceId());
      ps.setString(13, vo.getLink());
      ps.setString(14, vo.getAuthor());
      ps.setInt(15, vo.getModelId());
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

  public static void deleteModel(int modelId)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      String sql = "DELETE FROM models WHERE
      model_id=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setInt(1, modelId);
      ps.executeUpdate();
    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

ViewProjectDAO.java

```
package com.virholex.models;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import
com.virholex.experiments.ViewProjectDetailsVO;

public class ViewProjectDAO {

  public static ArrayList<String>
  getProjects() throws SQLException,
  NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    ArrayList<String> projectList = new
    ArrayList<String>();

    try {
      String sql = "SELECT * FROM project WHERE
      project_name NOT LIKE '%VirhoHotspot%'
      AND project_name NOT LIKE
      '%VirhoReference%' ";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.executeQuery(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        projectList.add(rs.getString("project_n
        ame"));
      }

    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return projectList;
  }

  public static ViewProjectDetailsVO
  getProjectDetails(ViewProjectDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "SELECT * FROM project WHERE
      project_name=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getProjectName());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setProjectName(rs.getString("project
        _name"));
        vo.setDescription(rs.getString("descrip
        tion"));
        vo.setAuthor(rs.getString("author"));
        vo.setStatus(rs.getString("status"));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

  public static String getProjectRole(String
  username, String project) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String role = "";

    try {
      String sql = "SELECT role_name FROM
      user_prev WHERE username=? AND
      involvement=? AND role_name LIKE
      '%Model%'";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, username);
      ps.setString(2, project);
      rs = ps.executeQuery();
```

```
    if(rs.next()) {
      role = rs.getString("role_name");
    }

   } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
   }

   return role;
   }

}
```

ViewProjectDetailsAction.java

```
package com.virholex.models;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import
com.virholex.experiments.ViewProjectDetailsFor
m;
import
com.virholex.experiments.ViewProjectDetailsVO;

public class ViewProjectDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewProjectDetailsForm projectForm =
  (ViewProjectDetailsForm) form;
  ViewProjectDetailsVO projectVO = new
  ViewProjectDetailsVO();
  ViewModelDetailsVO modelVO = new
  ViewModelDetailsVO();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.models.viewProject";
  String username =
  (String)session.getAttribute("userLogin");
  String classification =
  (String)session.getAttribute("classification
  ");
  String roleName = "", project = "";

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_MODELS);
  if(Utilities.isEmpty(classification)) {
    classification = "Conceptual Diagrammatic";
    session.setAttribute("classification",
    classification);
  }

  if(projectForm.getCommand()!=null &&
  projectForm.getCommand().equals("Add
  Model")) {
    forward = "virholex.models.addModel";
  } else if(projectForm.getCommand()!=null &&
  projectForm.getCommand().equals("View
  Members")) {
    forward = "virholex.models.viewUsers";
  } else if(projectForm.getCommand()!=null &&
  projectForm.getCommand().equals("Send
  Membership Request")) {
    forward =
    "virholex.virus.requestPrivilege";
  } else {

    if(request.getParameter("project") != null)
      project =
      request.getParameter("project");
    else
      project =
      (String)session.getAttribute("project");

    modelVO.setProjectName(project);
    modelVO.setClassification(classification);
    projectVO.setProjectName(project);
    projectVO = (ViewProjectDetailsVO)
    modelsWorker.getProjectDetails(projectVO);
    projectVO.setModels(modelsWorker.getModels(
    modelVO));
    roleName =
    modelsWorker.getProjectRole(username,
    project);

    if(roleName.equals(VirholexConstants.MODEL_
    CONTRIBUTOR_DESC))
      projectVO.setPrivilege(VirholexConstants.
      MODEL_CONTRIBUTOR);
    else
    if(roleName.equals(VirholexConstants.MODEL_
    COORDINATOR_DESC))
      projectVO.setPrivilege(VirholexConstants.
      MODEL_COORDINATOR);
    else
      projectVO.setPrivilege(VirholexConstants.
      RESTRICTED_USER);

    PropertyUtils.copyProperties(projectForm,
    projectVO);

    session.setAttribute("project", project);
  }

  return mapping.findForward(forward);
}

}
```

ViewTaxonomyAction.java

```
package com.virholex.models;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;
```

```
public class ViewTaxonomyAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {

  HttpSession session =
  request.getSession(true);
  String forward =
  "virholex.models.viewTaxonomy";

  session.setAttribute("module",
  VirholexConstants.VIRHO_MODELS);

  return mapping.findForward(forward);
}
}
```

ViewUsersAction.java

```
package com.virholex.models;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;
import com.virholex.virus.*;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  VirhoModelsWorker modelsWorker = new
  VirhoModelsWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<String> rolesList = new
  ArrayList<String>();
  ArrayList<String> usersList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String project =
  (String)session.getAttribute("project");
  String forward =
  "virholex.models.viewUsers";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", roleName = "";
  int privilege = 0;

if(!Utilities.isEmpty(request.getParameter("
list")))
  list= request.getParameter("list");

for(int i=0; i < letters.length() ; i++){
  if(virusWorker.getUsers(project,
  Character.toString(letters.charAt(i)),
  VirholexConstants.VIRHO_MODELS).size() > 0)
    letterList.add(Character.toString(letters
    .charAt(i)));
  else
    letterList.add("-");
}

usersForm.setLetters(letterList);

if(usersForm.getCommand()!=null &&
usersForm.getCommand().equals("Change
Privilege Role")) {
  String[] roles =
  request.getParameterValues("role");
  String[] oldRoles =
  request.getParameterValues("oldRole");
  String[] users =
  request.getParameterValues("userName");

  if(roles!=null && oldRoles!=null &&
  users!=null) {
    for(int i=0; i<oldRoles.length; i++) {
      if(!roles[i].equals(oldRoles[i])) {
        rolesList.add(roles[i]);
        usersList.add(users[i]);
      }
    }

    if(!rolesList.isEmpty() &&
    !usersList.isEmpty()) {
      virusWorker.updateProjectPrivilege(user
      sList, rolesList, project,
      VirholexConstants.VIRHO_MODELS);
      rolesList = new ArrayList<String>();
    }
  }
}

userVO.setUsers(virusWorker.getUsers(project
, list, VirholexConstants.VIRHO_MODELS));
rolesList =
virusWorker.getUserRoles(VirholexConstants.V
IRHO_MODELS);
roleName =
modelsWorker.getProjectRole(username,
project);

if(roleName.equals(VirholexConstants.MODEL_C
ONTRIBUTOR_DESC))
  privilege =
  VirholexConstants.MODEL_CONTRIBUTOR;
else
if(roleName.equals(VirholexConstants.MODEL_C
OORDINATOR_DESC))
  privilege =
  VirholexConstants.MODEL_COORDINATOR;
else
  privilege =
  VirholexConstants.REGISTERED_USER;

PropertyUtils.copyProperties(usersForm,
userVO);

request.setAttribute("privilege",
privilege);
request.setAttribute("roles", rolesList);

return mapping.findForward(forward);
```

```
}
}
```

VirhoModelsWorker.java

```
package com.virholex.models;

import java.util.ArrayList;

import com.virholex.general.VO;
import com.virholex.general.Worker;
import
com.virholex.experiments.ViewProjectDetailsVO;

public class VirhoModelsWorker implements
Worker {

  public ArrayList<String> getProjects()
  throws Exception {
    return ViewProjectDAO.getProjects();
  }

  public ViewProjectDetailsVO
  getProjectDetails(ViewProjectDetailsVO vo)
  throws Exception {
    return
    ViewProjectDAO.getProjectDetails(vo);
  }

  public String getProjectRole(String
  username, String project) throws Exception {
    return
    ViewProjectDAO.getProjectRole(username,
    project);
  }

  public ArrayList<ViewModelDetailsVO>
  getModels(ViewModelDetailsVO mvo) throws
  Exception {
    return ViewModelsDAO.getModels(mvo);
  }

  public ArrayList<String> getModels(String
  projectName) throws Exception {
    return
    ViewModelsDAO.getModels(projectName);
  }

  public ViewModelDetailsVO
  getModelDetails(ViewModelDetailsVO vo)
  throws Exception {
    return ViewModelsDAO.getModelDetails(vo);
  }

  public boolean checkModel(int modelId)
  throws Exception {
    return ViewModelsDAO.checkModel(modelId);
  }

  public boolean checkModel(ViewModelDetailsVO
  vo) throws Exception {
    return ViewModelsDAO.isModelExisted(vo);
  }

  public void addModel(ViewModelDetailsVO vo)
  throws Exception {
    ViewModelsDAO.addModel(vo);
  }

  public void editModel(ViewModelDetailsVO vo)
  throws Exception {
    ViewModelsDAO.editModel(vo);
  }

  public void deleteModel(int modelId) throws
  Exception {
    ViewModelsDAO.deleteModel(modelId);
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VIRHO REFERENCES

AddCollectionAction.java

```
package com.virholex.references;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.VirholexConstants;

public class AddCollectionAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewCollectionDetailsForm collectionForm =
  (ViewCollectionDetailsForm) form;
  ViewCollectionDetailsVO collectionVO = new
  ViewCollectionDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.references.addCollection";
  String username =
  (String)session.getAttribute("userLogin");
  String collection = "";

  PropertyUtils.copyProperties(collectionVO,
  collectionForm);

  if(collectionForm.getCommand() != null &&
  collectionForm.getCommand().equals("Submit")
  ) {

    if(collectionVO.getCollectionName()!=null)
    {
```

```
      collection =
      collectionVO.getCollectionName();
      if(referenceWorker.checkCollection(collec
      tion)) {
        collectionForm.setHasCollection(true);
      }
    }

    errors = collectionForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      collectionVO.setAuthor(username);
      collectionVO.setType(VirholexConstants.PR
      OJECT);
      referenceWorker.insertCollection(collecti
      onVO);

      session.setAttribute("collection",
      collection);

      forward =
      "virholex.references.viewReferences";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("collection",
      collection);
      return redirect;
    } else {
      saveErrors(request, errors);
    }

  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Cancel")
  ) {
    forward =
    "virholex.references.viewCollections";
  }

  PropertyUtils.copyProperties(collectionForm,
  collectionVO);

  return mapping.findForward(forward);
}
}
```

AddCollectionDAO.java

```
package com.virholex.references;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.VirholexConstants;

public class AddCollectionDAO {

  public static void
  insertCollection(ViewCollectionDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {

      conn = DBConnect.getConnection();

      String sql = "INSERT INTO collections
      (title, description, date_added,
      date_modified, downloadable)
      VALUES(?,?,NOW(),NOW(),?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCollectionName());
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getDownloadable());
      ps.executeUpdate();
      ps.close ();

      sql = "INSERT INTO project (project_name,
      description, author, date_added,
      date_modified)
      VALUES(?,?,?,NOW(),NOW())";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCollectionName() +
      " " + VirholexConstants.VIRHO_REFERENCE);
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getAuthor());
      ps.executeUpdate();
      ps.close ();

      sql = "INSERT INTO user_prev (username,
      role_name, involvement, date_added, kind)
      VALUES(?,?,?,NOW(),?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getAuthor());
      ps.setString(2,
      VirholexConstants.COLLECTION_COORDINATOR_
      DESC);
      ps.setString(3, vo.getCollectionName() +
      " " + VirholexConstants.VIRHO_REFERENCE);
      ps.setString(4, vo.getType());
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

  public static boolean checkCollection(String
  collection) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean hasCollection = false;

    try {
      conn = DBConnect.getConnection();

      String sql = "SELECT * FROM collections
      WHERE title=? OR title LIKE ?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, collection);
      ps.setString(2, collection);
      rs = ps.executeQuery();

      if(rs.next()) {
        hasCollection = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return hasCollection;
  }

}
```

AddReferenceAction.java

```
package com.virholex.references;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class AddReferenceAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewReferenceDetailsForm referenceForm =
  (ViewReferenceDetailsForm) form;
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  FileOutputStream outputStream = null;
  FormFile formFile = null;

  String forward =
  "virholex.references.addReference";
  String collection = "", reference = "";
  int refId = 0;

  if(request.getParameter("collection")!=null)
    collection =
    request.getParameter("collection");
  else
    collection =
    (String)session.getAttribute("collection");

  PropertyUtils.copyProperties(referenceVO,
  referenceForm);
  referenceVO.setCollection(collection);

  if(referenceForm.getCommand() != null &&
  referenceForm.getCommand().equals("Submit"))
  {
    errors = referenceForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      if(!referenceVO.getFileName().toString().
      equals("")) {
        try {
          // Save file
          formFile = referenceVO.getFileName();
          String filename =
          Utilities.formatFileName(formFile);
          String filePath =
          getServlet().getServletContext().getRe
          alPath(VirholexConstants.REFERENCES_RE
          POSITORY) + File.separator + filename;
          outputStream = new
          FileOutputStream(new File(filePath));
          outputStream.write(formFile.getFileDat
          a());

          referenceVO.setFileDB(filename);
        } catch(FileNotFoundException e) {
          e.printStackTrace();
        } finally {
          outputStream.close();
        }
      }

      if(referenceVO.getMonth().equals(Virholex
      Constants.DEFAULT_VALUE))
        referenceVO.setMonth("");

      referenceWorker.insertReference(reference
      VO);
      reference = referenceVO.getTitle();
      refId = referenceVO.getReferenceId();

      session.setAttribute("reference",
      reference);

      forward =
      "virholex.references.viewReferenceDetails
      ";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("reference",
      reference);
      redirect.addParameter("refId", refId);
      return redirect;

    } else {
      saveErrors(request, errors);
    }

  } else if(referenceForm.getCommand() != null
  &&
  referenceForm.getCommand().equals("Cancel"))
  {
    forward =
    "virholex.references.viewReferences";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("collection",
    collection);
    return redirect;
  }

  PropertyUtils.copyProperties(referenceForm,
  referenceVO);

  session.setAttribute("collection",
  collection);

  return mapping.findForward(forward);
}
}
```

AddReferenceDAO.java

```
package com.virholex.references;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.naming.NamingException;

import com.virholex.general.DBConnect;

public class AddReferenceDAO {

  public static void
  insertReference(ViewReferenceDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {

      conn = DBConnect.getConnection();

      String sql = "INSERT INTO coll_reference
      (title, author, publisher, year, chapter,
      pages, booktitle, school, institution,
      note, volume, series, address, edition,
      month, type, editor, organization,
      number, howpublished, collection,
      journal, date_added, date_modified, link,
      path, subtype)
      VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
      ?,?,?,?,?,NOW(),NOW(),?,?,?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getTitle());
      ps.setString(2, vo.getAuthor());
      ps.setString(3, vo.getPublisher());
      ps.setInt(4, vo.getYear());
      ps.setString(5, vo.getChapter());
      ps.setString(6, vo.getPages());
      ps.setString(7, vo.getBooktitle());
      ps.setString(8, vo.getSchool());
      ps.setString(9, vo.getInstitution());
      ps.setString(10, vo.getNote());
      ps.setString(11, vo.getVolume());
      ps.setString(12, vo.getSeries());
      ps.setString(13, vo.getAddress());
      ps.setString(14, vo.getEdition());
      ps.setString(15, vo.getMonth());
      ps.setString(16, vo.getType());
      ps.setString(17, vo.getEditor());
      ps.setString(18, vo.getOrganization());
      ps.setString(19, vo.getNumber());
      ps.setString(20, vo.getHowpublished());
      ps.setString(21, vo.getCollection());
      ps.setString(22, vo.getJournal());
      ps.setString(23, vo.getLink());
      ps.setString(24, vo.getFileDB());
      ps.setString(25, vo.getSubtype());
      ps.executeUpdate();
      ps.close();

      sql = "SELECT LAST_INSERT_ID() AS id";
      ps = conn.prepareStatement(sql);
      rs = ps.executeQuery();
      rs.next();
      vo.setReferenceId(rs.getInt("id"));

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

  }

  public static boolean
  checkReference(ViewReferenceDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    boolean hasReference = false;

    try {

      conn = DBConnect.getConnection();

      String sql = "SELECT * FROM
      coll_reference WHERE title=? OR title
      LIKE ? AND collection=? AND type=? AND
      author=? AND year=? AND month=? AND
      volume=? AND series=? AND pages=? AND
      school=? AND organization=? AND
      institution=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getTitle());
      ps.setString(2, vo.getTitle());
      ps.setString(3, vo.getCollection());
      ps.setString(4, vo.getType());
      ps.setString(5, vo.getAuthor());
      ps.setInt(6, vo.getYear());
      ps.setString(7, vo.getMonth());
      ps.setString(8, vo.getVolume());
      ps.setString(9, vo.getSeries());
      ps.setString(10, vo.getPages());
      ps.setString(11, vo.getSchool());
      ps.setString(12, vo.getOrganization());
      ps.setString(13, vo.getInstitution());
      rs = ps.executeQuery();

      if(rs.next()) {
        hasReference = true;
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return hasReference;
  }

}
```

DisplayReferencesAction.java

```
package com.virholex.references;

import java.util.ArrayList;

public class DisplayReferencesAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewReferenceDetailsForm referenceForm =
  (ViewReferenceDetailsForm) form;
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  ArrayList<ViewReferenceDetailsVO>
  referenceList = new
  ArrayList<ViewReferenceDetailsVO>();
```

```
  String forward =
  "virholex.references.displayReferences";

  PropertyUtils.copyProperties(referenceVO,
  referenceForm);

  referenceList =
  referenceWorker.getReferences();

  PropertyUtils.copyProperties(referenceForm,
  referenceVO);

  request.setAttribute("references",
  referenceList);

  return mapping.findForward(forward);
}

public String getReference(HttpServletRequest
request, int referenceId) {
  return
  request.getParameter("id"+referenceId);
}

}
```

EditCollectionAction.java

```
package com.virholex.references;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import
org.apache.struts.action.ActionRedirect;

import com.virholex.general.VirholexConstants;

public class EditCollectionAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewCollectionDetailsForm collectionForm =
  (ViewCollectionDetailsForm) form;
  ViewCollectionDetailsVO collectionVO = new
  ViewCollectionDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);

  String forward =
  "virholex.references.editCollection";
  String username =
  (String)session.getAttribute("userLogin");
  String collection = "";

  if(request.getParameter("collection")!=null)
    collection =
    request.getParameter("collection");
  else
    collection =
    (String)session.getAttribute("collection");

  PropertyUtils.copyProperties(collectionVO,
  collectionForm);
  collectionVO.setOldCollectionName(collection
  );

  if(collectionForm.getCommand() != null &&
  collectionForm.getCommand().equals("Submit")
  ) {
    if(collectionVO.getCollectionName()!=null
    &&
    !collectionVO.getCollectionName().equals(co
    llectionVO.getOldCollectionName())) {
      collection =
      collectionVO.getCollectionName();
      if(referenceWorker.checkCollection(collec
      tion)) {
        collectionForm.setHasCollection(true);
      }
    }

    errors = collectionForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      collectionVO.setAuthor(username);
      collectionVO.setType(VirholexConstants.PR
      OJECT);
      referenceWorker.updateCollection(collecti
      onVO);

      forward =
      "virholex.references.viewReferences";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("collection",
      collection);
      return redirect;
    } else {
      saveErrors(request, errors);
    }

  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Cancel")
  ) {
    forward =
    "virholex.references.viewReferences";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("collection",
    collection);
    return redirect;
  } else {
    collectionVO = (ViewCollectionDetailsVO)
    referenceWorker.readCollection(collectionVO
    );
  }

  PropertyUtils.copyProperties(collectionForm,
  collectionVO);

  session.setAttribute("collection",
  collection);

  return mapping.findForward(forward);
}
}
```

EditCollectionDAO.java

```
package com.virholex.references;

import java.sql.Connection;

public class EditCollectionDAO {

  public static void
  updateCollection(ViewCollectionDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();

      String sql = "UPDATE collections SET
      title=?, description=?, downloadable=?,
      date_modified=NOW() WHERE title=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCollectionName());
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getDownloadable());
      ps.setString(4,
      vo.getOldCollectionName());
      ps.executeUpdate();
      ps.close ();

      sql = "UPDATE user_prev SET
      involvement=?, date_modified=NOW() WHERE
      involvement=? AND kind=? AND username=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCollectionName() +
      " " + VirholexConstants.VIRHO_REFERENCE);
      ps.setString(2, vo.getOldCollectionName()
      + " " +
      VirholexConstants.VIRHO_REFERENCE);
      ps.setString(3, vo.getType());
      ps.setString(4, vo.getAuthor());
      ps.executeUpdate();
      ps.close ();

      sql = "UPDATE project SET project_name=?,
      description=? WHERE project_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCollectionName() +
      " " + VirholexConstants.VIRHO_REFERENCE);
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getOldCollectionName()
      + " " +
      VirholexConstants.VIRHO_REFERENCE);
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

EditReferenceAction.java

```
package com.virholex.references;

import java.io.File;

public class EditReferenceAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewReferenceDetailsForm referenceForm =
  (ViewReferenceDetailsForm) form;
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  ActionErrors errors = new ActionErrors();
  HttpSession session =
  request.getSession(true);
  FileOutputStream outputStream = null;
  FormFile formFile = null;

  String forward =
  "virholex.references.editReference";
  String collection = "", reference = "";
  String link = "", pdf = "";
  int refId = 0;

  if(request.getParameter("collection")!=null)
    collection =
    request.getParameter("collection");
  else
    collection =
    (String)session.getAttribute("collection");

  PropertyUtils.copyProperties(referenceVO,
  referenceForm);
  referenceVO.setCollection(collection);

  if(referenceForm.getCommand() != null &&
  referenceForm.getCommand().equals("Submit"))
  {
    errors = referenceForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      if(request.getParameter("removeFile")!=nu
      ll) {
        Utilities.deleteFile(request,
        VirholexConstants.REFERENCES_REPOSITORY
        , referenceVO.getFileDB());
        referenceVO.setFile("");
        referenceVO.setFileDB("");
      } else
      if(!referenceVO.getFileName().toString().
      equals("")) {
        // Delete old file
        Utilities.deleteFile(request,
        VirholexConstants.REFERENCES_REPOSITORY
        , referenceVO.getFileDB());
        // Save file
        try {
          formFile = referenceVO.getFileName();
          String filename =
          Utilities.formatFileName(formFile);
          String filePath =
          getServlet().getServletContext().getRe
          alPath(VirholexConstants.REFERENCES_RE
          POSITORY) + File.separator + filename;
          outputStream = new
          FileOutputStream(new File(filePath));
          outputStream.write(formFile.getFileDat
          a());

          referenceVO.setFileDB(filename);
        } catch(FileNotFoundException e) {
          e.printStackTrace();
        } finally {
          outputStream.close();
        }
      }
```

```
    if(referenceVO.getMonth().equals(Virholex
    Constants.DEFAULT_VALUE))
      referenceVO.setMonth("");

    referenceWorker.updateReference(reference
    VO);
    reference = referenceVO.getTitle();
    refId = referenceVO.getReferenceId();

    session.setAttribute("reference",
    reference);

    forward =
    "virholex.references.viewReferenceDetails
    ";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forwar
    d));
    redirect.addParameter("reference",
    reference);
    redirect.addParameter("refId", refId);
    return redirect;

  } else {
    saveErrors(request, errors);
  }
 } else if(referenceForm.getCommand() != null
 &&
 referenceForm.getCommand().equals("Cancel"))
 {
  forward =
  "virholex.references.viewReferences";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("collection",
  collection);
  return redirect;
 } else {
  referenceVO =
  (ViewReferenceDetailsVO)referenceWorker.rea
  dReference(referenceVO);
  referenceVO.setCollection(collection);

  // Remove application protocol
  link = referenceVO.getLink();
  referenceVO.setLink(Utilities.removeProtoco
  l(link));
  // Remove underscore and set of random
  numbers from PDF file
  pdf = referenceVO.getFileDB();
  referenceVO.setFile(Utilities.removeFileExt
  ension(pdf, ".pdf")); // path
 }

 PropertyUtils.copyProperties(referenceForm,
 referenceVO);

 return mapping.findForward(forward);
}
}
```

EditReferenceDAO.java

```
package com.virholex.references;

import java.sql.Connection;

public class EditReferenceDAO {

  public static void
  updateReference(ViewReferenceDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {

      String sql = "UPDATE coll_reference SET
      title=?, author=?, publisher=?, year=?,
      chapter=?, pages=?, booktitle=?,
      school=?, institution=?, note=?,
      volume=?, series=?, address=?, edition=?,
      month=?, type=?, editor=?,
      organization=?, number=?, howpublished=?,
      collection=?, journal=?,
      date_modified=NOW(), link=?, path=?,
      subtype=? WHERE id=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getTitle());
      ps.setString(2, vo.getAuthor());
      ps.setString(3, vo.getPublisher());
      ps.setInt(4, vo.getYear());
      ps.setString(5, vo.getChapter());
      ps.setString(6, vo.getPages());
      ps.setString(7, vo.getBooktitle());
      ps.setString(8, vo.getSchool());
      ps.setString(9, vo.getInstitution());
      ps.setString(10, vo.getNote());
      ps.setString(11, vo.getVolume());
      ps.setString(12, vo.getSeries());
      ps.setString(13, vo.getAddress());
      ps.setString(14, vo.getEdition());
      ps.setString(15, vo.getMonth());
      ps.setString(16, vo.getType());
      ps.setString(17, vo.getEditor());
      ps.setString(18, vo.getOrganization());
      ps.setString(19, vo.getNumber());
      ps.setString(20, vo.getHowpublished());
      ps.setString(21, vo.getCollection());
      ps.setString(22, vo.getJournal());
      ps.setString(23, vo.getLink());
      ps.setString(24, vo.getFileDB());
      ps.setString(25, vo.getSubtype());
      ps.setInt(26, vo.getReferenceId());
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

}
```

ExportReferencesAction.java

```
package com.virholex.references;

import java.io.BufferedWriter;

public class ExportReferencesAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewCollectionDetailsForm collectionForm =
  (ViewCollectionDetailsForm) form;
  ViewCollectionDetailsVO collectionVO = new
  ViewCollectionDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
```

```
HttpSession session =
request.getSession(true);

String forward =
"virholex.references.exportReferences";
String collection =
(String)session.getAttribute("collection");

PropertyUtils.copyProperties(collectionVO,
collectionForm);

if(session.getAttribute("collectionVO")!=nul
l) {

  collectionVO =
  (ViewCollectionDetailsVO)session.getAttribu
  te("collectionVO");
  session.removeAttribute("collectionVO");
}

if(collectionForm.getCommand() != null &&
collectionForm.getCommand().equals("Export")
) {

  ViewReferenceDetailsVO vo = new
  ViewReferenceDetailsVO();
  String[] references =
  request.getParameterValues("myReferences");
  String pathname =
  request.getSession().getServletContext().ge
  tRealPath(VirholexConstants.REFERENCES_REPO
  SITORY) + File.separator;
  String fileType =
  request.getParameter("fileType");
  String filename = "", str = "";
  final String newline = "\n";
  int referenceId = 0, id = 0;

  BufferedWriter out = null;
  InputStream is = null;
  ServletOutputStream sos =
  response.getOutputStream();
  File file = null;

  if(fileType!=null) {

    if(fileType.equals("xml")) {

      filename =
      Utilities.formatFileName(collection,
      ".xml");
      pathname = pathname + filename;

      response.setContentType("plain/text");
      response.setHeader("Content-
      Disposition", "attachment;filename=" +
      filename);

      try {

        out = new BufferedWriter(new
        FileWriter(pathname, true));
        out.write("<XML>" + newline);
        out.write("  <RECORDS>" + newline);

        for(int i=0; i<references.length; i++)
        {
          str = references[i];
          id = str.lastIndexOf("id");
          referenceId =
          Integer.parseInt(str.substring(id+2)
          );
          vo.setReferenceId(referenceId);
          vo =
          (ViewReferenceDetailsVO)referenceWor
          ker.readReference(vo);

          out.write("    <RECORD>" + newline);

          if(vo.getType().equals("Article")) {

            out.write("      <COLLECTION>" +
            vo.getCollection() +
            "</COLLECTION>" + newline);
            out.write("      <REFERENCE_TYPE>"
            + vo.getType() +
            "</REFERENCE_TYPE>" + newline);
            out.write("      <TITLE>" +
            vo.getTitle() + "</TITLE>" +
            newline);
            out.write("      <AUTHOR>" +
            vo.getAuthor() + "</AUTHOR>" +
            newline);
            out.write("      <JOURNAL>" +
            vo.getJournal() + "</JOURNAL>" +
            newline);
            out.write("      <VOLUME>" +
            vo.getVolume() + "</VOLUME>" +
            newline);
            out.write("      <NUMBER>" +
            vo.getNumber() + "</NUMBER>" +
            newline);
            out.write("      <PAGES>" +
            vo.getPages() + "</PAGES>" +
            newline);
            out.write("      <MONTH>" +
            vo.getMonth() + "</MONTH>" +
            newline);
            out.write("      <YEAR>" +
            vo.getYear() + "</YEAR>" +
            newline);
            out.write("      <NOTE>" +
            vo.getNote() + "</NOTE>" +
            newline);
            out.write("      <URL>" +
            vo.getLink() + "</URL>" + newline);
            out.write("      <FILENAME>" +
            vo.getFileDB() + "</FILENAME>" +
            newline);

          } else
          if(vo.getType().equals("Book")) {

            out.write("      <COLLECTION>" +
            vo.getCollection() +
            "</COLLECTION>" + newline);
            out.write("      <REFERENCE_TYPE>"
            + vo.getType() +
            "</REFERENCE_TYPE>" + newline);
            out.write("      <TITLE>" +
            vo.getTitle() + "</TITLE>" +
            newline);
            out.write("      <AUTHOR>" +
            vo.getAuthor() + "</AUTHOR>" +
            newline);
            out.write("      <VOLUME>" +
            vo.getVolume() + "</VOLUME>" +
            newline);
            out.write("      <SERIES>" +
            vo.getSeries() + "</SERIES>" +
            newline);
            out.write("      <EDITION>" +
            vo.getEdition() + "</EDITION>" +
            newline);
            out.write("      <MONTH>" +
            vo.getMonth() + "</MONTH>" +
            newline);
            out.write("      <YEAR>" +
            vo.getYear() + "</YEAR>" +
            newline);
```

```
  out.write("      <PUBLISHER>" +
  vo.getPublisher() + "</PUBLISHER>"
  + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Inbook")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <AUTHOR>" +
  vo.getAuthor() + "</AUTHOR>" +
  newline);
  out.write("      <CHAPTER>" +
  vo.getChapter() + "</CHAPTER>" +
  newline);
  out.write("      <VOLUME>" +
  vo.getVolume() + "</VOLUME>" +
  newline);
  out.write("      <SERIES>" +
  vo.getSeries() + "</SERIES>" +
  newline);
  out.write("      <PAGES>" +
  vo.getPages() + "</PAGES>" +
  newline);
  out.write("      <SUBTYPE>" +
  vo.getSubtype() + "</SUBTYPE>" +
  newline);
  out.write("      <EDITION>" +
  vo.getEdition() + "</EDITION>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <PUBLISHER>" +
  vo.getPublisher() + "</PUBLISHER>"
  + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Incollection
")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <AUTHOR>" +
  vo.getAuthor() + "</AUTHOR>" +
  newline);
  out.write("      <BOOKTITLE>" +
  vo.getBooktitle() + "</BOOKTITLE>"
  + newline);
  out.write("      <VOLUME>" +
  vo.getVolume() + "</VOLUME>" +
  newline);
  out.write("      <SERIES>" +
  vo.getSeries() + "</SERIES>" +
  newline);
  out.write("      <CHAPTER>" +
  vo.getChapter() + "</CHAPTER>" +
  newline);
  out.write("      <PAGES>" +
  vo.getPages() + "</PAGES>" +
  newline);
  out.write("      <SUBTYPE>" +
  vo.getSubtype() + "</SUBTYPE>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <EDITION>" +
  vo.getEdition() + "</EDITION>" +
  newline);
  out.write("      <EDITOR>" +
  vo.getEditor() + "</EDITOR>" +
  newline);
  out.write("      <PUBLISHER>" +
  vo.getPublisher() + "</PUBLISHER>"
  + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Proceedings"
)) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <VOLUME>" +
  vo.getVolume() + "</VOLUME>" +
  newline);
  out.write("      <SERIES>" +
  vo.getSeries() + "</SERIES>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
```

```
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <ORGANIZATION>" +
  vo.getOrganization() +
  "</ORGANIZATION>" + newline);
  out.write("      <EDITOR>" +
  vo.getEditor() + "</EDITOR>" +
  newline);
  out.write("      <PUBLISHER>" +
  vo.getPublisher() + "</PUBLISHER>"
  + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Inproceeding
s")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <AUTHOR>" +
  vo.getAuthor() + "</AUTHOR>" +
  newline);
  out.write("      <BOOKTITLE>" +
  vo.getBooktitle() + "</BOOKTITLE>"
  + newline);
  out.write("      <VOLUME>" +
  vo.getVolume() + "</VOLUME>" +
  newline);
  out.write("      <SERIES>" +
  vo.getSeries() + "</SERIES>" +
  newline);
  out.write("      <PAGES>" +
  vo.getPages() + "</PAGES>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <ORGANIZATION>" +
  vo.getOrganization() +
  "</ORGANIZATION>" + newline);
  out.write("      <EDITOR>" +
  vo.getEditor() + "</EDITOR>" +
  newline);
  out.write("      <PUBLISHER>" +
  vo.getPublisher() + "</PUBLISHER>"
  + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Masters
Thesis") || vo.getType().equals("PhD
Thesis")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <AUTHOR>" +
  vo.getAuthor() + "</AUTHOR>" +
  newline);
  out.write("      <SCHOOL>" +
  vo.getSchool() + "</SCHOOL>" +
  newline);
  out.write("      <SUBTYPE>" +
  vo.getSubtype() + "</SUBTYPE>" +
  newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Technical
Report")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <INSTITUTION>" +
  vo.getInstitution() +
  "</INSTITUTION>" + newline);
  out.write("      <SUBTYPE>" +
  vo.getSubtype() + "</SUBTYPE>" +
  newline);
  out.write("      <NUMBER>" +
  vo.getNumber() + "</NUMBER>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
```

```
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Unpublished"
)) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

} else
if(vo.getType().equals("Manual")) {

  out.write("      <COLLECTION>" +
  vo.getCollection() +
  "</COLLECTION>" + newline);
  out.write("      <REFERENCE_TYPE>"
  + vo.getType() +
  "</REFERENCE_TYPE>" + newline);
  out.write("      <TITLE>" +
  vo.getTitle() + "</TITLE>" +
  newline);
  out.write("      <AUTHOR>" +
  vo.getAuthor() + "</AUTHOR>" +
  newline);
  out.write("      <ORGANIZATION>" +
  vo.getOrganization() +
  "</ORGANIZATION>" + newline);
  out.write("      <ADDRESS>" +
  vo.getAddress() + "</ADDRESS>" +
  newline);
  out.write("      <EDITION>" +
  vo.getEdition() + "</EDITION>" +
  newline);
  out.write("      <MONTH>" +
  vo.getMonth() + "</MONTH>" +
  newline);
  out.write("      <YEAR>" +
  vo.getYear() + "</YEAR>" +
  newline);
  out.write("      <NOTE>" +
  vo.getNote() + "</NOTE>" +
  newline);
  out.write("      <URL>" +
  vo.getLink() + "</URL>" + newline);
  out.write("      <FILENAME>" +
  vo.getFileDB() + "</FILENAME>" +
  newline);

  } else
  if(vo.getType().equals("Miscellaneou
  s")) {

    out.write("      <COLLECTION>" +
    vo.getCollection() +
    "</COLLECTION>" + newline);
    out.write("      <REFERENCE_TYPE>"
    + vo.getType() +
    "</REFERENCE_TYPE>" + newline);
    out.write("      <TITLE>" +
    vo.getTitle() + "</TITLE>" +
    newline);
    out.write("      <AUTHOR>" +
    vo.getAuthor() + "</AUTHOR>" +
    newline);
    out.write("      <HOWPUBLISHED>" +
    vo.getHowpublished() +
    "</HOWPUBLISHED>" + newline);
    out.write("      <MONTH>" +
    vo.getMonth() + "</MONTH>" +
    newline);
    out.write("      <YEAR>" +
    vo.getYear() + "</YEAR>" +
    newline);
    out.write("      <NOTE>" +
    vo.getNote() + "</NOTE>" +
    newline);
    out.write("      <URL>" +
    vo.getLink() + "</URL>" + newline);
    out.write("      <FILENAME>" +
    vo.getFileDB() + "</FILENAME>" +
    newline);

  }
  out.write("    </RECORD>" +
  newline);
 }

 out.write("  </RECORDS>" + newline);
 out.write("</XML>");
 out.close();

 file = new File(pathname);

 if(file != null && file.exists()) {
   is = new FileInputStream(file);
   int bit = is.read();
   while(bit > -1) {
     sos.write(bit);
     bit = is.read();
   }
 }

} catch(NumberFormatException e) {
 e.printStackTrace();
} catch (IOException e) {
 e.printStackTrace();
} finally {
 sos.flush();
 sos.close();
 is.close();
 file.delete();
}

} else if(fileType.equals("bibtex")) {

 filename =
 Utilities.formatFileName(collection,
 ".bib");
 pathname = pathname + filename;

 response.setContentType("plain/text");
 response.setHeader("Content-
 Disposition", "attachment;filename=" +
 filename);
```

```
try {
  out = new BufferedWriter(new
  FileWriter(pathname, true));

  for(int i=0; i<references.length; i++)
  {
    str = references[i];
    id = str.lastIndexOf("id");
    referenceId =
    Integer.parseInt(str.substring(id+2)
    );
    vo.setReferenceId(referenceId);
    vo =
    (ViewReferenceDetailsVO)referenceWor
    ker.readReference(vo);

    if(vo.getType().equals("Article")) {

      out.write("@article{" + newline);
      out.write("title = {" +
      vo.getTitle() + "}," + newline);
      out.write("author = {" +
      vo.getAuthor() + "}," + newline);
      out.write("journal = {" +
      vo.getJournal() + "}," + newline);
      if(!Utilities.isEmpty(vo.getVolume(
      )))
        out.write("volume = {" +
        vo.getVolume() + "}," + newline);
      if(!Utilities.isEmpty(vo.getNumber(
      )))
        out.write("number = {" +
        vo.getNumber() + "}," + newline);
      if(!Utilities.isEmpty(vo.getPages()
      ))
        out.write("pages = {" +
        vo.getPages() + "}," + newline);
      if(!Utilities.isEmpty(vo.getMonth()
      ))
        out.write("month = {" +
        vo.getMonth() + "}," + newline);
      out.write("year = {" + vo.getYear()
      + "}," + newline);
      if(!Utilities.isEmpty(vo.getNote())
      )
        out.write("note = {" +
        vo.getNote() + "}," + newline);
      if(!Utilities.isEmpty(vo.getLink())
      )
        out.write("URL = {" +
        vo.getLink() + "}," + newline);
      out.write("}" + newline);

    } else
    if(vo.getType().equals("Book")) {

      out.write("@book{" + newline);
      out.write("title = {" +
      vo.getTitle() + "}," + newline);
      out.write("author = {" +
      vo.getAuthor() + "}," + newline);
      if(!Utilities.isEmpty(vo.getVolume(
      )))
        out.write("volume = {" +
        vo.getVolume() + "}," + newline);
      if(!Utilities.isEmpty(vo.getSeries(
      )))
        out.write("series = {" +
        vo.getSeries() + "}," + newline);
      if(!Utilities.isEmpty(vo.getEdition
      ()))
        out.write("edition = {" +
        vo.getEdition() + "}," +
        newline);
      if(!Utilities.isEmpty(vo.getMonth()
      ))
        out.write("month = {" +
        vo.getMonth() + "}," + newline);
      out.write("year = {" + vo.getYear()
      + "}," + newline);
      out.write("publisher = {" +
      vo.getPublisher() + "}," +
      newline);
      if(!Utilities.isEmpty(vo.getAddress
      ()))
        out.write("address = {" +
        vo.getAddress() + "}," +
        newline);
      if(!Utilities.isEmpty(vo.getNote())
      )
        out.write("note = {" +
        vo.getNote() + "}," + newline);
      if(!Utilities.isEmpty(vo.getLink())
      )
        out.write("URL = {" +
        vo.getLink() + "}," + newline);
      out.write("}" + newline);

    } else
    if(vo.getType().equals("Inbook")) {

      out.write("@inbook{" + newline);
      out.write("title = {" +
      vo.getTitle() + "}," + newline);
      out.write("author = {" +
      vo.getAuthor() + "}," + newline);
      out.write("chapter = {" +
      vo.getChapter() + "}," + newline);
      if(!Utilities.isEmpty(vo.getVolume(
      )))
        out.write("volume = {" +
        vo.getVolume() + "}," + newline);
      if(!Utilities.isEmpty(vo.getSeries(
      )))
        out.write("series = {" +
        vo.getSeries() + "}," + newline);
      out.write("pages = {" +
      vo.getPages() + "}," + newline);
      if(!Utilities.isEmpty(vo.getSubtype
      ()))
        out.write("subtype = {" +
        vo.getSubtype() + "}," +
        newline);
      if(!Utilities.isEmpty(vo.getEdition
      ()))
        out.write("edition = {" +
        vo.getEdition() + "}," +
        newline);
      if(!Utilities.isEmpty(vo.getMonth()
      ))
        out.write("month = {" +
        vo.getMonth() + "}," + newline);
      out.write("year = {" + vo.getYear()
      + "}," + newline);
      out.write("publisher = {" +
      vo.getPublisher() + "}," +
      newline);
      if(!Utilities.isEmpty(vo.getAddress
      ()))
        out.write("address = {" +
        vo.getAddress() + "}," +
        newline);
      if(!Utilities.isEmpty(vo.getNote())
      )
        out.write("note = {" +
        vo.getNote() + "}," + newline);
      if(!Utilities.isEmpty(vo.getLink())
      )
        out.write("URL = {" +
        vo.getLink() + "}," + newline);
```

```
  out.write("}" + newline);

} else
if(vo.getType().equals("Incollection
")) {

  out.write("@incollection{" +
  newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  out.write("author = {" +
  vo.getAuthor() + "}," + newline);
  out.write("booktitle = {" +
  vo.getBooktitle() + "}," +
  newline);
  if(!Utilities.isEmpty(vo.getVolume(
  )))
    out.write("volume = {" +
    vo.getVolume() + "}," + newline);
  if(!Utilities.isEmpty(vo.getSeries(
  )))
    out.write("series = {" +
    vo.getSeries() + "}," + newline);
  if(!Utilities.isEmpty(vo.getChapter
  ()))
    out.write("chapter = {" +
    vo.getChapter() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getPages()
  ))
    out.write("pages = {" +
    vo.getPages() + "}," + newline);
  if(!Utilities.isEmpty(vo.getSubtype
  ()))
    out.write("subtype = {" +
    vo.getSubtype() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  out.write("year = {" + vo.getYear()
  + "}," + newline);
  if(!Utilities.isEmpty(vo.getEdition
  ()))
    out.write("edition = {" +
    vo.getEdition() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getEditor(
  )))
    out.write("editor = {" +
    vo.getEditor() + "}," + newline);
  out.write("publisher = {" +
  vo.getPublisher() + "}," +
  newline);
  if(!Utilities.isEmpty(vo.getAddress
  ()))
    out.write("address = {" +
    vo.getAddress() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getNote())
  )
    out.write("note = {" +
    vo.getNote() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Proceedings"
)) {

  out.write("@proceedings{" +
  newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  if(!Utilities.isEmpty(vo.getVolume(
  )))
    out.write("volume = {" +
    vo.getVolume() + "}," + newline);
  if(!Utilities.isEmpty(vo.getSeries(
  )))
    out.write("series = {" +
    vo.getSeries() + "}," + newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  out.write("year = {" + vo.getYear()
  + "}," + newline);
  if(!Utilities.isEmpty(vo.getOrganiz
  ation()))
    out.write("organization = {" +
    vo.getOrganization() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getEditor(
  )))
    out.write("editor = {" +
    vo.getEditor() + "}," + newline);
  if(!Utilities.isEmpty(vo.getPublish
  er()))
    out.write("publisher = {" +
    vo.getPublisher() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getAddress
  ()))
    out.write("address = {" +
    vo.getAddress() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getNote())
  )
    out.write("note = {" +
    vo.getNote() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Inproceeding
s")) {

  out.write("@inproceedings{" +
  newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  out.write("author = {" +
  vo.getAuthor() + "}," + newline);
  out.write("booktitle = {" +
  vo.getBooktitle() + "}," +
  newline);
  if(!Utilities.isEmpty(vo.getVolume(
  )))
    out.write("volume = {" +
    vo.getVolume() + "}," + newline);
  if(!Utilities.isEmpty(vo.getSeries(
  )))
    out.write("series = {" +
    vo.getSeries() + "}," + newline);
  if(!Utilities.isEmpty(vo.getPages()
  ))
    out.write("pages = {" +
    vo.getPages() + "}," + newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  out.write("year = {" + vo.getYear()
  + "}," + newline);
```

```
  if(!Utilities.isEmpty(vo.getOrganiz
  ation()))
    out.write("organization = {" +
    vo.getOrganization() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getEditor(
  )))
    out.write("editor = {" +
    vo.getEditor() + "}," + newline);
  if(!Utilities.isEmpty(vo.getPublish
  er()))
    out.write("publisher = {" +
    vo.getPublisher() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getAddress
  ()))
    out.write("address = {" +
    vo.getAddress() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getNote())
  )
    out.write("note = {" +
    vo.getNote() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Masters
Thesis") || vo.getType().equals("PhD
Thesis")) {

  if(vo.getType().equals("Masters
  Thesis"))
    out.write("@masters thesis{" +
    newline);
  else
  out.write("@PhD thesis{" +
  newline);
    out.write("title = {" +
    vo.getTitle() + "}," + newline);
  out.write("author = {" +
  vo.getAuthor() + "}," + newline);
  out.write("school = {" +
  vo.getSchool() + "}," + newline);
  if(!Utilities.isEmpty(vo.getSubtype
  ()))
    out.write("subtype = {" +
    vo.getSubtype() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getAddress
  ()))
    out.write("address = {" +
    vo.getAddress() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  out.write("year = {" + vo.getYear()
  + "}," + newline);
  if(!Utilities.isEmpty(vo.getNote())
  )
    out.write("note = {" +
    vo.getNote() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Technical
Report")) {

  out.write("@technical report{" +
  newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  out.write("institution = {" +
  vo.getInstitution() + "}," +
  newline);
  if(!Utilities.isEmpty(vo.getSubtype
  ()))
    out.write("subtype = {" +
    vo.getSubtype() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getNumber(
  )))
    out.write("number = {" +
    vo.getNumber() + "}," + newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  out.write("year = {" + vo.getYear()
  + "}," + newline);
  if(!Utilities.isEmpty(vo.getAddress
  ()))
    out.write("address = {" +
    vo.getAddress() + "}," +
    newline);
  if(!Utilities.isEmpty(vo.getNote())
  )
    out.write("note = {" +
    vo.getNote() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Unpublished"
)) {

  out.write("@unpublished{" +
  newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  out.write("note = {" + vo.getNote()
  + "}," + newline);
  if(!Utilities.isEmpty(vo.getMonth()
  ))
    out.write("month = {" +
    vo.getMonth() + "}," + newline);
  if(vo.getYear()!=0000)
    out.write("year = {" +
    vo.getYear() + "}," + newline);
  if(!Utilities.isEmpty(vo.getLink())
  )
    out.write("URL = {" +
    vo.getLink() + "}," + newline);
  out.write("}" + newline);

} else
if(vo.getType().equals("Manual")) {

  out.write("@manual{" + newline);
  out.write("title = {" +
  vo.getTitle() + "}," + newline);
  if(!Utilities.isEmpty(vo.getAuthor(
  )))
    out.write("author = {" +
    vo.getAuthor() + "}," + newline);
  if(!Utilities.isEmpty(vo.getOrganiz
  ation()))
    out.write("organization = {" +
    vo.getOrganization() + "}," +
    newline);
```

```
        if(!Utilities.isEmpty(vo.getAddress
        ()))
          out.write("address = {" +
          vo.getAddress() + "}," +
          newline);
        if(!Utilities.isEmpty(vo.getEdition
        ()))
          out.write("edition = {" +
          vo.getEdition() + "}," +
          newline);
        if(!Utilities.isEmpty(vo.getMonth()
        ))
          out.write("month = {" +
          vo.getMonth() + "}," + newline);
        if(vo.getYear()!=0000)
          out.write("year = {" +
          vo.getYear() + "}," + newline);
        if(!Utilities.isEmpty(vo.getNote())
        )
          out.write("note = {" +
          vo.getNote() + "}," + newline);
        if(!Utilities.isEmpty(vo.getLink())
        )
          out.write("URL = {" +
          vo.getLink() + "}," + newline);
        out.write("}" + newline);

      } else
      if(vo.getType().equals("Miscellaneou
      s")) {

        out.write("@miscellaneous{" +
        newline);
        out.write("title = {" +
        vo.getTitle() + "}," + newline);
        if(!Utilities.isEmpty(vo.getAuthor(
        )))
          out.write("author = {" +
          vo.getAuthor() + "}," + newline);
        if(!Utilities.isEmpty(vo.getHowpubl
        ished()))
          out.write("howpublished = {" +
          vo.getHowpublished() + "}," +
          newline);
        if(!Utilities.isEmpty(vo.getMonth()
        ))
          out.write("month = {" +
          vo.getMonth() + "}," + newline);
        if(vo.getYear()!=0000)
          out.write("year = {" +
          vo.getYear() + "}," + newline);
        if(!Utilities.isEmpty(vo.getNote())
        )
          out.write("note = {" +
          vo.getNote() + "}," + newline);
        if(!Utilities.isEmpty(vo.getLink())
        )
          out.write("URL = {" +
          vo.getLink() + "}," + newline);
        out.write("}" + newline);

      }
}

    out.close();

    file = new File(pathname);

    if(file != null && file.exists()) {
      is = new FileInputStream(file);
      int bit = is.read();
      while(bit > -1) {
        sos.write(bit);
        bit = is.read();
      }
    }
```

```
        } catch(NumberFormatException e) {
          e.printStackTrace();
        } catch (IOException e) {
          e.printStackTrace();
        } finally {
          sos.flush();
          sos.close();
          is.close();
          file.delete();
        }

      }
    }

  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Cancel")
  ) {
    forward =
    "virholex.references.viewReferences";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("collection",
    collection);
    return redirect;
  }

  PropertyUtils.copyProperties(collectionForm,
  collectionVO);

  return mapping.findForward(forward);
}

}
```

ViewCollectionDetailsForm.java

```
package com.virholex.references;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;

import com.virholex.general.Utilities;

public class ViewCollectionDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  4577093286852466949L;
  private String collectionName;
  private String oldCollectionName;
  private String description;
  private String type;
  private String author;
  private String downloadable;
  private String command;
  private ArrayList<String> letters;
  private ArrayList<ViewReferenceDetailsVO>
  references;
  private int privilege;
  private boolean hasCollection;
  private String dateAdded;

  public void setCollectionName(String
  collectionName) {
    this.collectionName = collectionName;
  }
```

```
public String getCollectionName() {
  return collectionName;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setLetters(ArrayList<String>
letters) {
  this.letters = letters;
}
public ArrayList<String> getLetters() {
  return letters;
}
public void
setReferences(ArrayList<ViewReferenceDetails
VO> references) {
  this.references = references;
}
public ArrayList<ViewReferenceDetailsVO>
getReferences() {
  return references;
}
public void setPrivilege(int privilege) {
  this.privilege = privilege;
}
public int getPrivilege() {
  return privilege;
}
public void setType(String type) {
  this.type = type;
}
public String getType() {
  return type;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setHasCollection(boolean
hasCollection) {
  this.hasCollection = hasCollection;
}
public boolean isHasCollection() {
  return hasCollection;
}
public void setDownloadable(String
downloadable) {
  this.downloadable = downloadable;
}
public String getDownloadable() {
  return downloadable;
}
public void setOldCollectionName(String
oldCollectionName) {
  this.oldCollectionName = oldCollectionName;
}
public String getOldCollectionName() {
  return oldCollectionName;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
  return dateAdded;
  }

  public ActionErrors validate(ActionMapping
  mapping, HttpServletRequest request) {

    ActionErrors errors = new ActionErrors();

    if
    (Utilities.isEmpty(this.getCollectionName()
    ) ||
    Utilities.isEmpty(this.getDownloadable()))
    {
      errors.add("errorMessage", new
      ActionMessage("error.required.asterisk"))
      ;

      if(Utilities.isEmpty(this.getCollectionNa
      me()))
        errors.add("collectionName", new
        ActionMessage(""));
      if(Utilities.isEmpty(this.getDownloadable
      ()))
        errors.add("downloadable", new
        ActionMessage(""));
    }

    if(this.isHasCollection()) {
      errors.add("collectionName", new
      ActionMessage("error.notAvailable", "
      name"));
    }

    return errors;
  }

}
```

ViewCollectionDetailsVO.java

```
package com.virholex.references;

import java.util.ArrayList;

import com.virholex.general.VO;

public class ViewCollectionDetailsVO
implements VO {

  private static final long serialVersionUID =
  2562435154176934781L;
  private String collectionName;
  private String oldCollectionName;
  private String description;
  private String type;
  private String author;
  private String downloadable;
  private String command;
  private ArrayList<String> letters;
  private ArrayList<ViewReferenceDetailsVO>
  references;
  private int privilege;
  private String dateAdded;

  public void setCollectionName(String
  collectionName) {
    this.collectionName = collectionName;
  }
  public String getCollectionName() {
    return collectionName;
  }
  public void setDescription(String
  description) {
    this.description = description;
  }
```

```
  public String getDescription() {
    return description;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setLetters(ArrayList<String>
  letters) {
    this.letters = letters;
  }
  public ArrayList<String> getLetters() {
    return letters;
  }
  public void
  setReferences(ArrayList<ViewReferenceDetails
  VO> references) {
    this.references = references;
  }
  public ArrayList<ViewReferenceDetailsVO>
  getReferences() {
    return references;
  }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
    return privilege;
  }
  public void setType(String type) {
    this.type = type;
  }
  public String getType() {
    return type;
  }
  public void setAuthor(String author) {
    this.author = author;
  }
  public String getAuthor() {
    return author;
  }
  public void setDownloadable(String
  downloadable) {
    this.downloadable = downloadable;
  }
  public String getDownloadable() {
    return downloadable;
  }
  public void setOldCollectionName(String
  oldCollectionName) {
    this.oldCollectionName = oldCollectionName;
  }
  public String getOldCollectionName() {
    return oldCollectionName;
  }
  public void setDateAdded(String dateAdded) {
    this.dateAdded = dateAdded;
  }
  public String getDateAdded() {
    return dateAdded;
  }

}
```

ViewCollectionsAction.java

```
package com.virholex.references;

import java.util.ArrayList;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import
org.apache.commons.beanutils.PropertyUtils;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.virholex.general.VirholexConstants;

public class ViewCollectionsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewCollectionDetailsForm collectionForm =
  (ViewCollectionDetailsForm) form;
  ViewCollectionDetailsVO collectionVO = new
  ViewCollectionDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<ViewCollectionDetailsVO>
  collectionList = new
  ArrayList<ViewCollectionDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.references.viewCollections";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", roleName = "", project =
  "";
  boolean isLead = false;

  if(collectionForm.getCommand() != null &&
  collectionForm.getCommand().equals("Add
  Collection")) {
    forward =
    "virholex.references.addCollection";
  } else {
    for(int i=0; i < letters.length() ; i++){
      if(referenceWorker.getCollections(collect
      ionVO,
      Character.toString(letters.charAt(i))).si
      ze() > 0)
        letterList.add(Character.toString(lette
        rs.charAt(i)));
      else
        letterList.add("-");
    }

    collectionVO.setLetters(letterList);

    if(request.getParameter("list") != null)
      list = request.getParameter("list");

    collectionList =
    referenceWorker.getCollections(collectionVO
    , list);
      isLead =
      referenceWorker.isCollectionCoordinator(u
      sername);

    for(int i=0; i< collectionList.size(); i++)
    {
      project =
      ((ViewCollectionDetailsVO)collectionList.
```

```
    get(i)).getCollectionName() + " " +
    VirholexConstants.VIRHO_REFERENCE;
    project = project.substring(0,
    project.lastIndexOf(VirholexConstants.VIR
    HO_REFERENCE)-1);
    roleName =
    referenceWorker.getProjectRole(username,
    project);

    if(roleName.equals(VirholexConstants.COLL
    ECTION_COORDINATOR_DESC))
      ((ViewCollectionDetailsVO)collectionLis
      t.get(i)).setPrivilege(VirholexConstant
      s.COLLECTION_COORDINATOR);
    else
    if(roleName.equals(VirholexConstants.COLL
    ECTION_CONTRIBUTOR_DESC))
      ((ViewCollectionDetailsVO)collectionLis
      t.get(i)).setPrivilege(VirholexConstant
      s.COLLECTION_CONTRIBUTOR);
    else
    if(roleName.equals(VirholexConstants.REST
    RICTED_USER_DESC))
      ((ViewCollectionDetailsVO)collectionLis
      t.get(i)).setPrivilege(VirholexConstant
      s.RESTRICTED_USER);
    else
      ((ViewCollectionDetailsVO)collectionLis
      t.get(i)).setPrivilege(VirholexConstant
      s.REGISTERED_USER);
   }

   request.setAttribute("isLead", isLead);
  }

  PropertyUtils.copyProperties(collectionForm,
  collectionVO);

  session.setAttribute("module",
  VirholexConstants.VIRHO_REFERENCES);
  request.setAttribute("collections",
  collectionList);

  return mapping.findForward(forward);
}
}
```

ViewCollectionsDAO.java

```
package com.virholex.references;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.VirholexConstants;

public class ViewCollectionsDAO {

  public static ViewCollectionDetailsVO
  readCollection(ViewCollectionDetailsVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {

      String sql = "SELECT title, description,
      downloadable, date_added FROM collections
      WHERE title=? AND title LIKE ?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1,
      vo.getOldCollectionName());
      ps.setString(2,
      vo.getOldCollectionName());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setCollectionName(rs.getString(1));
        vo.setDescription(rs.getString(2));
        vo.setDownloadable(rs.getString(3));
        vo.setDateAdded(rs.getString(4));
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }

  public static
  ArrayList<ViewCollectionDetailsVO>
  getCollections(ViewCollectionDetailsVO vo,
  String letter) throws SQLException,
  NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ArrayList<ViewCollectionDetailsVO>
    collectionList = new
    ArrayList<ViewCollectionDetailsVO>();

    try {
      String sql = "SELECT title, description,
      downloadable, date_added FROM collections
      WHERE title LIKE ? ORDER BY title,
      description";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, letter+"%");
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setCollectionName(rs.getString(1));
        vo.setDescription(rs.getString(2));
        vo.setDownloadable(rs.getString(3));
        vo.setDateAdded(rs.getString(4));
        collectionList.add(vo);
        vo = new ViewCollectionDetailsVO();
      }

    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return collectionList;
  }

  public static String getProjectRole(String
  username, String project) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    String role = "";
```

```
  try {
    String sql = "SELECT role_name FROM
    user_prev WHERE username=? AND
    involvement=? AND (role_name LIKE
    '%Collection%' OR role_name='Restricted
    User')";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, project+"
    "+VirholexConstants.VIRHO_REFERENCE);
    rs = ps.executeQuery();

    if(rs.next()) {
      role = rs.getString(1);
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return role;
}

public static boolean
isCollectionCoordinator(String username)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean isLead = false;

  try {
    String sql = "SELECT role_name FROM
    user_prev WHERE username=? AND
    involvement='DEFAULT_PROJECT' AND
    role_name='Collection Coordinator'";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    rs = ps.executeQuery();

    if(rs.next()) {
      isLead = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isLead;
}

public static void
deleteCollection(ViewCollectionDetailsVO vo)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  int referenceId = 0;

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE FROM collections
    WHERE title=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getCollectionName());
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM project WHERE
    project_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getCollectionName()+"
    "+VirholexConstants.VIRHO_REFERENCE);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM user_prev WHERE
    involvement=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getCollectionName()+"
    "+VirholexConstants.VIRHO_REFERENCE);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM coll_reference WHERE
    id=?";

    for(int i=0; i<vo.getReferences().size();
    i++) {
      referenceId =
      ((ViewReferenceDetailsVO)vo.getReferenc
      es().get(i)).getReferenceId();

      ps = conn.prepareStatement(sql);
      ps.setInt(1, referenceId);
      ps.executeUpdate();
      ps.close();
    }

   } finally {
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }
  }

}
```

ViewReferenceDetailsAction.java

```
package com.virholex.references;

import javax.servlet.http.HttpServletRequest;

public class ViewReferenceDetailsAction
extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewReferenceDetailsForm referenceForm =
  (ViewReferenceDetailsForm) form;
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.references.viewReferenceDetails";
  String collection =
  (String)session.getAttribute("collection");
  String reference = "";
  String link = "", pdf = "", roleName = "";
  int referenceId = 0;
```

```
PropertyUtils.copyProperties(referenceVO,
referenceForm);
referenceVO.setCollection(collection);

if(request.getParameter("module")!=null)
  session.setAttribute("module",
  VirholexConstants.VIRHO_REFERENCES);
if(request.getParameter("collection")!=null)
{
  collection =
  request.getParameter("collection");
  session.setAttribute("collection",
  collection);
}
if(request.getParameter("reference")!=null)
  reference =
  request.getParameter("reference");
else
  reference =
  (String)session.getAttribute("reference");
if(request.getParameter("refId")!=null)
  referenceId =
  Integer.parseInt(request.getParameter("refI
  d"));

if(referenceForm.getCommand() != null &&
referenceForm.getCommand().equals("Add
Reference")) {
  forward =
  "virholex.references.addReference";
} else if(referenceForm.getCommand() != null
&& referenceForm.getCommand().equals("Edit
Reference")) {
  forward =
  "virholex.references.editReference";
} else if(referenceForm.getCommand() != null
&& referenceForm.getCommand().equals("Delete
Reference")) {
  referenceVO.setTitle(reference);
  referenceVO = (ViewReferenceDetailsVO)
  referenceWorker.readReference(referenceVO);

  if(referenceWorker.checkReference(reference
  VO.getReferenceId())) {
    if(referenceVO.getFileDB()!=null) {
      // delete the files from the repository
      Utilities.deleteFile(request,
      VirholexConstants.REFERENCES_REPOSITORY
      , referenceVO.getFileDB());
    }
    // delete the reference
    referenceWorker.deleteReference(reference
    VO.getReferenceId());
  }

  forward =
  "virholex.references.viewReferences";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forward)
  );
  redirect.addParameter("collection",
  collection);
  return redirect;

} else if(referenceForm.getCommand() != null
&& referenceForm.getCommand().equals("Send
Membership Request")) {
  session.setAttribute("project",
  collection);
  forward =
  "virholex.virus.privilegeRequest";
} else {
  referenceVO.setTitle(reference);
  referenceVO.setReferenceId(referenceId);
  referenceVO = (ViewReferenceDetailsVO)
  referenceWorker.readReference(referenceVO);
    // Remove application protocol
    link = referenceVO.getLink();
    referenceVO.setLink(Utilities.removeProtoco
    l(link));
    // Remove underscore and set of random
    numbers from PDF file
    pdf = referenceVO.getFileDB();
    referenceVO.setFile(Utilities.removeFileExt
    ension(pdf, ".pdf")); // path

    roleName =
    referenceWorker.getProjectRole(username,
    collection);

    if(roleName.equals(VirholexConstants.COLLEC
    TION_COORDINATOR_DESC))
      referenceVO.setPrivilege(VirholexConstant
      s.COLLECTION_COORDINATOR);
    else
    if(roleName.equals(VirholexConstants.COLLEC
    TION_CONTRIBUTOR_DESC))
      referenceVO.setPrivilege(VirholexConstant
      s.COLLECTION_CONTRIBUTOR);
    else
    if(roleName.equals(VirholexConstants.RESTRI
    CTED_USER_DESC))
      referenceVO.setPrivilege(VirholexConstant
      s.RESTRICTED_USER);
    else
      referenceVO.setPrivilege(VirholexConstant
      s.REGISTERED_USER);

  }

  PropertyUtils.copyProperties(referenceForm,
  referenceVO);

  session.setAttribute("reference",
  reference);

  return mapping.findForward(forward);
}
}
```

ViewReferenceDetailsForm.java

```
package com.virholex.references;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;
import com.virholex.general.VirholexConstants;

public class ViewReferenceDetailsForm extends
ActionForm {

  private static final long serialVersionUID =
  -2462555334252961991L;
  private String title;
  private String author;
  private String collection;
  private String publisher;
  private String chapter;
  private String pages;
  private String booktitle;
  private String school;
  private String institution;
  private String note;
```

```
private String volume;
private String series;
private String address;
private String subtype;
private String edition;
private String month;
private String type;
private String editor;
private String organization;
private String number;
private String howpublished;
private int year;
private String journal;
private String link;
private String file;
private FormFile fileName;
private String fileDB;
private String command;
private boolean downloadable;
private int privilege;
private boolean hasReference;
private int referenceId;

public void setTitle(String title) {
  this.title = title;
}
public String getTitle() {
  return title;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setCollection(String collection)
{
  this.collection = collection;
}
public String getCollection() {
  return collection;
}
public void setPublisher(String publisher) {
  this.publisher = publisher;
}
public String getPublisher() {
  return publisher;
}
public void setChapter(String chapter) {
  this.chapter = chapter;
}
public String getChapter() {
  return chapter;
}
public void setPages(String pages) {
  this.pages = pages;
}
public String getPages() {
  return pages;
}
public void setBooktitle(String booktitle) {
  this.booktitle = booktitle;
}
public String getBooktitle() {
  return booktitle;
}
public void setSchool(String school) {
  this.school = school;
}
public String getSchool() {
  return school;
}
public void setInstitution(String
institution) {
  this.institution = institution;
}
public String getInstitution() {
  return institution;
}
public void setNote(String note) {
  this.note = note;
}
public String getNote() {
  return note;
}
public void setVolume(String volume) {
  this.volume = volume;
}
public String getVolume() {
  return volume;
}
public void setSeries(String series) {
  this.series = series;
}
public String getSeries() {
  return series;
}
public void setAddress(String address) {
  this.address = address;
}
public String getAddress() {
return address;
  }
public void setEdition(String edition) {
  this.edition = edition;
}
public String getEdition() {
  return edition;
}
public void setMonth(String month) {
  this.month = month;
}
public String getMonth() {
  return month;
}
public void setType(String type) {
  this.type = type;
}
public String getType() {
  return type;
}
public void setEditor(String editor) {
  this.editor = editor;
}
public String getEditor() {
  return editor;
}
public void setOrganization(String
organization) {
  this.organization = organization;
}
public String getOrganization() {
  return organization;
}
public void setNumber(String number) {
  this.number = number;
}
public String getNumber() {
  return number;
}
public void setYear(int year) {
  this.year = year;
}
public int getYear() {
  return year;
}
public void setJournal(String journal) {
  this.journal = journal;
}
public String getJournal() {
  return journal;
}
```

```
public void setLink(String link) {
  this.link = link;
}
public String getLink() {
  return link;
}
public void setFile(String file) {
  this.file = file;
}
public String getFile() {
  return file;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setDownloadable(boolean
downloadable) {
  this.downloadable = downloadable;
}
public boolean isDownloadable() {
  return downloadable;
}
public void setHowpublished(String
howpublished) {
  this.howpublished = howpublished;
}
public String getHowpublished() {
  return howpublished;
}
public void setSubtype(String subtype) {
  this.subtype = subtype;
}
public String getSubtype() {
  return subtype;
}
public void setPrivilege(int privilege) {
  this.privilege = privilege;
}
public int getPrivilege() {
  return privilege;
}
public void setHasReference(boolean
hasReference) {
  this.hasReference = hasReference;
}
public boolean isHasReference() {
  return hasReference;
}
public void setFileDB(String fileDB) {
  this.fileDB = fileDB;
}
public String getFileDB() {
  return fileDB;
}
public void setFileName(FormFile fileName) {
  this.fileName = fileName;
}
public FormFile getFileName() {
  return fileName;
}
public void setReferenceId(int referenceId)
{
  this.referenceId = referenceId;
}
public int getReferenceId() {
  return referenceId;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if (Utilities.isEmpty(this.getTitle()) ||
  this.getType().equals(VirholexConstants.DEF
  AULT_VALUE) ||
  (this.getType().equals("Book") &&
  (Utilities.isEmpty(this.author) ||
  Utilities.isEmpty(this.getPublisher()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Inbook") &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getPublisher()) ||
  Utilities.isEmpty(this.getChapter()) ||
  Utilities.isEmpty(this.getPages()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Incollection") &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getPublisher()) ||
  Utilities.isEmpty(this.getBooktitle()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Inproceedings") &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getBooktitle()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Article") &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getJournal()) ||
  this.getYear()==0000)) ||
  ((this.getType().equals("Masters Thesis")
  || this.getType().equals("PhD Thesis")) &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getSchool()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Technical Report")
  && (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getInstitution()) ||
  this.getYear()==0000)) ||
  (this.getType().equals("Unpublished") &&
  (Utilities.isEmpty(this.getAuthor()) ||
  Utilities.isEmpty(this.getNote()))) ||
  (this.getType().equals("Proceedings") &&
  (this.getYear()==0000))) {

    errors.add("errorMessage", new
    ActionMessage("error.required.asterisk"))
    ;

    if(Utilities.isEmpty(this.getTitle()))
      errors.add("title", new
      ActionMessage(""));
    if(this.getType().equals(VirholexConstant
    s.DEFAULT_VALUE))
      errors.add("type", new
      ActionMessage(""));
    if(!(this.getType().equals(VirholexConsta
    nts.DEFAULT_VALUE) ||
    this.getType().equals("Manual") ||
    this.getType().equals("Proceedings") ||
    this.getType().equals("Miscellaneous")))
    {
      if(Utilities.isEmpty(this.getAuthor()))
        errors.add("author", new
        ActionMessage(""));
    }
    if(this.getType().equals("Book") ||
    this.getType().equals("Inbook") ||
    this.getType().equals("Incollection")) {
      if(Utilities.isEmpty(this.getPublisher(
      )))
        errors.add("publisher", new
        ActionMessage(""));
    }
    if(!(this.getType().equals(VirholexConsta
    nts.DEFAULT_VALUE) ||
    this.getType().equals("Manual") ||
    this.getType().equals("Unpublished") ||
    this.getType().equals("Miscellaneous")))
    {
```

```
      if(this.getYear()==0000)
        errors.add("year", new
        ActionMessage(""));
    }
    if(this.getType().equals("Inbook")) {
      if(Utilities.isEmpty(this.getChapter())
      )
        errors.add("chapter", new
        ActionMessage(""));
    }
    if(this.getType().equals("Inbook")) {
      if(Utilities.isEmpty(this.getPages()))
        errors.add("pages", new
        ActionMessage(""));
    }
    if(this.getType().equals("Article")) {
      if(Utilities.isEmpty(this.getJournal())
      )
        errors.add("journal", new
        ActionMessage(""));
    }
    if(this.getType().equals("Inproceedings")
    || this.getType().equals("Incollection"))
    {
      if(Utilities.isEmpty(this.getBooktitle(
      )))
        errors.add("booktitle", new
        ActionMessage(""));
    }
    if(this.getType().equals("Masters
    Thesis") || this.getType().equals("PhD
    Thesis")) {
      if(Utilities.isEmpty(this.getSchool()))
        errors.add("school", new
        ActionMessage(""));
    }
    if(this.getType().equals("Technical
    Report")) {
      if(Utilities.isEmpty(this.getInstitutio
      n()))
        errors.add("institution", new
        ActionMessage(""));
    }
    if(this.getType().equals("Unpublished"))
    {
      if(Utilities.isEmpty(this.getNote()))
        errors.add("note", new
        ActionMessage(""));
    }
   }

   return errors;
  }

}
```

ViewReferenceDetailsVO.java

```
package com.virholex.references;

import org.apache.struts.upload.FormFile;

import com.virholex.general.VO;

public class ViewReferenceDetailsVO implements
VO {

  private static final long serialVersionUID =
  -7715544719413080939L;
  private String title;
  private String author;
  private String collection;
  private String publisher;
  private String chapter;
  private String pages;
  private String booktitle;
  private String school;
  private String institution;
  private String note;
  private String volume;
  private String series;
  private String address;
  private String edition;
  private String month;
  private String type;
  private String editor;
  private String subtype;
  private String organization;
  private String number;
  private String howpublished;
  private int year;
  private String journal;
  private String link;
  private String file;
  private FormFile fileName;
  private String fileDB;
  private boolean downloadable;
  private int privilege;
  private int referenceId;

  public void setTitle(String title) {
    this.title = title;
  }
  public String getTitle() {
    return title;
  }
  public void setAuthor(String author) {
    this.author = author;
  }
  public String getAuthor() {
    return author;
  }
  public void setCollection(String collection)
  {
    this.collection = collection;
  }
  public String getCollection() {
    return collection;
  }
  public void setPublisher(String publisher) {
    this.publisher = publisher;
  }
  public String getPublisher() {
    return publisher;
  }
  public void setChapter(String chapter) {
    this.chapter = chapter;
  }
  public String getChapter() {
    return chapter;
  }
  public void setPages(String pages) {
    this.pages = pages;
  }
  public String getPages() {
    return pages;
  }
  public void setBooktitle(String booktitle) {
    this.booktitle = booktitle;
  }
  public String getBooktitle() {
    return booktitle;
  }
  public void setSchool(String school) {
    this.school = school;
  }
  public String getSchool() {
    return school;
  }
```

```
public void setInstitution(String
institution) {
  this.institution = institution;
}
public String getInstitution() {
  return institution;
}
public void setNote(String note) {
  this.note = note;
}
public String getNote() {
  return note;
}
public void setVolume(String volume) {
  this.volume = volume;
}
public String getVolume() {
  return volume;
}
public void setSeries(String series) {
  this.series = series;
}
public String getSeries() {
  return series;
}
public void setAddress(String address) {
  this.address = address;
}
public String getAddress() {
  return address;
}
public void setEdition(String edition) {
  this.edition = edition;
}
public String getEdition() {
  return edition;
}
public void setMonth(String month) {
  this.month = month;
}
public String getMonth() {
  return month;
}
public void setType(String type) {
  this.type = type;
}
public String getType() {
  return type;
}
public void setEditor(String editor) {
  this.editor = editor;
}
public String getEditor() {
  return editor;
}
public void setOrganization(String
organization) {
  this.organization = organization;
}
public String getOrganization() {
  return organization;
}
public void setNumber(String number) {
  this.number = number;
}
public String getNumber() {
  return number;
}
public void setYear(int year) {
  this.year = year;
}
public int getYear() {
  return year;
}
public void setJournal(String journal) {
this.journal = journal;
  }
  public String getJournal() {
    return journal;
  }
  public void setLink(String link) {
  this.link = link;
    }
  public String getLink() {
  return link;
    }
  public void setFile(String file) {
    this.file = file;
  }
  public String getFile() {
    return file;
  }
  public void setDownloadable(boolean
  downloadable) {
    this.downloadable = downloadable;
  }
  public boolean isDownloadable() {
  return downloadable;
    }
  public void setHowpublished(String
  howpublished) {
    this.howpublished = howpublished;
  }
  public String getHowpublished() {
    return howpublished;
  }
  public void setSubtype(String subtype) {
    this.subtype = subtype;
  }
  public String getSubtype() {
  return subtype;
    }
  public void setPrivilege(int privilege) {
    this.privilege = privilege;
  }
  public int getPrivilege() {
  return privilege;
    }
  public void setFileName(FormFile fileName) {
    this.fileName = fileName;
  }
  public FormFile getFileName() {
    return fileName;
  }
  public void setFileDB(String fileDB) {
    this.fileDB = fileDB;
  }
  public String getFileDB() {
    return fileDB;
  }
  public void setReferenceId(int referenceId)
  {
    this.referenceId = referenceId;
  }
  public int getReferenceId() {
    return referenceId;
  }

}
```

ViewReferencesAction.java

```
package com.virholex.references;

import java.util.ArrayList;

public class ViewReferencesAction extends
Action {

@Override
```

```
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewCollectionDetailsForm collectionForm =
  (ViewCollectionDetailsForm) form;
  ViewCollectionDetailsVO collectionVO = new
  ViewCollectionDetailsVO();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewReferenceDetailsVO>
  referencesList = new
  ArrayList<ViewReferenceDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.references.viewReferences";
  String collection = "";
  String roleName  = "";

  if(request.getParameter("module")!=null)
    session.setAttribute("module",
    VirholexConstants.VIRHO_REFERENCES);
  if(request.getParameter("collection") !=
  null)
    collection =
    request.getParameter("collection");
  else
    collection =
    (String)session.getAttribute("collection");

  if(collectionForm.getCommand() != null &&
  collectionForm.getCommand().equals("Add
  Collection")) {
    forward =
    "virholex.references.addCollection";
  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Edit
  Collection")) {
    forward =
    "virholex.references.editCollection";
  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Delete
  Collection")) {
    collectionVO.setCollectionName(collection);
    collectionVO.setReferences(referenceWorker.
    getReferences(collectionVO));
    referencesList =
    collectionVO.getReferences();

    for(int i=0;
    i<collectionVO.getReferences().size(); i++)
    {
      if(referenceWorker.checkReference(((ViewR
      eferenceDetailsVO)referencesList.get(i)).
      getReferenceId())) {
        if(((ViewReferenceDetailsVO)referencesL
        ist.get(i)).getFileDB()!=null) {
          // delete the files from the
          repository
          Utilities.deleteFile(request,
          VirholexConstants.REFERENCES_REPOSITOR
          Y,
          ((ViewReferenceDetailsVO)referencesLis
          t.get(i)).getFileDB());
        }
      }
    }

    referenceWorker.deleteCollection(collection
    VO);
    forward =
    "virholex.references.viewCollections";
  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Add
  Reference")) {
    forward =
    "virholex.references.addReference";
  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("View
  Members")) {
    forward = "virholex.references.viewUsers";
  } else if(collectionForm.getCommand() !=
  null &&
  collectionForm.getCommand().equals("Send
  Membership Request")) {
    session.setAttribute("project",
    collection);
    forward =
    "virholex.virus.privilegeRequest";
  } else {
    collectionVO.setCollectionName(collection);
    collectionVO.setOldCollectionName(collectio
    n);
    collectionVO = (ViewCollectionDetailsVO)
    referenceWorker.readCollection(collectionVO
    );
    collectionVO.setDateAdded(Utilities.dateFor
    mat(collectionVO.getDateAdded()));
    collectionVO.setReferences(referenceWorker.
    getReferences(collectionVO));
    roleName =
    referenceWorker.getProjectRole(username,
    collection);

    if(roleName.equals(VirholexConstants.COLLEC
    TION_COORDINATOR_DESC))
      collectionVO.setPrivilege(VirholexConstan
      ts.COLLECTION_COORDINATOR);
    else
    if(roleName.equals(VirholexConstants.COLLEC
    TION_CONTRIBUTOR_DESC))
      collectionVO.setPrivilege(VirholexConstan
      ts.COLLECTION_CONTRIBUTOR);
    else
    if(roleName.equals(VirholexConstants.RESTRI
    CTED_USER_DESC))
      collectionVO.setPrivilege(VirholexConstan
      ts.RESTRICTED_USER);
    else
      collectionVO.setPrivilege(VirholexConstan
      ts.REGISTERED_USER);

    if(collectionForm.getCommand()!=null &&
    collectionForm.getCommand().equals("Export"
    )) {
      ViewReferenceDetailsVO vo = new
      ViewReferenceDetailsVO();
      ArrayList<ViewReferenceDetailsVO> refList
      = new
      ArrayList<ViewReferenceDetailsVO>();
      String[] references =
      request.getParameterValues("referencesLis
      t");
      String str = "";
      int ref = 0, id = 0;

      try {
        for(int i=0; i<references.length; i++)
        {
          str = getReference(request,
          Integer.parseInt(references[i]));
          if(str!=null) {
            ref = str.indexOf("ref");
```

```
          id = str.lastIndexOf("id");
          vo.setTitle(str.substring(ref+3,
          id));
          vo.setReferenceId(Integer.parseInt(s
          tr.substring(id+2)));
          refList.add(vo);
          vo = new ViewReferenceDetailsVO();
        }
      }
    } catch(NumberFormatException e) {
      e.printStackTrace();
    }

    if(!refList.isEmpty()) {
      collectionVO.setReferences(refList);
      session.setAttribute("collectionVO",
      collectionVO);
      session.setMaxInactiveInterval(60*5);
      forward =
      "virholex.references.exportReferences";
      return new
      ActionRedirect(mapping.findForward(forw
      ard));
    }
  }
 }

 PropertyUtils.copyProperties(collectionForm,
 collectionVO);

 session.setAttribute("collection",
 collection);

 return mapping.findForward(forward);
}

public String getReference(HttpServletRequest
request, int referenceId) {
 return
 request.getParameter("id"+referenceId);
}

}
```

ViewReferencesDAO.java

```
package com.virholex.references;

import java.sql.Connection;

public class ViewReferencesDAO {

 public static
 ArrayList<ViewReferenceDetailsVO>
 getReferences(ViewCollectionDetailsVO vo)
 throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewReferenceDetailsVO referenceVO = new
  ViewReferenceDetailsVO();
  ArrayList<ViewReferenceDetailsVO>
  referenceList = new
  ArrayList<ViewReferenceDetailsVO>();

  try {
    String sql = "SELECT title, author, year,
    type, id, path FROM coll_reference WHERE
    collection=? ORDER BY title";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getCollectionName());
    rs = ps.executeQuery();

    while(rs.next()) {
      referenceVO.setTitle(rs.getString(1));
      referenceVO.setAuthor(rs.getString(2));
      referenceVO.setYear(rs.getInt(3));
      referenceVO.setType(rs.getString(4));
      referenceVO.setReferenceId(rs.getInt(5)
      );
      referenceVO.setFileDB(rs.getString(6));
      referenceList.add(referenceVO);
      referenceVO = new
      ViewReferenceDetailsVO();
    }

  } catch(Exception e) {
    e.printStackTrace();
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return referenceList;
}

public static
ArrayList<ViewReferenceDetailsVO>
getReferences() throws SQLException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewReferenceDetailsVO vo = new
  ViewReferenceDetailsVO();
  ArrayList<ViewReferenceDetailsVO>
  referenceList = new
  ArrayList<ViewReferenceDetailsVO>();

  try {
    String sql = "SELECT title, author, type,
    collection, id FROM coll_reference ORDER
    BY title";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setTitle(rs.getString(1));
      vo.setAuthor(rs.getString(2));
      vo.setType(rs.getString(3));
      vo.setCollection(rs.getString(4));
      vo.setReferenceId(rs.getInt(5));
      referenceList.add(vo);
      vo = new ViewReferenceDetailsVO();
    }

  } catch(Exception e) {
    e.printStackTrace();
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return referenceList;
}

public static ViewReferenceDetailsVO
getReferenceDetails(ViewReferenceDetailsVO
vo) throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;

  try {
```

```
    String sql = "SELECT * FROM
    coll_reference WHERE id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, vo.getReferenceId());
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setTitle(rs.getString("title"));
      vo.setAuthor(rs.getString("author"));
      vo.setCollection(rs.getString("collecti
      on"));
      vo.setPublisher(rs.getString("publisher
      "));
      vo.setYear(rs.getInt("year"));
      vo.setChapter(rs.getString("chapter"));
      vo.setPages(rs.getString("pages"));
      vo.setBooktitle(rs.getString("booktitle
      "));
      vo.setSchool(rs.getString("school"));
      vo.setInstitution(rs.getString("institu
      tion"));
      vo.setNote(rs.getString("note"));
      vo.setVolume(rs.getString("volume"));
      vo.setSeries(rs.getString("series"));
      vo.setAddress(rs.getString("address"));
      vo.setEdition(rs.getString("edition"));
      vo.setMonth(rs.getString("month"));
      vo.setType(rs.getString("type"));
      vo.setEditor(rs.getString("editor"));
      vo.setOrganization(rs.getString("organi
      zation"));
      vo.setNumber(rs.getString("number"));
      vo.setHowpublished(rs.getString("howpub
      lished"));
      vo.setSubtype(rs.getString("subtype"));
      vo.setJournal(rs.getString("journal"));
      vo.setLink(rs.getString("link"));
      vo.setFileDB(rs.getString("path"));
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return vo;
}

public static boolean checkReference(int
referenceId) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean hasReference = false;

  try {
    String sql = "Select title FROM
    coll_reference WHERE id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, referenceId);
    rs = ps.executeQuery();

    if(rs.next()) {
      hasReference = true;
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return hasReference;
}

public static void deleteReference(int
referenceId) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    String sql = "DELETE FROM coll_reference
    WHERE id=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setInt(1, referenceId);
    ps.executeUpdate();
  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

}
```

ViewUsersAction.java

```
package com.virholex.references;

import java.util.ArrayList;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  VirhoReferencesWorker referenceWorker = new
  VirhoReferencesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<String> rolesList = new
  ArrayList<String>();
  ArrayList<String> usersList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String collection =
  (String)session.getAttribute("collection");
  String forward =
  "virholex.references.viewUsers";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", roleName = "";
  int privilege = 0;

  if(!Utilities.isEmpty(request.getParameter("
  list")))
    list= request.getParameter("list");

  for(int i=0; i < letters.length() ; i++){
    if(virusWorker.getUsers(collection,
    Character.toString(letters.charAt(i)),
```

```
  VirholexConstants.VIRHO_REFERENCES).size()
  > 0)
    letterList.add(Character.toString(letters
    .charAt(i)));
  else
    letterList.add("-");
}

usersForm.setLetters(letterList);

if(usersForm.getCommand()!=null &&
usersForm.getCommand().equals("Change
Privilege Role")) {

  String[] roles =
  request.getParameterValues("role");
  String[] oldRoles =
  request.getParameterValues("oldRole");
  String[] users =
  request.getParameterValues("userName");

  if(roles!=null && oldRoles!=null &&
  users!=null) {
    for(int i=0; i<oldRoles.length; i++) {
      if(!roles[i].equals(oldRoles[i])) {
        rolesList.add(roles[i]);
        usersList.add(users[i]);
      }
    }

    if(!rolesList.isEmpty() &&
    !usersList.isEmpty()) {
      virusWorker.updateProjectPrivilege(user
      sList, rolesList, collection,
      VirholexConstants.VIRHO_REFERENCES);
      rolesList = new ArrayList<String>();
    }
  }
}

userVO.setUsers(virusWorker.getUsers(collect
ion, list,
VirholexConstants.VIRHO_REFERENCES));
rolesList =
virusWorker.getUserRoles(VirholexConstants.V
IRHO_REFERENCES);
roleName =
referenceWorker.getProjectRole(username,
collection);

if(roleName.equals(VirholexConstants.COLLECT
ION_COORDINATOR_DESC))
  privilege =
  VirholexConstants.COLLECTION_COORDINATOR;
else
if(roleName.equals(VirholexConstants.COLLECT
ION_CONTRIBUTOR_DESC))
  privilege =
  VirholexConstants.COLLECTION_CONTRIBUTOR;
else
if(roleName.equals(VirholexConstants.RESTRIC
TED_USER_DESC))
  privilege =
  VirholexConstants.RESTRICTED_USER;
else
  privilege =
  VirholexConstants.REGISTERED_USER;

PropertyUtils.copyProperties(usersForm,
userVO);

request.setAttribute("privilege",
privilege);
request.setAttribute("roles", rolesList);

return mapping.findForward(forward);
}
}
```

VirhoReferencesWorker.java

```
package com.virholex.references;

import java.util.ArrayList;

public class VirhoReferencesWorker implements
Worker {

  public ArrayList<ViewCollectionDetailsVO>
  getCollections(ViewCollectionDetailsVO vo,
  String letter) throws Exception {
    return
    ViewCollectionsDAO.getCollections(vo,
    letter);
  }

  public ArrayList<ViewReferenceDetailsVO>
  getReferences(ViewCollectionDetailsVO vo)
  throws Exception {
    return ViewReferencesDAO.getReferences(vo);
  }

  public ArrayList<ViewReferenceDetailsVO>
  getReferences() throws Exception {
    return ViewReferencesDAO.getReferences();
  }

  public String getProjectRole(String
  username, String project) throws Exception {
    return
    ViewCollectionsDAO.getProjectRole(username,
    project);
  }

  public boolean
  isCollectionCoordinator(String username)
  throws Exception {
    return
    ViewCollectionsDAO.isCollectionCoordinator(
    username);
  }

  public ViewReferenceDetailsVO
  readReference(ViewReferenceDetailsVO vo)
  throws Exception {
    return
    ViewReferencesDAO.getReferenceDetails(vo);
  }

  public ViewCollectionDetailsVO
  readCollection(ViewCollectionDetailsVO vo)
  throws Exception {
    return
    ViewCollectionsDAO.readCollection(vo);
  }

  public void
  insertCollection(ViewCollectionDetailsVO vo)
  throws Exception {
    AddCollectionDAO.insertCollection(vo);
  }

  public void
  updateCollection(ViewCollectionDetailsVO vo)
  throws Exception {
    EditCollectionDAO.updateCollection(vo);
  }

  public boolean checkCollection(String
  collection) throws Exception {
```

```
    return
    AddCollectionDAO.checkCollection(collection
    );
  }

  public void
  insertReference(ViewReferenceDetailsVO vo)
  throws Exception {
    AddReferenceDAO.insertReference(vo);
  }

  public boolean
  checkReference(ViewReferenceDetailsVO vo)
  throws Exception {
    return AddReferenceDAO.checkReference(vo);
  }

  public boolean checkReference(int
  referenceId) throws Exception {
    return
    ViewReferencesDAO.checkReference(referenceI
    d);
  }

  public void deleteReference(int referenceId)
  throws Exception {
    ViewReferencesDAO.deleteReference(reference
    Id);
  }

  public void
  deleteCollection(ViewCollectionDetailsVO vo)
  throws Exception {
    ViewCollectionsDAO.deleteCollection(vo);
  }

  public void
  updateReference(ViewReferenceDetailsVO vo)
  throws Exception {
    EditReferenceDAO.updateReference(vo);
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VIRHO INFORMATION SERVICES

AddHotspotServlet.java

```
package com.virholex.hotspots;

import java.io.IOException;

public class AddHotspotServlet extends
HttpServlet {

  private static final long serialVersionUID =
  1L;

  protected void doPost(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {
    doGet(request, response);
  }

  protected void doGet(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {

    String name =
    request.getParameter("photoID").trim();
    String stage =
    request.getParameter("message").trim();
    int height=0, width=0, x=0, y=0;

    try {
      height =
      (int)Double.parseDouble(request.getParame
      ter("height").trim());
      width =
      (int)Double.parseDouble(request.getParame
      ter("width").trim());
      x =
      (int)Double.parseDouble(request.getParame
      ter("x").trim());
      y =
      (int)Double.parseDouble(request.getParame
      ter("y").trim());
    } catch (NumberFormatException e) {
      e.printStackTrace();
    }

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement("INSERT INTO
      hotspots (name, stage, height, width, x ,
      y, date_added, date_modified)
      VALUES(?,?,?,?,?,?,NOW(),NOW())");
      ps.setString(1, name);
      ps.setString(2, stage);
      ps.setInt(3, height);
      ps.setInt(4, width);
      ps.setInt(5, x);
      ps.setInt(6, y);
      ps.executeUpdate();
      ps.close();

      ps = conn.prepareStatement("SELECT id
      FROM hotspots WHERE name = ? AND stage =
      ?");
      ps.setString(1, name);
      ps.setString(2, stage);
      rs = ps.executeQuery();
      if(rs.first()) {
        request.setAttribute("result",
        rs.getString("id"));
      }
      ps.close();
      conn.close();

    } catch (SQLException e) {
      e.printStackTrace();
    } catch (NamingException e) {
      e.printStackTrace();
    }

    request.getRequestDispatcher("/rd_virhohots
    pot/AJAXResult.jsp").include(request,
    response);

  }

}
```

AddVirusAction.java

```
package com.virholex.basicinfo;

import javax.servlet.http.HttpServletRequest;

public class AddVirusAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirusForm virusForm = (VirusForm) form;
  VirusVO virusVO = new VirusVO();
  VirhoBasicInfoWorker basicInfoWorker = new
  VirhoBasicInfoWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.basicinfo.addVirus";
  String virus = "";

  PropertyUtils.copyProperties(virusVO,
  virusForm);

  if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Submit")) {
    if(virusVO.getCommonName()!=null) {
      virus = virusVO.getCommonName();
      if(basicInfoWorker.checkVirus(virus)) {
        virusForm.setHasVirus(true);
      }
    }

    errors = virusForm.validate(mapping,
    request);

    if(errors.isEmpty()) {
      virusVO.setAuthor(username);
      basicInfoWorker.insertVirus(virusVO);
      Utilities.updateVirusList(request);

      PropertyUtils.copyProperties(virusForm,
      virusVO);
      session.setAttribute("virus", virus);

      forward =
      "virholex.basicinfo.viewVirusDetails";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("virus", virus);
      return redirect;
    } else {
      saveErrors(request, errors);
    }

  } else if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Cancel")) {
    forward = "virholex.basicinfo.viewVirus";
  }

  return mapping.findForward(forward);
}
}
```

AddVirusDAO.java

```
package com.virholex.basicinfo;

import java.sql.Connection;

public class AddVirusDAO {

  public static void insertVirus(VirusVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();

      String sql = "INSERT INTO
      basic_virus_info (scientific_name,
      common_name, serotype, host,
      virus_architecture, infectivity,
      manner_of_infection, vector,
      antimicrobial_treatment, receptors)
      VALUES (?,?,?,?,?,?,?,?,?,?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getScientificName());
      ps.setString(2, vo.getCommonName());
      ps.setString(3, vo.getSerotype());
      ps.setString(4, vo.getHost());
      ps.setString(5,
      vo.getVirusArchitecture());
      ps.setString(6, vo.getInfectivity());
      ps.setString(7,
      vo.getMannerOfInfection());
      ps.setString(8, vo.getVector());
      ps.setString(9,
      vo.getAntimicrobialTreatment());
      ps.setString(10, vo.getReceptors());
      ps.executeUpdate();
      ps.close();

      sql = "INSERT INTO virus (virus,
      description, date_added, date_modified)
      VALUES (?,?,NOW(),NOW())";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCommonName());
      ps.setString(2, vo.getDescription());
      ps.executeUpdate();
      ps.close();

      sql = "INSERT INTO project (project_name,
      virus, description, author, date_added,
      date_modified) VALUES
      (?,?,?,?,NOW(),NOW())";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCommonName() + " "
      + VirholexConstants.VIRHO_HOTSPOT);
      ps.setString(2, vo.getCommonName());
      ps.setString(3, vo.getDescription());
      ps.setString(4, vo.getAuthor());
      ps.executeUpdate();
      ps.close();

      sql = "INSERT INTO user_prev (username,
      role_name, involvement, date_added,
      date_modified, kind) VALUES
      (?,?,?,NOW(),NOW(),?)";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getAuthor());
      ps.setString(2,
      VirholexConstants.HOTSPOT_MANAGER_DESC);
      ps.setString(3, vo.getCommonName() + " "
      + VirholexConstants.VIRHO_HOTSPOT);
      ps.setString(4,
      VirholexConstants.PROJECT);
      ps.executeUpdate();

    } finally {
      if(ps != null) ps.close();
```

```
    if(conn != null) conn.close();
   }
  }

  public static boolean checkVirus(String
  virus) throws SQLException, NamingException
  {

   Connection conn = null;
   PreparedStatement ps = null;
   ResultSet rs = null;
   boolean hasVirus = false;

   try {
    String sql = "SELECT common_name FROM
    basic_virus_info WHERE common_name=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    rs = ps.executeQuery();

    if(rs.next()) {
     hasVirus = true;
    }
   } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
   }

   return hasVirus;
  }

  public static void uploadVirusImage(VirusVO
  vo) throws SQLException, NamingException {

   Connection conn = null;
   PreparedStatement ps = null;

   try {
    conn = DBConnect.getConnection();

    String sql = "UPDATE virus SET image=?,
    date_modified=NOW() WHERE virus=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, vo.getImageDB());
    ps.setString(2, vo.getVirusName());
    ps.executeUpdate();

   } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
   }
  }

}
```

AdvancedSearchAction.java

```
package com.virholex.internalsearch;

import java.util.ArrayList;

public class AdvancedSearchAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
 SearchForm searchForm = (SearchForm) form;
 SearchVO searchVO = new SearchVO();
 VirhoInternalSearchWorker searchWorker = new
 VirhoInternalSearchWorker();
 VirhoImagesWorker imageWorker = new
 VirhoImagesWorker();
 HttpSession session =
 request.getSession(true);
 ArrayList<ViewReferenceDetailsVO>
 referencesList = new
 ArrayList<ViewReferenceDetailsVO>();
 ArrayList<ViewExperimentDetailsVO>
 experimentsList = new
 ArrayList<ViewExperimentDetailsVO>();
 ArrayList<ViewModelDetailsVO> modelsList =
 new ArrayList<ViewModelDetailsVO>();
 ArrayList<ViewImageDetailsVO> imagesList =
 new ArrayList<ViewImageDetailsVO>();

 String username =
 (String)session.getAttribute("userLogin");
 String forward =
 "virholex.internalsearch.advancedSearch";
 String sortBy = "", roleName = "";
 boolean hasPrivilege = false;

 PropertyUtils.copyProperties(searchVO,
 searchForm);

 if(searchForm.getCommand()!=null &&
 searchForm.getCommand().equals("Search") ||
 searchVO.getExpSortBy()!=null ||
 searchVO.getModSortBy()!=null ||
 searchVO.getRefSortBy()!=null ||
 searchVO.getImgSortBy()!=null) {

  if(!Utilities.isEmpty(searchVO.getRefSearch
  Term())) {
   if(searchVO.getRefSortBy()==null)
    sortBy = "title";
   else {
    sortBy = searchVO.getRefSortBy();
    if(sortBy.equals("reference"))
     sortBy = "title";
   }
   referencesList =
   searchWorker.advancedSearchReferences(sea
   rchVO, sortBy);
   searchForm.setReferences(referencesList);
  }

  if(!Utilities.isEmpty(searchVO.getModSearch
  Term())) {
   if(searchVO.getModSortBy()==null)
    sortBy = "name";
   else {
    sortBy = searchVO.getModSortBy();
    if(sortBy.equals("model"))
     sortBy = "name";
    else if(sortBy.equals("project"))
     sortBy = "project_name";
   }
   modelsList =
   searchWorker.advancedSearchModels(searchV
   O, sortBy);
   searchForm.setModels(modelsList);
  }

  if(!Utilities.isEmpty(searchVO.getExpSearch
  Term())) {
   if(searchVO.getExpSortBy()==null)
    sortBy = "experiment_name";
   else {
    sortBy = searchVO.getExpSortBy();
    if(sortBy.equals("experiment"))
     sortBy = "experiment_name";
    else if(sortBy.equals("author"))
     sortBy = "e.author";
```

```
      else if(sortBy.equals("project"))
        sortBy = "project_name";
      }
      experimentsList =
      searchWorker.advancedSearchExperiments(se
      archVO, sortBy);
      searchForm.setExperiments(experimentsList
      );
    }

    if(!Utilities.isEmpty(searchVO.getImgSearch
    Term())) {
      if(searchVO.getImgSortBy()==null)
        sortBy = "image_path";
      else {
        sortBy = searchVO.getImgSortBy();
        if(sortBy.equals("image"))
          sortBy = "image_path";
        else if(sortBy.equals("imageSet"))
          sortBy = "project_set_name";
      }
      imagesList =
      searchWorker.advancedSearchImages(searchV
      O, sortBy);
      searchForm.setImages(imagesList);

      for(int i=0; i<imagesList.size(); i++) {
        if(((ViewImageDetailsVO)imagesList.get(
        i)).getAccessibility().equals("Restrict
        ed")) {
          roleName =
          imageWorker.getProjectRole(username,
          ((ViewImageDetailsVO)imagesList.get(i)
          ).getImageSet());

          if(roleName.equals(VirholexConstants.I
          MAGE_SET_COORDINATOR_DESC) ||
          roleName.equals(VirholexConstants.IMAG
          E_SET_CONTRIBUTOR_DESC) ||
          roleName.equals(VirholexConstants.REST
          RICTED_USER_DESC)) {
            hasPrivilege = true;
            break;
          }
        }
      }

      request.setAttribute("privilege",
      hasPrivilege);
    }

    forward =
    "virholex.internalsearch.advancedSearchResu
    lt";
  }

  PropertyUtils.copyProperties(searchForm,
  searchVO);

  return mapping.findForward(forward);
}
}
```

DeleteHotspotServlet.java

```
package com.virholex.hotspots;

import java.io.IOException;

public class DeleteHotspotServlet extends
HttpServlet {

  private static final long serialVersionUID =
  1L;

  protected void doGet(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {

    String id =
    request.getParameter("id").trim();
    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement("DELETE FROM
      hotspots WHERE id=?");
      ps.setString(1, id);

      request.setAttribute("result",
      ps.executeUpdate());
      ps.close();
      conn.close();
    } catch (SQLException e) {
      e.printStackTrace();
    } catch (NamingException e) {
      e.printStackTrace();
    }

    request.getRequestDispatcher("/rd_virhohots
    pot/AJAXResult.jsp").include(request,
    response);
  }

  protected void doPost(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {
    doGet(request, response);
  }
}
```

EditVirusAction.java

```
package com.virholex.basicinfo;

import javax.servlet.http.HttpServletRequest;

public class EditVirusAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirusForm virusForm = (VirusForm) form;
  VirusVO virusVO = new VirusVO();
  VirhoBasicInfoWorker basicInfoWorker = new
  VirhoBasicInfoWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();

  String forward =
  "virholex.basicinfo.editVirus";
  String virus =
  (String)session.getAttribute("virus");
  String description = "";

  PropertyUtils.copyProperties(virusVO,
  virusForm);

  if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Submit")) {

    if(virusVO.getCommonName()!=null &&
    !virusVO.getCommonName().equals(virusVO.get
    CommonNameDB())) {
```

```
      virus = virusVO.getCommonName();
      if(basicInfoWorker.checkVirus(virus)) {
        virusForm.setHasVirus(true);
      }
    }

    errors = virusForm.validate(mapping,
    request);

    if(errors.isEmpty()) {

      basicInfoWorker.updateVirus(virusVO);
      Utilities.updateVirusList(request);
      session.setAttribute("virus", virus);

      forward =
      "virholex.basicinfo.viewVirusDetails";
      ActionRedirect redirect = new
      ActionRedirect(mapping.findForward(forwar
      d));
      redirect.addParameter("virus", virus);
      return redirect;
    } else {
      saveErrors(request, errors);
    }

  } else if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Cancel")) {
    forward =
    "virholex.basicinfo.viewVirusDetails";
    ActionRedirect redirect = new
    ActionRedirect(mapping.findForward(forward)
    );
    redirect.addParameter("virus", virus);
    return redirect;
  } else {
    virusVO.setVirusName(virus);
    virusVO =
    (VirusVO)basicInfoWorker.readVirus(virusVO)
    ;
    description =
    basicInfoWorker.getVirusDescription(virus);
    virusVO.setDescription(description);
    virusVO.setDescriptionDB(description);
    virusVO.setCommonNameDB(virusVO.getCommonNa
    me());
  }

  PropertyUtils.copyProperties(virusForm,
  virusVO);

  return mapping.findForward(forward);
}
}
```

DeleteHotspotServlet.java

```
package com.virholex.basicinfo;

import java.sql.Connection;

public class EditVirusDAO {

  public static void updateVirus(VirusVO vo)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();

      String sql = "UPDATE basic_virus_info SET
      scientific_name=?, common_name=?,
      serotype=?, host=?, virus_architecture=?,
      infectivity=?, manner_of_infection=?,
      vector=?, antimicrobial_treatment=?,
      receptors=? WHERE common_name=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getScientificName());
      ps.setString(2, vo.getCommonName());
      ps.setString(3, vo.getSerotype());
      ps.setString(4, vo.getHost());
      ps.setString(5,
      vo.getVirusArchitecture());
      ps.setString(6, vo.getInfectivity());
      ps.setString(7,
      vo.getMannerOfInfection());
      ps.setString(8, vo.getVector());
      ps.setString(9,
      vo.getAntimicrobialTreatment());
      ps.setString(10, vo.getReceptors());
      ps.setString(11, vo.getCommonNameDB());
      ps.executeUpdate();
      ps.close();

      sql = "UPDATE virus SET virus=?,
      description=?, date_modified=NOW() WHERE
      virus=?";
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getCommonName());
      ps.setString(2, vo.getDescription());
      ps.setString(3, vo.getCommonNameDB());
      ps.executeUpdate();
      ps.close();

      if(!(vo.getCommonNameDB().equals(vo.getCo
      mmonName()) &&
      vo.getDescriptionDB().equals(vo.getDescri
      ption()))) {

        sql = "UPDATE project SET
        project_name=?, virus=?, description=?,
        date_modified=NOW() WHERE virus=?";
        ps = conn.prepareStatement(sql);
        ps.setString(1, vo.getCommonName() + "
        " + VirholexConstants.VIRHO_HOTSPOT);
        ps.setString(2, vo.getCommonName());
        ps.setString(3, vo.getDescription());
        ps.setString(4, vo.getCommonNameDB());
        ps.executeUpdate();
        ps.close();

        if(!vo.getCommonNameDB().equals(vo.getC
        ommonName())) {
          sql = "UPDATE user_prev SET
          involvement=?, date_modified=NOW()
          WHERE involvement LIKE ?";
          ps = conn.prepareStatement(sql);
          ps.setString(1, vo.getCommonName() + "
          " + VirholexConstants.VIRHO_HOTSPOT);
          ps.setString(2, vo.getCommonNameDB() +
          " " +
          VirholexConstants.VIRHO_HOTSPOT);
          ps.executeUpdate();
        }
      }

    } finally {
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }
  }

  public static String
  getVirusDescription(String virus) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
```

```
  ResultSet rs = null;
  String description = "";

  try {
    String sql = "SELECT description FROM
    virus WHERE virus=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    rs = ps.executeQuery();

    if(rs.next()) {
      description =
      rs.getString("description");
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return description;
}

public static void deleteImage(String virus,
String image) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE FROM hotspots WHERE
    name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, image);
    ps.executeUpdate();
    ps.close();

    sql = "UPDATE virus SET image=NULL WHERE
    virus=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    ps.executeUpdate();
    ps.close();

  } finally {
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }
}

public static void deleteVirus(String virus,
String image) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;

  try {
    conn = DBConnect.getConnection();

    String sql = "DELETE FROM hotspots WHERE
    name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, image);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM virus WHERE virus=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM basic_virus_info WHERE
    common_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM project WHERE
    project_name=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus + " " +
    VirholexConstants.VIRHO_HOTSPOT);
    ps.executeUpdate();
    ps.close();

    sql = "DELETE FROM user_prev WHERE
    involvement=?";
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus + " " +
    VirholexConstants.VIRHO_HOTSPOT);
    ps.executeUpdate();
    ps.close();

   } finally {
     if(ps != null) ps.close();
     if(conn != null) conn.close();
   }
  }

}
```

FetchHotspotInfoServlet.java

```
package com.virholex.hotspots;

import java.io.IOException;

public class FetchHotspotInfoServlet extends
HttpServlet {

  private static final long serialVersionUID =
  1L;

  protected void doGet(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {

    VirhoBasicInfoWorker basicInfoWorker = new
    VirhoBasicInfoWorker();
    HttpSession session =
    request.getSession(true);

    String username =
    (String)session.getAttribute("userLogin");
    String virus =
    (String)session.getAttribute("virus");
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      if(!Utilities.isEmpty(request.getParamete
      r("id"))) {
        conn = DBConnect.getConnection();
        int id =
        Integer.parseInt(request.getParameter("
        id").trim());

        ps = conn.prepareStatement("SELECT
        stage, information FROM hotspots WHERE
        id=?");
        ps.setInt(1, id);
```

```
      rs = ps.executeQuery();

      if(rs.first()) {
        request.setAttribute("info",
        rs.getString(1));
        request.setAttribute("stage",
        rs.getString(2));
      }

      conn.close();

      String roleName =
      basicInfoWorker.getProjectRole(username
      , virus);
      if(roleName.equals(VirholexConstants.HO
      TSPOT_MANAGER_DESC))
        request.setAttribute("privileged",
        true);
      else
        request.setAttribute("privileged",
        false);
    }
   } catch (SQLException e) {
    e.printStackTrace();
   } catch (NamingException e) {
    e.printStackTrace();
   } catch (NumberFormatException e) {
    e.printStackTrace();
   } catch (Exception e) {
    e.printStackTrace();
   }

   request.getRequestDispatcher("/rd_virhohots
   pot/DisplayHS.jsp").include(request,
   response);
  }

  protected void doPost(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {
    doGet(request, response);
  }

}
```

FetchHotspotsServlet.java

```
package com.virholex.hotspots;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.virholex.general.DBConnect;

public class FetchHotspotsServlet extends
HttpServlet {

  private static final long serialVersionUID =
  1L;

  protected void doGet(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {

    String virus =
    request.getParameter("photoID").trim();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement("SELECT id,
      name, x, y, width, height, stage FROM
      hotspots WHERE name = ?");
      ps.setString(1, virus);
      rs = ps.executeQuery();

      ArrayList<HashMap<String, String>> result
      = new
      ArrayList<HashMap<String,String>>();

      while(rs.next()) {
        HashMap<String, String> current = new
        HashMap<String, String>();
        current.put("id", rs.getString(1));
        current.put("name", rs.getString(2));
        current.put("x", rs.getString(3));
        current.put("y", rs.getString(4));
        current.put("width", rs.getString(5));
        current.put("height", rs.getString(6));
        current.put("stage", rs.getString(7));
        result.add(current);
      }

      request.setAttribute("result", result);

      ps.close();
      conn.close();
    } catch (SQLException e) {
      e.printStackTrace();
    } catch (NamingException e) {
      e.printStackTrace();
    }

    request.getRequestDispatcher("/rd_virhohots
    pot/HotspotJSON.jsp").include(request,
    response);
  }

  protected void doPost(HttpServletRequest
  request, HttpServletResponse response)
  throws ServletException, IOException {
    doGet(request, response);
  }

}
```

ManageHSServlet.java

```
package com.virholex.hotspots;

import javax.naming.NamingException;
import javax.servlet.http.HttpServlet;
import java.sql.*;

import com.virholex.general.DBConnect;

public class ManageHSServlet extends
HttpServlet {

  private static final long serialVersionUID =
  1L;

  public ManageHSServlet() {
    super();
  }
```

```
  public boolean saveChanges(String info,
  String id) {

    boolean saved = false;
    Connection conn = null;
    PreparedStatement ps = null;

    try {
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement("UPDATE
      hotspots SET information=? WHERE id=?");
      ps.setString(1, info);
      ps.setString(2, id);
      ps.executeUpdate();
      ps.close();
      conn.close();
      saved = true;
    } catch(SQLException e) {
      e.printStackTrace();
    } catch(NamingException e) {
      e.printStackTrace();
    }

    return saved;
  }

}
```

QuickSearchAction.java

```
package com.virholex.internalsearch;

import java.util.ArrayList;

public class QuickSearchAction extends Action
{

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  SearchForm searchForm = (SearchForm) form;
  SearchVO searchVO = new SearchVO();
  VirhoInternalSearchWorker searchWorker = new
  VirhoInternalSearchWorker();
  VirhoImagesWorker imageWorker = new
  VirhoImagesWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<ViewReferenceDetailsVO>
  referencesList = new
  ArrayList<ViewReferenceDetailsVO>();
  ArrayList<ViewExperimentDetailsVO>
  experimentsList = new
  ArrayList<ViewExperimentDetailsVO>();
  ArrayList<ViewModelDetailsVO> modelsList =
  new ArrayList<ViewModelDetailsVO>();
  ArrayList<ViewImageDetailsVO> imagesList =
  new ArrayList<ViewImageDetailsVO>();

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.internalsearch.searchResult";
  String key = "", sortBy = "", roleName = "";
  boolean hasPrivilege = false;

  PropertyUtils.copyProperties(searchVO,
  searchForm);

  if(searchForm.getCommand()!=null &&
  searchForm.getCommand().equals("Search") ||
searchVO.getExpSortBy()!=null ||
searchVO.getModSortBy()!=null ||
searchVO.getRefSortBy()!=null ||
searchVO.getImgSortBy()!=null) {

  key = searchVO.getSearchKey();

  if(searchVO.getModule().equals(VirholexCons
  tants.VIRHO_REFERENCES) ||
  searchVO.getModule().equals("All Modules"))
  {
    if(searchVO.getRefSortBy()==null)
      sortBy = "title";
    else {
      sortBy = searchVO.getRefSortBy();
      if(sortBy.equals("reference"))
        sortBy = "title";
    }
    referencesList =
    searchWorker.searchReferences(key,
    sortBy);
    searchForm.setReferences(referencesList);
  }

  if(searchVO.getModule().equals(VirholexCons
  tants.VIRHO_EXPERIMENTS) ||
  searchVO.getModule().equals("All Modules"))
  {
    if(searchVO.getExpSortBy()==null)
      sortBy = "experiment_name";
    else {
      sortBy = searchVO.getExpSortBy();
      if(sortBy.equals("experiment"))
        sortBy = "experiment_name";
      else if(sortBy.equals("author"))
        sortBy = "e.author";
      else if(sortBy.equals("project"))
        sortBy = "project_name";
    }
    experimentsList =
    searchWorker.searchExperiments(key,
    sortBy);
    searchForm.setExperiments(experimentsList
    );
  }

  if(searchVO.getModule().equals(VirholexCons
  tants.VIRHO_MODELS) ||
  searchVO.getModule().equals("All Modules"))
  {
    if(searchVO.getModSortBy()==null)
      sortBy = "name";
    else {
      sortBy = searchVO.getModSortBy();
      if(sortBy.equals("model"))
        sortBy = "name";
      else if(sortBy.equals("project"))
        sortBy = "project_name";
    }
    modelsList =
    searchWorker.searchModels(key, sortBy);
    searchForm.setModels(modelsList);
  }

  if(searchVO.getModule().equals(VirholexCons
  tants.VIRHO_IMAGES) ||
  searchVO.getModule().equals("All Modules"))
  {
    if(searchVO.getImgSortBy()==null)
      sortBy = "image_path";
    else {
      sortBy = searchVO.getImgSortBy();
      if(sortBy.equals("image"))
        sortBy = "image_path";
      else if(sortBy.equals("imageSet"))
        sortBy = "project_set_name";
```

```
    }
    imagesList =
    searchWorker.searchImages(key, sortBy);
    searchForm.setImages(imagesList);

    for(int i=0; i<imagesList.size(); i++) {
      if(((ViewImageDetailsVO)imagesList.get(
      i)).getAccessibility().equals("Restrict
      ed")) {
        roleName =
        imageWorker.getProjectRole(username,
        ((ViewImageDetailsVO)imagesList.get(i)
        ).getImageSet());

        if(roleName.equals(VirholexConstants.I
        MAGE_SET_COORDINATOR_DESC) ||
        roleName.equals(VirholexConstants.IMAG
        E_SET_CONTRIBUTOR_DESC) ||
        roleName.equals(VirholexConstants.REST
        RICTED_USER_DESC)) {
          hasPrivilege = true;
          break;
        }
      }
    }

    request.setAttribute("privilege",
    hasPrivilege);
   }
  }

  PropertyUtils.copyProperties(searchForm,
  searchVO);

  return mapping.findForward(forward);
}
}
```

SearchDAO.java

```
package com.virholex.internalsearch;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.naming.NamingException;

import com.virholex.general.DBConnect;
import com.virholex.general.Utilities;
import
com.virholex.references.ViewReferenceDetailsVO
;
import
com.virholex.experiments.ViewExperimentDetails
VO;
import com.virholex.models.ViewModelDetailsVO;
import com.virholex.images.ViewImageDetailsVO;

public class SearchDAO {

  public static
  ArrayList<ViewReferenceDetailsVO>
  searchReferences(String key, String sortBy)
  throws SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    ViewReferenceDetailsVO vo = new
    ViewReferenceDetailsVO();

    ArrayList<ViewReferenceDetailsVO>
    referencesList = new
    ArrayList<ViewReferenceDetailsVO>();

    try {
      String sql = "SELECT DISTINCT id, title,
      author, collection FROM coll_reference
      WHERE " +
       "title LIKE ? " +
       "OR author LIKE ? " +
       "OR publisher LIKE ? " +
       "OR year LIKE ? " +
       "OR chapter LIKE ? " +
       "OR pages LIKE ? " +
       "OR booktitle LIKE ? " +
       "OR school LIKE ? " +
       "OR institution LIKE ? " +
       "OR note LIKE ? " +
       "OR volume LIKE ? " +
       "OR series LIKE ? " +
       "OR address LIKE ? " +
       "OR edition LIKE ? " +
       "OR month LIKE ? " +
       "OR editor LIKE ? " +
       "OR organization LIKE ? " +
       "OR howpublished LIKE ? " +
       "OR collection LIKE ? " +
       "OR keyword LIKE ? " +
       "OR number LIKE ? ORDER BY " + sortBy;
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, "%"+key+"%");
      ps.setString(2, "%"+key+"%");
      ps.setString(3, "%"+key+"%");
      ps.setString(4, "%"+key+"%");
      ps.setString(5, "%"+key+"%");
      ps.setString(6, "%"+key+"%");
      ps.setString(7, "%"+key+"%");
      ps.setString(8, "%"+key+"%");
      ps.setString(9, "%"+key+"%");
      ps.setString(10, "%"+key+"%");
      ps.setString(11, "%"+key+"%");
      ps.setString(12, "%"+key+"%");
      ps.setString(13, "%"+key+"%");
      ps.setString(14, "%"+key+"%");
      ps.setString(15, "%"+key+"%");
      ps.setString(16, "%"+key+"%");
      ps.setString(17, "%"+key+"%");
      ps.setString(18, "%"+key+"%");
      ps.setString(19, "%"+key+"%");
      ps.setString(20, "%"+key+"%");
      ps.setString(21, "%"+key+"%");
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setReferenceId(rs.getInt(1));
        vo.setTitle(rs.getString(2));
        vo.setAuthor(rs.getString(3));
        vo.setCollection(rs.getString(4));
        referencesList.add(vo);
        vo = new ViewReferenceDetailsVO();
      }
    } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
    }

    return referencesList;
  }

  public static
  ArrayList<ViewExperimentDetailsVO>
  searchExperiments(String key, String sortBy)
  throws SQLException, NamingException {
```

```
Connection conn = null;
PreparedStatement ps = null;
ResultSet rs = null;
ViewExperimentDetailsVO vo = new
ViewExperimentDetailsVO();
ArrayList<ViewExperimentDetailsVO>
experimentsList = new
ArrayList<ViewExperimentDetailsVO>();

try {
  String sql = "SELECT DISTINCT
  e.experiment_id, p.project_id,
  project_name, experiment_name, e.author
  FROM experiments AS e, " +
   "project AS p WHERE
  (p.project_id=e.project_id) AND (" +
   "project_name LIKE ? " +
   "OR project_name LIKE ? " +
   "OR experiment_name LIKE ? " +
   "OR e.author LIKE ?" +
   "OR e.description LIKE ? " +
   "OR virus_used LIKE ? " +
   "OR measurement_interval LIKE ? " +
   "OR biological_replicates LIKE ? " +
   "OR technical_replicates LIKE ? " +
   "OR (SELECT COUNT(*) FROM conditions AS
  c WHERE condition_value LIKE ? AND
  e.experiment_id=c.experiment_id AND
  p.project_id=c.project_id) > 0 " +
   "OR (SELECT COUNT(*) FROM
  chemical_activators AS ca WHERE
  chem_activator_name LIKE ? AND
  e.experiment_id=ca.experiment_id AND
  p.project_id=ca.project_id) > 0 " +
   "OR (SELECT COUNT(*) FROM methods AS m
  WHERE (method_name LIKE ? OR
  m.description LIKE ?) AND
  e.experiment_id=m.experiment_id AND
  p.project_id=m.project_id) > 0 " +
   "OR (SELECT COUNT(*) FROM variables AS v
  WHERE (variable_type LIKE ? OR
  variable_value LIKE ?) AND
  e.experiment_id=v.experiment_id AND
  p.project_id=v.project_id) > 0 " +
   ") ORDER BY " + sortBy;
  conn = DBConnect.getConnection();
  ps = conn.prepareStatement(sql);
  ps.setString(1, "%"+key+"%");
  ps.setString(2, "%"+key+"%");
  ps.setString(3, "%"+key+"%");
  ps.setString(4, "%"+key+"%");
  ps.setString(5, "%"+key+"%");
  ps.setString(6, "%"+key+"%");
  ps.setString(7, "%"+key+"%");
  ps.setString(8, "%"+key+"%");
  ps.setString(9, "%"+key+"%");
  ps.setString(10, "%"+key+"%");
  ps.setString(11, "%"+key+"%");
  ps.setString(12, "%"+key+"%");
  ps.setString(13, "%"+key+"%");
  ps.setString(14, "%"+key+"%");
  ps.setString(15, "%"+key+"%");
  rs = ps.executeQuery();

  while(rs.next()) {
    vo.setExperimentId(rs.getInt(1));
    vo.setProjectId(rs.getInt(2));
    vo.setProjectName(rs.getString(3));
    vo.setExperimentName(rs.getString(4));
    vo.setAuthor(rs.getString(5));
    experimentsList.add(vo);
    vo = new ViewExperimentDetailsVO();
  }

} finally {
  if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return experimentsList;
}

public static ArrayList<ViewImageDetailsVO>
searchImages(String key, String sortBy)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewImageDetailsVO vo = new
  ViewImageDetailsVO();
  ArrayList<ViewImageDetailsVO> imagesList =
  new ArrayList<ViewImageDetailsVO>();

  try {
    String sql = "SELECT DISTINCT image_id,
    project_set_id, project_set_name,
    image_path, category, description,
    accessibility FROM image, project_set " +
     "WHERE project_set_id=image_set_id AND
    (" +
     "project_set_name LIKE ? " +
     "OR image_path LIKE ? " +
     "OR accessibility LIKE ? " +
     "OR category LIKE ? " +
     "OR description LIKE ? " +
     "OR cell_type LIKE ? " +
     "OR virus_type LIKE ? " +
     "OR mult_of_infection LIKE ? " +
     "OR medium LIKE ? " +
     "OR temperature LIKE ? " +
     "OR co2_level LIKE ? " +
     "OR owner LIKE ? " +
     "OR container LIKE ? " +
     "OR phenol_red LIKE ? " +
     "OR magnification LIKE ? " +
     "OR objective LIKE ? " +
     "OR illumination_type LIKE ? " +
     "OR illumination_level LIKE ? " +
     "OR filter LIKE ? " +
     "OR project_set_name LIKE ?) ORDER BY "
    + sortBy;

    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, "%"+key+"%");
    ps.setString(2, "%"+key+"%");
    ps.setString(3, "%"+key+"%");
    ps.setString(4, "%"+key+"%");
    ps.setString(5, "%"+key+"%");
    ps.setString(6, "%"+key+"%");
    ps.setString(7, "%"+key+"%");
    ps.setString(8, "%"+key+"%");
    ps.setString(9, "%"+key+"%");
    ps.setString(10, "%"+key+"%");
    ps.setString(11, "%"+key+"%");
    ps.setString(12, "%"+key+"%");
    ps.setString(13, "%"+key+"%");
    ps.setString(14, "%"+key+"%");
    ps.setString(15, "%"+key+"%");
    ps.setString(16, "%"+key+"%");
    ps.setString(17, "%"+key+"%");
    ps.setString(18, "%"+key+"%");
    ps.setString(19, "%"+key+"%");
    ps.setString(20, "%"+key+"%");
    rs = ps.executeQuery();

    while(rs.next()) {
      vo.setImageId(rs.getInt(1));
      vo.setImageSetId(rs.getInt(2));
      vo.setImageSet(rs.getString(3));
```

```
    vo.setImagePathDB(rs.getString(4));
    vo.setCategory(rs.getString(5));
    vo.setDescription(rs.getString(6));
    vo.setAccessibility(rs.getString(7));
    vo.setImagePath(Utilities.removeFileExt
    ension(vo.getImagePathDB(),"."+Utilitie
    s.getFileExtension(vo.getImagePathDB())
    ));
    imagesList.add(vo);
    vo = new ViewImageDetailsVO();
   }

  } finally {
   if(rs != null) rs.close();
   if(ps != null) ps.close();
   if(conn != null) conn.close();
  }

  return imagesList;
}

public static ArrayList<ViewModelDetailsVO>
searchModels(String key, String sortBy)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewModelDetailsVO vo = new
  ViewModelDetailsVO();
  ArrayList<ViewModelDetailsVO> modelsList =
  new ArrayList<ViewModelDetailsVO>();

  try {
   String sql = "SELECT DISTINCT model_id,
   name, classification, project_name,
   author FROM models WHERE " +
    "name LIKE ? " +
    "OR name LIKE ? " +
    "OR classification LIKE ? " +
    "OR project_name LIKE ? " +
    "OR description LIKE ? " +
    "OR scale LIKE ? " +
    "OR scope LIKE ? " +
    "OR size LIKE ? " +
    "OR type LIKE ? " +
    "OR aspects LIKE ? " +
    "OR tools LIKE ? " +
    "OR properties LIKE ? " +
    "OR link LIKE ? " +
    "OR author LIKE ? ORDER BY " + sortBy;
   conn = DBConnect.getConnection();
   ps = conn.prepareStatement(sql);
   ps.setString(1, "%"+key+"%");
   ps.setString(2, "%"+key+"%");
   ps.setString(3, "%"+key+"%");
   ps.setString(4, "%"+key+"%");
   ps.setString(5, "%"+key+"%");
   ps.setString(6, "%"+key+"%");
   ps.setString(7, "%"+key+"%");
   ps.setString(8, "%"+key+"%");
   ps.setString(9, "%"+key+"%");
   ps.setString(10, "%"+key+"%");
   ps.setString(11, "%"+key+"%");
   ps.setString(12, "%"+key+"%");
   ps.setString(13, "%"+key+"%");
   ps.setString(14, "%"+key+"%");
   rs = ps.executeQuery();

   while(rs.next()) {
    vo.setModelId(rs.getInt("model_id"));
    vo.setModelName(rs.getString("name"));
    vo.setClassification(rs.getString("clas
    sification"));
    vo.setProjectName(rs.getString("project
    _name"));
    vo.setAuthor(rs.getString("author"));
    modelsList.add(vo);
    vo = new ViewModelDetailsVO();
   }

  } finally {
   if(rs != null) rs.close();
   if(ps != null) ps.close();
   if(conn != null) conn.close();
  }

  return modelsList;
}

public static
ArrayList<ViewReferenceDetailsVO>
advancedSearchReferences(SearchVO vo, String
sortBy) throws SQLException, NamingException
{

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewReferenceDetailsVO refVO = new
  ViewReferenceDetailsVO();
  ArrayList<ViewReferenceDetailsVO>
  referencesList = new
  ArrayList<ViewReferenceDetailsVO>();

  try {
   String sql = "SELECT DISTINCT id, title,
   author, collection FROM coll_reference
   WHERE " +
    "title LIKE ? " +
    "OR author LIKE ? " +
    "OR publisher LIKE ? " +
    "OR year LIKE ? " +
    "OR chapter LIKE ? " +
    "OR pages LIKE ? " +
    "OR booktitle LIKE ? " +
    "OR school LIKE ? " +
    "OR institution LIKE ? " +
    "OR note LIKE ? " +
    "OR volume LIKE ? " +
    "OR series LIKE ? " +
    "OR address LIKE ? " +
    "OR edition LIKE ? " +
    "OR month LIKE ? " +
    "OR editor LIKE ? " +
    "OR organization LIKE ? " +
    "OR howpublished LIKE ? " +
    "OR collection LIKE ? " +
    "OR keyword LIKE ? " +
    "OR number LIKE ? ORDER BY " + sortBy;

   conn = DBConnect.getConnection();
   ps = conn.prepareStatement(sql);
   ps.setString(1,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(2,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(3,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(4,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(5,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(6,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(7,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(8,
   "%"+vo.getRefSearchTerm()+"%");
   ps.setString(9,
   "%"+vo.getRefSearchTerm()+"%");
```

```
    ps.setString(10,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(11,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(12,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(13,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(14,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(15,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(16,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(17,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(18,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(19,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(20,
    "%"+vo.getRefSearchTerm()+"%");
    ps.setString(21,
    "%"+vo.getRefSearchTerm()+"%");
    rs = ps.executeQuery();

    while(rs.next()) {
      refVO.setReferenceId(rs.getInt(1));
      refVO.setTitle(rs.getString(2));
      refVO.setAuthor(rs.getString(3));
      refVO.setCollection(rs.getString(4));

      referencesList.add(refVO);
      refVO = new ViewReferenceDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return referencesList;
}

public static ArrayList<ViewModelDetailsVO>
advancedSearchModels(SearchVO vo, String
sortBy) throws SQLException, NamingException
{

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewModelDetailsVO modVO = new
  ViewModelDetailsVO();
  ArrayList<ViewModelDetailsVO> modelsList =
  new ArrayList<ViewModelDetailsVO>();

  try {
    String sql = "SELECT DISTINCT model_id,
    name, classification, project_name,
    author FROM models WHERE " +
     "name LIKE ? " +
     "OR name LIKE ? " +
     "OR classification LIKE ? " +
     "OR project_name LIKE ? " +
     "OR description LIKE ? " +
     "OR scale LIKE ? " +
     "OR scope LIKE ? " +
     "OR size LIKE ? " +
     "OR type LIKE ? " +
     "OR aspects LIKE ? " +
     "OR tools LIKE ? " +
     "OR properties LIKE ? " +
     "OR link LIKE ? " +
     "OR author LIKE ? ORDER BY " + sortBy;
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(2,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(3,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(4,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(5,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(6,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(7,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(8,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(9,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(10,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(11,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(12,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(13,
    "%"+vo.getModSearchTerm()+"%");
    ps.setString(14,
    "%"+vo.getModSearchTerm()+"%");
    rs = ps.executeQuery();

    while(rs.next()) {
      modVO.setModelId(rs.getInt("model_id"))
      ;
      modVO.setModelName(rs.getString("name")
      );
      modVO.setClassification(rs.getString("c
      lassification"));
      modVO.setProjectName(rs.getString("proj
      ect_name"));
      modVO.setAuthor(rs.getString("author"))
      ;
      modelsList.add(modVO);
      modVO = new ViewModelDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return modelsList;
}

public static
ArrayList<ViewExperimentDetailsVO>
advancedSearchExperiments(SearchVO vo,
String sortBy) throws SQLException,
NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewExperimentDetailsVO expVO = new
  ViewExperimentDetailsVO();
  ArrayList<ViewExperimentDetailsVO>
  experimentsList = new
  ArrayList<ViewExperimentDetailsVO>();

  try {
    String sql = "SELECT DISTINCT
    e.experiment_id, p.project_id,
    project_name, experiment_name, e.author
    FROM experiments AS e, " +
```

```
 "project AS p WHERE
(p.project_id=e.project_id) AND (" +
 "project_name LIKE ? " +
 "OR project_name LIKE ? " +
 "OR experiment_name LIKE ? " +
 "OR e.author LIKE ?" +
 "OR e.description LIKE ? " +
 "OR virus_used LIKE ? " +
 "OR measurement_interval LIKE ? " +
 "OR biological_replicates LIKE ? " +
 "OR technical_replicates LIKE ? " +
 "OR (SELECT COUNT(*) FROM conditions AS
c WHERE condition_value LIKE ? AND
e.experiment_id=c.experiment_id AND
p.project_id=c.project_id) > 0 " +
 "OR (SELECT COUNT(*) FROM
chemical_activators AS ca WHERE
chem_activator_name LIKE ? AND
e.experiment_id=ca.experiment_id AND
p.project_id=ca.project_id) > 0 " +
 "OR (SELECT COUNT(*) FROM methods AS m
WHERE (method_name LIKE ? OR
m.description LIKE ?) AND
e.experiment_id=m.experiment_id AND
p.project_id=m.project_id) > 0 " +
 "OR (SELECT COUNT(*) FROM variables AS v
WHERE (variable_type LIKE ? OR
variable_value LIKE ?) AND
e.experiment_id=v.experiment_id AND
p.project_id=v.project_id) > 0 " +
 ") ORDER BY " + sortBy;
conn = DBConnect.getConnection();
ps = conn.prepareStatement(sql);
ps.setString(1,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(2,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(3,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(4,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(5,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(6,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(7,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(8,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(9,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(10,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(11,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(12,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(13,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(14,
"%"+vo.getExpSearchTerm()+"%");
ps.setString(15,
"%"+vo.getExpSearchTerm()+"%");
rs = ps.executeQuery();

while(rs.next()) {
  expVO.setExperimentId(rs.getInt(1));
  expVO.setProjectId(rs.getInt(2));
  expVO.setProjectName(rs.getString(3));
  expVO.setExperimentName(rs.getString(4)
  );
  expVO.setAuthor(rs.getString(5));
  experimentsList.add(expVO);
  expVO = new ViewExperimentDetailsVO();
}
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return experimentsList;
}

public static ArrayList<ViewImageDetailsVO>
advancedSearchImages(SearchVO vo, String
sortBy) throws SQLException, NamingException
{

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  ViewImageDetailsVO imgVO = new
  ViewImageDetailsVO();
  ArrayList<ViewImageDetailsVO> imagesList =
  new ArrayList<ViewImageDetailsVO>();

  try {
  String sql = "SELECT DISTINCT image_id,
  project_set_id, project_set_name,
  image_path, category, description,
  accessibility FROM image, project_set " +
   "WHERE project_set_id=image_set_id AND ("
  +
   "project_set_name LIKE ? " +
   "OR image_path LIKE ? " +
   "OR accessibility LIKE ? " +
   "OR category LIKE ? " +
   "OR description LIKE ? " +
   "OR cell_type LIKE ? " +
   "OR virus_type LIKE ? " +
   "OR mult_of_infection LIKE ? " +
   "OR medium LIKE ? " +
   "OR temperature LIKE ? " +
   "OR co2_level LIKE ? " +
   "OR owner LIKE ? " +
   "OR container LIKE ? " +
   "OR phenol_red LIKE ? " +
   "OR magnification LIKE ? " +
   "OR objective LIKE ? " +
   "OR illumination_type LIKE ? " +
   "OR illumination_level LIKE ? " +
   "OR filter LIKE ? " +
   "OR project_set_name LIKE ?) ORDER BY " +
  sortBy;

  conn = DBConnect.getConnection();
  ps = conn.prepareStatement(sql);
  ps.setString(1,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(2,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(3,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(4,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(5,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(6,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(7,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(8,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(9,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(10,
  "%"+vo.getImgSearchTerm()+"%");
  ps.setString(11,
  "%"+vo.getImgSearchTerm()+"%");
```

```
    ps.setString(12,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(13,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(14,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(15,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(16,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(17,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(18,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(19,
    "%"+vo.getImgSearchTerm()+"%");
    ps.setString(20,
    "%"+vo.getImgSearchTerm()+"%");
    rs = ps.executeQuery();

    while(rs.next()) {
      imgVO.setImageId(rs.getInt(1));
      imgVO.setImageSetId(rs.getInt(2));
      imgVO.setImageSet(rs.getString(3));
      imgVO.setImagePathDB(rs.getString(4));
      imgVO.setCategory(rs.getString(5));
      imgVO.setDescription(rs.getString(6));
      imgVO.setAccessibility(rs.getString(7));
      imgVO.setImagePath(Utilities.removeFileEx
      tension(imgVO.getImagePathDB(),"."+Utilit
      ies.getFileExtension(imgVO.getImagePathDB
      ())));
      imagesList.add(imgVO);
      imgVO = new ViewImageDetailsVO();
    }

  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return imagesList;
  }

}
```

SearchForm.java

```
package com.virholex.internalsearch;

import java.util.ArrayList;

import org.apache.struts.action.ActionForm;

import
com.virholex.references.ViewReferenceDetailsVO
;
import
com.virholex.experiments.ViewExperimentDetails
VO;
import com.virholex.models.ViewModelDetailsVO;
import com.virholex.images.ViewImageDetailsVO;

public class SearchForm extends ActionForm {

  private static final long serialVersionUID =
  -8313667513322730768L;
  private String searchKey;
  private String command;
  private String module;
  private String refSortBy;
  private String expSortBy;
  private String modSortBy;
  private String imgSortBy;
  private String expSearchTerm;
  private String modSearchTerm;
  private String refSearchTerm;
  private String imgSearchTerm;
  private ArrayList<ViewReferenceDetailsVO>
  references = new
  ArrayList<ViewReferenceDetailsVO>();
  private ArrayList<ViewExperimentDetailsVO>
  experiments = new
  ArrayList<ViewExperimentDetailsVO>();
  private ArrayList<ViewModelDetailsVO> models
  = new ArrayList<ViewModelDetailsVO>();
  private ArrayList<ViewImageDetailsVO> images
  = new ArrayList<ViewImageDetailsVO>();

  public void setSearchKey(String searchKey) {
    this.searchKey = searchKey;
  }
  public String getSearchKey() {
    return searchKey;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setModule(String module) {
    this.module = module;
  }
  public String getModule() {
    return module;
  }
  public void setRefSortBy(String refSortBy) {
    this.refSortBy = refSortBy;
  }
  public String getRefSortBy() {
    return refSortBy;
  }
  public void setExpSortBy(String expSortBy) {
    this.expSortBy = expSortBy;
  }
  public String getExpSortBy() {
    return expSortBy;
  }
  public void setModSortBy(String modSortBy) {
    this.modSortBy = modSortBy;
  }
  public String getModSortBy() {
    return modSortBy;
  }
  public void setImgSortBy(String imgSortBy) {
    this.imgSortBy = imgSortBy;
  }
  public String getImgSortBy() {
    return imgSortBy;
  }
  public void setExpSearchTerm(String
  expSearchTerm) {
    this.expSearchTerm = expSearchTerm;
  }
  public String getExpSearchTerm() {
    return expSearchTerm;
  }
  public void setModSearchTerm(String
  modSearchTerm) {
    this.modSearchTerm = modSearchTerm;
  }
  public String getModSearchTerm() {
    return modSearchTerm;
  }
  public void setRefSearchTerm(String
  refSearchTerm) {
    this.refSearchTerm = refSearchTerm;
  }
```

```
  public String getRefSearchTerm() {
    return refSearchTerm;
  }
  public void setImgSearchTerm(String
  imgSearchTerm) {
    this.imgSearchTerm = imgSearchTerm;
  }
  public String getImgSearchTerm() {
    return imgSearchTerm;
  }
  public void
  setReferences(ArrayList<ViewReferenceDetails
  VO> references) {
    this.references = references;
  }
  public ArrayList<ViewReferenceDetailsVO>
  getReferences() {
    return references;
  }
  public void
  setExperiments(ArrayList<ViewExperimentDetai
  lsVO> experiments) {
    this.experiments = experiments;
  }
  public ArrayList<ViewExperimentDetailsVO>
  getExperiments() {
    return experiments;
  }
  public void
  setModels(ArrayList<ViewModelDetailsVO>
  models) {
    this.models = models;
  }
  public ArrayList<ViewModelDetailsVO>
  getModels() {
    return models;
  }
  public void
  setImages(ArrayList<ViewImageDetailsVO>
  images) {
    this.images = images;
  }
  public ArrayList<ViewImageDetailsVO>
  getImages() {
    return images;
  }

}
```

SearchVO.java

```
package com.virholex.internalsearch;

import com.virholex.general.VO;

public class SearchVO implements VO {

  private static final long serialVersionUID =
  -2076917457336861877L;
  private String searchKey;
  private String command;
  private String module;
  private String refSortBy;
  private String expSortBy;
  private String modSortBy;
  private String imgSortBy;
  private String expSearchTerm;
  private String modSearchTerm;
  private String refSearchTerm;
  private String imgSearchTerm;

  public void setSearchKey(String searchKey) {
    this.searchKey = searchKey;
  }
  public String getSearchKey() {
    return searchKey;
  }
  public void setCommand(String command) {
    this.command = command;
  }
  public String getCommand() {
    return command;
  }
  public void setModule(String module) {
    this.module = module;
  }
  public String getModule() {
    return module;
  }
  public void setRefSortBy(String refSortBy) {
    this.refSortBy = refSortBy;
  }
  public String getRefSortBy() {
    return refSortBy;
  }
  public void setExpSortBy(String expSortBy) {
    this.expSortBy = expSortBy;
  }
  public String getExpSortBy() {
    return expSortBy;
  }
  public void setModSortBy(String modSortBy) {
    this.modSortBy = modSortBy;
  }
  public String getModSortBy() {
    return modSortBy;
  }
  public void setImgSortBy(String imgSortBy) {
    this.imgSortBy = imgSortBy;
  }
  public String getImgSortBy() {
    return imgSortBy;
  }
  public void setExpSearchTerm(String
  expSearchTerm) {
    this.expSearchTerm = expSearchTerm;
  }
  public String getExpSearchTerm() {
    return expSearchTerm;
  }
  public void setModSearchTerm(String
  modSearchTerm) {
    this.modSearchTerm = modSearchTerm;
  }
  public String getModSearchTerm() {
    return modSearchTerm;
  }
  public void setRefSearchTerm(String
  refSearchTerm) {
    this.refSearchTerm = refSearchTerm;
  }
  public String getRefSearchTerm() {
    return refSearchTerm;
  }
  public void setImgSearchTerm(String
  imgSearchTerm) {
    this.imgSearchTerm = imgSearchTerm;
  }
  public String getImgSearchTerm() {
    return imgSearchTerm;
  }

}
```

ViewUsersAction.java

```
package com.virholex.basicinfo;
```

```
import java.util.ArrayList;

public class ViewUsersAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  ViewUsersForm usersForm = (ViewUsersForm)
  form;
  UserProfileVO userVO = new UserProfileVO();
  VirhoVIRUSWorker virusWorker = new
  VirhoVIRUSWorker();
  VirhoBasicInfoWorker basicInfoWorker = new
  VirhoBasicInfoWorker();
  HttpSession session =
  request.getSession(true);
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<String> rolesList = new
  ArrayList<String>();
  ArrayList<String> usersList = new
  ArrayList<String>();

  String username =
  (String)session.getAttribute("userLogin");
  String virus =
  (String)session.getAttribute("virus");
  String forward =
  "virholex.basicinfo.viewUsers";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = "", roleName = "";
  int privilege = 0;

  if(!Utilities.isEmpty(request.getParameter("
  list")))
    list= request.getParameter("list");

  for(int i=0; i < letters.length() ; i++){
    if(virusWorker.getUsers(virus,
    Character.toString(letters.charAt(i)),
    VirholexConstants.VIRHO_HOTSPOTS).size() >
    0)
      letterList.add(Character.toString(letters
      .charAt(i)));
    else
      letterList.add("-");
  }

  usersForm.setLetters(letterList);

  if(usersForm.getCommand()!=null &&
  usersForm.getCommand().equals("Change
  Privilege Role")) {
    String[] roles =
    request.getParameterValues("role");
    String[] oldRoles =
    request.getParameterValues("oldRole");
    String[] users =
    request.getParameterValues("userName");

    if(roles!=null && oldRoles!=null &&
    users!=null) {
      for(int i=0; i<oldRoles.length; i++) {
        if(!roles[i].equals(oldRoles[i])) {
          rolesList.add(roles[i]);
          usersList.add(users[i]);
        }
      }

      if(!rolesList.isEmpty() &&
      !usersList.isEmpty()) {
        virusWorker.updateProjectPrivilege(user
        sList, rolesList, virus,
        VirholexConstants.VIRHO_HOTSPOTS);
        rolesList = new ArrayList<String>();
      }
    }
  }

  userVO.setUsers(virusWorker.getUsers(virus,
  list, VirholexConstants.VIRHO_HOTSPOTS));
  rolesList =
  virusWorker.getUserRoles(VirholexConstants.V
  IRHO_HOTSPOTS);

  roleName =
  basicInfoWorker.getProjectRole(username,
  virus);

  if(roleName.equals(VirholexConstants.HOTSPOT
  _MANAGER_DESC))
    privilege =
    VirholexConstants.HOTSPOT_MANAGER;
  else
    privilege =
    VirholexConstants.REGISTERED_USER;

  PropertyUtils.copyProperties(usersForm,
  userVO);

  request.setAttribute("privilege",
  privilege);
  request.setAttribute("roles", rolesList);

  return mapping.findForward(forward);
}
}
```

ViewVirusAction.java

```
package com.virholex.basicinfo;

import java.util.ArrayList;

public class ViewVirusAction extends Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirusForm virusForm = (VirusForm) form;
  VirusVO virusVO = new VirusVO();
  VirhoBasicInfoWorker basicInfoWorker = new
  VirhoBasicInfoWorker();
  ArrayList<String> letterList = new
  ArrayList<String>();
  ArrayList<VirusVO> virusList = new
  ArrayList<VirusVO>();
  HttpSession session =
  request.getSession(true);

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.basicinfo.viewVirus";
  String letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  String list = request.getParameter("list");
  boolean isManager = false;

  if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Add Virus"))
  {
    forward = "virholex.basicinfo.addVirus";
```

```
  } else {
    for(int i=0; i < letters.length() ; i++){
      if(basicInfoWorker.checkVirus(letters.cha
      rAt(i)).size() > 0)
        letterList.add(Character.toString(lette
        rs.charAt(i)));
      else
        letterList.add("-");
    }
    // Get list of viruses
    if(list != null) {
      virusList =
      basicInfoWorker.getVirus(virusVO, list);
    }
    // Check if the user is a Hotspot Manager
    isManager =
    basicInfoWorker.isManager(username);

    request.setAttribute("letters",
    letterList);
    request.setAttribute("virus", virusList);
    request.setAttribute("privilege",
    isManager);
  }

  return mapping.findForward(forward);
}
}
```

ViewVirusDAO.java

```
package com.virholex.basicinfo;

import java.sql.Connection;

public class ViewVirusDAO {

  public static ArrayList<String>
  checkVirus(char letter) throws SQLException,
  NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    ArrayList<String> virusList = new
    ArrayList<String>();

    try {
      String sql = "SELECT * FROM virus WHERE
      virus LIKE '" + letter + "%'";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.execute(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        virusList.add(rs.getString("virus"));
      }
    } finally {
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return virusList;
  }

  public static ArrayList<VirusVO>
  getVirus(VirusVO vo, String letter) throws
  SQLException, NamingException {

    Connection conn = null;
    Statement s = null;
    ResultSet rs = null;
    VirusVO virusVO = new VirusVO();
    ArrayList<VirusVO> virusList = new
    ArrayList<VirusVO>();

    try {
      String sql = "SELECT * FROM virus WHERE
      virus LIKE '" + letter + "%' ORDER BY
      virus";
      conn = DBConnect.getConnection();
      s = conn.createStatement();
      s.execute(sql);
      rs = s.getResultSet();

      while(rs.next()) {
        virusVO.setVirusName(rs.getString("viru
        s"));
        virusVO.setDescription(rs.getString("de
        scription"));
        virusList.add(virusVO);
        virusVO = new VirusVO();
      }
    } finally {
      if(rs != null) rs.close();
      if(s != null) s.close();
      if(conn != null) conn.close();
    }

    return virusList;
  }

  public static VirusVO
  readVirusDetails(VirusVO vo) throws
  SQLException, NamingException {

    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
      String sql = "SELECT * FROM
      basic_virus_info WHERE common_name=?";
      conn = DBConnect.getConnection();
      ps = conn.prepareStatement(sql);
      ps.setString(1, vo.getVirusName());
      rs = ps.executeQuery();

      while(rs.next()) {
        vo.setScientificName(rs.getString("scie
        ntific_name"));
        vo.setCommonName(rs.getString("common_n
        ame"));
        vo.setSerotype(rs.getString("serotype")
        );
        vo.setHost(rs.getString("host"));
        vo.setVirusArchitecture(rs.getString("v
        irus_architecture"));
        vo.setInfectivity(rs.getString("infecti
        vity"));
        vo.setMannerOfInfection(rs.getString("m
        anner_of_infection"));
        vo.setVector(rs.getString("vector"));
        vo.setAntimicrobialTreatment(rs.getStri
        ng("antimicrobial_treatment"));
        vo.setReceptors(rs.getString("receptors
        "));
        vo.setVirusId(rs.getInt("info_id"));
      }
    } finally {
      if(rs != null) rs.close();
      if(ps != null) ps.close();
      if(conn != null) conn.close();
    }

    return vo;
  }
```

```
public static String getImage(String virus)
throws SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String image = "";

  try {
    String sql = "SELECT image FROM virus
    WHERE virus=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, virus);
    rs = ps.executeQuery();

    if(rs.next()) {
      image = rs.getString("image");
    }
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return image;
}

public static boolean
isHotspotManager(String username) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  boolean isManager = false;

  try {
    String sql = "SELECT username FROM
    user_prev WHERE username=? AND
    role_name='Hotspot Manager'";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    rs = ps.executeQuery();

    if(rs.next()) {
      isManager = true;
    }
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return isManager;
}

public static String getProjectRole(String
username, String project) throws
SQLException, NamingException {

  Connection conn = null;
  PreparedStatement ps = null;
  ResultSet rs = null;
  String role = "";

  try {
    String sql = "SELECT role_name FROM
    user_prev WHERE username=? AND
    involvement=?";
    conn = DBConnect.getConnection();
    ps = conn.prepareStatement(sql);
    ps.setString(1, username);
    ps.setString(2, project+"
    "+VirholexConstants.VIRHO_HOTSPOT);
    rs = ps.executeQuery();

    if(rs.next()) {
      role = rs.getString(1);
    }
  } finally {
    if(rs != null) rs.close();
    if(ps != null) ps.close();
    if(conn != null) conn.close();
  }

  return role;
 }

}
```

ViewVirusDetailsAction.java

```
package com.virholex.basicinfo;

import java.io.File;

public class ViewVirusDetailsAction extends
Action {

@Override
public ActionForward execute(ActionMapping
mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws
Exception {
  VirusForm virusForm = (VirusForm) form;
  VirusVO virusVO = new VirusVO();
  VirhoBasicInfoWorker basicInfoWorker = new
  VirhoBasicInfoWorker();
  HttpSession session =
  request.getSession(true);
  ActionErrors errors = new ActionErrors();
  FileOutputStream outputStream = null;
  FormFile formFile = null;

  String username =
  (String)session.getAttribute("userLogin");
  String forward =
  "virholex.basicinfo.viewVirusDetails";
  String virus = "", image = "", roleName =
  "";
  int privilege = 0;

  if(request.getParameter("virus") != null)
    virus = request.getParameter("virus");
  else
    virus =
    (String)session.getAttribute("virus");

  if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Edit Virus"))
  {
    forward = "virholex.basicinfo.editVirus";
  } else if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("View
  Members")) {
    forward = "virholex.basicinfo.viewUsers";
  } else if(virusForm.getCommand()!=null &&
  virusForm.getCommand().equals("Send
  Privilege Request")) {
    session.setAttribute("project", virus);
    forward =
    "virholex.virus.requestPrivilege";
  } else {
    PropertyUtils.copyProperties(virusVO,
    virusForm);
    virusVO.setVirusName(virus);
```

```
virusVO =
(VirusVO)basicInfoWorker.readVirus(virusVO)
;

if(virusForm.getCommand()!=null &&
virusForm.getCommand().equals("Upload") &&
!virusForm.getFile().toString().equals(""))
{
  errors =
  virusForm.validateUploadSVG(mapping,
  request);

  if(errors.isEmpty()) {
    try {
      // Save file
      formFile = virusVO.getFile();
      String filename =
      Utilities.formatFileName(formFile);
      String filePath =
      getServlet().getServletContext().getRe
      alPath(VirholexConstants.HOTSPOTS_REPO
      SITORY) + File.separator + filename;
      outputStream = new
      FileOutputStream(new File(filePath));
      outputStream.write(formFile.getFileDat
      a());
      Utilities.resizeImage(request,
      filePath);
      virusVO.setImageDB(filename);
      basicInfoWorker.uploadSVG(virusVO);

    } catch(FileNotFoundException e) {
      e.printStackTrace();
    } catch(IOException e) {
      e.printStackTrace();
    } finally {
      outputStream.close();
    }
  } else {
    saveErrors(request, errors);
  }
}

image = basicInfoWorker.getImage(virus);

if(virusForm.getCommand()!=null &&
virusForm.getCommand().equals("Delete
Virus")) {
  Utilities.deleteFile(request,
  VirholexConstants.HOTSPOTS_REPOSITORY,
  image);
  basicInfoWorker.deleteVirus(virus,
  image);
  Utilities.updateVirusList(request);

  forward = "virholex.basicinfo.viewVirus";
  ActionRedirect redirect = new
  ActionRedirect(mapping.findForward(forwar
  d));
  redirect.addParameter("list", "");
  return redirect;
} else if(virusForm.getCommand()!=null &&
virusForm.getCommand().equals("here")) {
  // delete the image from the repository
  Utilities.deleteFile(request,
  VirholexConstants.HOTSPOTS_REPOSITORY,
  image);
  basicInfoWorker.deleteImage(virus,
  image);
  Utilities.updateVirusList(request);
} else {
  if(!Utilities.isEmpty(image)) {
    virusVO.setHasImage(true);
    virusVO.setImageDB(image);
    request.setAttribute("result", image);
  }
    }
    // Check if the user is a Hotspot Manager
    roleName =
    basicInfoWorker.getProjectRole(username,
    virus);

    if(roleName.equals(VirholexConstants.HOTSPO
    T_MANAGER_DESC))
      privilege =
      VirholexConstants.HOTSPOT_MANAGER;
    else
      privilege =
      VirholexConstants.REGISTERED_USER;

    request.setAttribute("privilege",
    privilege);

  }

  PropertyUtils.copyProperties(virusForm,
  virusVO);

  session.setAttribute("virus", virus);
  session.setAttribute("module",
  VirholexConstants.VIRHO_HOTSPOTS);

  return mapping.findForward(forward);
}
}
```

VirhoBasicInfoWorker.java

```
package com.virholex.basicinfo;

import java.util.ArrayList;

public class VirhoBasicInfoWorker implements
Worker {

  public VirusVO readVirus(VirusVO vo) throws
  Exception {
    return
    (VirusVO)ViewVirusDAO.readVirusDetails(vo);
  }

  public ArrayList<String> checkVirus(char
  letter) throws Exception {
    return ViewVirusDAO.checkVirus(letter);
  }

  public ArrayList<VirusVO> getVirus(VirusVO
  vo, String letter) throws Exception {
    return ViewVirusDAO.getVirus(vo, letter);
  }

  public boolean isManager(String username)
  throws Exception {
    return
    ViewVirusDAO.isHotspotManager(username);
  }

  public String getProjectRole(String
  username, String project) throws Exception {
    return
    ViewVirusDAO.getProjectRole(username,
    project);
  }

  public boolean checkVirus(String virus)
  throws Exception {
    return AddVirusDAO.checkVirus(virus);
  }
```

```
  public void insertVirus(VirusVO vo) throws
  Exception {
    AddVirusDAO.insertVirus(vo);
  }

  public void updateVirus(VirusVO vo) throws
  Exception {
    EditVirusDAO.updateVirus(vo);
  }

  public String getVirusDescription(String
  virus) throws Exception {
    return
    EditVirusDAO.getVirusDescription(virus);
  }

  public void uploadSVG(VirusVO vo) throws
  Exception {
    AddVirusDAO.uploadVirusImage(vo);
  }

  public String getImage(String virus) throws
  Exception {
    return ViewVirusDAO.getImage(virus);
  }

  public void deleteImage(String virus, String
  image) throws Exception {
    EditVirusDAO.deleteImage(virus, image);
  }

  public void deleteVirus(String virus, String
  image) throws Exception {
    EditVirusDAO.deleteVirus(virus, image);
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VirhoInternalSearchWorker.java

```
package com.virholex.internalsearch;

import java.util.ArrayList;

public class VirhoInternalSearchWorker
implements Worker {

  public ArrayList<ViewReferenceDetailsVO>
  searchReferences(String key, String sortBy)
  throws Exception {
    return SearchDAO.searchReferences(key,
    sortBy);
  }

  public ArrayList<ViewReferenceDetailsVO>
  advancedSearchReferences(SearchVO vo, String
  sortBy) throws Exception {
    return
    SearchDAO.advancedSearchReferences(vo,
    sortBy);
  }

  public ArrayList<ViewExperimentDetailsVO>
  searchExperiments(String key, String sortBy)
  throws Exception {
    return SearchDAO.searchExperiments(key,
    sortBy);
  }

  public ArrayList<ViewExperimentDetailsVO>
  advancedSearchExperiments(SearchVO vo,
  String sortBy) throws Exception {
    return
    SearchDAO.advancedSearchExperiments(vo,
    sortBy);
  }

  public ArrayList<ViewImageDetailsVO>
  searchImages(String key, String sortBy)
  throws Exception {
    return SearchDAO.searchImages(key, sortBy);
  }

  public ArrayList<ViewImageDetailsVO>
  advancedSearchImages(SearchVO vo, String
  sortBy) throws Exception {
    return SearchDAO.advancedSearchImages(vo,
    sortBy);
  }

  public ArrayList<ViewModelDetailsVO>
  searchModels(String key, String sortBy)
  throws Exception {
  return SearchDAO.searchModels(key, sortBy);
    }

  public ArrayList<ViewModelDetailsVO>
  advancedSearchModels(SearchVO vo, String
  sortBy) throws Exception {
    return SearchDAO.advancedSearchModels(vo,
    sortBy);
  }

  public VO read(VO vo) throws Exception {
    // TODO Auto-generated method stub
    return null;
  }

  public int store(VO vo) {
    // TODO Auto-generated method stub
    return 0;
  }

}
```

VirusForm.java

```
package com.virholex.basicinfo;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionMessage;
import org.apache.struts.upload.FormFile;

import com.virholex.general.Utilities;

public class VirusForm extends ActionForm {

  private static final long serialVersionUID =
  -4735625968524520 21L;

  private String virusName;
  private String commonName;
```

```
private String scientificName;
private String serotype;
private String host;
private String virusArchitecture;
private String infectivity;
private String mannerOfInfection;
private String vector;
private String antimicrobialTreatment;
private String receptors;
private String description;
private String dateAdded;
private String dateModified;
private String status;
private int virusId;
private String command;
private String author;
private boolean hasVirus;
private String commonNameDB;
private String descriptionDB;
private FormFile file;
private String image;
private String imageDB;
private boolean hasImage;

public void setVirusName(String virusName) {
  this.virusName = virusName;
}
public String getVirusName() {
  return virusName;
}
public void setCommonName(String commonName)
{
  this.commonName = commonName;
}
public String getCommonName() {
  return commonName;
}
public void setScientificName(String
scientificName) {
  this.scientificName = scientificName;
}
public String getScientificName() {
return scientificName;
  }
public void setSerotype(String serotype) {
  this.serotype = serotype;
}
public String getSerotype() {
  return serotype;
}
public void setHost(String host) {
this.host = host;
  }
public String getHost() {
return host;
  }
public void setVirusArchitecture(String
virusArchitecture) {
  this.virusArchitecture = virusArchitecture;
}
public String getVirusArchitecture() {
return virusArchitecture;
  }
public void setInfectivity(String
infectivity) {
  this.infectivity = infectivity;
}
public String getInfectivity() {
  return infectivity;
}
public void setMannerOfInfection(String
mannerOfInfection) {
  this.mannerOfInfection = mannerOfInfection;
}
public String getMannerOfInfection() {
  return mannerOfInfection;
}
public void setVector(String vector) {
  this.vector = vector;
}
public String getVector() {
return vector;
  }
public void setAntimicrobialTreatment(String
antimicrobialTreatment) {
  this.antimicrobialTreatment =
  antimicrobialTreatment;
}
public String getAntimicrobialTreatment() {
  return antimicrobialTreatment;
}
public void setReceptors(String receptors) {
this.receptors = receptors;
  }
public String getReceptors() {
  return receptors;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
return description;
  }
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
return dateAdded;
  }
public void setDateModified(String
dateModified) {
  this.dateModified = dateModified;
}
public String getDateModified() {
  return dateModified;
}
public void setStatus(String status) {
this.status = status;
  }
public String getStatus() {
  return status;
}
public void setVirusId(int virusId) {
  this.virusId = virusId;
}
public int getVirusId() {
  return virusId;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setCommand(String command) {
  this.command = command;
}
public String getCommand() {
  return command;
}
public void setHasVirus(boolean hasVirus) {
  this.hasVirus = hasVirus;
}
public boolean isHasVirus() {
  return hasVirus;
}
public void setCommonNameDB(String
commonNameDB) {
  this.commonNameDB = commonNameDB;
}
public String getCommonNameDB() {
```

```
  return commonNameDB;
}
public void setDescriptionDB(String
descriptionDB) {
  this.descriptionDB = descriptionDB;
}
public String getDescriptionDB() {
  return descriptionDB;
}
public void setFile(FormFile file) {
  this.file = file;
}
public FormFile getFile() {
  return file;
}
public void setImage(String image) {
  this.image = image;
}
public String getImage() {
  return image;
}
public void setImageDB(String imageDB) {
  this.imageDB = imageDB;
}
public String getImageDB() {
  return imageDB;
}
public void setHasImage(boolean hasImage) {
  this.hasImage = hasImage;
}
public boolean isHasImage() {
  return hasImage;
}

public ActionErrors validate(ActionMapping
mapping, HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if
  (Utilities.isEmpty(this.getCommonName())) {
    errors.add("errorMessage", new
    ActionMessage("error.required.asterisk"))
    ;
    errors.add("commonName", new
    ActionMessage(""));
  } else {
    if(this.isHasVirus()) {
      errors.add("commonName", new
      ActionMessage("error.notAvailable", "
      virus"));
    }
  }

  return errors;
}

public ActionErrors
validateUploadSVG(ActionMapping mapping,
HttpServletRequest request) {

  ActionErrors errors = new ActionErrors();

  if
  (!Utilities.isEmpty(this.getFile().toString
  ())) {
    if(!(Utilities.getFileExtension(this.getF
    ile().toString()).equals("jpg") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("JPG") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("jpeg") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("JPEG") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("gif") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("GIF") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("png") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("PNG") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("bmp") ||
    Utilities.getFileExtension(this.getFile()
    .toString()).equals("BMP"))) {
      errors.add("file", new
      ActionMessage("error.image"));
    }
  }

  return errors;
 }

}
```

VirusVO.java

```
package com.virholex.basicinfo;

import org.apache.struts.upload.FormFile;

public class VirusVO implements VO {

  private static final long serialVersionUID =
  4288334405363057456L;

  private String virusName;
  private String commonName;
  private String scientificName;
  private String serotype;
  private String host;
  private String virusArchitecture;
  private String infectivity;
  private String mannerOfInfection;
  private String vector;
  private String antimicrobialTreatment;
  private String receptors;
  private String description;
  private String dateAdded;
  private String dateModified;
  private String status;
  private int virusId;
  private String author;
  private boolean hasVirus;
  private String commonNameDB;
  private String descriptionDB;
  private FormFile file;
  private String image;
  private String imageDB;
  private boolean hasImage;

  public void setVirusName(String virusName) {
    this.virusName = virusName;
  }
  public String getVirusName() {
    return virusName;
  }
  public void setCommonName(String commonName)
  {
    this.commonName = commonName;
  }
  public String getCommonName() {
  return commonName;
    }
  public void setScientificName(String
  scientificName) {
    this.scientificName = scientificName;
  }
  public String getScientificName() {
```

```
  return scientificName;
}
public void setSerotype(String serotype) {
  this.serotype = serotype;
}
public String getSerotype() {
  return serotype;
}
public void setHost(String host) {
  this.host = host;
}
public String getHost() {
  return host;
}
public void setVirusArchitecture(String
virusArchitecture) {
  this.virusArchitecture = virusArchitecture;
}
public String getVirusArchitecture() {
  return virusArchitecture;
}
public void setInfectivity(String
infectivity) {
  this.infectivity = infectivity;
}
public String getInfectivity() {
  return infectivity;
}
public void setMannerOfInfection(String
mannerOfInfection) {
  this.mannerOfInfection = mannerOfInfection;
}
public String getMannerOfInfection() {
  return mannerOfInfection;
}
public void setVector(String vector) {
  this.vector = vector;
}
public String getVector() {
  return vector;
}
public void setAntimicrobialTreatment(String
antimicrobialTreatment) {
  this.antimicrobialTreatment =
  antimicrobialTreatment;
}
public String getAntimicrobialTreatment() {
  return antimicrobialTreatment;
}
public void setReceptors(String receptors) {
  this.receptors = receptors;
}
public String getReceptors() {
  return receptors;
}
public void setDescription(String
description) {
  this.description = description;
}
public String getDescription() {
  return description;
}
public void setDateAdded(String dateAdded) {
  this.dateAdded = dateAdded;
}
public String getDateAdded() {
  return dateAdded;
}
public void setDateModified(String
dateModified) {
  this.dateModified = dateModified;
}
public String getDateModified() {
  return dateModified;
}
public void setStatus(String status) {
  this.status = status;
}
public String getStatus() {
  return status;
}
public void setVirusId(int virusId) {
  this.virusId = virusId;
}
public int getVirusId() {
  return virusId;
}
public void setAuthor(String author) {
  this.author = author;
}
public String getAuthor() {
  return author;
}
public void setHasVirus(boolean hasVirus) {
  this.hasVirus = hasVirus;
}
public boolean isHasVirus() {
  return hasVirus;
}
public void setCommonNameDB(String
commonNameDB) {
  this.commonNameDB = commonNameDB;
}
public String getCommonNameDB() {
  return commonNameDB;
}
public void setDescriptionDB(String
descriptionDB) {
  this.descriptionDB = descriptionDB;
}
public String getDescriptionDB() {
  return descriptionDB;
}
public void setFile(FormFile file) {
  this.file = file;
}
public FormFile getFile() {
  return file;
}
public void setImage(String image) {
  this.image = image;
}
public String getImage() {
  return image;
}
public void setImageDB(String imageDB) {
  this.imageDB = imageDB;
}
public String getImageDB() {
  return imageDB;
}
public void setHasImage(boolean hasImage) {
  this.hasImage = hasImage;
}
public boolean isHasImage() {
  return hasImage;
}

}
```

GENERAL FILES

DBConnect.java

```
package com.virholex.general;

import javax.naming.Context;

public class DBConnect {
```

```
  public DBConnect() {
    super();
  }

  public static Connection getConnection()
  throws NamingException, SQLException {
    Context ctx = new InitialContext();
    DataSource ds =
    (DataSource)ctx.lookup("java:comp/env/jdbc/
    virholexPool");
    Connection conn = ds.getConnection();
    return conn;
  }

}
```

SendEmail.java

```
package com.virholex.general;

import java.util.*;

public class SendEmail extends HttpServlet {

  private static final long serialVersionUID =
  1L;

  public SendEmail() {
    super();
  }

  public static void sendEmail (String to,
  String subject, String msg, String name)
  throws Exception {

    String from = "virholex@cas.upm.edu.ph";
    String host = "localhost";

    String msgHead = "Hi " + name + ",\n\n";
    String msgBody = msg;
    String msgFoot = "\n\nIf you have any
    questions, queries, or suggestions
    regarding this email or Virholex, please
    email virholexhelp@cas.upm.edu.ph.
    \n\n\Thank you.\n\n" + from;

    //get system properties
    Properties properties =
    System.getProperties();

    //setup mail server
    properties.setProperty("mail.smtp.host",
    host);

    //get default session object
    Session session =
    Session.getDefaultInstance(properties);

    //create a default MimeMessage object
    MimeMessage message = new
    MimeMessage(session);

    // set sender address
    message.setFrom(new InternetAddress(from));

    // set recipient
    message.addRecipient(Message.RecipientType.
    TO, new InternetAddress(to));

    // Set the "Subject" header field.
    message.setSubject(subject);

    Multipart fullMessage = new
    MimeMultipart();

    BodyPart head = new MimeBodyPart();
    head.setText(msgHead);

    BodyPart body = new MimeBodyPart();
    body.setText(msgBody);

    BodyPart foot = new MimeBodyPart();
    foot.setText(msgFoot);

    fullMessage.addBodyPart(head);
    fullMessage.addBodyPart(body);
    fullMessage.addBodyPart(foot);

    message.setContent(fullMessage);

    // Sets the given String as this part's
    content,
    // with a MIME type of "text/plain".
    // message.setText("This is a test!");

    // Send message
    Transport.send(message);
  }

}
```

Utilities.java

```
package com.virholex.general;

import java.awt.Graphics2D;
import java.awt.RenderingHints;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.SQLException;
import java.sql.Timestamp;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.Random;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.util.zip.Deflater;
import java.util.zip.ZipEntry;
import java.util.zip.ZipOutputStream;
import javax.naming.NamingException;
import javax.servlet.http.HttpServletRequest;

import org.apache.struts.upload.FormFile;

import com.virholex.basicinfo.VirusVO;
import com.virholex.virui.UpdatesDAO;

public class Utilities {

  public static void
  deleteFile(HttpServletRequest request,
  String repository, String fileName) {
    String filePath =
    request.getSession().getServletContext().ge
    tRealPath(repository) + File.separator +
    fileName;
    File file = new File(filePath);
    file.delete();
  }
```

```
public static String removeProtocol(String
link) {
  String newLink = link;
  if(link!=null && (link.contains("http://")
  || link.contains("https://"))) {
    if
    (link.substring(0,7).equals("http://"))
      newLink = link.substring(7);
    else if
    (link.substring(0,8).equals("https://"))
      newLink = link.substring(8);
  }

  return newLink;
}

public static String formatFileName(FormFile
formFile) {
  String filename = formFile.toString();

  // Generate random numbers that will be
  appended on file name
  Random generator = new Random();
  int randNum =
  Math.abs(generator.nextInt());

  int lastIndexOf =
  filename.lastIndexOf(".");
  String fileExtension =
  filename.substring(lastIndexOf);
  filename = filename.substring(0,
  lastIndexOf) + "_" +
  String.valueOf(randNum) + fileExtension;

  return filename;
}

public static String formatFileName(String
filename, String fileExtension){
  filename = filename.trim().replace(" ",
  "_");
  filename = filename + "_" + generateDate()
  + fileExtension;

  return filename;
}

public static String generateDate() {
  Date now = new Date();
  String DATE_FORMAT = "MMddyyyy_HHmm";
  SimpleDateFormat sdf = new
  SimpleDateFormat(DATE_FORMAT);
  String date = sdf.format(now);

  return date;
}

public static String
removeFileExtension(String file, String
fileExtension) {
  String newFile = "";
  if(file!=null &&
  file.contains(fileExtension)) {
    int lastIndex = file.lastIndexOf("_");
    newFile = file.substring(0, lastIndex) +
    fileExtension;
  }

  return newFile;
}

public static void
resizeImage(HttpServletRequest request,
String filePath) throws IOException {
  File file = new File(filePath);
  BufferedImage in =
  javax.imageio.ImageIO.read(file);
  final int MAXWIDTH = 350;
  int width = in.getWidth();
  int height = in.getHeight();

  if(width > MAXWIDTH){
    height = (int)Math.ceil((double)height *
    MAXWIDTH/(double)width);
  } else if(width < MAXWIDTH){
    height += (int)Math.ceil((double)height *
    MAXWIDTH/(double)width);
  }

  width = MAXWIDTH;

  BufferedImage out = scaleImage(in,
  BufferedImage.TYPE_INT_RGB, width, height);
  javax.imageio.ImageIO.write(out, "JPG",
  file);
}

public static BufferedImage
scaleImage(BufferedImage image, int
imageType, int newWidth, int newHeight)
  // Make sure the aspect ratio is
  maintained, so the image is not distorted
  double thumbRatio = (double) newWidth /
  (double) newHeight;
  int imageWidth = image.getWidth(null);
  int imageHeight = image.getHeight(null);
  double aspectRatio = (double) imageWidth /
  (double) imageHeight;

  if (thumbRatio < aspectRatio) {
    newHeight = (int) (newWidth /
    aspectRatio);
  } else {
    newWidth = (int) (newHeight *
    aspectRatio);
  }

  // Draw the scaled image
  BufferedImage newImage = new
  BufferedImage(newWidth, newHeight,
  imageType);
  Graphics2D graphics2D =
  newImage.createGraphics();
  graphics2D.setRenderingHint(RenderingHints.
  KEY_INTERPOLATION,
  RenderingHints.VALUE_INTERPOLATION_BILINEAR
  );
  graphics2D.drawImage(image, 0, 0, newWidth,
  newHeight, null);

  return newImage;
}

public static boolean isEmpty(String str) {
  return (str == null || str.trim().length()
  < 1);
}

public static String getFileExtension(String
file) {
  int lastIndexOf = file.lastIndexOf(".");
  return file.substring(lastIndexOf+1);
}

public static boolean validInput(String str,
int type) {
  Pattern pattern = null;
  Matcher matcher;

  switch(type) {
```

```
    case 1: pattern =
    Pattern.compile(VirholexConstants.USERNAM
    E_PATTERN);
      break;
    case 2: pattern =
    Pattern.compile(VirholexConstants.PASSWOR
    D_PATTERN);
      break;
    case 3: pattern =
    Pattern.compile(VirholexConstants.EMAIL_P
    ATTERN);
      break;
    case 4: pattern =
    Pattern.compile(VirholexConstants.INTEGER
    _PATTERN);
      break;
    case 5: pattern =
    Pattern.compile(VirholexConstants.DOUBLE_
    PATTERN);
      break;
    default:System.out.println("ERROR: NO
    PATTERN");
      break;
  }

  matcher = pattern.matcher(str);

  return matcher.matches();
  }

public static String dateFormat(String date)
{
  if(!(isEmpty(date) || date.contains("0000-
  00-00 00:00:00"))) {
    Timestamp ts = Timestamp.valueOf(date);
    SimpleDateFormat formatter = new
    SimpleDateFormat("MMMMM dd, yyyy");
    date = formatter.format(ts);
  } else {
    date = "";
  }

  return date;
}

public static String
dateCalendarFormat(String date) {
  String newDate = "";

  if(!(isEmpty(date) || date.contains("0000-
  00-00 00:00:00"))) {
    int firstIndex = date.indexOf("-");
    int lastIndex = date.lastIndexOf("-");
    newDate = date.substring(firstIndex+1,
    lastIndex) + "/";
    newDate += date.substring(lastIndex+1,
    date.indexOf(" ")) + "/";
    newDate += date.substring(0, firstIndex);
    date = newDate;
  } else {
    date = "";
  }

  return date;
}

public static String
generateTitle(ArrayList<String> titleDBList,
String title) {
  int ctr = 0, start = 0, end = 0;
  String titleDB = "";

  for(int i=0; i<titleDBList.size(); i++) {
    titleDB = titleDBList.get(i);
    start = titleDB.lastIndexOf("(");
    end = titleDB.lastIndexOf(")");
    try {
      if(titleDB.equals(title)) {
        ctr++;
      } else
      if(Integer.valueOf(titleDB.substring(st
      art+1, end)) > 0) {
        ctr = Math.max(ctr,
        Integer.valueOf(titleDB.substring(star
        t+1, end))+1);
      }
    } catch(NumberFormatException e) {
      e.printStackTrace();
    }
  }

  return (ctr > 0) ? title + " (" + ctr + ")"
  : title;
}

public static void
copyFile(HttpServletRequest request, String
file, String repository, String
tempRepository) {
  InputStream inStream = null;
  OutputStream outStream = null;

  try{
    File afile = new
    File(request.getSession().getServletConte
    xt().getRealPath(repository) +
    File.separator + file);
    File bfile = new
    File(request.getSession().getServletConte
    xt().getRealPath(tempRepository) +
    File.separator + file);
    inStream = new FileInputStream(afile);
    outStream = new FileOutputStream(bfile);

    byte[] buffer = new byte[1024];
    int length;
    //copy the file content in bytes
    while ((length = inStream.read(buffer)) >
    0){
      outStream.write(buffer, 0, length);
    }

    inStream.close();
    outStream.close();

    System.out.println("File is copied
    successful!");
  }catch(IOException e){
    e.printStackTrace();
  }

}

public static String
zipFile(HttpServletRequest request,
ArrayList<String> files, String file, String
repository) {
  byte[] buffer = new byte[18024];
  String zipFileName = formatFileName(file,
  ".zip");
  String zipFilePath =
  request.getSession().getServletContext().ge
  tRealPath(repository) + File.separator +
  zipFileName;
  int len = 0;

  try {
    ZipOutputStream out = new
    ZipOutputStream(new
    FileOutputStream(zipFilePath));
```

```
    out.setLevel(Deflater.DEFAULT_COMPRESSION
    );

    for (int i = 0; i < files.size(); i++){
      String pathname =
      request.getSession().getServletContext(
      ).getRealPath(repository) +
      File.separator + files.get(i);

      if((new File(pathname)).exists()) {
        FileInputStream in = new
        FileInputStream(pathname);
        out.putNextEntry(new
        ZipEntry(files.get(i)));
        while ((len = in.read(buffer)) > 0){
          out.write(buffer, 0, len);
        }
        out.closeEntry();
        in.close();
      } else {
        System.out.println("File " + pathname
        + " is not in the images
        repository.");
      }
    }

    out.close();
  } catch(IllegalArgumentException e) {
    e.printStackTrace();
  } catch(FileNotFoundException e) {
    e.printStackTrace();
  } catch(IOException e) {
    e.printStackTrace();
  }

  return zipFileName;
}

public static String formatDate(String date)
{
  String newDate = "0000-00-00";

  if(!isEmpty(date)) {
    int firstIndex = date.indexOf("/");
    int lastIndex = date.lastIndexOf("/");

    newDate = date.substring(lastIndex+1) +
    "-";

    if(firstIndex == 1)
      newDate = newDate + "0" +
      date.substring(0, firstIndex) + "-";
    else
      newDate = newDate + date.substring(0,
      firstIndex) + "-";
    if(lastIndex - firstIndex == 2)
      newDate = newDate + "0" +
      date.substring(firstIndex+1,
      lastIndex);
    else
      newDate = newDate +
      date.substring(firstIndex+1,
      lastIndex);
  }

  return newDate;
}

public static String formatTime(String hour,
String minute, String second) {
  String time = "";

  if(hour.length() == 1)
    time = "0" + hour + ":";
  else
    time = hour + ":";

  if(minute.length() == 1)
    time = time + "0" + minute + ":";
  else
    time = time + minute + ":";
  if(second.length() == 1)
    time = time + "0" + second;
  else
    time = time + second;

  return time;
}

public static void
updateVirusList(HttpServletRequest request)
throws SQLException, NamingException {
UpdatesDAO updatesDAO = new UpdatesDAO();
ArrayList<VirusVO> updates = new
ArrayList<VirusVO>();
  updates = updatesDAO.readVirusUpdates();

  if(!updates.isEmpty()) {
    for(int i=0; i<updates.size(); i++) {
      String conv1 =
      ((VirusVO)updates.get(i)).getDateAdded(
      );
      String conv2 =
      ((VirusVO)updates.get(i)).getDateModifi
      ed();
      Timestamp ts1 =
      Timestamp.valueOf(conv1);
      Timestamp ts2 =
      Timestamp.valueOf(conv2);
      long time1 = ts1.getTime();
      long time2 = ts2.getTime();
      @SuppressWarnings("deprecation")
      long date1 = ts1.getDate();
      @SuppressWarnings("deprecation")
      long date2 = ts2.getDate();

      if(date1 == date2) {
        if(time1 == time2)
          ((VirusVO)updates.get(i)).setStatus(
          "added");
        else if(time2 != time1)
          ((VirusVO)updates.get(i)).setStatus(
          "modified");
      } else if(date2 != date1) {
        ((VirusVO)updates.get(i)).setStatus("m
        odified");
      }
    }
  }

  request.getSession().setAttribute("updates"
  , updates);
 }

}
```

VirholexConstants.java

```
package com.virholex.general;

public class VirholexConstants {

  public static final String TRUE = "true";
  public static final String FALSE = "false";

  public static final int
  IMAGE_SET_CONTRIBUTOR = 3;
  public static final int
  IMAGE_SET_COORDINATOR = 4;
```

```
public static final int MODEL_CONTRIBUTOR =
3;
public static final int MODEL_COORDINATOR =
4;

public static final int
COLLECTION_COORDINATOR = 4;
public static final int
COLLECTION_CONTRIBUTOR = 3;

public static final int
EXPERIMENT_LEAD_INVESTIGATOR = 4;
public static final int
EXPERIMENT_INVESTIGATOR = 3;

public static final int HOTSPOT_MANAGER = 4;

public static final int RESTRICTED_USER = 2;
public static final int REGISTERED_USER = 1;
public static final int GUESS_ANONYMOUS = 0;

public static final String
RESTRICTED_USER_DESC = "Restricted User";

public static final String
COLLECTION_COORDINATOR_DESC = "Collection
Coordinator";
public static final String
COLLECTION_CONTRIBUTOR_DESC = "Collection
Contributor";

public static final String
IMAGE_SET_COORDINATOR_DESC = "Image Set
Coordinator";
public static final String
IMAGE_SET_CONTRIBUTOR_DESC = "Image Set
Contributor";

public static final String
MODEL_CONTRIBUTOR_DESC = "Model
Contributor";
public static final String
MODEL_COORDINATOR_DESC = "Model
Coordinator";

public static final String
EXPERIMENT_LEAD_INVESTIGATOR_DESC =
"Experiment Lead Investigator";
public static final String
EXPERIMENT_INVESTIGATOR_DESC = "Experiment
Investigator";

public static final String
HOTSPOT_MANAGER_DESC = "Hotspot Manager";

public static final String VIRHO_MODELS =
"VirhoModels";
public static final String VIRHO_REFERENCES
= "VirhoReferences";
public static final String VIRHO_EXPERIMENTS
= "VirhoExperiments";
public static final String VIRHO_IMAGES =
"VirhoImages";
public static final String VIRHO_HOTSPOTS =
"VirhoHotspots";
public static final String VIRHO_BASICINFO =
"VirhoBasicInfo";

public static final String VIRHO_REFERENCE =
"VirhoReference";
public static final String VIRHO_HOTSPOT =
"VirhoHotspot";

public static final String DEFAULT_VALUE =
"Default";

 public static final String DEFAULT_PROJECT =
 "DEFAULT_PROJECT";
 public static final String REMOVE_PRIVILEGE
 = "Remove Privilege";

 public static final String STATUS_APPROVED =
 "approved";
 public static final String STATUS_DENIED =
 "denied";
 public static final String STATUS_SUSPENDED
 = "suspended";
 public static final String STATUS_REACTIVATE
 = "reactivate";
 public static final String STATUS_ACTIVATED
 = "activated";
 public static final String STATUS_ADMIN =
 "admin";
 public static final String STATUS_SUPERADMIN
 = "superadmin";

 public static final String PROJECT =
 "project";
 public static final String PROJECTSET =
 "projectset";

 public static final String USERNAME_PATTERN
 = "^[a-zA-Z0-9_-]{6,}$";
 public static final String PASSWORD_PATTERN
 = "^[a-zA-Z0-9@#$%_-]{6,}$";
 public static final String EMAIL_PATTERN =
 "^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-
 z0-9]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";
 public static final String DOUBLE_PATTERN =
 "^[0-9]+(\\.[0-9]+)$";
 public static final String INTEGER_PATTERN =
 "^[0-9]{0,}$";

 public static final String IMAGES_REPOSITORY
 = "/repository/virhoimages";
 public static final String
 IMAGES_TEMP_REPOSITORY =
 "/repository/virhoimages/temp";
 public static final String
 HOTSPOTS_REPOSITORY =
 "/repository/virhohotspots";
 public static final String
 REFERENCES_REPOSITORY =
 "/repository/virhoreferences";
 public static final String MODELS_REPOSITORY
 = "/repository/virhomodels";

}
```

VO.java

```
package com.virholex.general;

import java.io.Serializable;

public interface VO extends Serializable {

}
```

Worker.java

```
package com.virholex.general;

public interface Worker {

  public VO read(VO vo) throws Exception;
  public int store(VO vo);

}
```

C. jsp files

index.jsp

```
<jsp:forward page="Home.do" />
```

VIRHO USER INTERFACE

contactUs.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
    <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
      <jsp:include page="/rd_virui/sidebar.jsp"
      />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.contactUs"
            /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.contactUs" /></td>
        </tr>
        <tr>
       <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Kindly send your feedback, questions
            and suggestions to <span
            class="font-
            blue">virholexhelp@cas.upm.edu.ph</s
            pan>.
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
</html>
```

header.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<tr>
  <td class="banner" colspan="2">
    <html:img src="images/banner.png"
    width="1150" height="140" />
  </td>
</tr>
<tr class="login-index">
  <td colspan="2">
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin eq
    'superadmin'))}">
      Hi, <html:link page="/MyProfile.do"
      ><b>${userLogin}</b></html:link> &nb
      sp;|  
    </c:if>
    <c:choose>
      <c:when test="${userLogin eq null}">
        <html:link page="/Login.do"
        ><bean:message key="link.common.login"
        /></html:link>
          |  
        <html:link page="/Register.do"
        ><bean:message
        key="link.common.register"
        /></html:link>
          |  
        <html:link page="/ForgotPassword.do"
        ><bean:message
        key="link.common.forgotPassword"
        /></html:link>
            
      </c:when>
      <c:otherwise>
        <c:if test="${(statusLogin eq 'admin')
        or (statusLogin eq 'superadmin')}">
          <html:link page="/AccountRequest.do"
          ><bean:message
          key="link.common.request"
          /></html:link>
            |  
          <html:link page="/VirholexUsers.do"
          ><bean:message
          key="link.common.member"
          /></html:link>
            |  
        </c:if>
        <html:link page="/MyProjects.do"
        ><bean:message
        key="link.common.project"
        /></html:link>
          |  
```

```
      <html:link page="/MyProfile.do"
      ><bean:message
      key="link.common.profile"
      /></html:link>
        |  
      <html:link page="/Logout.do"
      ><bean:message key="link.common.logout"
      /></html:link>
          
    </c:otherwise>
  </c:choose>
 </td>
</tr>
```

index.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr class="content" valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <span class="breadcrumb-
      selected">Home</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or
    (statusLogin eq 'superadmin'))}">
     <html:form action="/QuickSearch"
     method="GET">
      <jsp:include
      page="/rd_virhointernalsearch/quickS
      earch.jsp" />
     </html:form>
    <tr>
     <td> </td>
    </tr>
    </c:if>
    <tr>
     <td class="content-title">VirHoLex
     (Virus-Host Interaction Lexicon)</td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-desc">
      VirHoLex is a system formulated by a
      research/study group composed of
      three academic institutions in the
      Philippines (University of the
      Philippines Manila, De La Salle
      University, Mapua Institute of
      Technology) to set up a Community
      ORiented Information (CORI) platform
      for virologists around the world.
      This initiative was born from the
      perspective of U. Reichl (Director,
      Max-Planck-Institute for the
      Dynamics of Complex Technical
      Systems, Magdeburg, Germany), J.
      Haas (LMU &amp;amp; Edinburgh University
      Medical Schools), J. R&auml;dler
      (LMU Faculty of Physics) and the
      author for the postdoctoral research
      of J. Bantang (on leave from the UPD
      National Institute of Physics). The
      project was inspired by EUCLIS
      (EUCLOCK Information System),
      another CORI system developed for
      the global community of
      chronobiologists. VirHoLex will
      provide information about three
      prevalent viruses in the country:
      Dengue, Influenza A, and Herpes
      Viruses.
      <p>The team is comprised of
      University of the Philippines
      graduating students mostly from the
      areas of computer science, molecular
      biology and mathematics. It is
      headed by Dr. Eduardo Mendoza of the
      Ludwig Maximilians Universität and
      the University of the
      Philippines.</p>
     </td>
    </tr>
   </table>
   </td>
  </tr>
  <tr class="footer">
   <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
   </td>
  </tr>
 </table>
</body>
</html>
```

sidebar.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ page
import="com.virholex.general.Utilities"  %>

<%
```

```
  if(null==session.getAttribute("updates") ||
  "".equals(session.getAttribute("updates")))
  {
    Utilities.updateVirusList(request);
  }
%>

<script type="text/javascript">
function DoNavigation(url) {
  document.location.href = url;
}
</script>

<table class="welcome-note">
<c:choose>
  <c:when test="${userLogin eq null}">
  <tr>
    <td class="welcome-title">Welcome to Virus-
    Host Interaction Lexicon</td>
  </tr>
  <tr>
    <td class="welcome-body">
      To access Virholex modules please log in
      or register.
      <html:link page="/VirholexPrivileges.do"
      >Why should you register?</html:link>
    </td>
  </tr>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="welcome-title">Welcome to Virus-
    Host Interaction Lexicon</td>
  </tr>
  <c:if test="${(statusLogin ne 'admin') and
  (statusLogin ne 'superadmin')}">
  <tr>
    <td class="welcome-body">
      For additional access rights, contact the
      administrator or submit a request.
    </td>
  </tr>
  </c:if>
  </c:otherwise>
</c:choose>
  <tr>
    <td> </td>
  </tr>
</table>
<table class="sidetabs">
<c:choose>
  <c:when test="${userLogin ne null}">
  <tr>
    <td
    onclick="DoNavigation('Home.do')"><html:lin
    k page="/Home.do" ><bean:message
    key="button.virho.home" /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('ViewProjects.do')"><
    html:link page="/ViewProjects.do"
    ><bean:message
    key="button.virho.experiments"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('ViewImageSets.do')">
    <html:link page="/ViewImageSets.do"
    ><bean:message key="button.virho.images"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('ViewModelsTaxonomy.d
    o')"><html:link
    page="/ViewModelsTaxonomy.do"
    ><bean:message key="button.virho.models"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('ViewCollections.do')
    "><html:link page="/ViewCollections.do"
    ><bean:message
    key="button.virho.references"
    /></html:link></td>
  </tr>
  </c:when>
  <c:otherwise>
  <tr>
    <td
    onclick="DoNavigation('Home.do')"><html:lin
    k page="/Home.do" ><bean:message
    key="button.virho.home" /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('VirhoExperiments.do'
    )"><html:link page="/VirhoExperiments.do"
    ><bean:message
    key="button.virho.experiments"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('VirhoImages.do')"><h
    tml:link page="/VirhoImages.do"
    ><bean:message key="button.virho.images"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('VirhoModels.do')"><h
    tml:link page="/VirhoModels.do"
    ><bean:message key="button.virho.models"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('VirhoReferences.do')
    "><html:link page="/VirhoReferences.do"
    ><bean:message
    key="button.virho.references"
    /></html:link></td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td
    onclick="DoNavigation('VirhoExternalSearch.
    do')"><html:link
    page="/VirhoExternalSearch.do"
    ><bean:message
    key="button.virho.externalInfo"
    /></html:link></td>
  </tr>
  <tr>
    <td
    onclick="DoNavigation('ContactUs.do')"><htm
    l:link page="/ContactUs.do" ><bean:message
    key="button.virho.contactUs"
    /></html:link></td>
  </tr>
</table>
<br/>
<table class="updatesbar">
  <tr>
    <td class="tab-updates" colspan="2">Virus
    Updates</td>
  </tr>
```

```
  <tr>
    <td>
    <table>
      <tr>
        <td colspan="2">
          <html:link page="/ViewVirus.do?list="
          >Click here to see the complete list
          of viruses</html:link>
        </td>
      </tr>
      <c:forEach var="update"
      items="${sessionScope.updates}">
      <tr>
        <td width="130px"><c:out
        value="${update.virusName}"/></td>
        <td><c:out
        value="${update.status}"/></td>
      </tr>
      </c:forEach>
    </table>
    </td>
  </tr>
  <tr>
    <td class="tab-updates-inv"> </td>
  </tr>
</table>
```

updatesbar.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<script src="./files/jquery-
1.8.0.js"></script>

<table class="updatesbar">
  <tr>
    <td class="tab-updates" colspan="2">Virus
    Updates</td>
  </tr>
  <tr>
    <td>
    <table>
      <tr>
        <td colspan="2">
          <html:link page="/ViewVirus.do?list="
          >Click here to see the complete list
          of viruses</html:link>
        </td>
      </tr>
      <c:forEach var="update"
      items="${sessionScope.updates}">
      <tr>
        <td width="130px"><c:out
        value="${update.virusName}"/></td>
        <td><c:out
        value="${update.status}"/></td>
      </tr>
      </c:forEach>
    </table>
    </td>
  </tr>
  <tr>
    <td class="tab-updates-inv"> </td>
  </tr>
</table>
```

virhoExperimentsDescription.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.experiments"
            /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.experiments" /></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Virho Experiments stores and manages
            data, metadata, and other relevant
            information on laboratory
            experiments. It allows the uploading
            of files such as images taken from
            experiments. It also includes tools
            for analysis of laboratory
            experiments, tabulation of
            laboratory experiments, image
            enhancement, and possibly for
            simulation of experiments.
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
```

```
</html>
```

virhoImagesDescription.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.images" /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.images" /></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Virho Images serves as a repository
            for images with their associated
            metadata for and from the users.
            Users may upload/download/search
            images that may be closely coupled
            with other modules in Virholex such
            as Virho Experiments, Virho Models,
            etc. Management of images, image
            meta-data, image analysis and
            analysis 15 results is done through
            the Open Microscopy Environment
            (OME), a set of software that
            interacts with a database for these
            functions.
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
</html>
```

virhoModelsDescription.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.models" /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.models" /></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Virho Models is a repository of
            summary descriptions of relevant
            models/modeling studies.
          </td>
        </tr>
      </table>
      </td>
```

```
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
</html>
```

virhoReferencesDescription.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.references"
            /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.references" /></td>
        </tr>
        <tr>
       <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Virho References is a repository of
            bibliographic entries or references
            of system virologists, which they
            can share among themselves. ShaRef
            will be used in managing this
            system.
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
</html>
```

VIRHO REGISTERED USER SERVICES

accountRequest.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="button.virho.references"
            /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-
          title"><bean:message
          key="button.virho.references" /></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
```

```
          Virho References is a repository of
          bibliographic entries or references
          of system virologists, which they
          can share among themselves. ShaRef
          will be used in managing this
          system.
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
</body>
</html>
```

accountRequestConfig.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/AccountRequestEvaluate"
 method="POST" >
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
   <tr>
     <td> </td>
   </tr>
   <tr valign="top">
     <td class="sidebar">
       <jsp:include
       page="/rd_virui/sidebar.jsp" />
     </td>
     <td>
     <c:if test="${(userLogin ne null) and
     ((statusLogin eq 'activated') or
     (statusLogin eq 'admin') or (statusLogin
     eq 'superadmin'))}">
     <table class="content" width="785px">
       <tr>
         <td class="breadcrumb">
           <html:link
           page="/Home.do"><bean:message
           key="button.virho.home"
           /></html:link> &#187;
           <span class="breadcrumb-
           selected">Account Request</span>
         </td>
       </tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Account Request</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td class="font-reg"
      height="40px"><b>Step 3 : Add
      User's Privileges</b></td>
    </tr>
    <tr>
      <td>
      <html:hidden property="nextStep"
      value="3"/>
      <table class="table-content">
        <tr>
          <th>Name</th>
          <th width="120px">Experiment
          Lead Investigator</th>
          <th width="120px">Image Set
          Coordinator</th>
          <th width="120px">Collection
          Coordinator</th>
          <th width="120px">Hotspot
          Manager</th>
        </tr>
       <c:choose>
        <c:when test="${not empty
        AccountRequestForm.users}">
        <c:forEach var="user"
        items="${AccountRequestForm.users
        }">
        <tr>
          <td>
            <c:out
            value="${user.firstName}"/>&nb
            sp;<c:out
            value="${user.lastName}"/>
            <input type="hidden"
            name="actionUser"
            value="${user.username}" />
          </td>
          <td align="center">
            <input type="checkbox"
            name="experiment${user.usernam
            e}" />
          </td>
          <td align="center">
            <input type="checkbox"
            name="image${user.username}"
            />
          </td>
          <td align="center">
            <input type="checkbox"
            name="collection${user.usernam
            e}" />
          </td>
          <td align="center">
            <input type="checkbox"
            name="hotspot${user.username}"
            />
          </td>
        </tr>
        </c:forEach>
        </c:when>
        <c:otherwise>
        <tr>
          <td colspan="5" class="font-
          italic">There are no users to
          be granted by privileges</td>
        </tr>
```

```
            </c:otherwise>
           </c:choose>
           </table>
           </td>
         </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td colspan="5">
              
          <html:submit property="command"
          value="Add Privileges"
          styleClass="button-long-blue" />
           
          <html:reset property="command"
          value="Reset" styleClass="button-
          long-blue" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

accountRequestEvaluate.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/AccountRequestEvaluate"
 method="POST" >
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
   <tr>
     <td> </td>
   </tr>
   <tr valign="top">
     <td class="sidebar">
       <jsp:include
       page="/rd_virui/sidebar.jsp" />
</td>
<td>
<c:if test="${(userLogin ne null) and
((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content" width="785px">
 <tr>
   <td class="breadcrumb">
     <html:link
     page="/Home.do"><bean:message
     key="button.virho.home"
     /></html:link> &#187;
     <span class="breadcrumb-
     selected">Account Request</span>
   </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
 <tr>
   <td class="content-title-form"
   colspan="2">Account Request</td>
 </tr>
 <tr>
   <td>
   <table class="content-desc-form">
     <tr>
       <td class="font-reg"
       height="40px"><b>Step 2 : Evaluate
       Accounts</b></td>
     </tr>
     <tr>
       <td>
       <table class="table-content">
         <tr>
           <th>Name</th>
           <th>Username</th>
           <th>Email Address</th>
           <th>Affiliation</th>
           <th>Action</th>
         </tr>
       <c:choose>
         <c:when test="${not empty
         AccountRequestForm.users}">
         <c:forEach var="user"
         items="${AccountRequestForm.users
         }">
         <tr>
           <td>${user.firstName} ${us
           er.lastName}</td>
           <td>${user.username}</td>
           <td>${user.emailAddress}</td>
           <td>${user.affiliation}</td>
           <td width="50px">
           <c:if
           test="${AccountRequestForm.requ
           estType eq 'PendingAccount'}">
             <html:hidden
             property="requestType"
             value="PendingAccount" />
             <input type="hidden"
             name="actionUser"
             value="${user.username}" />
             <select
             name="action${user.username}"
             class="font-black-small">
               <option value="No Change">No
               Change</option>
               <option
               value="Approve">Approve</opt
               ion>
               <option
               value="Deny">Deny</option>
             </select>
           </c:if>
```

```
            <c:if
            test="${AccountRequestForm.requ
            estType eq
            'ReactivateAccount'}">
              <html:hidden
              property="requestType"
              value="RequestAccount" />
              <input type="hidden"
              name="actionUser"
              value="${user.username}" />
              <select
              name="action${user.username}"
              class="font-black-small">
                <option value="No Change">No
                Change</option>
                <option
                value="Reactivate">Reactivat
                e</option>
              </select>
            </c:if>
            </td>
          </tr>
          </c:forEach>
          </c:when>
          <c:otherwise>
          <tr>
            <td colspan="5" class="font-
            italic">There are no users to
            be evaluated</td>
          </tr>
          </c:otherwise>
        </c:choose>
        </table>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td colspan="5">
          
      <html:submit property="command"
      value="Evaluate" styleClass="button-
      long-blue" />
       
      <html:reset property="command"
      value="Reset" styleClass="button-
      long-blue" />
    </td>
  </tr>
 </table>
 </c:if>
 </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

accountRequestFinal.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Account Request</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title">Account
          Request</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Done processing the pending
            requests. Click <html:link
            page="/VirholexUsers.do"
            >here</html:link> to see the list of
            Virholex users.
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
</body>
</html>
```

cancelAccount.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/UnsubscribeAccount"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link page="/MyProfile.do">My
            Profile</html:link> &#187;
            <span class="breadcrumb-
            selected">Account
            Unsubscription</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Account
          Unsubscription</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:160px">
            <tr>
              <td height="50px">
                You are about to cancel your
                VirhoLex account. This action
                cannot be undone when
                cancellation proceeds.
                <br/><br/>
                Please enter your password for
                verification
              </td>
            </tr>
            <tr>
              <td align="center">
              <table style="height:70px;"
              class="font-medium-reg">
                <tr>
                  <td width="100px"><bean:message
                  key="label.common.password"
                  /></td>
                  <td>
                    <html:password
                    property="password" size="35"
                    style="remove-error-field"
                    errorStyleClass="error-field"
                    errorKey="org.apache.struts.ac
                    tion.ERROR" />
                    <span class="error-
                    text"><html:errors
                    property="password" /></span>
                  </td>
                </tr>
                <tr>
                  <td align="center" colspan="2">
                     <html:submit
                     property="cancelAccount"
                     value="Cancel Account"
                     styleClass="button-long" />
                  </td>
                </tr>
              </table>
              </td>
            </tr>
          </table>
          </td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

editUserProfile.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  $(document).ready(function() {
    var pass = $("#editPassword").val();

    if(pass != 'true')
      $("#password").hide();

    $("#changePassword").click(function() {
    $("#password").slideToggle('slow',
    function() {});
    if($("#tableForm").hasClass('shortForm'))
      $("#tableForm").removeClass('shortForm').
      addClass('longForm');
```

```
    if($("#tableForm").hasClass('longForm'))
      $("#tableForm").removeClass('longForm').a
      ddClass('shortForm');
    });
  });

  </script>
  <style>
  .shortForm {
    height:350px
  }

  .longForm {
    height:500px
  }

  </style>
</head>
<body>
  <html:form action="/EditProfile"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link page="/MyProfile.do">My
            Profile</html:link> &#187;
            <span class="breadcrumb-
            selected">Account Manager</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Account Manager</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form
          shortForm" id="tableForm">
        <tr>
          <td class="note" colspan="2">
            Note: All fields are required. The
            username and password must have a
            minimum of six characters. They can
            be alphanumeric and are case
            sensitive.
          </td>
        </tr>
        <tr>
          <td colspan="2"><span class="error-
          text-big"><html:errors
          property="errorMessage" /></span></td>
        </tr>
        <tr>
          <td width="220px"><bean:message
          key="label.common.firstName" /></td>
        <td>
          <html:text property="firstName"
          size="30" style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.ERR
          OR"/>
          <span class="error-text"><html:errors
          property="firstName" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.lastName" /></td>
        <td>
          <html:text property="lastName"
          size="30" style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.ERR
          OR"/>
          <span class="error-text"><html:errors
          property="lastName" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.email" /></td>
        <td>
          <html:text property="emailAddress"
          size="30" style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.ERR
          OR"/>
          <span class="error-text"><html:errors
          property="emailAddress" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.affiliation" /></td>
        <td>
          <html:text property="affiliation"
          size="30" style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.ERR
          OR"/>
          <span class="error-text"><html:errors
          property="affiliation" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.jobTitle" /></td>
        <td>
          <html:text property="jobTitle"
          size="30" style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.ERR
          OR"/>
          <span class="error-text"><html:errors
          property="jobTitle" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.password" /></td>
        <td class="font-medium-reg">
          <span class="font-gray">Password has
          not changed</span> 
          <html:checkbox
          property="changePassword"
          styleId="changePassword" /> Edit
          <html:hidden property="editPassword"
          styleId="editPassword" />
        </td>
```

```
  </tr>
  <tr>
    <td colspan="2">
    <div id="password">
    <table class="font-medium-reg"
    height="120px">
      <tr>
        <td width="220px"><bean:message
        key="label.common.oldPassword"
        /></td>
        <td>
          <html:password
          property="oldPassword" size="30"
          style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.
          ERROR" />
          <span class="error-
          text"><html:errors
          property="oldPassword" /></span>
        </td>
      </tr>
      <tr>
        <td><bean:message
        key="label.common.newPassword"
        /></td>
        <td>
          <html:password
          property="newPassword" size="30"
          style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.action.
          ERROR" />
          <span class="error-
          text"><html:errors
          property="newPassword" /></span>
      </td>
    </tr>
    <tr>
      <td><bean:message
      key="label.common.reNewPassword"
      /></td>
      <td>
        <html:password
        property="reNewPassword" size="30"
        style="remove-error-field"
        errorStyleClass="error-field"
        errorKey="org.apache.struts.action.E
        RROR" />
        <span class="error-
        text"><html:errors
        property="reNewPassword" /></span>
           </td>
        </tr>
      </table>
      </div>
      </td>
    </tr>
  </table>
    <br/>
        
    <html:submit property="command"
    value="Submit" styleClass="button" />
     
    <html:reset property="command"
    value="Reset" styleClass="button"
    onclick="javacript:resetForm()" />
     
    <html:cancel property="command"
    value="Cancel" styleClass="button" />
  </td>
 </tr>
</table>
</c:if>
</td>
</tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

forgotPassword.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/ForgotPassword"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Forgotten Password</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Forgotten Password</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td>
                Don't remember your password?
                Simply enter your username and
                registered email below and we
                will send you instructions on how
                to reset your password.
              </td>
            </tr>
            <tr>
```

```
            <td class="note">
              Your email server may treat our
              email as spam. Please check your
              spam folder.
            </td>
          </tr>
          <tr>
            <td align="center">
            <span class="error-text-
            big"><html:errors
            property="errorMessage" /></span>
            <table class="font-medium-reg">
              <tr>
                <td width="120px"><bean:message
                key="label.common.username"
                /></td>
                <td>
                  <html:text property="username"
                  size="35" style="remove-error-
                  field" errorStyleClass="error-
                  field"
                  errorKey="org.apache.struts.ac
                  tion.ERROR" />
                  <span class="error-
                  text"><html:errors
                  property="username" /></span>
                </td>
              </tr>
              <tr>
                <td><bean:message
                key="label.common.email"
                /></td>
                <td>
                  <html:text
                  property="emailAddress"
                  size="35" style="remove-error-
                  field" errorStyleClass="error-
                  field"
                  errorKey="org.apache.struts.ac
                  tion.ERROR" />
                  <span class="error-
                  text"><html:errors
                  property="emailAddress"
                  /></span>
                </td>
              </tr>
              <tr>
                <td align="center" colspan="2">
                  <html:submit
                  property="command"
                  value="Reset Password"
                  styleClass="button-long" />
                </td>
              </tr>
            </table>
            </td>
          </tr>
        </table>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

loginAccount.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/Login" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Login Account</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Login Account</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr valign="top">
              <td rowspan="6" width="470px">
                If you log in to the Virholex
                system, all your privileges will
                be<br/>
                loaded to enable access to the
                different modules.<br><br>
                Don't have an account yet?
                <html:link page="/Register.do"
                >Register Now!</html:link>
              </td>
            </tr>
            <tr>
              <td colspan="2">
                <span class="error-text-
                big"><html:errors
                property="errorMessage" /></span>
                <br/>
                <span style="position:relative;
                top:-6px;">Username:</span>
                <br/>
                <html:text property="username"
                size="45" style="remove-error-
```

```
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR" />
            <span class="error-
            text"><html:errors
            property="username" /></span>
          </td>
        </tr>
        <tr>
          <td colspan="2">
            <span style="position:relative;
            top:-6px;">Password: </span>
            <br/>
            <html:password
            property="password" size="45"
            style="remove-error-field"
            errorStyleClass="error-field"
            errorKey="org.apache.struts.actio
            n.ERROR" />
            <span class="error-
            text"><html:errors
            property="password" /></span>
          </td>
        </tr>
        <tr>
          <td height="50px">
            <html:link
            page="/ForgotPassword.do" >Forgot
            your password?</html:link>
          </td>
          <td align="right">
            <html:submit property="command"
            value="Submit"
            styleClass="button" />
          </td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

projectRequestEvaluate.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/ProjectRequest"
 method="POST">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <c:if test="${(userLogin ne null) and
   ((statusLogin eq 'activated') or
   (statusLogin eq 'admin') or (statusLogin
   eq 'superadmin'))}">
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link page="/MyProjects.do">My
      Project</html:link> &#187;
      <span class="breadcrumb-
      selected">Privilege Request</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2">Privilege Request</td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr>
       <td class="font-bold"
       width="200px">Project Name </td>
       <td>
        <bean:write
        name="RequestPrivilegeForm"
        property="project" />
       </td>
      </tr>
      <tr>
       <td class="font-bold">Requested
       by</td>
       <td>
        <bean:write
        name="RequestPrivilegeForm"
        property="firstName" />
        <bean:write
        name="RequestPrivilegeForm"
        property="lastName" />
        <input type="hidden"
        name="userEval"
        value="${RequestPrivilegeForm.use
        rname}" />
       </td>
      </tr>
      <tr>
       <td class="font-bold">Date
       Requested </td>
       <td>
        <bean:write
        name="RequestPrivilegeForm"
        property="dateAdded" />
       </td>
      </tr>
```

```
          <tr>
            <td class="font-bold"
            valign="top">Message</td>
            <td><bean:write
            name="RequestPrivilegeForm"
            property="requestMessage" /></td>
          </tr>
          <tr>
            <td class="font-bold">Select
            Privilege to grant</td>
            <td>
              <html:select property="privilege"
              style="remove-error-field-file"
              errorStyleClass="error-field"
              errorKey="org.apache.struts.actio
              n.ERROR" >
              <html:option value="Default"
              styleClass="font-gray">Select
              Privilege</html:option>
              <c:forEach var="priv"
              items="${RequestPrivilegeForm.rol
              es}">
                <html:option
                value="${priv}">${priv}</html:o
                ption>
              </c:forEach>
              </html:select>
              <span class="error-
              text"><html:errors
              property="privilege" /></span>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>
              
          <html:submit property="command"
          value="Approve" styleClass="button-
          long" />
           
          <html:submit property="command"
          value="Deny" styleClass="button-
          long" />
           
          <html:submit property="command"
          value="No Change"
          styleClass="button-long" />
           
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

reactivateAccount.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/ReactivateAccount"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title">Account
          Reactivation</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Your account has been
            <i>disabled</i>.<br/><br/>
            Do you want to request for account
            reactivation?
            <br/><br/>
            <html:submit property="command"
            value="Yes" styleClass="button" />
            <html:cancel property="command"
            value="No" styleClass="button" />
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

registerAccount.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
```

```
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/Register" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Registration</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Registration</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td colspan="2">
                Register to Virus-Host
                Interaction Lexicon to take
                advantage of the privileges the
                user can get. Click <html:link
                page="/VirholexPrivileges.do"
                >here</html:link> to find out
                what these are.
              </td>
            </tr>
            <tr>
              <td class="note" colspan="2">
                Note: All fields are required.
                The username and password must
                have a minimum of six characters.
                They can be alphanumeric and are
                case sensitive.
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td width="200px"><bean:message
              key="label.common.username" /></td>
              <td>
                <html:text property="username"
                size="30" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="username" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.password" /></td>
              <td>
                <html:password
                property="password" size="30"
                style="remove-error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="password" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.repassword"
              /></td>
              <td>
                <html:password
                property="repassword" size="30"
                style="remove-error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR"/>
                <span class="error-
                text"><html:errors
                property="repassword" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.email" /></td>
              <td>
                <html:text
                property="emailAddress" size="30"
                style="remove-error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR"/>
                <span class="error-
                text"><html:errors
                property="emailAddress" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.firstName"
              /></td>
              <td>
                <html:text property="firstName"
                size="30" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR"/>
                <span class="error-
                text"><html:errors
                property="firstName" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.lastName" /></td>
              <td>
```

```
            <html:text property="lastName"
            size="30" style="remove-error-
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR"/>
            <span class="error-
            text"><html:errors
            property="lastName" /></span>
          </td>
        </tr>
        <tr>
          <td><bean:message
          key="label.common.affiliation"
          /></td>
          <td>
            <html:text property="affiliation"
            size="30" style="remove-error-
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR"/>
            <span class="error-
            text"><html:errors
            property="affiliation" /></span>
          </td>
        </tr>
        <tr>
          <td><bean:message
          key="label.common.jobTitle" /></td>
          <td>
            <html:text property="jobTitle"
            size="30" style="remove-error-
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR"/>
            <span class="error-
            text"><html:errors
            property="jobTitle" /></span>
          </td>
        </tr>
      </table>
      <br/>
          
      <html:submit property="command"
      value="Submit" styleClass="button" />
      <html:reset property="command"
      value="Reset" styleClass="button" />
      <html:cancel property="command"
      value="Cancel" styleClass="button" />
      </td>
    </tr>
    <tr>
      <td height="50px"> </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

requestPrivilege.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/PrivilegeRequest"
 method="POST">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <c:if test="${(userLogin ne null) and
   ((statusLogin eq 'activated') or
   (statusLogin eq 'admin') or (statusLogin
   eq 'superadmin'))}">
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <span class="breadcrumb-
      selected">Privilege Request</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2">Privilege Request</td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr>
       <td class="font-bold"
       width="130px">Project Name : </td>
       <td>
        <bean:write
        name="RequestPrivilegeForm"
        property="project" />
       </td>
      </tr>
      <tr>
       <td class="font-bold"
       colspan="2">How can you contribute
       to this project?</td>
      </tr>
      <tr>
       <td colspan="2">
        <html:textarea
        property="requestMessage"
        cols="90"
        rows="10"></html:textarea>
```

```
          </td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <html:submit property="command"
            value="Submit Request"
            styleClass="button-long" />
             
            <html:cancel property="command"
            value="Cancel"
            styleClass="button-long" />
          </td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

requestPrivilegeFinal.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <span class="breadcrumb-
      selected">Privilege Request</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title">Privilege
     Request</td>
    </tr>
    <tr>
     <td> </td>
    </tr>
   <c:choose>
    <c:when test="${module eq
    'VirhoExperiments'}" >
     <c:set var="url"
     value="/ViewProjectExperimentDetails.d
     o?project=${project}"/>
    </c:when>
    <c:when test="${module eq
    'VirhoImages'}" >
     <c:set var="url"
     value="/ViewImageSetDetails.do?project
     Set=${project}"/>
    </c:when>
    <c:when test="${module eq
    'VirhoModels'}" >
     <c:set var="url"
     value="/ViewProjectDetails.do?project=
     ${project}"/>
    </c:when>
    <c:when test="${module eq
    'VirhoReferences'}" >
     <c:set var="url"
     value="/ViewReferences.do?collection=$
     {project}"/>
    </c:when>
    <c:when test="${module eq
    'VirhoHotspots'}" >
     <c:set var="url"
     value="/ViewVirusDetails.do?virus=${pr
     oject}"/>
    </c:when>
    <c:otherwise>
     <c:set var="url" value="/Home.do"/>
    </c:otherwise>
   </c:choose>
   <c:choose>
    <c:when test="${hasRequest eq 'true'}">
    <tr>
     <td class="content-desc">
      You have already requested for a
      privilege in
      <html:link page="${url}"><c:out
      value="${project}" /></html:link>
      project. The Virholex Team is still
      processing your request.
     </td>
    </tr>
    </c:when>
    <c:otherwise>
    <tr>
     <td class="content-desc">
      Your request for a privilege in
      <html:link page="${url}"><c:out
      value="${project}" /></html:link>
      project will be evaluated by the
      Virholex Team.
     </td>
    </tr>
    </c:otherwise>
   </c:choose>
   </table>
   </td>
  </tr>
  <tr class="footer">
   <td colspan="2">
```

```
      &copy; <bean:message
      key="virho.copyright" />
     </td>
   </tr>
  </table>
</body>
</html>
```

resetPassword.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/ResetPassword"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Reset Password</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Reset Password</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
          <c:choose>
            <c:when test="${(requestScope.state
            ne null) or (requestScope.crack ne
            null) or (requestScope.key ne null)
            or (requestScope.randomSequence ne
            null)}">
            <tr>
              <td colspan="2">
                <input type="hidden" name="state"
                value="${requestScope.state}" />
                <input type="hidden" name="crack"
                value="${requestScope.crack}" />
                <input type="hidden" name="key"
                value="${requestScope.key}" />
                <input type="hidden" name="rand"
                value="${requestScope.randomSeque
                nce}" />
                In order to complete this
                process, please supply your
                <b>username</b> and <b>new
                password</b>.
              </td>
            </tr>
            <tr>
              <td class="note" colspan="2">
                The new password must have a
                minimum of six characters. It can
                be alphanumeric and it is case
                sensitive.
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td width="200px"><bean:message
              key="label.common.username" /></td>
              <td>
                <html:text property="username"
                size="35" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="username" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.newPassword"
              /></td>
              <td>
                <html:password
                property="newPassword" size="35"
                style="remove-error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="newPassword" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.renewPassword"
              /></td>
              <td>
                <html:password
                property="renewPassword"
                size="35" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="renewPassword"
                /></span>
              </td>
            </tr>
```

```
          </c:when>
        </c:choose>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <c:if test="${(requestScope.state ne
      null) or (requestScope.crack ne null)
      or (requestScope.key ne null) or
      (requestScope.randomSequence ne
      null)}">
      <tr>
        <td colspan="2">
              
          <html:submit property="command"
          value="Change Password"
          styleClass="button-superlong" />
        </td>
      </tr>
    </c:if>
    </table>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

statusAccountMessage.jsp

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld"
prefix="logic" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td> </td>
        </tr>
      <c:choose>
        <c:when test="${status eq 'approved'}">
        <tr>
          <td class="content-title">Account
          Activated</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Congratulations! Your request for
            the account <span class="font-
            red"><c:out value="${username}"
            /></span> has been approved.
            <br/><br/>
            You are now a registered user of
            VirHoLex. Feel free to browse the
            website.
          </td>
        </tr>
        </c:when>
        <c:when test="${status eq 'pending'}">
        <tr>
          <td class="content-title">Account
          Validation</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Your request for the account <span
            class="font-red"><c:out
            value="${username}" /></span> is
            still being processed. Please check
            again later.
          </td>
        </tr>
        </c:when>
        <c:when test="${status eq 'denied'}">
        <tr>
          <td class="content-title">Account
          Rejection</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            We regret to inform you that your
            account request <span class="font-
            red"><c:out value="${username}"
            /></span> is rejected by the
            VirhoLex team.
            <br/><br/>
            You can contact the administrator to
            reconsider your account.
          </td>
        </tr>
        </c:when>
        <c:when test="${status eq
        'reactivate'}">
        <tr>
          <td class="content-title">Account
          Reactivation</td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-desc">
            Your reactivation request is still
            being processed.
          </td>
```

```
</tr>
</c:when>
<c:when test="${status eq
'suspended'}">
<tr>
  <td class="content-title">Account
  Reactivation</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    Your request for a reactivation will
    be evaluated within 24 hours.
  </td>
</tr>
</c:when>
<c:when test="${status eq
'registered'}">
<tr>
  <td class="content-title">Account
  Registration</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    Your request for Virholex account
    <span class="font-red"><c:out
    value="${username}" /></span> has
    been submitted.<br/><br/>
    We will evaluate your request
    according to VirHoLex' policy within
    24 hours.
  </td>
</tr>
</c:when>
<c:when test="${status eq
'unsubscribe'}">
<tr>
  <td class="content-title">Account
  Unsubscription</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    You have successfully canceled your
    Account.<br/><br/>
    If you want to reactivate your
    account, just log in again and
    follow the instructions.
  </td>
</tr>
</c:when>
<c:when test="${status eq 'reset
password'}">
<tr>
  <td class="content-title">Reset
  Password</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    You have successfully changed the
    password for your Virholex account
    <span class="font-red"><c:out
    value="${userLogin}"
    /></span>.<br/><br/>
    Click <html:link
    page="/Login.do">here</html:link> to
    log in.
  </td>
</tr>
</c:when>
<c:when test="${status eq 'invalid
key'}">
<tr>
  <td class="content-title">Invalid
  Activation Key</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    We are sorry to inform you that the
    link containing the activation key
    is invalid. Either the link was used
    already to reset the password, or
    the key is incorrect.
    <p>Click <html:link
    page="/ForgotPassword.do">here</html
    :link> to reset your password.</p>
  </td>
</tr>
</c:when>
<c:otherwise>
<tr>
  <td class="content-title">Forgotten
  Password</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-desc">
    Instructions will be sent within 24
    hours.
  </td>
</tr>
</c:otherwise>
</c:choose>
</table>
</td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</body>
</html>
```

userProfile.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
```

```
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/MyProfile"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-selected">My
            Profile</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">My Profile</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:220px">
            <tr>
              <td width="200px"><bean:message
              key="label.common.firstName"
              /></td>
              <td><bean:write
              name="UserProfileForm"
              property="firstName" /></td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.lastName" /></td>
              <td><bean:write
              name="UserProfileForm"
              property="lastName" /></td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.email" /></td>
              <td><bean:write
              name="UserProfileForm"
              property="emailAddress" /></td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.affiliation"
              /></td>
              <td><bean:write
              name="UserProfileForm"
              property="affiliation" /></td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.jobTitle" /></td>
              <td><bean:write
              name="UserProfileForm"
              property="jobTitle" /></td>
            </tr>
          </table>
          <br/>
              
          <html:submit property="command"
          value="Account Manager"
          styleClass="button-long" />
           
          <c:choose>
            <c:when
            test="${(UserProfileForm.unsubscribe
            eq 'disabled') and (statusLogin ne
            'superadmin')}">
              <html:submit property="command"
              value="Unsubscribe"
              styleClass="button-long"
              disabled="true" />
              <br/><br/><br/>
              <div class="note">
              Disabled Unsubscribe button:
              <ul>
                <li>You are the sole
                administrator of Virholex
                system</li>
                <li>You have one or more projects
                that do not have successor</li>
              </ul>
              </div>
            </c:when>
            <c:otherwise>
              <html:submit property="command"
              value="Unsubscribe"
              styleClass="button-long" />
            </c:otherwise>
          </c:choose>
          </td>
        </tr>
        <tr>
          <td height="50px"> </td>
        </tr>
      </table>
      </c:if>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

userProjects.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
```

```
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-selected">My
            Projects</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">My Projects</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td>
              <table class="table-content">
                <tr>
                  <th>Involvement</th>
                  <th>Privilege</th>
                  <th>Module</th>
                  <th>Requests</th>
                </tr>
              <c:choose>
                <c:when
                test="${fn:length(projects) gt
                0}">
                <c:forEach var="detail"
                items="${projects}">
                <tr>
                  <td>
                <c:choose>
                  <c:when test="${detail.module
                  eq 'VirhoExperiments'}">
                    <c:set var="url"
                    value="/ViewProjectExperimentD
                    etails.do?module=VirhoExperime
                    nts&project=" />
                  </c:when>
                  <c:when test="${detail.module
                  eq 'VirhoImages'}">
                    <c:set var="url"
                    value="/ViewImageSetDetails.do
                    ?module=VirhoImages&projectSet
                    =" />
                  </c:when>
                  <c:when test="${detail.module
                  eq 'VirhoModels'}">
                    <c:set var="url"
                    value="/ViewProjectDetails.do?
                    module=VirhoModels&project="
                    />
                  </c:when>
                  <c:when test="${detail.module
                  eq 'VirhoReferences'}">
                    <c:set var="url"
                    value="/ViewReferences.do?modu
                    le=VirhoReferences&collection=
                    " />
                  </c:when>
                  <c:when test="${detail.module
                  eq 'VirhoHotspots'}">
                    <c:set var="url"
                    value="/ViewVirusDetails.do?mo
                    dule=VirhoHotspots&virus=" />
                  </c:when>
                </c:choose>
                <c:choose>
                  <c:when
                  test="${(detail.privilege eq 4)
                  and (detail.requestCount gt
                  0)}">
                    <html:link
                    page="${url}${detail.projectNa
                    me}">
                    <span class="font-blue font-
                    bold"><c:out
                    value="${detail.projectName}"
                    /></span>
                    </html:link>
                  </c:when>
                  <c:otherwise>
                    <html:link
                    page="${url}${detail.projectNa
                    me}">
                    <span class="font-blue"><c:out
                    value="${detail.projectName}"
                    /></span>
                    </html:link>
                  </c:otherwise>
                </c:choose>
                  </td>
                  <td><c:out
                  value="${detail.role}" /></td>
                  <td><c:out
                  value="${detail.module}"
                  /></td>
                  <td align="center">
                  <c:if test="${(detail.privilege
                  eq 4) and (detail.requestCount
                  gt 0)}">
                    <html:link
                    page="/ProjectRequest.do?modul
                    e=${detail.module}&project=${d
                    etail.projectName}">
                    <span class="font-blue font-
                    bold"><c:out
                    value="${detail.requestCount}"
                    /></span>
                    </html:link>
                  </c:if>
                  </td>
                </tr>
                </c:forEach>
                </c:when>
                <c:otherwise>
                <tr>
                  <td class="font-italic"
                  colspan="4">
```

```
              You are currently not part of
              any Virholex projects.
            </td>
          </tr>
          </c:otherwise>
        </c:choose>
        </table>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
</table>
</c:if>
</td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</body>
</html>
```

userProjects.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/ProjectRequest"
  method="POST" >
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link page="/MyProjects.do">My
            Project</html:link> &#187;
            <span class="breadcrumb-
            selected">Privilege Request</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Privilege Request</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td><span class="font-bold">Project
              Name :</span> <c:out
              value="${RequestPrivilegeForm.proje
              ct}" /></td>
            </tr>
            <tr>
              <td>
              <table class="table-content">
                <tr>
                  <th>Name</th>
                  <th>Username</th>
                  <th>Affiliation</th>
                  <th>Action</th>
                </tr>
              <c:choose>
                <c:when test="${not empty
                RequestPrivilegeForm.users}">
                <c:forEach var="user"
                items="${RequestPrivilegeForm.use
                rs}">
                <tr>
                  <td>${user.firstName} ${us
                  er.lastName}</td>
                  <td>${user.username}</td>
                  <td>${user.affiliation}</td>
                  <td align="center">
                    <html:link
                    page="/ProjectRequest.do?userE
                    val=${user.username}">
                    <span class="font-
                    blue">Evaluate</span>
                    </html:link>
                  </td>
                </tr>
                </c:forEach>
                </c:when>
                <c:otherwise>
                <tr>
                  <td colspan="4" class="font-
                  italic">There are no users to
                  be evaluated</td>
                </tr>
                </c:otherwise>
              </c:choose>
              </table>
              </td>
            </tr>
          </table>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td colspan="5">
                
```

```
        <html:cancel property="command"
        value="Cancel" styleClass="button-
        long-blue" />
       </td>
      </tr>
    </table>
    </c:if>
    </td>
   </tr>
   <tr class="footer">
    <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
    </td>
   </tr>
  </table>
  </html:form>
</body>
</html>
```

viewUserProfileDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  function changeAction(username) {
    document.forms[0].action =
    "UserProfileDetails.do?user=" + username;
  }
  </script>
</head>
<body>
  <html:form action="/UserProfileDetails"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
  <tr>
  <td> </td>
  </tr>
  <tr valign="top">
  <td class="sidebar">
  <jsp:include page="/rd_virui/sidebar.jsp" />
  </td>
  <td>
  <c:if test="${(userLogin ne null) and
  ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or (statusLogin eq
  'superadmin'))}">
  <table class="content">
  <tr>
  <td class="breadcrumb">
  <html:link page="/Home.do"><bean:message
  key="button.virho.home" /></html:link>
  &#187;
  <html:link page="/VirholexUsers.do">Virholex
  Users</html:link> &#187;
  <span class="breadcrumb-selected">Profile
  Details</span>
  </td>
  </tr>
  <tr>
  <td> </td>
  </tr>
  <tr>
  <td class="content-title-form"
  colspan="2">Profile Details</td>
  </tr>
  <tr>
  <td>
  <table class="content-desc-form"
  style="height:200px">
  <tr>
  <td width="200px"><bean:message
  key="label.common.name" /></td>
  <td><bean:write name="UserProfileForm"
  property="firstName" /> 
  <bean:write name="UserProfileForm"
  property="lastName" /></td>
  </tr>
  <tr>
  <td width="200px"><bean:message
  key="label.common.username" /></td>
  <td><bean:write name="UserProfileForm"
  property="username" /></td>
  </tr>
  <tr>
  <td><bean:message key="label.common.email"
  /></td>
  <td><bean:write name="UserProfileForm"
  property="emailAddress" /></td>
  </tr>
  <tr>
  <td><bean:message
  key="label.common.affiliation" /></td>
  <td><bean:write name="UserProfileForm"
  property="affiliation" /></td>
  </tr>
  <tr>
  <td><bean:message
  key="label.common.jobTitle" /></td>
  <td><bean:write name="UserProfileForm"
  property="jobTitle" /></td>
  </tr>
  </table>
  <br/>
      
  <c:if test="${(statusLogin eq 'admin') or
  (statusLogin eq 'superadmin')}">
  <c:choose>
  <c:when test="${(UserProfileForm.status eq
  'admin') or (UserProfileForm.status eq
  'superadmin')}">
  <html:submit property="command"
  value="Remove from Admin"
  styleClass="button-superlong"
  onclick="changeAction('${UserProfileForm.use
  rname}')" />
  </c:when>
  <c:otherwise>
  <html:submit property="command"
  value="Promote to Admin" styleClass="button-
  superlong"
  onclick="changeAction('${UserProfileForm.use
  rname}')" />
  </c:otherwise>
  </c:choose>
  </c:if>
   
  </td>
  </tr>
  </table>
  </c:if>
```

```
  </td>
  </tr>
  <tr class="footer">
  <td colspan="2">
  &copy; <bean:message key="virho.copyright"
  />
  </td>
  </tr>
  </table>
  </html:form>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Virholex Users</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Virholex Users</td>
        </tr>
        <tr>
<td>
<table class="content-desc-form">
  <tr>
    <td>
    <table class="letterList">
      <tr>
        <td>
        <c:forEach var="letter"
        items="A,B,C,D,E,F,G,H,I,J,K,L,
        M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
        varStatus="loop">
      <c:choose>
        <c:when
        test="${ViewUsersForm.letters[l
        oop.index] eq letter}">
          <html:link
          page="/VirholexUsers.do?list=$
          {letter}"><span class="font-
          letter">${letter}</span></html
          :link> 
        </c:when>
        <c:otherwise>
          ${letter} 
        </c:otherwise>
      </c:choose>
        </c:forEach>
        <html:link
        page="/VirholexUsers.do?list=">
        View All</html:link>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td>
    <table class="table-content">
      <tr>
        <th>Name</th>
        <th>Username</th>
        <th>Email Address</th>
        <th>Member since</th>
      </tr>
    <c:choose>
      <c:when
      test="${fn:length(ViewUsersForm.u
      sers) gt 0}">
      <c:forEach var="user"
      items="${ViewUsersForm.users}">
      <tr>
        <td>
          <html:link
          page="/UserProfileDetails.do?u
          ser=${user.username}" >
          <span class="font-
          blue">${user.firstName} $
          {user.lastName}</span></html:l
          ink>
        </td>
        <td>${user.username}</td>
        <td>${user.emailAddress}</td>
        <td
        align="center">${user.approvalD
        ate}</td>
      </tr>
      </c:forEach>
      </c:when>
      <c:otherwise>
      <tr>
        <td class="font-medium-reg
        font-italic" colspan="4">
          Your query for <b><c:out
          value="${ViewUsersForm.searchT
          erm}" /></b> in <i><c:out
          value="${field}" /></i> did
          not return any results.
```

```
            </td>
          </tr>
          </c:otherwise>
        </c:choose>
        </table>
        </td>
      </tr>
      <tr>
        <td class="note" align="center">
          <br/>
          <html:form
          action="/VirholexUsers"
          method="GET">
          Search <html:text
          property="searchTerm"
          styleClass="note" /> in
          <html:select
          property="searchOption"
          styleClass="note">
            <html:option
            value="username">username</html
            :option>
            <html:option
            value="name">name</html:option>
            <html:option
            value="email">email</html:optio
            n>
          </html:select>
           
          <html:submit property="search"
          value="Search Members"
          styleClass="button-long-white" />
          </html:form>
        </td>
      </tr>
    </table>
    </td>
  </tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</body>
</html>
```

WhyShouldIRegister.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
<table class="body" align="center">
<jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <span class="breadcrumb-
          selected">Why Should I
          Register?</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title">Why Should
        You Register?</td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-desc">
          As a registered VirhoLex user, you
          will be entitled to the following
          privileges:
          <ul>
            <li>Get involved with the latest
            virus-host Experiments</li>
            <li>Search and view Experiment
            models, images and references</li>
            <li>Download available data</li>
            <li>Contribute to the VirhoLex
            Community</li>
          </ul>
          <html:link page="/Register.do"
          >Click here to register</html:link>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</body>
</html>
```

VIRHO EXPERIMENTS

addExperiment.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
```

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <link rel="stylesheet" type="text/css"
  href="./files/datechooser.css"
  media="screen"/>
  <script type="text/javascript"
  src="./files/date-functions.js"></script>
  <script type="text/javascript"
  src="./files/datechooser.js"></script>
  <script type="text/javascript"
  src="./files/javascript.js"></script>
</head>
<body>
  <html:form action="/AddExperiment"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewProjects.do">Virho
            Experiments Projects</html:link>
            &#187;
            <html:link
            page="/ViewProjectExperimentDetails.
            do?project=${project}">${project}</h
            tml:link> &#187;
            <span class="breadcrumb-
            selected">Add Experiment</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Add Experiment</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:300px">
            <tr>
              <td class="note" colspan="2">
                Note: All fields that are marked
                with asterisk (*) are required.
                <p>Please refrain from refreshing
                the page, or navigating to
                another page without canceling
                this page</p>
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td width="190px" class="font-
              bold">Experiment *</td>
              <td>
                <html:text
                property="experimentName"
                size="65" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="experimentName"
                /></span>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Author *</td>
              <td>
                <html:text property="author"
                size="40" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="author" /></span>
              </td>
            </tr>
            <tr>
              <td class="font-bold"
              valign="top"><bean:message
              key="label.common.description"
              /></td>
              <td>
                <html:textarea
                property="description" rows="4"
                cols="68" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="description" /></span>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Experiment
              Date *</td>
              <td>
                <html:text
                property="experimentDate"
                size="20" styleId="sd"
                readonly="true" style="remove-
                error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR"
                onfocus="showChooser(this, 'sd',
                'chooserSpan1', 1950, 2020,
                Date.patterns.ShortDatePattern,
                false);" />
                <img src="./images/calendar.gif"
                onclick="showChooser(this, 'sd',
                'chooserSpan1', 1950, 2020,
```

```
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan1"
    class="dateChooser"
    style="display: none; visibility:
    hidden; width: 200px;"></div>
    <span class="error-
    text"><html:errors
    property="experimentDate"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Virus
  Name/Type *</td>
  <td>
    <html:text property="virus"
    size="65" style="remove-error-
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
    <span class="error-
    text"><html:errors
    property="virus" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Measurement
  Intervals</td>
  <td>
    <html:text
    property="measurementInterval"
    size="5" style="remove-error-
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.actio
    n.ERROR" /> minutes
    <span class="error-
    text"><html:errors
    property="measurementInterval"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold" valign="top"
  rowspan="2">Time Points</td>
  <td>
    Date:
    <html:text property="timePoints"
    size="20" styleId="ed"
    readonly="true" style="remove-
    error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR"
    onfocus="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" />
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'ed',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan2"
    class="dateChooser"
    style="display: none; visibility:
    hidden; width: 200px;"></div>
  </td>
</tr>
<tr>
  <td>
    Hours: <html:text
    property="tpHours" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
    Minutes: <html:text
    property="tpMinutes" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
    Seconds: <html:text
    property="tpSeconds" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" /><br/>
    <span class="error-
    text"><html:errors
    property="timePoints" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Biological
  Replicates</td>
  <td>
    <html:text
    property="biologicalReplicates"
    size="5" />
    <span class="error-
    text"><html:errors
    property="biologicalReplicates"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Technical
  Replicates</td>
  <td>
    <html:text
    property="technicalReplicates"
    size="5" />
    <span class="error-
    text"><html:errors
    property="technicalReplicates"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Experiment
  Photos</td>
  <td>
    <html:select
    property="imageSetId">
      <html:option value="0"
      styleClass="font-gray">Select
      Image Set</html:option>
    <c:forEach var="image"
    items="${requestScope.imagesList}
    ">
      <html:option
      value="${image.projectSetId}"><
      c:out
      value="${image.projectSetName}"
      /></html:option>
    </c:forEach>
    </html:select>
  </td>
</tr>
<tr>
  <td colspan="2"> </td>
</tr>
<tr>
  <td colspan="2">
  <table class="table-content"
  id="expComponentId">
    <tr>
```

```
      <th colspan="3">Experiment
      Components</th>
    </tr>
    <tr style="background-
    color:#E1E8F6;">
      <td>
      <table>
        <tr>
          <td class="font-
          bold">Condition</td>
          <td><input type="text"
          name="condition" size="90"
          /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Method</td>
          <td><input type="text"
          name="methodName" size="90"
          /></td>
        </tr>
        <tr>
          <td class="font-bold"
          valign="top">Description</td
          >
          <td><input type="text"
          name="methodDesc" size="90"
          /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Variable Type</td>
          <td><input type="text"
          name="varType" size="90"
          /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Value</td>
          <td><input type="text"
          name="varValue" size="90"
          /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Chemical
          Activator</td>
          <td><input type="text"
          name="chemActivator"
          size="90" /></td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr align="right">
  <td colspan="2">
    <a href="#"
    onclick="addField('expComponent')
    ">Add Experiment
    Component</a> | 
    <a href="#"
    onclick="deleteField('expComponen
    t')">Delete Experiment
    Component</a>   &n
    bsp;
  </td>
 </tr>
</table>
<br/>
    
<html:submit property="command"
value="Submit" styleClass="button" />
      <html:reset property="command"
      value="Reset" styleClass="button" />
      <html:cancel property="command"
      value="Cancel" styleClass="button" />
      </td>
    </tr>
    <tr>
      <td height="50px"> </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

addExperimentComponent.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script
  src="./files/javascript.js"></script>
  <script src="./files/jquery-
  1.8.0.js"></script>
</head>
<body>
  <html:form action="/AddExperimentComponent"
  method="POST" styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
```

```
    key="button.virho.home"
    /></html:link> &#187;
    <html:link
    page="/ViewProjects.do">Virho
    Experiments Projects</html:link>
    &#187;
    <html:link
    page="/ViewProjectExperimentDetails.
    do?project=${project}">${project}</h
    tml:link> &#187;
    <html:link
    page="/ViewExperimentDetails.do?expe
    riment=${experiment}">${experiment}<
    /html:link> &#187;
    <span class="breadcrumb-
    selected">Add Experiment
    Component</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Add Experiment
  Component</td>
</tr>
<tr>
  <td>
  <html:hidden property="experimentId"
  value="${ExperimentComponentsForm.expe
  rimentId}" />
  <html:hidden property="projectId"
  value="${ExperimentComponentsForm.proj
  ectId}" />
  <table class="content-desc-form">
    <tr>
      <td class="note">
        Note: Please refrain from
        refreshing the page, or
        navigating to another page
        without canceling this page
      </td>
    </tr>
    <tr>
      <td colspan="2">
      <table class="table-content"
      id="expComponentId">
        <tr>
          <th colspan="3">Experiment
          Component</th>
        </tr>
        <tr style="background-
        color:#E1E8F6;">
          <td>
          <table>
            <tr>
              <td class="font-
              bold">Condition</td>
              <td><input type="text"
              name="condition" size="90"
              /></td>
            </tr>
            <tr>
              <td class="font-
              bold">Method</td>
              <td><input type="text"
              name="methodName" size="90"
              /></td>
            </tr>
            <tr>
              <td class="font-bold"
              valign="top">Description</td
              >
                <td><input type="text"
                name="methodDesc" size="90"
                /></td>
              </tr>
              <tr>
                <td class="font-
                bold">Variable Type</td>
                <td><input type="text"
                name="varType" size="90"
                /></td>
              </tr>
              <tr>
                <td class="font-
                bold">Value</td>
                <td><input type="text"
                name="varValue" size="90"
                /></td>
              </tr>
              <tr>
                <td class="font-
                bold">Chemical
                Activator</td>
                <td><input type="text"
                name="chemActivator"
                size="90" /></td>
              </tr>
            </table>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr align="right">
      <td>
        <a href="#"
        onclick="addField('expComponent')">
        Add Experiment
        Component</a> | 
        <a href="#"
        onclick="deleteField('expComponent'
        )">Delete Experiment
        Component</a>   &nbs
        p;
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button" />
    <html:reset property="command"
    value="Reset" styleClass="button" />
    <html:cancel property="command"
    value="Cancel" styleClass="button" />
  </td>
</tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
```

```
  </html:form>
</body>
</html>
```

addProject.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/AddProject"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewProjects.do">Virho
            Experiments Projects</html:link>
            &#187;
            <span class="breadcrumb-
            selected">Add Project</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Add Project</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:300px">
            <tr>
              <td class="note" colspan="2">
                Note: All fields that are marked
                with asterisk (*) are required
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td width="150px"><bean:message
              key="label.common.project" />
              *</td>
              <td>
                <html:text property="projectName"
                size="65" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="projectName" /></span>
              </td>
            </tr>
            <tr>
              <td valign="top"><bean:message
              key="label.common.description"
              /></td>
              <td>
                <html:textarea
                property="description" rows="7"
                cols="75" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="description" /></span>
              </td>
            </tr>
            <tr>
              <td><bean:message
              key="label.common.status" /> *</td>
              <td>
                <html:select property="status"
                style="remove-error-field"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR">
                  <html:option value="Default"
                  styleClass="font-gray">Select
                  Status</html:option>
                  <html:option
                  value="Planned">Planned</html:o
                  ption>
                  <html:option value="On
                  Going">On Going</html:option>
                  <html:option
                  value="Completed">Completed</ht
                  ml:option>
                </html:select>
                <span class="error-
                text"><html:errors
                property="status" /></span>
              </td>
            </tr>
          </table>
          <br/>
              
          <html:submit property="command"
          value="Submit" styleClass="button" />
          <html:reset property="command"
          value="Reset" styleClass="button" />
          <html:cancel property="command"
          value="Cancel" styleClass="button" />
          </td>
        </tr>
        <tr>
          <td height="50px"> </td>
        </tr>
      </table>
      </td>
    </tr>
```

```
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

editExperiment.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <link rel="stylesheet" type="text/css"
  href="./files/datechooser.css"
  media="screen"/>
  <script type="text/javascript"
  src="./files/date-functions.js"></script>
  <script type="text/javascript"
  src="./files/datechooser.js"></script>
</head>
<body>
  <html:form action="/EditExperiment"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewProjects.do">Virho
            Experiments Projects</html:link>
            &#187;
            <html:link
            page="/ViewProjectExperimentDetails.
            do?project=${project}">${project}</h
            tml:link> &#187;
            <html:link
            page="/ViewExperimentDetails.do?expe
            riment=${experiment}">${experiment}<
            /html:link> &#187;
            <span class="breadcrumb-
            selected">Edit Experiment</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Edit Experiment</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:300px">
            <tr>
              <td class="note" colspan="2">
                Note: All fields that are marked
                with asterisk (*) are required
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td width="190px" class="font-
              bold">Experiment *</td>
              <td>
                <html:text
                property="experimentName"
                size="65" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="experimentName"
                /></span>
                <html:hidden
                property="experimentId"
                value="${ExperimentDetailsForm.ex
                perimentId}" />
                <html:hidden property="projectId"
                value="${ExperimentDetailsForm.pr
                ojectId}" />
              </td>
            </tr>
            <tr>
              <td class="font-bold">Author *</td>
              <td>
                <html:text property="author"
                size="40" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="author" /></span>
            </td>
          </tr>
          <tr>
            <td class="font-bold"
            valign="top"><bean:message
            key="label.common.description"
            /></td>
            <td>
              <html:textarea
              property="description" rows="4"
              cols="68" style="remove-error-
```

```
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.action.
    ERROR" />
    <span class="error-
    text"><html:errors
    property="description" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Experiment
  Date *</td>
  <td>
    <html:text
    property="experimentDate" size="20"
    styleId="sd" readonly="true"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" onfocus="showChooser(this,
    'sd', 'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" />
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan1"
    class="dateChooser" style="display:
    none; visibility: hidden; width:
    200px;"></div>
    <span class="error-
    text"><html:errors
    property="experimentDate" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Virus
  Name/Type *</td>
  <td>
    <html:text property="virus"
    size="65" style="remove-error-
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.action.
    ERROR" />
    <span class="error-
    text"><html:errors property="virus"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Measurement
  Intervals</td>
  <td>
    <html:text
    property="measurementInterval"
    size="5" style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" /> minutes
    <span class="error-
    text"><html:errors
    property="measurementInterval"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold" valign="top"
  rowspan="2">Time Points</td>
  <td>
    Date:
    <html:text property="timePoints"
    size="20" styleId="ed"
    readonly="true" style="remove-
    error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" onfocus="showChooser(this,
    'sd', 'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" />
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'ed',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan2"
    class="dateChooser" style="display:
    none; visibility: hidden; width:
    200px;"></div>
  </td>
</tr>
<tr>
  <td>
    Hours: <html:text
    property="tpHours" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" />
    Minutes: <html:text
    property="tpMinutes" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" />
    Seconds: <html:text
    property="tpSeconds" size="5"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" /><br/>
    <span class="error-
    text"><html:errors
    property="timePoints" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Biological
  Replicates</td>
  <td>
    <html:text
    property="biologicalReplicates"
    size="5" />
    <span class="error-
    text"><html:errors
    property="biologicalReplicates"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Technical
  Replicates</td>
  <td>
    <html:text
    property="technicalReplicates"
    size="5" />
    <span class="error-
    text"><html:errors
    property="technicalReplicates"
    /></span>
  </td>
</tr>
<tr>
  <td class="font-bold" rowspan="2"
  valign="top">Experiment Image
  Sets</td>
  <td>
    <html:select property="imageSetId">
```

```
      <html:option value="0"
      styleClass="font-gray">Select
      Image Set</html:option>
    <c:forEach var="image"
    items="${requestScope.imagesList}">
      <html:option
      value="${image.projectSetId}"><c:
      out
      value="${image.projectSetName}"/>
      </html:option>
    </c:forEach>
    </html:select>
  </td>
</tr>
<tr>
  <td>
    OLD IMAGE SETS<br/>
    <span class="font-supersmall">Tick
    the checkbox of Image set you want
    to remove</span>
    <br/><br/>
    <c:forEach var="imageset"
    items="${ExperimentDetailsForm.imag
    es}">
      <input type="checkbox"
      name="myImageSets"
      value="${imageset.projectSetId}"
      />
      <c:out
      value="${imageset.projectSetName}
      "/><br/>
    </c:forEach>
  </td>
</tr>
<tr>
  <td colspan="2">
  <table class="table-content"
  id="expComponentId">
    <tr>
      <th colspan="3">Experiment
      Components</th>
    </tr>
  <c:choose>
    <c:when
    test="${fn:length(ExperimentDetails
    Form.experimentComponents) gt 0}">
    <c:forEach var="detail"
    items="${ExperimentDetailsForm.expe
    rimentComponents}"
    varStatus="loop">
    <tr style="background-
    color:#E1E8F6;">
      <td>
      <table>
        <tr>
          <td># ${loop.index+1}
          <input type="hidden"
          name="componentId"
          value="${detail.experimentComp
          onentId}" />
          </td>
          <td align="right"><input
          type="checkbox"
          name="deleteComponent"
          value="${detail.experimentComp
          onentId}" /> Delete</td>
        </tr>
        <tr>
          <td class="font-
          bold">Condition</td>
          <td><input type="text"
          name="condition" size="90"
          value="${detail.conditionValue
          }" /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Method</td>
          <td><input type="text"
          name="methodName" size="90"
          value="${detail.methodName}"
          /></td>
        </tr>
        <tr>
          <td class="font-bold"
          valign="top">Description</td>
          <td><input type="text"
          name="methodDesc" size="90"
          value="${detail.methodDescript
          ion}" /></td>
        </tr>
        <tr>
          <td class="font-bold">Variable
          Type</td>
          <td><input type="text"
          name="varType" size="90"
          value="${detail.variableType}"
          /></td>
        </tr>
        <tr>
          <td class="font-
          bold">Value</td>
          <td><input type="text"
          name="varValue" size="90"
          value="${detail.variableValue}
          " /></td>
        </tr>
        <tr>
          <td class="font-bold">Chemical
          Activator</td>
          <td><input type="text"
          name="chemActivator" size="90"
          value="${detail.chemActivatorN
          ame}" /></td>
        </tr>
      </table>
      </td>
    </tr>
    </c:forEach>
    </c:when>
    <c:otherwise>
    <tr>
      <td class="font-italic"
      colspan="2">No Experimental
      Components Yet</td>
    </tr>
    </c:otherwise>
  </c:choose>
  </table>
  </td>
</tr>
</table>
<br/>
    
<html:submit property="command"
value="Submit" styleClass="button" />
<html:reset property="command"
value="Reset" styleClass="button" />
<html:cancel property="command"
value="Cancel" styleClass="button" />
</td>
</tr>
<tr>
  <td height="50px"> </td>
</tr>
</table>
</td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
```

```
    </td>
   </tr>
  </table>
  </html:form>
</body>
</html>
```

editProject.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/EditProject"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
  <tr>
  <td> </td>
  </tr>
  <tr valign="top">
  <td class="sidebar">
  <jsp:include page="/rd_virui/sidebar.jsp" />
  </td>
  <td>
  <table class="content">
  <tr>
  <td class="breadcrumb">
  <html:link page="/Home.do"><bean:message
  key="button.virho.home" /></html:link>
  &#187;
  <html:link page="/ViewProjects.do">Virho
  Experiments Projects</html:link> &#187;
  <html:link
  page="/ViewProjectExperimentDetails.do?proje
  ct=${project}">${project}</html:link> &#187;
  <span class="breadcrumb-selected">Edit
  Project</span>
  </td>
  </tr>
  <tr>
  <td> </td>
  </tr>
  <tr>
  <td class="content-title-form"
  colspan="2">Edit Project</td>
  </tr>
  <tr>
  <td>
  <table class="content-desc-form"
  style="height:300px">
  <tr>
  <td class="note" colspan="2">
  Note: All fields that are marked with * are
  required
  </td>
  </tr>
  <tr>
  <td colspan="2"><span class="error-text-
  big"><html:errors property="errorMessage"
  /></span></td>
</tr>
<tr>
<td width="150px"><bean:message
key="label.common.project" /> *</td>
<td>
<html:text property="projectName" size="65"
style="remove-error-field"
errorStyleClass="error-field"
errorKey="org.apache.struts.action.ERROR" />
<span class="error-text"><html:errors
property="projectName" /></span>
<html:hidden property="projectOldName"
value="${ProjectDetailsForm.projectOldName}"
/>
<html:hidden property="projectId"
value="${ProjectDetailsForm.projectId}" />
</td>
</tr>
<tr>
<td valign="top"><bean:message
key="label.common.description" /></td>
<td>
<html:textarea property="description"
rows="7" cols="75" style="remove-error-
field" errorStyleClass="error-field"
errorKey="org.apache.struts.action.ERROR" />
<span class="error-text"><html:errors
property="description" /></span>
</td>
</tr>
<tr>
<td><bean:message key="label.common.status"
/> *</td>
<td>
<html:select property="status"
style="remove-error-field"
errorStyleClass="error-field"
errorKey="org.apache.struts.action.ERROR">
<html:option value="Default"
styleClass="font-gray">Select
Status</html:option>
<html:option
value="Planned">Planned</html:option>
<html:option value="On Going">On
Going</html:option>
<html:option
value="Completed">Completed</html:option>
</html:select>
<span class="error-text"><html:errors
property="status" /></span>
</td>
</tr>
</table>
<br/>
    
<html:submit property="command"
value="Submit" styleClass="button" />
<html:reset property="command" value="Reset"
styleClass="button" />
<html:cancel property="command"
value="Cancel" styleClass="button" />
</td>
</tr>
<tr>
<td height="50px"> </td>
</tr>
</table>
</td>
</tr>
<tr class="footer">
<td colspan="2">
&copy; <bean:message key="virho.copyright"
/>
</td>
</tr>
</table>
```

```
  </html:form>
</body>
</html>
```

viewExperimentDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  $(document).ready(function() {
    $(".confirm").easyconfirm();
    $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/ViewExperimentDetails"
  method="POST" styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewProjects.do">Virho
            Experiments Projects</html:link>
            &#187;
            <html:link
            page="/ViewProjectExperimentDetails.
            do?project=${project}">${project}</h
            tml:link> &#187;
            <span class="breadcrumb-
            selected">${experiment}</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Experiment Details</td>
        </tr>
        <tr>
          <td>
          <c:set var="privilege"
          value="${ExperimentDetailsForm.privile
          ge}" />
          <html:hidden property="experimentId"
          value="${ExperimentDetailsForm.experim
          entId}" />
          <html:hidden property="projectId"
          value="${ExperimentDetailsForm.project
          Id}" />
          <table class="content-desc-form">
            <tr>
              <td>
              <table class="description-
              container">
                <tr>
                  <td class="font-bold"
                  width="250px">Experiment
                  Name</td>
                  <td><c:out
                  value="${ExperimentDetailsForm.
                  experimentName}" /></td>
                </tr>
                <tr>
                  <td class="font-
                  bold">Author</td>
                  <td><c:out
                  value="${ExperimentDetailsForm.
                  author}" /></td>
                </tr>
                <tr>
                  <td class="font-
                  bold">Description</td>
                  <td>
                    <c:out
                    value="${ExperimentDetailsForm
                    .description}" />
                    <c:if test="${empty
                    ExperimentDetailsForm.descript
                    ion}" >-</c:if>
                  </td>
                </tr>
                <tr>
                  <td class="font-bold">Virus
                  Used</td>
                  <td>
                    <c:out
                    value="${ExperimentDetailsForm
                    .virus}" />
                    <c:if test="${empty
                    ExperimentDetailsForm.virus}"
                    >-</c:if>
                  </td>
                </tr>
                <tr>
```

```
    <td class="font-
    bold">Experiment Date</td>
    <td><c:out
    value="${ExperimentDetailsForm.
    experimentDate}" /></td>
  </tr>
  <tr>
    <td class="font-
    bold">Measurement Interval</td>
    <td>
    <c:choose>
      <c:when
      test="${ExperimentDetailsForm.
      measurementInterval eq 0.0}"
      >-</c:when>
      <c:otherwise><c:out
      value="${ExperimentDetailsForm
      .measurementInterval}"
      /></c:otherwise>
    </c:choose>
    </td>
  </tr>
<c:choose>
  <c:when test="${not empty
  ExperimentDetailsForm.timePoints}
  " >
  <tr>
    <td class="font-bold"
    rowspan="2" valign="top">Time
    Points</td>
    <td>
      <c:out
      value="${ExperimentDetailsForm
      .timePoints}" />
    </td>
  </tr>
  <tr>
    <td>
      Hour: <c:out
      value="${ExperimentDetailsForm
      .tpHours}"/>
      Minute: <c:out
      value="${ExperimentDetailsForm
      .tpMinutes}"/>
      Second: <c:out
      value="${ExperimentDetailsForm
      .tpSeconds}"/>
    </td>
  </tr>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-bold"
    valign="top">Time Points</td>
    <td>-</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td class="font-
    bold">Biological
    Replicates</td>
    <td>
  <c:choose>
    <c:when
    test="${ExperimentDetailsForm.b
    iologicalReplicates eq 0}" >-
    </c:when>
    <c:otherwise><c:out
    value="${ExperimentDetailsForm.
    biologicalReplicates}"
    /></c:otherwise>
  </c:choose>
    </td>
  </tr>
  <tr>
    <td class="font-bold">Technical
    Replicates</td>
    <td>
  <c:choose>
    <c:when
    test="${ExperimentDetailsForm.t
    echnicalReplicates eq 0}" >-
    </c:when>
    <c:otherwise><c:out
    value="${ExperimentDetailsForm.
    technicalReplicates}"
    /></c:otherwise>
  </c:choose>
    </td>
  </tr>
  <tr>
    <td class="font-
    bold">Experiment Photos </td>
  <c:forEach var="image"
  items="${ExperimentDetailsForm.imag
  es}" varStatus="loop">
  <c:choose>
    <c:when test="${loop.index eq
    0}">
      <td>
    </c:when>
    <c:otherwise>
    <tr>
      <td> </td>
      <td>
    </c:otherwise>
  </c:choose>
       <html:link
       page="/ViewImageSetDetails.do?
       module=VirhoImages&projectSet=
       ${image.projectSetName}" >
       <span class="font-blue"><c:out
       value="${image.projectSetName}
       " /></span>
       </html:link>
      </td>
    </tr>
  </c:forEach>
  </table>
  </td>
</tr>
<tr>
  <td colspan="2">
  <table class="table-content">
    <tr>
      <th colspan="3">Experiment
      Components</th>
    </tr>
  <c:choose>
    <c:when
    test="${fn:length(ExperimentDetai
    lsForm.experimentComponents) gt
    0}">
    <c:forEach var="detail"
    items="${ExperimentDetailsForm.ex
    perimentComponents}"
    varStatus="loop">
    <tr style="background-
    color:#E1E8F6;">
      <td align="center" class="font-
      bold">${loop.index+1}</td>
      <td>
      <table width="100%">
        <tr><td><b>Condition :
        </b><c:out
        value="${detail.conditionValue
        }" /></td></tr>
        <tr><td><b>Method Name :
        </b><c:out
        value="${detail.methodName}"
        /></td></tr>
```

```
            <tr><td><b>Description :
            </b><c:out
            value="${detail.methodDescript
            ion}" /></td></tr>
            <tr><td><b>Variable Type :
            </b><c:out
            value="${detail.variableType}"
            /></td></tr>
            <tr><td><b>Value : </b><c:out
            value="${detail.variableValue}
            " /></td></tr>
            <tr><td><b>Chemical Activator
            : </b><c:out
            value="${detail.chemActivatorN
            ame}" /></td></tr>
          </table>
          </td>
        </tr>
        </c:forEach>
        </c:when>
        <c:otherwise>
        <tr>
          <td class="font-italic"
          colspan="2">No Experimental
          Components Yet</td>
        </tr>
        </c:otherwise>
      </c:choose>
      </table>
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
  <c:choose>
    <c:when test="${privilege ge 3}">
      <html:submit property="command"
      value="Edit Experiment"
      styleClass="button-long-blue" />
      <html:submit property="command"
      value="Delete Experiment"
      styleId="delete" styleClass="button-
      long-blue confirm" />
      <html:submit property="command"
      value="Add Experiment Component"
      styleClass="button-superlong-blue"
      />
    </c:when>
    <c:otherwise>
      <html:submit property="command"
      value="Send Membership Request"
      styleClass="button-superlong-blue"
      />
    </c:otherwise>
  </c:choose>
    <html:hidden property="privilege"
    value="${ViewProjectsForm.privilege}"/
    >
    </td>
  </tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

viewProjectDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/fmt"
prefix="fmt" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  $(document).ready(function() {
    $(".confirm").easyconfirm();
    $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form
  action="/ViewProjectExperimentDetails"
  styleId="form" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
```

```
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content">
  <tr>
    <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewProjects.do">Virho
      Experiments Projects</html:link>
      &#187;
      <span class="breadcrumb-
      selected">${project}</span>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td class="content-title-form"
    colspan="2">Project Details</td>
  </tr>
  <tr>
    <td>
    <table class="content-desc-form">
      <tr>
        <td>
        <table class="description-
        container">
          <tr>
            <td>
              <span class="font-
              big"><bean:write
              name="ProjectDetailsForm"
              property="projectName"/></span
              >  
              <html:hidden
              property="projectId"
              value="${ProjectDetailsForm.pr
              ojectId}" />
              <html:hidden
              property="projectName"
              value="${ProjectDetailsForm.pr
              ojectName}" />
              <c:if
              test="${ProjectDetailsForm.pri
              vilege ge 3}" >
                <html:submit
                property="command"
                value="Edit Project"
                styleClass="button-remove"
                />
                <c:if
                test="${ProjectDetailsForm.p
                rivilege eq 4}" >
                  <html:submit
                  property="command"
                  value="Delete Project"
                  styleId="delete"
                  styleClass="button-remove
                  confirm"/>
                </c:if>
              </c:if>
              <br/>
              <span class="font-darkgray">
              Created by <bean:write
              name="ProjectDetailsForm"
              property="author"/> /&nbs
              p;
              Date added <bean:write
              name="ProjectDetailsForm"
              property="dateAdded"/> /&
              nbsp;
                Status <bean:write
                name="ProjectDetailsForm"
                property="status"/></span>
              </td>
            </tr>
            <tr>
              <td>
                <bean:write
                name="ProjectDetailsForm"
                property="description"/>
              </td>
            </tr>
          </table>
          </td>
        </tr>
        <tr>
          <td colspan="2">
          <table class="table-content">
            <tr>
              <th>Experiment</th>
              <th>Description</th>
            </tr>
          <c:choose>
            <c:when
            test="${fn:length(ProjectDetailsF
            orm.experiments) gt 0}">
            <c:forEach var="detail"
            items="${ProjectDetailsForm.exper
            iments}">
            <tr>
              <td>
                <html:link
                page="/ViewExperimentDetails.d
                o?projectId=${ProjectDetailsFo
                rm.projectId}&experiment=${det
                ail.experimentName}">
                <span class="font-blue"><c:out
                value="${detail.experimentName
                }" /></span></html:link>
              </td>
              <td><c:out
              value="${detail.description}"
              /></td>
            </tr>
            </c:forEach>
            </c:when>
            <c:otherwise>
            <tr>
              <td class="font-italic"
              colspan="2">No Experiments
              Yet</td>
            </tr>
            </c:otherwise>
          </c:choose>
          </table>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>
          
      <c:set var="privilege"
      value="${ProjectDetailsForm.privilege}
      " />
      <c:choose>
        <c:when test="${privilege ge 3}">
          <html:submit property="command"
          value="Add Experiment"
          styleClass="button-long-blue" />
          <c:if test="${privilege eq 4}">
```

```
            <html:submit property="command"
            value="View Members"
            styleClass="button-long-blue" />
          </c:if>
        </c:when>
        <c:otherwise>
          <html:submit property="command"
          value="Send Membership Request"
          styleClass="button-superlong-blue"
          />
        </c:otherwise>
      </c:choose>
      </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

viewProjects.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
 <script src="./files/jquery-
 1.8.0.js"></script>
 <script>
 function loadPage(project, id){
  if(id == 'editProject')
   window.open
   ('EditProject.do?project='+project,
   '_self', false);
  else if(id == 'addExperiment')
   window.open
   ('AddExperiment.do?project='+project,
   '_self', false);
 }
 </script>
</head>
<body>
 <html:form action="/ViewProjects"
 method="POST">
 <table class="body" align="center">
<jsp:include page="/rd_virui/header.jsp" />
 <tr>
  <td> </td>
 </tr>
 <tr valign="top">
  <td class="sidebar">
   <jsp:include
   page="/rd_virui/sidebar.jsp" />
  </td>
  <td>
  <c:if test="${(userLogin ne null) and
  ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or (statusLogin
  eq 'superadmin'))}">
  <table class="content">
   <tr>
    <td class="breadcrumb">
     <html:link
     page="/Home.do"><bean:message
     key="button.virho.home"
     /></html:link> &#187;
     <span class="breadcrumb-
     selected">Virho Experiments
     Projects</span>
    </td>
   </tr>
   <tr>
    <td> </td>
   </tr>
   <tr>
    <td class="content-title-form"
    colspan="2">List of Projects</td>
   </tr>
   <tr>
    <td>
    <table class="content-desc-form">
     <tr>
      <td>
      <table class="table-content">
       <tr>
        <th>Project</th>
        <th>Description</th>
       </tr>
       <c:choose>
        <c:when
        test="${fn:length(projects) gt
        0}">
        <c:forEach var="detail"
        items="${projects}">
        <tr>
         <td width="250px">
          <html:link
          page="/ViewProjectExperiment
          Details.do?project=${detail.
          projectName}">
          <span class="font-
          blue"><c:out
          value="${detail.projectName}
          " /></span></html:link><br/>
          <c:if
          test="${detail.privilege ge
          3}" >
           <html:submit
           property="command"
           value="Add Experiment"
           styleClass="button-remove"
           onclick="loadPage('${detail
           .projectName}',
           'addExperiment'); return
           false;" />
           <html:submit
           property="command"
           value="Edit Project"
           styleClass="button-remove"
           onclick="loadPage('${detail
           .projectName}',
```

```
                    'editProject'); return
                    false;" />
                    <c:if
                    test="${detail.privilege eq
                    4}" >
                      <c:set var="privilege"
                      value="${detail.privilege
                      }" />
                    </c:if>
                  </c:if>
                </td>
                <td><c:out
                value="${detail.description}"
                /></td>
              </tr>
              </c:forEach>
              </c:when>
              <c:otherwise>
              <tr>
                <td class="font-italic"
                colspan="2">No Projects
                Yet</td>
              </tr>
              </c:otherwise>
            </c:choose>
            </table>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>
            
        <c:if test="${(privilege eq 4) or
        (requestScope.isLead eq true)}" >
          <html:submit property="command"
          value="Add Project"
          styleClass="button-long-blue" />
        </c:if>
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/VirhoExperimentsUsers"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewProjects.do">Virho
            Experiments Projects</html:link>
            &#187;
            <html:link
            page="/ViewProjectExperimentDetails.
            do?project=${project}">${project}</h
            tml:link> &#187;
            <span class="breadcrumb-
            selected">View Members</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">List of Project
          Members</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td>
              <table class="letterList">
                <tr>
                  <td>
                  <c:forEach var="letter"
                  items="A,B,C,D,E,F,G,H,I,J,K,L,
                  M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
                  varStatus="loop">
                  <c:choose>
```

```
      <c:when
      test="${ViewUsersForm.letters[
      loop.index] eq letter}">
        <html:link
        page="/VirhoExperimentsUsers
        .do?list=${letter}&projectId
        =${requestScope.projectId}">
        <span class="font-
        letter">${letter}</span></ht
        ml:link> 
      </c:when>
      <c:otherwise>
        ${letter} 
      </c:otherwise>
    </c:choose>
    </c:forEach>
    <html:link
    page="/VirhoExperimentsUsers.do
    ?list=&projectId=${requestScope
    .projectId}">View
    All</html:link>
    </td>
  </tr>
 </table>
 </td>
</tr>
<tr>
 <td>
 <table class="table-content">
  <tr>
   <th>Name</th>
   <th>Username</th>
   <th>Affiliation</th>
   <th>Role</th>
  </tr>
 <c:choose>
  <c:when
  test="${fn:length(ViewUsersForm.u
  sers) gt 0}">
  <c:forEach var="user"
  items="${ViewUsersForm.users}">
  <tr>
   <td>
  <c:choose>
   <c:when
   test="${requestScope.privilege
   eq 4}" >
    <html:link
    page="/UserProfileDetails.do?u
    ser=${user.username}" >
    <span class="font-
    blue">${user.firstName} $
    {user.lastName}</span>
    </html:link>
   </c:when>
   <c:otherwise>
    ${user.firstName} ${user.
    lastName}
   </c:otherwise>
  </c:choose>
   </td>
   <td>${user.username}</td>
   <td>${user.affiliation}</td>
   <td>
  <c:choose>
   <c:when
   test="${(requestScope.privilege
   eq 4) and (user.username ne
   userLogin)}">
   <select name="role">
    <option value="Remove
    Privilege">Remove
    Privilege</option>
   <c:forEach var="roleName"
   items="${requestScope.roles}">
    <option value="${roleName}"
    <c:if test="${roleName eq
    user.projectRole}">selected="s
    elected"</c:if>>${roleName}</o
    ption>
   </c:forEach>
   </select>
   <input type="hidden"
   name="oldRole"
   value="${user.projectRole}" />
   <input type="hidden"
   name="userName"
   value="${user.username}" />
  </c:when>
  <c:otherwise>
   ${user.projectRole}
  </c:otherwise>
 </c:choose>
 <input type="hidden"
 name="projectId"
 value="${requestScope.projectId}"
 />
    </td>
  </tr>
  </c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
   <td class="font-italic"
   colspan="4">
    There are no members yet
   </td>
  </tr>
  </c:otherwise>
 </c:choose>
 </table>
 </td>
</tr>
</table>
</td>
</tr>
<tr>
 <td> </td>
</tr>
<tr>
 <td>
      
  <c:if test="${requestScope.privilege
  eq 4}" >
   <html:submit property="command"
   value="Change Privilege Role"
   styleClass="button-superlong-blue"
   />
  </c:if>
 </td>
</tr>
<tr>
 <td> </td>
</tr>
</table>
</c:if>
</td>
</tr>
<tr>
 <td> </td>
</tr>
<tr class="footer">
 <td colspan="2">
  &copy; <bean:message
  key="virho.copyright" />
 </td>
</tr>
</table>
</html:form>
</body>
</html>
```

VIRHO IMAGES

addImage.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/fmt"
prefix="fmt" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <link rel="stylesheet" type="text/css"
  href="./files/datechooser.css"
  media="screen"/>
  <script type="text/javascript"
  src="./files/date-functions.js"></script>
  <script type="text/javascript"
  src="./files/datechooser.js"></script>
  <script type="text/javascript"
  src="./files/javascript.js"></script>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  function displayReferences(){
    window.open("DisplayReferences.do","Referen
    ces","alwaysRaised=yes, height=700,
    width=800 location=no, scrollbars=yes");
  }

  function clearReference(){
    document.getElementById("refTitle").value =
    "";
    document.getElementById("refId").value =
    "";
  }
  </script>
</head>
<body onload="showMetadata()">
  <html:form action="/AddImage" method="POST"
  enctype="multipart/form-data" styleId="form"
  >
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
<tr>
  <td class="breadcrumb">
    <html:link
    page="/Home.do"><bean:message
    key="button.virho.home"
    /></html:link> &#187;
    <html:link
    page="/ViewImageSets.do">Virho Image
    Sets</html:link> &#187;
    <html:link
    page="/ViewImageSetDetails.do?projec
    tSet=${projectSet}">${projectSet}</h
    tml:link> &#187;
    <span class="breadcrumb-
    selected">Add Images</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Add Images</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td colspan="2">
        Click the 'Add to Queue' button
        to include the image and its
        metadata for upload. Click
        'Upload All' button to upload the
        images listed in the queue.
      </td>
    </tr>
    <tr>
      <td class="note" colspan="2">
        Note: All fields marked with
        asterisk (*) are required. You
        can only upload files of type
        jpg, gif, png, bmp, tiff, svg,
        flv, avi, mp4, wmv, exe, and jar.
        For image files and .ImageJ plug-
        ins, you can only upload up to
        4MB. For video files, you can
        upload up to 20MB.
        <p>Please refrain from refreshing
        the page, or navigating to
        another page without canceling
        this page</p>
      </td>
    </tr>
    <tr>
      <td colspan="2">
      <table class="table-content">
        <tr>
          <th colspan="6"
          align="left">Image Queue</th>
        </tr>
        <tr style="background-
        color:#E1E8F1;">
          <td class="font-bold">File</td>
          <td class="font-
          bold">Description</td>
          <td class="font-
          bold">Accessibility</td>
          <td class="font-
          bold">Category</td>
          <td width="15px"> </td>
          <td width="15px"> </td>
        </tr>
      <c:choose>
        <c:when
        test="${fn:length(sessionScope.im
        ageVOs) gt 0}">
```

```
    <c:forEach var="image"
    items="${sessionScope.imageVOs}"
    varStatus="loop">
    <tr>
      <td><c:out
      value="${image.imagePath}"
      /></td>
      <td><c:out
      value="${image.description}"
      /></td>
      <td><c:out
      value="${image.accessibility}"
      /></td>
      <td><c:out
      value="${image.category}"
      /></td>
      <td><html:link
      page="/AddImage.do?edit=${loop.
      index}" ><img
      src="./images/edit.png"
      width="13px" height="13px"
      title="Edit"
      /></html:link></td>
      <td><html:link
      page="/AddImage.do?delete=${loo
      p.index}" ><img
      src="./images/delete2.png"
      width="13px" height="13px"
      title="Remove"
      /></html:link></td>
    </tr>
    </c:forEach>
    </c:when>
    <c:otherwise>
    <tr>
      <td class="font-italic"
      colspan="6">No Uploaded Images
      Yet</td>
    </tr>
    </c:otherwise>
  </c:choose>
  </table>
  </td>
</tr>
<tr>
  <td colspan="2" align="right">
    <html:submit property="command"
    value="Upload Images"
    styleClass="button-long-white" />
    <html:cancel property="command"
    value="Cancel"
    styleClass="button-long-white" />
  </td>
</tr>
<tr>
  <td colspan="2"><hr/></td>
</tr>
<tr>
  <td class="font-bold"
  width="200px">Image File * </td>
  <td>
<c:choose>
  <c:when test="${isEdit eq false}">
    <html:file property="file"
    style="remove-error-field-file"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
    <span class="error-
    text"><html:errors
    property="file" /></span>
  </c:when>
  <c:otherwise>
    <c:out
    value="${ImageForm.imagePath}"></
    c:out>
    <html:hidden property="imagePath"
    value="${ImageForm.imagePath}" />
    <html:hidden
    property="imagePathDB"
    value="${ImageForm.imagePathDB}"
    />
    <input type="hidden"
    name="imgIndex"
    value="${requestScope.imgIndex}"
    />
  </c:otherwise>
</c:choose>
  </td>
</tr>
<tr>
  <td class="font-bold"
  valign="top">Description</td>
  <td><html:textarea
  property="description" cols="67"
  rows="4"></html:textarea></td>
</tr>
<tr>
  <td class="font-bold">Accessibilty
  * </td>
  <td>
  <html:select
  property="accessibility"
  style="remove-error-field-file"
  errorStyleClass="error-field"
  errorKey="org.apache.struts.action.
  ERROR" >
    <html:option value="Default"
    styleClass="font-gray">Select
    Accessibility</html:option>
    <html:option
    value="Public">Publicly
    Available</html:option>
    <html:option
    value="Restricted">Restricted to
    Members</html:option>
  </html:select>
  <span class="error-
  text"><html:errors
  property="accessibility" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Image
  Category *</td>
  <td>
  <html:select property="category"
  styleId="category" style="remove-
  error-field-file"
  errorStyleClass="error-field"
  errorKey="org.apache.struts.action.
  ERROR" onchange="showMetadata()" >
    <html:option value="Default"
    styleClass="font-gray">Select
    Image Category</html:option>
    <html:option
    value="General">General</html:opt
    ion>
    <html:option value="Infection
    Assays">Infection
    Assays</html:option>
    <html:option value="Cell
    Culture">Cell
    Culture</html:option>
    <html:option
    value="Fluorenscence">Fluorenscen
    ce</html:option>
    <html:option value="Brightfield
    Microscopy">Brightfield
    Microscopy</html:option>
```

```
    <html:option
    value="Immunofluorenscence">Immun
    ofluorenscence</html:option>
    <html:option value="Electron
    Micrograph">Electron
    Micrograph</html:option>
    <html:option
    value="Video">Video</html:option>
    <html:option value="ImageJ Plug-
    in">ImageJ Plug-in</html:option>
  </html:select>
  <span class="error-
  text"><html:errors
  property="category" /></span>
  </td>
</tr>
<tr id="version"
style="display:none">
  <td class="font-bold">Version</td>
  <td><html:text property="version"
  styleId="versionVal" size="50" />
  </td>
</tr>
<tr>
  <td class="font-bold">Link a
  Reference</td>
  <td>
    <html:text property="reference"
    size="50" readonly="true"
    styleId="refTitle" />
    <html:hidden
    property="referenceId"
    styleId="refId" /> 
    <a href="javascript:void(1)"
    onclick="displayReferences()">
    <img src="./images/reference.png"
    height="16px" width="16px"
    title="Choose Reference"/>
    </a>
     
    <a href="javascript:void(1)"
    onclick="clearReference()">
    <img src="./images/delete2.png"
    height="15px" width="15px"
    title="Remove Reference"/>
    </a>
  </td>
</tr>
<tr id="startDate"
style="display:none">
  <td class="font-bold">Actual Start
  Date</td>
  <td>
    <html:text property="startDate"
    size="20" styleId="sd"
    readonly="true"
    onfocus="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" />
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan1"
    class="dateChooser"
    style="display: none; visibility:
    hidden; width: 200px;"></div>
  </td>
</tr>
<tr id="endDate"
style="display:none">
  <td class="font-bold">Expected End
  Date</td>
  <td>
    <html:text property="endDate"
    size="20" styleId="ed"
    readonly="true"
    onfocus="showChooser(this, 'ed',
    'chooserSpan2', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);"/>
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'ed',
    'chooserSpan2', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan2"
    class="dateChooser"
    style="display: none; visibility:
    hidden; width: 200px;"> </div>
  </td>
</tr>
<tr id="cellType"
style="display:none">
  <td class="font-bold">Cell Type
  Used</td>
  <td><html:text property="cellType"
  styleId="cellTypeVal" size="50"
  /></td>
</tr>
<tr id="virusType"
style="display:none">
  <td class="font-bold">Virus Type
  Used</td>
  <td><html:text property="virusType"
  styleId="virusTypeVal" size="50"
  /></td>
</tr>
<tr id="infection"
style="display:none">
  <td class="font-bold">Multiplicity
  of Infection</td>
  <td><html:text
  property="multOfInfection"
  styleId="infectionVal" size="50"
  /></td>
</tr>
<tr id="medium"
style="display:none">
  <td class="font-bold">Medium
  Used</td>
  <td><html:text property="medium"
  styleId="mediumVal" size="50"
  /></td>
</tr>
<tr id="temperature"
style="display:none">
  <td class="font-
  bold">Temperature</td>
  <td><html:text
  property="temperature"
  styleId="temperatureVal" size="50"
  /></td>
</tr>
<tr id="co2level"
style="display:none">
  <td class="font-
  bold">CO<sub>2</sub> Level</td>
  <td><html:text property="CO2Level"
  styleId="co2levelVal" size="50"
  /></td>
</tr>
<tr id="owner" style="display:none">
  <td class="font-bold">Owner</td>
  <td><html:text property="owner"
  styleId="ownerVal" size="50"
  /></td>
</tr>
<tr id="container"
style="display:none">
```

```
      <td class="font-bold">Container
      Used</td>
      <td><html:text property="container"
      styleId="containerVal" size="50"
      /></td>
    </tr>
    <tr id="phenolRed"
    style="display:none">
      <td class="font-bold">Phenol Red
      Present?</td>
      <td>
        <html:radio property="phenolRed"
        value="Yes" />Yes 
        <html:radio property="phenolRed"
        value="No" />No
      </td>
    </tr>
    <tr id="magnification"
    style="display:none">
      <td class="font-bold">Magnification
      Used</td>
      <td><html:text
      property="magnification"
      styleId="magnificationVal"
      size="50" /></td>
    </tr>
    <tr id="objective"
    style="display:none">
      <td class="font-bold">Objective
      Used</td>
      <td><html:text property="objective"
      styleId="objectiveVal" size="50"
      /></td>
    </tr>
    <tr id="illuminationType"
    style="display:none">
      <td class="font-bold">Illumination
      Type</td>
      <td><html:text
      property="illuminationType"
      styleId="illuminationTypeVal"
      size="50" /></td>
    </tr>
    <tr id="illuminationLevel"
    style="display:none">
      <td class="font-bold">Illumination
      Level</td>
      <td><html:text
      property="illuminationLevel"
      styleId="illuminationLevelVal"
      size="50" /></td>
    </tr>
    <tr id="filter"
    style="display:none">
      <td class="font-bold">Filter
      Used</td>
      <td><html:text property="filter"
      styleId="filterVal" size="50"
      /></td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
  <c:choose>
    <c:when test="${isEdit eq false}">
      <html:submit property="command"
      value="Add to Queue"
      styleClass="button-long-blue" />
    </c:when>
    <c:otherwise>
      <html:submit property="command"
      value="Edit Metadata"
      styleClass="button-long-blue" />
      <html:cancel property="command"
      value="Cancel Edit"
      styleClass="button-long-blue" />
    </c:otherwise>
  </c:choose>
  </td>
</tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

addImageSet.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/AddImageSet"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
```

```
    <html:link
    page="/ViewImageSets.do">Virho Image
    Sets</html:link> &#187;
    <span class="breadcrumb-
    selected">Add Image Set</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Add Image Set</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td class="note" colspan="2">
        Note: All fields marked with
        asterisk (*) are required.
      </td>
    </tr>
    <tr>
      <td colspan="2"><span class="error-
      text-big"><html:errors
      property="errorMessage"
      /></span></td>
    </tr>
    <tr>
      <td class="font-bold"
      width="150px">Title * </td>
      <td>
        <html:text
        property="projectSetName"
        size="75" style="remove-error-
        field" errorStyleClass="error-
        field"
        errorKey="org.apache.struts.actio
        n.ERROR" />
        <span class="error-
        text"><html:errors
        property="projectSetName"
        /></span>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      colspan="2">Description</td>
    </tr>
    <tr>
      <td colspan="2">
        <html:textarea
        property="description" cols="90"
        rows="8"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Experiment</td>
      <td>
      <c:set var="project" value="" />
      <html:select
      property="experimentId">
        <html:option value="0"
        styleClass="font-gray">Select
        Experiment</html:option>
      <c:forEach var="experiment"
      items="${requestScope.experiments}"
      >
      <c:if test="${project ne
      experiment.projectName}">
        <html:option
        value="${experiment.projectId}"
        disabled="true" ><c:out
        value="${experiment.projectName}"
        /></html:option>
        <c:set var="project"
        value="${experiment.projectName}"
        />
      </c:if>
        <html:option
        value="${experiment.experimentId}
        " ><c:out
        value="${experiment.experimentNam
        e}"/></html:option>
      </c:forEach>
      </html:select>
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button-
    long-blue" />
    <html:reset property="command"
    value="Reset" styleClass="button-
    long-blue" />
    <html:cancel property="command"
    value="Cancel" styleClass="button-
    long-blue" />
  </td>
</tr>
</table>
</c:if>
</td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

addSavedView.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
```

```
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  function checkAll(value) {
    if(value == 'true')
      $('form input:checkbox').attr('checked',
      'checked');
    else
      $('form
      input:checkbox').removeAttr('checked');
  }

  function selectButton(id) {
    if(id=="oldView") {
      $('#oldView').attr('checked', 'checked');
      $('#newView').removeAttr('checked');
    } else if(id=="newView") {
      $('#newView').attr('checked', 'checked');
      $('#oldView').removeAttr('checked');
    }
  }

  $(document).ready(function() {
    $(".confirm").easyconfirm();
      $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/AddSavedView"
  method="POST" >
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewImageSets.do">Virho Image
            Sets</html:link> &#187;
            <html:link
            page="/ViewImageSetDetails.do?projec
            tSet=${projectSet}">${projectSet}</h
            tml:link> &#187;
            <span class="breadcrumb-
            selected">Include in Image
            View</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Include in Image View</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td colspan="5">
              <table class="description-
              container">
                <tr>
                  <td class="font-bold"
                  onclick="selectButton('newView'
                  )">
                    <html:radio
                    property="optionView"
                    value="New View"
                    styleId="newView" />
                    Save as :
                    <html:text property="viewName"
                    size="70px" style="remove-
                    error-field"
                    errorStyleClass="error-field"
                    errorKey="org.apache.struts.ac
                    tion.ERROR" />  
                  </td>
                </tr>
                <tr>
                  <td class="font-bold"
                  onclick="selectButton('oldView'
                  )">
                  <html:radio
                  property="optionView"
                  value="Old View"
                  disabled="${requestScope.disabl
                  ed}" styleId="oldView" />
                  Merge with :
                  <html:select
                  property="viewNameDB"
                  style="remove-error-field"
                  errorStyleClass="error-field"
                  errorKey="org.apache.struts.act
                  ion.ERROR">
                    <option value="Default"
                    class="font-gray">Select Old
                    View</option>
                  <c:forEach var="view"
                  items="${requestScope.views}"
                  varStatus="loop">
                    <option
                    value="${view.viewName}"
                    style="font-gray"><c:out
                    value="${view.viewName}"/></op
                    tion>
                  </c:forEach>
                  </html:select>
                  <c:forEach var="view"
                  items="${requestScope.views}">
                    <input type="hidden"
                    name="view${view.viewName}"
                    value="${view.viewId}" />
                  </c:forEach>
                  </td>
                </tr>
              </table>
              </td>
            </tr>
            <tr>
              <td colspan="5"><span class="error-
              text-big"><html:errors
```

```
    property="errorMessage"
    /></span></td>
  </tr>
<c:choose>
  <c:when
  test="${fn:length(SavedViewForm.imag
  esDB) gt 0}">
  <c:forEach var="imageDB"
  items="${SavedViewForm.imagesDB}"
  varStatus="loop">
  <c:if test="${loop.index mod 5 eq
  0}">
  <tr valign="top">
  </c:if>
  <c:set var="image"
  value="${SavedViewForm.images}" />
    <td align="center" width="220px"
    valign="middle">
  <c:choose>
    <c:when
    test="${fn:endsWith(imageDB, 'exe')
    or fn:endsWith(imageDB, 'avi') or
    fn:endsWith(imageDB, 'mp4') ||
    fn:endsWith(imageDB, 'wmv') or
    fn:endsWith(imageDB, 'jar')}">
      <img
      src="images/defaultThumb.gif"
      class="thumbnail" />
    </c:when>
    <c:when
    test="${fn:endsWith(imageDB,
    'flv')}">
      <object classid="clsid:d27cdb6e-
      ae6d-11cf-96b8-444553540000"
      codebase="http://download.macrome
      dia.com/pub/shockwave/cabs/flash/
      swflash.cab#version=9,0,0,0"
      width="100" height="75"
      id="VideoPlayer" align="middle">
        <param name="allowScriptAccess"
        value="*" />
        <param name="allowFullScreen"
        value="true" />
        <param name="movie"
        value="http://www.platipus.nl/f
        lvplayer/download/1.0/FLVPlayer
        .swf?video=http://www.bioinfo.m
        pg.de/virholex/repository/virho
        images/${imageDB}&autoplay=fals
        e" />
        <param name"quality"
        value="high" />
        <param name="bgcolor"
        value="#ffffff" />
        <embed
        src="http://www.platipus.nl/flv
        player/download/1.0/FLVPlayer.s
        wf?video=http://www.bioinfo.mpg
        .de/virholex/repository/virhoim
        ages/${imageDB}&autoplay=false"
        quality="high"
        bgcolor="#000000" width="100"
        height="75" name="VideoPlayer"
        align="middle"
        allowScriptAccess="*"
        allowFullScreen="true"
        type="application/x-shockwave-
        flash"
        pluginspage="http://www.macrome
        dia.com/go/getflashplayer" />
      </object>
    </c:when>
    <c:otherwise>
      <img
      src="repository/virhoimages/${ima
      geDB}" class="thumbnail" />
    </c:otherwise>
  </c:choose>
    <div class="font-supersmall"
    style="word-wrap: break-word;
    width: 120px;">
    <input type="checkbox" name="img-
    ${imageDB}" value="true" /><br/>
    <input type="hidden"
    name="myImages" value="${imageDB}"
    />
    ${image[loop.index]}
    </div>
    </td>
  <c:if test="${(loop.index mod 5 eq
  4) or
  (fn:length(SavedViewForm.imagesDB)
  eq loop.index+1)}">
    <c:if test="${loop.index mod 5 ne
    4}">
    <c:forEach var="counter" begin="1"
    end="${4-(loop.index mod 5)}">
    <td width="220px"> </td>
    </c:forEach>
    </c:if>
  </tr>
  </c:if>
  </c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-italic"
    colspan="5">No Saved View Images
    Yet</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td colspan="5">
      <span style="color:#0000FF">
      <a href="javascript:void(1)"
      onclick="checkAll('true')">Check
      All</a> |
      <a href="javascript:void(1)"
      onclick="checkAll('false')">Unche
      ck All</a> 
      </span>
      <span class="font-italic">With
      Selected : </span>  
      <img src="./images/delete.png"
      width="15px" height="15px"
      title="Delete"
      style="position:relative;
      left:5px; top:2px" />
      <html:submit property="command"
      value="Delete" styleId="delete"
      styleClass="button-small confirm"
      />
    </td>
  </tr>
</table>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button"
    />
    <html:reset property="command"
    value="Reset" styleClass="button" />
    <html:cancel property="command"
    value="Cancel" styleClass="button"
    />
```

```
        </td>
       </tr>
      </table>
      </c:if>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

editImage.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/fmt"
prefix="fmt" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <link rel="stylesheet" type="text/css"
  href="./files/datechooser.css"
  media="screen"/>
  <script type="text/javascript"
  src="./files/date-functions.js"></script>
  <script type="text/javascript"
  src="./files/datechooser.js"></script>
  <script type="text/javascript"
  src="./files/javascript.js"></script>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  function displayReferences(){
    window.open("DisplayReferences.do","Referen
    ces","alwaysRaised=yes, height=700,
    width=800 location=no, scrollbars=yes");
  }

  function clearReference(){
    document.getElementById("refTitle").value =
    "";
    document.getElementById("refId").value =
    "";
  }
  </script>
</head>
<body onload="showMetadata()">
<html:form action="/EditImage" method="POST"
enctype="multipart/form-data" styleId="form"
>
<table class="body" align="center">
<jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin
    eq 'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <html:link
          page="/ViewImageSets.do">Virho Image
          Sets</html:link> &#187;
          <html:link
          page="/ViewImageSetDetails.do?projec
          tSet=${projectSet}">${projectSet}</h
          tml:link> &#187;
          <html:link
          page="/ViewImage.do?image=${image}">
          ${image}</html:link> &#187;
          <span class="breadcrumb-
          selected">Edit Images</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2">Edit Images</td>
      </tr>
      <tr>
        <td>
        <table class="content-desc-form">
          <tr>
            <td class="note" colspan="2">
              Note: All fields marked with
              asterisk (*) are required.
            </td>
          </tr>
          <tr>
            <td class="font-bold"
            width="200px">Image File </td>
            <td>
              <c:out
              value="${ImageForm.imagePath}" />
              <html:hidden property="imagePath"
              value="${ImageForm.imagePath}" />
            </td>
          </tr>
          <tr>
            <td class="font-bold"
            valign="top">Description</td>
            <td><html:textarea
            property="description" cols="67"
            rows="4"></html:textarea></td>
          </tr>
          <tr>
            <td class="font-bold">Accessibilty
            * </td>
            <td>
```

```
    <html:select
    property="accessibility"
    style="remove-error-field-file"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.action.
    ERROR" >
      <html:option value="Default"
      styleClass="font-gray">Select
      Accessibility</html:option>
      <html:option
      value="Public">Publicly
      Available</html:option>
      <html:option
      value="Restricted">Restricted to
      Members</html:option>
    </html:select>
  <span class="error-
  text"><html:errors
  property="accessibility" /></span>
  </td>
</tr>
<tr>
  <td class="font-bold">Image Category
  *</td>
  <td>
  <html:select property="category"
  styleId="category" style="remove-
  error-field-file"
  errorStyleClass="error-field"
  errorKey="org.apache.struts.action.E
  RROR" onchange="showMetadata()" >
    <html:option value="Default"
    styleClass="font-gray">Select Image
    Category</html:option>
    <html:option
    value="General">General</html:optio
    n>
    <html:option value="Infection
    Assays">Infection
    Assays</html:option>
    <html:option value="Cell
    Culture">Cell Culture</html:option>
    <html:option
    value="Fluorenscence">Fluorenscence
    </html:option>
    <html:option value="Brightfield
    Microscopy">Brightfield
    Microscopy</html:option>
    <html:option
    value="Immunofluorenscence">Immunof
    luorenscence</html:option>
    <html:option value="Electron
    Micrograph">Electron
    Micrograph</html:option>
    <html:option
    value="Video">Video</html:option>
    <html:option value="ImageJ Plug-
    in">ImageJ Plug-in</html:option>
  </html:select>
  <span class="error-
  text"><html:errors
  property="category" /></span>
  </td>
</tr>
<tr id="version" style="display:none">
  <td class="font-bold">Version</td>
  <td><html:text property="version"
  styleId="versionVal" size="50" />
  </td>
</tr>
<tr>
  <td class="font-bold">Link a
  Reference</td>
  <td>
    <html:text property="reference"
    size="50" readonly="true"
    styleId="refTitle" />
    <html:hidden property="referenceId"
    value="${ImageForm.referenceId}"
    styleId="refId" /> 
    <a href="javascript:void(1)"
    onclick="displayReferences()">
    <img src="./images/reference.png"
    height="16px" width="16px"
    title="Choose Reference"/>
    </a>
     
    <a href="javascript:void(1)"
    onclick="clearReference()">
    <img src="./images/delete2.png"
    height="15px" width="15px"
    title="Remove Reference"/>
  </a>
  </td>
</tr>
<tr id="startDate"
style="display:none">
  <td class="font-bold">Actual Start
  Date</td>
  <td>
    <html:text property="startDate"
    size="20" styleId="sd"
    readonly="true"
    onfocus="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" />
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'sd',
    'chooserSpan1', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan1"
    class="dateChooser" style="display:
    none; visibility: hidden; width:
    200px;"></div>
  </td>
</tr>
<tr id="endDate" style="display:none">
  <td class="font-bold">Expected End
  Date</td>
  <td>
    <html:text property="endDate"
    size="20" styleId="ed"
    readonly="true"
    onfocus="showChooser(this, 'ed',
    'chooserSpan2', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);"/>
    <img src="./images/calendar.gif"
    onclick="showChooser(this, 'ed',
    'chooserSpan2', 1950, 2020,
    Date.patterns.ShortDatePattern,
    false);" width="16">
    <div id="chooserSpan2"
    class="dateChooser" style="display:
    none; visibility: hidden; width:
    200px;"> </div>
    </td>
  </tr>
  <tr id="cellType"
  style="display:none">
    <td class="font-bold">Cell Type
    Used</td>
    <td><html:text property="cellType"
    styleId="cellTypeVal" size="50"
    /></td>
  </tr>
  <tr id="virusType"
  style="display:none">
```

```
  <td class="font-bold">Virus Type
  Used</td>
  <td><html:text property="virusType"
  styleId="virusTypeVal" size="50"
  /></td>
</tr>
<tr id="infection"
style="display:none">
  <td class="font-bold">Multiplicity
  of Infection</td>
  <td><html:text
  property="multOfInfection"
  styleId="infectionVal" size="50"
  /></td>
</tr>
<tr id="medium"
style="display:none">
  <td class="font-bold">Medium
  Used</td>
  <td><html:text property="medium"
  styleId="mediumVal" size="50"
  /></td>
</tr>
<tr id="temperature"
style="display:none">
  <td class="font-
  bold">Temperature</td>
  <td><html:text
  property="temperature"
  styleId="temperatureVal" size="50"
  /></td>
</tr>
<tr id="co2level"
style="display:none">
  <td class="font-
  bold">CO<sub>2</sub> Level</td>
  <td><html:text property="CO2Level"
  styleId="co2levelVal" size="50"
  /></td>
</tr>
<tr id="owner" style="display:none">
  <td class="font-bold">Owner</td>
  <td><html:text property="owner"
  styleId="ownerVal" size="50"
  /></td>
</tr>
<tr id="container"
style="display:none">
  <td class="font-bold">Container
  Used</td>
  <td><html:text property="container"
  styleId="containerVal" size="50"
  /></td>
</tr>
<tr id="phenolRed"
style="display:none">
  <td class="font-bold">Phenol Red
  Present?</td>
  <td>
    <html:radio property="phenolRed"
    value="Yes" />Yes 
    <html:radio property="phenolRed"
    value="No" />No
  </td>
</tr>
<tr id="magnification"
style="display:none">
  <td class="font-bold">Magnification
  Used</td>
  <td><html:text
  property="magnification"
  styleId="magnificationVal"
  size="50" /></td>
</tr>
<tr id="objective"
style="display:none">
  <td class="font-bold">Objective
  Used</td>
  <td><html:text property="objective"
  styleId="objectiveVal" size="50"
  /></td>
</tr>
<tr id="illuminationType"
style="display:none">
  <td class="font-bold">Illumination
  Type</td>
  <td><html:text
  property="illuminationType"
  styleId="illuminationTypeVal"
  size="50" /></td>
</tr>
<tr id="illuminationLevel"
style="display:none">
  <td class="font-bold">Illumination
  Level</td>
  <td><html:text
  property="illuminationLevel"
  styleId="illuminationLevelVal"
  size="50" /></td>
</tr>
<tr id="filter"
style="display:none">
  <td class="font-bold">Filter
  Used</td>
  <td><html:text property="filter"
  styleId="filterVal" size="50"
  /></td>
</tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button-
    long-blue" />
    <html:reset property="command"
    value="Reset" styleClass="button-
    long-blue" />
    <html:cancel property="command"
    value="Cancel" styleClass="button-
    long-blue" />
  </td>
</tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

editImageSet.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
```

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/EditImageSet"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewImageSets.do">Virho Image
            Sets</html:link> &#187;
            <span class="breadcrumb-
            selected">Edit Image Set</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Edit Image Set</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td class="note" colspan="2">
                Note: All fields marked with
                asterisk (*) are required.
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td class="font-bold">Title * </td>
              <td>
                <html:text
                property="projectSetName"
                size="75" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="projectSetName"
                /></span>
                <html:hidden
                property="projectSetId"
                value="${ImageSetsForm.projectSet
                Id}" />
              </td>
            </tr>
            <tr>
              <td class="font-bold"
              colspan="2">Description</td>
            </tr>
            <tr>
              <td colspan="2">
                <html:textarea
                property="description" cols="90"
                rows="8"></html:textarea>
              </td>
            </tr>
            <tr>
              <td class="font-bold" width="180px"
              valign="top"
              rowspan="2">Experiment</td>
              <td>
              <c:set var="project" value="" />
              <html:select
              property="experimentId">
                <html:option value="Default"
                styleClass="font-gray">Select
                Experiment</html:option>
              <c:forEach var="experiment"
              items="${requestScope.experiments}"
              >
              <c:if test="${project ne
              experiment.projectName}">
                <html:option
                value="${experiment.projectId}"
                disabled="true" ><c:out
                value="${experiment.projectName}"
                /></html:option>
              <c:set var="project"
              value="${experiment.projectName}"
              />
              </c:if>
                <html:option
                value="${experiment.experimentId}
                "><c:out
                value="${experiment.experimentNam
                e}"/></html:option>
              </c:forEach>
              </html:select>
              </td>
            </tr>
            <tr>
              <td>
                OLD EXPERIMENT<br/>
                <span class="font-
                supersmall">Tick the checkbox of
                experiment you want to
                remove</span>
                <br/>
                <c:set var="project" value="" />
                <c:forEach var="experiment"
                items="${ImageSetsForm.experiment
                s}">
```

```
            <c:if test="${project ne
            experiment.projectName}">
              <br/><c:out
              value="${experiment.projectName
              }"/><br/>
              <c:set var="project"
              value="${experiment.projectName
              }" />
            </c:if>
            <input type="checkbox"
            name="myExperiments"
            value="${experiment.experimentId}
            " />
            <c:out
            value="${experiment.experimentNam
            e}"/><br/>
            </c:forEach>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>
          
      <html:submit property="command"
      value="Submit" styleClass="button-
      long-blue" />
      <html:reset property="command"
      value="Reset" styleClass="button-long-
      blue" />
      <html:cancel property="command"
      value="Cancel" styleClass="button-
      long-blue" />
    </td>
  </tr>
</table>
</c:if>
</td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

editSavedView.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>

  <script>
  function checkAll(value) {
    if(value == 'true')
      $('form input:checkbox').attr('checked',
      'checked');
    else
      $('form
      input:checkbox').removeAttr('checked');
  }

  function selectButton(id) {
    if(id=="oldView") {
      $('#oldView').attr('checked', 'checked');
      $('#newView').removeAttr('checked');
    } else if(id=="newView") {
      $('#newView').attr('checked', 'checked');
      $('#oldView').removeAttr('checked');
    }
  }

  $(document).ready(function() {
    $(".confirm").easyconfirm();
      $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/EditSavedView"
  method="POST" >
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewImageSets.do">Virho Image
            Sets</html:link> &#187;
```

```
    <html:link
    page="/ViewSavedViews.do">My Saved
    Views</html:link> &#187;
    <html:link
    page="/SavedViewDetails.do?savedView
    =${savedView}"><c:out
    value="${savedView}" /></html:link>
    &#187;
    <span class="breadcrumb-
    selected">Edit Saved View</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Edit Saved View</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td colspan="5">
      <table class="description-
      container">
        <tr>
          <td class="font-bold"
          onclick="selectButton('newView'
          )">
            <html:radio
            property="optionView"
            value="New View"
            styleId="newView" />
            Save as :
            <html:text property="viewName"
            size="70px" style="remove-
            error-field"
            errorStyleClass="error-field"
            errorKey="org.apache.struts.ac
            tion.ERROR" />  
          </td>
        </tr>
        <tr>
          <td class="font-bold"
          onclick="selectButton('oldView'
          )">
          <html:radio
          property="optionView"
          value="Old View"
          disabled="${requestScope.disabl
          ed}" styleId="oldView" />
          Merge with :
          <html:select
          property="viewNameDB"
          style="remove-error-field"
          errorStyleClass="error-field"
          errorKey="org.apache.struts.act
          ion.ERROR">
            <option value="Default"
            class="font-gray">Select Old
            View</option>
          <c:forEach var="view"
          items="${requestScope.views}"
          varStatus="loop">
          <c:if test="${view.viewName ne
          SavedViewForm.oldViewName}">
            <option
            value="${view.viewName}"
            style="font-gray"><c:out
            value="${view.viewName}"/></op
            tion>
          </c:if>
          </c:forEach>
          </html:select>
          <c:forEach var="view"
          items="${requestScope.views}">
          <input type="hidden"
          name="view${view.viewName}"
          value="${view.viewId}" />
          </c:forEach>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td colspan="5"><span class="error-
      text-big"><html:errors
      property="errorMessage"
      /></span></td>
    </tr>
  <c:choose>
    <c:when
    test="${fn:length(SavedViewForm.imag
    esDB) gt 0}">
    <c:forEach var="imageDB"
    items="${SavedViewForm.imagesDB}"
    varStatus="loop">
    <c:if test="${loop.index mod 5 eq
    0}">
    <tr valign="top">
    </c:if>
    <c:set var="image"
    value="${SavedViewForm.images}" />
      <td align="center" width="220px"
      valign="middle">
    <c:choose>
      <c:when
      test="${fn:endsWith(imageDB, 'exe')
      or fn:endsWith(imageDB, 'avi') or
      fn:endsWith(imageDB, 'mp4') ||
      fn:endsWith(imageDB, 'wmv') or
      fn:endsWith(imageDB, 'jar')}">
        <img
        src="images/defaultThumb.gif"
        class="thumbnail" />
      </c:when>
      <c:when
      test="${fn:endsWith(imageDB,
      'flv')}">
        <object classid="clsid:d27cdb6e-
        ae6d-11cf-96b8-444553540000"
        codebase="http://download.macrome
        dia.com/pub/shockwave/cabs/flash/
        swflash.cab#version=9,0,0,0"
        width="100" height="75"
        id="VideoPlayer" align="middle">
          <param name="allowScriptAccess"
          value="*" />
          <param name="allowFullScreen"
          value="true" />
          <param name="movie"
          value="http://www.platipus.nl/f
          lvplayer/download/1.0/FLVPlayer
          .swf?video=http://www.bioinfo.m
          pg.de/virholex/repository/virho
          images/${imageDB}&autoplay=fals
          e" />
          <param name"quality"
          value="high" />
          <param name="bgcolor"
          value="#ffffff" />
          <embed
          src="http://www.platipus.nl/flv
          player/download/1.0/FLVPlayer.s
          wf?video=http://www.bioinfo.mpg
          .de/virholex/repository/virhoim
          ages/${imageDB}&autoplay=false"
          quality="high"
          bgcolor="#000000" width="100"
```

```
        height="75" name="VideoPlayer"
        align="middle"
        allowScriptAccess="*"
        allowFullScreen="true"
        type="application/x-shockwave-
        flash"
        pluginspage="http://www.macrome
        dia.com/go/getflashplayer" />
      </object>
      </c:when>
      <c:otherwise>
        <img
        src="repository/virhoimages/${i
        mageDB}" class="thumbnail" />
      </c:otherwise>
    </c:choose>
      <div class="font-supersmall"
      style="word-wrap: break-word;
      width: 120px;">
      <input type="checkbox" name="img-
      ${imageDB}" value="true" /><br/>
      <input type="hidden"
      name="myImages"
      value="${imageDB}" />
      ${image[loop.index]}
      </div>
    </td>
  <c:if test="${(loop.index mod 5 eq
  4) or
  (fn:length(SavedViewForm.imagesDB)
  eq loop.index+1)}">
    <c:if test="${loop.index mod 5 ne
    4}">
    <c:forEach var="counter" begin="1"
    end="${4-(loop.index mod 5)}">
    <td width="220px"> </td>
    </c:forEach>
    </c:if>
  </tr>
  </c:if>
  </c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-italic"
    colspan="5">No Saved View Images
    Yet</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td colspan="5">
      <span style="color:#0000FF">
      <a href="javascript:void(1)"
      onclick="checkAll('true')">Check
      All</a> |
      <a href="javascript:void(1)"
      onclick="checkAll('false')">Unche
      ck All</a> 
      </span>
      <span class="font-italic">With
      Selected : </span>  
      <img src="./images/delete.png"
      width="15px" height="15px"
      title="Delete"
      style="position:relative;
      left:5px; top:2px" />
      <html:submit property="command"
      value="Delete" styleId="delete"
      styleClass="button-small confirm"
      />
    </td>
  </tr>
</table>
</td>
</tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>
              
          <html:submit property="command"
          value="Submit" styleClass="button"
          />
          <html:reset property="command"
          value="Reset" styleClass="button" />
          <html:cancel property="command"
          value="Cancel" styleClass="button"
          />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

viewImageDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>

  <script>
  $(document).ready(function() {
    $(".confirm").easyconfirm();
```

```
    $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/ViewImage" method="POST"
  styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewImageSets.do">Virho Image
            Sets</html:link> &#187;
            <html:link
            page="/ViewImageSetDetails.do?projec
            tSet=${projectSet}">${projectSet}</h
            tml:link> &#187;
            <span class="breadcrumb-
            selected">Image Metadata</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Image Metadata</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td style="color:#0000FF">
                <html:link
                page="/ViewImageSetDetails.do?pro
                jectSet=${projectSet}">Thumbnail
                View</html:link>
              </td>
              <td align="right">
              <c:forEach var="image"
              items="${requestScope.images}"
              varStatus="loop">
              <c:if test="${ImageForm.imagePathDB
              eq image}">
                <c:set var="imgPos"
                value="${loop.index}" />
              </c:if>
              </c:forEach>
              Image <c:out value="${imgPos+1}" />
              of <c:out
              value="${fn:length(requestScope.ima
              ges)}" />
            </td>
          </tr>
          <tr>
<td class="image-container"
colspan="2" align="center">
<table>
  <tr>
    <td width="100px">
    <c:set var="prevIndex"
    value="${imgPos-1}" />
    <c:if test="${prevIndex eq -
    1}"><c:set var="prevIndex"
    value="${fn:length(requestScope.i
    mages)-1}"/></c:if>
      <html:link
      page="/ViewImage.do?image=${req
      uestScope.images[prevIndex]}">
      <img src="images/arrowprev.png"
      style="cursor: pointer;" />
    </html:link>
    </td>
    <td>
  <c:choose>
    <c:when
    test="${fn:endsWith(ImageForm.ima
    gePathDB, 'exe') or
    fn:endsWith(ImageForm.imagePathDB
    , 'avi') or
    fn:endsWith(ImageForm.imagePathDB
    , 'mp4') ||
    fn:endsWith(ImageForm.imagePathDB
    , 'wmv') or
    fn:endsWith(ImageForm.imagePathDB
    , 'jar')}">
      <img
      src="images/defaultThumb.gif"
      class="imageView" />
    </c:when>
    <c:when
    test="${fn:endsWith(ImageForm.ima
    gePathDB, 'flv')}">
      <object
      classid="clsid:d27cdb6e-ae6d-
      11cf-96b8-444553540000"
      codebase="http://download.macro
      media.com/pub/shockwave/cabs/fl
      ash/swflash.cab#version=9,0,0,0
      " width="360" height="270"
      id="VideoPlayer"
      align="middle">
        <param
        name="allowScriptAccess"
        value="*" />
        <param name="allowFullScreen"
        value="true" />
        <param name="movie"
        value="http://www.platipus.nl/
        flvplayer/download/1.0/FLVPlay
        er.swf?video=http://www.bioinf
        o.mpg.de/virholex/repository/v
        irhoimages/${ImageForm.imagePa
        thDB}&autoplay=false" />
        <param name"quality"
        value="high" />
        <param name="bgcolor"
        value="#ffffff" />
        <embed
        src="http://www.platipus.nl/fl
        vplayer/download/1.0/FLVPlayer
        .swf?video=http://www.bioinfo.
        mpg.de/virholex/repository/vir
        hoimages/${ImageForm.imagePath
        DB}&autoplay=false"
        quality="high"
        bgcolor="#000000" width="360"
        height="270"
        name="VideoPlayer"
        align="middle"
        allowScriptAccess="*"
```

```
        allowFullScreen="true"
        type="application/x-shockwave-
        flash"
        pluginspage="http://www.macrom
        edia.com/go/getflashplayer" />
      </object>
    </c:when>
    <c:when
    test="${fn:endsWith(ImageForm.ima
    gePathDB, 'svg')}">
      <img
      src="repository/virhoimages/${I
      mageForm.imagePathDB}"
      height="400px" width="400px" />
    </c:when>
    <c:otherwise>
      <img
      src="repository/virhoimages/${I
      mageForm.imagePathDB}"
      class="imageView" />
    </c:otherwise>
  </c:choose>
    </td>
    <td width="100px" align="right">
      <c:set var="nextIndex"
      value="${imgPos+1}" />
      <c:if test="${nextIndex eq
      fn:length(requestScope.images)}
      "><c:set var="nextIndex"
      value="0"/></c:if>
        <html:link
        page="/ViewImage.do?image=${re
        questScope.images[nextIndex]}"
        >
        <img
        src="images/arrownext.png"
        style="cursor: pointer;" />
        </html:link>
    </td>
  </tr>
</table>
</td>
</tr>
<tr>
  <td class="font-supersmall font-
  gray" colspan="2">
    Date Uploaded <c:out
    value="${ImageForm.dateUploaded}"
    />  / Uploaded by <c:out
    value="${ImageForm.uploadedBy}" />
  </td>
</tr>
<tr>
  <td colspan="2">
  <table class="description-
  container">
    <tr>
      <td class="font-bold"
      width="180px">Image Name</td>
      <td><c:out
      value="${ImageForm.imagePath}"
      /></td>
  </tr>
  <tr>
    <td class="font-bold">Category</td>
    <td><c:out
    value="${ImageForm.category}"
    /></td>
  </tr>
  <tr>
    <td class="font-bold"
    valign="top">Description</td>
    <td>
      <c:out
      value="${ImageForm.description}"
      />
      <c:if test="${empty
      ImageForm.description}">-</c:if>
    </td>
  </tr>
  <c:if test="${(ImageForm.category
  eq 'ImageJ Plug-in')}">
  <tr>
    <td class="font-bold"
    valign="top">Version</td>
    <td>
      <c:out
      value="${ImageForm.version}" />
      <c:if test="${empty
      ImageForm.version}">-</c:if>
    </td>
  </tr>
  </c:if>
  <tr>
    <td class="font-
    bold">Reference</td>
    <td>
      <html:link
      page="/ViewReferenceDetails.do?
      module=VirhoReferences&collecti
      on=${requestScope.referenceDeta
      ils.collection}&reference=${Ima
      geForm.reference}&refId=${Image
      Form.referenceId}"><c:out
      value="${ImageForm.reference}"
      /></html:link>
      <c:if test="${empty
      ImageForm.reference}">-</c:if>
    </td>
  </tr>
<c:if test="${(ImageForm.category eq
'Infection Assays') or
(ImageForm.category eq 'Cell
Culture') or (ImageForm.category eq
'Fluorenscence') or
(ImageForm.category eq 'Brightfield
Microscopy')}">
  <tr>
    <td class="font-bold">Start
    Date</td>
    <td>
      <c:out
      value="${ImageForm.startDate}"
      />
      <c:if test="${empty
      ImageForm.startDate}">-</c:if>
    </td>
  </tr>
<c:if test="${(ImageForm.category eq
'Infection Assays') or
(ImageForm.category eq 'Cell
Culture')}">
  <tr>
    <td class="font-bold">End
    Date</td>
    <td>
      <c:out
      value="${ImageForm.endDate}" />
      <c:if test="${empty
      ImageForm.endDate}">-</c:if>
    </td>
  </tr>
</c:if>
  <tr>
    <td class="font-bold">Cell
    Type</td>
    <td>
      <c:out
      value="${ImageForm.cellType}"
      />
      <c:if test="${empty
      ImageForm.cellType}">-</c:if>
```

```
  </td>
</tr>
<tr>
  <td class="font-bold">Virus
  Type</td>
  <td>
    <c:out
    value="${ImageForm.virusType}"
    />
    <c:if test="${empty
    ImageForm.virusType}">-</c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">Multiplicity of
  Infection</td>
  <td>
    <c:out
    value="${ImageForm.multOfInfect
    ion}" />
    <c:if test="${empty
    ImageForm.multOfInfection}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-bold">Medium</td>
  <td>
    <c:out
    value="${ImageForm.medium}" />
    <c:if test="${empty
    ImageForm.medium}">-</c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">Temperature</td>
  <td>
    <c:out
    value="${ImageForm.temperature}
    " />
    <c:if test="${empty
    ImageForm.temperature}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">CO<sub>2</sub> Level</td>
  <td>
    <c:out
    value="${ImageForm.CO2Level}"
    />
    <c:if test="${empty
    ImageForm.CO2Level}">-</c:if>
  </td>
</tr>
<tr>
  <td class="font-bold">Owner</td>
  <td>
    <c:out
    value="${ImageForm.owner}" />
    <c:if test="${empty
    ImageForm.owner}">-</c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">Container</td>
  <td>
    <c:out
    value="${ImageForm.container}"
    />
    <c:if test="${empty
    ImageForm.container}">-</c:if>
    </td>
  </tr>
<c:if test="${(ImageForm.category eq
'Fluorenscence') or
(ImageForm.category eq 'Brightfield
Microscopy')}">
  <tr>
    <td class="font-bold">Phenol
    Red</td>
    <td><c:out
    value="${ImageForm.phenolRed}"
    /></td>
  </tr>
  <tr>
    <td class="font-
    bold">Magnification</td>
    <td>
      <c:out
      value="${ImageForm.magnificatio
      n}" />
      <c:if test="${empty
      ImageForm.magnification}">-
      </c:if>
    </td>
  </tr>
  <tr>
    <td class="font-
    bold">Objective</td>
    <td>
      <c:out
      value="${ImageForm.objective}"
      />
      <c:if test="${empty
      ImageForm.objective}">-</c:if>
    </td>
  </tr>
  <tr>
    <td class="font-
    bold">Illumination Type</td>
    <td>
      <c:out
      value="${ImageForm.illumination
      Type}" />
      <c:if test="${empty
      ImageForm.illuminationType}">-
      </c:if>
    </td>
  </tr>
  <tr>
    <td class="font-
    bold">Illumination Level</td>
    <td>
      <c:out
      value="${ImageForm.illumination
      Level}" />
      <c:if test="${empty
      ImageForm.illuminationLevel}">-
      </c:if>
    </td>
  </tr>
  <tr>
    <td class="font-bold">Filter</td>
    <td>
      <c:out
      value="${ImageForm.filter}" />
      <c:if test="${empty
      ImageForm.filter}">-</c:if>
    </td>
  </tr>
</c:if>
</c:if>
</table>
</td>
</tr>
</table>
</td>
```

```
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>    
        <c:set var="privilege"
        value="${ImageForm.privilege}" />
      <c:choose>
        <c:when test="${privilege ge 2}" >
          <c:if test="${privilege ge 3}" >
            <html:submit property="command"
            value="Edit Metadata"
            styleClass="button-long-blue" />
            <html:submit property="command"
            value="Delete Image"
            styleId="delete"
            styleClass="button-long-blue
            confirm" />
          </c:if>
          <html:submit property="command"
          value="Download Image"
          styleClass="button-long-blue" />
        </c:when>
        <c:otherwise>
          <html:submit property="command"
          value="Send Membership Request"
          styleClass="button-superlong-blue"
          />
        </c:otherwise>
      </c:choose>
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

viewImageSetDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
 <script src="./files/jquery-
 1.8.0.js"></script>
 <script
 src="http://ajax.googleapis.com/ajax/libs/jq
 uery/1.3.2/jquery.min.js"
 type="text/javascript"></script>
 <script
 src="http://ajax.googleapis.com/ajax/libs/jq
 ueryui/1.7.2/jquery-ui.min.js"></script>
 <link rel="stylesheet" href="./files/jquery-
 ui.css" type="text/css" />
 <script src="./files/jquery.easy-confirm-
 dialog.js"></script>
 <script>

 function checkAll(value) {
  if(value == 'true')
    $('form input:checkbox').attr('checked',
    'checked');
  else
    $('form
    input:checkbox').removeAttr('checked');
 }

 $(document).ready(function() {
  $(".confirm").easyconfirm();
  $("#image").click(function() {
    return true;
  });
  $("#imageset").click(function() {
    return true;
  });
 });

 </script>
</head>
<body>
 <html:form action="/ViewImageSetDetails"
 method="POST" styleId="form">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin
    eq 'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <html:link
          page="/ViewImageSets.do">Virho Image
          Sets</html:link> &#187;
          <span class="breadcrumb-
          selected"><c:out
          value="${projectSet}" /></span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2"><c:out
```

```
  value="${ImageSetsForm.projectSetName}
  " /></td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td>
      <c:set var="privilege"
      value="${ImageSetsForm.privilege}"
      />
      <table class="description-
      container">
        <tr>
          <td colspan="2">
          <span class="font-
          big"><bean:write
          name="ImageSetsForm"
          property="projectSetName"/></sp
          an>  
          <c:if
          test="${ImageSetsForm.privilege
          ge 3}" >
            <html:submit
            property="command" value="Edit
            Image Set" styleClass="button-
            remove"  />
            <c:if
            test="${ImageSetsForm.privileg
            e eq 4}" >
              <html:submit
              property="command"
              value="Delete Image Set"
              styleId="imageset"
              styleClass="button-remove
              confirm" />
            </c:if>
          </c:if>
          <br/>
          <span class="font-darkgray">
          Created by <bean:write
          name="ImageSetsForm"
          property="addedBy"/> /&nbs
          p;
          Date added <bean:write
          name="ImageSetsForm"
          property="dateAdded"/> /&n
          bsp;
          Date modified <bean:write
          name="ImageSetsForm"
          property="dateModified"/></span
          >
          </td>
        </tr>
      <c:if test="${not empty
      ImageSetsForm.description}" >
        <tr>
          <td colspan="2">
            <bean:write
            name="ImageSetsForm"
            property="description"/>
          </td>
        </tr>
      </c:if>
        <tr>
          <td class="font-bold"
          width="130px"
          valign="top">Experiment Link :
          </td>
          <td>
          <c:forEach var="experiment"
          items="${ImageSetsForm.experime
          nts}" varStatus="loop">
            <html:link
            page="/ViewExperimentDetails.d
            o?module=VirhoExperiments&proj
            ectId=${experiment.projectId}&
            experiment=${experiment.experi
            mentName}" >
            <span class="font-blue"><c:out
            value="${experiment.experiment
            Name}" /></span>
            </html:link>
            <c:if test="${loop.index lt
            fn:length(ImageSetsForm.experi
            ments)}">
              <br/>
            </c:if>
          </c:forEach>
          </td>
        </tr>
      </table>
      </td>
    </tr>
  <c:choose>
    <c:when
    test="${fn:length(ImageSetsForm.imag
    es) gt 0}">
    <tr>
      <td>
      <table>
      <c:forEach var="imageDB"
      items="${ImageSetsForm.imagesDB}"
      varStatus="loop">
      <c:if test="${loop.index mod 5 eq
      0}">
        <tr valign="top">
      </c:if>
      <c:set var="image"
      value="${ImageSetsForm.images}" />
         <td align="center"
         width="220px" valign="middle">
          <html:link
          page="/ViewImage.do?image=${im
          ageDB}">
          <c:choose>
            <c:when
            test="${fn:endsWith(imageDB,
            'exe') or fn:endsWith(imageDB,
            'avi') or fn:endsWith(imageDB,
            'mp4') || fn:endsWith(imageDB,
            'wmv') or fn:endsWith(imageDB,
            'jar')}">
              <img
              src="images/defaultThumb.gif
              " class="thumbnail" />
            </c:when>
            <c:when
            test="${fn:endsWith(imageDB,
            'flv')}">
              <object
              classid="clsid:d27cdb6e-
              ae6d-11cf-96b8-444553540000"
              codebase="http://download.ma
              cromedia.com/pub/shockwave/c
              abs/flash/swflash.cab#versio
              n=9,0,0,0" width="100"
              height="75" id="VideoPlayer"
              align="middle">
                <param
                name="allowScriptAccess"
                value="*" />
                <param
                name="allowFullScreen"
                value="true" />
                <param name="movie"
                value="http://www.platipus.
                nl/flvplayer/download/1.0/F
                LVPlayer.swf?video=http://w
                ww.bioinfo.mpg.de/virholex/
                repository/virhoimages/${im
                ageDB}&autoplay=false" />
```

```
            <param name"quality"
            value="high" />
            <param name="bgcolor"
            value="#ffffff" />
            <embed
            src="http://www.platipus.nl
            /flvplayer/download/1.0/FLV
            Player.swf?video=http://www
            .bioinfo.mpg.de/virholex/re
            pository/virhoimages/${imag
            eDB}&autoplay=false"
            quality="high"
            bgcolor="#000000"
            width="100" height="75"
            name="VideoPlayer"
            align="middle"
            allowScriptAccess="*"
            allowFullScreen="true"
            type="application/x-
            shockwave-flash"
            pluginspage="http://www.mac
            romedia.com/go/getflashplay
            er" />
          </object>
        </c:when>
        <c:otherwise>
          <img
          src="repository/virhoimages/
          ${imageDB}"
          class="thumbnail" />
        </c:otherwise>
      </c:choose>
        </html:link>
        <div class="font-supersmall"
        style="word-wrap: break-word;
        width: 120px;">
        <c:if test="${privilege ge
        2}">
          <input type="checkbox"
          name="img-${imageDB}"
          value="true" /><br/>
          <input type="hidden"
          name="imagesList"
          value="${imageDB}" />
        </c:if>
        <html:link
        page="/ViewImage.do?image=${im
        ageDB}">
        <span class="font-image-
        link">${image[loop.index]}</sp
        an>
        </html:link>
        </div>
      </td>
    <c:if test="${(loop.index mod 5
    eq 4) or
    (fn:length(ImageSetsForm.images)
    eq loop.index+1)}">
      <c:if test="${loop.index mod 5
      ne 4}">
      <c:forEach var="counter"
      begin="1" end="${4-(loop.index
      mod 5)}">
      <td width="220px"> </td>
      </c:forEach>
      </c:if>
    </tr>
  </c:if>
  </c:forEach>
  </table>
  </td>
</tr>
</c:when>
<c:otherwise>
<tr>
    <td class="font-italic"
    colspan="2">No Images Yet</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td>
    <c:if test="${privilege ge 2}">
      <span style="color:#0000FF">
      <a href="javascript:void(1)"
      onclick="checkAll('true')">Check
      All</a> |
      <a href="javascript:void(1)"
      onclick="checkAll('false')">Unche
      ck All</a> 
      </span>
      <span class="font-italic">With
      Selected : </span>  
      <img src="./images/add2.png"
      width="15px" height="15px"
      title="Include in View"
      style="position:relative;
      left:5px; top:2px" />
      <html:submit property="command"
      value="Include in View"
      styleClass="button-small" />
      <img src="./images/download4.png"
      width="14px" height="14px"
      title="Download Images"
      style="position:relative;
      left:5px; top:2px" />
      <html:submit property="command"
      value="Download"
      styleClass="button-small" />
      <c:if test="${privilege ge 3}">
        <img src="./images/delete.png"
        width="15px" height="15px"
        title="Delete"
        style="position:relative;
        left:5px; top:2px" />
        <html:submit property="command"
        value="Delete" styleId="image"
        styleClass="button-small
        confirm" />
      </c:if>
    </c:if>
    </td>
  </tr>
</table>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>    
<c:choose>
  <c:when test="${privilege ge 2}" >
    <c:if test="${privilege ge 3}" >
      <html:submit property="command"
      value="Add Images"
      styleClass="button-long-blue" />
    </c:if>
    <c:if test="${privilege eq 4}" >
      <html:submit property="command"
      value="View Members"
      styleClass="button-long-blue" />
    </c:if>
    <html:submit property="command"
    value="My Saved View"
    styleClass="button-long-blue" />
  </c:when>
  <c:otherwise>
    <html:submit property="command"
    value="Send Membership Request"
```

```
          styleClass="button-superlong-blue"
          />
        </c:otherwise>
      </c:choose>
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

viewImageSets.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>

 <script>
 function loadPage(projectSet, id){
  if(id == 'editImageSet')
    window.open
    ('EditImageSet.do?projectSet='+projectSet
    , '_self', false);
  else if(id == 'addImages')
    window.open
    ('AddImage.do?projectSet='+projectSet,
    '_self', false);
 }
 </script>
</head>
<body>
 <html:form action="/ViewImageSets"
 method="POST" >
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
<td>
<c:if test="${(userLogin ne null) and
((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content">
 <tr>
   <td class="breadcrumb">
     <html:link
     page="/Home.do"><bean:message
     key="button.virho.home"
     /></html:link> &#187;
     <span class="breadcrumb-
     selected">Virho Image Sets</span>
   </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
 <tr>
   <td class="content-title-form"
   colspan="2">List of Image Sets</td>
 </tr>
 <tr>
   <td>
   <table class="content-desc-form">
     <tr>
       <td>
       <table class="table-content">
         <tr>
           <th>Title</th>
           <th>Description</th>
         </tr>
       <c:choose>
         <c:when
         test="${fn:length(imageSets) gt
         0}">
         <c:forEach var="detail"
         items="${imageSets}">
         <tr>
           <td width="250px">
             <html:link
             page="/ViewImageSetDetails.do?
             projectSet=${detail.projectSet
             Name}">
             <span class="font-
             blue">${detail.projectSetName}
             </span>
             </html:link><br/>
             <c:if test="${detail.privilege
             ge 3}" >
               <html:submit
               property="command"
               value="Add Images"
               styleClass="button-remove"
               onclick="loadPage('${detail.
               projectSetName}',
               'addImages'); return false;"
               />
               <html:submit
               property="command"
               value="Edit Image Set"
               styleClass="button-remove"
               onclick="loadPage('${detail.
               projectSetName}',
               'editImageSet'); return
               false;" />
               <c:if
               test="${detail.privilege eq
               4}" >
                 <c:set var="privilege"
                 value="${detail.privilege}"
                 />
               </c:if>
             </c:if>
           </td>
```

```
        <td>${detail.description}</td>
       </tr>
       <c:if test="${detail.privilege ge
       2}">
        <c:set var="hasPrivilege"
        value="true"/>
       </c:if>
       </c:forEach>
       </c:when>
       <c:otherwise>
       <tr>
        <td class="font-italic"
        colspan="2">No Image Sets
        Yet</td>
       </tr>
       </c:otherwise>
      </c:choose>
      </table>
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
  <table class="table-content">
    <tr>
     <th colspan="5">More Recent
     Images</th>
    </tr>
  <c:choose>
    <c:when
    test="${fn:length(recentImages) gt
    0}">
    <c:forEach var="image"
    items="${recentImages}"
    varStatus="loop">
    <c:if test="${loop.index mod 5 eq
    0}">
    <tr valign="top">
    </c:if>
     <td align="center" width="220px"
     valign="middle">
    <c:choose>
     <c:when test="${(hasPrivilege eq
     true) or (image.accessibility eq
     'Public')}">
      <html:link
      page="/ViewImage.do?projectSet=${
      image.imageSet}&image=${image.ima
      gePathDB}">
     <c:choose>
      <c:when
      test="${fn:endsWith(image.imagePa
      thDB, 'exe') or
      fn:endsWith(image.imagePathDB,
      'avi') or
      fn:endsWith(image.imagePathDB,
      'mp4') ||
      fn:endsWith(image.imagePathDB,
      'wmv') or
      fn:endsWith(image.imagePathDB,
      'jar')}">
       <img
       src="images/defaultThumb.gif"
       class="thumbnail" />
      </c:when>
      <c:when
      test="${fn:endsWith(image.imagePa
      thDB, 'flv')}">
       <object
       classid="clsid:d27cdb6e-ae6d-
       11cf-96b8-444553540000"
       codebase="http://download.macro
       media.com/pub/shockwave/cabs/fl
       ash/swflash.cab#version=9,0,0,0
       " width="100" height="75"
       id="VideoPlayer"
       align="middle">
        <param
        name="allowScriptAccess"
        value="*" />
        <param name="allowFullScreen"
        value="true" />
        <param name="movie"
        value="http://www.platipus.nl/
        flvplayer/download/1.0/FLVPlay
        er.swf?video=http://www.bioinf
        o.mpg.de/virholex/repository/v
        irhoimages/${image.imagePathDB
        }&autoplay=false" />
        <param name"quality"
        value="high" />
        <param name="bgcolor"
        value="#ffffff" />
        <embed
        src="http://www.platipus.nl/fl
        vplayer/download/1.0/FLVPlayer
        .swf?video=http://www.bioinfo.
        mpg.de/virholex/repository/vir
        hoimages/${image.imagePathDB}&
        autoplay=false" quality="high"
        bgcolor="#000000" width="100"
        height="75" name="VideoPlayer"
        align="middle"
        allowScriptAccess="*"
        allowFullScreen="true"
        type="application/x-shockwave-
        flash"
        pluginspage="http://www.macrom
        edia.com/go/getflashplayer" />
       </object>
     </c:when>
     <c:otherwise>
      <img
      src="repository/virhoimages/${i
      mage.imagePathDB}"
      alt="${image.imagePath}"
      title="${image.imagePath}"
      class="thumbnail" />
     </c:otherwise>
   </c:choose>
    </html:link>
   </c:when>
   <c:otherwise>
    <img src="images/restricted.png"
    class="thumbnail" alt="Restricted
    Image" title="Restricted Image"/>
   </c:otherwise>
 </c:choose>
  </td>
 <c:if test="${(loop.index mod 5 eq
 4) or (fn:length(recentImages) eq
 loop.index+1)}">
  <c:if test="${loop.index mod 5 ne
  4}">
  <c:forEach var="counter" begin="1"
  end="${4-(loop.index mod 5)}">
  <td width="220px"> </td>
  </c:forEach>
  </c:if>
 </tr>
 </c:if>
 </c:forEach>
 </c:when>
 <c:otherwise>
 <tr>
  <td class="font-italic"
  colspan="2">No Images Yet</td>
```

```
          </tr>
          </c:otherwise>
        </c:choose>
          <tr>
            <td class="note" colspan="5">
              <c:if test="${hasPrivilege ne
              true}" >
              Note: You need to be a Restricted
              User, Image Contributor or Image
              Coordinator to gain access to the
              restricted images
              </c:if>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>    
          <c:if test="${(privilege eq 4) or
          (isLead eq true)}" >
            <html:submit property="command"
            value="Add Image Set"
            styleClass="button-long-blue" />
          </c:if>
          <html:submit property="command"
          value="My Saved Views"
          styleClass="button-long-blue" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

viewSavedViewDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
 <script src="./files/jquery-
 1.8.0.js"></script>
 <script
 src="http://ajax.googleapis.com/ajax/libs/jq
 uery/1.3.2/jquery.min.js"
 type="text/javascript"></script>
 <script
 src="http://ajax.googleapis.com/ajax/libs/jq
 ueryui/1.7.2/jquery-ui.min.js"></script>
 <link rel="stylesheet" href="./files/jquery-
 ui.css" type="text/css" />
 <script src="./files/jquery.easy-confirm-
 dialog.js"></script>
 <script>
 function checkAll(value) {
  if(value == 'true')
    $('form input:checkbox').attr('checked',
    'checked');
  else
    $('form
    input:checkbox').removeAttr('checked');
 }

 $(document).ready(function() {
  $(".confirm").easyconfirm();
    $("#image").click(function() {
    return true;
  });
  $("#view").click(function() {
    return true;
  });
 });

 </script>
</head>
<body>
 <html:form action="/SavedViewDetails"
 method="POST" styleId="form">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin
    eq 'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <html:link
          page="/ViewImageSets.do">Virho Image
          Sets</html:link> &#187;
          <html:link
          page="/ViewSavedViews.do">My Saved
          Views</html:link> &#187;
          <span class="breadcrumb-
          selected">${savedView}</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
```

```
  <td class="content-title-form"
  colspan="2"><c:out
  value="${savedView}" /></td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
<c:choose>
  <c:when
  test="${fn:length(SavedViewForm.images
  DB) gt 0}">
  <c:forEach var="imageDB"
  items="${SavedViewForm.imagesDB}"
  varStatus="loop">
    <c:if test="${loop.index mod 5 eq
    0}">
    <tr valign="top">
    </c:if>
    <c:set var="image"
    value="${SavedViewForm.images}" />
      <td align="center" width="220px">
        <html:link
        page="/ViewSavedViewImage.do?imag
        e=${imageDB}">
      <c:choose>
        <c:when
        test="${fn:endsWith(imageDB,
        'exe') or fn:endsWith(imageDB,
        'avi') or fn:endsWith(imageDB,
        'mp4') || fn:endsWith(imageDB,
        'wmv') or fn:endsWith(imageDB,
        'jar')}">
          <img
          src="images/defaultThumb.gif"
          class="thumbnail" />
        </c:when>
        <c:when
        test="${fn:endsWith(imageDB,
        'flv')}">
          <object
          classid="clsid:d27cdb6e-ae6d-
          11cf-96b8-444553540000"
          codebase="http://download.macro
          media.com/pub/shockwave/cabs/fl
          ash/swflash.cab#version=9,0,0,0
          " width="100" height="75"
          id="VideoPlayer"
          align="middle">
            <param
            name="allowScriptAccess"
            value="*" />
            <param name="allowFullScreen"
            value="true" />
            <param name="movie"
            value="http://www.platipus.nl/
            flvplayer/download/1.0/FLVPlay
            er.swf?video=http://www.bioinf
            o.mpg.de/virholex/repository/v
            irhoimages/${imageDB}&autoplay
            =false" />
            <param name"quality"
            value="high" />
            <param name="bgcolor"
            value="#ffffff" />
            <embed
            src="http://www.platipus.nl/fl
            vplayer/download/1.0/FLVPlayer
            .swf?video=http://www.bioinfo.
            mpg.de/virholex/repository/vir
            hoimages/${imageDB}&autoplay=f
            alse" quality="high"
            bgcolor="#000000" width="100"
            height="75" name="VideoPlayer"
            align="middle"
            allowScriptAccess="*"
            allowFullScreen="true"
            type="application/x-shockwave-
            flash"
            pluginspage="http://www.macrom
            edia.com/go/getflashplayer" />
          </object>
        </c:when>
        <c:otherwise>
          <img
          src="repository/virhoimages/${i
          mageDB}" class="thumbnail" />
        </c:otherwise>
      </c:choose>
        </html:link>
        <div class="font-supersmall"
        style="word-wrap: break-word;
        width: 120px;">
        <input type="checkbox" name="img-
        ${imageDB}" value="true" /><br/>
        <input type="hidden"
        name="imagesList"
        value="${imageDB}" />
        <html:link
        page="/ViewSavedViewImage.do?imag
        e=${imageDB}">
        <span class="font-image-
        link">${image[loop.index]}</span>
        </html:link>
        </div>
      </td>
    <c:if test="${(loop.index mod 5 eq
    4) or
    (fn:length(SavedViewForm.imagesDB)
    eq loop.index+1)}">
      <c:if test="${loop.index mod 5 ne
      4}">
      <c:forEach var="counter" begin="1"
      end="${4-(loop.index mod 5)}">
      <td width="220px"> </td>
      </c:forEach>
      </c:if>
    </tr>
    </c:if>
  </c:forEach>
  </c:when>
  <c:otherwise>
    <tr>
      <td class="font-italic"
      colspan="5">No Saved View Images
      Yet</td>
    </tr>
    </c:otherwise>
  </c:choose>
    <tr>
      <td colspan="5">
        <span style="color:#0000FF">
        <a href="javascript:void(1)"
        onclick="checkAll('true')">Check
        All</a> |
        <a href="javascript:void(1)"
        onclick="checkAll('false')">Unche
        ck All</a> 
        </span>
        <span class="font-italic">With
        Selected : </span>  
        <img src="./images/download4.png"
        width="14px" height="14px"
        title="Download Images"
        style="position:relative;
        left:5px; top:2px" />
        <html:submit property="command"
        value="Download"
        styleClass="button-small" />
        <img src="./images/delete.png"
        width="15px" height="15px"
        title="Delete"
```

```
            style="position:relative;
            left:5px; top:2px" />
            <html:submit property="command"
            value="Delete" styleId="image"
            styleClass="button-small confirm"
            />
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>    
        <html:submit property="command"
        value="Edit View"
        styleClass="button-long-blue" />
        <html:submit property="command"
        value="Delete View" styleId="view"
        styleClass="button-long-blue
        confirm" />
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

viewSavedImage.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  $(document).ready(function() {
    $(".confirm").easyconfirm();
      $("#delete").click(function() {
      return true;
    });
  });

  </script>
</head>
<body>
  <html:form action="/ViewImage" method="POST"
  styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewImageSets.do">Virho Image
            Sets</html:link> &#187;
            <html:link
            page="/ViewSavedViews.do">My Saved
            Views</html:link> &#187;
            <html:link
            page="/SavedViewDetails.do?savedView
            =${savedView}">${savedView}</html:li
            nk> &#187;
            <span class="breadcrumb-
            selected">${requestScope.image}</spa
            n>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2"><c:out
          value="${savedView}" /> Images</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td style="color:#0000FF">
                <html:link
                page="/SavedViewDetails.do?savedV
                iew=${savedView}">Thumbnail
                View</html:link>
```

```
  </td>
  <td align="right">
    <c:forEach var="image"
    items="${SavedViewForm.imagesDB}"
    varStatus="loop">
    <c:if
    test="${requestScope.imageDB eq
    image}">
      <c:set var="imgPos"
      value="${loop.index}" />
    </c:if>
    </c:forEach>
    Image <c:out value="${imgPos+1}"
    /> of <c:out
    value="${fn:length(SavedViewForm.
    imagesDB)}" />
  </td>
</tr>
<tr>
  <td class="image-container"
  colspan="2" align="center">
  <table>
    <tr>
      <td width="100px">
        <c:set var="prevIndex"
        value="${imgPos-1}" />
        <c:if test="${prevIndex eq -
        1}"><c:set var="prevIndex"
        value="${fn:length(SavedViewFo
        rm.imagesDB)-1}"/></c:if>
        <html:link
        page="/ViewSavedViewImage.do?i
        mage=${SavedViewForm.imagesDB[
        prevIndex]}">
        <img
        src="images/arrowprev.png"
        style="cursor: pointer;" />
        </html:link>
      </td>
      <td>
    <c:choose>
      <c:when
      test="${fn:endsWith(requestScop
      e.imageDB, 'exe') or
      fn:endsWith(requestScope.imageD
      B, 'avi') or
      fn:endsWith(requestScope.imageD
      B, 'mp4') ||
      fn:endsWith(requestScope.imageD
      B, 'wmv') or
      fn:endsWith(requestScope.imageD
      B, 'jar')}">
        <img
        src="images/defaultThumb.gif"
        class="imageView" />
      </c:when>
      <c:when
      test="${fn:endsWith(requestScop
      e.imageDB, 'flv')}">
        <object
        classid="clsid:d27cdb6e-ae6d-
        11cf-96b8-444553540000"
        codebase="http://download.macr
        omedia.com/pub/shockwave/cabs/
        flash/swflash.cab#version=9,0,
        0,0" width="360" height="270"
        id="VideoPlayer"
        align="middle">
          <param
          name="allowScriptAccess"
          value="*" />
          <param
          name="allowFullScreen"
          value="true" />
            <param name="movie"
            value="http://www.platipus.
              nl/flvplayer/download/1.0/F
              LVPlayer.swf?video=http://w
              ww.bioinfo.mpg.de/virholex/
              repository/virhoimages/${re
              questScope.imageDB}&autopla
              y=false" />
            <param name"quality"
            value="high" />
            <param name="bgcolor"
            value="#ffffff" />
            <embed
            src="http://www.platipus.nl/
            flvplayer/download/1.0/FLVPl
            ayer.swf?video=http://www.bi
            oinfo.mpg.de/virholex/reposi
            tory/virhoimages/${requestSc
            ope.imageDB}&autoplay=false"
            quality="high"
            bgcolor="#000000"
            width="360" height="270"
            name="VideoPlayer"
            align="middle"
            allowScriptAccess="*"
            allowFullScreen="true"
            type="application/x-
            shockwave-flash"
            pluginspage="http://www.macr
            omedia.com/go/getflashplayer
            " />
          </object>
        </c:when>
        <c:when
        test="${fn:endsWith(requestScop
        e.imageDB, 'svg')}">
          <img
          src="repository/virhoimages/${
          requestScope.imageDB}"
          height="400px" width="400px"
          />
        </c:when>
        <c:otherwise>
          <img
          src="repository/virhoimages/${
          requestScope.imageDB}"
          class="imageView" />
        </c:otherwise>
      </c:choose>
        </td>
        <td width="100px"
        align="right">
          <c:set var="nextIndex"
          value="${imgPos+1}" />
          <c:if test="${nextIndex eq
          fn:length(SavedViewForm.images
          DB)}"><c:set var="nextIndex"
          value="0"/></c:if>
          <html:link
          page="/ViewSavedViewImage.do?i
          mage=${SavedViewForm.imagesDB[
          nextIndex]}">
          <img
          src="images/arrownext.png"
          style="cursor: pointer;" />
          </html:link>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td class="font-supersmall font-
    gray" colspan="2">
      Image Filename <c:out
      value="${requestScope.image}" />
    </td>
  </tr>
```

```
        </table>
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

viewSavedViews.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
 <script>
 function loadPage(view, id){
  if(id == 'editViewName')
    window.open
    ('EditSavedView.do?savedView='+view,
    '_self', false);
 }
 </script>
</head>
<body>
 <html:form action="/ViewSavedViews"
 method="POST" >
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin
    eq 'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <html:link
          page="/ViewImageSets.do">Virho Image
          Sets</html:link> &#187;
          <span class="breadcrumb-selected">My
          Saved Views</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2">My Saved Views</td>
      </tr>
      <tr>
        <td>
        <table class="content-desc-form">
          <tr>
            <td>
            <table class="table-content">
              <tr>
                <th>Title</th>
                <th>Date Modified</th>
              </tr>
            <c:choose>
              <c:when
              test="${fn:length(savedViews) gt
              0}">
              <c:forEach var="detail"
              items="${savedViews}">
              <tr>
                <td width="550px">
                  <html:link
                  page="/SavedViewDetails.do?sav
                  edView=${detail.viewName}">
                  <span class="font-
                  blue">${detail.viewName}</span
                  >
                  </html:link>
                  <html:submit
                  property="command" value="Edit
                  Saved View"
                  styleClass="button-remove"
                  onclick="loadPage('${detail.vi
                  ewName}', 'editViewName');
                  return false;" />
                </td>
                <td>${detail.dateModified}</td>
              </tr>
              </c:forEach>
              </c:when>
              <c:otherwise>
              <tr>
                <td class="font-italic"
                colspan="2">No Saved Views
                Yet</td>
              </tr>
              </c:otherwise>
            </c:choose>
            </table>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
```

```
    <tr>
      <td> </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/VirhoImagesUsers"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin eq
    'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home" /></html:link>
          &#187;
          <html:link
          page="/ViewImageSets.do">Virho Image
          Sets</html:link> &#187;
          <html:link
          page="/ViewImageSetDetails.do?projectS
          et=${projectSet}">${projectSet}</html:
          link> &#187;
          <span class="breadcrumb-selected">View
          Members</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2">List of Image Set
        Members</td>
      </tr>
      <tr>
        <td>
        <table class="content-desc-form">
          <tr>
            <td>
            <table class="letterList">
              <tr>
                <td>
                <c:forEach var="letter"
                items="A,B,C,D,E,F,G,H,I,J,K,L,M,
                N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
                varStatus="loop">
                <c:choose>
                  <c:when
                  test="${ViewUsersForm.letters[loo
                  p.index] eq letter}">
                    <html:link
                    page="/VirhoImagesUsers.do?list
                    =${letter}"><span class="font-
                    letter">${letter}</span></html:
                    link> 
                  </c:when>
                  <c:otherwise>
                    ${letter} 
                  </c:otherwise>
                </c:choose>
                  </c:forEach>
                  <html:link
                  page="/VirhoImagesUsers.do?list="
                  >View All</html:link>
                  </td>
                </tr>
              </table>
              </td>
            </tr>
            <tr>
              <td>
              <table class="table-content">
                <tr>
                  <th>Name</th>
                  <th>Username</th>
                  <th>Affiliation</th>
                  <th>Role</th>
                </tr>
              <c:choose>
                <c:when
                test="${fn:length(ViewUsersForm.use
                rs) gt 0}">
                <c:forEach var="user"
                items="${ViewUsersForm.users}">
                <tr>
                  <td>
                <c:choose>
                  <c:when
                  test="${requestScope.privilege eq
                  4}" >
                    <html:link
                    page="/UserProfileDetails.do?us
                    er=${user.username}" >
                    <span class="font-
                    blue">${user.firstName} ${
                    user.lastName}</span>
                    </html:link>
                  </c:when>
                  <c:otherwise>
```

```
          ${user.firstName} ${user.l
          astName}
        </c:otherwise>
      </c:choose>
        </td>
        <td>${user.username}</td>
        <td>${user.affiliation}</td>
        <td>
        <c:choose>
          <c:when
          test="${(requestScope.privilege
          eq 4) and (user.username ne
          userLogin)}">
          <select name="role">
            <option value="Remove
            Privilege">Remove
            Privilege</option>
          <c:forEach var="roleName"
          items="${requestScope.roles}">
            <option value="${roleName}"
            <c:if test="${roleName eq
            user.projectRole}">selected="s
            elected"</c:if>>${roleName}</o
            ption>
          </c:forEach>
          </select>
          <input type="hidden"
          name="oldRole"
          value="${user.projectRole}" />
          <input type="hidden"
          name="userName"
          value="${user.username}" />
          </c:when>
          <c:otherwise>
            ${user.projectRole}
          </c:otherwise>
        </c:choose>
          </td>
        </tr>
        </c:forEach>
        </c:when>
        <c:otherwise>
        <tr>
          <td class="font-italic"
          colspan="4">
            There are no members yet
          </td>
        </tr>
        </c:otherwise>
      </c:choose>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
 <tr>
  <td>
       
   <c:if test="${requestScope.privilege
   eq 4}" >
    <html:submit property="command"
    value="Change Privilege Role"
    styleClass="button-superlong-blue"
    />
   </c:if>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
</table>
</c:if>
```

```
    </td>
   </tr>
   <tr>
    <td> </td>
   </tr>
   <tr class="footer">
    <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
    </td>
   </tr>
  </table>
  </html:form>
</body>
</html>
```

VIRHO MODELS

addModelDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script type="text/javascript">
  function displayReferences(){
   window.open("DisplayReferences.do","Referen
   ces","alwaysRaised=yes, height=700,
   width=800 location=no, scrollbars=yes");
  }

  function clearReference(){
   document.getElementById("refTitle").value =
   "";
   document.getElementById("refId").value =
   "";
  }
  </script>
</head>
<body>
  <html:form action="/AddModelDetails"
  method="POST" enctype="multipart/form-data">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
   <tr>
    <td> </td>
   </tr>
   <tr valign="top">
    <td class="sidebar">
     <jsp:include
     page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
   <c:if test="${(userLogin ne null) and
   ((statusLogin eq 'activated') or
```

```
(statusLogin eq 'admin') or (statusLogin eq
'superadmin'))}">
  <c:if test="${(classification ne null)
  and (project ne null)}">
  <table class="content">
    <tr>
      <td class="breadcrumb">
        <html:link
        page="/Home.do"><bean:message
        key="button.virho.home"
        /></html:link> &#187;
        <html:link
        page="/ViewModelsTaxonomy.do">Models
        Taxonomy</html:link> &#187;
        <html:link
        page="/ViewModels.do?classification=
        ${classification}"><c:out
        value="${classification}"/></html:li
        nk> &#187;
        <html:link
        page="/ViewProjectDetails.do?project
        =${project}"><c:out
        value="${project}"/></html:link>
        &#187;
        <span class="breadcrumb-
        selected">Add Model</span>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td class="content-title-form"
      colspan="2">Add Model</td>
    </tr>
    <tr>
      <td>
      <table class="content-desc-form">
        <tr>
          <td class="note" colspan="2">
            Note: All required fields are
            marked with an asterisk (*). The
            model description should be a pdf
            file.
          </td>
        </tr>
        <tr>
          <td colspan="2"><span class="error-
          text-big"><html:errors
          property="errorMessage"
          /></span></td>
        </tr>
        <tr>
          <td class="font-bold"
          width="210px">Project Name *</td>
          <td><c:out value="${project}"/>
          <html:hidden property="projectName"
          value="${project}"/></td>
        </tr>
        <tr>
          <td class="font-bold">Model Name
          *</td>
          <td>
            <html:text property="modelName"
            size="50" style="remove-error-
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR" />
            <span class="error-
            text"><html:errors
            property="modelName" /></span>
          </td>
        </tr>
        <tr>
          <td class="font-
          bold">Classification *</td>
          <td>
          <html:select
          property="classification">
            <html:option value="Conceptual
            Diagrammatic">Conceptual
            Diagrammatic Model</html:option>
            <html:option value="Conceptual
            Qualitative Discrete">Conceptual
            Qualitative Discrete
            Model</html:option>
            <html:option value="Conceptual
            Qualitative
            Continuous">Conceptual
            Qualitative Continuous
            Model</html:option>
            <html:option value="Conceptual
            Quantitative
            Stochastic">Conceptual
            Quantitative Stochastic
            Model</html:option>
            <html:option value="Conceptual
            Quantitative
            Deterministic">Conceptual
            Quantitative Deterministic
            Model</html:option>
            <html:option value="Contextual
            Diagrammatic">Contextual
            Diagrammatic Model</html:option>
            <html:option value="Contextual
            Qualitative Discrete">Contextual
            Qualitative Discrete
            Model</html:option>
            <html:option value="Contextual
            Qualitative
            Continuous">Contextual
            Qualitative Continuous
            Model</html:option>
            <html:option value="Contextual
            Quantitative
            Stochastic">Contextual
            Quantitative Stochastic
            Model</html:option>
            <html:option value="Contextual
            Quantitative
            Deterministic">Contextual
            Quantitative Deterministic
            Model</html:option>
          </html:select>
          </td>
        </tr>
        <tr>
          <td class="font-bold">Description
          *</td>
          <td>
            <html:file property="file"
            style="remove-error-field-file"
            errorStyleClass="error-field"
            errorKey="org.apache.struts.actio
            n.ERROR" />
            <span class="error-
            text"><html:errors
            property="file" /></span>
          </td>
        </tr>
        <tr>
          <td class="font-bold">Scale</td>
          <td><html:text property="scale"
          size="50" /></td>
        </tr>
        <tr>
          <td class="font-bold">Scope</td>
          <td><html:text property="scope"
          size="50" /></td>
        </tr>
```

```
    <tr>
      <td class="font-bold">Size</td>
      <td><html:text property="size"
      size="50" /></td>
    </tr>
    <tr>
      <td class="font-bold">Type</td>
      <td><html:text property="type"
      size="50" /></td>
    </tr>
    <tr>
      <td class="font-bold">Biological
      Aspects</td>
      <td><html:text property="aspects"
      size="50" /></td>
    </tr>
    <tr>
      <td class="font-bold">Tools</td>
      <td><html:text property="tools"
      size="50" /></td>
    </tr>
    <tr>
      <td class="font-bold">Computational
      Properties</td>
      <td><html:text
      property="properties" size="50"
      /></td>
    </tr>
    <tr>
      <td class="font-bold">Experimental
      Evidence</td>
      <td>
        <html:text property="evidence"
        size="50" readonly="true"
        styleId="refTitle" />
        <html:hidden
        property="referenceId" value="0"
        styleId="refId" /> 
        <a href="javascript:void(1)"
        onclick="displayReferences()">
        <img src="./images/reference.png"
        height="16px" width="16px"
        title="Choose Reference"/>
        </a>
         
        <a href="javascript:void(1)"
        onclick="clearReference()">
        <img src="./images/delete2.png"
        height="15px" width="15px"
        title="Remove Reference"/>
        </a>
      </td>
    </tr>
    <tr>
      <td class="font-bold">External
      Link</td>
      <td><html:text property="link"
      size="50" /></td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button"
    />
    <html:reset property="command"
    value="Reset" styleClass="button" />
    <html:cancel property="command"
    value="Cancel" styleClass="button"
    />
      </td>
    </tr>
  </table>
  </c:if>
  </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

editModelDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script type="text/javascript">
  function displayReferences(){
    window.open("DisplayReferences.do","Referen
    ces","alwaysRaised=yes, height=700,
    width=800 location=no, scrollbars=yes");
  }

  function clearReference(){
    document.getElementById("refTitle").value =
    "";
    document.getElementById("refId").value =
    "";
  }
  </script>
</head>
<body>
  <html:form action="/EditModelDetails"
  method="POST" enctype="multipart/form-data">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
```

```
<c:if test="${(userLogin ne null) and
((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin eq
'superadmin'))}">
 <c:if test="${(classification ne null)
 and (project ne null)}">
 <table class="content">
  <tr>
   <td class="breadcrumb">
    <html:link
    page="/Home.do"><bean:message
    key="button.virho.home"
    /></html:link> &#187;
    <html:link
    page="/ViewModelsTaxonomy.do">Models
    Taxonomy</html:link> &#187;
    <html:link
    page="/ViewModels.do?classification=
    ${classification}"><c:out
    value="${classification}"/></html:li
    nk> &#187;
    <html:link
    page="/ViewProjectDetails.do?project
    =${project}"><c:out
    value="${project}"/></html:link>
    &#187;
    <html:link
    page="/ViewModelDetails.do?project=$
    {project}&model=${model}"><c:out
    value="${model}"/></html:link>
    &#187;
    <span class="breadcrumb-
    selected">Edit Model</span>
   </td>
  </tr>
  <tr>
   <td> </td>
  </tr>
  <tr>
   <td class="content-title-form"
   colspan="2">Edit Model</td>
  </tr>
  <tr>
   <td>
   <table class="content-desc-form">
    <tr>
     <td class="note" colspan="2">
      Note: All required fields are
      marked with an asterisk (*). The
      model description should be a pdf
      file. If no file is selected, the
      old file would be used instead.
     </td>
    </tr>
    <tr>
     <td colspan="2"><span class="error-
     text-big"><html:errors
     property="errorMessage"
     /></span></td>
    </tr>
    <tr>
     <td class="font-bold"
     width="210px">Project Name *</td>
     <td><c:out value="${project}"/>
     <html:hidden property="projectName"
     value="${project}"/></td>
    </tr>
    <tr>
     <td class="font-bold">Model Name
     *</td>
     <td>
      <html:text property="modelName"
      size="50" style="remove-error-
      field" errorStyleClass="error-
      field"
      errorKey="org.apache.struts.actio
      n.ERROR" />
      <span class="error-
      text"><html:errors
      property="modelName" /></span>
      <html:hidden property="modelId"
      value="${ModelDetailsForm.modelId
      }" />
     </td>
    </tr>
    <tr>
     <td class="font-
     bold">Classification *</td>
     <td>
     <html:select
     property="classification">
      <html:option value="Conceptual
      Diagrammatic">Conceptual
      Diagrammatic Model</html:option>
      <html:option value="Conceptual
      Qualitative Discrete">Conceptual
      Qualitative Discrete
      Model</html:option>
      <html:option value="Conceptual
      Qualitative
      Continuous">Conceptual
      Qualitative Continuous
      Model</html:option>
      <html:option value="Conceptual
      Quantitative
      Stochastic">Conceptual
      Quantitative Stochastic
      Model</html:option>
      <html:option value="Conceptual
      Quantitative
      Deterministic">Conceptual
      Quantitative Deterministic
      Model</html:option>
      <html:option value="Contextual
      Diagrammatic">Contextual
      Diagrammatic Model</html:option>
      <html:option value="Contextual
      Qualitative Discrete">Contextual
      Qualitative Discrete
      Model</html:option>
      <html:option value="Contextual
      Qualitative
      Continuous">Contextual
      Qualitative Continuous
      Model</html:option>
      <html:option value="Contextual
      Quantitative
      Stochastic">Contextual
      Quantitative Stochastic
      Model</html:option>
      <html:option value="Contextual
      Quantitative
      Deterministic">Contextual
      Quantitative Deterministic
      Model</html:option>
     </html:select>
     </td>
    </tr>
    <tr>
     <td class="font-bold"
     valign="top">Description *</td>
     <td>
      <html:file property="file"
      style="remove-error-field-file"
      errorStyleClass="error-field"
      errorKey="org.apache.struts.actio
      n.ERROR" />
      <span class="error-
      text"><html:errors
      property="file" /></span>
      <br/><br/>
```

```
    OLD FILE :
    <html:link
    page="/repository/virhomodels/${M
    odelDetailsForm.fileDescriptionDB
    }" target="_blank" >
    <bean:write
    name="ModelDetailsForm"
    property="fileDescription"
    /></html:link>
    <html:hidden
    property="fileDescriptionDB"
    value="${ModelDetailsForm.fileDes
    criptionDB}" />
    <html:hidden
    property="fileDescription"
    value="${ModelDetailsForm.fileDes
    cription}" />
  </td>
</tr>
<tr>
  <td class="font-bold">Scale</td>
  <td><html:text property="scale"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Scope</td>
  <td><html:text property="scope"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Size</td>
  <td><html:text property="size"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Type</td>
  <td><html:text property="type"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Biological
  Aspects</td>
  <td><html:text property="aspects"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Tools</td>
  <td><html:text property="tools"
  size="50" /></td>
</tr>
<tr>
  <td class="font-bold">Computational
  Properties</td>
  <td><html:text
  property="properties" size="50"
  /></td>
</tr>
<tr>
  <td class="font-bold">Experimental
  Evidence</td>
  <td>
    <html:text property="evidence"
    size="50" readonly="true"
    styleId="refTitle" />
    <html:hidden
    property="referenceId" value="0"
    styleId="refId" /> 
    <a href="javascript:void(1)"
    onclick="displayReferences()">
    <img src="./images/reference.png"
    height="16px" width="16px"
    title="Choose Reference"/>
    </a>
     
    <a href="javascript:void(1)"
    onclick="clearReference()">
      <img src="./images/delete2.png"
      height="15px" width="15px"
      title="Remove Reference"/>
      </a>
    </td>
  </tr>
  <tr>
    <td class="font-bold">External
    Link</td>
    <td><html:text property="link"
    size="50" /></td>
  </tr>
 </table>
 </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <html:submit property="command"
    value="Submit" styleClass="button"
    />
    <html:reset property="command"
    value="Reset" styleClass="button" />
    <html:cancel property="command"
    value="Cancel" styleClass="button"
    />
  </td>
 </tr>
</table>
</c:if>
</c:if>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

editUserPrivilege.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script>
```

```
  function removeConfirmation() {
    var message = confirm("Are you sure you
    want to remove the privilege of this
    user?");
    if (message==true){
      document.getElementById("removePrivilege"
      ).val("Remove Privilege");
      document.forms[0].action =
      "EditUserPrivilege.do";
      document.forms[0].submit();
    }
  }
  </script>
</head>
<body>
  <html:form action="/EditUserPrivilege"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewModelsTaxonomy.do">Models
            Taxonomy</html:link> &#187;
            <html:link
            page="/ViewModels.do?classification=
            ${classification}"><c:out
            value="${classification}"/></html:li
            nk> &#187;
            <html:link
            page="/ViewProjectDetails.do?project
            =${project}"><c:out
            value="${project}" /></html:link>
            &#187;
            <html:link
            page="/ViewProjectUsers.do">Project
            Users</html:link> &#187;
            <span class="breadcrumb-
            selected">Change User
            Privilege</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Change User Privilege</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form"
          style="height:210px">
            <tr>
              <td class="note" colspan="2">
                You can change the privilege of
                the user in this project.
              </td>
            </tr>
          <c:if test="${fn:length(userDetails)
          gt 0}">
            <tr>
              <td class="font-bold"
              width=200px">Name</td>
              <td>${userDetails[0]} ${userDe
              tails[1]}</td>
            </tr>
            <tr>
              <td class="font-bold">Username</td>
              <td>
                ${userDetails[2]}
                <html:hidden property="username"
                value="${userDetails[2]}"/>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Email
              Address</td>
              <td>${userDetails[3]}</td>
            </tr>
            <tr>
              <td class="font-bold">Privilege
              Role</td>
              <td>
              <html:select property="privilege">
                <html:option value="Model
                Contributor">Model
                Contributor</html:option>
                <html:option value="Model
                Coordinator">Model
                Coordinator</html:option>
              </html:select>
              </td>
            </tr>
          </c:if>
          </table>
          <br/>
              
          <html:submit property="command"
          value="Update Privilege"
          styleClass="button-superlong" />
          <html:submit property="command"
          value="Remove Privilege"
          styleClass="button-superlong"
          styleId="removePrivilege"
          onclick="removeConfirmation()" />
          <html:submit property="command"
          value="Cancel" styleClass="button-
          long" />
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

viewModelDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
```

```
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  function changeAction(id) {
    if(id == "privilegeRequest") {
      document.forms[0].action =
      "PrivilegeRequest.do";
    } else if(id == "editModel") {
      document.forms[0].action =
      "EditModelDetails.do";
    } else if(id == "addModel") {
      document.forms[0].action =
      "AddModelDetails.do";
    }
  }

  $(document).ready(function() {
    $(".confirm").easyconfirm();
    $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/ViewModelDetails"
  styleId="form" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin eq
    'superadmin'))}">
      <c:if test="${(classification ne null)
      and (project ne null)}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewModelsTaxonomy.do">Models
            Taxonomy</html:link> &#187;
            <html:link
            page="/ViewModels.do?classification=
            ${classification}"><c:out
            value="${classification}"/></html:li
            nk> &#187;
            <html:link
            page="/ViewProjectDetails.do?project
            =${project}"><c:out
            value="${project}"/></html:link>
            &#187;
            <span class="breadcrumb-
            selected"><c:out
            value="${model}"/></span>
            </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Model Details</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td class="font-bold"
              width="210px">Model Name</td>
              <td><c:out
              value="${ModelDetailsForm.modelNa
              me}"/></td>
            </tr>
            <tr>
              <td class="font-bold">Author</td>
              <td><c:out
              value="${ModelDetailsForm.author}
              "/></td>
            </tr>
            <tr>
              <td class="font-
              bold">Description</td>
              <td><html:link
              page="/repository/virhomodels/${M
              odelDetailsForm.fileDescriptionDB
              }" target="_blank" ><c:out
              value="${ModelDetailsForm.fileDes
              cription}"/></html:link></td>
            </tr>
            <tr>
              <td class="font-bold">Scale</td>
              <td>
                <c:out
                value="${ModelDetailsForm.scale
                }"/>
                <c:if test="${empty
                ModelDetailsForm.scale}">-
                </c:if>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Scope</td>
              <td>
                <c:out
                value="${ModelDetailsForm.scope
                }"/>
                <c:if test="${empty
                ModelDetailsForm.scope}">-
                </c:if>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Size</td>
              <td>
```

```
    <c:out
    value="${ModelDetailsForm.size}
    "/>
    <c:if test="${empty
    ModelDetailsForm.size}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-bold">Type</td>
  <td>
    <c:out
    value="${ModelDetailsForm.type}
    "/>
    <c:if test="${empty
    ModelDetailsForm.type}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-bold">Biological
  Aspects</td>
  <td>
    <c:out
    value="${ModelDetailsForm.aspec
    ts}"/>
    <c:if test="${empty
    ModelDetailsForm.aspects}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-bold">Tools</td>
  <td>
    <c:out
    value="${ModelDetailsForm.tools
    }"/>
    <c:if test="${empty
    ModelDetailsForm.tools}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">Computational
  Properties</td>
  <td>
    <c:out
    value="${ModelDetailsForm.prope
    rties}"/>
    <c:if test="${empty
    ModelDetailsForm.properties}">-
    </c:if>
  </td>
</tr>
<tr>
  <td class="font-
  bold">Experimental Evidence</td>
  <td>
    <html:link
    page="/ViewReferenceDetails.do?
    module=VirhoReferences&collecti
    on=${reference.collection}&refe
    rence=${reference.title}&refId=
    ${reference.referenceId}" >
    <c:out
    value="${ModelDetailsForm.evide
    nce}"/>
    </html:link>
    <c:if test="${empty
    ModelDetailsForm.evidence}">-
    </c:if>
  </td>
</tr>
<tr>
      <td class="font-bold">External
      Link</td>
      <td>
        <div style="word-wrap: break-
        word; width: 540px;">
        <a
        href="///${ModelDetailsForm.lin
        k}" target="_blank"><c:out
        value="${ModelDetailsForm.link}
        "/></a>
        <c:if test="${empty
        ModelDetailsForm.link}">-
        </c:if>
        </div>
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
        
    <input type="hidden"
    name="statusAction" value=""
    id="statusAction" />
    <c:set var="role"
    value="${ModelDetailsForm.privilege
    }" />
  <c:choose>
    <c:when test="${role ge 3}">
    <html:submit property="command"
    value="Add Model"
    styleClass="button-long-blue"
    onclick="changeAction('addModel')"
    />
      <html:submit property="command"
      value="Edit Model"
      styleClass="button-long-blue"
      onclick="changeAction('editModel'
      )" />
      <c:if test="${role eq 4}">
        <html:submit property="command"
        value="Delete Model"
        styleId="delete"
        styleClass="button-long-blue
        confirm" />
      </c:if>
    </c:when>
    <c:otherwise>
      <html:submit property="command"
      value="Send Membership Request"
      styleClass="button-superlong-
      blue"
      onclick="changeAction('privilegeR
      equest')" />
    </c:otherwise>
  </c:choose>
  </td>
 </tr>
</table>
</c:if>
</c:if>
 </td>
</tr>
<tr>
 <td> </td>
</tr>
<tr class="footer">
 <td colspan="2">
  &copy; <bean:message
  key="virho.copyright" />
 </td>
</tr>
```

```
  </table>
  </html:form>
</body>
</html>
```

viewModels.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin eq
    'superadmin'))}">
      <c:if test="${classification ne null}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewModelsTaxonomy.do">Models
            Taxonomy</html:link> &#187;
            <span class="breadcrumb-
            selected"><c:out
            value="${classification}"/></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2"><c:out
          value="${classification}" />
          Model</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
            <td>
            <table class="table-content">
              <tr>
                <th>Project</th>
                <th>Model</th>
              </tr>
            <c:choose>
              <c:when
              test="${fn:length(projects) gt
              0}">
              <c:forEach var="project"
              items="${projects}"
              varStatus="loop">
              <tr>
                <td valign="top" <c:if
                test="${fn:length(project.model
                s) ge
                2}">rowspan="${fn:length(projec
                t.models)}"</c:if>>
                  <html:link
                  page="/ViewProjectDetails.do?p
                  roject=${project.projectName}"
                  >
                  <span class="font-
                  blue">${project.projectName}</
                  span>
                  </html:link>
                </td>
              <c:choose>
                <c:when
                test="${fn:length(project.model
                s) gt 0}">
                <c:forEach var="model"
                items="${project.models}"
                varStatus="counter">
                <c:if test="${counter.index ge
                1}">
              <tr>
                </c:if>
                <td>
                  <html:link
                  page="/ViewModelDetails.do?pro
                  ject=${project.projectName}&mo
                  del=${model.modelName}">
                  <span class="font-
                  blue">${model.modelName}</span
                  ></html:link>
                </td>
              </tr>
                </c:forEach>
                </c:when>
                <c:otherwise>
                  <td class="font-italic">No
                  Models Yet</td>
              </tr>
                </c:otherwise>
              </c:choose>
              </c:forEach>
              </c:when>
              <c:otherwise>
              <tr>
                <td class="font-italic"
                colspan="2">No Projects
                Yet</td>
              </tr>
              </c:otherwise>
            </c:choose>
            </table>
            </td>
          </tr>
        </table>
        </td>
      </tr>
    </table>
    </c:if>
  </c:if>
```

```
    </td>
   </tr>
   <tr>
    <td> </td>
   </tr>
   <tr class="footer">
    <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
    </td>
   </tr>
  </table>
</body>
</html>
```

viewProjectDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/ViewProjectDetails"
 method="POST">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
  <c:if test="${(userLogin ne null) and
  ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or (statusLogin eq
  'superadmin'))}">
   <c:if test="${(classification ne null)
   and (project ne null)}">
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewModelsTaxonomy.do">Models
      Taxonomy</html:link> &#187;
      <html:link
      page="/ViewModels.do?classification=
      ${classification}"><c:out
      value="${classification}"/></html:li
      nk> &#187;
      <span class="breadcrumb-
      selected"><c:out
      value="${project}"/></span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2">Project Details</td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr>
       <td class="font-bold"
       width="200px">Project Name</td>
       <td><bean:write
       name="ProjectDetailsForm"
       property="projectName"/></td>
      </tr>
      <tr>
       <td class="font-
       bold">Description</td>
       <td><bean:write
       name="ProjectDetailsForm"
       property="description"/></td>
      </tr>
      <tr>
       <td class="font-bold">Author</td>
       <td><bean:write
       name="ProjectDetailsForm"
       property="author"/></td>
      </tr>
      <tr>
       <td class="font-bold">Status</td>
       <td><bean:write
       name="ProjectDetailsForm"
       property="status"/></td>
      </tr>
      <tr>
       <td class="font-bold">Models</td>
      <c:choose>
       <c:when
       test="${fn:length(ProjectDetailsFor
       m.models) gt 0}">
       <c:forEach var="model"
       items="${ProjectDetailsForm.models}
       " varStatus="loop">
       <c:if test="${loop.index ge 1}">
      <tr>
       <td> </td>
       </c:if>
       <td>
        <html:link
        page="/ViewModelDetails.do?projec
        t=${ProjectDetailsForm.projectNam
        e}&model=${model.modelName}">${mo
        del.modelName}</html:link>
       </td>
      </tr>
       </c:forEach>
       </c:when>
       <c:otherwise>
       <td><span class="font-italic">No
       Models Yet</span></td>
      </tr>
       </c:otherwise>
      </c:choose>
     </table>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
```

```
      <tr>
        <td>
            
        <c:set var="role"
        value="${ProjectDetailsForm.privilege}
        " />
      <c:choose>
        <c:when test="${role ge 3}">
        <html:submit property="command"
        value="Add Model" styleClass="button-
        long-blue" />
        <c:if test="${role eq 4}">
          <html:submit property="command"
          value="View Members"
          styleClass="button-long-blue" />
        </c:if>
        </c:when>
        <c:otherwise>
          <html:submit property="command"
          value="Send Membership Request"
          styleClass="button-superlong-blue"
          />
        </c:otherwise>
      </c:choose>
        </td>
      </tr>
    </table>
    </c:if>
  </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

viewProjectUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@page trimDirectiveWhitespaces="true" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
   <tr>
    <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <c:if test="${(userLogin ne null) and
   ((statusLogin eq 'activated') or
   (statusLogin eq 'admin') or (statusLogin
   eq 'superadmin'))}">
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewModelsTaxonomy.do">Models
      Taxonomy</html:link> &#187;
      <html:link
      page="/ViewModels.do?classification=
      ${classification}"><c:out
      value="${classification}"/></html:li
      nk> &#187;
      <html:link
      page="/ViewProjectDetails.do?project
      =${project}"><c:out
      value="${project}" /></html:link>
      &#187;
      <span class="breadcrumb-
      selected">Project Users</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2">List of Project Users</td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr>
       <td>
       <table class="table-content">
        <tr>
         <th>Name</th>
         <th>Username</th>
         <th>Email Address</th>
         <th>Rolename</th>
         <th>Action</th>
        </tr>
        <c:if test="${fn:length(users) gt
        0}">
        <c:forEach var="user"
        items="${users}">
        <tr>
         <td>${user.firstName} ${us
         er.lastName}</td>
         <td>${user.username}</td>
         <td>${user.emailAddress}</td>
         <td>${user.projectRole}</td>
         <td align="center">
         <c:if test="${user.username ne
         userLogin}">
          <html:link
          page="/EditUserPrivilege.do?us
          ername=${user.username}">
          <input type="image"
          src="./images/edit.gif"
          width="15px" height="15px"
          title="Edit Privilege"/>
```

```
              </html:link>
             </c:if>
             <td>
            </tr>
            </c:forEach>
            </c:if>
           </table>
           </td>
          </tr>
        </table>
        </td>
       </tr>
     </table>
     </c:if>
     </td>
   </tr>
   <tr>
     <td> </td>
   </tr>
   <tr class="footer">
     <td colspan="2">
       &copy; <bean:message
       key="virho.copyright" />
     </td>
   </tr>
  </table>
</body>
</html>
```

viewTaxonomy.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Models Taxonomy</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Models Taxonomy</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =cd"><span class="font-
              blue">Conceptual Diagrammatic
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =cqd"><span class="font-
              blue">Conceptual Qualitative
              Discrete
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =cqc"><span class="font-
              blue">Conceptual Qualitative
              Continuous
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =cqns"><span class="font-
              blue">Conceptual Quantitative
              Stochastic
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =cqnd"><span class="font-
              blue">Conceptual Quantitative
              Deterministic
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =ctd"><span class="font-
              blue">Contextual Diagrammatic
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =ctqd"><span class="font-
              blue">Contextual Qualitative
              Discrete
              Model</span></html:link></td>
            </tr>
            <tr>
              <td><html:link
              page="/ViewModels.do?classification
              =ctqc"><span class="font-
              blue">Contextual Qualitative
              Continuous
              Model</span></html:link></td>
            </tr>
```

```
          <tr>
            <td><html:link
            page="/ViewModels.do?classification
            =ctqns"><span class="font-
            blue">Contextual Quantitative
            Stochastic
            Model</span></html:link></td>
          </tr>
          <tr>
            <td><html:link
            page="/ViewModels.do?classification
            =ctqnd"><span class="font-
            blue">Contextual Quantitative
            Deterministic
            Model</span></html:link></td>
          </tr>
        </table>
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/VirhoModelsUsers"
 method="POST">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
</td>
<td>
<c:if test="${(userLogin ne null) and
((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content">
 <tr>
  <td class="breadcrumb">
   <html:link
   page="/Home.do"><bean:message
   key="button.virho.home"
   /></html:link> &#187;
   <html:link
   page="/ViewModelsTaxonomy.do">Models
   Taxonomy</html:link> &#187;
   <html:link
   page="/ViewModels.do?classification=
   ${classification}"><c:out
   value="${classification}"/></html:li
   nk> &#187;
   <html:link
   page="/ViewProjectDetails.do?project
   =${project}"><c:out
   value="${project}"/></html:link>
   &#187;
   <span class="breadcrumb-
   selected">View Members</span>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
 <tr>
  <td class="content-title-form"
  colspan="2">List of Project
  Members</td>
 </tr>
 <tr>
  <td>
  <table class="content-desc-form">
   <tr>
    <td>
    <table class="letterList">
     <tr>
      <td>
       <c:forEach var="letter"
       items="A,B,C,D,E,F,G,H,I,J,K,L
       ,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
       varStatus="loop">
      <c:choose>
       <c:when
       test="${ViewUsersForm.letters[
       loop.index] eq letter}">
        <html:link
        page="/VirhoModelsUsers.do?l
        ist=${letter}"><span
        class="font-
        letter">${letter}</span></ht
        ml:link> 
       </c:when>
       <c:otherwise>
        ${letter} 
       </c:otherwise>
      </c:choose>
       </c:forEach>
       <html:link
       page="/VirhoModelsUsers.do?lis
       t=">View All</html:link>
      </td>
     </tr>
    </table>
    </td>
   </tr>
   <tr>
    <td>
```

```
<table class="table-content">
  <tr>
    <th>Name</th>
    <th>Username</th>
    <th>Affiliation</th>
    <th>Role</th>
  </tr>
<c:choose>
  <c:when
  test="${fn:length(ViewUsersForm.u
  sers) gt 0}">
  <c:forEach var="user"
  items="${ViewUsersForm.users}">
  <tr>
    <td>
  <c:choose>
    <c:when
    test="${requestScope.privilege
    eq 4}" >
      <html:link
      page="/UserProfileDetails.do?u
      ser=${user.username}" >
      <span class="font-
      blue">${user.firstName} $
      {user.lastName}</span>
      </html:link>
    </c:when>
    <c:otherwise>
    ${user.firstName} ${user.l
    astName}
    </c:otherwise>
  </c:choose>
    </td>
    <td>${user.username}</td>
    <td>${user.affiliation}</td>
    <td>
  <c:choose>
    <c:when
    test="${(requestScope.privilege
    eq 4) and (user.username ne
    userLogin)}">
    <select name="role">
      <option value="Remove
      Privilege">Remove
      Privilege</option>
    <c:forEach var="roleName"
    items="${requestScope.roles}">
      <option value="${roleName}"
      <c:if test="${roleName eq
      user.projectRole}">selected="s
      elected"</c:if>>${roleName}</o
      ption>
    </c:forEach>
    </select>
    <input type="hidden"
    name="oldRole"
    value="${user.projectRole}" />
    <input type="hidden"
    name="userName"
    value="${user.username}" />
    </c:when>
    <c:otherwise>
      ${user.projectRole}
    </c:otherwise>
  </c:choose>
    </td>
  </tr>
  </c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-italic"
    colspan="4">
      There are no members yet
    </td>
  </tr>
        </c:otherwise>
      </c:choose>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
 <tr>
  <td>
        
    <c:if test="${requestScope.privilege
    eq 4}" >
      <html:submit property="command"
      value="Change Privilege Role"
      styleClass="button-superlong-blue"
      />
    </c:if>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

VIRHO REFERENCES

addCollection.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/AddCollection"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
```

```
  <td> </td>
</tr>
<tr valign="top">
  <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
  </td>
  <td>
  <c:if test="${(userLogin ne null) and
  ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or (statusLogin
  eq 'superadmin'))}">
  <table class="content">
    <tr>
      <td class="breadcrumb">
        <html:link
        page="/Home.do"><bean:message
        key="button.virho.home"
        /></html:link> &#187;
        <html:link
        page="/ViewCollections.do">Virho
        References Collection</html:link>
        &#187;
        <span class="breadcrumb-
        selected">Add Collection</span>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td class="content-title-form"
      colspan="2">Add Collection</td>
    </tr>
    <tr>
      <td>
      <table class="content-desc-form">
        <tr>
          <td class="note" colspan="2">
            Note: All fields marked with
            asterisk (*) are required
          </td>
        </tr>
        <tr>
          <td colspan="2"><span class="error-
          text-big"><html:errors
          property="errorMessage"
          /></span></td>
        </tr>
        <tr>
          <td class="font-bold"
          width="140px">Collection Name *
          </td>
          <td>
            <html:text
            property="collectionName"
            size="75" style="remove-error-
            field" errorStyleClass="error-
            field"
            errorKey="org.apache.struts.actio
            n.ERROR" />
            <span class="error-
            text"><html:errors
            property="collectionName"
            /></span>
          </td>
        </tr>
        <tr>
          <td class="font-bold"
          width="130px">Downloadable * </td>
          <td>
            <html:radio
            property="downloadable"
            value="Yes"/>Yes   
            ;
            <html:radio
            property="downloadable"
            value="No"/>No
          </td>
        </tr>
        <tr>
          <td class="font-bold"
          colspan="2">Description</td>
        </tr>
        <tr>
          <td colspan="2">
            <html:textarea
            property="description" cols="90"
            rows="8"></html:textarea>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>
            
        <html:submit property="command"
        value="Submit" styleClass="button"
        />
        <html:reset property="command"
        value="Reset" styleClass="button" />
        <html:cancel property="command"
        value="Cancel" styleClass="button"
        />
      </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

addReference.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/fmt"
prefix="fmt" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
```

```
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script type="text/javascript"
  src="./files/javascript.js"></script>
</head>
<body onload="changeDetails()">
  <html:form action="/AddReference"
  method="POST" enctype="multipart/form-data">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewCollections.do">Virho
            References Collection</html:link>
            &#187;
            <html:link
            page="/ViewReferences.do?collection=
            ${collection}">${collection}</html:l
            ink> &#187;
            <span class="breadcrumb-
            selected">Add Reference</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Reference Details</td>
        </tr>
        <tr>
          <td>
          <c:set var="type"
          value="${ReferenceDetailsForm.type}"/>
          <table class="content-desc-form"
          id="table-content">
            <tr>
              <td class="note" colspan="2">
                Note: All fields marked with
                asterisk (*) are required. The
                reference file should be in PDF.
              </td>
            </tr>
            <tr>
              <td colspan="2"><span class="error-
              text-big"><html:errors
              property="errorMessage"
              /></span></td>
            </tr>
            <tr>
              <td class="font-bold"
              width="150px">Title * </td>
              <td>
                <html:text property="title"
                size="75" style="remove-error-
                field" errorStyleClass="error-
                field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="title" /></span>
              </td>
            </tr>
            <tr>
              <td class="font-bold">Type * </td>
              <td>
              <html:select property="type"
              styleId="type" style="remove-error-
              field" errorStyleClass="error-
              field"
              errorKey="org.apache.struts.action.
              ERROR" onchange="changeDetails()">
                <html:option value="Default"
                styleClass="font-gray">Select
                Reference Type</html:option>
                <html:option
                value="Book">Book</html:option>
                <html:option
                value="Inbook">Inbook</html:optio
                n>
                <html:option
                value="Incollection">Incollection
                </html:option>
                <html:option
                value="Inproceedings">Inproceedin
                gs</html:option>
                <html:option
                value="Article">Article</html:opt
                ion>
                <html:option value="Masters
                Thesis">Master's
                Thesis</html:option>
                <html:option value="PhD
                Thesis">PhD Thesis</html:option>
                <html:option value="Technical
                Report">Technical
                Report</html:option>
                <html:option
                value="Unpublished">Unpublished</
                html:option>
                <html:option
                value="Manual">Manual</html:optio
                n>
                <html:option
                value="Proceedings">Proceedings</
                html:option>
                <html:option
                value="Miscellaneous">Miscellaneo
                us</html:option>
              </html:select>
              </td>
            </tr>
            <tr>
              <td class="font-bold">File</td>
              <td>
                <html:file property="fileName"
                style="remove-error-field-file"
                errorStyleClass="error-field"
                errorKey="org.apache.struts.actio
                n.ERROR" />
                <span class="error-
                text"><html:errors
                property="fileName" /></span>
              </td>
            </tr>
            <tr id="author" style="display:
            none">
              <td class="font-bold">Author <span
              id="authorRequired">*</span> </td>
              <td>
                <html:text property="author"
                size="50" styleId="authorVal"
```

```
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="editor" style="display:
none">
  <td class="font-bold">Editor</td>
  <td>
    <html:text property="editor"
    size="50" styleId="editorVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="publisher" style="display:
none">
  <td class="font-bold">Publisher
  <span
  id="publisherRequired">*</span></td
  >
  <td>
    <html:text property="publisher"
    size="50" styleId="publisherVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold">Year <span
  id="yearRequired" style="display:
  none">*</span> </td>
  <td>
  <jsp:useBean id="date"
  class="java.util.Date" />
  <fmt:formatDate var="currentYear"
  value="${date}" pattern="yyyy" />
  <html:select property="year"
  style="remove-error-field"
  errorStyleClass="error-field"
  errorKey="org.apache.struts.action.
  ERROR">
    <html:option value="0000"
    styleClass="font-gray">Select
    Year</html:option>
  <c:forEach var="number"
  begin="1800" end="${currentYear}" >
    <html:option
    value="${currentYear-(number-
    1800)}"><c:out
    value="${currentYear-(number-
    1800)}" /></html:option>
  </c:forEach>
  </html:select>
  </td>
</tr>
<tr>
  <td class="font-bold">Month</td>
  <td>
  <html:select property="month">
    <html:option value="Default"
    styleClass="font-gray">Select
    Month</html:option>
    <html:option
    value="January">January</html:opt
    ion>
    <html:option
    value="February">February</html:o
    ption>
    <html:option
    value="March">March</html:option>
    <html:option
    value="April">April</html:option>
    <html:option
    value="May">May</html:option>
    <html:option
    value="June">June</html:option>
    <html:option
    value="July">July</html:option>
    <html:option
    value="August">August</html:optio
    n>
    <html:option
    value="September">September</html
    :option>
    <html:option
    value="October">October</html:opt
    ion>
    <html:option
    value="November">November</html:o
    ption>
    <html:option
    value="December">December</html:o
    ption>
  </html:select>
  </td>
</tr>
<tr id="booktitle" style="display:
none">
  <td class="font-bold">Book Title
  <span
  id="booktitleRequired">*</span></td
  >
  <td>
    <html:text property="booktitle"
    size="75" styleId="booktitleVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="chapter" style="display:
none">
  <td class="font-bold">Chapter <span
  id="chapterRequired">*</span></td>
  <td>
    <html:text property="chapter"
    size="30" styleId="chapterVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="journal" style="display:
none">
  <td class="font-bold">Journal <span
  id="journalRequired">*</span></td>
  <td>
    <html:text property="journal"
    size="75" styleId="journalVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="volume" style="display:
none">
  <td class="font-bold">Volume</td>
  <td>
    <html:text property="volume"
    size="30" styleId="volumeVal"
    style="remove-error-field"
    errorStyleClass="error-field"
```

```
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="series" style="display:
none">
  <td class="font-bold">Series</td>
  <td>
    <html:text property="series"
    size="30" styleId="seriesVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="edition" style="display:
none">
  <td class="font-bold">Edition</td>
  <td>
    <html:text property="edition"
    size="30" styleId="editionVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="number" style="display:
none">
  <td class="font-bold">Number</td>
  <td>
    <html:text property="number"
    size="30" styleId="numberVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="pages" style="display:
none">
  <td class="font-bold">Pages <span
  id="pagesRequired">*</span></td>
  <td>
    <html:text property="pages"
    size="30" styleId="pagesVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="school" style="display:
none">
  <td class="font-bold">School <span
  id="schoolRequired">*</span></td>
  <td>
    <html:text property="school"
    size="75" styleId="schoolVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="institution" style="display:
none">
  <td class="font-bold">Institution
  <span
  id="institutionRequired">*</span></
  td>
  <td>
    <html:text property="institution"
    size="75"
    styleId="institutionVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="organization"
style="display: none">
  <td class="font-
  bold">Organization</td>
  <td>
    <html:text
    property="organization" size="75"
    styleId="organizationVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="address" style="display:
none">
  <td class="font-bold"
  valign="top">Address</td>
  <td>
    <html:textarea property="address"
    cols="70" rows="2"
    styleId="addressVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold"
  valign="top">Note <span
  id="noteRequired" style="display:
  none">*</span> </td>
  <td>
    <html:textarea property="note"
    cols="70" rows="4" style="remove-
    error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="subtype" style="display:
none">
  <td class="font-bold">Subtype</td>
  <td>
    <html:text property="subtype"
    size="75" styleId="subtypeVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="howpublished"
style="display: none">
  <td class="font-bold">How
  Published</td>
  <td>
    <html:text
    property="howpublished" size="75"
    styleId="howpublishedVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold">Link</td>
```

```
            <td>
              <html:text property="link"
              size="75" style="remove-error-
              field" errorStyleClass="error-
              field"
              errorKey="org.apache.struts.actio
              n.ERROR" />
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>    
          <html:submit property="command"
          value="Submit" styleClass="button"
          />
          <html:reset property="command"
          value="Reset" styleClass="button" />
          <html:cancel property="command"
          value="Cancel" styleClass="button"
          />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

displayReferences.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>

 <script type="text/javascript">
 function filter (id){
  var collection =
  document.getElementById("collection").value
  .toLowerCase();
  var reftitle =
  document.getElementById("reference").value.
  toLowerCase();
  var author =
  document.getElementById("author").value.toL
  owerCase();
  var type =
  document.getElementById("type").value.toLow
  erCase();

  var table = document.getElementById(id);
  var collectioncell;
  var lnamecell;
  var reftitlecell;
  var authorcell;
  var typecell;
  var fieldName;

  for (var r = 2; r < table.rows.length;
  r++){

    collectioncell =
    table.rows[r].cells[0].innerHTML.replace(
    /<[^>]+>/g,"");
    reftitlecell =
    table.rows[r].cells[1].innerHTML.replace(
    /<[^>]+>/g,"");
    authorcell =
    table.rows[r].cells[2].innerHTML.replace(
    /<[^>]+>/g,"");
    typecell =
    table.rows[r].cells[3].innerHTML.replace(
    /<[^>]+>/g,"");

    if
    (collectioncell.toLowerCase().indexOf(col
    lection)>=0 &&
    reftitlecell.toLowerCase().indexOf(reftit
    le)>=0 &&
    authorcell.toLowerCase().indexOf(author)>
    =0 &&
    typecell.toLowerCase().indexOf(type)>=0)
      table.rows[r].style.display = '';
    else
      table.rows[r].style.display = 'none';
  }
 }

 function returnSelection(refid, reftitle){
  window.opener.document.getElementById("refT
  itle").value = reftitle;
  window.opener.document.getElementById("refI
  d").value = refid;
  self.close();
 }

 </script>
</head>
<body>
 <html:form action="/DisplayReferences"
 method="POST" >
 <table class="content">
  <tr>
    <td class="content-title-form"
    colspan="2">List of References</td>
  </tr>
  <tr>
    <td>
    <table class="content-desc-form">
      <tr>
        <td>
        <table id="references" class="table-
        content">
          <tr>
            <th>Collection</th>
            <th>Reference Title</th>
```

```
            <th>Author</th>
            <th>Type</th>
          </tr>
          <tr bgcolor="#E1E8F1"
          align="center">
            <td><input type="text"
            name="collectionText"
            id="collection" size="20px"
            onkeyup="filter('references')"/></t
            d>
            <td><input type="text"
            name="referenceText" id="reference"
            size="20px"
            onkeyup="filter('references')"/></t
            d>
            <td><input type="text"
            name="authorText" id="author"
            size="20px"
            onkeyup="filter('references')"/></t
            d>
            <td><input type="text"
            name="typeText" id="type"
            size="20px"
            onkeyup="filter('references')"/></t
            d>
          </tr>
        <c:choose>
          <c:when
          test="${fn:length(requestScope.refer
          ences) gt 0}">
          <c:forEach var="detail"
          items="${requestScope.references}">
          <tr class="record">
            <td
            onclick="returnSelection('${detail.
            referenceId}',
            '${detail.title}')">${detail.collec
            tion}</td>
            <td
            onclick="returnSelection('${detail.
            referenceId}',
            '${detail.title}')">${detail.title}
            </td>
            <td
            onclick="returnSelection('${detail.
            referenceId}',
            '${detail.title}')">${detail.author
            }</td>
            <td
            onclick="returnSelection('${detail.
            referenceId}',
            '${detail.title}')">${detail.type}<
            /td>
          </tr>
          </c:forEach>
          </c:when>
          <c:otherwise>
          <tr>
            <td class="font-italic"
            colspan="4">No References Yet</td>
          </tr>
          </c:otherwise>
        </c:choose>
        </table>
        </td>
      </tr>
    </table>
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

editCollection.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/EditCollection"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewCollections.do">Virho
            References Collection</html:link>
            &#187;
            <html:link
            page="/ViewReferences.do?collection=
            ${collection}">${collection}</html:l
            ink> &#187;
            <span class="breadcrumb-
            selected">Edit Collection</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">Edit Collection</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td class="note" colspan="2">
                Note: All fields marked with
                asterisk (*) are required
              </td>
            </tr>
            <tr>
```

```
        <td colspan="2"><span class="error-
        text-big"><html:errors
        property="errorMessage"
        /></span></td>
      </tr>
      <tr>
        <td class="font-bold"
        width="140px">Collection Name *
        </td>
        <td>
          <html:text
          property="collectionName"
          size="75" style="remove-error-
          field" errorStyleClass="error-
          field"
          errorKey="org.apache.struts.actio
          n.ERROR" />
          <span class="error-
          text"><html:errors
          property="collectionName"
          /></span>
        </td>
      </tr>
      <tr>
        <td class="font-bold"
        width="130px">Downloadable * </td>
        <td>
          <html:radio
          property="downloadable"
          value="Yes"/>Yes   
          ;
          <html:radio
          property="downloadable"
          value="No"/>No
        </td>
      </tr>
      <tr>
        <td class="font-bold"
        colspan="2">Description</td>
      </tr>
      <tr>
        <td colspan="2">
          <html:textarea
          property="description" cols="90"
          rows="8"></html:textarea>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td align="left">
          
      <html:submit property="command"
      value="Submit" styleClass="button-
      long" />
      <html:reset property="command"
      value="Reset" styleClass="button-
      long" />
      <html:cancel property="command"
      value="Cancel" styleClass="button-
      long" />
    </td>
  </tr>
 </table>
 </c:if>
 </td>
</tr>
<tr class="footer">
 <td colspan="2">
  &copy; <bean:message
  key="virho.copyright" />
 </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

editReference.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/fmt"
prefix="fmt" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
 <script type="text/javascript"
 src="./files/javascript.js"></script>
</head>
<body onload="changeDetails()">
 <html:form action="/EditReference"
 method="POST" enctype="multipart/form-data">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <c:if test="${(userLogin ne null) and
   ((statusLogin eq 'activated') or
   (statusLogin eq 'admin') or (statusLogin
   eq 'superadmin'))}">
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewCollections.do">Virho
      References Collection</html:link>
      &#187;
      <html:link
      page="/ViewReferences.do?collection=
      ${collection}">${collection}</html:l
      ink> &#187;
      <span class="breadcrumb-
      selected">Edit Reference</span>
     </td>
    </tr>
    <tr>
     <td> </td>
```

```
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Reference Details</td>
</tr>
<tr>
  <td>
  <html:hidden property="referenceId"
  value="${ReferenceDetailsForm.referenc
  eId}" />
  <c:set var="type"
  value="${ReferenceDetailsForm.type}"/>
  <table class="content-desc-form"
  id="table-content">
   <tr>
    <td class="note" colspan="2">
     Note: All fields marked with
     asterisk (*) are required. The
     reference file should be in PDF.
    </td>
   </tr>
   <tr>
    <td colspan="2"><span class="error-
    text-big"><html:errors
    property="errorMessage"
    /></span></td>
   </tr>
   <tr>
    <td class="font-bold"
    width="150px">Title * </td>
    <td>
     <html:text property="title"
     size="75" style="remove-error-
     field" errorStyleClass="error-
     field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
     <span class="error-
     text"><html:errors
     property="title" /></span>
    </td>
   </tr>
   <tr>
    <td class="font-bold">Type * </td>
    <td>
    <html:select property="type"
    styleId="type" style="remove-error-
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.action.
    ERROR" onchange="changeDetails()">
     <html:option value="Default"
     styleClass="font-gray">Select
     Reference Type</html:option>
     <html:option
     value="Book">Book</html:option>
     <html:option
     value="Inbook">Inbook</html:optio
     n>
     <html:option
     value="Incollection">Incollection
     </html:option>
     <html:option
     value="Inproceedings">Inproceedin
     gs</html:option>
     <html:option
     value="Article">Article</html:opt
     ion>
     <html:option value="Masters
     Thesis">Master's
     Thesis</html:option>
     <html:option value="PhD
     Thesis">PhD Thesis</html:option>
     <html:option value="Technical
     Report">Technical
     Report</html:option>
     <html:option
     value="Unpublished">Unpublished</
     html:option>
     <html:option
     value="Manual">Manual</html:optio
     n>
     <html:option
     value="Proceedings">Proceedings</
     html:option>
     <html:option
     value="Miscellaneous">Miscellaneo
     us</html:option>
    </html:select>
    </td>
  </tr>
  <tr>
    <td class="font-bold"
    valign="top">File</td>
    <td>
     <html:file property="fileName"
     style="remove-error-field-file"
     errorStyleClass="error-field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
     <span class="error-
     text"><html:errors
     property="fileName" /></span>
     <c:if test="${not empty
     ReferenceDetailsForm.file}">
      <br/><br/>
      OLD FILE :
      <html:link
      page="/repository/virhoreferenc
      es/${ReferenceDetailsForm.fileD
      B}" target="_blank" >
      <bean:write
      name="ReferenceDetailsForm"
      property="file" /></html:link>
      <html:hidden property="fileDB"
      value="${ReferenceDetailsForm.f
      ileDB}" />
      <html:hidden property="file"
      value="${ReferenceDetailsForm.f
      ile}" />
      <br/><br/>
      <input type="checkbox"
      name="removeFile"> Remove
      reference files
     </c:if>
    </td>
  </tr>
  <tr id="author" style="display:
  none">
    <td class="font-bold">Author <span
    id="authorRequired">*</span> </td>
    <td>
     <html:text property="author"
     size="50" styleId="authorVal"
     style="remove-error-field"
     errorStyleClass="error-field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
    </td>
  </tr>
  <tr id="editor" style="display:
  none">
    <td class="font-bold">Editor</td>
    <td>
     <html:text property="editor"
     size="50" styleId="editorVal"
     style="remove-error-field"
     errorStyleClass="error-field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
    </td>
  </tr>
```

```
<tr id="publisher" style="display:
none">
  <td class="font-bold">Publisher
  <span
  id="publisherRequired">*</span></td
  >
  <td>
    <html:text property="publisher"
    size="50" styleId="publisherVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold">Year <span
  id="yearRequired" style="display:
  none">*</span> </td>
  <td>
  <jsp:useBean id="date"
  class="java.util.Date" />
  <fmt:formatDate var="currentYear"
  value="${date}" pattern="yyyy" />
  <html:select property="year"
  style="remove-error-field"
  errorStyleClass="error-field"
  errorKey="org.apache.struts.action.
  ERROR">
    <html:option value="0000"
    styleClass="font-gray">Select
    Year</html:option>
  <c:forEach var="number"
  begin="1800" end="${currentYear}" >
    <html:option
    value="${currentYear-(number-
    1800)}"><c:out
    value="${currentYear-(number-
    1800)}" /></html:option>
  </c:forEach>
  </html:select>
  </td>
</tr>
<tr>
  <td class="font-bold">Month</td>
  <td>
  <html:select property="month">
    <html:option value="Default"
    styleClass="font-gray">Select
    Month</html:option>
    <html:option
    value="January">January</html:opt
    ion>
    <html:option
    value="February">February</html:o
    ption>
    <html:option
    value="March">March</html:option>
    <html:option
    value="April">April</html:option>
    <html:option
    value="May">May</html:option>
    <html:option
    value="June">June</html:option>
    <html:option
    value="July">July</html:option>
    <html:option
    value="August">August</html:optio
    n>
    <html:option
    value="September">September</html
    :option>
    <html:option
    value="October">October</html:opt
    ion>
    <html:option
    value="November">November</html:o
    ption>
    <html:option
    value="December">December</html:o
    ption>
  </html:select>
  </td>
</tr>
<tr id="booktitle" style="display:
none">
  <td class="font-bold">Book Title
  <span
  id="booktitleRequired">*</span></td
  >
  <td>
    <html:text property="booktitle"
    size="75" styleId="booktitleVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="chapter" style="display:
none">
  <td class="font-bold">Chapter <span
  id="chapterRequired">*</span></td>
  <td>
    <html:text property="chapter"
    size="30" styleId="chapterVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="journal" style="display:
none">
  <td class="font-bold">Journal <span
  id="journalRequired">*</span></td>
  <td>
    <html:text property="journal"
    size="75" styleId="journalVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="volume" style="display:
none">
  <td class="font-bold">Volume</td>
  <td>
    <html:text property="volume"
    size="30" styleId="volumeVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="series" style="display:
none">
  <td class="font-bold">Series</td>
  <td>
    <html:text property="series"
    size="30" styleId="seriesVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="edition" style="display:
none">
```

```
  <td class="font-bold">Edition</td>
  <td>
    <html:text property="edition"
    size="30" styleId="editionVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="number" style="display:
none">
  <td class="font-bold">Number</td>
  <td>
    <html:text property="number"
    size="30" styleId="numberVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="pages" style="display:
none">
  <td class="font-bold">Pages <span
  id="pagesRequired">*</span></td>
  <td>
    <html:text property="pages"
    size="30" styleId="pagesVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="school" style="display:
none">
  <td class="font-bold">School <span
  id="schoolRequired">*</span></td>
  <td>
    <html:text property="school"
    size="75" styleId="schoolVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="institution" style="display:
none">
  <td class="font-bold">Institution
  <span
  id="institutionRequired">*</span></
  td>
  <td>
    <html:text property="institution"
    size="75"
    styleId="institutionVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="organization"
style="display: none">
  <td class="font-
  bold">Organization</td>
  <td>
    <html:text
    property="organization" size="75"
    styleId="organizationVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="address" style="display:
none">
  <td class="font-bold"
  valign="top">Address</td>
  <td>
    <html:textarea property="address"
    cols="70" rows="2"
    styleId="addressVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold"
  valign="top">Note <span
  id="noteRequired" style="display:
  none">*</span> </td>
  <td>
    <html:textarea property="note"
    cols="70" rows="4" style="remove-
    error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="subtype" style="display:
none">
  <td class="font-bold">Subtype</td>
  <td>
    <html:text property="subtype"
    size="75" styleId="subtypeVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr id="howpublished"
style="display: none">
  <td class="font-bold">How
  Published</td>
  <td>
    <html:text
    property="howpublished" size="75"
    styleId="howpublishedVal"
    style="remove-error-field"
    errorStyleClass="error-field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
<tr>
  <td class="font-bold">Link</td>
  <td>
    <html:text property="link"
    size="75" style="remove-error-
    field" errorStyleClass="error-
    field"
    errorKey="org.apache.struts.actio
    n.ERROR" />
  </td>
</tr>
</table>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>    
```

```
          <html:submit property="command"
          value="Submit" styleClass="button-
          long" />
          <html:reset property="command"
          value="Reset" styleClass="button-
          long" />
          <html:cancel property="command"
          value="Cancel" styleClass="button-
          long" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

exportReferences.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Virus-Host Interaction
 Lexicon</title>
 <link rel="shortcut icon"
 href="./images/favicon.ico" >
 <link rel="stylesheet" type="text/css"
 href="./files/custom.css" media="screen"/>
</head>
<body>
 <html:form action="/ExportReferences"
 method="POST">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin
    eq 'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <html:link
          page="/ViewCollections.do">Virho
          References Collection</html:link>
          &#187;
          <html:link
          page="/ViewReferences.do?collection=
          ${collection}">${collection}</html:l
          ink> &#187;
          <span class="breadcrumb-
          selected">Export References</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2">Export References</td>
      </tr>
      <tr>
        <td>
        <table class="content-desc-form">
          <tr>
            <td class="note" colspan="2">Select
            options to export references</td>
          </tr>
          <tr>
            <td width="130px">XML</td>
            <td><input type="radio"
            name="fileType" value="xml"
            checked="checked" /></td>
          </tr>
          <tr>
            <td>BibTex</td>
            <td><input type="radio"
            name="fileType"
            value="bibtex"/></td>
          </tr>
        </table>
        <c:forEach var="reference"
        items="${CollectionsForm.references}">
          <input type="hidden"
          name="myReferences"
          value="ref${reference.title}id${refe
          rence.referenceId}" />
        </c:forEach>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>
              
          <html:submit property="command"
          value="Export" styleClass="button-
          long-blue" />
          <html:cancel property="command"
          value="Cancel" styleClass="button-
          long-blue" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
```

```
   </tr>
  </table>
  </html:form>
</body>
</html>
```

viewCollections.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script>
  function loadPage(collection, id){
    if(id == 'editCollection')
      window.open
      ('EditCollection.do?collection='+collecti
      on, '_self', false);
    else if(id == 'addReference')
      window.open
      ('AddReference.do?collection='+collection
      , '_self', false);
  }
  </script>
</head>
<body>
  <html:form action="/ViewCollections"
  method="POST" >
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Virho References
            Collection</span>
          </td>
        </tr>
        <tr>
          <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">List of References
  Collection</td>
</tr>
<tr>
<td>
<table class="content-desc-form">
  <tr>
    <td>
    <table class="letterList">
      <tr>
        <td>
        <c:forEach var="letter"
        items="A,B,C,D,E,F,G,H,I,J,K,L,M,
        N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
        varStatus="loop">
      <c:choose>
        <c:when
        test="${CollectionsForm.letters[l
        oop.index] eq letter}">
          <html:link
          page="/ViewCollections.do?list=
          ${letter}"><span class="font-
          letter">${letter}</span></html:
          link> 
        </c:when>
        <c:otherwise>
          ${letter} 
        </c:otherwise>
      </c:choose>
        </c:forEach>
        <html:link
        page="/ViewCollections.do?list=">
        View All</html:link>
        </td>
      </tr>
    </table>
    </td>
  </tr>
  <tr>
    <td>
    <html:form action="/ViewCollections"
    method="POST">
    <table class="table-content">
      <tr>
        <th>Title</th>
        <th>Description</th>
      </tr>
    <c:choose>
      <c:when
      test="${fn:length(collections) gt
      0}">
      <c:forEach var="collection"
      items="${collections}">
      <tr>
        <td width="250px">
        <html:link
        page="/ViewReferences.do?collecti
        on=${collection.collectionName}">
        <span class="font-
        blue">${collection.collectionName
        }</span>
        <c:if
        test="${collection.downloadable
        eq 'Yes'}" >
          <input type="image"
          src="./images/download3.png"
          width="10px" height="10px"
          title="Downloadable Reference"
          />
        </c:if>
        </html:link><br/>
```

```
            <c:if
            test="${collection.privilege ge
            3}" >
              <html:submit property="command"
              value="Add Reference"
              styleClass="button-remove"
              onclick="loadPage('${collection
              .collectionName}',
              'addReference'); return false;"
              />
              <html:submit property="command"
              value="Edit Collection"
              styleClass="button-remove"
              onclick="loadPage('${collection
              .collectionName}',
              'editCollection'); return
              false;" />
              <c:if
              test="${collection.privilege eq
              4}" >
                <c:set var="privilege"
                value="${collection.privilege}
                " />
              </c:if>
            </c:if>
            </td>
            <td>${collection.description}</td
            >
          </tr>
          </c:forEach>
          </c:when>
          <c:otherwise>
          <tr>
            <td class="font-italic"
            colspan="2">No References
            Yet</td>
          </tr>
          </c:otherwise>
        </c:choose>
          </table>
          </html:form>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>    
      <c:if test="${(privilege eq 4) or
      (isLead eq true)}" >
        <html:submit property="command"
        value="Add Collection"
        styleClass="button-long-blue" />
      </c:if>
      </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
 <tr class="footer">
   <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
   </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

viewReferenceDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
  <script>
  $(document).ready(function() {
    $(".confirm").easyconfirm();
      $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
<html:form action="/ViewReferenceDetails"
styleId="form" method="POST">
<table class="body" align="center">
<jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr valign="top">
    <td class="sidebar">
      <jsp:include page="/rd_virui/sidebar.jsp"
      />
    </td>
    <td>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or (statusLogin eq
    'superadmin'))}">
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home" /></html:link>
          &#187;
          <html:link
          page="/ViewCollections.do">Virho
          References Collection</html:link>
          &#187;
```

```
    <html:link
    page="/ViewReferences.do?collection=${
    collection}">${collection}</html:link>
    &#187;
    <span class="breadcrumb-
    selected">${reference}</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Reference Details</td>
</tr>
<tr>
  <td>
  <html:hidden property="referenceId"
  value="${ReferenceDetailsForm.reference
  Id}" />
  <c:set var="type"
  value="${ReferenceDetailsForm.type}"/>
  <table class="content-desc-form"
  id="table-content">
    <tr>
      <td class="font-bold"
      width="150px">Title</td>
      <td><bean:write
      name="ReferenceDetailsForm"
      property="title"/></td>
    </tr>
    <c:if test="${type ne 'Proceedings'}"
    >
    <tr>
      <td class="font-bold">Author</td>
      <td>
        <bean:write
        name="ReferenceDetailsForm"
        property="author"/>
        <c:if test="${empty
        ReferenceDetailsForm.author}">-
        </c:if>
      </td>
    </tr>
    </c:if>
    <tr>
      <td class="font-bold">Type</td>
      <td><bean:write
      name="ReferenceDetailsForm"
      property="type"/></td>
    </tr>
    <tr>
      <td class="font-bold">Year</td>
      <td>
        <c:if
        test="${ReferenceDetailsForm.year
        ne 0}" >
          <bean:write
          name="ReferenceDetailsForm"
          property="year"/>
        </c:if>
        <c:if
        test="${ReferenceDetailsForm.year
        eq 0}" >-</c:if>
      </td>
    </tr>
    <tr>
      <td class="font-bold">Month</td>
      <td>
        <bean:write
        name="ReferenceDetailsForm"
        property="month"/>
        <c:if test="${empty
        ReferenceDetailsForm.month}">-
        </c:if>
      </td>
    </tr>
<c:if test="${type eq 'Inproceedings'}"
>
  <tr>
    <td class="font-bold">Book
    Title</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="booktitle"/>
      <c:if test="${empty
      ReferenceDetailsForm.booktitle}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${type eq 'Article'}" >
  <tr>
    <td class="font-bold">Journal</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="journal"/>
      <c:if test="${empty
      ReferenceDetailsForm.journal}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Book') or (type
eq 'Inbook') or (type eq 'Manual')}" >
  <tr>
    <td class="font-bold">Edition</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="edition"/>
      <c:if test="${empty
      ReferenceDetailsForm.edition}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Book') or (type
eq 'Inbooking') or (type eq
'Inproceedings') or (type eq
'Proceedings') or (type eq 'Article')}"
>
  <tr>
    <td class="font-bold">Volume</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="volume"/>
      <c:if test="${empty
      ReferenceDetailsForm.volume}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Book') or (type
eq 'Inbooking') or (type eq
'Inproceedings') or (type eq
'Proceedings')}" >
  <tr>
    <td class="font-bold">Series</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="series"/>
      <c:if test="${empty
      ReferenceDetailsForm.series}">-
      </c:if>
    </td>
  </tr>
</c:if>
```

```
<c:if test="${(type eq 'Article') or
(type eq 'Technical Report')}" >
  <tr>
    <td class="font-bold">Number</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="number"/>
      <c:if test="${empty
      ReferenceDetailsForm.number}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Inbooking')}" >
  <tr>
    <td class="font-bold">Chapter</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="chapter"/>
      <c:if test="${empty
      ReferenceDetailsForm.chapter}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Inbooking') or
(type eq 'Inproceedings') or (type eq
'Article')}" >
  <tr>
    <td class="font-bold">Pages</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="pages"/>
      <c:if test="${empty
      ReferenceDetailsForm.pages}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Inbooking') or
(type eq 'Masters Thesis') or (type eq
'PhD Thesis') or (type eq 'Technical
Report')}" >
  <tr>
    <td class="font-bold">Subtype</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="subtype"/>
      <c:if test="${empty
      ReferenceDetailsForm.subtype}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Masters
Thesis') or (type eq 'PhD Thesis')}" >
  <tr>
    <td class="font-bold">School</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="school"/>
      <c:if test="${empty
      ReferenceDetailsForm.school}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${type eq 'Technical
Report'}" >
  <tr>
    <td class="font-
    bold">Institution</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="institution"/>
      <c:if test="${empty
      ReferenceDetailsForm.institution}">
      -</c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Inproceedings')
or (type eq 'Manual') or (type eq
'Proceedings')}" >
  <tr>
    <td class="font-
    bold">Organization</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="organization"/>
      <c:if test="${empty
      ReferenceDetailsForm.organization}"
      >-</c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type ne 'Article') and
(type ne 'Miscellaneous') and (type ne
'Unpublished')}" >
  <tr>
    <td class="font-bold">Address</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="address"/>
      <c:if test="${empty
      ReferenceDetailsForm.address}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Inproceedings')
or (type eq 'Proceedings')}" >
  <tr>
    <td class="font-bold">Editor</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="editor"/>
      <c:if test="${empty
      ReferenceDetailsForm.editor}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${(type eq 'Book') or (type
eq 'Inbooking') or (type eq
'Inproceedings') or (type eq
'Proceedings')}" >
  <tr>
    <td class="font-bold">Publisher</td>
    <td>
      <bean:write
      name="ReferenceDetailsForm"
      property="publisher"/>
      <c:if test="${empty
      ReferenceDetailsForm.publisher}">-
      </c:if>
    </td>
  </tr>
</c:if>
<c:if test="${type eq 'Miscellaneous'}"
>
  <tr>
```

```
      <td class="font-bold">How
      Published</td>
      <td>
        <bean:write
        name="ReferenceDetailsForm"
        property="howpublished"/>
        <c:if test="${empty
        ReferenceDetailsForm.howpublished}"
        >-</c:if>
      </td>
    </tr>
  </c:if>
    <tr>
      <td class="font-bold">Note</td>
      <td>
        <bean:write
        name="ReferenceDetailsForm"
        property="note"/>
        <c:if test="${empty
        ReferenceDetailsForm.note}">-
        </c:if>
      </td>
    </tr>
    <tr>
      <td class="font-bold">Link</td>
      <td>
        <div style="word-wrap: break-word;
        width: 600px;">
        <a
        href="///${ReferenceDetailsForm.lin
        k}" target="_blank"><bean:write
        name="ReferenceDetailsForm"
        property="link"/></a>
        </div>
        <c:if test="${empty
        ReferenceDetailsForm.link}">-
        </c:if>
      </td>
    </tr>
    <tr>
      <td class="font-bold">File</td>
      <td>
        <html:link
        page="/repository/virhoreferences/$
        {ReferenceDetailsForm.fileDB}"
        target="_blank" >
        <bean:write
        name="ReferenceDetailsForm"
        property="file"/></html:link>
        <c:if test="${empty
        ReferenceDetailsForm.file}">-
        </c:if>
        <html:hidden property="fileDB"
        value="${ReferenceDetailsForm.fileD
        B}" />
        <html:hidden property="file"
        value="${ReferenceDetailsForm.file}
        " />
      </td>
    </tr>
  </table>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>    
  <c:set var="privilege"
  value="${ReferenceDetailsForm.privilege
  }"/>
  <c:if test="${privilege ge 3}" >
    <html:submit property="command"
    value="Add Reference"
    styleClass="button-long-blue" />
    <html:submit property="command"
    value="Edit Reference"
    styleClass="button-long-blue" />
    <c:if test="${privilege eq 4}" >
      <html:submit property="command"
      styleId="delete" value="Delete
      Reference" styleClass="button-long-
      blue confirm" />
    </c:if>
  </c:if>
  <c:if test="${privilege le 1}" >
    <html:submit property="command"
    value="Send Membership Request"
    styleClass="button-superlong-blue" />
  </c:if>
  </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

viewReferences.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>
```

```
  <script type="text/javascript">
  function exportReference(loop) {
    if(hasNoSelected(loop)==true) {
      alert("Where do you want to export the
      selected references?");
      return true;
    } else {
      alert("Please check at least one of the
      references to export.");
      return false;
    }
  }

  function hasNoSelected(loop) {
    var noSelected = false;

    for(var i=0; i<loop+1; i++) {
      if(document.getElementById("export"+i).ch
      ecked == true) {
        noSelected = true;
        break;
      }
      if(i<loop+1) {
        break;
      }
    }

    return noSelected;
  }

  function checkAll(loop) {
    for(var i=0; i<loop+1; i++) {
      document.getElementById("export"+i).check
      ed = true;
    }
  }

  function uncheckAll(loop) {
    for(var i=0; i<loop+1; i++) {
      document.getElementById("export"+i).check
      ed = false;
    }
  }

  function changeAction(id) {
    if(id == "addReference") {
      document.forms[0].action =
      "ViewReferences.do";
    }
  }

  $(document).ready(function() {
    $(".confirm").easyconfirm();
      $("#delete").click(function() {
      return true;
    });
  });
  </script>
</head>
<body>
  <html:form action="/ViewReferences"
  method="POST" styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content">
  <tr>
    <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewCollections.do">Virho
      References Collection</html:link>
      &#187;
      <span class="breadcrumb-
      selected">${collection}</span>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td class="content-title-form"
    colspan="2">List of References</td>
  </tr>
  <tr>
    <td>
    <table class="content-desc-form">
      <tr>
        <td>
        <table class="description-
        container">
          <tr>
            <td>
            <span class="font-
            big"><bean:write
            name="CollectionsForm"
            property="collectionName"/></sp
            an>  
            <c:if
            test="${(CollectionsForm.privil
            ege eq 3) or
            (CollectionsForm.privilege eq
            4)}" >
              <html:submit
              property="command" value="Edit
              Collection"
              styleClass="button-remove"  />
            </c:if>
            <c:if
            test="${CollectionsForm.privile
            ge eq 4}" >
              <html:submit
              property="command"
              value="Delete Collection"
              styleId="delete"
              styleClass="button-remove
              confirm" />
            </c:if>
            <br/>
            <span class="font-darkgray">
            Date added <bean:write
            name="CollectionsForm"
            property="dateAdded"/>
            </span>
            </td>
          </tr>
          <tr>
            <td>
              <bean:write
              name="CollectionsForm"
              property="description"/>
            </td>
          </tr>
        </table>
        </td>
      </tr>
```

```
<tr>
  <td>
  <html:form
  action="/ViewCollections"
  method="POST">
  <table class="table-content">
    <tr>
      <th>Title</th>
      <th>Author</th>
      <th>Year</th>
      <th>Type</th>
    <c:if
    test="${CollectionsForm.privilege
    ge 2 &&
    CollectionsForm.downloadable eq
    'Yes'}" >
      <th>Export</th>
    </c:if>
  </tr>
<c:choose>
  <c:when
  test="${fn:length(CollectionsForm.r
  eferences) gt 0}">
  <c:forEach var="reference"
  items="${CollectionsForm.references
  }" varStatus="loop">
  <tr>
    <td>
      <html:link
      page="/ViewReferenceDetails.do?
      reference=${reference.title}&re
      fId=${reference.referenceId}">
      <span class="font-
      blue">${reference.title}</span>
      </html:link>
    </td>
    <td>${reference.author}</td>
    <td align="center">
      <c:if test="${reference.year ne
      0}">${reference.year}</c:if>
      <c:if test="${reference.year eq
      0}">-</c:if>
    </td>
    <td>${reference.type}</td>
  <c:if
  test="${CollectionsForm.privilege
  ge 2 &&
  CollectionsForm.downloadable eq
  'Yes'}" >
    <td align="center">
      <input type="checkbox"
      name="id${reference.referenceId
      }"
      value="ref${reference.title}id$
      {reference.referenceId}"
      id="export${loop.index}"/>
      <input type="hidden"
      name="referencesList"
      value="${reference.referenceId}
      "/>
      <c:set var="counter"
      value="${loop.index}"/>
    </td>
  </c:if>
  </tr>
  </c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
  <c:choose>
    <c:when
    test="${CollectionsForm.privilege
    ge 2 &&
    CollectionsForm.downloadable eq
    'Yes'}" >
      <td class="font-italic"
      colspan="5">No References
      Yet</td>
    </c:when>
    <c:otherwise>
      <td class="font-italic"
      colspan="4">No References
      Yet</td>
    </c:otherwise>
  </c:choose>
    </tr>
    </c:otherwise>
  </c:choose>
  </table>
  </html:form>
  </td>
 </tr>
<c:if
test="${CollectionsForm.privilege ge 2
&& CollectionsForm.downloadable eq
'Yes'}" >
  <tr>
    <td>
      <span style="color:#0000FF">
      <a href="javascript:void(1)"
      onclick="checkAll('${counter}')">
      Check All</a> |
      <a href="javascript:void(1)"
      onclick="uncheckAll('${counter}')
      ">Uncheck All</a> 
      </span>
      <span class="font-italic">With
      Selected : </span>  
      <img src="./images/export2.png"
      width="15px" height="15px"
      title="Export References"
      style="position:relative;
      left:5px; top:2px"/>
      <html:submit property="command"
      value="Export"
      styleClass="button-small" />
    </td>
  </tr>
</c:if>
</table>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>    
<c:choose>
  <c:when
  test="${CollectionsForm.privilege ge
  3}" >
    <html:submit property="command"
    value="Add Reference"
    styleClass="button-long-blue" />
    <c:if
    test="${CollectionsForm.privilege eq
    4}" >
      <html:submit property="command"
      value="View Members"
      styleClass="button-long-blue" />
    </c:if>
  </c:when>
  <c:when
  test="${CollectionsForm.privilege le
  1}" >
    <html:submit property="command"
    value="Send Membership Request"
    styleClass="button-superlong-blue"
    />
  </c:when>
</c:choose>
```

```
        </td>
       </tr>
       <tr>
        <td> </td>
       </tr>
      </table>
      </c:if>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
       &copy; <bean:message
       key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/VirhoReferencesUsers"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
       <jsp:include
       page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
       <tr>
        <td class="breadcrumb">
         <html:link
         page="/Home.do"><bean:message
         key="button.virho.home"
         /></html:link> &#187;
         <html:link
         page="/ViewCollections.do">Virho
         References Collection</html:link>
         &#187;
         <html:link
         page="/ViewReferences.do?collection=
         ${collection}">${collection}</html:l
         ink> &#187;
         <span class="breadcrumb-
         selected">View Members</span>
        </td>
       </tr>
       <tr>
        <td> </td>
       </tr>
       <tr>
        <td class="content-title-form"
        colspan="2">List of Collection
        Members</td>
       </tr>
       <tr>
        <td>
        <table class="content-desc-form">
         <tr>
          <td>
          <table class="letterList">
           <tr>
            <td>
            <c:forEach var="letter"
            items="A,B,C,D,E,F,G,H,I,J,K,L,
            M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
            varStatus="loop">
           <c:choose>
            <c:when
            test="${ViewUsersForm.letters[l
            oop.index] eq letter}">
             <html:link
             page="/VirhoReferencesUsers.do
             ?list=${letter}"><span
             class="font-
             letter">${letter}</span></html
             :link> 
            </c:when>
            <c:otherwise>
             ${letter} 
            </c:otherwise>
           </c:choose>
            </c:forEach>
            <html:link
            page="/VirhoReferencesUsers.do?
            list=">View All</html:link>
            </td>
           </tr>
          </table>
          </td>
         </tr>
         <tr>
          <td>
          <table class="table-content">
           <tr>
            <th>Name</th>
            <th>Username</th>
            <th>Affiliation</th>
            <th>Role</th>
           </tr>
          <c:choose>
           <c:when
           test="${fn:length(ViewUsersForm.u
           sers) gt 0}">
           <c:forEach var="user"
           items="${ViewUsersForm.users}">
           <tr>
            <td>
           <c:choose>
```

```
        <c:when
        test="${requestScope.privilege
        eq 4}" >
          <html:link
          page="/UserProfileDetails.do?u
          ser=${user.username}" >
          <span class="font-
          blue">${user.firstName} $
          {user.lastName}</span>
        </html:link>
        </c:when>
        <c:otherwise>
          ${user.firstName} ${user.
          lastName}
        </c:otherwise>
      </c:choose>
        </td>
        <td>${user.username}</td>
        <td>${user.affiliation}</td>
        <td>
      <c:choose>
        <c:when
        test="${(requestScope.privilege
        eq 4) and (user.username ne
        userLogin)}">
          <select name="role">
            <option value="Remove
            Privilege">Remove
            Privilege</option>
          <c:forEach var="roleName"
          items="${requestScope.roles}">
            <option value="${roleName}"
            <c:if test="${roleName eq
            user.projectRole}">selected=
            "selected"</c:if>>${roleName
            }</option>
          </c:forEach>
          </select>
          <input type="hidden"
          name="oldRole"
          value="${user.projectRole}" />
          <input type="hidden"
          name="userName"
          value="${user.username}" />
        </c:when>
        <c:otherwise>
          ${user.projectRole}
        </c:otherwise>
      </c:choose>
        </td>
      </tr>
      </c:forEach>
      </c:when>
      <c:otherwise>
      <tr>
        <td class="font-italic"
        colspan="4">
          There are no members yet
        </td>
      </tr>
      </c:otherwise>
    </c:choose>
    </table>
    </td>
  </tr>
</table>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
      
  <c:if test="${requestScope.privilege
  eq 4}" >
          <html:submit property="command"
          value="Change Privilege Role"
          styleClass="button-superlong-blue"
          />
        </c:if>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

VIRHO INFORMATION SERVICES

addVirus.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/AddVirus" styleId="form"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
```

```
<tr>
  <td class="breadcrumb">
    <html:link
    page="/Home.do"><bean:message
    key="button.virho.home"
    /></html:link> &#187;
    <html:link
    page="/ViewVirus.do?list=" >Virus
    List</html:link> &#187;
    <span class="breadcrumb-
    selected">Add Virus</span>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Add Virus</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
    <tr>
      <td class="note" colspan="2">
        Note: All required fields are
        marked with an asterisk (*).
      </td>
    </tr>
    <tr>
      <td colspan="2"><span class="error-
      text-big"><html:errors
      property="errorMessage"
      /></span></td>
    </tr>
    <tr>
      <td class="font-bold"
      width="200px">Common Name * </td>
      <td>
        <html:text property="commonName"
        size="60" style="remove-error-
        field" errorStyleClass="error-
        field"
        errorKey="org.apache.struts.actio
        n.ERROR" />
        <span class="error-
        text"><html:errors
        property="commonName" /></span>
      </td>
    </tr>
    <tr>
      <td class="font-bold">Scientific
      Name</td>
      <td>
        <html:text
        property="scientificName"
        size="60" style="remove-error-
        field" errorStyleClass="error-
        field"
        errorKey="org.apache.struts.actio
        n.ERROR" />
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Description</td>
      <td>
        <html:textarea
        property="description" cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold">Serotype</td>
      <td>
        <html:text property="serotype"
        size="50" style="remove-error-
        field" errorStyleClass="error-
        field"
        errorKey="org.apache.struts.actio
        n.ERROR" />
      </td>
    </tr>
    <tr>
      <td class="font-bold">Host</td>
      <td>
        <html:text property="host"
        size="50" style="remove-error-
        field" errorStyleClass="error-
        field"
        errorKey="org.apache.struts.actio
        n.ERROR" />
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Virus
      Architecture</td>
      <td>
        <html:textarea
        property="virusArchitecture"
        cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Infectivity</td>
      <td>
        <html:textarea
        property="infectivity" cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Manner of
      Infection</td>
      <td>
        <html:textarea
        property="mannerOfInfection"
        cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Vector</td>
      <td>
        <html:textarea property="vector"
        cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Antimicrobial
      Treatment</td>
      <td>
        <html:textarea
        property="antimicrobialTreatment"
        cols="60"
        rows="3"></html:textarea>
      </td>
    </tr>
    <tr>
      <td class="font-bold"
      valign="top">Receptors</td>
      <td>
```

```
            <html:textarea
            property="receptors" cols="60"
            rows="3"></html:textarea>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>    
        <html:submit property="command"
        value="Submit" styleClass="button-
        long-blue" />
        <html:reset property="command"
        value="Reset" styleClass="button-
        long-blue" />
        <html:cancel property="command"
        value="Cancel" styleClass="button-
        long-blue" />
      </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
 <tr class="footer">
   <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
   </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

advancedSearch.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/AdvancedSearch"
  styleId="form" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
<tr valign="top">
  <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
  </td>
  <td>
  <c:if test="${(userLogin ne null) and
  ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or (statusLogin
  eq 'superadmin'))}">
  <table class="content">
    <tr>
      <td class="breadcrumb">
        <html:link
        page="/Home.do"><bean:message
        key="button.virho.home"
        /></html:link> &#187;
        <span class="breadcrumb-
        selected">Advanced Search</span>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td class="content-title-form"
      colspan="2">Advanced Search</td>
    </tr>
    <tr>
      <td>
      <table class="content-desc-form">
        <tr>
          <td class="font-bold"
          colspan="2">VirhoExperiments</td>
        </tr>
        <tr>
          <td
          width="100px">   &nb
          sp;Search for</td>
          <td><html:text
          property="expSearchTerm" size="60"
          /></td>
        </tr>
        <tr>
          <td class="font-bold"
          colspan="2">VirhoImages</td>
        </tr>
        <tr>
          <td
          width="100px">   &nb
          sp;Search for</td>
          <td><html:text
          property="imgSearchTerm" size="60"
          /></td>
        </tr>
        <tr>
          <td class="font-bold"
          colspan="2">VirhoModels</td>
        </tr>
        <tr>
          <td
          width="100px">   &nb
          sp;Search for</td>
          <td><html:text
          property="modSearchTerm" size="60"
          /></td>
        </tr>
        <tr>
          <td class="font-bold"
          colspan="2">VirhoReferences</td>
        </tr>
        <tr>
          <td
          width="100px">   Sea
          rch for</td>
```

```
            <td><html:text
            property="refSearchTerm" size="60"
            /></td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>
              
          <html:submit property="command"
          value="Search" styleClass="button-
          long-blue" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
</table>
</html:form>
</body>
</html>
```

advancedSearchResult.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  function sortBy() {
    $('#form').submit();
  }
  </script>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr class="content" valign="top">
<td class="sidebar">
  <jsp:include
  page="/rd_virui/sidebar.jsp" />
</td>
<td>
<c:if test="${(userLogin ne null) and
((statusLogin eq 'activated') or
(statusLogin eq 'admin') or (statusLogin
eq 'superadmin'))}">
<table class="content">
  <tr>
    <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <span class="breadcrumb-
      selected">Search Results</span>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <html:form action="/QuickSearch"
  method="GET">
    <jsp:include
    page="/rd_virhointernalsearch/quickSea
    rch.jsp" />
  </html:form>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td class="content-title">Search
    Results</td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr>
    <td>
    <html:form action="/AdvancedSearch"
    method="GET" styleId="form">
    <table class="table-content">
    <c:if test="${not empty
    SearchForm.expSearchTerm}">
      <tr>
        <th
        align="left">VirhoExperiments</th>
      </tr>
      <html:hidden
      property="expSearchTerm"
      value="${SearchForm.expSearchTerm}"
      />
    <c:choose>
      <c:when test="${not empty
      SearchForm.experiments}">
      <tr style="background-
      color:#E1E8F1;">
        <td>
          Sort results by :
          <html:select property="expSortBy"
          onchange="sortBy()">
            <html:option
            value="experiment">Experiment</
            html:option>
            <html:option
            value="author">Author</html:opt
            ion>
            <html:option
            value="project">Project</html:o
            ption>
          </html:select>
        </td>
      </tr>
      <tr>
```

```
    <td class="font-italic">Search
    results related to key term
    <b><c:out
    value="${SearchForm.expSearchTerm}"
    /></b> in VirhoExperiments</td>
  </tr>
<c:forEach var="detail"
items="${SearchForm.experiments}">
  <tr>
    <td>
      <span class="font-
      bold">Experiment:</span>
      <html:link
      page="/ViewExperimentDetails.do?m
      odule=VirhoExperiments&projectId=
      ${detail.projectId}&experiment=${
      detail.experimentName}">
      <span class="font-blue"><c:out
      value="${detail.experimentName}"
      /></span>
      </html:link><br/>
      <span class="font-
      bold">Author:</span> <c:out
      value="${detail.author}" /><br/>
      <span class="font-
      bold">Project:</span> <c:out
      value="${detail.projectName}" />
    </td>
  </tr>
</c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-italic">No results
    related to key term <b><c:out
    value="${SearchForm.expSearchTerm}"
    /></b> were found in
    VirhoExperiments</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td> </td>
  </tr>
</c:if>
<c:if test="${not empty
SearchForm.imgSearchTerm}">
  <tr>
    <th align="left">VirhoImages</th>
  </tr>
    <html:hidden
    property="imgSearchTerm"
    value="${SearchForm.imgSearchTerm}"
    />
  <c:choose>
    <c:when test="${not empty
    SearchForm.images}">
    <tr style="background-
    color:#E1E8F1;">
    <td>
      Sort results by :
      <html:select property="imgSortBy"
      onchange="sortBy()">
        <html:option
        value="image">Image
        Name</html:option>
        <html:option
        value="category">Category</html
        :option>
        <html:option
        value="accessibility">Accessibi
        lity</html:option>
        <html:option
        value="description">Description
        </html:option>
        <html:option
        value="imageSet">Image
        Set</html:option>
      </html:select>
    </td>
  </tr>
  <tr>
    <td class="font-italic">Search
    results related to key term
    <b><c:out
    value="${SearchForm.imgSearchTerm}"
    /></b> in VirhoImages</td>
  </tr>
<c:forEach var="detail"
items="${SearchForm.images}">
  <tr>
    <td>
  <c:choose>
    <c:when test="${(privilege eq true)
    or (detail.accessibility eq
    'Public')}">
    <span style="float:left; padding-
    right:15px">
    <html:link
    page="/ViewImage.do?module=VirhoIma
    ges&projectSet=${detail.imageSet}&i
    mage=${detail.imagePathDB}">
    <c:choose>
      <c:when
      test="${fn:endsWith(detail.imageP
      athDB, 'exe') or
      fn:endsWith(detail.imagePathDB,
      'avi') or
      fn:endsWith(detail.imagePathDB,
      'mp4') ||
      fn:endsWith(detail.imagePathDB,
      'wmv') or
      fn:endsWith(detail.imagePathDB,
      'jar')}">
        <img
        src="images/defaultThumb.gif"
        class="thumbnail" />
      </c:when>
      <c:when
      test="${fn:endsWith(detail.imageP
      athDB, 'flv')}">
        <object
        classid="clsid:d27cdb6e-ae6d-
        11cf-96b8-444553540000"
        codebase="http://download.macro
        media.com/pub/shockwave/cabs/fl
        ash/swflash.cab#version=9,0,0,0
        " width="100" height="75"
        id="VideoPlayer"
        align="middle">
          <param
          name="allowScriptAccess"
          value="*" />
          <param name="allowFullScreen"
          value="true" />
          <param name="movie"
          value="http://www.platipus.nl/
          flvplayer/download/1.0/FLVPlay
          er.swf?video=http://www.bioinf
          o.mpg.de/virholex/repository/v
          irhoimages/${detail.imagePathD
          B}&autoplay=false" />
          <param name"quality"
          value="high" />
          <param name="bgcolor"
          value="#ffffff" />
          <embed
          src="http://www.platipus.nl/fl
          vplayer/download/1.0/FLVPlayer
          .swf?video=http://www.bioinfo.
          mpg.de/virholex/repository/vir
```

```
        hoimages/${detail.imagePathDB}
        &autoplay=false"
        quality="high"
        bgcolor="#000000" width="100"
        height="75" name="VideoPlayer"
        align="middle"
        allowScriptAccess="*"
        allowFullScreen="true"
        type="application/x-shockwave-
        flash"
        pluginspage="http://www.macrom
        edia.com/go/getflashplayer" />
      </object>
    </c:when>
    <c:otherwise>
      <img
      src="repository/virhoimages/${d
      etail.imagePathDB}"
      class="thumbnail" />
    </c:otherwise>
  </c:choose>
  </html:link>
  </span>
  <span class="font-bold">Image
  Name:</span>
  <html:link
  page="/ViewImage.do?module=VirhoIma
  ges&projectSet=${detail.imageSet}&i
  mage=${detail.imagePathDB}">
  <span class="font-blue"><c:out
  value="${detail.imagePath}"
  /></span>
  </html:link><br/>
  </c:when>
  <c:otherwise>
  <span style="float:left; padding-
  right:15px">
  <img src="images/restricted.png"
  class="thumbnail" />
  </span>
  <span class="font-bold">Image
  Name:</span>
  <span class="font-blue"><c:out
  value="${detail.imagePath}"
  /></span><br/>
  </c:otherwise>
 </c:choose>
  <span class="font-
  bold">Category:</span> <c:out
  value="${detail.category}" /><br/>
  <span class="font-
  bold">Accessibility:</span> <c:out
  value="${detail.accessibility}"
  /><br/>
  <span class="font-
  bold">Description:</span> <c:out
  value="${detail.description}"
  /><br/>
  <span class="font-bold">Image
  Set:</span> <c:out
  value="${detail.imageSet}" />
  </td>
 </tr>
</c:forEach>
 </c:when>
 <c:otherwise>
 <tr>
  <td class="font-italic">No results
  related to key term <b><c:out
  value="${SearchForm.imgSearchTerm}"
  /></b> were found in
  VirhoImages</td>
 </tr>
 </c:otherwise>
</c:choose>
 <tr>
  <td> </td>
 </tr>
</c:if>
<c:if test="${not empty
SearchForm.modSearchTerm}">
 <tr>
  <th align="left">VirhoModels</th>
 </tr>
 <html:hidden
 property="modSearchTerm"
 value="${SearchForm.modSearchTerm}"
 />
<c:choose>
 <c:when test="${not empty
 SearchForm.models}">
 <tr style="background-
 color:#E1E8F1;">
  <td>
  Sort results by :
  <html:select property="modSortBy"
  onchange="sortBy()">
   <html:option
   value="model">Model</html:option>
   <html:option
   value="author">Author</html:optio
   n>
   <html:option
   value="project">Project</html:opt
   ion>
   <html:option
   value="classification">Classifica
   tion</html:option>
  </html:select>
  </td>
 </tr>
 <tr>
  <td class="font-italic">Search
  results related to key term
  <b><c:out
  value="${SearchForm.modSearchTerm}"
  /></b> in VirhoModels</td>
 </tr>
<c:forEach var="detail"
items="${SearchForm.models}">
 <tr>
  <td>
   <span class="font-
   bold">Model:</span>
   <html:link
   page="/ViewModelDetails.do?module
   =VirhoModels&classification=${det
   ail.classification}&project=${det
   ail.projectName}&model=${detail.m
   odelName}">
   <span class="font-blue"><c:out
   value="${detail.modelName}"
   /></span>
   </html:link><br/>
   <span class="font-
   bold">Author:</span> <c:out
   value="${detail.author}" /><br/>
   <span class="font-
   bold">Project:</span> <c:out
   value="${detail.projectName}"
   /><br/>
   <span class="font-
   bold">Classification:</span>
   <c:out
   value="${detail.classification}"
   />
  </td>
 </tr>
</c:forEach>
 </c:when>
 <c:otherwise>
 <tr>
```

```
    <td class="font-italic">No results
    related to key term <b><c:out
    value="${SearchForm.modSearchTerm}"
    /></b> were found in
    VirhoModels</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td> </td>
  </tr>
</c:if>
<c:if test="${not empty
SearchForm.refSearchTerm}">
  <tr>
    <th
    align="left">VirhoReferences</th>
  </tr>
  <html:hidden
  property="refSearchTerm"
  value="${SearchForm.refSearchTerm}"
  />
<c:choose>
  <c:when test="${not empty
  SearchForm.references}">
  <tr style="background-
  color:#E1E8F1;">
    <td>
    Sort results by :
    <html:select property="refSortBy"
    onchange="sortBy()">
      <html:option
      value="reference">Reference</html
      :option>
      <html:option
      value="author">Author</html:optio
      n>
      <html:option
      value="collection">Collection</ht
      ml:option>
    </html:select>
    </td>
  </tr>
  <tr>
    <td class="font-italic">Search
    results related to key term
    <b><c:out
    value="${SearchForm.refSearchTerm}"
    /></b> in VirhoReferences</td>
  </tr>
<c:forEach var="detail"
items="${SearchForm.references}">
  <tr>
    <td>
      <span class="font-
      bold">Reference:</span>
      <html:link
      page="/ViewReferenceDetails.do?mo
      dule=VirhoReferences&collection=$
      {detail.collection}&reference=${d
      etail.title}&refId=${detail.refer
      enceId}">
      <span class="font-blue"><c:out
      value="${detail.title}" /></span>
      </html:link><br/>
      <span class="font-
      bold">Author:</span> <c:out
      value="${detail.author}" /><br/>
      <span class="font-
      bold">Collection:</span> <c:out
      value="${detail.collection}" />
    </td>
  </tr>
</c:forEach>
  </c:when>
  <c:otherwise>
        <tr>
          <td class="font-italic">No results
          related to key term <b><c:out
          value="${SearchForm.refSearchTerm}"
          /></b> were found in
          VirhoReferences</td>
        </tr>
        </c:otherwise>
      </c:choose>
        <tr>
          <td> </td>
        </tr>
      </c:if>
      </table>
      </html:form>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
  </table>
  </c:if>
  </td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</body>
</html>
```

AJAXResult.jsp

```
<%@ page language="java"
contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
${result}
```

alternativeExternalSearch.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <link type="text/css" href="./files/jquery-
  ui-1.8.23.custom.css" rel="stylesheet" />

  <script type="text/javascript">
  function subsearch(){
    open(document.searchF.base.value+"&db="+doc
    ument.searchF.db.value+"&term="+document.se
    archF.key.value+"&dispmax="+document.search
    F.limit.value,"new");
  }

  function subretrieve(){
```

```
  open(document.retrieveF.base.value+"&db="+d
  ocument.retrieveF.db.value+"&list_uids="+do
  cument.retrieveF.uids.value,"new");
 }

 function sublink(){
  open(document.linkF.base.value+"&db="+docum
  ent.linkF.db2.value+"&dbFrom="+document.lin
  kF.db1.value+"&uid="+document.linkF.uid.val
  ue,"new");
 }

 var ids = new Array();
 ids[0] = "sbut";
 ids[1] = "rbut";
 ids[2] = "lbut";

 function selectSearchPanel(tabIndex){
  for (var i=0;i<ids.length;i++){
   document.getElementById(ids[i]+"Search").
   style.display = 'none';
   document.getElementById(ids[i]).style.bac
   kgroundColor = '#FFFFFF';
  }
  document.getElementById(ids[tabIndex]+"Sear
  ch").style.display = 'block';
  document.getElementById(ids[tabIndex]).styl
  e.backgroundColor = '#DDDDFF';
 }
 </script>
</head>
<body>
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/VirhoExternalSearch.do"><bean
      :message key="virho.externalSearch"
      /></html:link> &#187;
      <span class="breadcrumb-
      selected"><bean:message
      key="virho.externalAltSearch"
      /></span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2"><bean:message
     key="virho.externalAltSearch" /></td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr align="center">
       <td onClick="selectSearchPanel(0)"
       id="sbut" bgcolor="#DDDDFF">
        <div class="button-
        search">Search</div>
       </td>
       <td onClick="selectSearchPanel(1)"
       id="rbut" bgcolor="#FFFFFF">
        <div class="button-
        search">Retrieve</div></td>
       <td onClick="selectSearchPanel(2)"
       id="lbut" bgcolor="#FFFFFF">
        <div class="button-
        search">Link</div>
       </td>
      </tr>
      <tr>
       <td colspan="3">
       <div id="sbutSearch"
       style="display:block">
       <form name="searchF">
       <input type="hidden" name="base"
       value="<bean:message
       key="link.search.ncbi.alt1" />">
       <table align="center"
       class="content-search">
        <tr>
         <td>Select Entrez Database</td>
         <td>
         <select id="db">
          <option value="domains">3d
          Domains </option>
          <option value="cdd">Domains
          </option>
          <option value="genome">Genome
          </option>
          <option
          value="nucleotide">Nucleotide
          </option>
          <option value="omim">OMIM
          </option>
          <option value="popset">PopSet
          </option>
          <option
          value="protein">Protein
          </option>
          <option
          value="geo">ProbeSet</option>
          <option value="pubmed">PubMed
          </option>
          <option
          value="structure">Structure
          </option>
          <option value="snp">SNP
          </option>
          <option
          value="taxonomy">Taxonomy
          </option>
          <option
          value="unigene">UniGene
          </option>
          <option value="unists">UniSTS
          </option>
         </select>
         </td>
        </tr>
        <tr>
         <td>Type in search term </td>
         <td><input type="text"
         size="20" id="key"
         name="key"></td>
        </tr>
        <tr>
         <td>Limit results display</td>
         <td>
         <select id="limit">
          <option value="5">5</option>
          <option value="10">10</option>
          <option value="15">15</option>
          <option value="20">20</option>
          <option value="25">25</option>
```

```
    <option value="30">30</option>
    <option value="35">35</option>
    <option value="40">40</option>
    <option value="45">45</option>
    <option value="50">50</option>
   </select>
   </td>
  </tr>
  <tr>
   <td colspan="2" align="center">
    <input type="button"
    value="Search"
    onClick="subsearch()"
    class="button-long-blue">
   </td>
  </tr>
</table>
</form>
</div>

<div id="rbutSearch"
style="display:none">
<form name="retrieveF">
<input type="hidden" name="base"
value="<bean:message
key="link.search.ncbi.alt2" />">
<table align="center"
class="content-search">
 <tr>
  <td>Select Entrez Database</td>
  <td>
  <select id="db">
   <option value="domains">3d
   Domains </option>
   <option value="cdd">Domains
   </option>
   <option value="genome">Genome
   </option>
   <option
   value="nucleotide">Nucleotide
   </option>
   <option value="omim">OMIM
   </option>
   <option value="popset">PopSet
   </option>
   <option
   value="protein">Protein
   </option>
   <option
   value="geo">ProbeSet</option>
   <option value="pubmed">PubMed
   </option>
   <option
   value="structure">Structure
   </option>
   <option value="snp">SNP
   </option>
   <option
   value="taxonomy">Taxonomy
   </option>
   <option
   value="unigene">UniGene
   </option>
   <option value="unists">UniSTS
   </option>
  </select>
  </td>
 </tr>
 <tr>
  <td>Enter UIDs to search</td>
  <td><input type="text"
  size="12" id="uids"
  name="uids"></td>
 </tr>
 <tr>
  <td colspan="2" align="center">
   <input type="button"
   value="Retrieve"
   onClick="subretrieve()"
   class="button-long-blue">
  </td>
 </tr>
</table>
</form>
</div>

<div id="lbutSearch"
style="display:none">
<form name="linkF">
<input type="hidden" name="base"
value="<bean:message
key="link.search.ncbi.alt3" />">
<table align="center"
class="content-search">
 <tr>
  <td>From database</td>
  <td>
  <select id="db1">
   <option value="domains">3d
   Domains </option>
   <option value="cdd">Domains
   </option>
   <option value="genome">Genome
   </option>
   <option
   value="nucleotide">Nucleotide
   </option>
   <option value="omim">OMIM
   </option>
   <option value="popset">PopSet
   </option>
   <option
   value="protein">Protein
   </option>
   <option
   value="geo">ProbeSet</option>
   <option value="pubmed">PubMed
   </option>
   <option
   value="structure">Structure
   </option>
   <option value="snp">SNP
   </option>
   <option
   value="taxonomy">Taxonomy
   </option>
   <option
   value="unigene">UniGene
   </option>
   <option value="unists">UniSTS
   </option>
  </select>
  </td>
 </tr>
 <tr>
  <td>To database</td>
  <td>
  <select id="db2">
   <option value="domains">3d
   Domains </option>
   <option value="cdd">Domains
   </option>
   <option value="genome">Genome
   </option>
   <option
   value="nucleotide">Nucleotide
   </option>
   <option value="omim">OMIM
   </option>
   <option value="popset">PopSet
   </option>
```

```
                <option
                value="protein">Protein
                </option>
                <option
                value="geo">ProbeSet</option>
                <option value="pubmed">PubMed
                </option>
                <option
                value="structure">Structure
                </option>
                <option value="snp">SNP
                </option>
                <option
                value="taxonomy">Taxonomy
                </option>
                <option
                value="unigene">UniGene
                </option>
                <option value="unists">UniSTS
                </option>
              </select>
              </td>
            </tr>
            <tr>
              <td>Primary ID</td>
              <td><input type="text"
              size="12" id="key"
              name="uid"></td>
            </tr>
            <tr>
              <td colspan="2" align="center">
              <input type="button"
              value="Link" size="30"
              onClick="sublink()"
              class="button-long-blue">
              </td>
            </tr>
          </table>
          </form>
          </div>
          </td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr class="footer">
  <td colspan="2">
   &copy; <bean:message
   key="virho.copyright" />
  </td>
 </tr>
</table>
</body>
</html>
```

DisplayHS.jsp

```
<%@ page language="java"
contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core"%>

<h3>
  <font face="verdana"
  color="red">${stage}</font>
  <font face="verdana"> Stage </font>
</h3>

<font face="verdana" size="2">
  ${info}
</font>
<c:if test="${privileged eq 'true' }">
<form method="post" action="ManageHS.jsp">
  <input type="hidden" name="stage"
  value="${stage}">
  <input type="hidden" name="info"
  value="${info}">
  <input type="hidden" name="id"
  value="${param.id}">
  <input type="submit" name="editContent"
  value="Edit This Information">
</form>
</c:if>
```

editVirus.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/EditVirus"
  styleId="form" method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewVirus.do?list=" >Virus
            List</html:link> &#187;
            <html:link
            page="/ViewVirusDetails.do?virus=${v
            irus}" >${virus}</html:link> &#187;
            <span class="breadcrumb-
            selected">Edit Virus</span>
          </td>
        </tr>
        <tr>
```

```
  <td> </td>
</tr>
<tr>
  <td class="content-title-form"
  colspan="2">Edit Virus</td>
</tr>
<tr>
  <td>
  <table class="content-desc-form">
   <tr>
    <td class="note" colspan="2">
     Note: All required fields are
     marked with an asterisk (*).
    </td>
   </tr>
   <tr>
    <td colspan="2"><span class="error-
    text-big"><html:errors
    property="errorMessage"
    /></span></td>
   </tr>
   <tr>
    <td class="font-bold"
    width="200px">Common Name * </td>
    <td>
     <html:text property="commonName"
     size="60" style="remove-error-
     field" errorStyleClass="error-
     field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
     <span class="error-
     text"><html:errors
     property="commonName" /></span>
     <html:hidden
     property="commonNameDB"
     value="${VirusForm.commonNameDB}"
     />
    </td>
   </tr>
   <tr>
    <td class="font-bold">Scientific
    Name</td>
    <td>
     <html:text
     property="scientificName"
     size="60" style="remove-error-
     field" errorStyleClass="error-
     field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Description</td>
    <td>
     <html:textarea
     property="description" cols="60"
     rows="3"></html:textarea>
     <html:hidden
     property="descriptionDB"
     value="${VirusForm.descriptionDB}
     " />
    </td>
   </tr>
   <tr>
    <td class="font-bold">Serotype</td>
    <td>
     <html:text property="serotype"
     size="50" style="remove-error-
     field" errorStyleClass="error-
     field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
    </td>
   </tr>
   <tr>
    <td class="font-bold">Host</td>
    <td>
     <html:text property="host"
     size="50" style="remove-error-
     field" errorStyleClass="error-
     field"
     errorKey="org.apache.struts.actio
     n.ERROR" />
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Virus
    Architecture</td>
    <td>
     <html:textarea
     property="virusArchitecture"
     cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Infectivity</td>
    <td>
     <html:textarea
     property="infectivity" cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Manner of
    Infection</td>
    <td>
     <html:textarea
     property="mannerOfInfection"
     cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Vector</td>
    <td>
     <html:textarea property="vector"
     cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Antimicrobial
    Treatment</td>
    <td>
     <html:textarea
     property="antimicrobialTreatment"
     cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
   <tr>
    <td class="font-bold"
    valign="top">Receptors</td>
    <td>
     <html:textarea
     property="receptors" cols="60"
     rows="3"></html:textarea>
    </td>
   </tr>
  </table>
  </td>
</tr>
<tr>
```

```
        <td> </td>
      </tr>
      <tr>
        <td>    
          <html:submit property="command"
          value="Submit" styleClass="button-
          long-blue" />
          <html:reset property="command"
          value="Reset" styleClass="button-
          long-blue" />
          <html:cancel property="command"
          value="Cancel" styleClass="button-
          long-blue" />
        </td>
      </tr>
    </table>
    </c:if>
    </td>
  </tr>
  <tr>
    <td> </td>
  </tr>
  <tr class="footer">
    <td colspan="2">
      &copy; <bean:message
      key="virho.copyright" />
    </td>
  </tr>
 </table>
 </html:form>
</body>
</html>
```

HotspotJSON.jsp

```
<%@ page trimDirectiveWhitespaces="true" %>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>
[
<c:forEach var="hotspot" items="${result}"
varStatus="status">
 {"id":"${hotspot.id}", "photoid":
 "${hotspot.name}", "x": ${hotspot.x},
 "y": ${hotspot.y}, "width":
 ${hotspot.width}, "height":
 ${hotspot.height}, "message":
 "${hotspot.stage}" }
 <c:if test="${!status.last}">,</c:if>
</c:forEach>
]
```

ManageHS.jsp

```
<%@ page language="java"
contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title>Insert title here</title>
</head>
 <body>
 <%
 if((request.getParameter("editContent")).equ
 als("Edit This Information")) {
   String info = request.getParameter("info");
   String msg = info.replace("<br>","");
 %>
 <form action="ManageHSTool.jsp"
 method="post">
 <table>
  <tr>
    <td><font face="verdana">Virus: <font
    color="red">
      <b><%=
      session.getAttribute("virus")%></b></fo
      nt>
      <input type="hidden" name="virus"
      value="<%=
      session.getAttribute("virus")%>">
      </font>
    </td>
  </tr>
  <tr>
    <td><font face="verdana">Stage: <font
    color="red">
      <b><%=
      request.getParameter("stage")%></b></fo
      nt>
      <input type="hidden" name="stage"
      value="<%=
      request.getParameter("stage")%>">
      </font>
    </td>
  </tr>
  <tr>
    <td><textarea rows="25" cols="55"
    name="info"><%= msg%></textarea>
    </td>
  </tr>
  <tr>
    <td>
      <input type="hidden" name="id"
      value="<%=
      request.getParameter("id")%>">
      <input type="submit" name="action"
      value="Save Changes">
      <input type="submit" name="action"
      value="Cancel">
    </td>
  </tr>
 </table>
 </form>
 <%  } %>
</body>
</html>
```

ManageHSTool.jsp

```
<%@ page language="java"
contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
 <meta http-equiv="Content-Type"
 content="text/html; charset=ISO-8859-1">
 <title></title>
</head>
<jsp:useBean id = "manage"
class="com.virholex.hotspots.ManageHSServlet"
scope="request"/>
<body>
 <%
 String action =
 request.getParameter("action");
 String stage =
 request.getParameter("stage");
 String virus =
 request.getParameter("virus");
```

```
String info = request.getParameter("info");
String id = request.getParameter("id");
String setname = virus + " VirhoHotspot";
if(action.equals("Save Changes") &&
manage.saveChanges(info, id)) {
%>
<form action="viewVirusDetails.jsp"
method="post">
<table>
  <tr>
    <td>
      <font face="verdana" size="2">
      You have successfully changed the
      content for
      <font color="red"><%= virus%> </font>
      <font color="blue"><%= stage%> </font>
      </font>
    </td>
  </tr>
  <tr>
    <td>
      <input type="hidden" name="virus"
      value="<%= virus %>">
      <input type="hidden" name="set_name"
      value="<%= setname %>">
      <center><input type="submit"
      name="success" value="OK"></center>
    </td>
  </tr>
</table>
</form>
<%   } else { %>
  <jsp:forward page="viewVirusDetails.jsp">
  <jsp:param name="virus" value="<%= virus
  %>"/>
  <jsp:param name="set_name" value="<%=
  setname %>"/>
  </jsp:forward>
<%  } %>
</body>
</html>
```

quickSearch.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<tr class="search-container">
  <td>
    <div style="position:relative; left:25px">
    Quick Search <html:text
    property="searchKey" size="40" /> in
    <html:select property="module">
      <html:option value="All Modules">All
      Modules</html:option>
      <html:option
      value="VirhoExperiments">VirhoExperiments
      </html:option>
      <html:option
      value="VirhoImages">VirhoImages</html:opt
      ion>
      <html:option
      value="VirhoModels">VirhoModels</html:opt
      ion>
      <html:option
      value="VirhoReferences">VirhoReferences</
      html:option>
    </html:select>
    <html:submit property="command"
    value="Search" />
      
    <span class="font-blue"><html:link
    page="/AdvancedSearch.do">Advanced
    Search</html:link></span>
    </div>
  </td>
</tr>
```

quickSearchResult.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script src="./files/jquery-
  1.8.0.js"></script>
  <script>
  function sortBy() {
    $('#form').submit();
  }
  </script>
</head>
<body>
  <html:form action="/QuickSearch"
  method="GET" styleId="form">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr class="content" valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected">Search Results</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <jsp:include
        page="/rd_virhointernalsearch/quickSear
        ch.jsp" />
        <tr>
          <td> </td>
        </tr>
        <tr>
```

```
  <td class="content-title">Search
  Results</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr>
  <td>
  <table class="table-content">
  <c:if test="${(SearchForm.module eq
  'VirhoExperiments') or
  (SearchForm.module eq 'All
  Modules')}">
    <tr>
      <th
      align="left">VirhoExperiments</th>
    </tr>
  <c:choose>
    <c:when test="${not empty
    SearchForm.experiments}">
    <tr style="background-
    color:#E1E8F1;">
      <td>
      Sort results by :
      <html:select property="expSortBy"
      onchange="sortBy()">
        <html:option
        value="experiment">Experiment</ht
        ml:option>
        <html:option
        value="author">Author</html:optio
        n>
        <html:option
        value="project">Project</html:opt
        ion>
      </html:select>
      </td>
    </tr>
    <tr>
      <td class="font-italic">Search
      results related to key term
      <b><c:out
      value="${SearchForm.searchKey}"
      /></b> in VirhoExperiments</td>
    </tr>
  <c:forEach var="detail"
  items="${SearchForm.experiments}">
    <tr>
      <td>
        <span class="font-
        bold">Experiment:</span>
        <html:link
        page="/ViewExperimentDetails.do?m
        odule=VirhoExperiments&projectId=
        ${detail.projectId}&experiment=${
        detail.experimentName}">
        <span class="font-blue"><c:out
        value="${detail.experimentName}"
        /></span>
        </html:link><br/>
        <span class="font-
        bold">Author:</span> <c:out
        value="${detail.author}" /><br/>
        <span class="font-
        bold">Project:</span> <c:out
        value="${detail.projectName}" />
      </td>
    </tr>
  </c:forEach>
    </c:when>
    <c:otherwise>
    <tr>
      <td class="font-italic">No results
      related to key term <b><c:out
      value="${SearchForm.searchKey}"
      /></b> were found in
      VirhoExperiments</td>
    </tr>
    </c:otherwise>
  </c:choose>
    <tr>
      <td> </td>
    </tr>
  </c:if>
  <c:if test="${(SearchForm.module eq
  'VirhoImages') or (SearchForm.module
  eq 'All Modules')}">
    <tr>
      <th align="left">VirhoImages</th>
    </tr>
  <c:choose>
    <c:when test="${not empty
    SearchForm.images}">
    <tr style="background-
    color:#E1E8F1;">
      <td>
      Sort results by :
      <html:select property="imgSortBy"
      onchange="sortBy()">
        <html:option value="image">Image
        Name</html:option>
        <html:option
        value="category">Category</html:o
        ption>
        <html:option
        value="accessibility">Accessibili
        ty</html:option>
        <html:option
        value="description">Description</
        html:option>
        <html:option
        value="imageSet">Image
        Set</html:option>
      </html:select>
      </td>
    </tr>
    <tr>
      <td class="font-italic">Search
      results related to key term
      <b><c:out
      value="${SearchForm.searchKey}"
      /></b> in VirhoImages</td>
    </tr>
  <c:forEach var="detail"
  items="${SearchForm.images}">
    <tr>
      <td>
    <c:choose>
      <c:when test="${(privilege eq true)
      or (detail.accessibility eq
      'Public')}">
      <span style="float:left; padding-
      right:15px">
      <html:link
      page="/ViewImage.do?module=VirhoIma
      ges&projectSet=${detail.imageSet}&i
      mage=${detail.imagePathDB}">
      <c:choose>
        <c:when
        test="${fn:endsWith(detail.imageP
        athDB, 'exe') or
        fn:endsWith(detail.imagePathDB,
        'avi') or
        fn:endsWith(detail.imagePathDB,
        'mp4') ||
        fn:endsWith(detail.imagePathDB,
        'wmv') or
        fn:endsWith(detail.imagePathDB,
        'jar')}">
```

```
    <img
    src="images/defaultThumb.gif"
    class="thumbnail" />
  </c:when>
  <c:when
  test="${fn:endsWith(detail.imageP
  athDB, 'flv')}">
    <object
    classid="clsid:d27cdb6e-ae6d-
    11cf-96b8-444553540000"
    codebase="http://download.macro
    media.com/pub/shockwave/cabs/fl
    ash/swflash.cab#version=9,0,0,0
    " width="100" height="75"
    id="VideoPlayer"
    align="middle">
      <param
      name="allowScriptAccess"
      value="*" />
      <param name="allowFullScreen"
      value="true" />
      <param name="movie"
      value="http://www.platipus.nl/
      flvplayer/download/1.0/FLVPlay
      er.swf?video=http://www.bioinf
      o.mpg.de/virholex/repository/v
      irhoimages/${detail.imagePathD
      B}&autoplay=false" />
      <param name"quality"
      value="high" />
      <param name="bgcolor"
      value="#ffffff" />
      <embed
      src="http://www.platipus.nl/fl
      vplayer/download/1.0/FLVPlayer
      .swf?video=http://www.bioinfo.
      mpg.de/virholex/repository/vir
      hoimages/${detail.imagePathDB}
      &autoplay=false"
      quality="high"
      bgcolor="#000000" width="100"
      height="75" name="VideoPlayer"
      align="middle"
      allowScriptAccess="*"
      allowFullScreen="true"
      type="application/x-shockwave-
      flash"
      pluginspage="http://www.macrom
      edia.com/go/getflashplayer" />
    </object>
  </c:when>
  <c:otherwise>
    <img
    src="repository/virhoimages/${d
    etail.imagePathDB}"
    class="thumbnail" />
  </c:otherwise>
</c:choose>
</html:link>
</span>
<span class="font-bold">Image
Name:</span>
<html:link
page="/ViewImage.do?module=VirhoIma
ges&projectSet=${detail.imageSet}&i
mage=${detail.imagePathDB}">
<span class="font-blue"><c:out
value="${detail.imagePath}"
/></span>
</html:link><br/>
</c:when>
<c:otherwise>
  <span style="float:left; padding-
  right:15px">
  <img src="images/restricted.png"
  class="thumbnail" />
    </span>
    <span class="font-bold">Image
    Name:</span>
    <span class="font-blue"><c:out
    value="${detail.imagePath}"
    /></span><br/>
  </c:otherwise>
</c:choose>
  <span class="font-
  bold">Category:</span> <c:out
  value="${detail.category}" /><br/>
  <span class="font-
  bold">Accessibility:</span> <c:out
  value="${detail.accessibility}"
  /><br/>
  <span class="font-
  bold">Description:</span> <c:out
  value="${detail.description}"
  /><br/>
  <span class="font-bold">Image
  Set:</span> <c:out
  value="${detail.imageSet}" />
  </td>
</tr>
</c:forEach>
</c:when>
<c:otherwise>
<tr>
  <td class="font-italic">No results
  related to key term <b><c:out
  value="${SearchForm.searchKey}"
  /></b> were found in
  VirhoImages</td>
</tr>
</c:otherwise>
</c:choose>
<tr>
  <td> </td>
</tr>
</c:if>
<c:if test="${(SearchForm.module eq
'VirhoModels') or (SearchForm.module
eq 'All Modules')}">
<tr>
  <th align="left">VirhoModels</th>
</tr>
<c:choose>
<c:when test="${not empty
SearchForm.models}">
<tr style="background-
color:#E1E8F1;">
  <td>
  Sort results by :
  <html:select property="modSortBy"
  onchange="sortBy()">
    <html:option
    value="model">Model</html:option>
    <html:option
    value="author">Author</html:optio
    n>
    <html:option
    value="project">Project</html:opt
    ion>
    <html:option
    value="classification">Classifica
    tion</html:option>
  </html:select>
  </td>
</tr>
<tr>
  <td class="font-italic">Search
  results related to key term
  <b><c:out
  value="${SearchForm.searchKey}"
  /></b> in VirhoModels</td>
</tr>
```

```
<c:forEach var="detail"
items="${SearchForm.models}">
  <tr>
    <td>
      <span class="font-
      bold">Model:</span>
      <html:link
      page="/ViewModelDetails.do?module
      =VirhoModels&classification=${det
      ail.classification}&project=${det
      ail.projectName}&model=${detail.m
      odelName}">
      <span class="font-blue"><c:out
      value="${detail.modelName}"
      /></span>
      </html:link><br/>
      <span class="font-
      bold">Author:</span> <c:out
      value="${detail.author}" /><br/>
      <span class="font-
      bold">Project:</span> <c:out
      value="${detail.projectName}"
      /><br/>
      <span class="font-
      bold">Classification:</span>
      <c:out
      value="${detail.classification}"
      />
    </td>
  </tr>
</c:forEach>
  </c:when>
  <c:otherwise>
  <tr>
    <td class="font-italic">No results
    related to key term <b><c:out
    value="${SearchForm.searchKey}"
    /></b> were found in
    VirhoModels</td>
  </tr>
  </c:otherwise>
</c:choose>
  <tr>
    <td> </td>
  </tr>
</c:if>
<c:if test="${(SearchForm.module eq
'VirhoReferences') or
(SearchForm.module eq 'All
Modules')}">
  <tr>
    <th
    align="left">VirhoReferences</th>
  </tr>
<c:choose>
  <c:when test="${not empty
  SearchForm.references}">
  <tr style="background-
  color:#E1E8F1;">
    <td>
    Sort results by :
    <html:select property="refSortBy"
    onchange="sortBy()">
      <html:option
      value="reference">Reference</html
      :option>
      <html:option
      value="author">Author</html:optio
      n>
      <html:option
      value="collection">Collection</ht
      ml:option>
    </html:select>
    </td>
  </tr>
  <tr>
      <td class="font-italic">Search
      results related to key term
      <b><c:out
      value="${SearchForm.searchKey}"
      /></b> in VirhoReferences</td>
    </tr>
  <c:forEach var="detail"
  items="${SearchForm.references}">
    <tr>
      <td>
        <span class="font-
        bold">Reference:</span>
        <html:link
        page="/ViewReferenceDetails.do?mo
        dule=VirhoReferences&collection=$
        {detail.collection}&reference=${d
        etail.title}&refId=${detail.refer
        enceId}">
        <span class="font-blue"><c:out
        value="${detail.title}" /></span>
        </html:link><br/>
        <span class="font-
        bold">Author:</span> <c:out
        value="${detail.author}" /><br/>
        <span class="font-
        bold">Collection:</span> <c:out
        value="${detail.collection}" />
      </td>
    </tr>
  </c:forEach>
    </c:when>
    <c:otherwise>
    <tr>
      <td class="font-italic">No results
      related to key term <b><c:out
      value="${SearchForm.searchKey}"
      /></b> were found in
      VirhoReferences</td>
    </tr>
    </c:otherwise>
  </c:choose>
    <tr>
      <td> </td>
    </tr>
  </c:if>
  </table>
  </td>
 </tr>
 <tr>
  <td> </td>
 </tr>
</table>
</c:if>
</td>
</tr>
<tr>
  <td> </td>
</tr>
<tr class="footer">
  <td colspan="2">
    &copy; <bean:message
    key="virho.copyright" />
  </td>
</tr>
</table>
</html:form>
</body>
</html>
```

viewUsers.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
```

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt"
prefix="fmt"%>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
  <html:form action="/VirhoBasicInfoUsers"
  method="POST">
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <c:if test="${(userLogin ne null) and
      ((statusLogin eq 'activated') or
      (statusLogin eq 'admin') or (statusLogin
      eq 'superadmin'))}">
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <html:link
            page="/ViewVirus.do?list=" >Virus
            List</html:link> &#187;
            <html:link
            page="/ViewVirusDetails.do?virus=${v
            irus}" >${virus}</html:link> &#187;
            <span class="breadcrumb-
            selected">View Members</span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2">List of Hotspot
          Members</td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr>
              <td>
              <table class="letterList">
                <tr>
                  <td>
                  <c:forEach var="letter"
                  items="A,B,C,D,E,F,G,H,I,J,K,L,
                  M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
                  varStatus="loop">
                  <c:choose>
                    <c:when
                    test="${ViewUsersForm.letters[l
                    oop.index] eq letter}">
                      <html:link
                      page="/VirhoHotspotsUsers.do?l
                      ist=${letter}"><span
                      class="font-
                      letter">${letter}</span></html
                      :link> 
                    </c:when>
                    <c:otherwise>
                      ${letter} 
                    </c:otherwise>
                  </c:choose>
                    </c:forEach>
                    <html:link
                    page="/VirhoHotspotsUsers.do?li
                    st=">View All</html:link>
                    </td>
                  </tr>
                </table>
                </td>
              </tr>
              <tr>
                <td>
                <table class="table-content">
                  <tr>
                    <th>Name</th>
                    <th>Username</th>
                    <th>Affiliation</th>
                    <th>Role</th>
                  </tr>
                <c:choose>
                  <c:when
                  test="${fn:length(ViewUsersForm.u
                  sers) gt 0}">
                <c:forEach var="user"
                items="${ViewUsersForm.users}">
                  <tr>
                    <td>
                  <c:choose>
                    <c:when
                    test="${requestScope.privilege
                    eq 4}" >
                      <html:link
                      page="/UserProfileDetails.do?u
                      ser=${user.username}" >
                      <span class="font-
                      blue">${user.firstName} $
                      {user.lastName}</span>
                      </html:link>
                    </c:when>
                    <c:otherwise>
                      ${user.firstName} ${user.
                      lastName}
                    </c:otherwise>
                  </c:choose>
                    </td>
                    <td>${user.username}</td>
                    <td>${user.affiliation}</td>
                    <td>
                  <c:choose>
                    <c:when
                    test="${(requestScope.privilege
                    eq 4) and (user.username ne
                    userLogin)}">
                    <select name="role">
                      <option value="Remove
                      Privilege">Remove
                      Privilege</option>
                    <c:forEach var="roleName"
                    items="${requestScope.roles}">
```

```
                <option value="${roleName}"
                <c:if test="${roleName eq
                user.projectRole}">selected="s
                elected"</c:if>>${roleName}</o
                ption>
              </c:forEach>
              </select>
              <input type="hidden"
              name="oldRole"
              value="${user.projectRole}" />
              <input type="hidden"
              name="userName"
              value="${user.username}" />
              </c:when>
              <c:otherwise>
                ${user.projectRole}
              </c:otherwise>
            </c:choose>
              </td>
            </tr>
          </c:forEach>
            </c:when>
            <c:otherwise>
            <tr>
              <td class="font-italic"
              colspan="4">
                There are no members yet
              </td>
            </tr>
            </c:otherwise>
          </c:choose>
          </table>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td>
          
      <c:if test="${requestScope.privilege
      eq 4}" >
        <html:submit property="command"
        value="Change Privilege Role"
        styleClass="button-superlong-blue"
        />
      </c:if>
      </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
  </table>
  </c:if>
  </td>
 </tr>
 <tr>
   <td> </td>
 </tr>
 <tr class="footer">
   <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
   </td>
 </tr>
</table>
</html:form>
</body>
</html>
```

viewVirusDetails.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="java" content="text/html;
  charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  uery/1.3.2/jquery.min.js"
  type="text/javascript"></script>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jq
  ueryui/1.7.2/jquery-ui.min.js"></script>
  <link rel="stylesheet" href="./files/jquery-
  ui.css" type="text/css" />
  <script src="./files/jquery.easy-confirm-
  dialog.js"></script>

  <style type="text/css">

  div.photo-container {
    border: 1px solid #333333 ;
  }

  a.activated {
    text-decoration:none;
    font-size: 10px;
    font-weight: bold;
    position: relative;
  }

  a.deactivated {
    text-decoration:none;
    font-size: 10px;
    position: relative;
  }

  </style>

  <script type="text/javascript"
  src="//ajax.googleapis.com/ajax/libs/jquery/
  1.5.1/jquery.min.js"></script>
  <script type="text/javascript"
  src="./files/phototagger.jquery.js"></script
  >

  <script type="text/javascript">

  var $jq151 = jQuery.noConflict(true);
  // When the DOM is ready, initialize the
  scripts.
  $jq151(function( $jq151 ){
    // Set up the photo tagger.
    $jq151( "div.photo-container"
    ).photoTagger({
      // The API urls.
      loadURL: "FetchHotspots",
      saveURL: "AddHotspot",
      deleteURL: "DeleteHotspot"
    });
```

```
// Hook up the enable create links.
$jq151( "a.enable-create" ).click(
  function( event ){
    // Prevent relocation.
    event.preventDefault();

    $jq151( this
    ).removeClass('deactivated').addClass('
    activated');
    $jq151( "a.disable-create"
    ).removeClass('activated').addClass('de
    activated');

    // Get the container and enable the tag
    creation on it.
    $jq151( this ).prevAll( "div.photo-
    container" ).photoTagger(
    "enableTagCreation" );
  }
);

// Hook up the disabled create links.
$jq151( "a.disable-create" ).click(
  function( event ){
    // Prevent relocation.
    event.preventDefault();

    $jq151( this
    ).removeClass('deactivated').addClass('
    activated');
    $jq151( "a.enable-create"
    ).removeClass('activated').addClass('de
    activated');

    // Get the container and enable the tag
    creation on it.
    $jq151( this ).prevAll( "div.photo-
    container" ).photoTagger(
    "disableTagCreation" );
  }
);

// Hook up the enable delete links.
$jq151( "a.enable-delete" ).click(
  function( event ){
    // Prevent relocation.
    event.preventDefault();

    $jq151( this
    ).removeClass('deactivated').addClass('
    activated');
    $jq151( "a.disable-delete"
    ).removeClass('activated').addClass('de
    activated');

    // Get the container and enable the tag
    deletion on it.
    $jq151( this ).prevAll( "div.photo-
    container" ).photoTagger(
    "enableTagDeletion" );
  }
);

// Hook up the disabled delete links.
$jq151( "a.disable-delete" ).click(
  function( event ){
    // Prevent relocation.
    event.preventDefault();

    $jq151( this
    ).removeClass('deactivated').addClass('
    activated');
    $jq151( "a.enable-delete"
    ).removeClass('activated').addClass('de
    activated');

    // Get the container and disabled the
    tag deletion on it.
    $jq151( this ).prevAll( "div.photo-
    container" ).photoTagger(
    "disableTagDeletion" );
   }
  );
 });

 jQuery(document).ready(function() {
  $(".confirm").easyconfirm();
  $("#image").click(function() {
   return true;
  });
  $("#virus").click(function() {
   return true;
  });
 });

 </script>
</head>
<body>
 <html:form action="/ViewVirusDetails"
 styleId="form" method="POST"
 enctype="multipart/form-data">
 <table class="body" align="center">
 <jsp:include page="/rd_virui/header.jsp" />
  <tr>
   <td> </td>
  </tr>
  <tr valign="top">
   <td class="sidebar">
    <jsp:include
    page="/rd_virui/sidebar.jsp" />
   </td>
   <td>
   <table class="content">
    <tr>
     <td class="breadcrumb">
      <html:link
      page="/Home.do"><bean:message
      key="button.virho.home"
      /></html:link> &#187;
      <html:link
      page="/ViewVirus.do?list=" >Virus
      List</html:link> &#187;
      <span class="breadcrumb-
      selected">${virus}</span>
     </td>
    </tr>
    <tr>
     <td> </td>
    </tr>
    <tr>
     <td class="content-title-form"
     colspan="2">Virus Information</td>
    </tr>
    <tr>
     <td>
     <table class="content-desc-form">
      <tr>
       <td width="350px" valign="top">
       <table class="font-medium-reg">
        <tr>
         <td>
          <b>Common Name</b> :
          <bean:write name="VirusForm"
          property="commonName"/>
         </td>
        <tr>
         <td>
          <b>Scientific Name</b> :
          <bean:write name="VirusForm"
          property="scientificName"/>
```

```
    <c:if test="${empty
    fn:trim(VirusForm.scientificNa
    me)}">-</c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Serotype</b> : <bean:write
    name="VirusForm"
    property="serotype"/>
    <c:if test="${empty
    fn:trim(VirusForm.serotype)}">
    -</c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Host</b> : <bean:write
    name="VirusForm"
    property="host"/>
    <c:if test="${empty
    fn:trim(VirusForm.host)}">-
    </c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Virus Architecture</b> :
    <bean:write name="VirusForm"
    property="virusArchitecture"/>
    <c:if test="${empty
    fn:trim(VirusForm.virusArchite
    cture)}">-</c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Infectivity</b> :
    <bean:write name="VirusForm"
    property="infectivity"/>
    <c:if test="${empty
    fn:trim(VirusForm.infectivity)
    }">-</c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Manner of Infection</b> :
    <bean:write name="VirusForm"
    property="mannerOfInfection"/>
    <c:if test="${empty
    fn:trim(VirusForm.mannerOfInfe
    ction)}">-</c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Vector</b> : <bean:write
    name="VirusForm"
    property="vector"/>
    <c:if test="${empty
    fn:trim(VirusForm.vector)}">-
    </c:if>
   </td>
  </tr>
  <tr>
   <td>
    <b>Antimicrobial Treatment</b>
    : <bean:write name="VirusForm"
    property="antimicrobialTreatme
    nt"/>
    <c:if test="${empty
    fn:trim(VirusForm.antimicrobia
    lTreatment)}">-</c:if>
   </td>
  </tr>
    <tr>
     <td>
      <b>Receptors</b> : <bean:write
      name="VirusForm"
      property="receptors"/>
      <c:if test="${empty
      fn:trim(VirusForm.receptors)}"
      >-</c:if>
     </td>
    </tr>
   </table>
   </td>
  <c:choose>
   <c:when test="${VirusForm.hasImage
   eq true}">
   <td valign="top">
   <table class="upload-container">
    <tr>
     <td>
      <div class="photo-container">
      <img id="${result}"
      src="./repository/virhohotspot
      s/${result}" />
      </div>
      <c:if test="${(privilege eq 4)
      and (userLogin ne null) and
      ((statusLogin eq 'activated')
      or (statusLogin eq 'admin') or
      (statusLogin eq
      'superadmin'))}">
       <!-- These will toggle the
       tag creation. -->
       <a href="#" class="enable-
       create deactivated">Enable
       Create</a><span class="font-
       gray-small"> |</span>
       <a href="#" class="disable-
       create activated">Disable
       Create</a><span class="font-
       gray-small"> |</span>
       <!-- These will toggle the
       tag deletion. -->
       <a href="#" class="enable-
       delete deactivated">Enable
       Delete</a><span class="font-
       gray-small"> |</span>
       <a href="#" class="disable-
       delete activated">Disable
       Delete</a>
      </c:if>
     </td>
    </tr>
   </table>
   </td>
   </c:when>
   <c:otherwise>
  <c:if test="${(userLogin ne null)
  and ((statusLogin eq 'activated') or
  (statusLogin eq 'admin') or
  (statusLogin eq 'superadmin'))}">
   <c:if test="${privilege eq 4}">
   <td valign="top">
   <table class="upload-container">
    <tr>
     <td class="font-bold">DOWNLOAD
     DIAGRAM</td>
    </tr>
    <tr>
     <td class="note">
      <p>Note: The preferred width
      size for the image is <b>400
      pixels or less</b>. Image that
      exceeds the given size will be
      automatically resized.</p>
```

```
            <p><i>The bigger the image,
            the lesser the quality it
            gets.</i></p>
            Acceptable image file
            extensions : jpg, jpeg, gif,
            png, bmp</td>
          </tr>
          <tr>
            <td class="font-
            bold"><br/>Choose Image
            File</td>
          </tr>
          <tr>
            <td>
              <html:file property="file"
              style="remove-error-field-
              file" errorStyleClass="error-
              field"
              errorKey="org.apache.struts.ac
              tion.ERROR" />
              <br/><span class="error-
              text"><html:errors
              property="file" /></span>
            </td>
          </tr>
          <tr>
            <td align="right"><html:submit
            property="command"
            value="Upload"
            styleClass="button-long-gray"
            /></td>
          </tr>
        </table>
        </td>
        </c:if>
      </c:if>
        </c:otherwise>
      </c:choose>
      </tr>
    <c:if test="${(userLogin ne null) and
    ((statusLogin eq 'activated') or
    (statusLogin eq 'admin') or
    (statusLogin eq 'superadmin'))}">
      <c:if test="${(privilege eq 4) and
      (VirusForm.hasImage eq true)}">
      <tr>
        <td colspan="2" class="note"></td>
      </tr>
      <tr>
        <td colspan="2" class="note">
          <b>How to use hotspots</b>
          <ul>
          <li>To create hotspot, draw a
          rectangle on the picture through
          click and drag.</li>
          <li>To delete hotspot, double
          click on the hotspot
          created.</li>
          </ul>
          Note: Disable first the Create
          Hotspot before deleting a
          hotspot. Avoid refreshing the
          page when you're using Chrome.
          <p>Click <html:submit
          property="command" value="here"
          styleId="image"
          styleClass="button-small confirm"
          style="color:#0000FF" /> if you
          want to delete the image.</p>
        </td>
      </tr>
      </c:if>
    </c:if>
    </table>
  </td>
</tr>
```

```
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td>    
          <c:if test="${(userLogin ne null) and
          ((statusLogin eq 'activated') or
          (statusLogin eq 'admin') or
          (statusLogin eq 'superadmin'))}">
          <c:choose>
            <c:when test="${privilege eq 4}" >
              <html:submit property="command"
              value="Edit Virus"
              styleClass="button-long-blue" />
              <html:submit property="command"
              value="Delete Virus" styleId="virus"
              styleClass="button-long-blue
              confirm" />
              <html:submit property="command"
              value="View Members"
              styleClass="button-long-blue" />
            </c:when>
            <c:otherwise>
              <html:submit property="command"
              value="Send Privilege Request"
              styleClass="button-superlong-blue"
              />
            </c:otherwise>
          </c:choose>
            </c:if>
            </td>
          </tr>
        </table>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr class="footer">
        <td colspan="2">
          &copy; <bean:message
          key="virho.copyright" />
        </td>
      </tr>
    </table>
    </html:form>
</body>
</html>
```

viewVirusList.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/functions"
prefix="fn" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
<body>
```

```
<html:form action="/ViewVirus"
method="POST">
<table class="body" align="center">
<jsp:include page="/rd_virui/header.jsp" />
  <tr>
    <td> </td>
  </tr>
  <tr class="content" valign="top">
    <td class="sidebar">
      <jsp:include
      page="/rd_virui/sidebar.jsp" />
    </td>
    <td>
    <table class="content">
      <tr>
        <td class="breadcrumb">
          <html:link
          page="/Home.do"><bean:message
          key="button.virho.home"
          /></html:link> &#187;
          <span class="breadcrumb-
          selected">Virus List</span>
        </td>
      </tr>
      <tr>
        <td> </td>
      </tr>
      <tr>
        <td class="content-title-form"
        colspan="2">Virus List</td>
      </tr>
      <tr>
        <td>
        <table class="content-desc-form">
          <tr>
            <td>
            <table class="letterList">
              <tr>
                <td>
                <c:forEach var="letter"
                items="A,B,C,D,E,F,G,H,I,J,K,L,
                M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"
                varStatus="loop">
              <c:choose>
                <c:when
                test="${letters[loop.index] eq
                letter}">
                  <html:link
                  page="/ViewVirus.do?list=${let
                  ter}"><span class="font-
                  letter">${letter}</span></html
                  :link> 
                </c:when>
                <c:otherwise>
                  ${letter} 
                </c:otherwise>
              </c:choose>
                </c:forEach>
                <html:link
                page="/ViewVirus.do?list=">View
                All</html:link>
                </td>
              </tr>
            </table>
            </td>
          </tr>
          <tr>
            <td>
            <table class="table-content">
              <tr>
                <th>Virus Name</th>
                <th>Description</th>
              </tr>
            <c:choose>
              <c:when test="${!empty virus}">
                <c:forEach var="detail"
                items="${virus}">
                <tr>
                  <td width="200px" valign="top">
                    <html:link
                    page="/ViewVirusDetails.do?vir
                    us=${detail.virusName}"><span
                    class="font-
                    blue">${detail.virusName}</spa
                    n></html:link>
                  </td>
                  <td>${detail.description} 
                  </td>
                </tr>
                </c:forEach>
                </c:when>
                <c:otherwise>
                <tr>
                  <td class="font-italic"
                  colspan="2">No Virus Yet</td>
                </tr>
                </c:otherwise>
              </c:choose>
              </table>
              </td>
            </tr>
          </table>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td>    
          <c:if test="${(userLogin ne null) and
          (privilege eq true)}" >
            <html:submit property="command"
            value="Add Virus"
            styleClass="button-long-blue" />
          </c:if>
          </td>
        </tr>
      </table>
      </td>
    </tr>
    <tr class="footer">
      <td colspan="2">
        &copy; <bean:message
        key="virho.copyright" />
      </td>
    </tr>
  </table>
  </html:form>
</body>
</html>
```

virhoExternalSearch.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
```

```
<link type="text/css" href="./files/jquery-
ui-1.8.23.custom.css" rel="stylesheet" />
<script src="./files/jquery-
1.8.0.js"></script>

<script type="text/javascript">
function createQuery(){
  var db =
  document.getElementById("db").value;
  var key =
  document.getElementById("key").value;
  var base =
  document.getElementById("base").value;

  if(key==""){
    alert("Please supply a search term.");
  } else {
    window.open(base+"&db="+db+"&term="+key,
    "new_window");
  }
}

function createQuery2(){
  var db =
  document.getElementById("db2").value;
  var key =
  document.getElementById("key2").value;
  var base =
  document.getElementById("base2").value;

  if(key==""){
    alert("Please supply a search term.");
  } else {
    window.open(base+db+"&keywords="+key+"&pa
    ge=1", "new_window");
  }
}

function createQuery3(){
  var db =
  document.getElementById("db3").value;
  var key =
  document.getElementById("key3").value;
  var base =
  document.getElementById("base3").value;

  if(key==""){
    alert("Please supply a search term.");
  } else {
    window.open(base+db+"&t="+key,
    "new_window");
  }
}

function loadAlt(){
  window.open
  ('AlternativeExternalSearch.do', '_self',
  false);
}

var ids = new Array();
ids[0] = "NCBI";
ids[1] = "KEGG";
ids[2] = "EBI";

function selectSearchPanel(tabIndex){
  for (var i=0;i<ids.length;i++){
    document.getElementById(ids[i]+"Search").
    style.display = 'none';
    document.getElementById(ids[i]).style.bac
    kgroundColor = '#FFFFFF';
  }
  document.getElementById(ids[tabIndex]+"Sear
  ch").style.display = 'block';
  document.getElementById(ids[tabIndex]).styl
  e.backgroundColor = '#DDDDFF';
  }
  </script>
</head>
<body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
    <tr>
      <td> </td>
    </tr>
    <tr valign="top">
      <td class="sidebar">
        <jsp:include
        page="/rd_virui/sidebar.jsp" />
      </td>
      <td>
      <table class="content">
        <tr>
          <td class="breadcrumb">
            <html:link
            page="/Home.do"><bean:message
            key="button.virho.home"
            /></html:link> &#187;
            <span class="breadcrumb-
            selected"><bean:message
            key="virho.externalSearch" /></span>
          </td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td class="content-title-form"
          colspan="2"><bean:message
          key="virho.externalSearch" /></td>
        </tr>
        <tr>
          <td>
          <table class="content-desc-form">
            <tr align="center">
              <td id="NCBI"
              bgcolor="#DDDDFF"><div
              onclick="selectSearchPanel(0);"
              class="button-
              search">NCBI</div></td>
              <td id="KEGG"
              bgcolor="#FFFFFF"><div
              onclick="selectSearchPanel(1);"
              class="button-
              search">KEGG</div></td>
              <td id="EBI" bgcolor="#FFFFFF"><div
              onclick="selectSearchPanel(2);"
              class="button-
              search">EBI</div></td>
            </tr>
            <tr>
              <td colspan="3">
              <div id="NCBISearch"
              style="display:block">
              <table align="center"
              class="content-search">
                <tr>
                  <td>Select Entrez Database</td>
                  <td>
                  <select id="db">
                    <option value="domains">3d
                    Domains </option>
                    <option value="cdd">Domains
                    </option>
                    <option value="genome">Genome
                    </option>
                    <option
                    value="nucleotide">Nucleotide
                    </option>
                    <option value="omim">OMIM
                    </option>
```

```
      <option value="popset">PopSet
      </option>
      <option
      value="protein">Protein
      </option>
      <option value="geo">ProbeSet
      </option>
      <option value="pubmed">PubMed
      </option>
      <option
      value="structure">Structure
      </option>
      <option value="snp">SNP
      </option>
      <option
      value="taxonomy">Taxonomy
      </option>
      <option
      value="unigene">UniGene
      </option>
      <option value="unists">UniSTS
      </option>
    </select>
    </td>
  </tr>
  <tr>
    <td>Type in search term </td>
    <td><input type="text"
    size="30" id="key"></td>
  </tr>
  <tr>
    <td colspan=2 align="center">
      <input type="submit"
      value="Perform Search"
      onClick="createQuery()"
      class="button-superlong-blue">
      <input type="submit"
      value="Alternative Search"
      onClick="loadAlt()"
      class="button-superlong-blue">
    </td>
  </tr>
</table>
<input type="hidden"
value="<bean:message
key="link.search.ncbi" />"
id="base">
</div>

<div id="KEGGSearch"
style="display:none">
<table align="center"
class="content-search">
  <tr>
    <td>Select KEGG Database</td>
    <td>
    <select id="db2">
      <option value="kegg">KEGG
      </option>
      <option
      value="pathway">PATHWAY
      </option>
      <option value="brite">BRITE
      </option>
      <option value="module">MODULE
      </option>
      <option
      value="disease">DISEASE
      </option>
      <option
      value="orthology">ORTHOLOGY
      </option>
      <option value="genes">GENES
      </option>
      <option
      value="dgenes">DGENES</option>
      <option value="egenes">EGENES
      </option>
      <option value="genome">GENOME
      </option>
      <option
      value="compound">COMPOUND
      </option>
      <option value="drug">DRUG
      </option>
      <option value="glycan">GLYCAN
      </option>
      <option
      value="reaction">REACTION
      </option>
      <option value="rpair">RPAIR
      </option>
      <option value="enzyme">ENZYME
      </option>
    </select>
    </td>
  </tr>
  <tr>
    <td>Type in search term </td>
    <td><input type="text"
    size="30" id="key2"></td>
  </tr>
  <tr>
    <td colspan=2 align="center">
    <input type="submit"
    value="Perform Search"
    onClick="createQuery2()"
    class="button-superlong-blue">
    </td>
  </tr>
</table>
<input type="hidden"
value="<bean:message
key="link.search.kegg" />"
id="base2">
</div>

<div id="EBISearch"
style="display:none">
<table align="center"
class="content-search">
  <tr>
    <td>Select EBI Database </td>
    <td>
    <select id="db3">
      <option
      value="genomes">Genomes
      </option>
      <option
      value="nucleotideSequences">Nu
      cleotide Sequences </option>
      <option
      value="proteinSequences">Prote
      in Sequences </option>
      <option
      value="macromolecularStructure
      s">Macromolecular Structures
      </option>
      <option
      value="smallMolecules">Small
      molecules </option>
      <option
      value="geneExpression">Gene
      Expression </option>
      <option
      value="molecularInteractions">
      Molecular Interactions
      </option>
      <option
      value="reactionsPathways">Reac
      tions &amp;amp; Pathways</option>
```

```
                <option
                value="proteinFamilies">Protei
                n Families </option>
                <option
                value="enzymes">Enzymes
                </option>
                <option
                value="literature">Literature
                </option>
                <option
                value="ontologies">Ontologies
                </option>
                <option value="ebiweb">EBI Web
                Site </option>
              </select>
              </td>
            </tr>
            <tr>
              <td>Type in search term </td>
              <td><input type="text"
              size="30" id="key3"></td>
            </tr>
            <tr>
              <td colspan=2 align="center">
                <input type="submit"
                value="Perform Search"
                onClick="createQuery3()"
                class="button-superlong-blue">
              </td>
            </tr>
          </table>
          <input type="hidden"
          value="<bean:message
          key="link.search.ebi" />"
          id="base3">
          </div>
          </td>
        </tr>
      </table>
      </td>
    </tr>
  </table>
  </td>
 </tr>
 <tr class="footer">
   <td colspan="2">
     &copy; <bean:message
     key="virho.copyright" />
   </td>
 </tr>
</table>
</body>
</html>
```

GENERAL FILES

errorPage.jsp

```
<%@ taglib uri="/WEB-INF/struts-bean.tld"
prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld"
prefix="html" %>
<%@ taglib
uri="http://java.sun.com/jsp/jstl/core"
prefix="c" %>

<html>
<head>
  <meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
  <title>Virus-Host Interaction
  Lexicon</title>
  <link rel="shortcut icon"
  href="./images/favicon.ico" >
  <link rel="stylesheet" type="text/css"
  href="./files/custom.css" media="screen"/>
</head>
  <body>
  <table class="body" align="center">
  <jsp:include page="/rd_virui/header.jsp" />
   <tr>
     <td> </td>
   </tr>
   <tr valign="top">
     <td class="sidebar">
       <jsp:include
       page="/rd_virui/sidebar.jsp" />
     </td>
     <td>
     <table class="content">
       <tr>
         <td class="breadcrumb">
           <html:link
           page="/Home.do"><bean:message
           key="button.virho.home"
           /></html:link>
         </td>
       </tr>
       <tr>
         <td> </td>
       </tr>
       <c:if test="${(userLogin ne null) and
       ((statusLogin eq 'activated') or
       (statusLogin eq 'admin') or
       (statusLogin eq 'superadmin'))}">
       <tr>
         <td class="content-title">System
         Announcement</td>
       </tr>
       <tr>
         <td> </td>
       </tr>
       <tr>
         <td class="content-desc">
           Problem existed while processing
           your request. Please try again
           later.
           <br/><br/>
           <html:link
           page="/ContactUs.do">Contact
           Us</html:link> for further
           inquiries.
         </td>
       </tr>
       </c:if>
     </table>
     </td>
   </tr>
   <tr class="footer">
     <td colspan="2">
       &copy; <bean:message
       key="virho.copyright" />
     </td>
   </tr>
  </table>
</body>
</html>
```

D. css and js files

custom.css

```
a:hover {
  color:#FF0000;
}

.body {
  width: 1150px;
  box-shadow: 0px -45px 100px 5px #DDDDDD;
}

.banner {
  text-align: center;
}

.login-index {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 11px;
  color: #FFFFFF;
  text-align: right;
  background-color: 001C6B;
  height: 25px;
}

.login-index a {
  color: #FFFFFF;
  text-decoration: none;
}

.welcome-note {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  border-bottom: 2px dashed #295991;
}

.welcome-title {
  font-size: 16px;
  font-weight: bold;
  font-style: italic;
  color: #003274;
  padding-bottom: 10px;
}

.welcome-body {
  font-size: 10px;
  color: #777777;
  padding-bottom: 10px;
}

.welcome-body a {
  font-size: 10px;
  color: #777777;
  font-style: italic;
}

.sidebar {
  width: 240px;
  padding: 0px 15px 15px 15px;
}

.sidetabs {
  width: 100%;
}

.sidetabs td {
  font: 14px Verdana, Geneva, Arial,
Helvetica, sans-serif;
  padding: 10px 30px 10px 30px;
  border-bottom: 1px dashed #295991;
}

.sidetabs a {
  text-decoration: none;
  color: #003274;
}

.sidetabs td:hover {
  font: 14px Verdana, Geneva, Arial,
Helvetica, sans-serif;
  padding: 10px 30px 10px 30px;
  border-bottom: 1px dashed #295991;
  background-color: #C0DDFF;
  cursor: pointer;
}

.updatesbar {
  width: 100%;
}

.updatesbar table {
  border: 1px solid #EEEEEE;
}

.updatesbar table td {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #023E60;
  font-size: 11px;
  padding: 5px 15px;
  border-bottom: 1px solid #EEEEEE;
}

.tab-updates {
  background: white url("../images/tab-
blue.gif") no-repeat bottom left;
  font: bold 14px Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #FFFFFF;
  text-align: center;
  padding-top: 10px;
  height: 50px;
}

.tab-updates-inv {
  background: white url("../images/tab-blue-
inv.gif") no-repeat top left;
  height: 20px;
}

.record:hover {
  border-bottom: 1px dashed #295991;
  background-color: #C0DDFF;
  cursor: pointer;
}

.content {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  margin-left: 25px;
  margin-right: 30px;
  width: 820px;
}

.search-container {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
  font-weight: bold;
  color: #555555;
  background: white url("../images/search-
bg.gif") no-repeat top left;
  height: 55px;
}

.image-container {
  height: 500px;
```

```
  width: 100%;
  border: 1px dotted #BBBBBB;
  background-color: #FFFFFF;
}

.content-title {
  font-size: 16px;
  font-weight: bold;
  color: #103900;
}

.content-title-form {
  font-size: 16px;
  font-weight: bold;
  color: #FFFFFF;
  padding-left: 30px;
  background: white url("../images/title-
blue.gif") no-repeat top center;
  height: 43px;
}

.content-desc {
  font-size: 14px;
  color: #434343;
}

.content-desc a {
  font-size: 14px;
  color: #2233FF;
  text-decoration: none;
}

.content-desc-form {
  background: white url("../images/white-
bg.jpg") repeat top center;
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
  width: 820px;
  padding: 15px 30px;
  border-bottom: 1px solid #CCCCCC;
}

.content-desc-form td {
  padding: 8px 0px;
}

.content-desc-form a {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #2233FF;
  font-size: 12px;
  text-decoration: none;
}

.description-container {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
  width: 100%;
  background-color: #EFEFEF;
  border: 1px solid #DDDDDD;
  padding: 0px 10px;
}

.upload-container {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
  margin-left: 20px;
  padding: 20px;
  background-color: #EEEEEE;
  border: 1px #DDDDDD solid;
  width: 400px;
}

.table-content {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #434343;
  width: 100%;
}

.table-content td {
  font-size: 12px;
  color: #212343;
  padding: 7px 10px;
  border-bottom: 1px dotted #DDDDFF;
}

.table-content td a {
  font-size: 12px;
  color: #212343;
  text-decoration: none;
}

.table-content th {
  font-size: 13px;
  padding: 7px;
  background-color: #ADC7E6;
}

.note {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #666666;
  border-bottom: 1px ridge #DDDDDD;
}

.note a {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #2233FF;
  text-decoration: none;
}

.breadcrumb {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  color: #888888;
}

.breadcrumb a {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  color: #295991;
  text-decoration: none;
}

.breadcrumb-selected {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  color: #888888;
}

.login {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 14px;
  padding: 10px 30px;
}

.letterList {
  font-size: 15px;
  text-align: center;
  color: #1947D1;
  width: 100%;
  border-bottom: 2px solid #DDDDDD;
  padding: 3px 0px;
}
```

```
.letterList a {
  font-size: 15px;
  color: #2959CD;
  text-decoration: none;
}

.footer {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 11px;
  color: #666666;
  text-align: center;
  background: white url("../images/footer-
bg.gif") no-repeat top left;
  height: 50px;
}

/*   ERROR STYLES    */
.error-field {
  background-color: #FAFEB8;
}

.error-text {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  color: #FF0000;
}

.remove-error-field {
  background-color: #FFFFFF;
}

.remove-error-field-file {
  background-color: transparent;
}

.error-text-big {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
  font-weight: bold;
  color: #FF0000;
}

/*       GENRERAL        */
.button {
  background: white url("../images/button-
blue.gif") repeat top center;
  height: 25px;
  width: 85px;
  cursor: pointer;
}

.button-long {
  background: white url("../images/button-
blue.gif") repeat top center;
  height: 25px;
  width: 125px;
  cursor: pointer;
}

.button-superlong {
  background: white url("../images/button-
blue.gif") repeat top center;
  height: 25px;
  width: 155px;
  cursor: pointer;
}

.button-long-white {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  background-color: white;
  height: 20px;
  width: 120px;
  cursor: pointer;
}

.button-superlong-white {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  background-color: white;
  height: 20px;
  width: 150px;
  cursor: pointer;
}

.button-superlong-blue {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 11px;
  background: white url("../images/button-
blue.gif") repeat top center;
  height: 25px;
  width: 200px;
  cursor: pointer;
}

.button-long-blue {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 11px;
  background: white url("../images/button-
blue.gif") repeat top center;
  height: 25px;
  width: 130px;
  cursor: pointer;
}

.button-long-gray {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  background-color: #CCCCCC;
  height: 20px;
  width: 100px;
  cursor: pointer;
}

.button-superlong-gray {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  background-color: #CCCCCC;
  height: 20px;
  width: 150px;
  cursor: pointer;
}

.button-search {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #434343;
  font-size: 14px;
  font-weight: bold;
  cursor: pointer;
  padding: 5px;
}

.button-remove {
  font-family: Arial, Helvetica, sans-serif;
  color: #676778;
  font-size: 9px;
  cursor: pointer;
  border: 1px solid #EDEDED;
}

.button-small {
```

```
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #000000;
  font-size: 12px;
  cursor: pointer;
  border: 0px;
  background-color: transparent;
}

.content-search {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #434343;
  font-size: 14px;
  padding: 15px;
}

.content-search td {
  padding: 15px;
}

.font-medium-reg {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 12px;
}

.font-bold {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-weight: bold;
}

.font-italic {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-style: italic;
}

.font-green {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #137A23;
}

.font-red {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #FF0000;
}

.font-blue {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #0033CC;
  text-decoration: none;
}

.font-blue:hover {
  color:#FF0000;
}

.font-darkgray {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #6C7967;
  text-decoration: none;
}

.font-letter {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #1947D1;
  cursor: pointer;
  text-decoration: underline;
}

.font-letter:hover {
  color:#FF0000;
  cursor: pointer;
  text-decoration: underline;
}

.font-image-link {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 10px;
  color: #0033CC;
}

.font-image-link:hover {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 10px;
  color: #FF0000;
}

.font-gray {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  color: #555555;
  text-decoration: none;
}

.font-gray-small {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #777777;
}

.font-black-small {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 12px;
  color: #000000;
}

.font-big {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 17px;
  font-weight: bold;
  color: #103900;
}

.font-supersmall {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 10px;
}

.font-page {
  font-family: Verdana, Geneva, Arial,
Helvetica, sans-serif;
  font-size: 10px;
  font-size:11px;
  color:#001C6B;
  font-weight: bold
}

.thumbnail {
  width: 100px;
  height: 100px;
  border: 1px solid #CCCCCC;
}

.thumbnail:hover {
  width: 100px;
  height: 100px;
  border: 1px solid #FF0000;
  cursor: pointer;
}

.imageView {
  width: 400px;
  height: 400px;
  border: 1px solid #CCCCCC;
```

}

javascript.js

```
function changeDetails() {

  var type =
  document.getElementById("type").value;

  if(type == "Book") {

    document.getElementById("author").style.dis
    play = '';
    document.getElementById("editor").style.dis
    play = 'none';
    document.getElementById("publisher").style.
    display = '';
    document.getElementById("chapter").style.di
    splay = 'none';
    document.getElementById("pages").style.disp
    lay = 'none';
    document.getElementById("booktitle").style.
    display = 'none';
    document.getElementById("school").style.dis
    play = 'none';
    document.getElementById("institution").styl
    e.display = 'none';
    document.getElementById("volume").style.dis
    play = '';
    document.getElementById("series").style.dis
    play = '';
    document.getElementById("address").style.di
    splay = '';
    document.getElementById("edition").style.di
    splay = '';
    document.getElementById("subtype").style.di
    splay = 'none';
    document.getElementById("organization").sty
    le.display = 'none';
    document.getElementById("number").style.dis
    play = 'none';
    document.getElementById("howpublished").sty
    le.display = 'none';
    document.getElementById("journal").style.di
    splay = 'none';

    document.getElementById("editorVal").value
    = "";
    document.getElementById("chapterVal").value
    = "";
    document.getElementById("pagesVal").value =
    "";
    document.getElementById("booktitleVal").val
    ue = "";
    document.getElementById("schoolVal").value
    = "";
    document.getElementById("institutionVal").v
    alue = "";
    document.getElementById("subtypeVal").value
    = "";
    document.getElementById("organizationVal").
    value = "";
    document.getElementById("numberVal").value
    = "";
    document.getElementById("howpublishedVal").
    value = "";
    document.getElementById("journalVal").value
    = "";

    document.getElementById("authorRequired").s
    tyle.display = '';
    document.getElementById("publisherRequired"
    ).style.display = '';
    document.getElementById("chapterRequired").
    style.display = 'none';
    document.getElementById("pagesRequired").st
    yle.display = 'none';
    document.getElementById("booktitleRequired"
    ).style.display = 'none';
    document.getElementById("schoolRequired").s
    tyle.display = 'none';
    document.getElementById("institutionRequire
    d").style.display = 'none';
    document.getElementById("journalRequired").
    style.display = 'none';
    document.getElementById("yearRequired").sty
    le.display = '';
    document.getElementById("noteRequired").sty
    le.display = 'none';

  } else if(type == "Inbook") {

    document.getElementById("author").style.dis
    play = '';
    document.getElementById("editor").style.dis
    play = 'none';
    document.getElementById("publisher").style.
    display = '';
    document.getElementById("chapter").style.di
    splay = '';
    document.getElementById("pages").style.disp
    lay = '';
    document.getElementById("booktitle").style.
    display = 'none';
    document.getElementById("school").style.dis
    play = 'none';
    document.getElementById("institution").styl
    e.display = 'none';
    document.getElementById("volume").style.dis
    play = '';
    document.getElementById("series").style.dis
    play = '';
    document.getElementById("address").style.di
    splay = '';
    document.getElementById("edition").style.di
    splay = '';
    document.getElementById("subtype").style.di
    splay = '';
    document.getElementById("organization").sty
    le.display = 'none';
    document.getElementById("number").style.dis
    play = 'none';
    document.getElementById("howpublished").sty
    le.display = 'none';
    document.getElementById("journal").style.di
    splay = 'none';

    document.getElementById("editorVal").value
    = "";
    document.getElementById("booktitleVal").val
    ue = "";
    document.getElementById("schoolVal").value
    = "";
    document.getElementById("institutionVal").v
    alue = "";
    document.getElementById("organizationVal").
    value = "";
    document.getElementById("numberVal").value
    = "";
    document.getElementById("howpublishedVal").
    value = "";
    document.getElementById("journalVal").value
    = "";

    document.getElementById("authorRequired").s
    tyle.display = '';
    document.getElementById("publisherRequired"
    ).style.display = '';
```

```
  document.getElementById("chapterRequired").
  style.display = '';
  document.getElementById("pagesRequired").st
  yle.display = '';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Incollection") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = '';
  document.getElementById("publisher").style.
  display = '';
  document.getElementById("chapter").style.di
  splay = '';
  document.getElementById("pages").style.disp
  lay = '';
  document.getElementById("booktitle").style.
  display = '';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = '';
  document.getElementById("series").style.dis
  play = '';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = '';
  document.getElementById("subtype").style.di
  splay = '';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = '';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = '';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Inproceedings") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = '';
  document.getElementById("publisher").style.
  display = '';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = '';
  document.getElementById("booktitle").style.
  display = '';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = '';
  document.getElementById("series").style.dis
  play = '';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = '';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("chapterVal").value
  = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
```

```
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = '';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Article") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = '';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = '';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = 'none';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = '';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = '';

  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("seriesVal").value
  = "";
  document.getElementById("addressVal").value
  = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = '';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Masters Thesis" || type
== "PhD Thesis") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = '';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = 'none';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = '';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("volumeVal").value
  = "";
  document.getElementById("seriesVal").value
  = "";
```

```
  document.getElementById("editionVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = '';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Technical Report") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = '';
  document.getElementById("volume").style.dis
  play = 'none';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = '';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = '';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("volumeVal").value
  = "";
  document.getElementById("seriesVal").value
  = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = '';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Unpublished") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = 'none';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = 'none';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';
```

```
  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("volumeVal").value
  = "";
  document.getElementById("seriesVal").value
  = "";
  document.getElementById("addressVal").value
  = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = '';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = 'none';
  document.getElementById("noteRequired").sty
  le.display = '';

} else if(type == "Manual") {

  document.getElementById("author").style.dis
  play = '';
  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = 'none';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = '';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = '';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("volumeVal").value
  = "";
  document.getElementById("seriesVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = 'none';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = 'none';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Proceedings") {

  document.getElementById("author").style.dis
  play = 'none';
  document.getElementById("editor").style.dis
  play = '';
  document.getElementById("publisher").style.
  display = '';
  document.getElementById("chapter").style.di
  splay = 'none';
```

```
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = '';
  document.getElementById("series").style.dis
  play = '';
  document.getElementById("address").style.di
  splay = '';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = '';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = 'none';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("authorVal").value
  = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("howpublishedVal").
  value = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = 'none';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
  document.getElementById("institutionRequire
  d").style.display = 'none';
  document.getElementById("journalRequired").
  style.display = 'none';
  document.getElementById("yearRequired").sty
  le.display = '';
  document.getElementById("noteRequired").sty
  le.display = 'none';

} else if(type == "Miscellaneous") {

  document.getElementById("author").style.dis
  play = '';

  document.getElementById("editor").style.dis
  play = 'none';
  document.getElementById("publisher").style.
  display = 'none';
  document.getElementById("chapter").style.di
  splay = 'none';
  document.getElementById("pages").style.disp
  lay = 'none';
  document.getElementById("booktitle").style.
  display = 'none';
  document.getElementById("school").style.dis
  play = 'none';
  document.getElementById("institution").styl
  e.display = 'none';
  document.getElementById("volume").style.dis
  play = 'none';
  document.getElementById("series").style.dis
  play = 'none';
  document.getElementById("address").style.di
  splay = 'none';
  document.getElementById("edition").style.di
  splay = 'none';
  document.getElementById("subtype").style.di
  splay = 'none';
  document.getElementById("organization").sty
  le.display = 'none';
  document.getElementById("number").style.dis
  play = 'none';
  document.getElementById("howpublished").sty
  le.display = '';
  document.getElementById("journal").style.di
  splay = 'none';

  document.getElementById("editorVal").value
  = "";
  document.getElementById("publisherVal").val
  ue = "";
  document.getElementById("chapterVal").value
  = "";
  document.getElementById("pagesVal").value =
  "";
  document.getElementById("booktitleVal").val
  ue = "";
  document.getElementById("schoolVal").value
  = "";
  document.getElementById("institutionVal").v
  alue = "";
  document.getElementById("volumeVal").value
  = "";
  document.getElementById("seriesVal").value
  = "";
  document.getElementById("addressVal").value
  = "";
  document.getElementById("editionVal").value
  = "";
  document.getElementById("subtypeVal").value
  = "";
  document.getElementById("organizationVal").
  value = "";
  document.getElementById("numberVal").value
  = "";
  document.getElementById("journalVal").value
  = "";

  document.getElementById("authorRequired").s
  tyle.display = 'none';
  document.getElementById("publisherRequired"
  ).style.display = 'none';
  document.getElementById("chapterRequired").
  style.display = 'none';
  document.getElementById("pagesRequired").st
  yle.display = 'none';
  document.getElementById("booktitleRequired"
  ).style.display = 'none';
  document.getElementById("schoolRequired").s
  tyle.display = 'none';
```

```
    document.getElementById("institutionRequire
    d").style.display = 'none';
    document.getElementById("journalRequired").
    style.display = 'none';
    document.getElementById("yearRequired").sty
    le.display = 'none';
    document.getElementById("noteRequired").sty
    le.display = 'none';

  } else {

    document.getElementById("author").style.dis
    play = 'none';
    document.getElementById("editor").style.dis
    play = 'none';
    document.getElementById("publisher").style.
    display = 'none';
    document.getElementById("chapter").style.di
    splay = 'none';
    document.getElementById("pages").style.disp
    lay = 'none';
    document.getElementById("booktitle").style.
    display = 'none';
    document.getElementById("school").style.dis
    play = 'none';
    document.getElementById("institution").styl
    e.display = 'none';
    document.getElementById("volume").style.dis
    play = 'none';
    document.getElementById("series").style.dis
    play = 'none';
    document.getElementById("address").style.di
    splay = 'none';
    document.getElementById("edition").style.di
    splay = 'none';
    document.getElementById("subtype").style.di
    splay = 'none';
    document.getElementById("organization").sty
    le.display = 'none';
    document.getElementById("number").style.dis
    play = 'none';
    document.getElementById("howpublished").sty
    le.display = 'none';
    document.getElementById("journal").style.di
    splay = 'none';

    document.getElementById("authorVal").value
    = "";
    document.getElementById("editorVal").value
    = "";
    document.getElementById("publisherVal").val
    ue = "";
    document.getElementById("chapterVal").value
    = "";
    document.getElementById("pagesVal").value =
    "";
    document.getElementById("booktitleVal").val
    ue = "";
    document.getElementById("schoolVal").value
    = "";
    document.getElementById("institutionVal").v
    alue = "";
    document.getElementById("volumeVal").value
    = "";
    document.getElementById("seriesVal").value
    = "";
    document.getElementById("addressVal").value
    = "";
    document.getElementById("editionVal").value
    = "";
    document.getElementById("subtypeVal").value
    = "";
    document.getElementById("organizationVal").
    value = "";
    document.getElementById("numberVal").value
    = "";
    document.getElementById("howpublishedVal").
    value = "";
    document.getElementById("journalVal").value
    = "";

    document.getElementById("authorRequired").s
    tyle.display = 'none';
    document.getElementById("publisherRequired"
    ).style.display = 'none';
    document.getElementById("chapterRequired").
    style.display = 'none';
    document.getElementById("pagesRequired").st
    yle.display = 'none';
    document.getElementById("booktitleRequired"
    ).style.display = 'none';
    document.getElementById("schoolRequired").s
    tyle.display = 'none';
    document.getElementById("institutionRequire
    d").style.display = 'none';
    document.getElementById("journalRequired").
    style.display = 'none';
    document.getElementById("yearRequired").sty
    le.display = 'none';
    document.getElementById("noteRequired").sty
    le.display = 'none';

  }
}

function showMetadata() {

  var category =
  document.getElementById("category").value;

  if(category == "Infection Assays" ||
  category == "Cell Culture" || category ==
  "Fluorenscence" || category == "Brightfield
  Microscopy") {

    document.getElementById("version").style.di
    splay = 'none';
    document.getElementById("startDate").style.
    display = '';
    document.getElementById("cellType").style.d
    isplay = '';
    document.getElementById("virusType").style.
    display = '';
    document.getElementById("infection").style.
    display = '';
    document.getElementById("medium").style.dis
    play = '';
    document.getElementById("temperature").styl
    e.display = '';
    document.getElementById("co2level").style.d
    isplay = '';
    document.getElementById("owner").style.disp
    lay = '';
    document.getElementById("container").style.
    display = '';

    document.getElementById("versionVal").value
    = "";

    if(category == "Infection Assays" ||
    category == "Cell Culture") {

      document.getElementById("endDate").style.
      display = '';
      document.getElementById("phenolRed").styl
      e.display = 'none';
      document.getElementById("magnification").
      style.display = 'none';
      document.getElementById("objective").styl
      e.display = 'none';
      document.getElementById("illuminationType
      ").style.display = 'none';
```

```
    document.getElementById("illuminationLeve
    l").style.display = 'none';
    document.getElementById("filter").style.d
    isplay = 'none';

    document.getElementById("magnificationVal
    ").value = "";
    document.getElementById("objectiveVal").v
    alue = "";
    document.getElementById("illuminationType
    Val").value = "";
    document.getElementById("illuminationLeve
    lVal").value = "";
    document.getElementById("filterVal").valu
    e = "";

  } else {

    document.getElementById("endDate").style.
    display = 'none';
    document.getElementById("phenolRed").styl
    e.display = '';
    document.getElementById("magnification").
    style.display = '';
    document.getElementById("objective").styl
    e.display = '';
    document.getElementById("illuminationType
    ").style.display = '';
    document.getElementById("illuminationLeve
    l").style.display = '';
    document.getElementById("filter").style.d
    isplay = '';

    document.getElementById("ed").val = "";
  }

} else {

  document.getElementById("startDate").style.
  display = 'none';
  document.getElementById("endDate").style.di
  splay = 'none';
  document.getElementById("cellType").style.d
  isplay = 'none';
  document.getElementById("virusType").style.
  display = 'none';
  document.getElementById("infection").style.
  display = 'none';
  document.getElementById("medium").style.dis
  play = 'none';
  document.getElementById("temperature").styl
  e.display = 'none';
  document.getElementById("co2level").style.d
  isplay = 'none';
  document.getElementById("owner").style.disp
  lay = 'none';
  document.getElementById("container").style.
  display = 'none';
  document.getElementById("phenolRed").style.
  display = 'none';
  document.getElementById("magnification").st
  yle.display = 'none';
  document.getElementById("objective").style.
  display = 'none';
  document.getElementById("illuminationType")
  .style.display = 'none';
  document.getElementById("illuminationLevel"
  ).style.display = 'none';
  document.getElementById("filter").style.dis
  play = 'none';

  document.getElementById("sd").value = "";
  document.getElementById("ed").value = "";
  document.getElementById("cellTypeVal").valu
  e = "";
    document.getElementById("virusTypeVal").val
    ue = "";
    document.getElementById("infectionVal").val
    ue = "";
    document.getElementById("mediumVal").value
    = "";
    document.getElementById("temperatureVal").v
    alue = "";
    document.getElementById("co2levelVal").valu
    e = "";
    document.getElementById("ownerVal").value =
    "";
    document.getElementById("containerVal").val
    ue = "";
    document.getElementById("magnificationVal")
    .value = "";
    document.getElementById("objectiveVal").val
    ue = "";
    document.getElementById("illuminationTypeVa
    l").value = "";
    document.getElementById("illuminationLevelV
    al").value = "";
    document.getElementById("filterVal").value
    = "";

    if(category == "ImageJ Plug-in") {
      document.getElementById("version").style.
      display = '';
    } else {
      document.getElementById("version").style.
      display = 'none';
      document.getElementById("versionVal").val
      = "";
    }

  }

}


function addField(id) {

  var tr=document.createElement("tr");
  var tr1=document.createElement("tr");
  var tr2=document.createElement("tr");
  var tr3=document.createElement("tr");
  var tr4=document.createElement("tr");
  var tr5=document.createElement("tr");
  var tr6=document.createElement("tr");
  var td=document.createElement("td");
  var td1=document.createElement("td");
  var td2=document.createElement("td");
  var td3=document.createElement("td");
  var td4=document.createElement("td");
  var td5=document.createElement("td");
  var td6=document.createElement("td");
  var td7=document.createElement("td");
  var td8=document.createElement("td");
  var td9=document.createElement("td");
  var td10=document.createElement("td");
  var td11=document.createElement("td");
  var td12=document.createElement("td");
  var table=document.createElement("table");
  var input1=document.createElement("input");
  var input2=document.createElement("input");
  var input3=document.createElement("input");
  var input4=document.createElement("input");
  var input5=document.createElement("input");
  var input6=document.createElement("input");

  var condition =
  document.createTextNode("Condition");
  var method =
  document.createTextNode("Method");
  var description =
  document.createTextNode("Description");
```

```
var variableType =
document.createTextNode("Variable Type");
var value =
document.createTextNode("Value");
var chemActivator =
document.createTextNode("Chemical
Activator");
var i = 0;

td1.appendChild(condition);
td1.style.fontWeight="bold";
tr1.appendChild(td1);

input1.type = "text";
input1.name = "condition";
input1.size="90";
td2.appendChild(input1);
tr1.appendChild(td2);
table.appendChild(tr1);

td3.appendChild(method);
td3.style.fontWeight="bold";
tr2.appendChild(td3);

input2.type = "text";
input2.name = "methodName";
input2.size="90";
td4.appendChild(input2);
tr2.appendChild(td4);
table.appendChild(tr2);

td5.appendChild(description);
td5.style.fontWeight="bold";
tr3.appendChild(td5);

input3.type = "text";
input3.name = "methodDesc";
input3.size="90";
td6.appendChild(input3);
tr3.appendChild(td6);
table.appendChild(tr3);

td7.appendChild(variableType);
td7.style.fontWeight="bold";
tr4.appendChild(td7);

input4.type = "text";
input4.name = "varType";
input4.size="90";
td8.appendChild(input4);
tr4.appendChild(td8);
table.appendChild(tr4);

td9.appendChild(value);
td9.style.fontWeight="bold";
tr5.appendChild(td9);

input5.type = "text";
input5.name = "varValue";
input5.size="90";
td10.appendChild(input5);
tr5.appendChild(td10);
table.appendChild(tr5);

td11.appendChild(chemActivator);
td11.style.fontWeight="bold";
tr6.appendChild(td11);

input6.type = "text";
input6.name = "chemActivator";
input6.size="90";
td12.appendChild(input6);
tr6.appendChild(td12);
table.appendChild(tr6);

td.appendChild(table);
  tr.appendChild(td);
  tr.style.backgroundColor="#E1E8F6";

  document.getElementById(id+"Id").appendChild
  (tr);

}

function deleteField(id) {
  var parentNode =
  document.getElementById(id+"Id");
  var numChildNode = parentNode.childNodes;

  if(numChildNode.length-1 != 1) {
    parentNode.removeChild(parentNode.lastChild
    );
  }
}
```

XI. ACKNOWLEDGEMENTS

Seems just like yesterday when I first entered the UP Manila premises, uncertain as to what future might be awaiting me. I was living in doubt for a year thinking if I have chosen a right track. There were times I almost gave up, but I never did because of the sincere help and encouragement I received from the people around me. They taught me to never be disheartened by the simple failures on school stuff, such as on homeworks, quizzes, projects and exams. I just hadn't tried by best they said. And that's not the basis of who you will be in the future. I am pleased that I listened to them. Now, the chapter of my life as a college student has finally ended. Sleepless nights are over. No regrets with the decision I made. As I reminiscing the past, I can say that my college life is one of the best chapters in my life.

Foremost, I would like to thank God, the omniscient and omnipotent Father, who has always been blessing me and giving me strength and wisdom to overcome my trials and hardships in life. The one helping me when I am on the verge of giving up.

Kudos to all my professors in UPM! Thank you for the knowledge you imparted us, and for making us ready to face the real world. My utmost gratitude is for my thesis adviser, Ma'am Sheila Magboo, for entrusting in me the VirHoLex system, and for guiding me throughout my SP proposal and defense. I would also like to extend my gratitude to Sir Bryann Chua who helped me and shared with me the instructions that I need during the deployment of the system on MaxPlanck server. Moreover, the deployment wouldn't be possible without the assistance of Dr. Markus Rampp from MaxPlanck Institute. Thank you for quickly and patiently answering my questions in the midst of your busy schedule.

I would also like to show my gratitude to Kuya Jeric Morales, for allowing me to bug you regarding the installation setup of VirHolex system into my computer, and for bearing

with my inquiries concerning the system. I know it would be difficult to recall all the details that I was asking, but you still tried your very best. I greatly appreciate your effort.

I owe my warmest appreciation to my Cambridge friends who played a significant role why I successfully completed my SP defense. The never-ending support (with a little bit of pressure on it) you've shown upon me gave me the drive and motivation to finish my thesis. You were lenient when it comes to work and leaves just so I could have time for my SP. Thank you also for the opportunity you've given to me to work with you. In fact, the things I've learned from the company have a great contribution to my SP. A special thanks to Ariel Tabag, for helping me during system deployment on MaxPlanck server.

To my ComSci Dose '80 family, I love you guys! And I miss you so much! No words can express how thankful I am to be part of this block. You made my college years memorable and unforgettable. Being with you guys is the best way to relieve my stress and temporarily forget my problems in life. To my Stat friends, thank you for believing in me and for helping me on my studies. Actually, you inspired me to be a diligent student. To the beloved mother of the block, Muhmi JJ, thank you so much for the support and the help you've given me during my SP proposal and defense ☺

Most importantly, I would like to express my sincerest gratitude to my family and relatives who are always there at my side during the ups and downs of my life. I won't be the person I am right now without your endless love, support and encouragement. You are my inspiration and the core of my strength. It is for you that I dedicated my diploma. To my dearest mom, I really really love you though I haven't told it to you yet in person. I maybe stubborn and 'pasaway' sometimes, but it doesn't mean that I fail to appreciate your genuine care and love for me. I truly value ever little thing you do. To my Uncle Erwin, thank you for

the financial support you've provided to me and my family. Education is the best gift you can ever give to a person, and I am lucky enough to have such a loving uncle. To my other relatives, I may not mention your names one by one, thank you for always present in times we're in trouble. I love you all! ☺

To all those people who supported me along the way and to those who made my college life worthwhile, my dearest friends, I offer you my sincerest gratitude for making my stay in UP Manila a happy and fruitful one.