

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

**WADoTS:
A Workflow Aware Document Tracking System for FDA**

**A Special Problem in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science**

Submitted by:

Jose Louie Mark Z. Año

2008-56492

Abstract

FDA processes hundreds of documents per day. The movements of these documents are only recorded as they arrive to a particular department. Since these records are not centralized, there is no way of monitoring the movements of the documents efficiently. A centralized information system would seem to provide a solution, but FDA processes its documents with constantly changing workflows. Such changes would require heavy maintenance on information systems.

WADoTS is a workflow aware document tracking information system that is capable of facilitating the dynamic changes in document processing requirements and workflows of the FDA. Workflows and requirements are customized by the users without changing the system's code. It provides user friendly interface in setting detailed routing and processing requirements of workflows. Authorized users are also able to see graphical representations of each document's status.

Keywords: *Document, Tracking System, Workflow, Workflow Aware, Dynamic Changes*

ACCEPTANCE SHEET

The Special Problem entitled “**WADoTS: A Workflow Aware Document Tracking System for FDA**” prepared and submitted by **Jose Louie Mark Z. Año** in partial fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science** has been examined and is recommended for acceptance.

Geoffrey A. Solano, Ph.D. (candidate)

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. Ma. Sheila A. Magboo, M.S.	_____	_____
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., Dr. Eng'g

Dean

College of Arts and Sciences

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	REVIEW OF RELATED LITERATURE	8
III.	THEORETICAL FRAMEWORK.....	13
IV.	DESIGN AND IMPLEMENTATION	22
V.	RESULTS	35
VI.	DISCUSSION.....	63
VII.	CONCLUSION.....	65
VIII.	RECOMMENDATION.....	66
XI.	BIBLIOGRAPHY	67
X.	APPENDIX	70
XI.	ACKNOWLEDGEMENT.....	191

I. Introduction

A. Background of the study

It is important to establish the country's current state of health, especially in a rapid population growth. By 2010 the population of the Philippines is approximated to be 94 million. Furthermore, population growth rate is at 2.04% (2000-2007 data) [1]. Food, pharmaceuticals, cosmetics, household hazardous materials and pesticides are increasingly critical components in improving the health of Filipinos, thus there is a need for an administration that develops plans policies, programs and strategies for regulating these products [2].

In the year 1961-1962 the late Department of Health (DOH) Secretary Francisco Duque Sr. created a subcommittee on Food and Drugs. This aims to initiate an administration bill to Congress to enact a law that ensures the safety, purity and quality of foods, drugs and cosmetics being made available to the Filipino public. On June 22, 1963 Republic Act No. 3720 known as the "Food, Drug and Cosmetic Act" was passed into law. To carry out the provisions of R.A. 3720, the Food and Drug Administration (FDA) was created [2].

FDA's core responsibility is assuring the safety, purity, quality and effectiveness of food, pharmaceuticals, cosmetics, household hazardous and pesticides. Working with patients, physicians, pharmacists, and industry, FDA plays a critical role in the detection and management of these product's safety issues, by helping to ensure the safe use of marketed products by providing the best possible information to the Filipino public [2].

Specifically, FDA has the following services: licensing and license renewal of establishments which includes manufacturers, traders, repackers, packers, importers, exporters, distributors and drug outlets (*Appendix 1, 2, 3, 4, 5 and 6*); product registration and license renewal of food, pharmaceuticals, cosmetics, household hazardous and pesticides (*Appendix 11, 12, 13, 14, 15, 16, 17, 18 and 19*) [3].

In all of these services' documents, the initial stage is the Information and Compliance Section (PAICS) or Policy, Planning and Advocacy Division (PPAD), and then they go to the administrator's office. In the administrator's office, these documents are assessed, commented, pointed to the next route then recorded for future references and tracking purposes. Recording the document means that the documents' time-in, time-out, comments, next route and other desired fields are stored electronically in a Microsoft Access file or in a Microsoft Excel file. These documents are then sent to other departments following their respective routes. All these departments including their extensions have the same function as of the admin department. They assess, comment, point to the next route and record the document [4].

B. Statement of the Problem

Conferring to interviews with the FDA director Dr. Suzette H. Lazo and her secretary Mrs. Susy Pauline, there is an estimate of five-hundred (500) documents that is being processed at the admin department each day. All these documents are then spread out to other departments. There are different independent recoding protocols, like Microsoft Access and Microsoft Excel, in every station and there is no centralized database. The only way to completely track a document is to physically go from one department to the other.

The processing speed of a document can depend on the document congestion of the departments on its path. It can also depend on its approval on every department. A document's process is slow if it has been frequently disapproved. In case of a delay in the process, there is no such way of knowing the reason for the delay. There is also no way of estimating how much time is left in a document's process because nothing is monitoring the amount of documents per department [5].

According to Mrs. Susy Pauline, FDA's workflows on document processing are complicated. This is due to having numerous services of FDA, resulting into many workflow patterns (*See Appendix 1 to 10*). In addition, workflow patterns are constantly changing. FDA's dynamic workflow is due to FDA's

adoption to new technology and international standards, not because of administrative weaknesses. On August 18, 2009 Republic Act No. 9711, an act strengthening and rationalizing the regulatory capacity of the Bureau of Food and Drugs by establishing adequate testing laboratories and field offices, upgrading its equipment, augmenting its human resources complement, giving authority to retain its income was signed by the president. This results in a change in workflow and staffs. The most recent change in workflow was on November 2011. Dynamic workflow has an influence on document flow. A specific document flow is only subject to the workflow version which it follows [6].

Applicants who submitted the documents have no way of knowing when the processes end and what are the status of their documents. Monitoring their applications cause them valuable resources because they must go to FDA just follow up to their documents [6].

C. Objectives

General Objective

To develop a workflow aware system that can be customized to facilitate the monitoring and tracking of the status of documents being processed in the FDA.

Specific Objectives

To develop a system that provides the following functionalities in correspondence with the users:

1. System Administrator (IT Personnel)
 - a. log in and manage account
 - b. view/add/edit/delete department
 - c. view all users in a department
 - d. view/add/edit/delete user account

- e. view/add/edit/delete document record type
 - I. parameters
 - II. parameter input type
 - III. parameter default values
 - f. view/add/edit/delete document record
 - g. view/add/edit/delete department to workflow path
 - h. view workflow traffic
 - i. view a specific document's workflow path traffic
2. Director/ Department Chief
- a. log in and manage account
 - b. view all departments
 - c. view all users in a department
 - d. view/add/edit/delete document record
 - e. approve/disapprove a document

 - f. view workflow traffic
 - g. notify applicant via e-mail
 - I. if there is a problem in the application/there is a need for a follow up by the applicant
 - II. if the application is complete
3. Director assistant/ Department Chief assistant
- a. log in and manage account
 - b. view all departments
 - c. view document records of own department
 - d. view/add/edit/delete document record
 - e. approve/disapprove document

- f. view workflow traffic
- g. notify applicant via e-mail
 - I. if there is a problem in the application/there is a need for a follow up by the applicant
 - II. if the application is complete
- 4. Applicants
 - a. logging in and manage account
 - b. view document status
 - I. location
 - II. check-in and check-out time at each department in the document's workflow path
 - III. document traffic of document's workflow path

D. Significance of the Study

The proposed system would improve the current way on how processed document movement within FDA is tracked, benefiting FDA staffs and establishments which owns the documents being processed in FDA. This serves as an electronic tracking system for monitoring the progress of document movements. With the system, the individual recording of document movements by each department of FDA is centralized. Having a centralized record of document movements enables concerned FDA staffs to locate a document easily as they can just type a keyword or identification code of the document they want to find. They would be able to know when a document checked in and checked out at a given department thus they know what department a document is delayed.

Moreover, concerned FDA staffs would be provided information about the document traffic in the departments. The information is relayed in a graphical manner so that they can really visualize what is happening. They would know where the bottleneck of the document movements is, so that if ever they want to optimize workflow they can have bases.

Applicants do not spend precious resources just to monitor the progress of their document's processes. They see online reports on their applications such as where their documents are and traffic information on offices of their document's workflow path

Instead of being a fixed, hard coded system, the proposed system is a generator of a system. It enables super users to add, edit and delete departments, documents and other users. This provides FDA personnel liberty on how the workflow is built because they themselves are the ones who design the structure of the system and they themselves would build it. The system is customizable so as to adapt to a constant change in the document's workflows. FDA's IT personnel can add, edit and delete departments, documents and users.

E. Scope and Limitations

- Verification of the workflow is outside of the system.
- Director/Department Chief and their assistants can only add, edit and delete document record of their own department.
- Only the system administrator can make and edit the system structure.
- There can only be two classes of users wherein the systems administrator can add in a department, the Department Chief and Department Chief assistant.

- When a document type's parameter of is edited or deleted, its existing instances remains unchanged. Changes apply to instances made after it is edited. When a document type is deleted, it can no longer be instantiated.
- A document record can only be edited or deleted if the document is still not read by the staffs of the document's next destination.
- An applicant account can only be added when a station is created.

F. Assumptions

This paper assumes that the FDA IT personnel were already built a system using the proposed system based on the desired workflow of the system given by the administrators. It is also assumed that all end-users are oriented and trained with how the system works. It is also assumed that actors in the system have an Ethernet and Internet connection to access FDA servers and machines.

II. Review of Related Literature

It is inevitable that paper is still the main medium of communication in every office environments. It was commonly believed that in the future offices, paper will no longer be used or its importance will be limited. These existing physical documents are supposed to be replaced by their electronic document versions. But the use of paper documents is increasing, especially in the form of mails and print outs. In general people feel comfortable with paper in reading long articles, in annotating them and also for record keeping purposes. Paper documents make people's work more convenient as compared to electronic documents. That is why there is a need to have a centralized system that provides users with a proactive real time environment that automatically retrieves and displays digital information associated with any actual document [7].

An example system is the University of the Philippines (UP) DocuTrak discussed in [8]. It makes use of barcode labels to track documents as they make their way through the various unit offices of a UP campus. DocuTrak can monitor and identify bottlenecks in unit offices. It supports service delivery by way of consistency, continuity, efficiency and productivity. To use the system, unit representatives have to submit the required information to the administrator. When these unit representatives are already added as system users, their unit office is able to create and monitor all in-process documents that are within their concern. Various types of documents that can be monitored through DocuTrak like correspondences and accounting forms. In order to create a document in DocuTrack, users must fill up the parameters like document title, author, document type, action needed and type of access, and its unique tracking number. The document's tracking number is its unique identifier that is attached as a barcode label. The tracking number enables unit offices involved in the processing documents to create, receive, act upon, release, and track these in-process documents by means of DocuTrak. These labels are administered only by the systems administrator.

Also in the academe, document tracking systems are significant especially in research management. A scientific research management system is designed in Wuhan University of Science and Engineering. The system aims to improve efficiency in management and will help in making decision for the management. It has four users, the scientific research administrators, the department administrators, the scientific research teachers and the visitors. The system assigns different functions for different users which are concentrated in different pages. All administrators and teachers have their unique ID number. Administrators have the highest authority. They have the unique right for final examination of scientific research information. The system also manages the scientific research workload of every teacher in a year. Research workers must have to input their scientific research information into the system to be managed and marked by the scientific research quantitative methods of Wuhan University of Science and Engineering. When querying data, the system makes use of data view which integrates data in one table or several tables to fasten querying speed and make management easier [9].

As of now, a lot of government institutions here in the Philippines use document tracking systems to speed up their transaction processes. The Department of Science and Technology has a system that maintain an organized database of incoming and outgoing documents that includes tracking on the sender/addressee, verifying accountability of documents of person-in-charge and actions taken. Department of Environment and Natural Resources (DENR) and Department of Social Welfare and Development (DSWD) separately maintain a document tracking system that monitors incoming and outgoing documents. This system allows a particular document to be delivered electronically from the sender to the receiver. The system speeds up inter-Departmental transactions [10].

Another example of an application of a document tracking system is the Army Records Information Management System (ARIMS) in the United States of America (USA) which is discussed in [11]. The purpose of Army recordkeeping and record tacking is to appropriately manage information from its creation through final disposition, according to Federal laws in USA. Provides life cycle tracking and management for the systematic identification, maintenance, storage, retrieval, retirement, and destruction

of Army information recorded on paper, microforms, electronic, or any other. The system supplies the only legal authority for destroying nonpermanent Army information. It preserves those records needed to defend the rights and interests of the Army and its current and former members, and those records that are of permanent value. It ensures records related to matters involved in administrative or legal proceedings are retained until the staff judge advocate or legal advisor authorizes resumption of normal disposition.

Government institutions, army and academe are not the only ones using this kind of system. In a private business setting, project progress control and monitoring are vital tasks for the project management. Every member needs to know how the project progresses. An example for this is in [12], where an integrated simulation model called DCM (Digitalizing the Construction Monitoring) is developed to integrate digital images of construction scene with AutoCAD drawings. It resolves the existing project progress reporting problems by enabling construction managers to develop progress reports in a more consistent and accurate manner as built project schedules can be transferred easily to facility managers for operation, maintenance, renovation and demolition. The Digitalizing the Construction Monitoring (DCM) model is developed using the Relational Database Management System (RDBMS). The addition of digital images and drawings enables construction managers to develop progress reports in a more consistent and accurate manner and more accurate as-built project schedules can be transferred to facility managers for operation, maintenance, renovation and demolition.

In a business setting, two major contributors to a company success are sales and inventory control. But, if a problem arises in any one of the two would mean the delaying the company daily operations and this can cause financial loss to the company. To counteract this problem, a relational database of a tracking system is designed and developed based on the user's requirements to overcome their daily problems/issues. The database system targets four groups of users which are the salesmen, accounts personnel, provisioning team and customers who will rely heavily on the system for their daily operations [13].

The notion of workflows has evolved from a means to describe the flow of paperwork through an organization to a more abstract and general technique used to express best practices and lessons learned in many application domains. Workflows provide the explanation how work results are meant to be attained and how problems that occurred in the past be avoided. To do this throughout the organization, workflow management systems (WfMS) is used. In order to make such support work the WfMS must be aware of the overall progress reached. Then it checks whether the workflow can be completed in time. Further, by evaluating a series of progress information recorded over time, detailed documentation about project progress can be created. However, users are often unwilling to manually enter information about work progress into a WfMS. Particularly when these users are experts and the tasks in question have been carried out many times before, the benefits of a WfMS soon appear marginal to people who are technical roles [14]. In [14], the agent-oriented Progress Information System (PIE) is presented which has been created to overcome this problem by means of an automatic workflow enactment tracking. The underlying model of workflow definition and enactment supports frequent changes, which is particularly crucial for organizational workflows as these have to be adapted often to respect shifted priorities. An example of a system that lets you evaluate a workflow is presented in [15]. It is a lightweight, distributed prototype of a process information system that allows the evaluation of workflow history data using the three perspectives process, resource and object. Security is important in these kinds of systems. So in order to maintain a flexible level of privacy, the prototype implements a security concept on the bases of access profiles for evaluation methods, data sources and information objects.

The need for dynamic modifications of workflow models while workflows are running is an also an important feature in scientific information systems. In [16] an approach called Workflow Aware Scientific Application (WASA) is introduced. WASA aims to have a flexible and platform independent workflow support. The conceptual design and functionality of a WASA prototype is outlined in its core workflow engine. In WASA workflow models may be built from scratch or existing workflow models may be reused. When the system is started the workflow engine reads workflow models from the

workflow library. Users access the workflow system using workflow clients. The basic functionality of a workflow client is to inform users in general of activities to perform. For each activity, a role is defined and when the activity is about to be started the workflow engine determines the users ready to perform it.

One advantage of a workflow aware document tracking system is that it can assist users in decision making. A model called QuESt (Quality E-management System) assimilates web-based technologies into an interconnected system to enable decision making by all stakeholders at the different levels of responsibility. It integrates participatory decisions which ensure greater efficiency in management. It connects, links, and monitors important decision making processes. Its main aim was to increase administrative efficiency, the system has a built in mechanism for minimizing wastage of resources, staff time as well as workload. It is a platform for administrators to monitor internal and external processes, for the academe to participate in group decision making and provide feedback to management [17].

A tracking system can not only track physical entities like documents, people or goods. Provided with the right algorithm, it can also monitor the cost of resources in the processing of those entities. In [18] Activity Based Cost Tracking Systems are discussed and patented. These systems relates on the monitoring of medical procedure cost by tracking the movements of personnel or equipment and supplies during the procedure. Then it is associated with a particular cost. An activity based tracking system tracks the costs of activities of persons and objects in a defined space such as a hallway or room. Each person and object is attached with a transponder, and the transponders transmit a unique identifying code. A reader of the transponder is located in a defined space. It reads the code from each of the transponders. These codes are associated by a database to the objects or persons to which the transponders are attached. In order to compute the costs of the activities of the persons and objects, a cost computer uses the times of the entry and exit of objects or persons. This cost tracking is important in establishing profitable pricing and monitoring expenses and client budgets.

III. Theoretical Framework

1. Food and Drug Administration (FDA)

I. Organizational Units

The administration has the following major organizational units according to [19]:

A. Office of the Director/Assistant Director

A.1. Director

A.2. Assistant Director

B. Product Evaluation Director

B.1. Product Evaluation Section

B.2. Product Registration Section

B.3. Product Advertisement Section

B.4. Adverse Drug Reaction Section

C. Regulation Division

C.1. Inspection Section

C.2. Licensing Section

C.3. Import and Export Section

D. Laboratory Division

D.1. Physico-chemical Section

D.2. Microbiological Section

D.3. Toxicology Section

E. Administrative Division

E.1. Planning and Budget Section

E.2. Personnel Section

E.3. Supply and Property Section

E.4. Cash Collection and disbursement Section

E.5. Accounting Section

E.6. Central Record Section

F. Technical Supervision:

Food and Drug Services under the Regional Health Offices including Integrated Provincial Health Offices.

G. Creation of Ad Hoc Committees on Food and Drug based on specialization from various sectors as the need arises.

A. Office of the Director

The Office of the Director provides consultative/advisory services and act as the administrative and executive arm of the DOH on matters pertinent to food, drug, including traditional medicine, cosmetic, device and household products containing hazardous substances. It develops and recommends policies, rules, regulations and standards in accordance with Republic Act No. 3720, P.D. 881, R.A. 5921, R.A. 6425 (*See Appendix*) and other pertinent laws and regulations. It coordinates with the regional health

Offices in the technical supervision of the personnel thereat in connection with the implementation of the aforestated functions. It plans, directs, coordinates and evaluates the over-all administrative, technical, legal and public affairs operation of the Administration.

B. Product Evaluation Division

Responsible for evaluation of food, drug, traditional medicine, device, cosmetic and household products containing hazardous substance and the registration of the same in compliance with the Food, Drug and Cosmetic Act, Decree on Hazardous Substances and other laws. Monitor and check the veracity of claims in advertisement of products and set-up a system for monitoring adverse drug reactions.

C. Regulation Division

Plans, directs and supervises the implementation or rule, regulations and standard operating procedures in inspection and licensing of establishments within FDA jurisdiction.

D. Laboratory Division

Plans, directs and supervises the implementation or rules, regulations and standard operation procedures in the laboratory testing of products for identity, purity, quality and fill of container and prepares certificate of compliance or non-compliance.

E. Administrative Division

The Administrative Division performs the following functions:

- a. Develops and recommends the short/long range plan and budget estimates of FDA.
- b. Provides FDA with general administrative support services related to:
 1. Personnel recruitment, training and development;
 2. Record and property management;
 3. Cash collection and disbursement;
 4. Accounting;
 5. Communication and transportation; and
 6. Other general maintenance services
- c. Performs other related functions as may be assigned by the Director.

2. Monitoring System

Monitoring system is a system is supervising of activities in progress to ensure that they are on course and on schedule in meeting the objectives [20]. It is the regular observation and recording of activities taking place in a project or event. It gathers information on all aspects of the project. It involves giving feedback about the progress of the project to the donors, implementers and beneficiaries of the project. Reporting enables the gathered information to be used in making decisions for improving project performance [21].

3. Tracking System

Tracking system is a system that follows something or someone. It basically follows the tracks of or trailing a record, project or a user [22]. By tracking, one can monitor the progress of something, to follow behind something or to try to find something. One can trace by means of vestiges, evidence, etc. or plot the path of and record data from (an aircraft, spacecraft, missile, etc.) using radar, a telescope, etc. [23].

4. Workflow

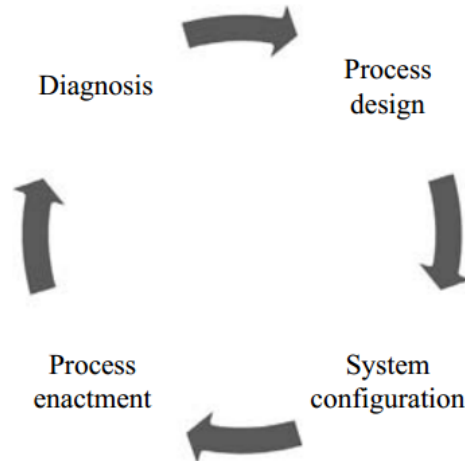
Workflow is the movement of documents and tasks through a business process. It can be a consecutive development of work activities or a compound set of processes each taking place concurrently, eventually impacting each other according to a set of rules, routes, and roles [24].

5. Workflow aware management system

Workflow aware management systems allow organizations to define and control the activities related with a business process. Moreover, these systems allow a business the opportunity to measure and analyze the execution of the process so that dynamic and continuous improvements can be made [24]. Dynamic modifications of workflow models can be done while workflows are running [16].

6. Business process management (BPM)

BPM is a systematic approach to making an organization's workflow more effective, more efficient and more capable of adapting to an ever-changing environment. The goal of BPM is to reduce human error and miscommunication [25]. The BPM life cycle can be illustrated in this way.



Where *process design* is the stage where fax- or paper-based as-is business processes are electronically modeled into BPM System. *System configuration* is where the configuring of the BPM System and the underlying system infrastructure. *Process enactment* is the step where electronically modeled business processes are deployed in BPM System engines. *Diagnosis* is where analysts identify and improve on bottlenecks and potential loopholes in the business processes [26].

7. Information System

An information system is a computer system that provides management and other staff within a specific organization with updated information regarding its performance. It is usually linked to a computer network where different computers join together in order to share data and resources. It is designed to capture, transmit, store, retrieve, manipulate, and display information used in one or more business processes. This kind of system outputs information in a form that is useable at any levels of the organization [27].

8. Database Management System (DBMS)

A Database management System is a software program that enables the creation and management of databases [28]. This system offers quick search and retrieval of information from a database. The DBMS determines how data are stored and retrieved. It must address problems such as security, accuracy, consistency among different records, response time, and memory requirements. [29]. With DBMS, one can sort the records according to a particular order indicated by one specific field, choose criteria that allow you to access specific records, count the number of records that meet a specific condition, and perform calculations [30].

9. PHP: Hypertext Preprocessor

PHP is an open source, server-side, HTML embedded scripting language used to create dynamic Web pages. The main goal of the language is to allow web developers to write dynamically generated web pages quickly [31]. PHP code is read or parsed by the server whenever a PHP page is accessed. The output from the PHP functions on the page is typically returned as HTML code, which can be read by the browser. Because the PHP code is transformed into HTML before the page is loaded, users cannot view the PHP code on the page. This makes the use of PHP pages a secure way to access databases and other confidential information [32].

10. CodeIgniter

CodeIgniter is a PHP framework with a very small footprint, built for PHP coders who need a simple and elegant toolkit to create full-featured web applications [33]. It provides useful libraries for common tasks. It also provides easy access to these libraries. It is also based on the Model-View-

Controller development pattern. View and controller classes are a necessary part of development under CodeIgniter while models are optional [34].

Technical Architecture

- A client is usually a program on the user's computer
- The server is usually a remote computer that holds interesting data.
- When the client needs data (a HTML page, for instance) it sends a request to the server, and the server responds by sending the requested data.
- After getting this response, the client may send a new request, which leads to a new response, and so on.

System Requirements

- Minimum Hardware Requirements:
 - Pentium/AMD 1 GHz processor
 - 512Mb RAM
 - 1024 x 768 screen resolution
 - At least 120GB hard drive
 - Existing L/WAN and Internet connection
 - Bluetooth
- Minimum Software Requirements:
 - Platform
 - Linux
 - MySQL version 5.0.67

- PHP 4.2.3 scripting language
- Apache web server 1.3
- JRE
- Web Browsers:
 - Chrome/Mozilla Firefox/Opera

IV. Design and Implementation

A. Entity Relationship Diagram

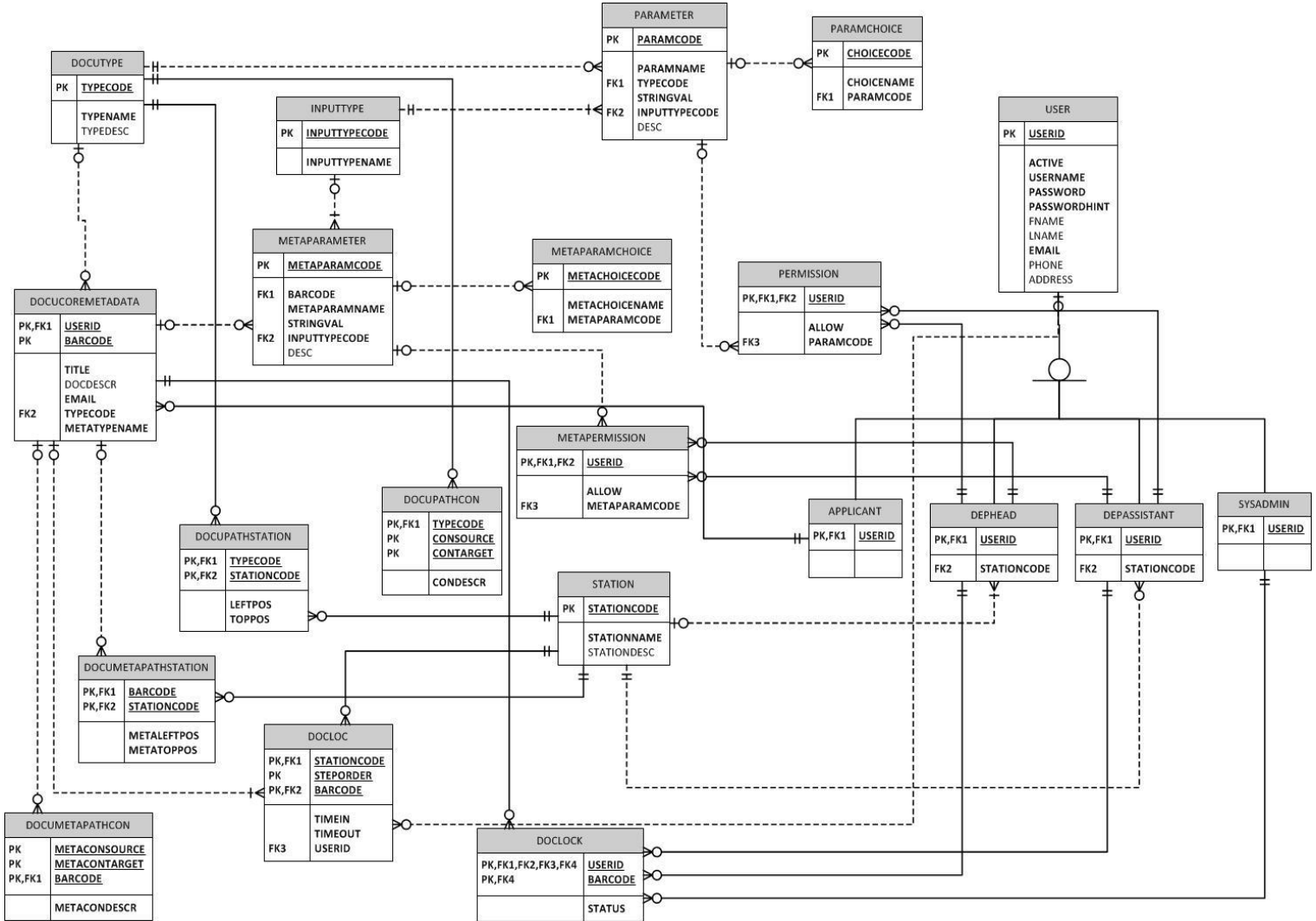


FIGURE 1. ENTITY RELATIONSHIP DIAGRAM, FDA Workflow Aware Document Tracking System (WADoTS)

The diagram for this system consists of 21 entities namely: DOCUTYPE, DOCUMENTMETADATA, DOCULOC, DOCUPATHSTATION, DOCUMETAPATHSTATION, DOCUPATHCON, DOCUMETAPATHCON, STATION, APPLICANT, DEPHEAD, DEPASSISTANT, SYSADMIN, PARAMETER, METAPARAMETER, PARAMCHOICE, METAPARAMCHOICE, DOCLOCK, PERMISSION, METAPERMISSION and INPUTTYPE.

DOCUPATHSTATION, DOCUPATHCON, PARAMETER, PARAMCHOICE and PERMISSION are tables that contain data that are replicated in DOCUMETAPATHSTATION, DOCUMETAPATHCON, METAPARAMETER, METAPARAMCHOICE and METAPERMISSION when a document is instantiated. This enables a specific document to exhibit the same properties throughout the process, even if the document type is edited or deleted. When a document is already in process, its properties (path and parameters) cannot be altered.

B. Data Dictionary

Data Field	Data Type	Description
TYPECODE	INT(11)	Identification number of the document type
TYPENAME	VARCHAR(20)	Name of the document type
TYPEDESC	VARCHAR(200)	Description of the document type

TABLE 1: DOCUTYPE contains the type of documents

Data Field	Data Type	Description
BARCODE	INT(11)	Identification number of the document
TITLE	VARCHAR(20)	Title of the document
TYPECODE	INT(11)	Identification number of the document's document type
DOCDESC	VARCHAR(200)	Description of the document
STEPCOUNT	INT(3)	The number of steps in the document's workflow
EMAIL	VARCHAR(20)	Email address of the applicants to notify them

TABLE 2: DOCUMENTMETADATA contains the documents data

Data Field	Data Type	Description
STATIONCODE	INT(11)	Identification number of the station
STATIONNAME	VARCHAR(20)	Name of the station
STATIONDESC	VARCHAR(200)	Description of the station

TABLE 3: STATION contains station details

Data Field	Data Type	Description
BARCODE	INT(11)	The type code of a stage in a document path
STATIONCODE	INT(11)	The identification number of the station
STEPORDER	INT(3)	The stage number of a document in a document path
TIMEIN	DATETIME	Checking in time of document in the station
TIMEOUT	DATETIME	Checking out time of document in the station
USERID	INT(11)	Identification number of user who checked in the document

TABLE 4: DOCUPATHLOC contains record of where the document as been through

Data Field	Data Type	Description
TYPECODE	INT(11)	Identification number of the document type
STATIONCODE	INT(11)	Identification number of the station
LEFTPOS	INT(11)	Left position of the graphical display of path in browser
TOPPOS	INT(3)	Top position of the graphical display of path in browser

TABLE 5: DOCUPATHSTATION contains the data needed for the document type path

Data Field	Data Type	Description
TYPECODE	INT(11)	Identification number of the document
STATIONCODE	INT(11)	Identification number of the station
METALEFTPOS	INT(11)	Left position of the graphical display of path in browser
METATOPPOS	VARCHAR(80)	Top position of the graphical display of path in browser

TABLE 6: DOCUMETAPATHSTATION contains the data needed for the document instantiation path

Data Field	Data Type	Description
TYPECODE	INT(11)	Identification number of the document type
CONSOURCE	INT(11)	Source station of the connection
CONTARGET	INT(11)	Target station of the connection
CONDESCR	INT(3)	Description of the connection

TABLE 7: DOCUPATHCON contains the connections needed for the document type path

Data Field	Data Type	Description
BARCODE	INT(11)	Identification number of the document
METACONSOURCE	INT(11)	Source station of the connection
METACONTARGET	INT(11)	Target station of the connection
METACONDESCR	VARCHAR(80)	Description of the connection

TABLE 8: DOCUPATHCON contains the connections needed for the document type path

Data Field	Data Type	Description
USERID	INT(11)	Identification number of the user
ACTIVE	INT	Determines if the user is still active or not
ROLE	ENUM(APPLICANT, DEPHEAD, DEPASSISTANT , SYSADMIN)	Role of the user
USERNAME	VARCHAR(20)	Desired name of the user
PASSWORD	VARCHAR(20)	Password of the user
PASSWORDHINT	VARCHAR(20)	Hint of the user's password
FNAME	VARCHAR(20)	First name of the user
LNAME	VARCHAR(20)	Last name of the user
EMAIL	VARCHAR(20)	Email address of the user
PHONE	VARCHAR(20)	Phone number of the user
ADDRESS	VARCHAR(20)	Address of the user
BARCODE	INT(11)	Identification number of applicants document
STATIONCODE	INT(11)	Identification number of the station of department chiefs and assistants

TABLE 9: USER contains the user's information

Data Field	Data Type	Description
PARAMCODE	INT(11)	Identification number of parameter
PARAMNAME	VARCHAR(20)	Name of parameter
TYPECODE	INT(11)	Identification number of document type
STRINGVAL	VARCHAR(40)	Holds default value of parameter
DATATYPECODE	INT(11)	Identification number of parameter's data type
INPUTTYPECODE	INT(11)	Identification number of parameter's input mode
DESC	VARCHAR(200)	Description of parameter

TABLE 11: PARAMETER contains parameter properties of document type

Data Field	Data Type	Description
METAPARAMCODE	INT(11)	Identification number of parameter
METAPARAMNAME	VARCHAR(20)	Name of parameter
BARCODE	INT(11)	Identification number of document
STRINGVAL	VARCHAR(40)	Holds default value of parameter
DATATYPECODE	INT(11)	Identification number of parameter's data type
INPUTTYPECODE	INT(11)	Identification number of parameter's input mode
DESC	VARCHAR(200))	Description of parameter

TABLE 12: METAPARAMETER contains parameter properties of document

Data Field	Data Type	Description
CHOICECODE	INT(11)	Identification number of parameter choice
CHOICENAME	VARCHAR(20)	Name of parameter choice
PARAMCODE	VARCHAR(20)	Identification number of parameter

TABLE 13: PARAMCHOICE contains parameter choices of document type

Data Field	Data Type	Description
METACHOICECODE	INT(11)	Identification number of document's parameter choice
METACHOICENAME	VARCHAR(20)	Name of document's parameter choice
METAPARAMCODE	VARCHAR(20)	Identification number of document's parameter

TABLE 14: PARAMCHOICE contains parameter choices of document type

Data Field	Data Type	Description
INPUTTYPECODE	INT(11)	Identification number of input type
INPUTTYPENAME	VARCHAR(20)	Name of input type

TABLE 15: INPUTTYPE contains input type choices of parameter

Data Field	Data Type	Description
USERID	INT(11)	Identification number of user
BARCODE	INT(11)	Identification number of document
STATUS	VARCHAR(20)	Status if the document is locked to a particular user

TABLE 16: DOCLOCK contains permission of a user for document manipulation, if locked then only the user that manipulated it can edit it.

Data Field	Data Type	Description
USERID	INT(11)	Identification number of data type
PARAMCODE	INT(11)	Identification number of parameter code
ALLOW	BOOLEAN	Determines if user can manipulate data in the parameter

TABLE 17: PERMISSION contains permission of a user for parameter manipulation

Data Field	Data Type	Description
USERID	INT(11)	Identification number of data type
METAPARAMCODE	INT(11)	Identification number of parameter code
ALLOW	BOOLEAN	Determines if user can manipulate data in the parameter

TABLE 18: PERMISSION contains permission of a user for parameter manipulation of document

C. Context Diagram

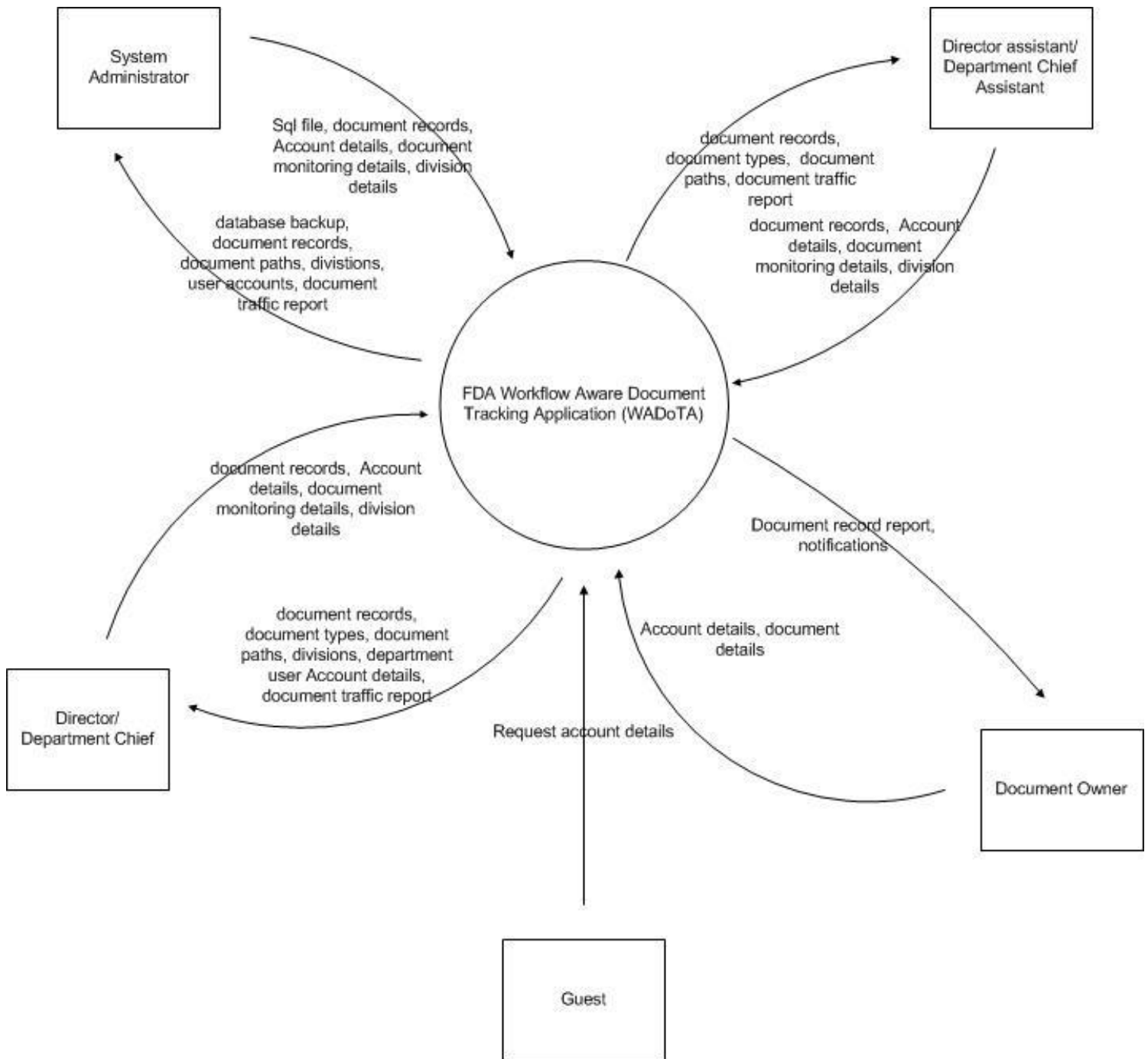


Figure 2: CONTEXT DIAGRAM, FDA Workflow Aware Document Tracking System (WADoTS)

D. Data Flow Diagram

Top-level

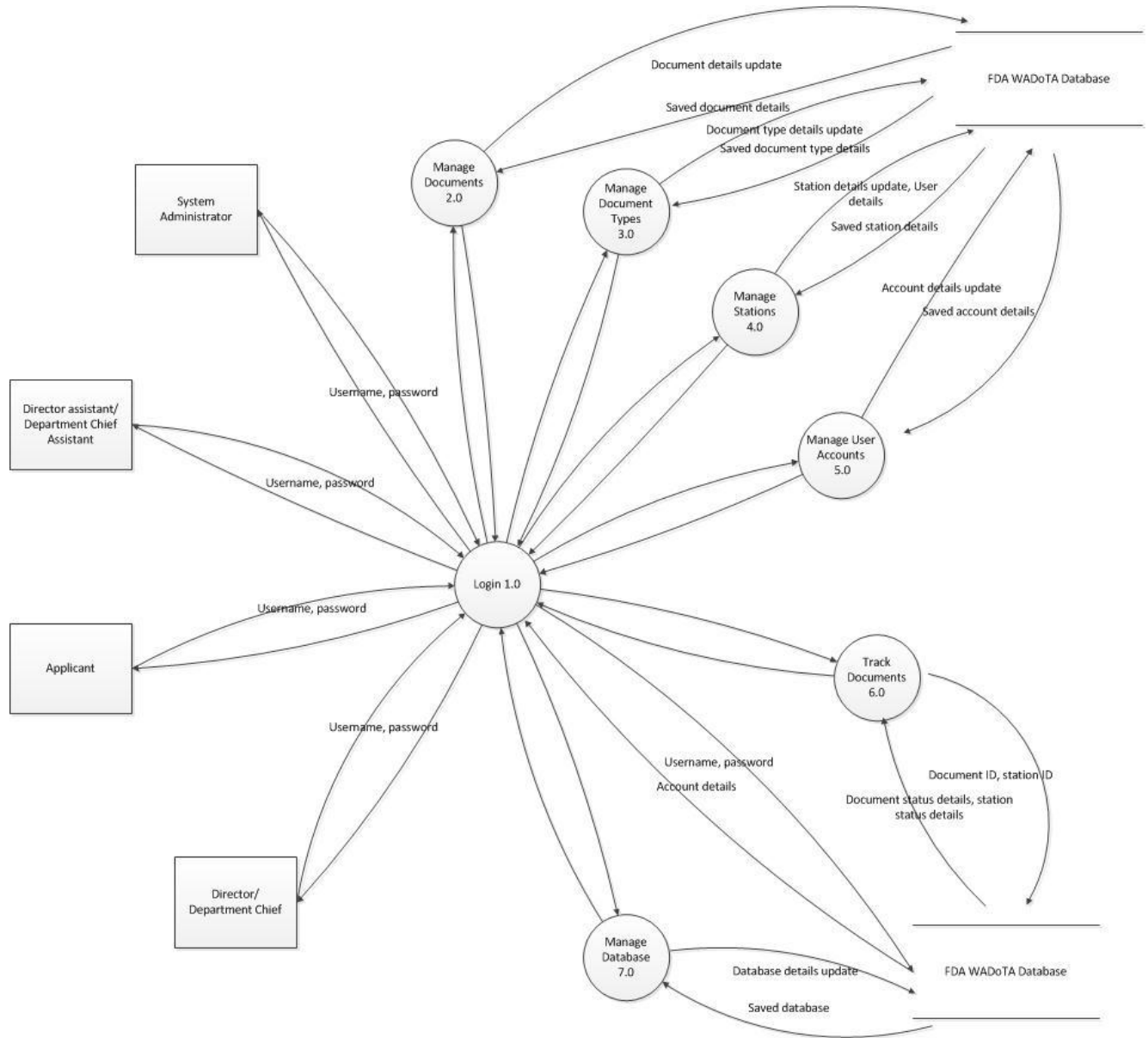
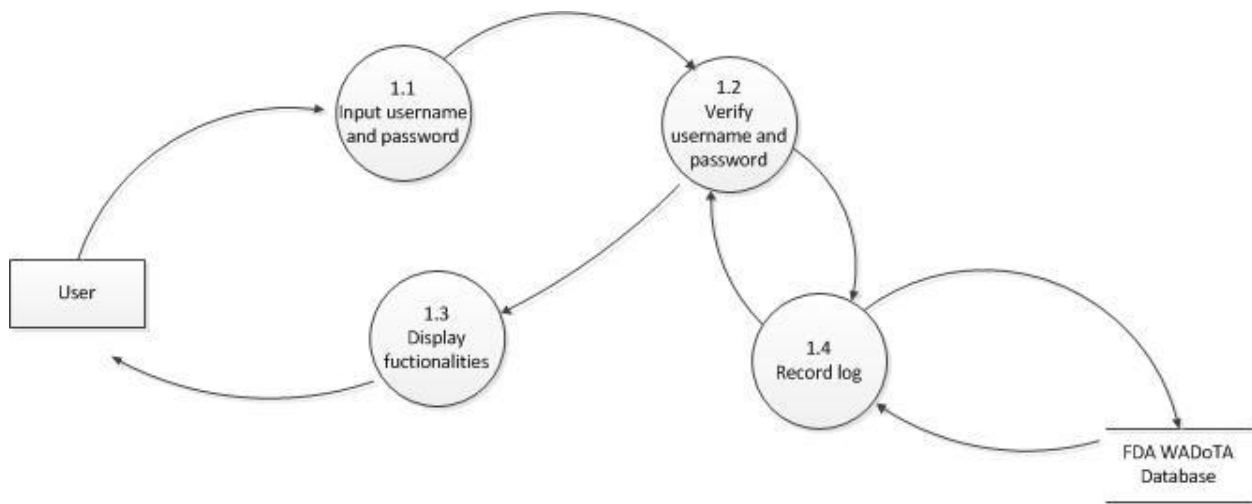


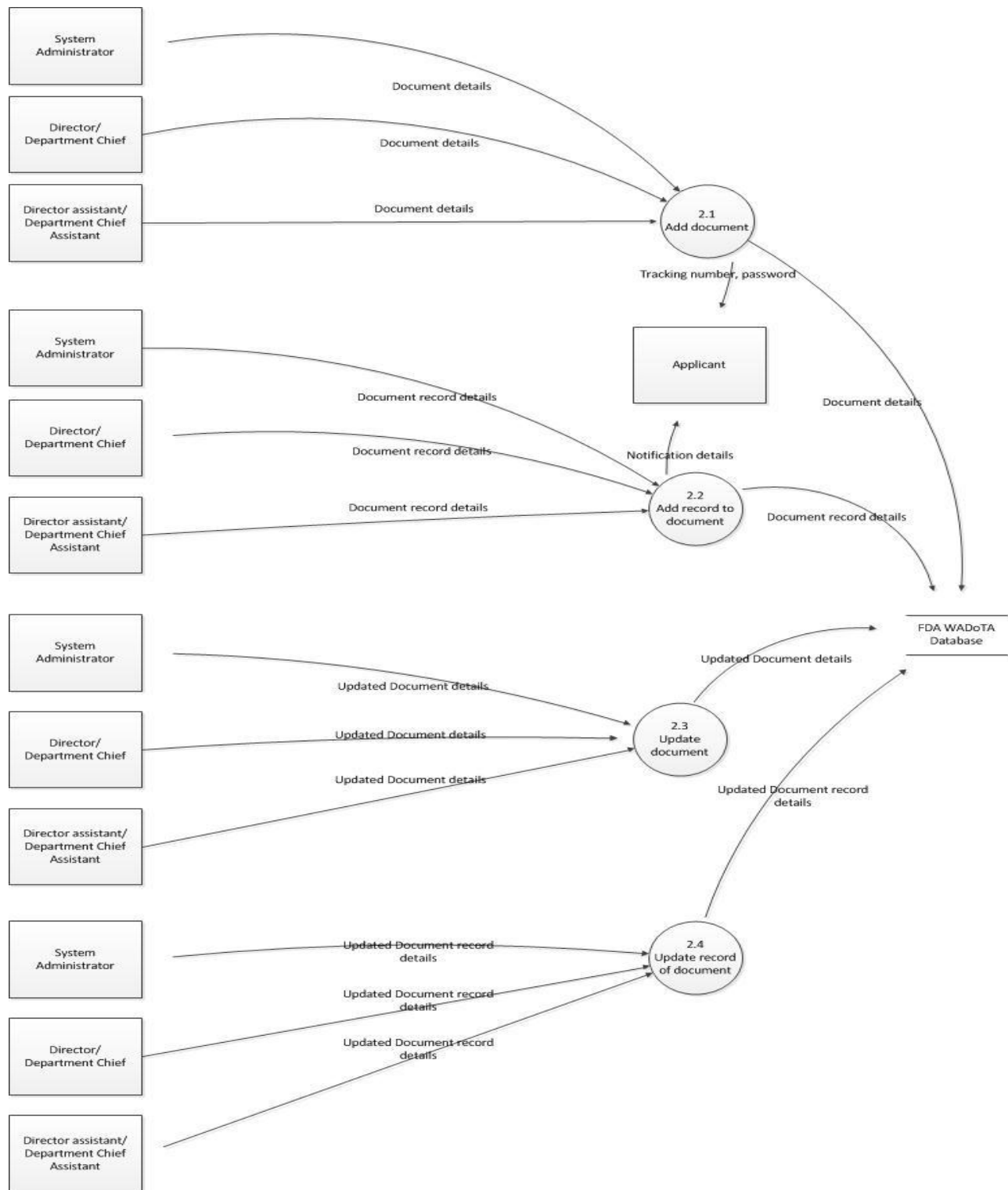
Figure 3: DATA FLOW DIAGRAM, FDA Workflow Aware Document Tracking System (WADoTS)

Data Flow Diagram Sub-explodings:

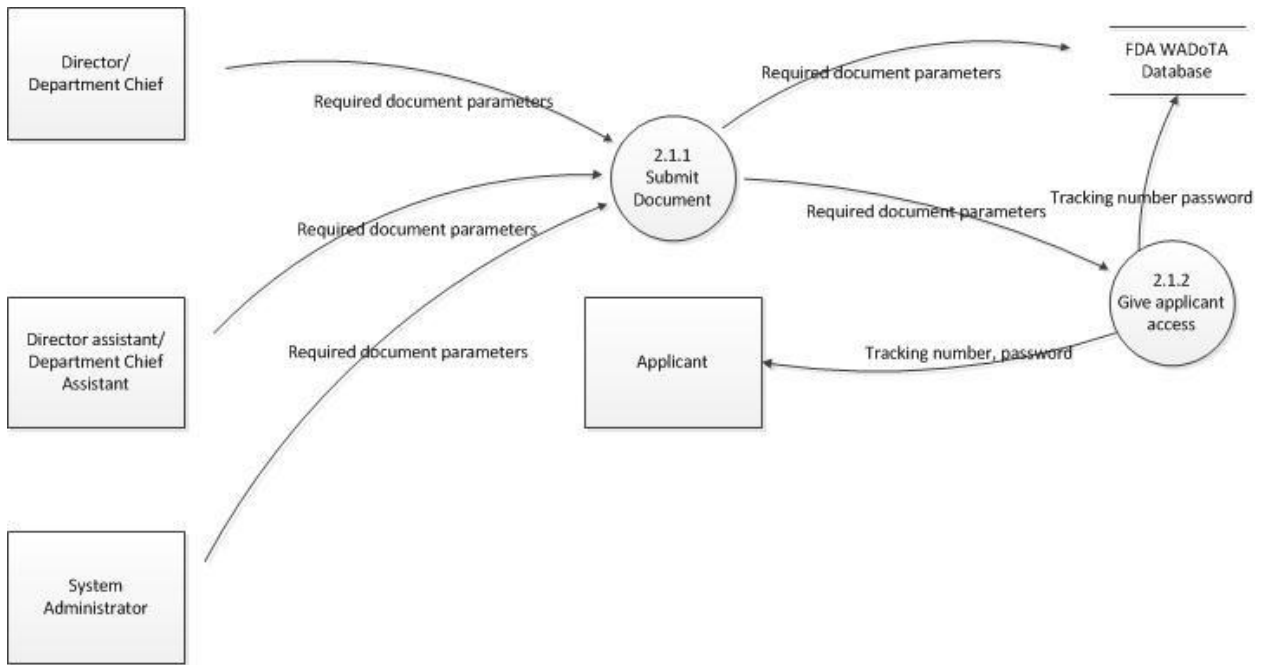
Sub-exploding: Level 1 - 1.0 Login



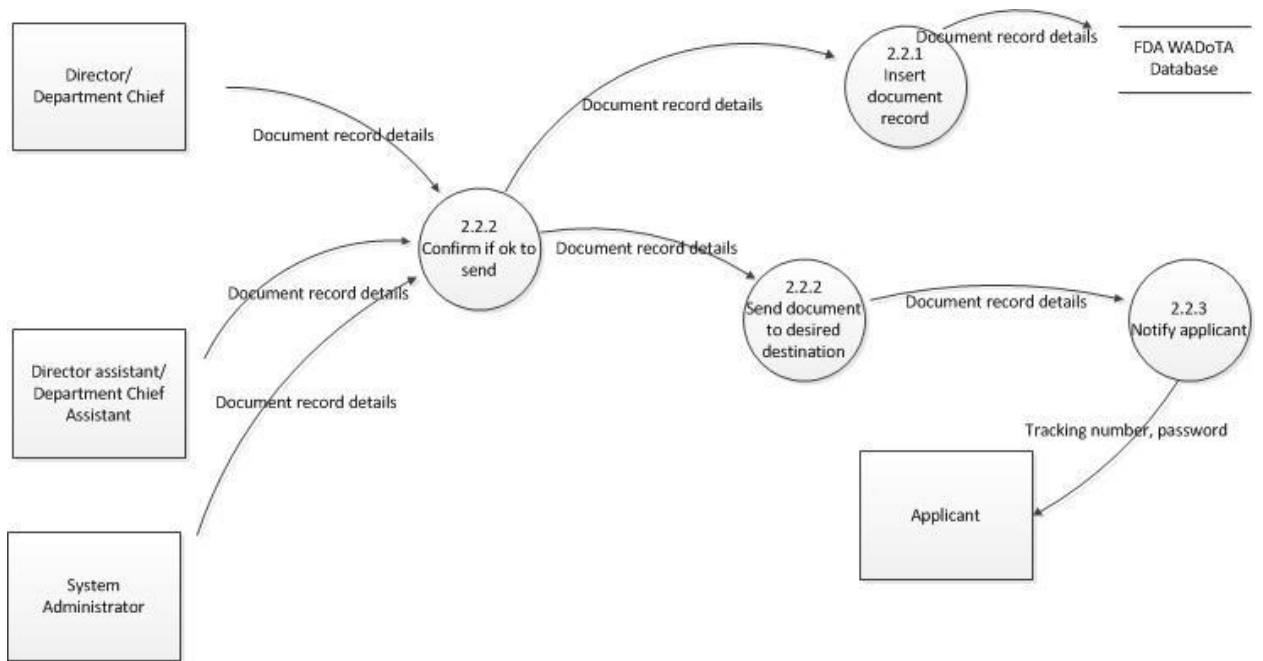
Sub-explosion: Level 1 – 2.0 Manage Documents



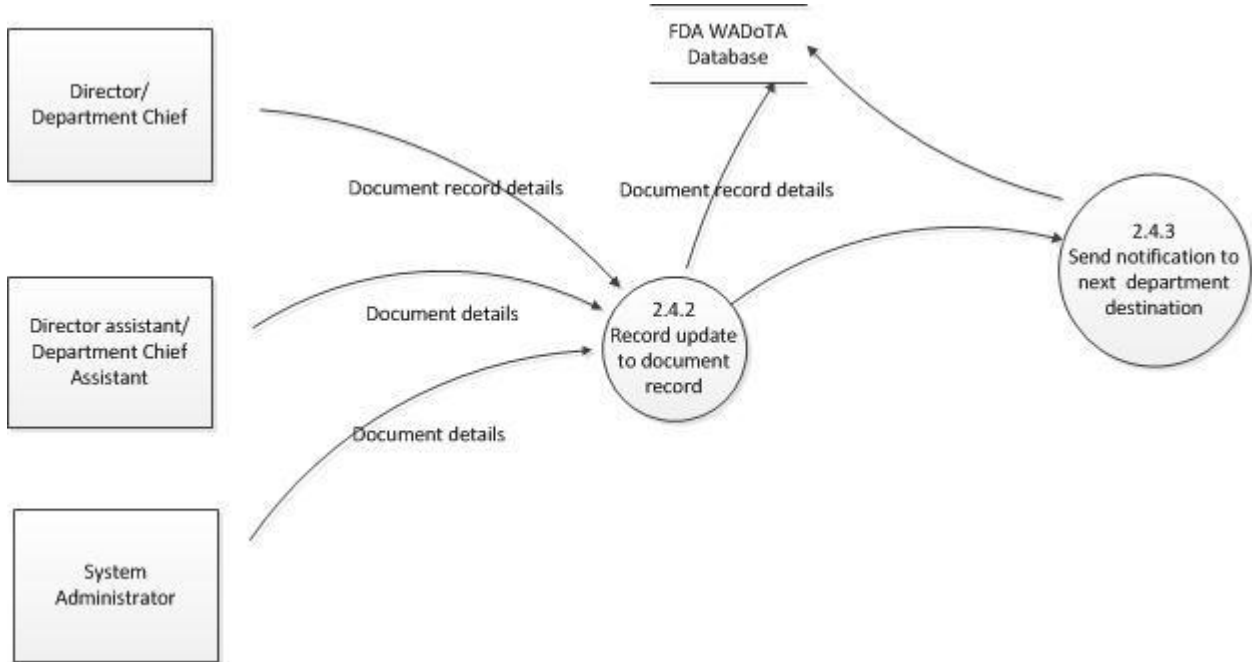
Sub-explosion: Level 2 – 2.1 Add document



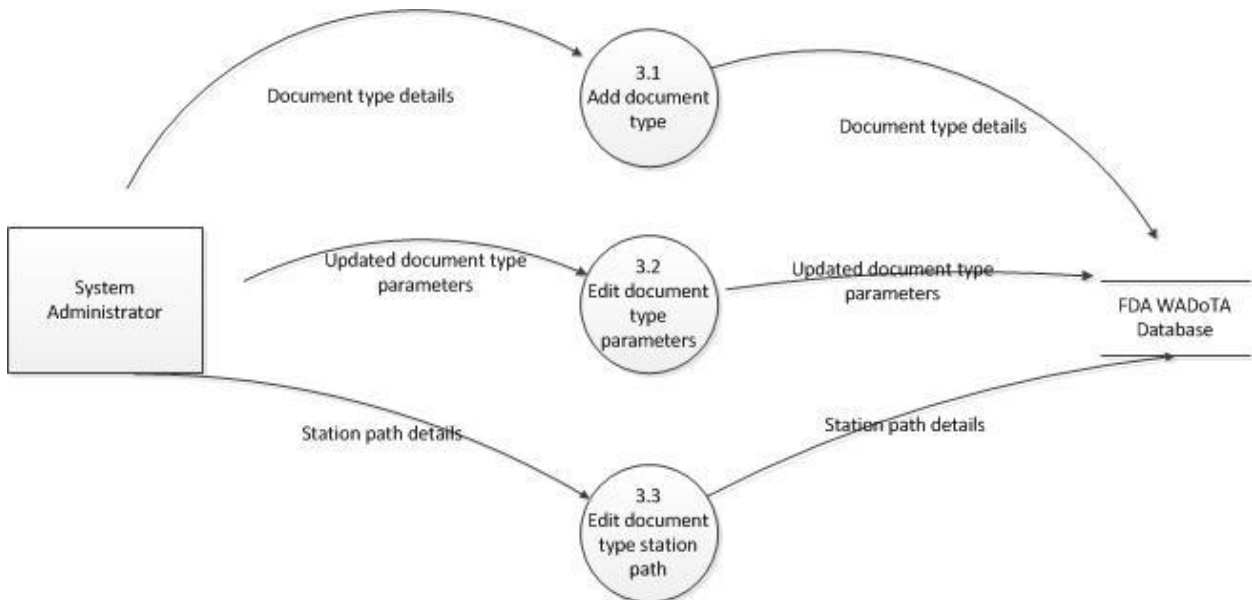
Sub-explosion: Level 2 – 2.2 Add record to document



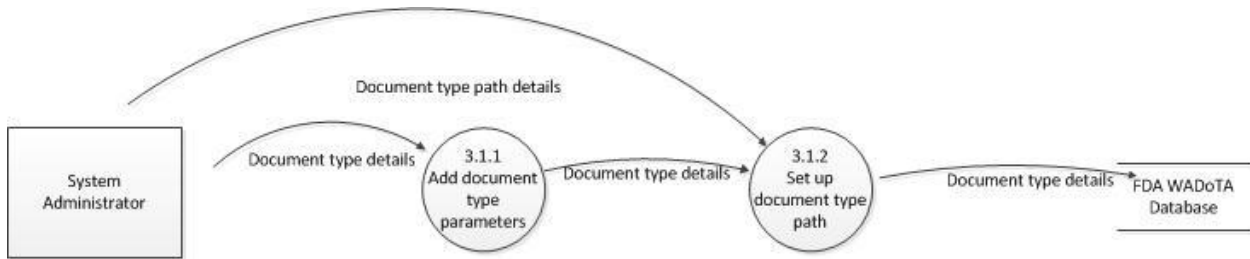
Sub-explosion: Level 2 – 2.4 Update document record



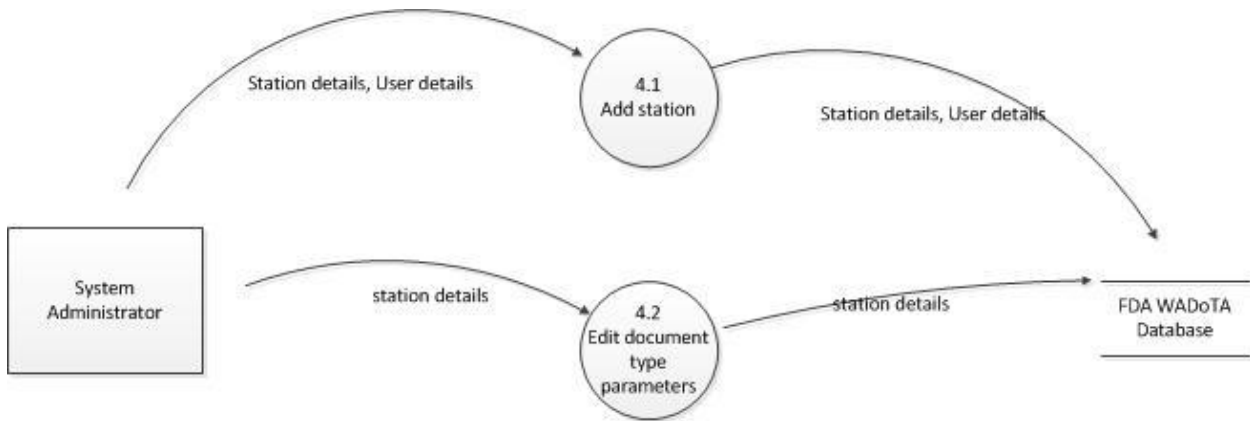
Sub-explosion: Level 1 – 3.0 Manage document types



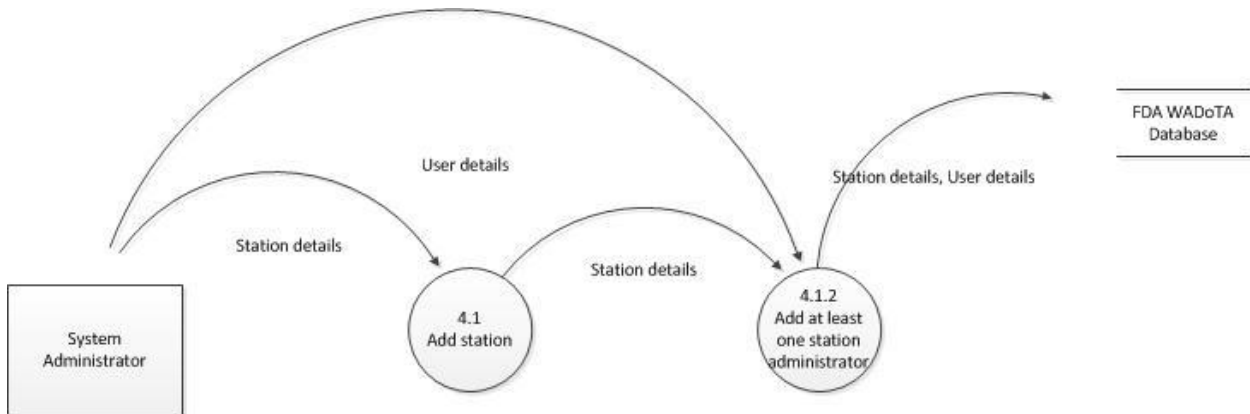
Sub-explosion: Level 2 – 3.1 Add document type



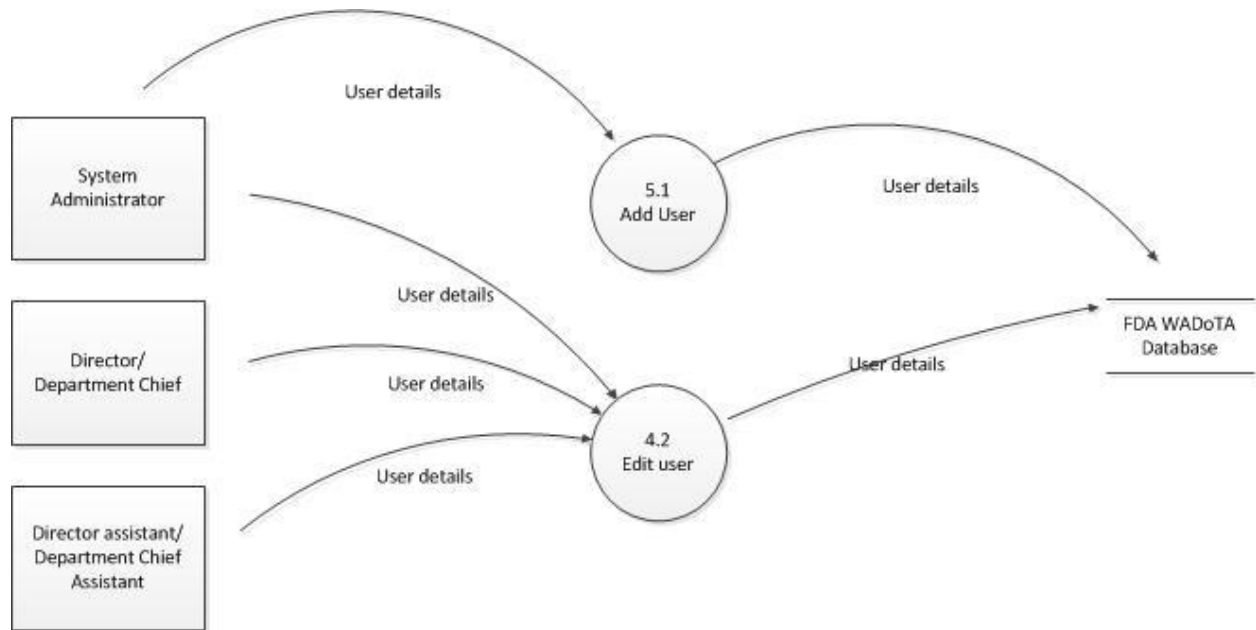
Sub-explosion: Level 1 – 4.0 Manage Stations



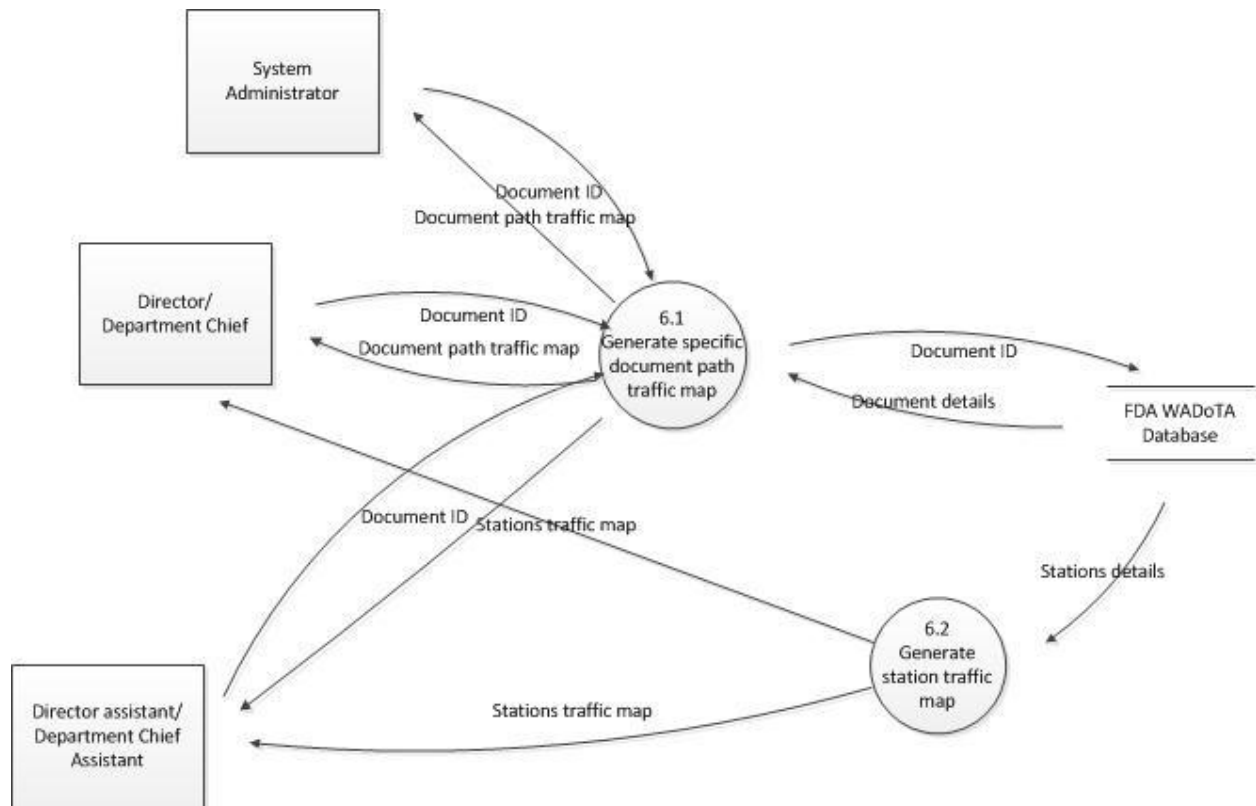
Sub-explosion: Level 2 – 4.1 Add station



Sub-explosion: Level 1 – 5.0 Manage user accounts



Sub-explosion: Level 1 – 6.0 Track documents

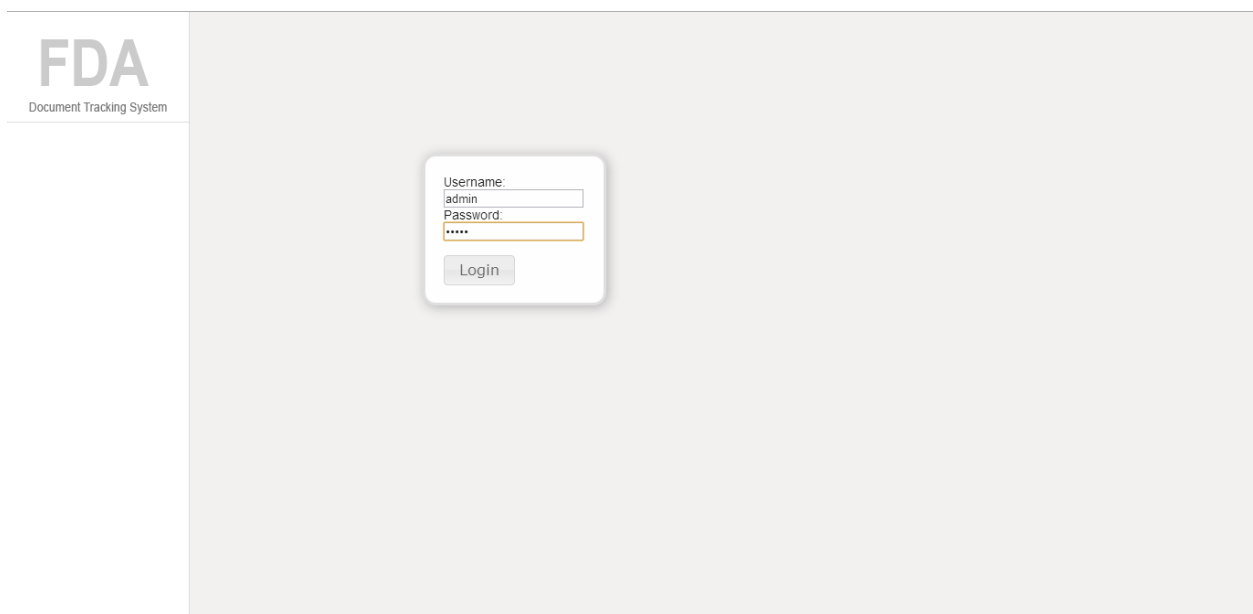


V. Results

5.1 Administrator Functionalities

5.1.1 Log in and manage account

Initially, users were directed to the login page as shown in Figure 5.1.1. The system allows them to login as administrator, head of a department and assistant of a department.



The screenshot shows the login page for the FDA Document Tracking System. On the left side, there is a logo with the text "FDA" in large letters and "Document Tracking System" below it. The main area of the page is a light gray background. In the center, there is a white login form with a gray border. The form contains two input fields: "Username:" with the text "admin" entered, and "Password:" with five dots representing a masked password. Below the password field is a "Login" button.

Figure 5.1: Login Page

When logged in to the system, the user was redirected to the homepage with a side panel. If the user was a system administrator, side panel displayed navigations for home, my profile, documents, document types, stations, users, companies and logout. (shown in Figure 52).

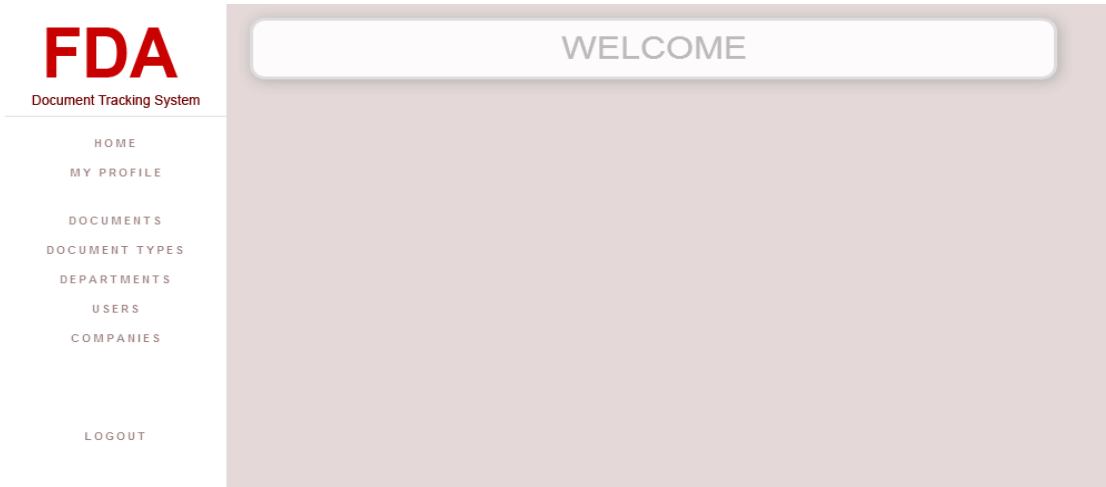


Figure 5.2: Home Page

The system provided an interface wherein the users had the ability to edit their account information and change the theme for their account's interface.

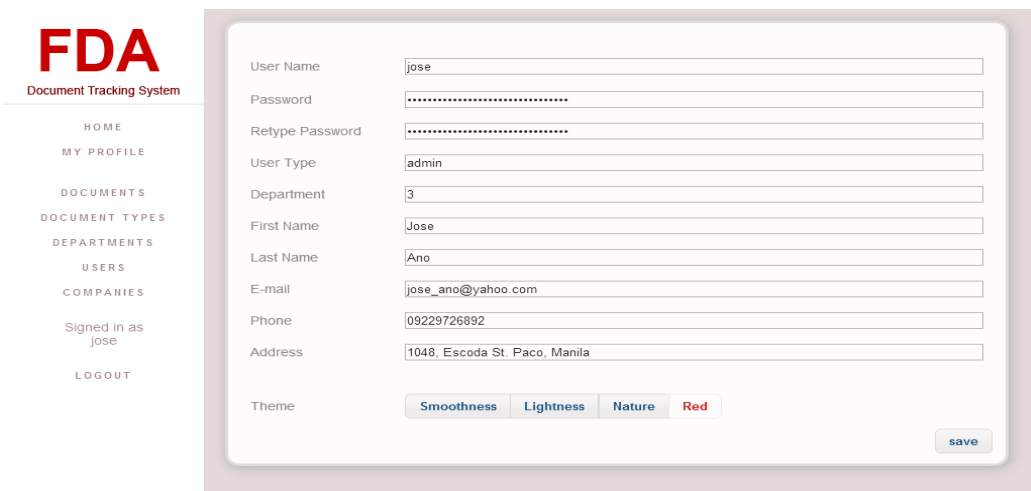


Figure 5.3: Manage account

5.1.2 View/add/edit/delete department

A non-applicant user was able to view all departments. Congestion of each departments were shown by its number of documents. Green ment that there were just a few documents in a department. The nearer it got to the shade of red, the more the documents observed in Figure 5.4.

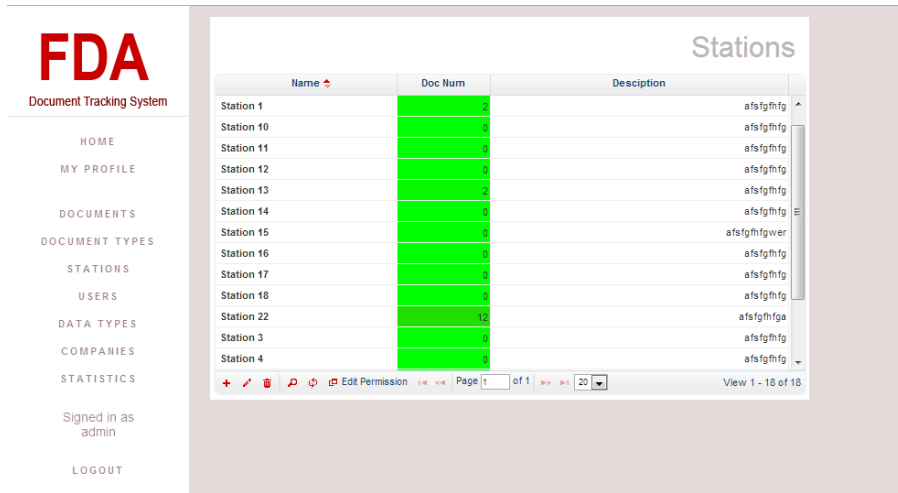


Figure 5.4: View all departments

An admin user was able add, edit and delete departments as seen in Figure 5.5 to Figure 5.7.

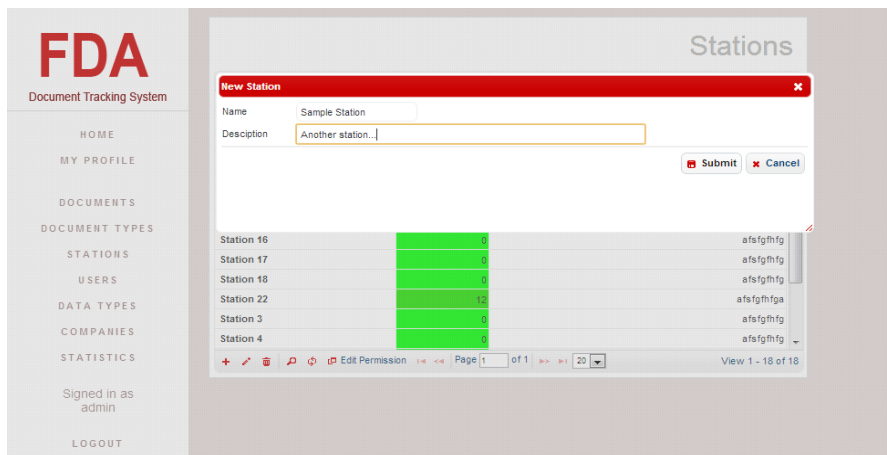


Figure 5.5: Add department

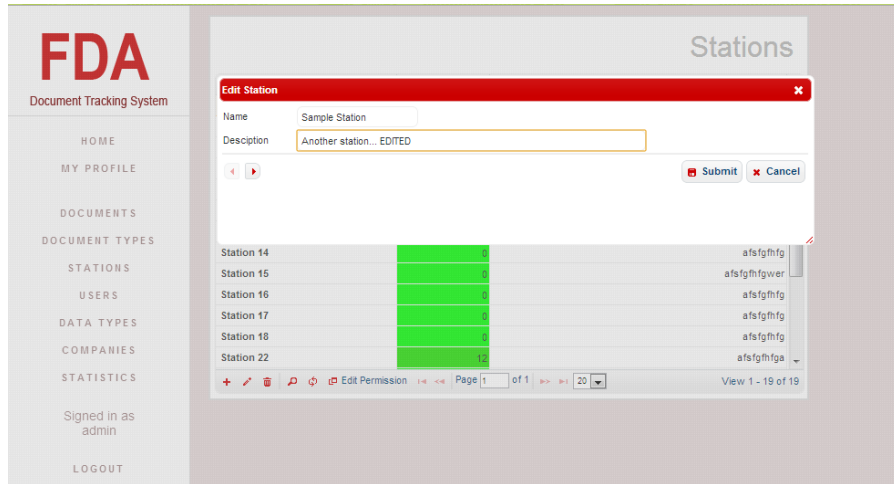


Figure 5.6: Edit department

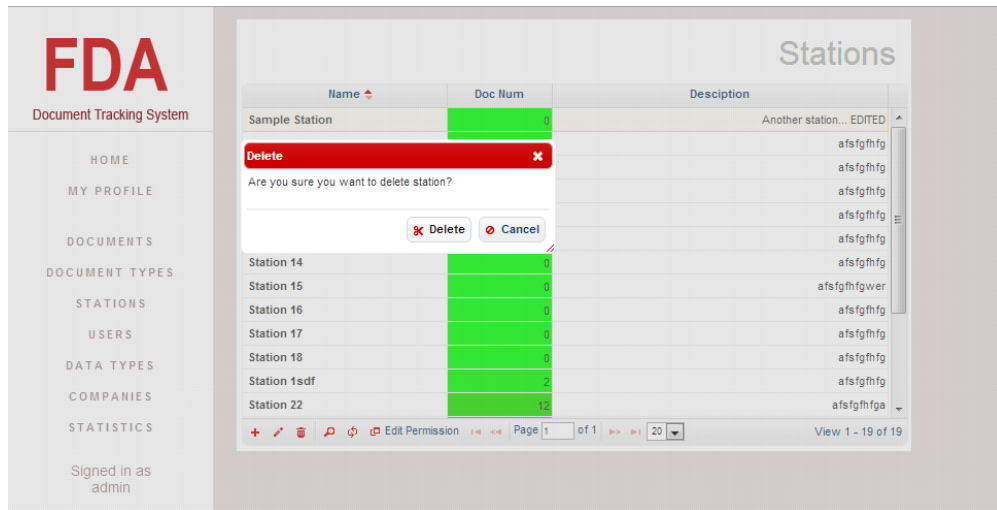


Figure 5.7: Delete department

5.1.3 View all users in a department

Users for a particular station were viewed by an administrator in two different ways. First was by filtering rows of the table of all users by station (Figure 5.8). Second was by opening a particular station (Figure 5.9).

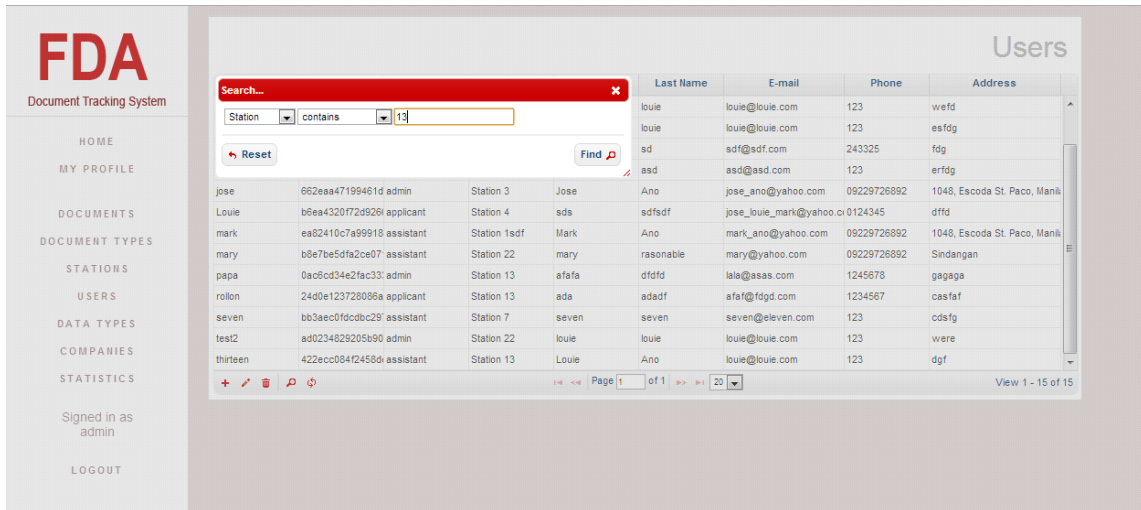


Figure 5.8: View all users in a particular station by user list filtering

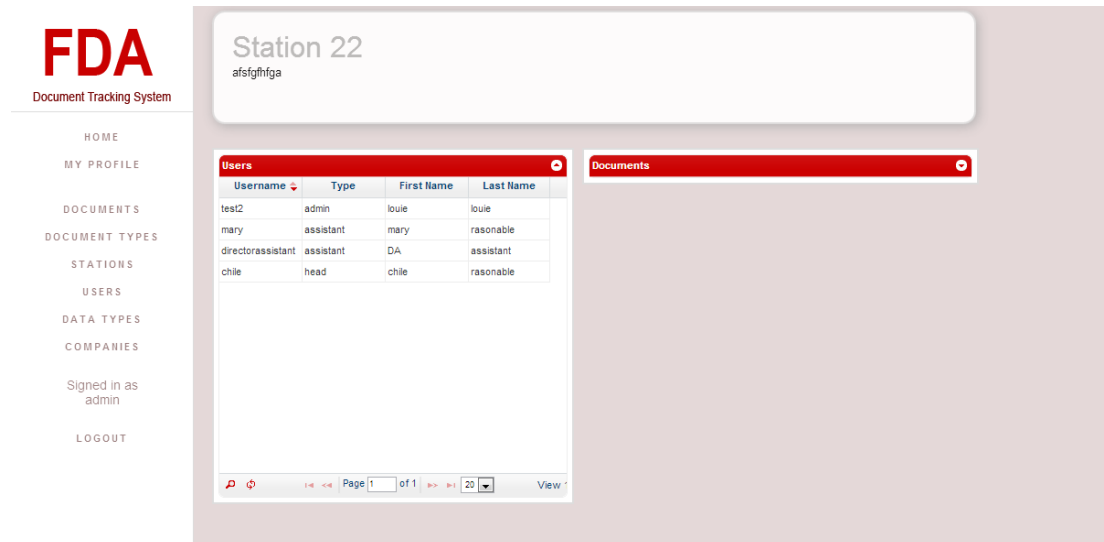


Figure 5.9: View all users in a particular station by viewing a department

5.1.4 View/add/edit/delete user account

All user accounts were viewed by the administrator (Figure 5.10). Department heads were allowed to view users in their department.

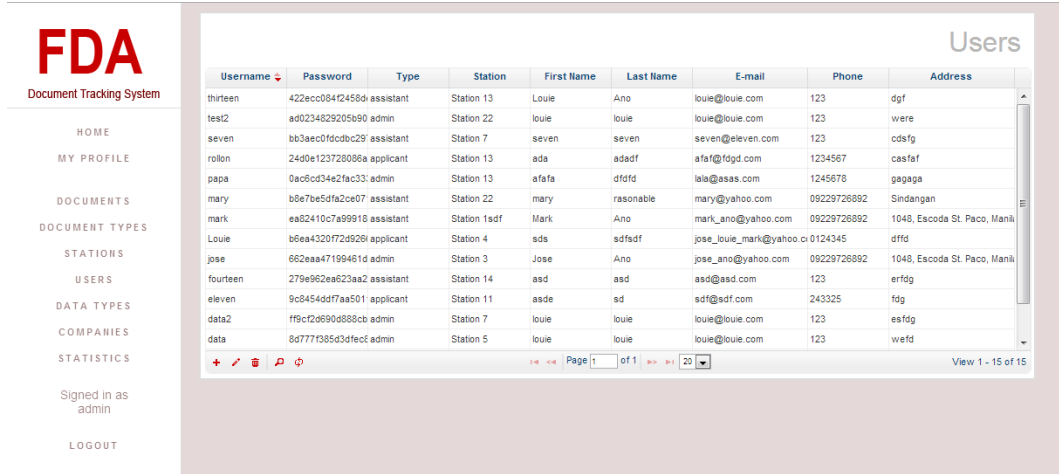


Figure 5.10: View user accounts

Administrators had the ability to add, edit and delete accounts as observed in Figure 5.11 to Figure 5.13. Department heads were able edit users that are in their department.

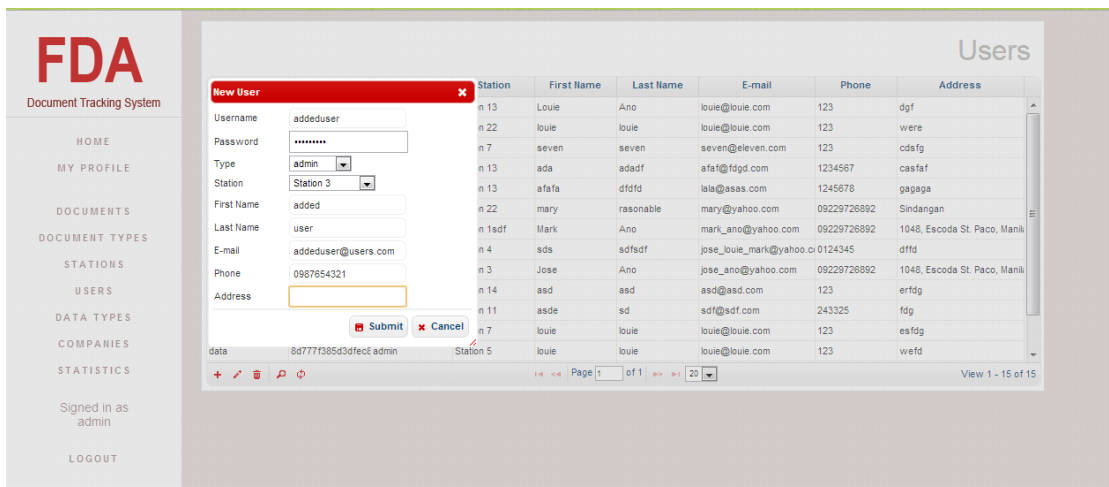


Figure 5.11: Add user account

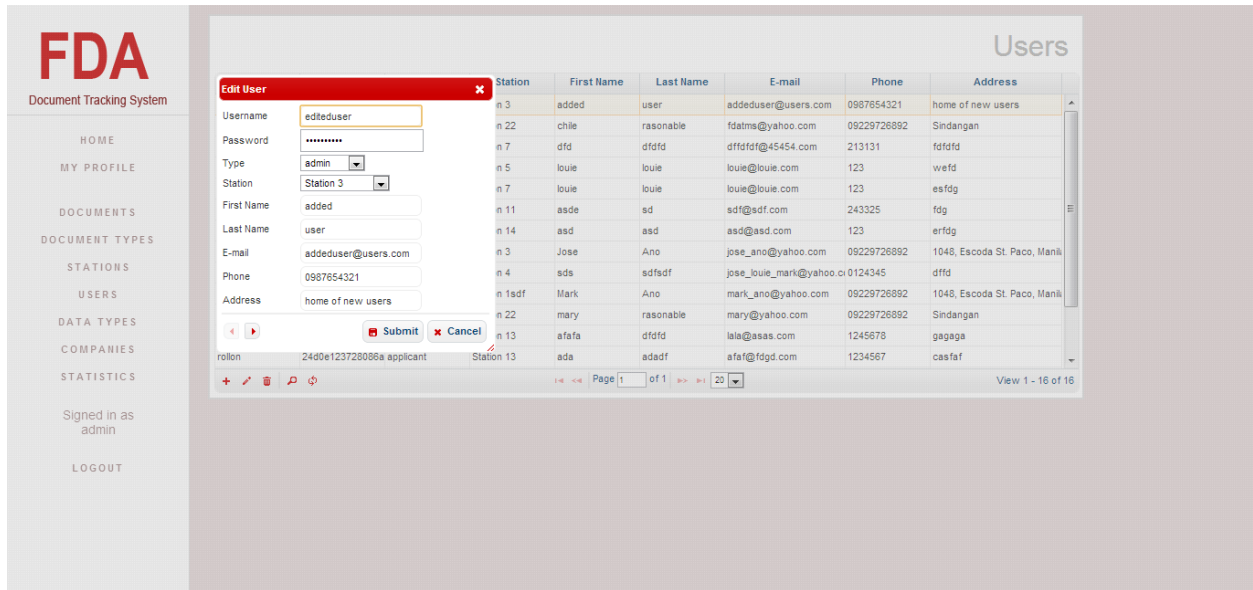


Figure 5.12: Edit user account

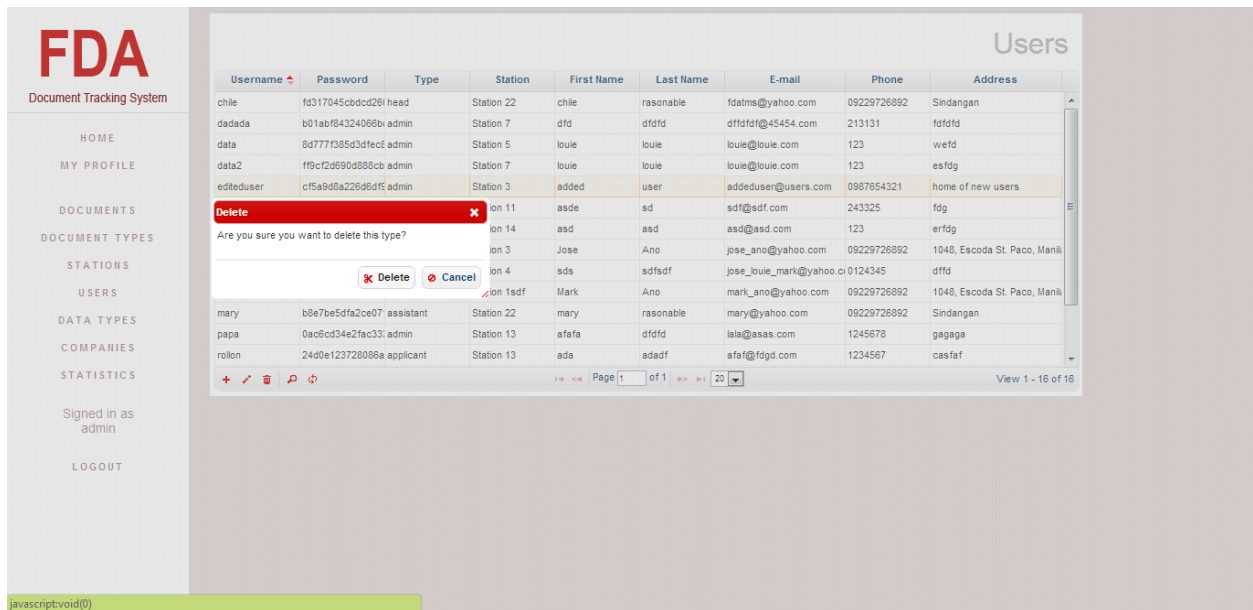


Figure 5.13: Delete user account

5.1.5 View/add/edit/delete document record type

An administrator was able to view the list of document types.

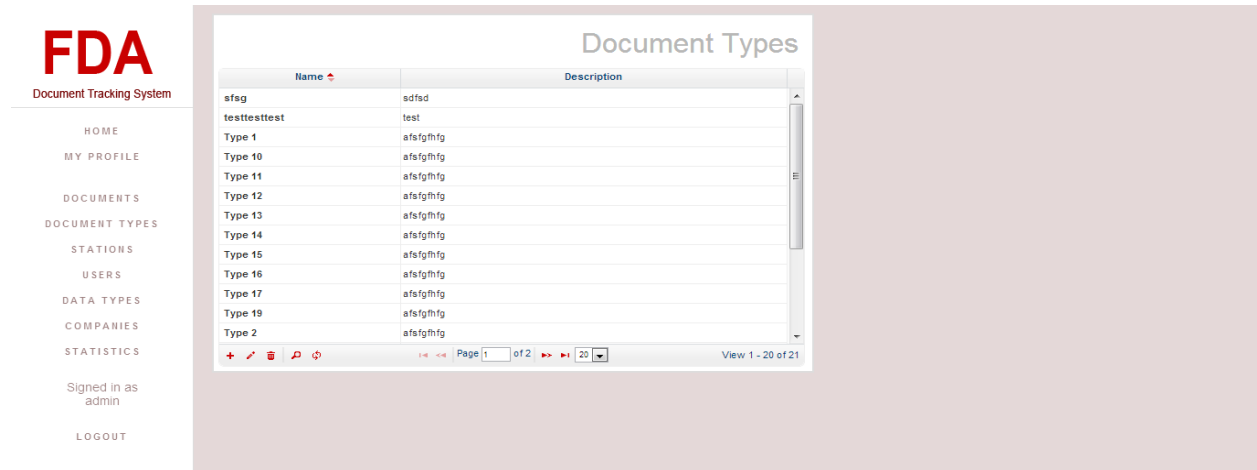


Figure 5.14: View document types

An administrator was responsible for adding a document type. Adding of document type involved several steps. The first step was the setting of document type meta data which included the name and the description of the document type (see Figure 5.15).

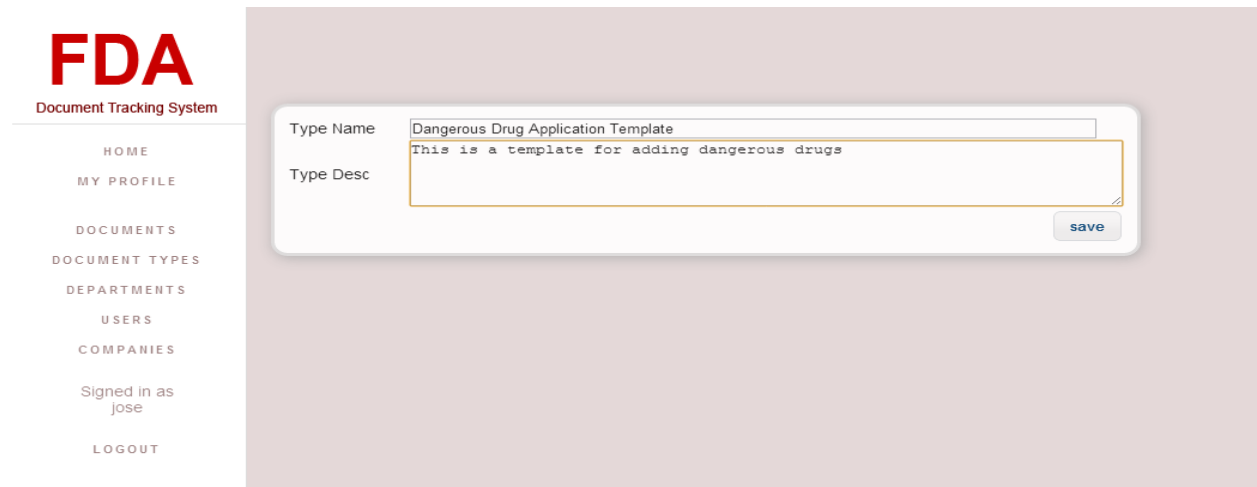


Figure 5.15: Add new document template - Metadata

The next step of setting up the document was defining its parameters (see Figure 5.16). Adding a parameters required administrators to enter an order number, name, input type, data type, options and default value. Options inputted as a string and each option was separated by the ‘&’ sign.

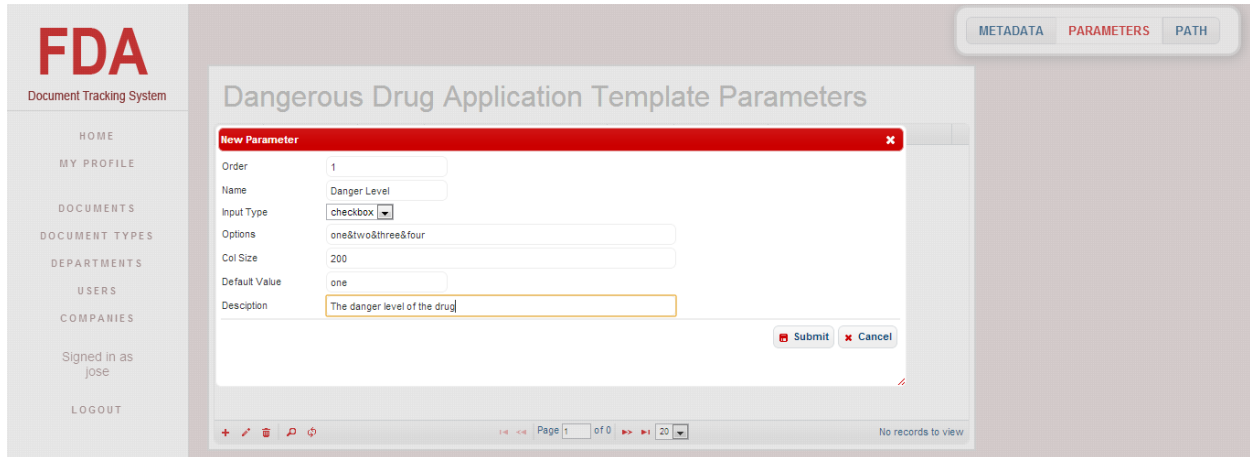


Figure 5.16: Add new document type - Parameters

The next step was the setting up of the document type’s workflow path which also involves several steps.



Figure 5.17: Add new document type - Path

To set up a document type's workflow, departments were added. These departments served as the nodes of the workflow illustration setup. A department was able to be deleted after it was added by clicking the black box at the right side of the department.



Figure 5.18: Path – Adding departments for document workflow

The departments were connected together by lines that would represent the possible paths of document in the workflow. In connecting two departments, a connection name was indicated to describe the nature of the connection.

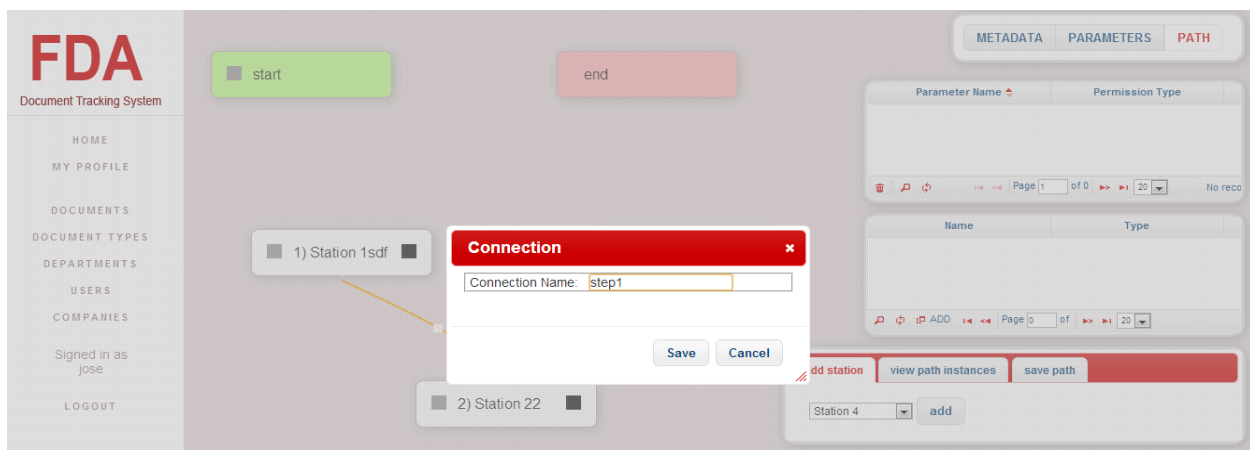


Figure 5.19: Path – Adding connection to the departments

The workflow path from start to end must be well defined before it can saved. A well defined workflow had all of its nodes connected from the start node to the end node. An example of this is illustrated in Figure 5.20.

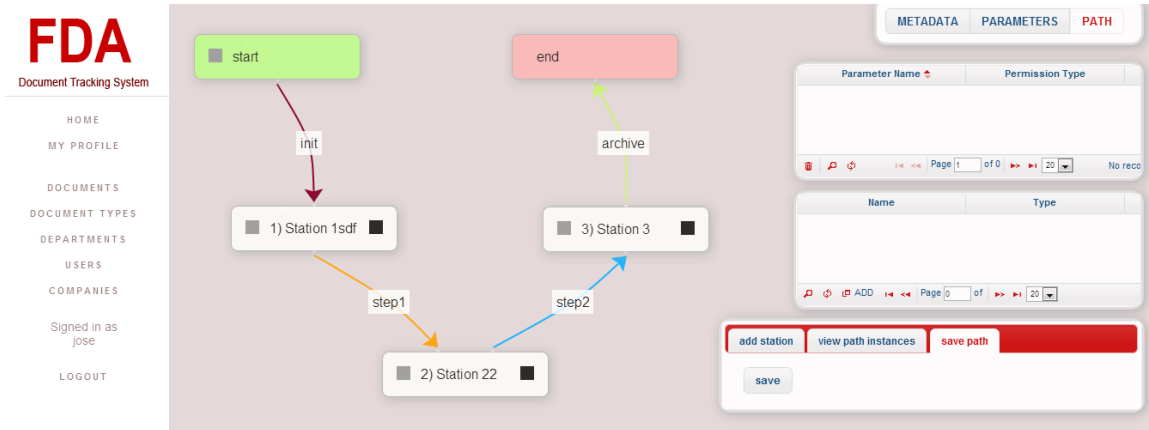


Figure 5.20: Path – Well-defined workflow.

The last step in setting up a document type was to add permissions to parameters with respect to department or station. When a station was selected from the diagram, parameters of the document type were displayed in the left panels. From there, permissions per parameter and station were selected (See Figure 5.21). Create permission was edited by ticking the “create permission” button.

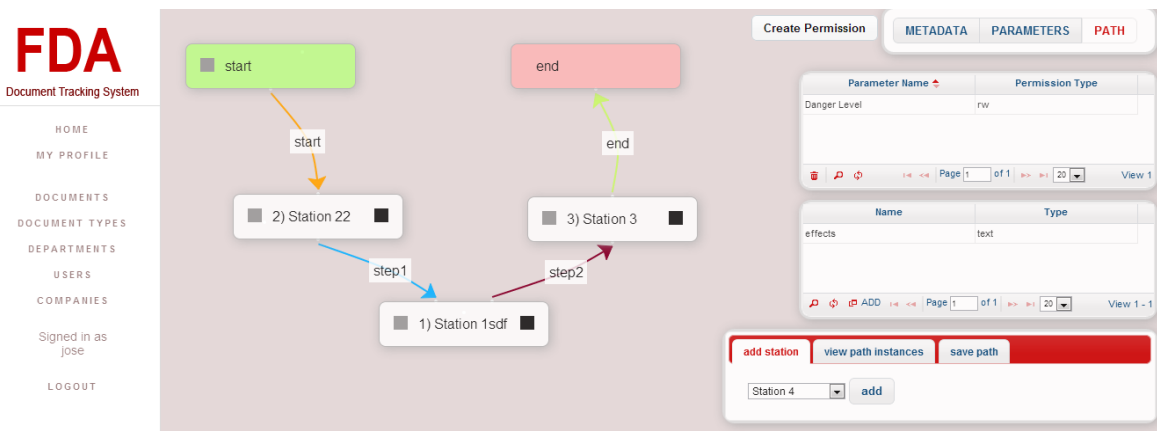


Figure 5.21: Path – Set up permissions.

When the workflow already had permissions and was well defined, it can already be saved.

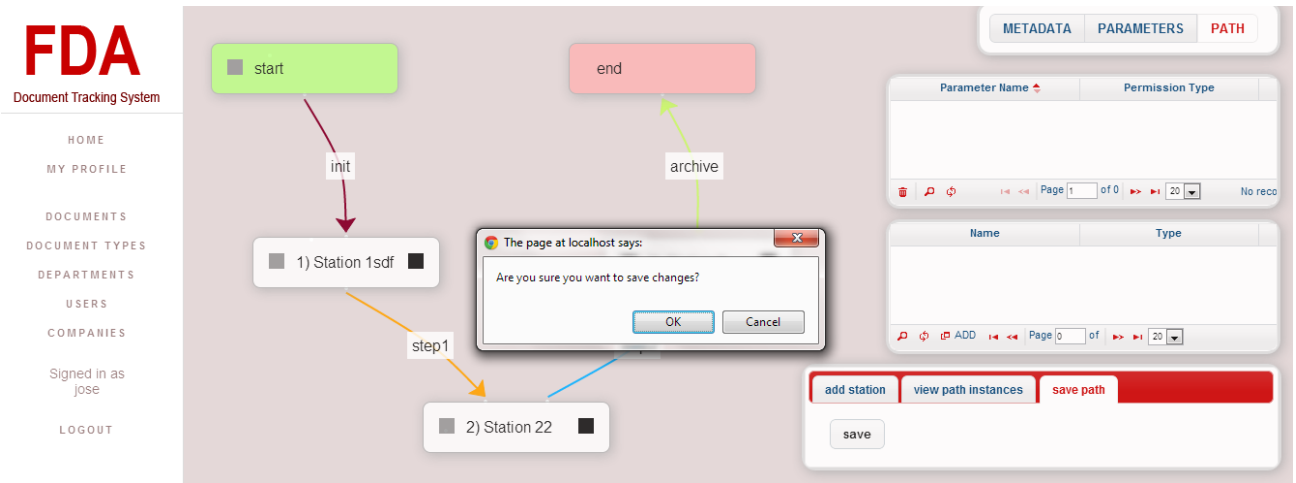


Figure 5.22: Path – Save workflow

Administrators were also the ones editing the document type. Steps in editing was the same as adding a document type, although some steps can be skipped since workflowpath and parameters were already defined. Editing document types won't affect exiting document instances. Documents with an edited document type still had the latter properties that the document type had before the edit. Figure 5.23 shows how metadata is edited.

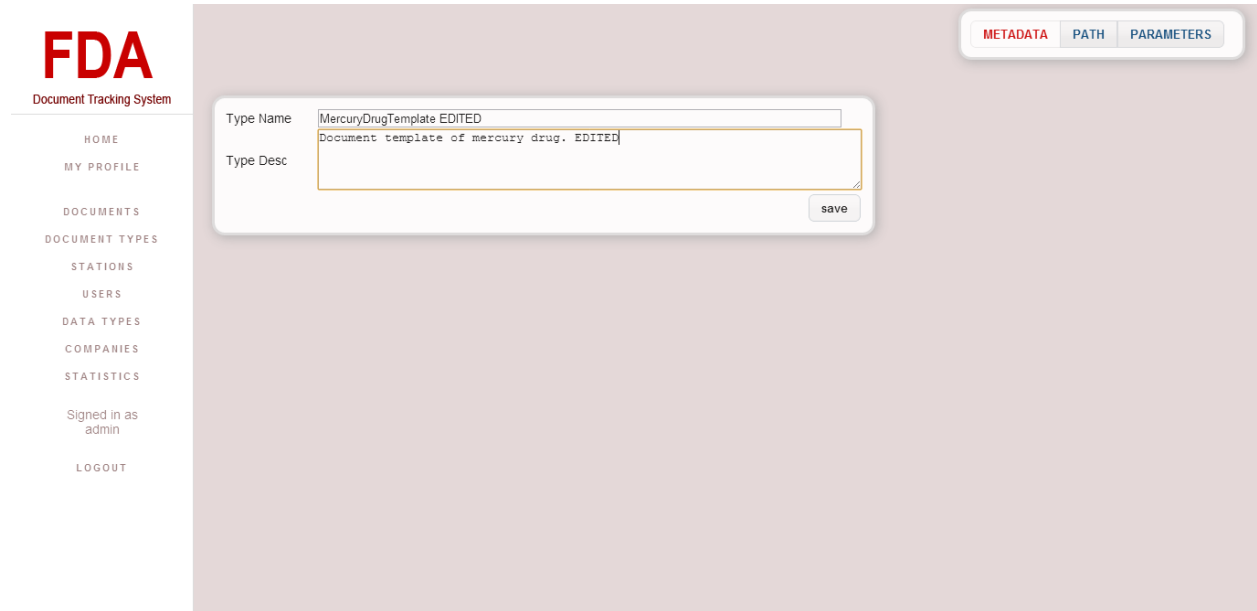


Figure 5.23: Edit document type

Document types can be deleted by an administrator. Document instances of a deleted document type would not be deleted.

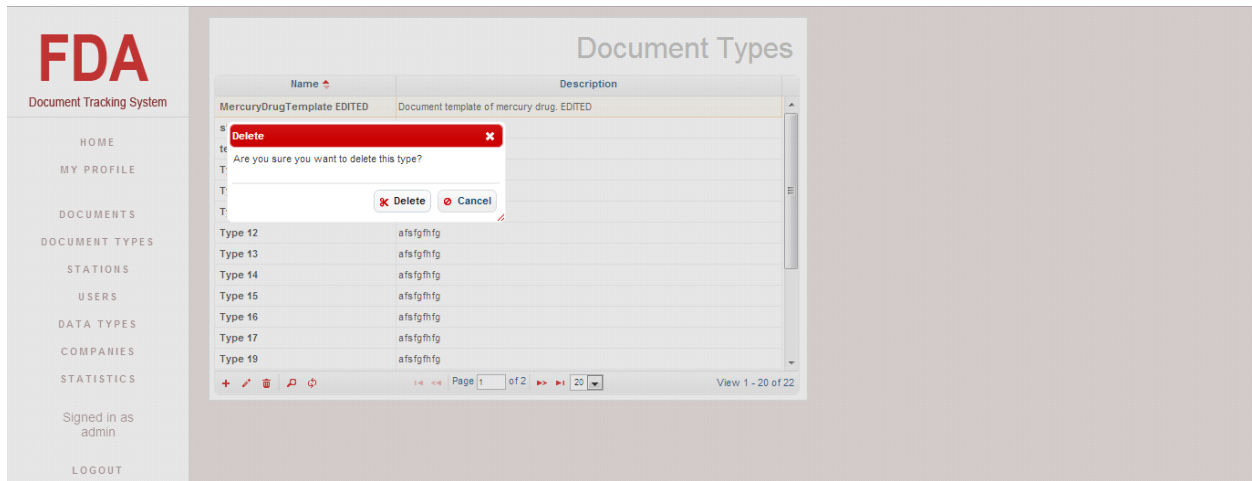


Figure 5.24: Delete document template

5.1.6 View/add/edit/delete document record

An administrator and department head was able see all documents. In the documents list page, the users viewed an overview of the document’s status. In Figure 5.25, location, time elapsed and progress percentage were displayed which showed the document’s status. Location was the current location of the document. Time elapsed was the difference between the current time and the time the document was created. It’s format is [year:month:day:hr]. Progress was the percentage of completion of the document. It was calculated by counting the number of minimum steps from the start of the workflow path to the current department over the total minimum number of steps required to finish the document.

Lock	Barcode	Title	Type	Location	Time Elapsed	Progress	Companies
<input type="checkbox"/>	101	ApproveDocument	Type 1	Station 9	00:00:22:06	50%	[Company 1]
<input type="checkbox"/>	100	DocumentToBeApprov	Type 3	Station 22	00:00:22:06	0%	[Company 1]
<input type="checkbox"/>	99	SampleDocument	Type 1	Station 9	00:00:22:07	50%	[Company 1]
<input type="checkbox"/>	95	testtest	Type 3	Station 22	00:00:24:20	0%	[Company 2]
<input type="checkbox"/>	94	y	Type 3	Station 14	00:00:26:03	33%	[Company 2]
<input type="checkbox"/>	91	josejose	Type 3	Station 13	00:00:26:05	67%	[Company 3]
<input type="checkbox"/>	90	joselouie	Type 3	Station 13	00:00:26:05	67%	[Company 2] [Company 5]
<input checked="" type="checkbox"/>	88	Mark	Type 3	Station 22	00:00:26:09	0%	[Company 2]
<input type="checkbox"/>	86	Ironic	Type 3	Station 22	00:00:26:22	0%	[Company 2]
<input type="checkbox"/>	85	Chance	Type 1	Station 1sdf	00:00:27:01	0%	[Company 3] [Company 5]
<input type="checkbox"/>	84	tst1	Type 1	Station 1sdf	00:00:29:03	0%	[Company 1] [Company 4]
<input checked="" type="checkbox"/>	81	testing2	Type 3	Station 22	00:00:29:05	0%	[Company 4]
<input type="checkbox"/>	82	testing3	Type 3	Station 22	00:00:29:05	0%	[Company 3]
<input type="checkbox"/>	83	testing4	Type 3	Station 22	00:00:29:05	0%	[Company 2]
<input type="checkbox"/>	78	test13	Type 3	Station 22	00:01:00:03	0%	[Company 2] [Company 4]
<input checked="" type="checkbox"/>	79	test14	Type 3	Station 22	00:01:00:03	0%	[Company 1] [Company 2] [Company 3]
<input type="checkbox"/>	5	five	Type 4	Station 22	01:10:15:00	0%	[Company 40]

Figure 5.25: View all document records

Administrators, department heads and assistants were able to add a document. Although department heads and assistants needed to have permission set up by the administrator to do so. In setting up a document, title, description and type was required. To help the user select the appropriate document type, format and instances of a document type were viewed when the document type was selected.

The screenshot shows the 'Add document record' form in the FDA Document Tracking System. The interface includes a sidebar with navigation links and a main form area. The form is divided into two sections: 'format' and 'instances'.

Form Fields:

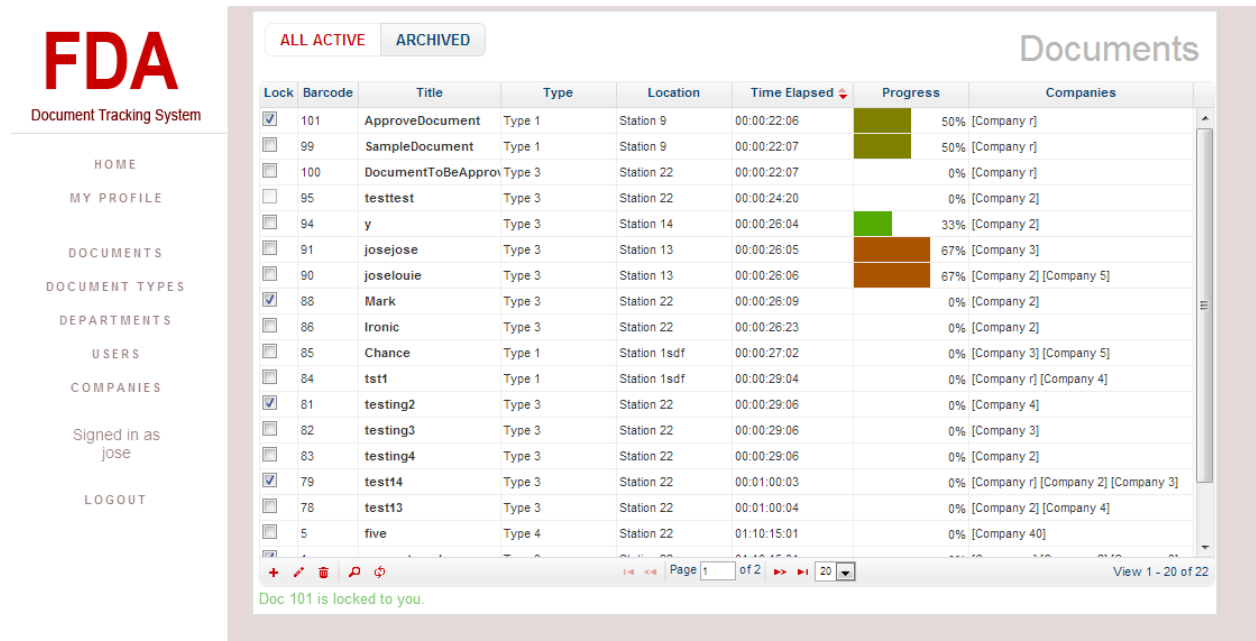
- Title:** SampleDocument
- Description:** A sample document record.
- Type:** A dropdown menu with options: Type 1, Type 2, Type 3, Type 4, Type 5. Type 1 is selected.
- Company:** A dropdown menu with options: Company 2, Company 3, Company 4, Company 5. Company 2 is selected.
- Company r:** A dropdown menu with the option: Company r.
- save:** A button to save the record.

Instances Section:

- dsfsd:** A dropdown menu with the option: three.
- radio:** A group of radio buttons with options: four, three (selected), two, one.
- wer:** A text input field with the value: 12.
- textarea:** A text area with the value: test.
- dsf1:** A text input field with the value: 12.
- hgfh:** A text input field with the value: 12.
- jaja:** A text input field with the value: 12.
- thhf:** A text input field with the value: 2.
- sdfgdh:** A text input field with the value: 12.
- sdfdg:** A text input field with the value: fgdhgd.
- dfg:** A text input field with the value: 12.
- dsd:** A text input field with the value: 12.
- sdf2:** A text input field with the value: gdfdf.

Figure 5.26: Add document record

Users edited document depending on what fields they have permission to. A document needed to be locked to a particular user before that user was able to edit it. A document was locked by simply clicking the lock checkbox.



FDA
Document Tracking System

HOME
MY PROFILE
DOCUMENTS
DOCUMENT TYPES
DEPARTMENTS
USERS
COMPANIES
Signed in as jose
LOGOUT

ALL ACTIVE ARCHIVED Documents

Lock	Barcode	Title	Type	Location	Time Elapsed	Progress	Companies
<input checked="" type="checkbox"/>	101	ApproveDocument	Type 1	Station 9	00:00:22:06	50%	[Company r]
<input type="checkbox"/>	99	SampleDocument	Type 1	Station 9	00:00:22:07	50%	[Company r]
<input type="checkbox"/>	100	DocumentToBeApprov	Type 3	Station 22	00:00:22:07	0%	[Company r]
<input type="checkbox"/>	95	testtest	Type 3	Station 22	00:00:24:20	0%	[Company 2]
<input type="checkbox"/>	94	y	Type 3	Station 14	00:00:26:04	33%	[Company 2]
<input type="checkbox"/>	91	josejose	Type 3	Station 13	00:00:26:05	67%	[Company 3]
<input type="checkbox"/>	90	joselouie	Type 3	Station 13	00:00:26:06	67%	[Company 2] [Company 5]
<input checked="" type="checkbox"/>	88	Mark	Type 3	Station 22	00:00:26:09	0%	[Company 2]
<input type="checkbox"/>	86	Ironic	Type 3	Station 22	00:00:26:23	0%	[Company 2]
<input type="checkbox"/>	85	Chance	Type 1	Station 1sdf	00:00:27:02	0%	[Company 3] [Company 5]
<input type="checkbox"/>	84	tst1	Type 1	Station 1sdf	00:00:29:04	0%	[Company r] [Company 4]
<input checked="" type="checkbox"/>	81	testing2	Type 3	Station 22	00:00:29:06	0%	[Company 4]
<input type="checkbox"/>	82	testing3	Type 3	Station 22	00:00:29:06	0%	[Company 3]
<input type="checkbox"/>	83	testing4	Type 3	Station 22	00:00:29:06	0%	[Company 2]
<input checked="" type="checkbox"/>	79	test14	Type 3	Station 22	00:01:00:03	0%	[Company r] [Company 2] [Company 3]
<input type="checkbox"/>	78	test13	Type 3	Station 22	00:01:00:04	0%	[Company 2] [Company 4]
<input type="checkbox"/>	5	five	Type 4	Station 22	01:10:15:01	0%	[Company 40]

Doc 101 is locked to you.

Page 1 of 2 View 1 - 20 of 22

Figure 5.27: A document record locked

If users already had permission to edit a field in the document, they were required to populate them. Blank fields were not accepted and the document cannot be sent to the next department.

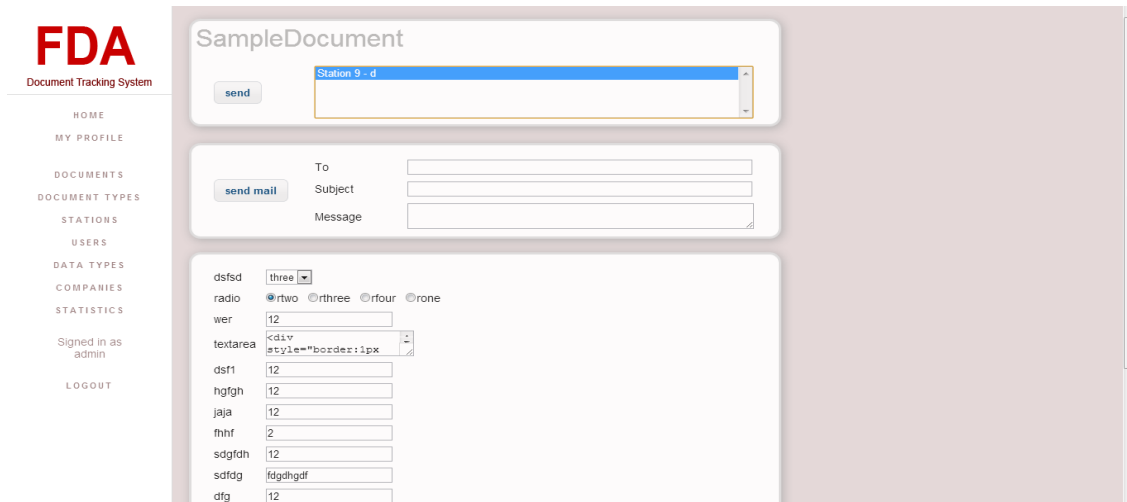


Figure 5.28: Edit document record

Non-applicant users were able to delete documents. When a document was deleted it was also archived for future reference.

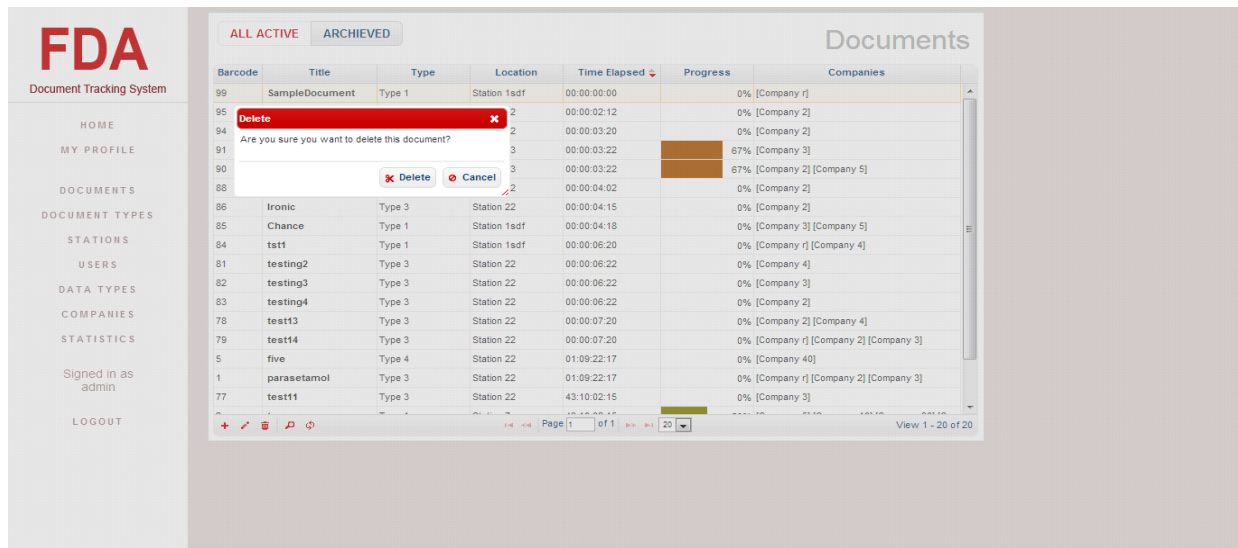


Figure 5.29: Delete document type

5.1.7 View/add/edit/delete department to workflow path

Only administrators were able to ad/edit/delete departments to a document workflow path.

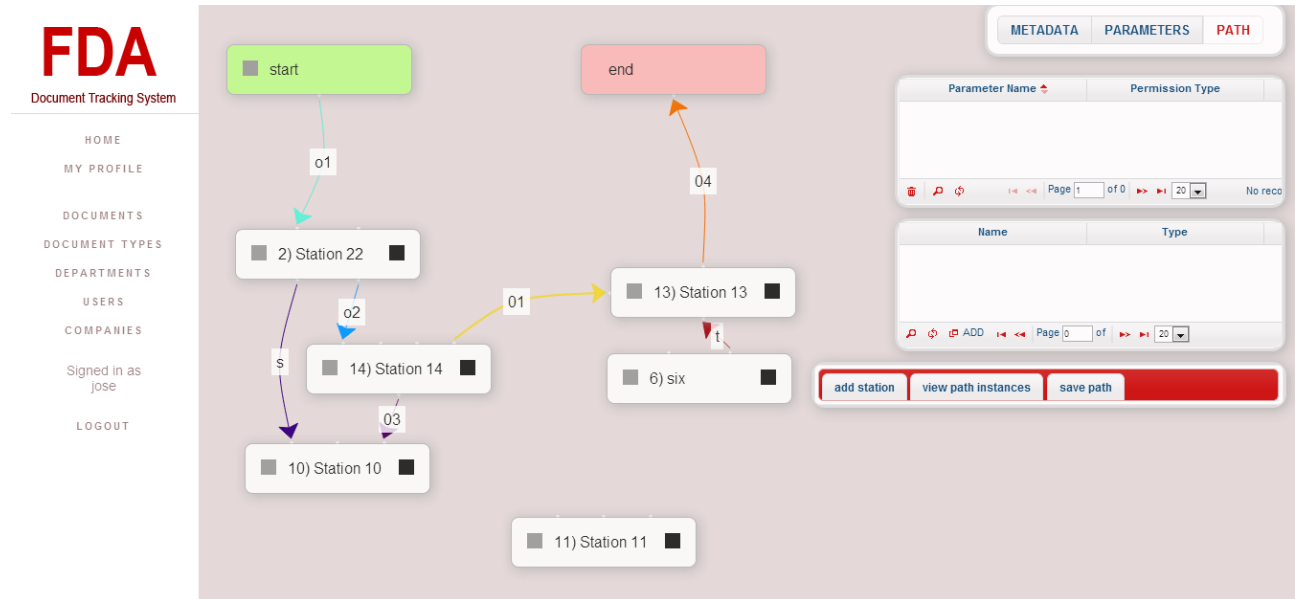


Figure 5.30: View/add/edit/delete departments to workflow

5.1.8 View workflow traffic

Workflow traffic was viewed on the document type.

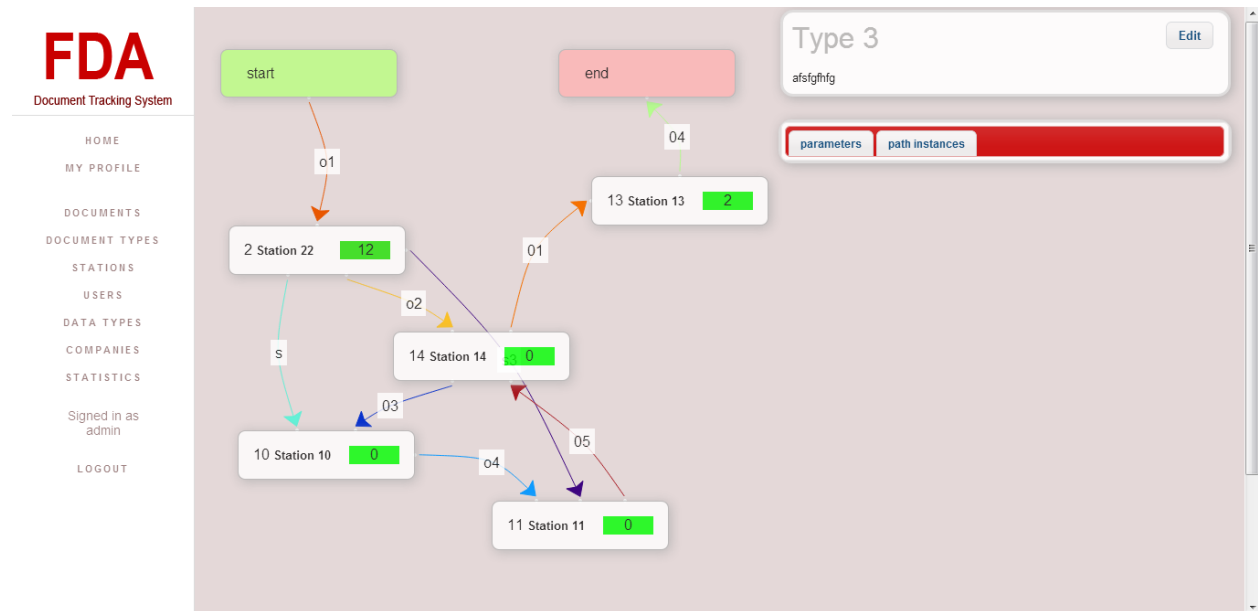


Figure 5.31: View workflow traffic

5.1.9 View a specific document's workflow path traffic

A specific document's workflow traffic was viewed by users with permission to view the document. Aside from the document counts indicated, the color of the box in a department node indicated its document congestion. The greener it was, the lesser congestion. The closer it was to the shade of red, the more congested it was.

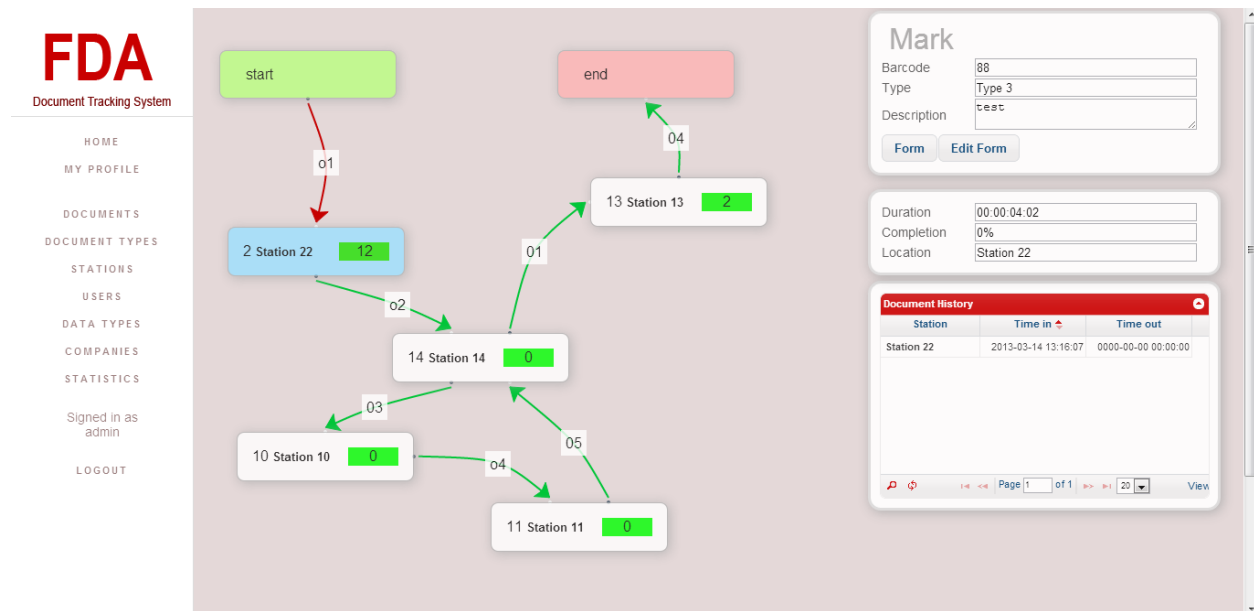


Figure 5.32: View a specific document's workflow traffic

Non-applicant users can view all stations' document congestion.

FDA
Document Tracking System

HOME
MY PROFILE
DOCUMENTS
DOCUMENT TYPES
STATIONS
USERS
DATA TYPES
COMPANIES
STATISTICS

Signed in as admin
LOGOUT

Stations

Name	Doc Num	Description
Station 22	12	afsfgthfga
Station 1sdf	9	afsfgthfg
Station 7	2	afsfgthfg
Station 13	4	afsfgthfg
six	1	afsfgthfg
Station 3	6	afsfgthfg
Station 18	8	afsfgthfg
Station 17	0	afsfgthfg
Station 16	0	afsfgthfg
Station 15	0	afsfgthfgwer
Station 14	0	afsfgthfg
Station 12	0	afsfgthfg
Station 11	0	afsfgthfg

+ Edit Permission Page 1 of 1 20 View 1 - 19 of 19

Figure 5.33: View all stations' document loads

5.2 Director's Functionalities

5.2.1 View Documents

When directors viewed documents, they were directed to pending documents in their station by default. They viewed all active documents as well as archived ones.

The screenshot displays the FDA Document Tracking System interface. On the left is a navigation menu with options: HOME, MY PROFILE, DOCUMENTS, STATIONS, USERS, COMPANIES, STATISTICS, Signed in as director, and LOGOUT. The main content area is titled 'Documents' and features three tabs: ALL ACTIVE, STATION, and ARCHIEVED. Below the tabs is a table with the following data:

Barcode	Title	Type	Time Elapsed	Progress	Companies
101	ApproveDocument	Type 1	00:00:00:00	0%	[Company r]
85	Chance	Type 1	00:00:04:19	0%	[Company 3] [Company 5]
84	tst1	Type 1	00:00:06:21	0%	[Company r] [Company 4]

At the bottom of the table, there is a pagination control showing 'Page 1 of 1' and '20' items per page. The status 'View 1 - 3 of 3' is also visible.

Figure 5.34: Pending documents in their station.

5.2.2 Approve/Disapprove document

Department heads and applicants were able to send a document to other departments depending if it is approved or disapproved.

The screenshot shows the 'ApproveDocument' interface. On the left is a navigation menu for the 'FDA Document Tracking System' with options like HOME, MY PROFILE, DOCUMENTS, STATIONS, USERS, COMPANIES, STATISTICS, Signed in as director, and LOGOUT. The main form area contains a 'send' button, a dropdown menu with 'Station 9 - d' selected, a 'send mail' button, and three input fields for 'To', 'Subject', and 'Message'. Below these are three input fields with values '12', '2', and '12', and a 'save' button.

Figure 5.35: Approve/Disapprove a document

The screenshot shows the 'Documents' table in the FDA Document Tracking System. The table has columns for Barcode, Title, Type, Time Elapsed, Progress, and Companies. There are two rows of data. The first row has Barcode 85, Title 'Chance', Type 1, Time Elapsed 00:00:04:19, Progress 0%, and Companies [Company 3] [Company 5]. The second row has Barcode 84, Title 'tstt', Type 1, Time Elapsed 00:00:06:21, Progress 0%, and Companies [Company r] [Company 4].

Barcode	Title	Type	Time Elapsed	Progress	Companies
85	Chance	Type 1	00:00:04:19	0%	[Company 3] [Company 5]
84	tstt	Type 1	00:00:06:21	0%	[Company r] [Company 4]

Figure 5.36: Pending documents after the approval/disapproval.

5.2.3 Email Notification

A department head and applicant were able send emails to the applicants if something was wrong and needs the applicant's attention.

The screenshot displays the FDA Document Tracking System interface. On the left is a navigation menu with the following items: HOME, MY PROFILE, DOCUMENTS, STATIONS, USERS, COMPANIES, STATISTICS, Signed in as director, and LOGOUT. The main content area is titled "Chance" and contains three form sections. The first section has a "send" button and a dropdown menu currently showing "Station 9 - d". The second section, titled "send mail", includes fields for "To" (jose_louie_mark@yahoo.com, sdf@sdf.com), "Subject" (Document Notification), and "Message" (There was a problem in your document. Please follow-up your application as soon as possible.). The third section contains three input fields labeled "hgfgh", "fhhf", and "sdgfdh", each with a "0" value, and a "save" button.

Figure 5.37: Notify applicant via e-mail.

5.3 Director Assistant's Functionalities

5.3.1 View pending documents

Department assistants only viewed documents on their own station.

The screenshot displays the FDA Document Tracking System interface. On the left is a navigation menu with options: HOME, MY PROFILE, DOCUMENTS, STATIONS, COMPANIES, STATISTICS, Signed in as directorassistant, and LOGOUT. The main content area is titled 'Documents' and features a table with columns: Barcode, Title, Type, Time Elapsed, Progress, and Companies. The table lists 13 documents, all with 0% progress. The 'Time Elapsed' column includes a dropdown arrow. At the bottom of the table, there are navigation icons and a page indicator showing 'Page 1 of 1' and '20' items per page. The status 'View 1 - 13 of 13' is visible in the bottom right corner.

Barcode	Title	Type	Time Elapsed	Progress	Companies
100	DocumentToBeAppro	Type 3	00:00:00:00	0%	[Company 1]
95	testtest	Type 3	00:00:02:13	0%	[Company 2]
94	y	Type 3	00:00:03:21	0%	[Company 2]
88	Mark	Type 3	00:00:04:03	0%	[Company 2]
86	Ironic	Type 3	00:00:04:16	0%	[Company 2]
83	testing4	Type 3	00:00:06:23	0%	[Company 2]
81	testing2	Type 3	00:00:06:23	0%	[Company 4]
82	testing3	Type 3	00:00:06:23	0%	[Company 3]
78	test13	Type 3	00:00:07:21	0%	[Company 2] [Company 4]
79	test14	Type 3	00:00:07:21	0%	[Company 1] [Company 2] [Company 3]
5	five	Type 4	01:09:22:18	0%	[Company 40]
1	parasetamol	Type 3	01:09:22:18	0%	[Company 1] [Company 2] [Company 3]
77	test11	Type 3	43:10:02:16	0%	[Company 3]

Figure 5.38: View document records in their station

5.3.2 Approve/Disapprove documents

Department assistants also had the ability to send documents to other departments.

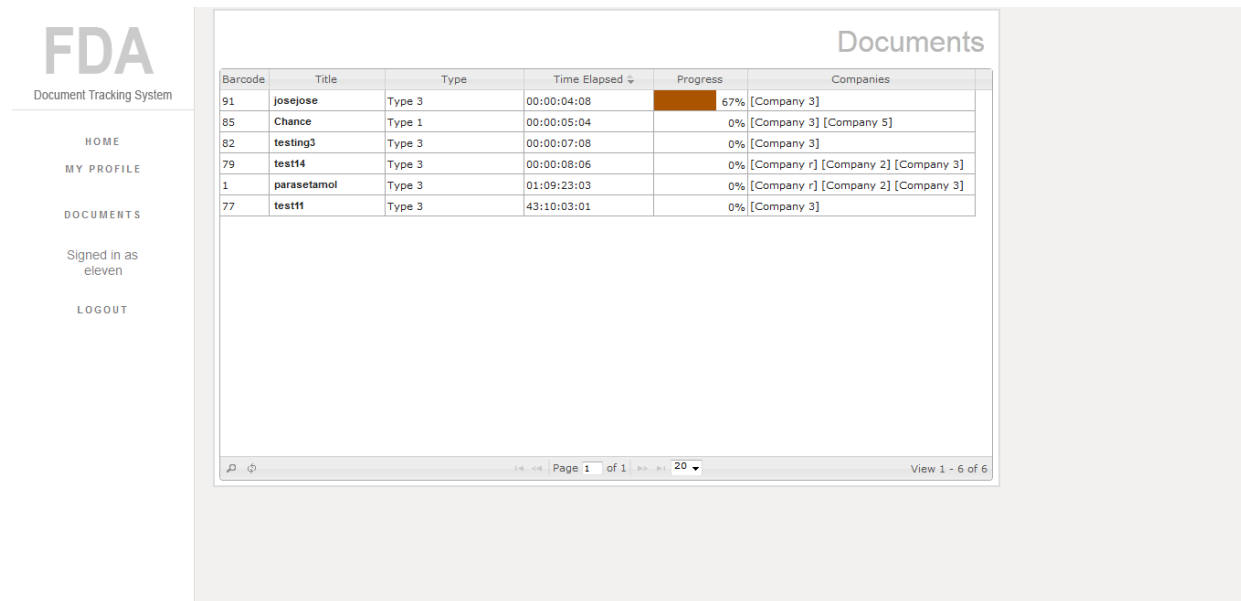
The screenshot shows the 'DocumentToBeApproved' interface. On the left is a navigation menu with 'HOME', 'MY PROFILE', 'DOCUMENTS', 'STATIONS', 'COMPANIES', 'STATISTICS', 'Signed in as directorassistant', and 'LOGOUT'. The main content area has a title 'DocumentToBeApproved' and a 'send' button. Below this is a list of stations: 'Station 10 - s', 'Station 11 - s3', and 'Station 14 - o2'. A 'send mail' button is also present. The form includes fields for 'To', 'Subject', and 'Message'. Below these are options for 'married' (checkbox), 'color' (radio buttons for red, blue, green), and 'selecttest' (dropdown menu with 'two' selected). There are also 'LastName' and 'Firstname' text input fields and a 'save' button.

Figure 5.39: Approve/Disapprove a pending document.

5.4 Applicant's Functionalities

5.4.1 Document Monitoring

Applicants only viewed documents that were assigned to the company that they were in.



The screenshot displays the 'Documents' section of the FDA Document Tracking System. The interface includes a sidebar with navigation options: HOME, MY PROFILE, DOCUMENTS, Signed in as eleven, and LOGOUT. The main content area shows a table of document records with the following data:

Barcode	Title	Type	Time Elapsed	Progress	Companies
91	josejose	Type 3	00:00:04:08	67%	[Company 3]
85	Chance	Type 1	00:00:05:04	0%	[Company 3] [Company 5]
82	testing3	Type 3	00:00:07:08	0%	[Company 3]
79	test4	Type 3	00:00:08:06	0%	[Company r] [Company 2] [Company 3]
1	parasetamol	Type 3	01:09:23:03	0%	[Company r] [Company 2] [Company 3]
77	test11	Type 3	43:10:03:01	0%	[Company 3]

At the bottom of the table, there is a pagination control showing 'Page 1 of 1' and 'View 1 - 6 of 6'.

Figure 5.40: View document records in the company that they were assigned in

VI. Discussion

WADoTS for FDA is a workflow aware information system that enables movement tracking of documents through a workflow in FDA. Users are able to see graphical representations of documents' statuses. It facilitates FDA's dynamic change in document processing requirements and workflows. Workflows and requirements are customized by the user's preference and needs. It provides user-friendly interface in defining and setting detailed routing and processing requirements of a typical workflow. Applicants of documents are updated with their document's status.

Being workflow aware, the system gives administrators the permission to modify the structure of each document's workflow. They have the access to define the name, source and destination of a specific route that the document takes. The system makes sure that routes are well defined, so that every nodes, stations or departments can be accessed from the start stage and can access the termination stage of the workflow. This assures a robust workflow with no documents trapped in a certain department. All documents are created and archived properly.

A document is made by copying a document type. In a document's creation, a document type workflow and permissions are cloned. By replicating document types to create document, the workflow would be more stable. Document workflow, parameters and permissions cannot be edited and changed. Permission parameter values are the only editable attributes of a document. When a document is created, and the administrator decides to change the document type in which the document was patterned, there are no conflicts that would happen, since workflow path and requirements are already copied to the document and are not editable.

Administrators set the document's requirement parameters to be approved by users of each department. These requirement parameters are filled up by the user that handles the document in a

particular department. Parameters require a value before a document is sent to another station. In this way, mistakes on requirement parameter values are prevented.

Parameters of requirements for each document are read and edited when permissions are indicated. These permissions define which users can access certain parameters. Permissions are in the stations level, which means that all types of users in a specified station share the same permissions. Permissions are given to the station where the user is assigned. There are three kinds of permissions in the system: read-only, read-write and create. Read-only is used as reference, which might be considered, in the input value of parameter with read-write permission. The create permission allows a certain document to be created by users but this does not mean that a document will directly go to their document list. A document will always go to the first department specified by its workflow.

A document is sent from one department to another by strictly following the workflow set up. A document cannot deviate from this and can only be removed from the path if it is deleted. Every transfer of document is recorded. The record is used in determining where the document is and its percent completion. Percent completion is calculated by getting the minimum total number of steps from the start date to the current place over the minimum total number of paths from start date to end date. When a document reaches the point where it will be archived, the document's requirement parameters and workflow are first copied to the archive and then deleted. When the document is already archived, it sends an email to the applicants that the document is completed.

Applicants are assigned to companies and companies are assigned to documents. Applicants can monitor all the documents in a company that they are assigned to. Applicants are able to track where a document is currently located, what path it already took and the path it can possibly take. Congestion of departments or stations is shown so that the applicant can get a rough estimate of the document's completion. When a document has a problem and needs the applicant's attention, the station user can send an email to the applicant.

VII. Conclusion

The study developed a workflow aware information system, WADoTS, which provides movement and requirement status tracking of documents through a business process within FDA. It facilitates FDA's dynamic document processing. Workflows and requirements are customized by the user's preference and needs. It offers user friendly interface in defining and setting detailed routing and processing requirements of a typical document workflow. Users are able to track graphically where the located document is, what paths it already took, the paths it can potentially take and the congestion of the departments along their paths. Documents are passed on from one department to another by users that have the right permissions to do so. Applicants are notified if their documents are finished or if there is a problem.

VIII. Recommendation

Home page is blank and it needs to be populated with news, feeds and more. Users also need to know what is happening within the system.

To have permission, a user has to communicate with the administrator first. I would be better if the system itself would have a request permission functionality so that users would not need to resort to other means of communication that would possibly waste effort.

Currently, the documents are physically passed in the processing Users input data to the system so that other users can monitor the document they have. Very soon, FDA would resort to electronic documents. File upload feature would be needed to replace physical documents.

Some workflows in FDA goes through stages where payment is required. If so, it would be convenient for the applicant to pay it online. This would save resources for both applicant and FDA.

For clients that are on the go and always busy, having a mobile monitoring of their documents is needed. Today, many people have smartphones, so to facilitate the need of time, clients can just monitor their documents in their mobile devises.

IX. Bibliography

- [1] The Aquino Health Agenda: Universal Health Care for All Filipinos. (2010). Retrieved March 20, 2012, from <http://www.doh.gov.ph/node/583>.
- [2] FDA History. (n.d.). Food and Drug Association. Retrieved March 20, 2012, from <http://www.fda.gov.ph/About%20FDA.html>.
- [3] FDA Services. (2012). Food and Drug Administration. Retrieved March 20, 2012 from http://www.fda.gov.ph/checklist_of_requirements.html
- [4] Aguing, M. *Personal Interview*. March 2012.
- [5] Pauline, S. *Personal Interview*. February 2012 - March 2012.
- [6] Lazo, S. *Personal Interview*. February 2012 - March 2012.
- [7] Wolfgang, P. and Zeeshan, J. (2005). Document Tracking and Collaboration Support using RFID. *Fraunhofer Institute FIT D-53754 St. Augustin*.292-296
- [8] Sueno, B. D. (2009). UP Docutrak: An assessment of an in-house developed document tracking system (DTS). *Philippine Librarianship*, 29, 24-45.
- [9] Zhang, J., Liu, Y., & Song, J. (2009). The Scientific Research Management System of Colleges. Second International Symposium on Knowledge Acquisition and Modeling. *The 2nd International Symposium on Knowledge Acquisition and Modeling*. 289-292.
- [10] Magno, F. and Serafica, R. (2002). *Information Technology for Good Governance*. Manila, Yuchengco Center for East Asia
- [11] The Army Records Information Management System (ARIMS), Army Regulation 25-400-2 (2007)
- [12] Memon, Z. A., Majid, M. Z. A., & Mustaffar, M. (2006). A systematic approach for monitoring and evaluating the construction project progress. *The Institution of Engineers, Malaysia*. 67. 26-32.
- [13] Shan, W. (2009). Sales and Inventory Tracking System For ReadySpace SG Pte Ltd. ICT499 Capstone Project SIM University.
- [14] Sauer T., Minor M. and Werno S. (2008). An Agent-oriented System for Workflow Enactment Tracking. *IEEE, Computer Society*. 235-240
- [15] Muehlen, M. and Rosemann, M. (2003). Workflow-based Process Monitoring and Controlling- Technical and Organizational Issues. *33rd Hawaii International Conference on System Sciences*. 1-10
- [16] Vossen, G. and Weske, M. (2010). The WASA Approach to Workflow Management for Scientific Applications. *Lehrstuhl fur Informatik, University of Muenster*. 1-22

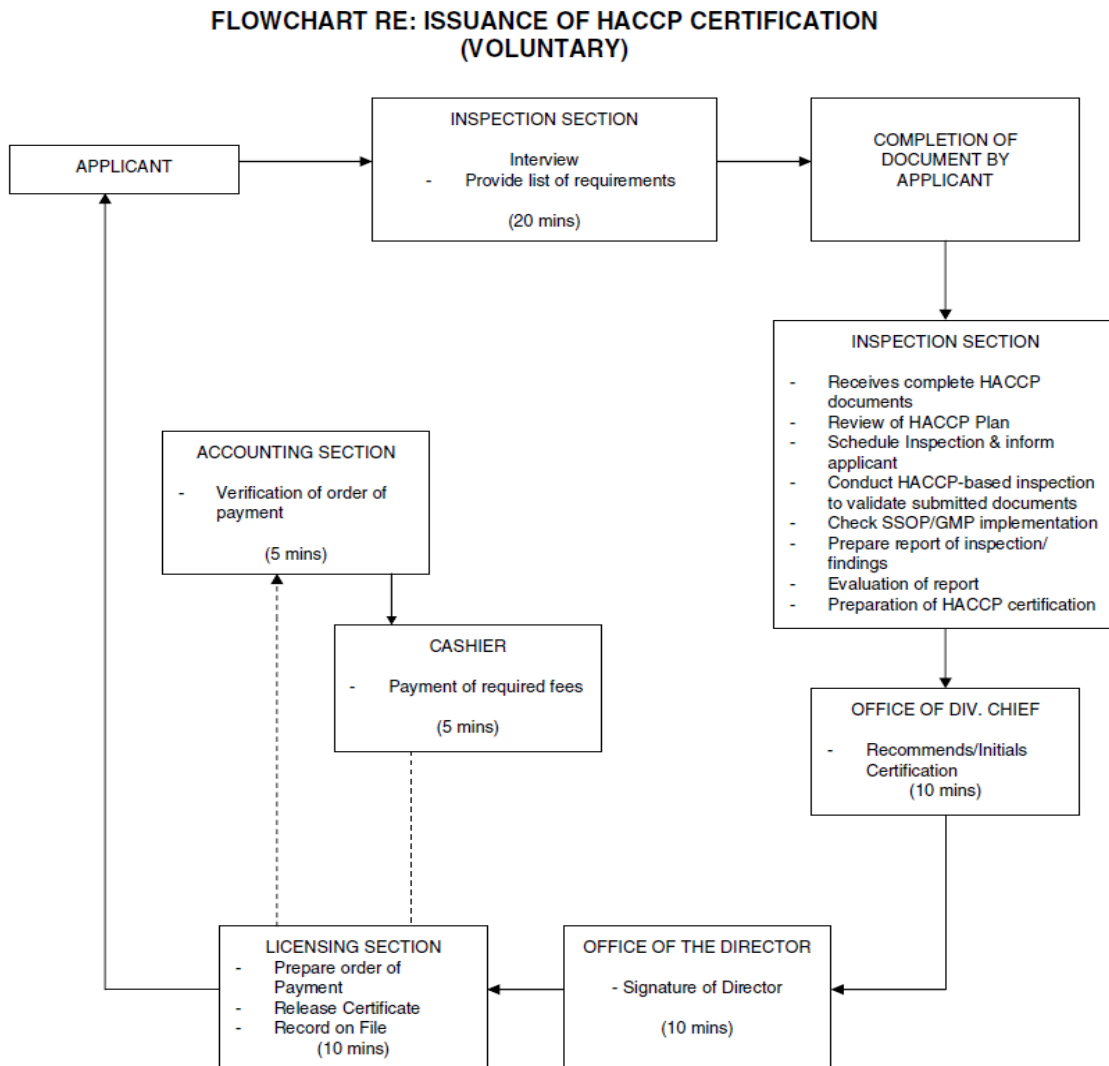
- [17] Hashim, F., Alam G. M., & Siraj, S. (2010). Information and communication technology for participatory based decision-making-E-management for administrative efficiency in Higher Education. *International Journal of Physical Sciences*. 5. 383-392.
- [18] Conway, D. (2002). *U.S. Patent No. 626,642*. Washington, DC: U.S. Patent and Trademark Office.
- [19] Food and Drug Administration. (n.d.). Department of Health. Retrieved March 20, 2012 from <http://www.doh.gov.ph/bfad/orgchart>
- [20] Monitoring. (2011). Business Dictionary. Retrieved March 18,2012 from <http://www.businessdictionary.com/definition/monitoring.html>.
- [21] Bartle, P. (n.d.). The nature of monitoring and evaluation. Retrieved March 18,2012 from <http://www.scn.org/cmp/modules/mon-wht.htm>
- [22] Tracking. (2011). The Free Dictionary. Retrieved March 18, 2012 from <http://www.thefreedictionary.com/track,tracking>
- [23] Track. (2011). Your Dictionary.com. Retrieved March 18, 2012 from <http://www.yourdictionary.com/track>
- [24] An Introduction to Workflow Management Systems. (2011). Center for Technology in Government University at Albany / SUNY. Retrieved March 18, 2012 from <http://bpt.hpi.unipotsdam.de/pub/Public/PaperArchive/habil.pdf>
- [25] Business process management (BPM). (2005). Retrieved March 22, 2012 <http://searchcio.techtarget.com/definition/business-process-management>
- [26] Ko, R., Lee, S. , Lee, E., (2009) Business process management (BPM) standards: a survey 746-748
- [27] Management Information Systems. (2011). Encyclopedia.com. Retrieved March 18, 2012 from http://www.encyclopedia.com/topic/Management_information_systems.aspx
- [28] Database Management Systems. (2011). Retrieved March 18, 2012 from http://www.quackit.com/database/tutorial/database_management_systems.cfm
- [29] Database management system (DBMS). (2011). In Encyclopedia Britannica. Retrieved March 18, 2012 from http://www.britannica.com/EBchecked/topic/152201/database-management_system
- [30] DBMS (Database Management System). (n.d.). Retrieved March 18, 2012 from <http://www.allbusiness.com/glossaries/dbms-database-management-system/4955874-1.html>.
- [31] PHP. (n.d.). Retrieved March 18, 2012 from <http://www.webopedia.com/TERM/P/PHP.html>
- [32] PHP (Hypertext Preprocessor). (n.d.). Retrieved March 18, 2012 from <http://www.techterms.com/definition/php>
- [33] Codeigniter Development. (n.d.). Retrieved March 18, 2012 from <http://www.codeigniter.com/>

[34] Codeigniter Development. (n.d.). Retrieved March 18, 2012 from
<http://www.softwareimc.com/services/web-development/codeigniter-development>

X. Appendix

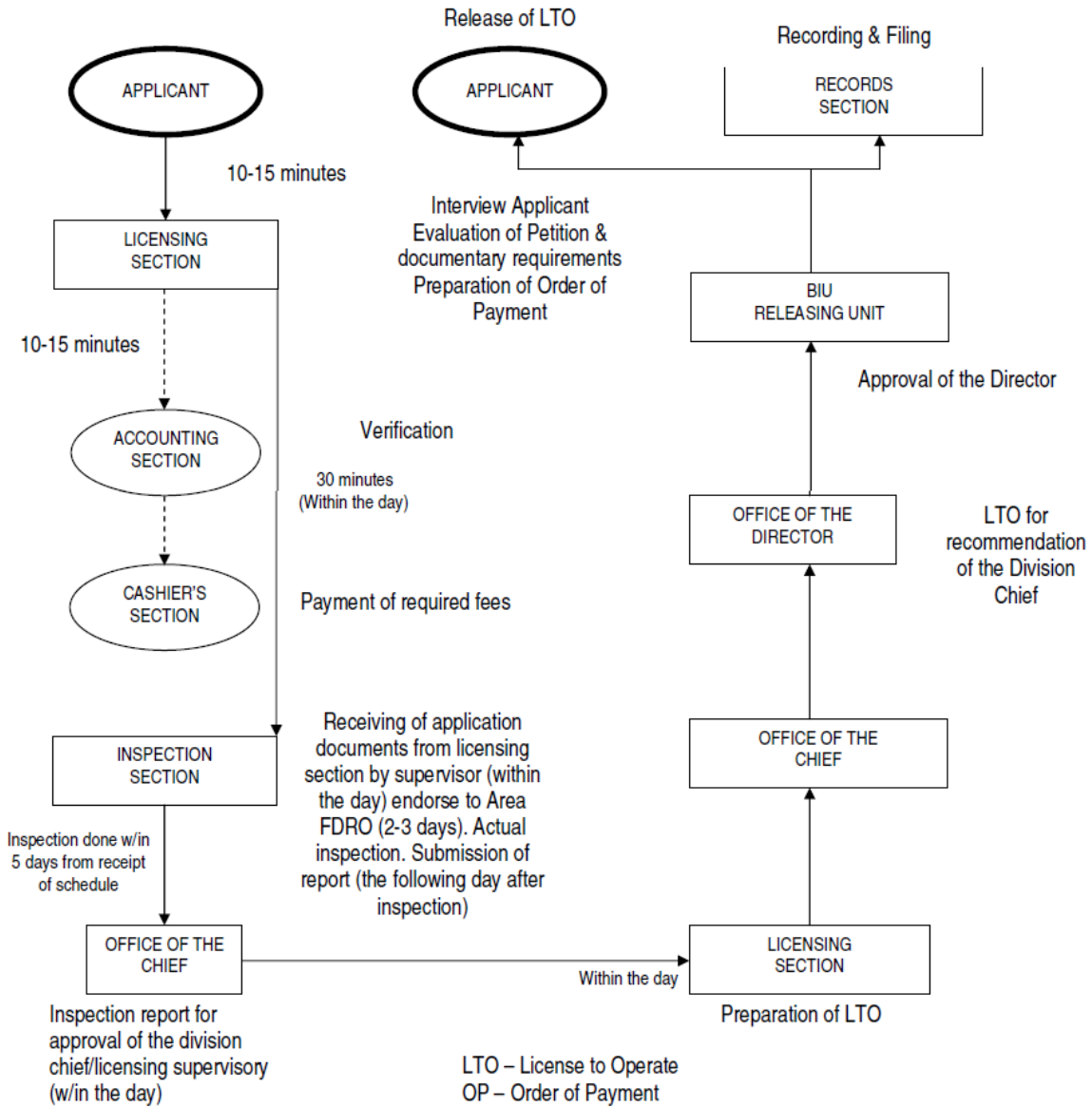
A. Flowcharts

i. Establishment



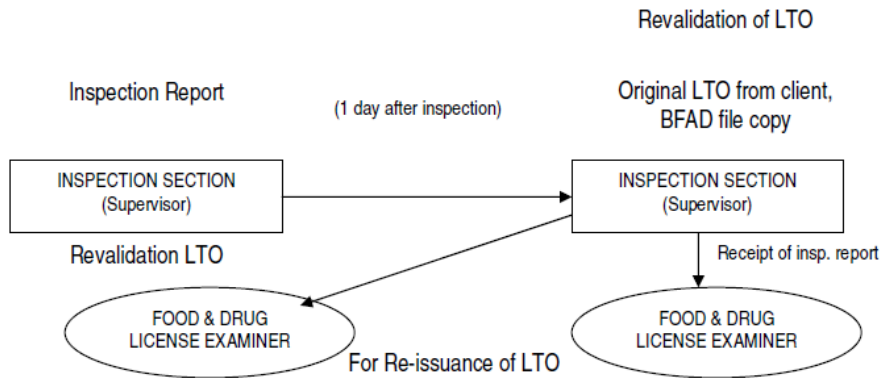
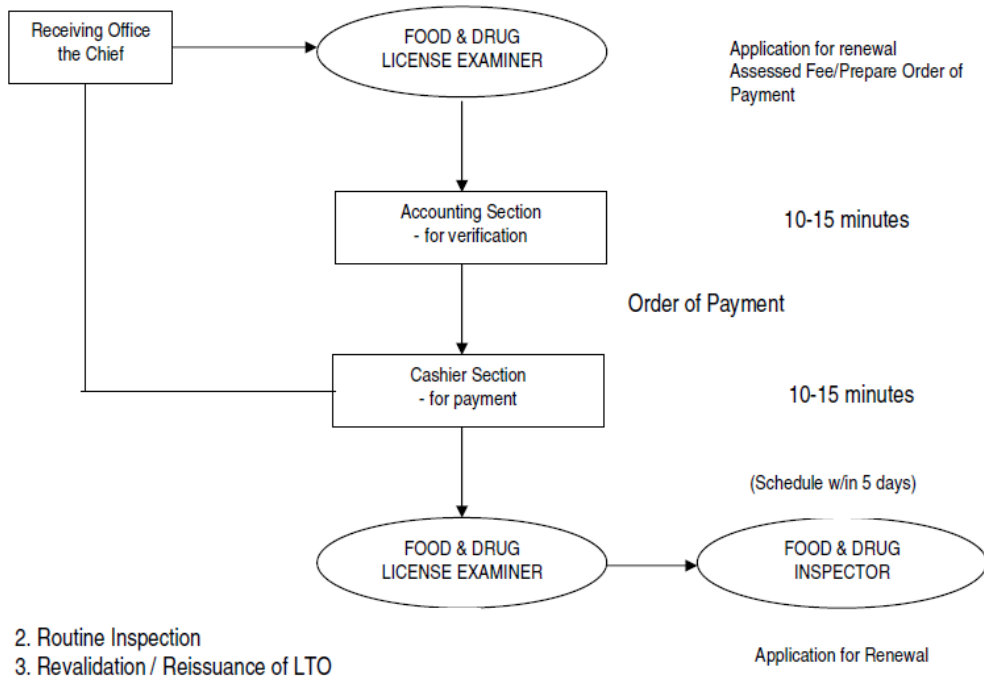
Appendix 1

**FLOWCHART FOR INITIAL LICENSING (OPENING) FOOD, DRUG,
COSMETIC MEDICAL DEVICE & HHS ESTABLISHMENTS AND DRUG
OUTLETS**



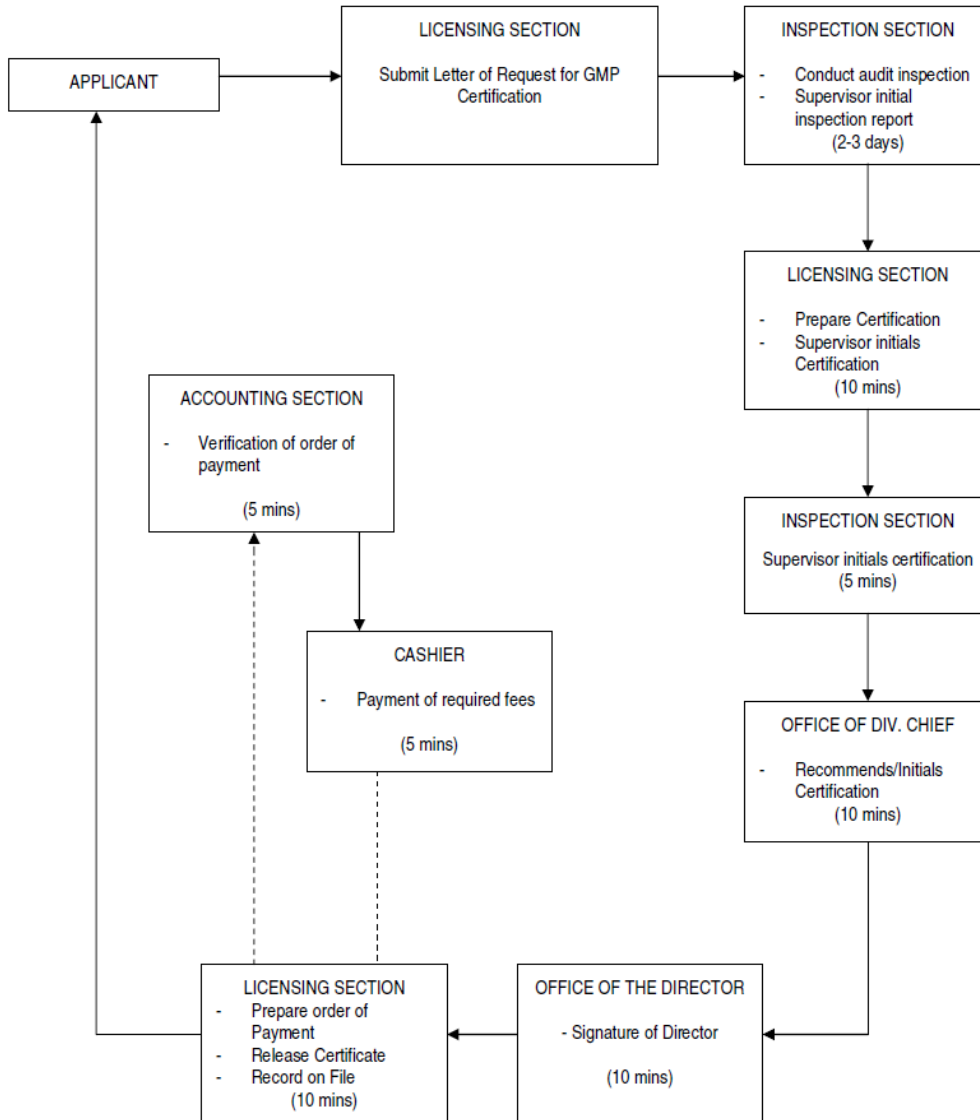
Appendix 3

**FLOWCHART FOR LICENSING (RENEWAL) FOOD, DRUG, COSMETIC
MEDICAL DEVICE & HHS ESTABLISHMENTS AND DRUG OUTLETS**



Appendix 4

**FLOWCHART RE: ISSUANCE OF GMP CERTIFICATION
(DRUG)**

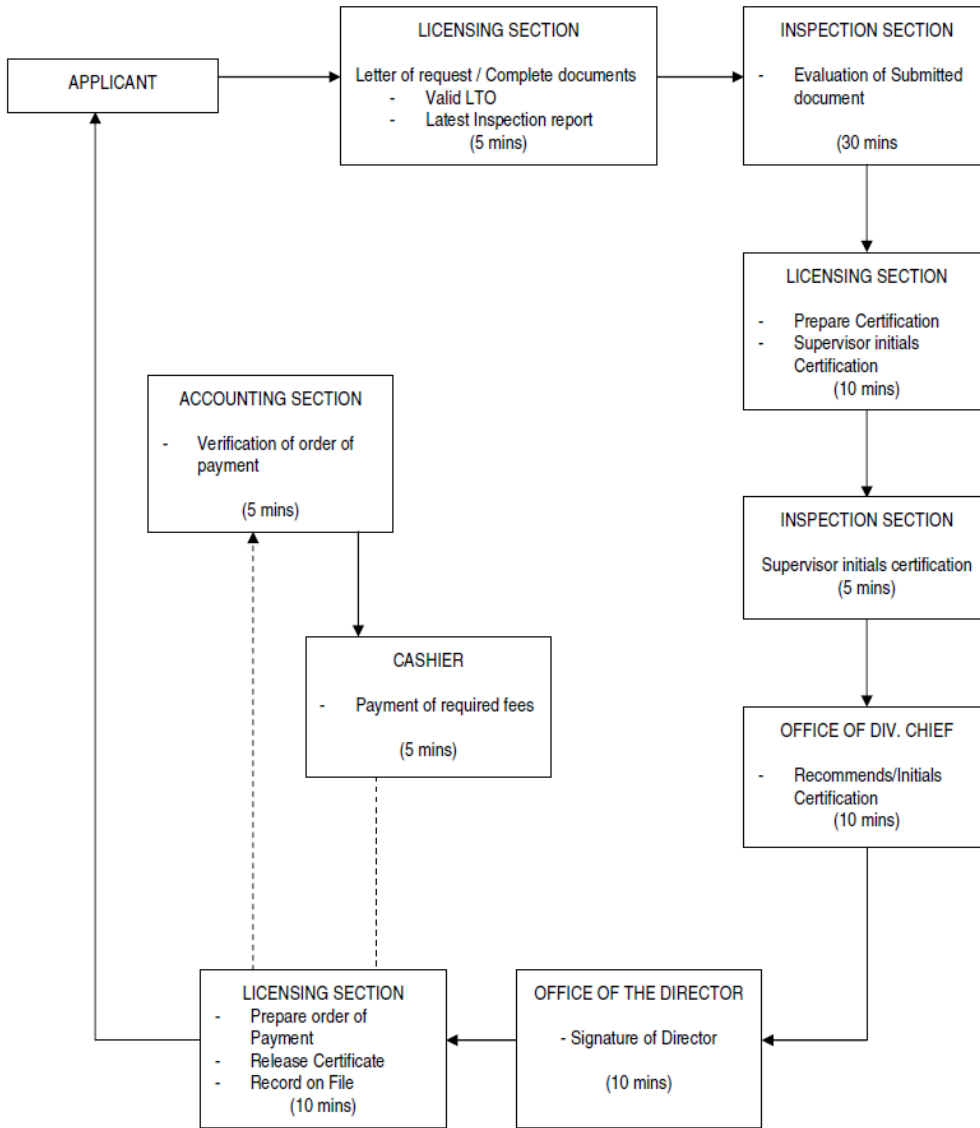


Note:

- For new establishment/transfer of plant site GMP Certificate will be issued only after one year of operation from issuance of License to Operate subject to audit inspection.

Appendix 5

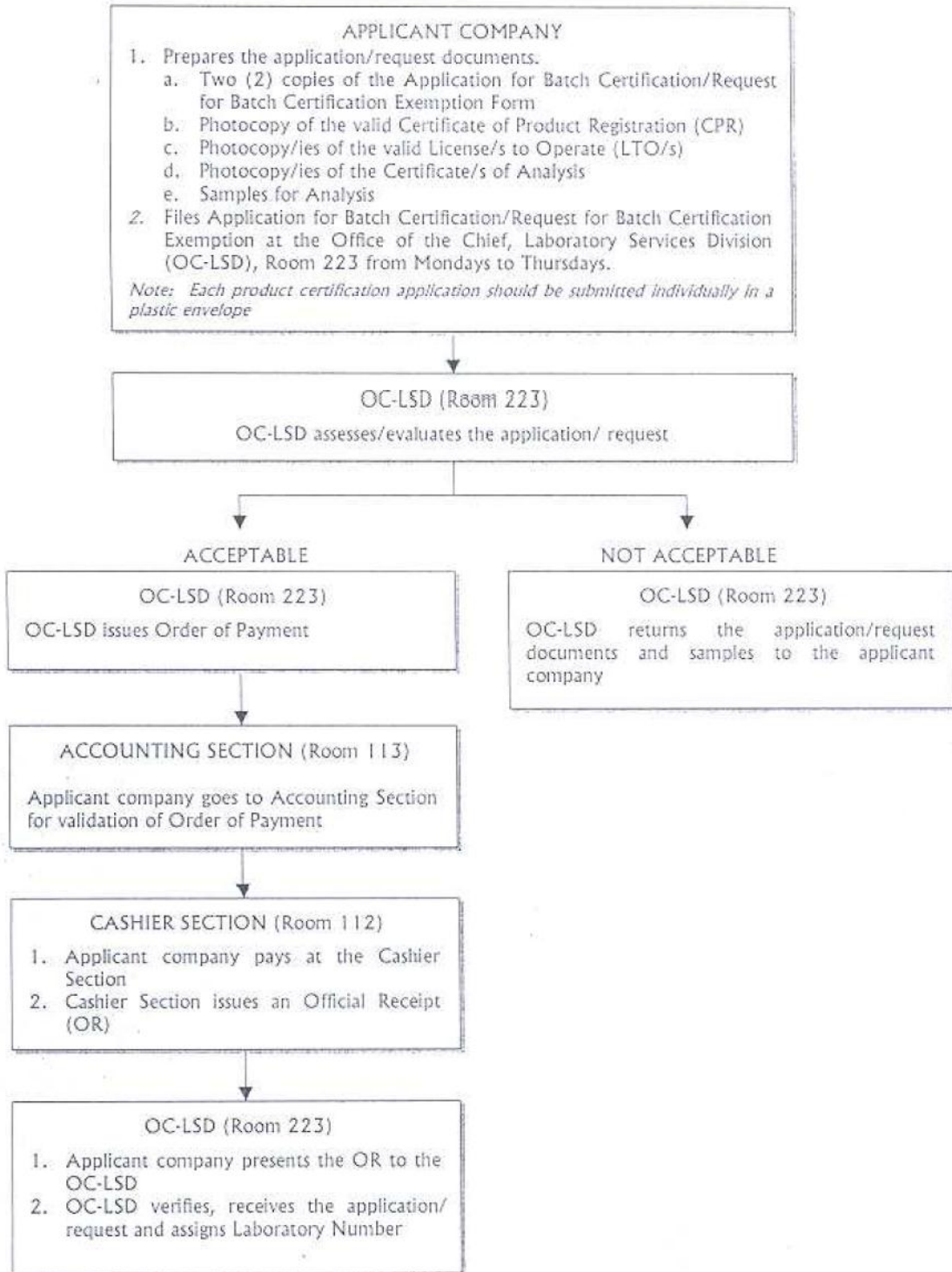
**FLOWCHART RE: ISSUANCE OF GMP CERTIFICATION
(FOOD)**



Appendix 6

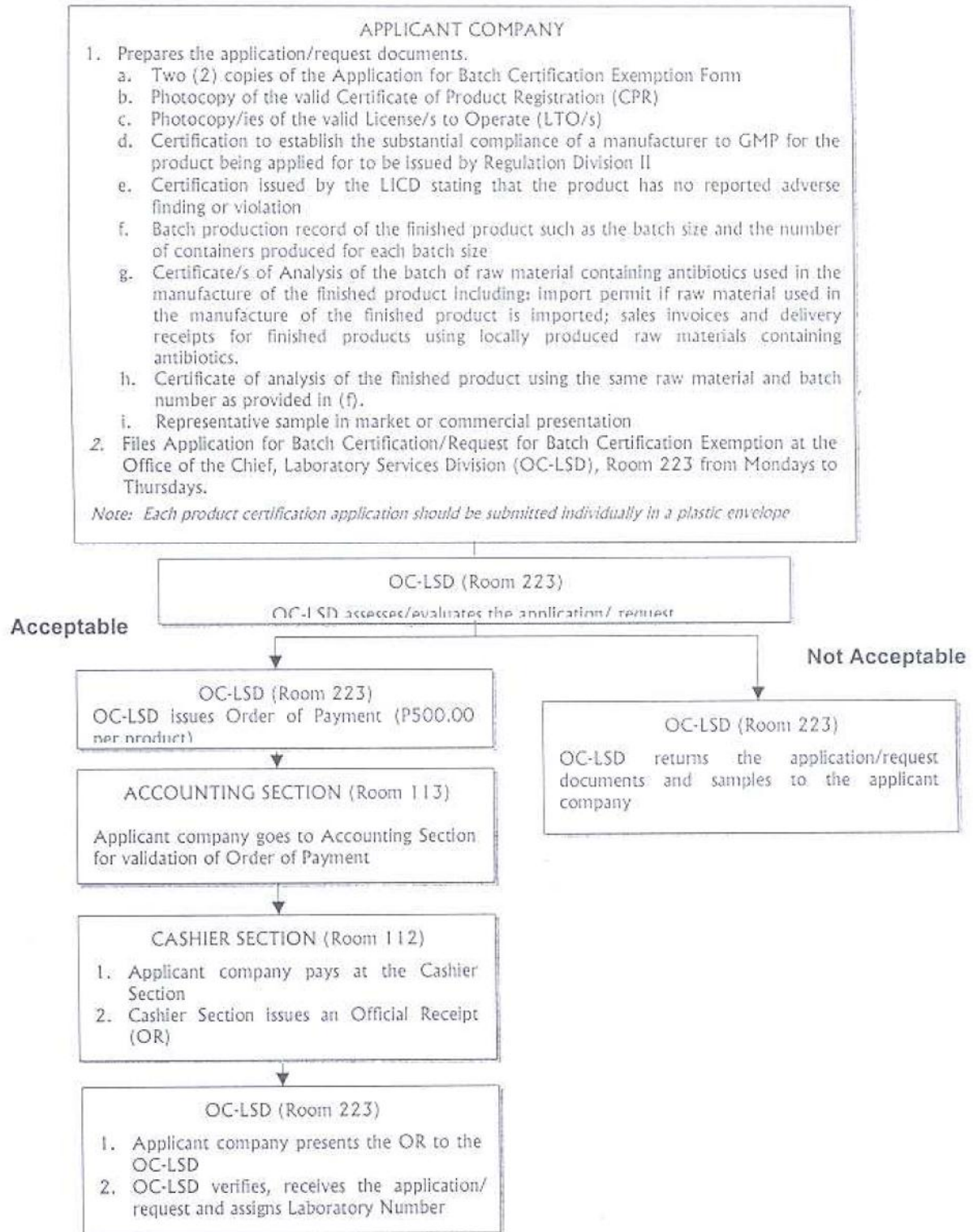
ii. Laboratory

PROCEDURE FOR FILING AN APPLICATION FOR BATCH CERTIFICATION/
A REQUEST FOR BATCH CERTIFICATION EXEMPTION (ANTIBIOTICS)



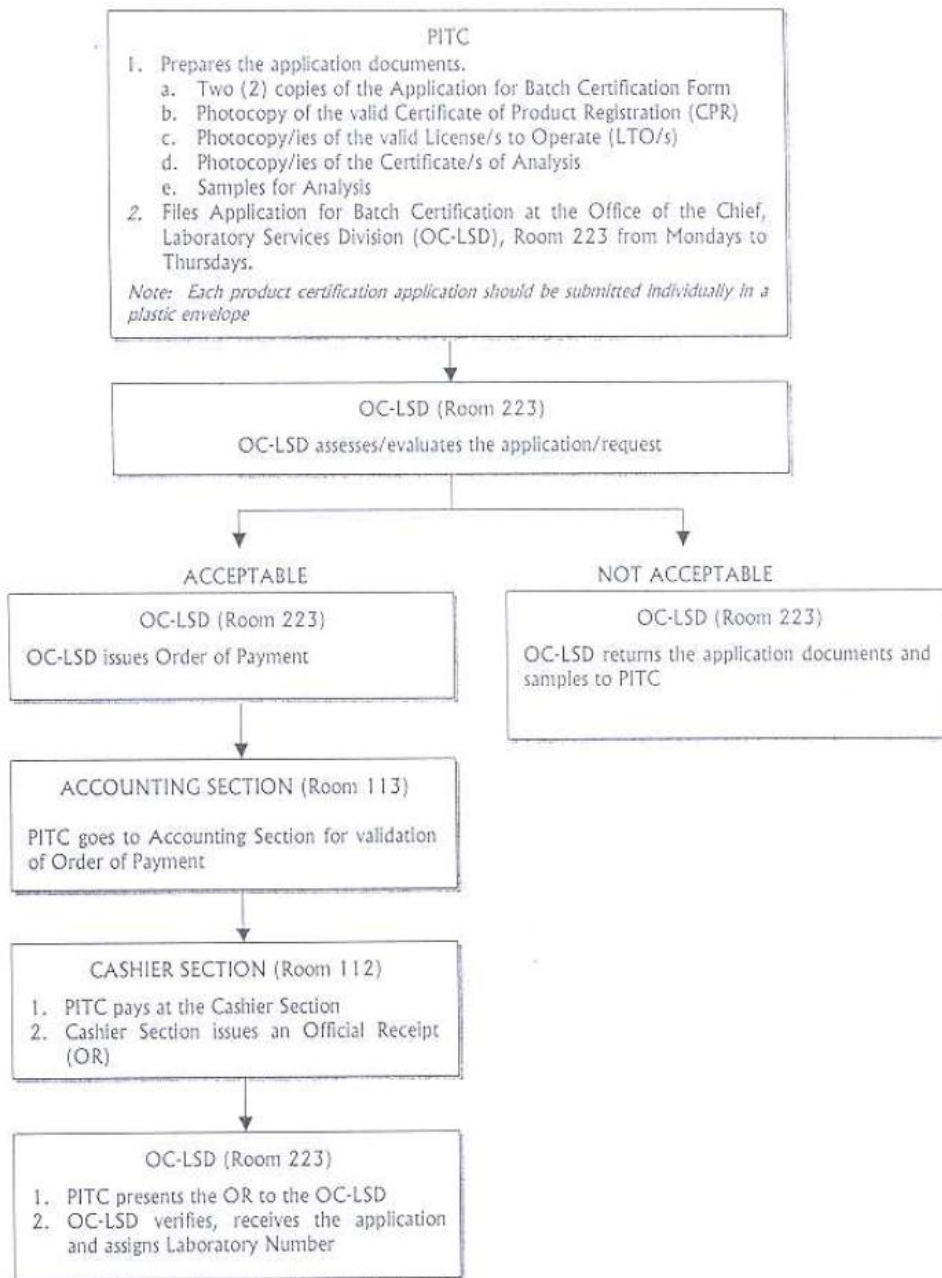
Appendix 7

PROCEDURE FOR FILING AN APPLICATION FOR EXEMPTION FROM BATCH CERTIFICATION OF LOCALLY MANUFACTURED PRODUCTS (ANTIBIOTICS)



Appendix 8

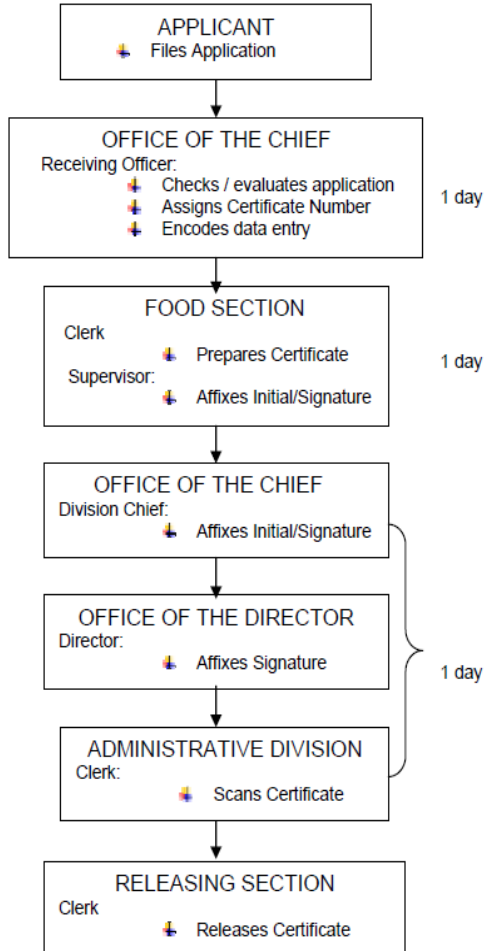
PROCEDURE FOR FILING AN APPLICATION FOR BATCH CERTIFICATION OF PHILIPPINE INTERNATIONAL TRADING CORPORATION (PITC) PRODUCTS



Appendix 9

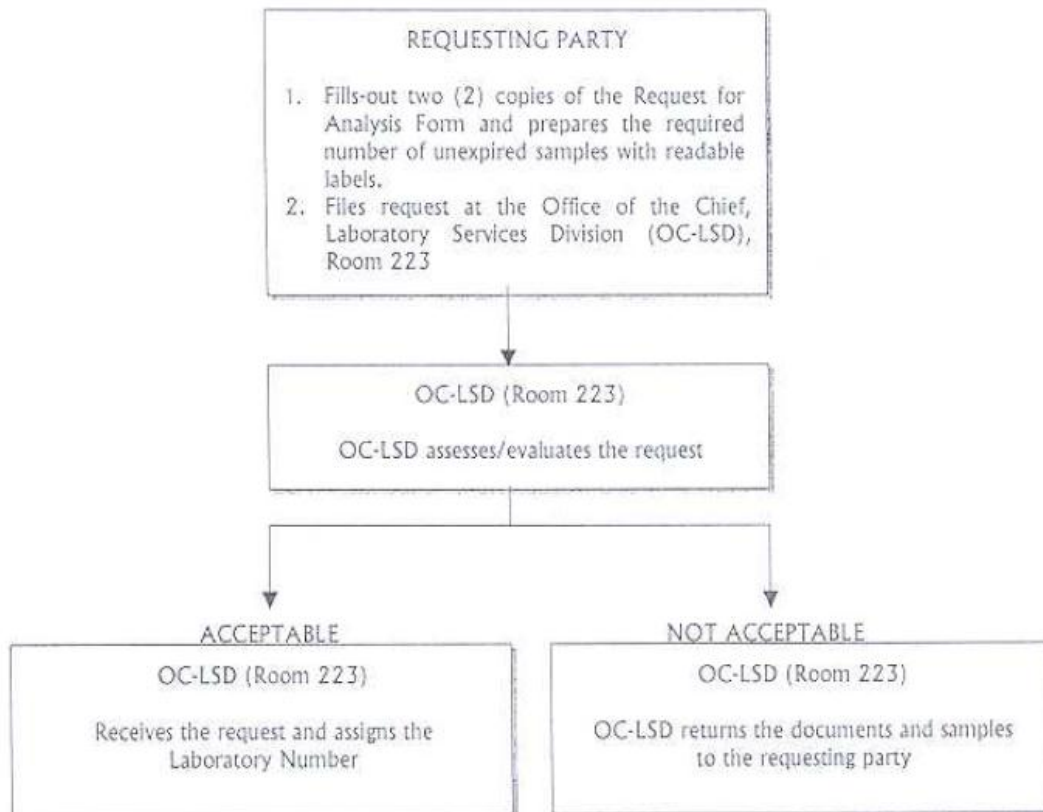
FLOWCHART FOR ISSUANCE OF FOOD EXPORT / COMMODITY CLEARANCE

LABORATORY SERVICES DIVISION



Appendix 10

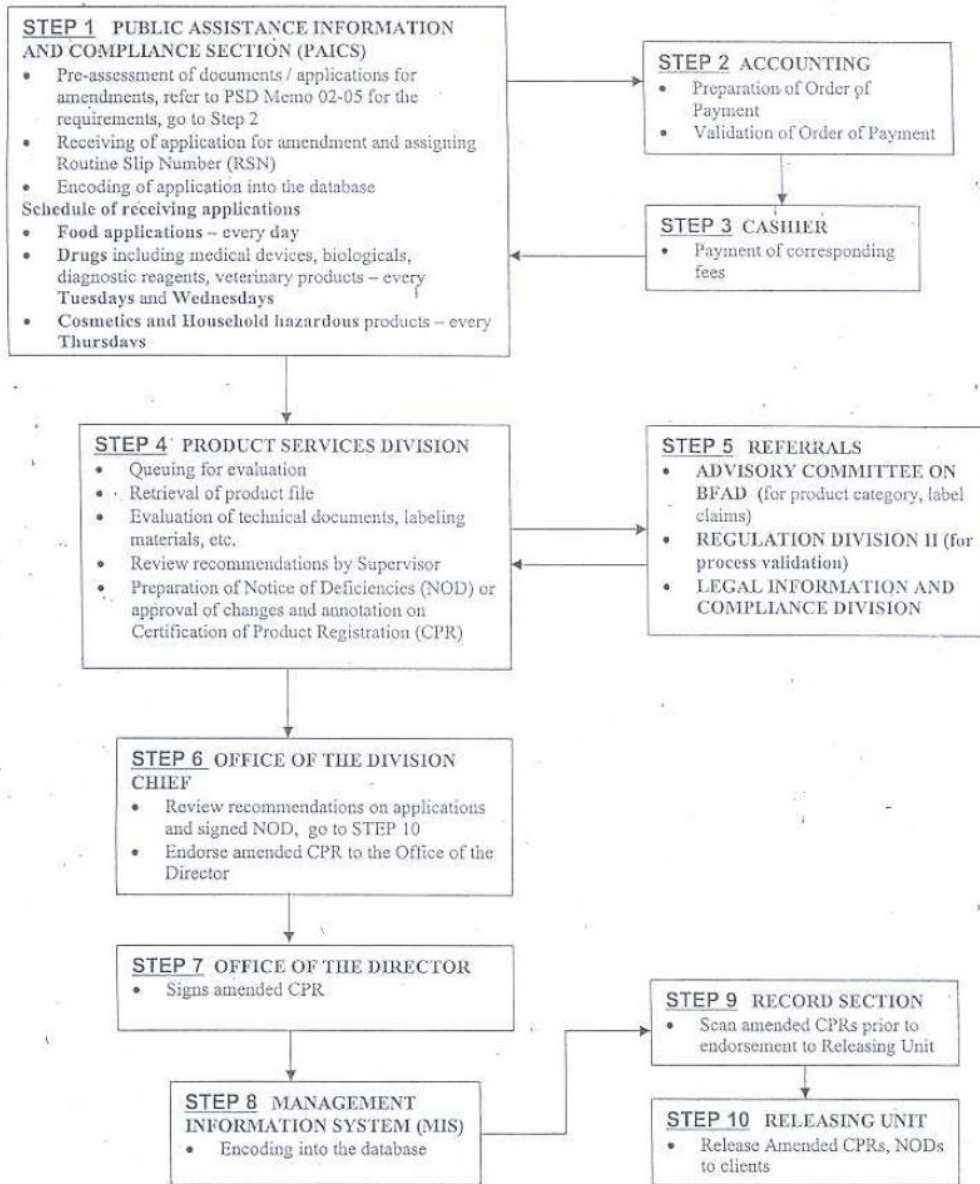
PROCEDURE FOR FILING A REQUEST FOR LABORATORY ANALYSIS
OF GOVERNMENT PROCURED OR COLLECTED/COMPLAINT SAMPLES



Appendix 11

iii. Products

PROCEDURE FOR REQUEST FOR POST APPROVAL CHANGES OF REGISTERED PRODUCT (AMENDMENTS)



Appendix 11

Annex A

**PROCEDURE IN PROCESSING AND RELEASING OF APPLICATIONS FOR
AUTOMATIC RENEWAL REGISTRATION**

1. All applications for automatic renewal registration received at the Public Assistance Information and Compliance Section (PAICS/Room 101) shall be forwarded at the different sections of the Product Services Division
PREPACKAGED FOOD AND FOOD PRODUCTS – (Room 110)
DRUGS, MEDICAL DEVICES AND HHS – (Room 106)
2. Assigned PSD staff shall assess the completeness and accuracy/correctness of the submitted documents.
3. Assigned PSD staff shall request for the duplicate copy of the Certificate of Product Registration (CPR) at Administrative Division - Records Section (Room 113).
4. The CPR of approved application shall be revalidated or a new CPR shall be issued.
5. Revalidated CPR shall be signed by the PSD Chief while re-issued CPR by the BFAD Director.
6. Signed CPR will be forwarded to Policy, Planning and Advocacy Division (PPAD) for encoding
7. PPAD will endorse the CPR to Administrative Division - Records Section for scanning
8. Administrative Division - Records Section will forward scanned CPR to Releasing Unit (Room 113)

All revalidated CPR(s) shall be released at the Releasing Section (Room100) according to the following schedule:

Receipt of Application		Release of CPR	
7:30 am	10:00 am	11:00	12:00
10:00 am	12:00 pm	2:00	3:00
1:00 pm	2:00 pm	3:00	4:00
2:00 pm	3:00 pm	4:00	5:00
After 3pm		The following day 9am – 10am	



FLOWCHART OF BFAD PRODUCT REGISTRATION

MONDAYS

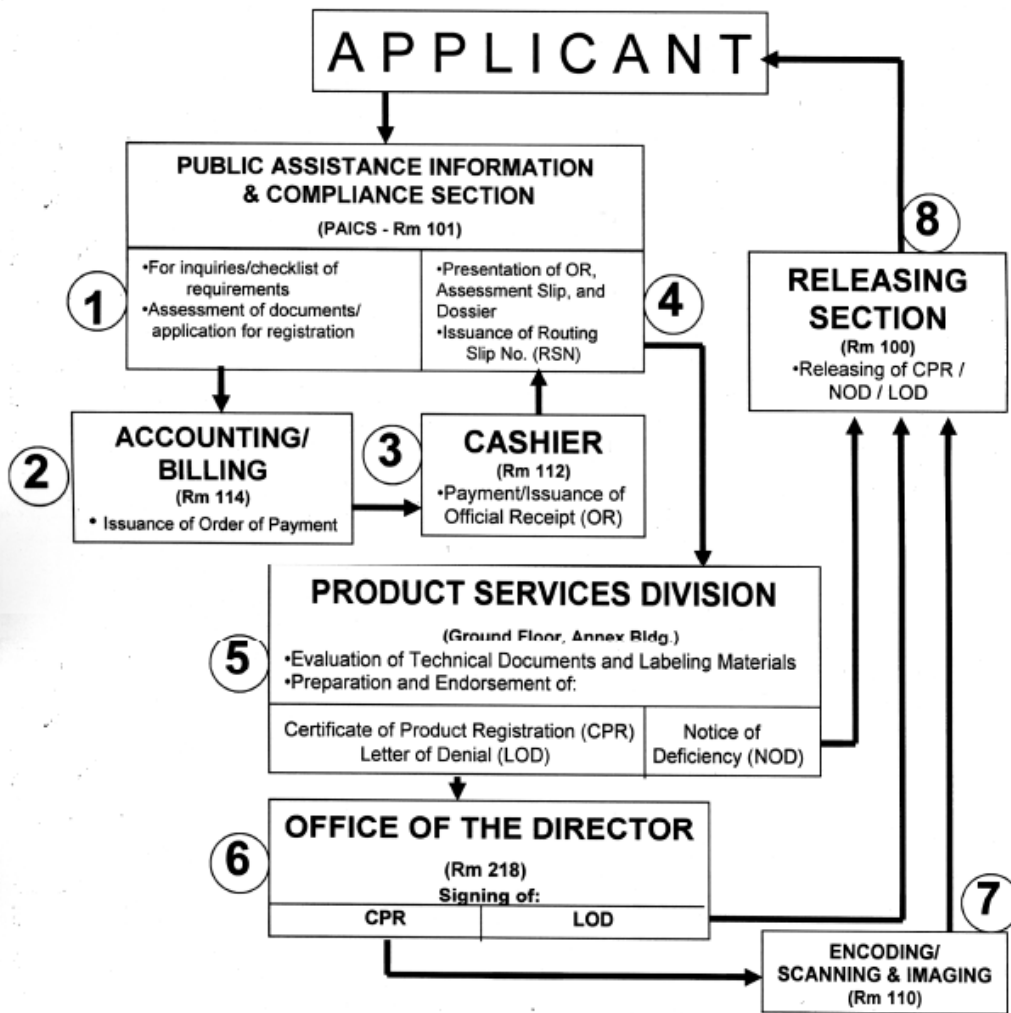
PROCESSED FOOD

TUESDAYS & WEDNESDAYS

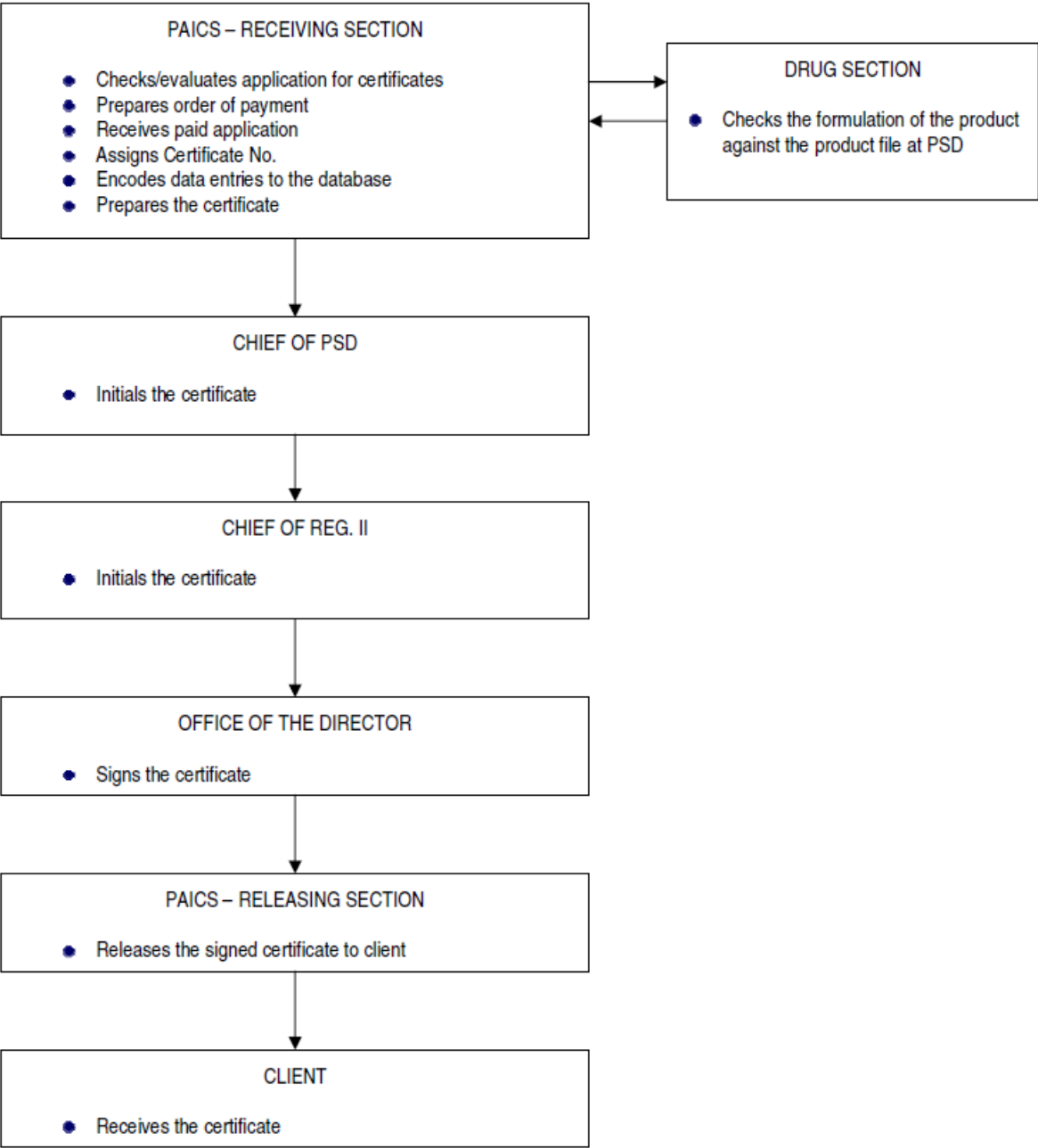
DRUGS, MEDICAL DEVICES, DIAGNOSTIC REAGENTS, DRUG TESTING KITS,
 BIOLOGICALS AND VACCINES, HERBAL & TRADITIONALLY USED MEDICINES

THURSDAYS

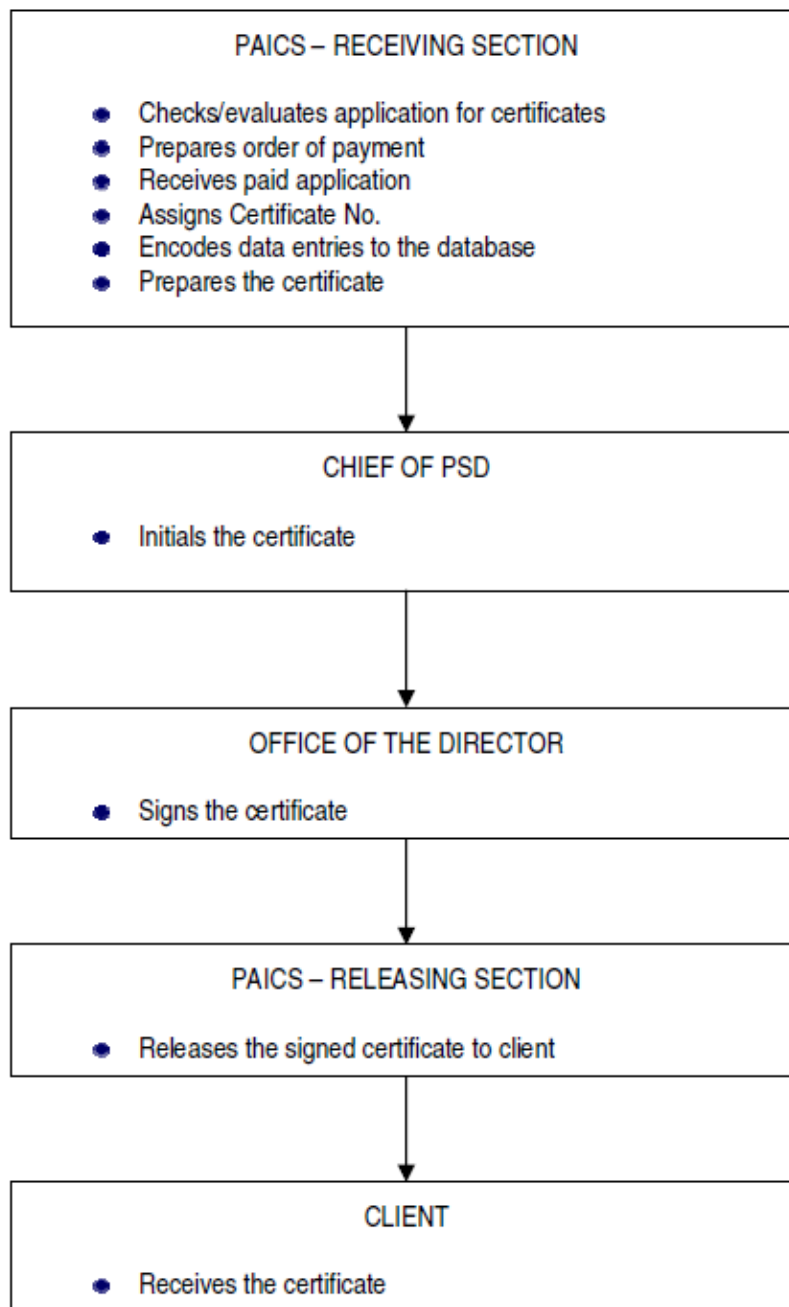
COSMETICS, HOUSEHOLD HAZARDOUS SUBSTANCES



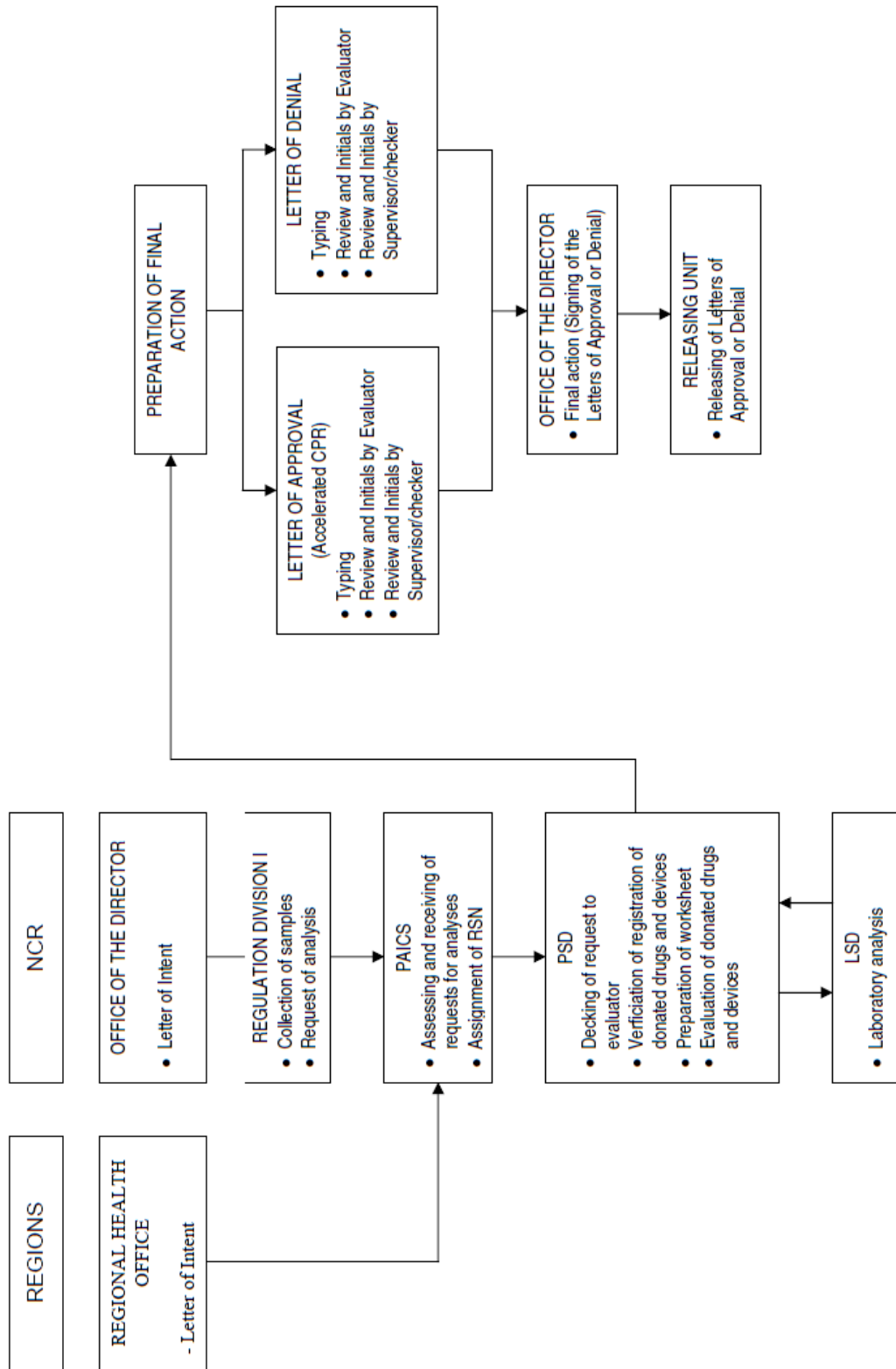
**FLOWCHART FOR ISSUANCE OF CERTIFICATE OF PHARM'L PRODUCTS
PRODUCT SERVICES DIVISION – PAICS SECTION**



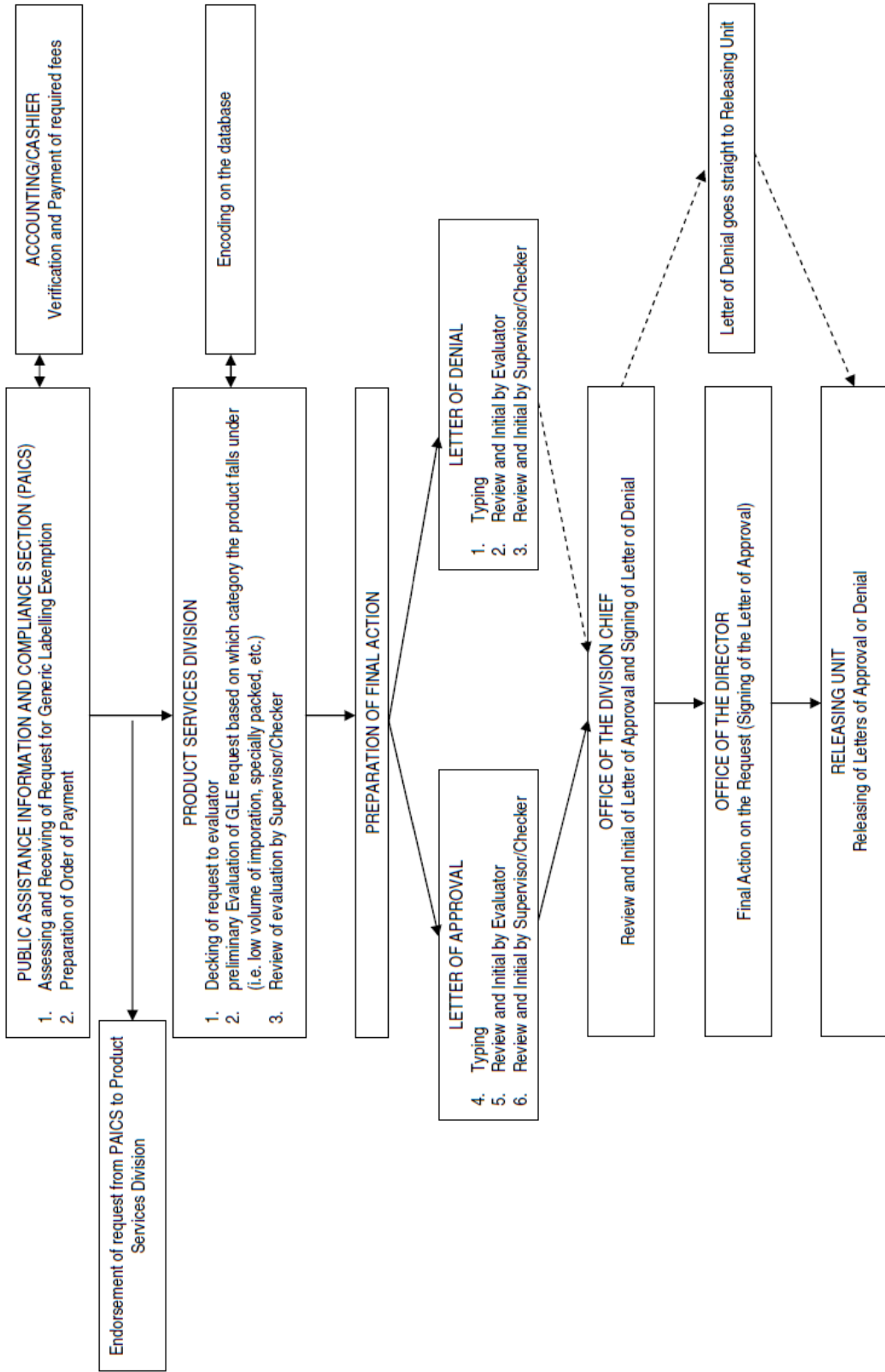
FLOWCHART FOR ISSUANCE OF FREE SALE & EXPORT CERTIFICATE PRODUCT SERVICES DIVISION – PAICS SECTION



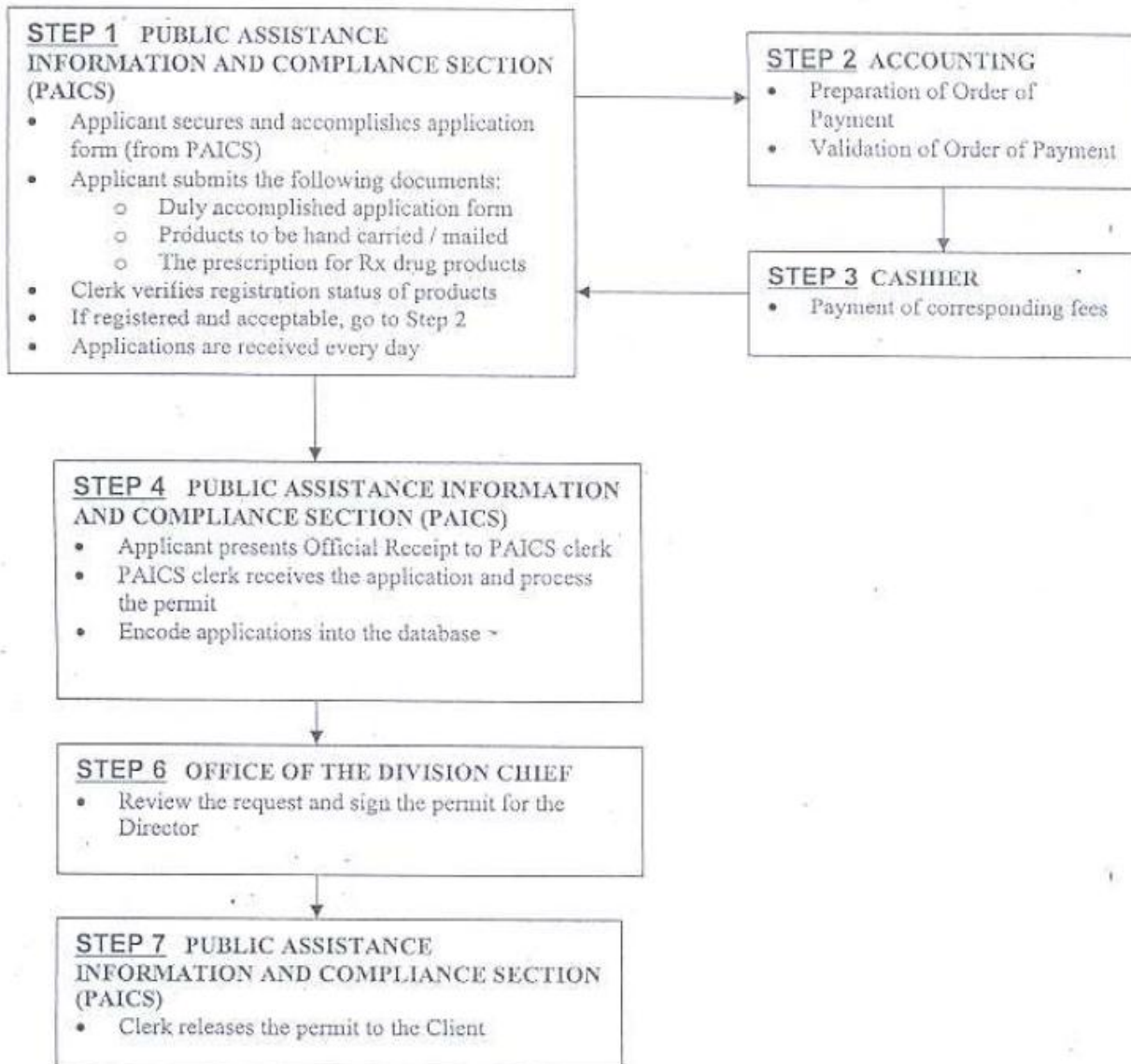
FLOWCHART OF REQUESTS FOR DONATION



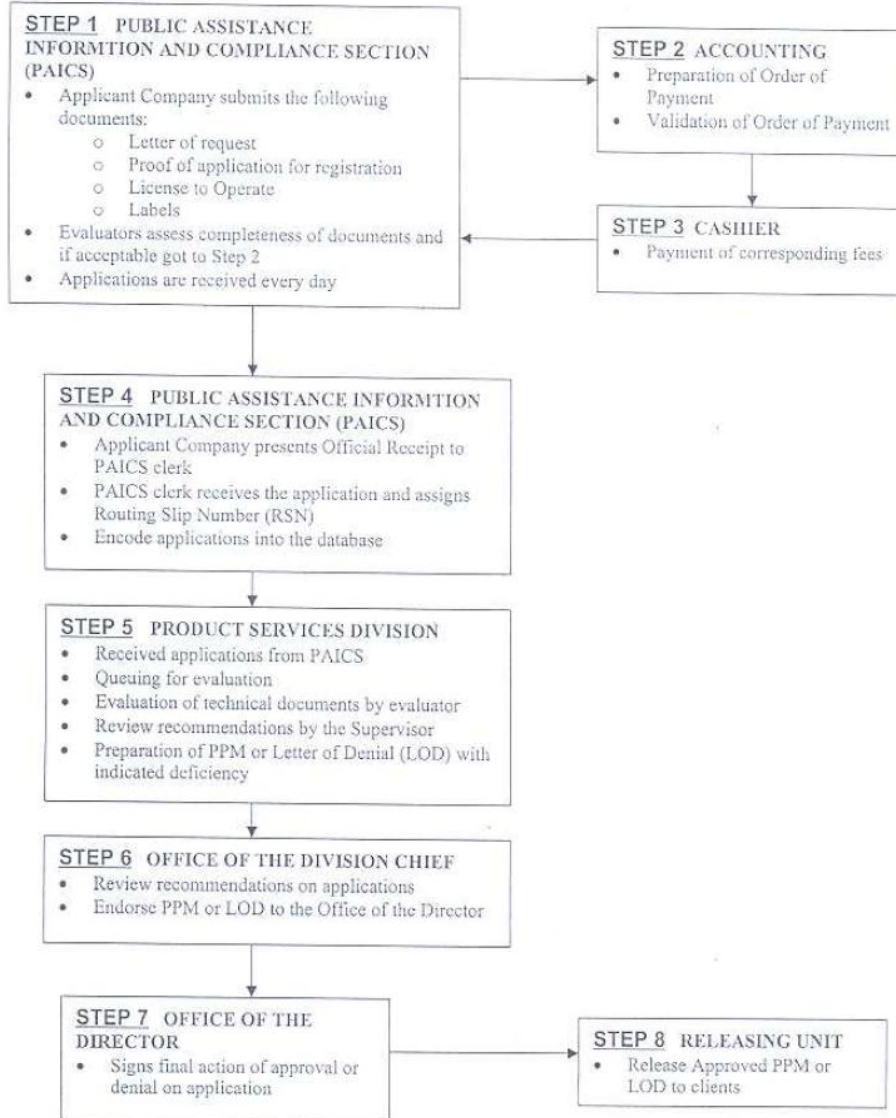
FLOWCHART OF REQUESTS FOR GNERIC LABELLING EXEMPTION



PROCEDURE FOR REQUESTING PERMIT TO HAND CARRY / MAILING PERMIT



PROCEDURE FOR REQUESTING PROVISIONAL PERMIT TO MARKET (PPM)



Appendix 19

B. Source Codes

i. models/Document_Model.php

```
<?php
class Document_model extends CI_Model {

public function __construct()
{
$this->load->database();
}

//for login

public function isUser($userName,$password) {
$query = $this->db->query("SELECT * FROM
`USERACCOUNT` WHERE `USERNAME`='".$userName.'" AND
`PASSWORD`='".$md5($password)."'");
if (!$query||count($query->result_array()) == 0){
return false;
} else {
$res = $query->row_array();
$newdata = array(
'userid' => $res['USERID'],
'username' => $res['USERNAME'],
'firstname' => $res['FIRSTNAME'],
'lastname' => $res['LASTNAME'],
'stationcode' => $res['STATIONCODE'],
'usertype' => $res['USERTYPE'],
'email' => $res['EMAIL'],
'phone' => $res['PHONE'],
'address' => $res['ADDRESS'],
'theme' => $res['THEME'],
'login_state' => TRUE
);
$this->session->set_userdata($newdata);

return true;
}
}

//for document type

//get

public function get_applicantDocuRowCount($userId){
$query0 = $this->db->query("SELECT * FROM
COMPANYUSER WHERE COMPANYUSER.USERID =
'$userId'");
$companies = $query0->result_array();

$returnArr = array();
$i = 0;
foreach ($companies as $company) {
//$query1 = $this->db->query("SELECT * FROM
`docucoremadata` DOCUCOR, DOCUTYPE DT,
COMPANYDOCU COMPDOC, DOCULOC DOCL, STATION
ST WHERE ST.STATIONCODE = DOCL.STATIONCODE
AND DT.TYPECODE = DOCUCOR.TYPECODE AND
DOCL.TIMEIN = (SELECT MAX(DOCLOC.TIMEIN) FROM
DOCULOC DOCLOC WHERE DOCLOC.BARCODE =
DOCL.BARCODE) AND COMPDOC.COMPANYCODE =
'".$company['COMPANYCODE']."' AND
DOCUCOR.BARCODE=DOCL.BARCODE");
$query1 = $this->db->query("SELECT * FROM
DOCUCOREMETADATA DOCUD, DOCUTYPE DOCUT,
STATION STAT, DOCULOC DOCUL, COMPANYDOCU
COMPDOC WHERE DOCUD.BARCODE =
DOCUL.BARCODE AND DOCUT.TYPECODE =
```

```
DOCUD.TYPECODE AND DOCUL.STATIONCODE =
STAT.STATIONCODE AND DOCUL.TIMEIN = (SELECT
MAX(DOCLOC.TIMEIN) FROM DOCULOC DOCLOC
WHERE DOCLOC.BARCODE = DOCUD.BARCODE) AND
COMPDOC.BARCODE = DOCUD.BARCODE AND
COMPDOC.COMPANYCODE =
'".$company['COMPANYCODE']."'";);
$docus = $query1->result_array();
foreach ($docus as $docu) {
$returnArr[$i] = $docu;
$i++;
}
}

return $returnArr;
}
```

```
public function get_docuRowCount($stationCode){
if (isset($stationCode)) {
$query = $this->db->query("SELECT * FROM DOCULOC
DOCL WHERE DOCL.STATIONCODE = $stationCode AND
DOCL.TIMEIN = (SELECT MAX(DOCLOC.TIMEIN) FROM
DOCULOC DOCLOC WHERE DOCLOC.BARCODE =
DOCL.BARCODE);");
} else {
$query = $this->db->query("SELECT * FROM DOCULOC
DOCL WHERE DOCL.TIMEIN = (SELECT
MAX(DOCLOC.TIMEIN) FROM DOCULOC DOCLOC
WHERE DOCLOC.BARCODE = DOCL.BARCODE);");
}
return $query->result_array();
}
```

```
public function get_docs($sidx, $sord, $start, $limit,
$queryString=1,$stationCode){
if (isset($stationCode)) {
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATA DOCUD, DOCUTYPE DOCUT,
STATION STAT, DOCULOC DOCUL WHERE
DOCUD.BARCODE = DOCUL.BARCODE AND
DOCUT.TYPECODE = DOCUD.TYPECODE AND
DOCUL.STATIONCODE = STAT.STATIONCODE AND
DOCUL.STATIONCODE = $stationCode AND DOCUL.TIMEIN
= (SELECT MAX(DOCLOC.TIMEIN) FROM DOCULOC
DOCLOC WHERE DOCLOC.BARCODE =
DOCUD.BARCODE) AND $queryString ORDER BY $sidx $sord
LIMIT $start, $limit;");
} else {
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATA DOCUD, DOCUTYPE DOCUT,
STATION STAT, DOCULOC DOCUL WHERE
DOCUD.BARCODE = DOCUL.BARCODE AND
DOCUT.TYPECODE = DOCUD.TYPECODE AND
DOCUL.STATIONCODE = STAT.STATIONCODE AND
DOCUL.TIMEIN = (SELECT MAX(DOCLOC.TIMEIN) FROM
DOCULOC DOCLOC WHERE DOCLOC.BARCODE =
DOCUD.BARCODE) AND $queryString ORDER BY $sidx $sord
LIMIT $start, $limit;");
}
}
```

```
return $query->result_array();
}
```

```
public function get_arcdocs($sidx, $sord, $start, $limit,
$queryString=1){
```

```

$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATAARC WHERE $queryString ORDER
BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_arcdoccount(){
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATAARC;");

return $query->result_array();
}

public function get_docuType($sidx, $sord, $start , $limit,
$queryString=1) {
$query = $this->db->query("SELECT * FROM `DOCUTYPE`
WHERE $queryString ORDER BY $sidx $sord LIMIT $start ,
$limit;");

return $query->result_array();
}

public function get_users($sidx, $sord, $start , $limit,
$queryString=1) {
$query = $this->db->query("SELECT * FROM
`USERACCOUNT`, `STATION` WHERE
STATION.STATIONCODE =
USERACCOUNT.STATIONCODE AND $queryString ORDER
BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_users_station($sidx, $sord, $start , $limit,
$queryString=1, $stationCode) {
$query = $this->db->query("SELECT * FROM
`USERACCOUNT`, `STATION` WHERE
STATION.STATIONCODE =
USERACCOUNT.STATIONCODE AND
STATION.STATIONCODE = $stationCode AND $queryString
ORDER BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_stationUsers($sidx, $sord, $start , $limit,
$queryString=1, $stationCode) {
$query = $this->db->query("SELECT * FROM
`USERACCOUNT`, `STATION` WHERE
STATION.STATIONCODE =
USERACCOUNT.STATIONCODE AND
USERACCOUNT.STATIONCODE = $stationCode AND
$queryString ORDER BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_docuTypeRowCount() {
$query = $this->db->query("SELECT * FROM DOCUTYPE;");

return $query->result_array();
}

public function get_docuTypeWithLimit($stationCode) {
$query = $this->db->query("SELECT * FROM DOCUTYPE,
CREATEPERMISSION WHERE DOCUTYPE.TYPECODE =
CREATEPERMISSION.TYPECODE AND
CREATEPERMISSION.STATIONCODE = $stationCode AND
CREATEPERMISSION.PERTYPE = 'y;");

```

```

return $query->result_array();
}

public function getDocLock($barCode,$userid) {
$query = $this->db->query("SELECT * FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode AND
DOCLOCK.USERID = $userid;");

return $query->result_array();
}

public function getDocLockExists($barCode,$userid) {
$query = $this->db->query("SELECT * FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode AND
DOCLOCK.USERID <> $userid AND DOCLOCK.STATUS <>
'n;");

return $query->result_array();
}

public function get_PermissionCount($stationCode,$typeCode) {
$query = $this->db->query("SELECT * FROM `PERMISSION`,
PARAM WHERE PARAM.PARAMCODE =
PERMISSION.PARAMCODE AND
PERMISSION.STATIONCODE=$stationCode' AND
PARAM.TYPECODE = '$typeCode;");

return $query->result_array();
}

public function get_stationRowCount() {
$query = $this->db->query("SELECT * FROM STATION;");

return $query->result_array();
}

public function get_usersRowCount() {
$query = $this->db->query("SELECT * FROM USERACCOUNT
WHERE USERACCOUNT.STATIONCODE != 0;");

return $query->result_array();
}

public function get_usersRowCountStation($stationCode) {
$query = $this->db->query("SELECT * FROM
`USERACCOUNT`, `STATION` WHERE
STATION.STATIONCODE =
USERACCOUNT.STATIONCODE AND
STATION.STATIONCODE = $stationCode;");

return $query->result_array();
}

public function get_usersStationRowCount($stationCode) {
$query = $this->db->query("SELECT * FROM USERACCOUNT
WHERE USERACCOUNT.STATIONCODE = $stationCode;");

return $query->result_array();
}

public function get_dataTypeCount() {
$query = $this->db->query("SELECT * FROM DATATYPE;");

return $query->result_array();
}

public function get_companies() {
$query = $this->db->query("SELECT * FROM COMPANY;");

```

```

return $query->result_array();
}

public function get_companies_with_limits($sidx, $sord, $start ,
$limit, $queryString=1) {
$query = $this->db->query("SELECT * FROM `COMPANY`
WHERE $queryString ORDER BY $sidx $sord LIMIT $start ,
$limit;");

return $query->result_array();
}

public function get_dataType() {
$query = $this->db->query("SELECT * FROM DATATYPE;");

return $query->result_array();
}

public function get_profile(){
$userid = $this->session->userdata('userid');
if($this->session->userdata('usertype')== "admin"){
$query = $this->db->query("SELECT * FROM USERACCOUNT
WHERE USERID = '$userid';");
return $query->row_array();
} else {
$query = $this->db->query("SELECT * FROM USERACCOUNT,
STATION WHERE
STATION.STATIONCODE=USERACCOUNT.STATIONCODE
AND USERID = '$userid';");
return $query->row_array();
}
}

public function get_documentTypeMeta($typeCode) {
$query = $this->db->query("SELECT * FROM DOCUTYPE
WHERE DOCUTYPE.TYPECODE = '$typeCode';");

return $query->row_array();
}

public function get_documentTypeStation($typeCode) {
$query = $this->db->query("SELECT * FROM
DOCUPATHSTATION, STATION WHERE
STATION.STATIONCODE =
DOCUPATHSTATION.STATIONCODE AND
DOCUPATHSTATION.TYPECODE = '$typeCode';");

return $query->result_array();
}

public function get_stationMetaData($stationCode) {
$query = $this->db->query("SELECT * FROM STATION
WHERE STATION.STATIONCODE = '$stationCode';");

return $query->row_array();
}

public function get_documentTypeParamStation($typeCode,
$stationCode , $sidx=1, $sord="", $start=0 , $limit=19,
$queryString="1") {
$query = $this->db->query("SELECT * FROM PARAM WHERE
PARAM.TYPECODE = '$typeCode' AND
PARAM.PARAMCODE NOT IN(SELECT
PERMISSION.PARAMCODE FROM `PERMISSION` WHERE
`STATIONCODE`=$stationCode) AND $queryString ORDER
BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

```

```

public function get_documentTypeParam($typeCode, $sidx=1,
$sord="", $start=0 , $limit=19, $queryString="1") {
$query = $this->db->query("SELECT * FROM PARAM WHERE
PARAM.TYPECODE = '$typeCode' AND $queryString ORDER
BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_documentTypeParaWOLimiters($typeCode) {
$query = $this->db->query("SELECT * FROM PARAM WHERE
PARAM.TYPECODE = '$typeCode' ORDER BY
PARAMORDER");

return $query->result_array();
}

public function
get_documentParaWOLimiters($barCode,$stationCode) {
$query = $this->db->query("SELECT * FROM METAPARAM,
STATION, METAPERMISSION WHERE
METAPARAM.BARCODE = '$barCode' AND
STATION.STATIONCODE = '$stationCode' AND
METAPARAM.METAPARAMCODE =
METAPERMISSION.METAPARAMCODE AND
STATION.STATIONCODE =
METAPERMISSION.STATIONCODE AND
(METAPERMISSION.PERTYPE = 'r' OR
METAPERMISSION.PERTYPE = 'rw') ORDER BY
PARAMORDER");

return $query->result_array();
}

public function get_documentParaWOLimiters($barCode) {
$query = $this->db->query("SELECT * FROM METAPARAM
WHERE METAPARAM.BARCODE = '$barCode' ORDER BY
PARAMORDER");

return $query->result_array();
}

public function get_documentMetaPara($sarcCode) {
$query = $this->db->query("SELECT * FROM
METAPARAMARC WHERE METAPARAMARC.ARCCODE =
'$sarcCode' ORDER BY PARAMORDER");

return $query->result_array();
}

public function get_dataTypeList($sidx, $sord, $start , $limit,
$queryString){
$query = $this->db->query("SELECT * FROM DATATYPE
WHERE $queryString ORDER BY $sidx $sord LIMIT $start ,
$limit;");

return $query->result_array();
}

public function get_permissionList($stationCode, $typeCode,
$sidx, $sord, $start , $limit, $queryString){
$query = $this->db->query("SELECT * FROM `PERMISSION`,
`PARAM` WHERE PERMISSION.PARAMCODE =
PARAM.PARAMCODE AND
PERMISSION.STATIONCODE=$stationCode AND
PARAM.TYPECODE = '$typeCode' AND $queryString ORDER
BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

```



```

public function get_documentTypeParamChoice() {
$query = $this->db->query("SELECT * FROM
PARAMCHOICE;");

return $query->result_array();
}

public function get_documentTypePath($typeCode) {
$query = $this->db->query("SELECT * FROM DOCUPATHCON
WHERE DOCUPATHCON.TYPECODE = '$typeCode';");

return $query->result_array();
}

public function get_otherStations($typeCode) {
$query = $this->db->query("SELECT * FROM STATION
WHERE STATION.STATIONCODE NOT IN (SELECT
DOCUPATHSTATION.STATIONCODE FROM
DOCUPATHSTATION WHERE
DOCUPATHSTATION.TYPECODE = '$typeCode');");

return $query->result_array();
}

public function get_pathInstances($typeCode,$sidx, $sord, $start ,
$limit, $queryString=1) {
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATA, DOCULOC, STATION WHERE
DOCULOC.STATIONCODE = STATION.STATIONCODE
AND DOCULOC.BARCODE =
DOCUCOREMETADATA.BARCODE AND
DOCULOC.TIMEIN = (SELECT MAX(DOCLOC.TIMEIN)
FROM DOCULOC DOCLOC WHERE DOCLOC.BARCODE =
DOCUCOREMETADATA.BARCODE) AND
DOCUCOREMETADATA.TYPECODE = '$typeCode' AND
$queryString ORDER BY $sidx $sord LIMIT $start , $limit;");

return $query->result_array();
}

public function get_pathInstancesCount($typeCode){
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATA, DOCULOC, STATION WHERE
DOCULOC.STATIONCODE = STATION.STATIONCODE
AND DOCULOC.BARCODE =
DOCUCOREMETADATA.BARCODE AND
DOCULOC.TIMEIN = (SELECT MAX(DOCLOC.TIMEIN)
FROM DOCULOC DOCLOC WHERE DOCLOC.BARCODE =
DOCUCOREMETADATA.BARCODE) AND
DOCUCOREMETADATA.TYPECODE = '$typeCode';");

return $query->result_array();
}

public function get_stations($sidx, $sord, $start , $limit,
$queryString=1) {
$query = $this->db->query("SELECT * FROM `STATION`
WHERE $queryString ORDER BY $sidx $sord LIMIT $start ,
$limit;");

return $query->result_array();
}

public function get_stationdoccount($stationCode){
$query = $this->db->query("SELECT COUNT(*) AS
DOCCOUNT FROM DOCULOC DOCL WHERE
DOCL.STATIONCODE = $stationCode AND DOCL.TIMEIN =
(SELECT MAX(DOCLOC.TIMEIN) FROM DOCULOC

```

```

DOCLOC WHERE DOCLOC.BARCODE =
DOCL.BARCODE);");

return $query->row_array();
}

public function getMetaConnections ($barcode) {
$query = $this->db->query("SELECT * FROM
`DOCUMETAPATHCON` WHERE
DOCUMETAPATHCON.BARCODE = $barcode;");

return $query->result_array();
}

public function getMetaArcConnections ($arcCode) {
$query = $this->db->query("SELECT * FROM
`DOCUMETAPATHCONARC` WHERE
DOCUMETAPATHCONARC.ARCCODE = $arcCode;");

return $query->result_array();
}

public function getMetaStations ($barcode) {
$query = $this->db->query("SELECT * FROM
`DOCUMETAPATHSTATION`, `STATION` WHERE
DOCUMETAPATHSTATION.BARCODE = $barcode AND
STATION.STATIONCODE =
DOCUMETAPATHSTATION.STATIONCODE;");

return $query->result_array();
}

public function getMetaArcStations ($arcCode) {
$query = $this->db->query("SELECT * FROM
`DOCUMETAPATHSTATIONARC` WHERE
DOCUMETAPATHSTATIONARC.ARCCODE = $arcCode;");

return $query->result_array();
}

public function getDocLocStartDate ($barcode) {
$query = $this->db->query("SELECT MIN(DOCULOC.TIMEIN)
AS MINDATE FROM `DOCULOC` WHERE
DOCULOC.BARCODE = $barcode;");

return $query->row_array();
}

public function getDocLocArcStartDate ($arcCode) {
$query = $this->db->query("SELECT
MIN(DOCULOCAARC.TIMEIN) AS MINDATE FROM
`DOCULOCAARC` WHERE DOCULOCAARC.ARCCODE =
$arcCode;");

return $query->row_array();
}

public function getDocLocArcEndDate ($arcCode) {
$query = $this->db->query("SELECT
MAX(DOCULOCAARC.TIMEOUT) AS MAXDATE FROM
`DOCULOCAARC` WHERE DOCULOCAARC.ARCCODE =
$arcCode;");

return $query->row_array();
}

public function getStationStartDate ($barcode,$stationCode) {
$query = $this->db->query("SELECT
MAX(DOCULOC.TIMEIN) AS MINDATE FROM `DOCULOC`

```

```

WHERE DOCULOC.STATIONCODE = $stationCode AND
DOCULOC.BARCODE = $barcode;");

return $query->row_array();
}

public function getDocLoc ($barcode) {
$query = $this->db->query("SELECT * FROM `DOCULOC`,
`STATION` WHERE DOCULOC.STATIONCODE =
STATION.STATIONCODE AND DOCULOC.BARCODE =
$barcode ORDER BY DOCULOC.TIMEIN;");

return $query->result_array();
}

public function getDocArcLoc ($arCcode) {
$query = $this->db->query("SELECT * FROM
`DOCULOCAARC` WHERE DOCULOCAARC.ARCCODE =
$arCcode ORDER BY DOCULOCAARC.TIMEIN;");

return $query->result_array();
}

public function getDocLocForTable ($sidx, $sord, $start, $limit,
$queryString=1, $barcode) {
$query = $this->db->query("SELECT * FROM `DOCULOC`,
`STATION` WHERE DOCULOC.STATIONCODE =
STATION.STATIONCODE AND DOCULOC.BARCODE =
$barcode AND $queryString ORDER BY $sidx $sord LIMIT
$start, $limit;");

return $query->result_array();
}

public function getDocArcLocForTable ($sidx, $sord, $start,
$limit, $queryString=1, $arCcode) {
$query = $this->db->query("SELECT * FROM
`DOCULOCAARC` WHERE DOCULOCAARC.ARCCODE =
$arCcode AND $queryString ORDER BY $sidx $sord LIMIT
$start, $limit;");

return $query->result_array();
}

public function getDocumentMetaData ($barcode) {
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATA DOCUD, DOCUTYPE DOCUT,
STATION STAT, DOCULOC DOCUL WHERE
DOCUD.BARCODE = $barcode AND DOCUD.BARCODE =
DOCUL.BARCODE AND DOCUT.TYPECODE =
DOCUD.TYPECODE AND DOCUL.STATIONCODE =
STAT.STATIONCODE AND DOCUL.TIMEIN = (SELECT
MAX(DOCLOC.TIMEIN) FROM DOCULOC DOCLOC
WHERE DOCLOC.BARCODE = DOCUD.BARCODE);");

return $query->row_array();
}

public function getArcDocumentMetaData ($arCcode) {
$query = $this->db->query("SELECT * FROM
DOCUCOREMETADATAARC WHERE
DOCUCOREMETADATAARC.ARCCODE = $arCcode ;");

return $query->row_array();
}

public function getUserMeta($userId) {
$query = $this->db->query("SELECT * FROM USERACCOUNT
WHERE USERACCOUNT.USERID = $userId ;");

```

```

return $query->row_array();
}

public function getCompanies ($barcode) {
$query = $this->db->query("SELECT * FROM
`COMPANYDOCU`, `COMPANY` WHERE
COMPANY.COMPANYCODE =
COMPANYDOCU.COMPANYCODE AND
COMPANYDOCU.BARCODE = $barcode");

return $query->result_array();
}

public function getArcDocCompanies ($arCcode) {
$query = $this->db->query("SELECT * FROM
`COMPANYDOCUARC` WHERE
COMPANYDOCUARC.ARCCODE = $arCcode;");

return $query->result_array();
}

public function getAllCompanies () {
$query = $this->db->query("SELECT * FROM `COMPANY`;");

return $query->result_array();
}

public function getUserCompaniesChoice ($userId) {
$query = $this->db->query("SELECT * FROM `COMPANY`
WHERE COMPANY.COMPANYCODE NOT IN (SELECT
COMPANYUSER.COMPANYCODE FROM COMPANYUSER
WHERE COMPANYUSER.USERID = '$userId');");

return $query->result_array();
}

public function getUserCompanies ($userId) {
$query = $this->db->query("SELECT * FROM
COMPANYUSER, COMPANY WHERE
COMPANY.COMPANYCODE =
COMPANYUSER.COMPANYCODE AND
COMPANYUSER.USERID = '$userId';");

return $query->result_array();
}

public function getCreatePermission($typeCode,$stationCode) {
$query = $this->db->query("SELECT * FROM
`CREATEPERMISSION` WHERE
CREATEPERMISSION.TYPECODE = $typeCode AND
CREATEPERMISSION.STATIONCODE = $stationCode");

return $query->row_array();
}

public function getMailTo($barCode){
$query = $this->db->query("SELECT * FROM
`COMPANYDOCU`,`COMPANYUSER`,`USERACCOUNT`
WHERE USERACCOUNT.USERID =
COMPANYUSER.USERID AND
COMPANYUSER.COMPANYCODE =
COMPANYDOCU.COMPANYCODE AND
COMPANYDOCU.BARCODE = $barCode");

return $query->result_array();
}

public function get_paramChoices($paramCode) {

```

```

$query = $this->db->query("SELECT * FROM
`PARAMCHOICE` WHERE PARAMCHOICE.PARAMCODE =
$paramsCode");

return $query->result_array();
}

public function get_metaParamChoices($metaParamCode) {
$query = $this->db->query("SELECT * FROM
`METAPARAMCHOICE` WHERE
METAPARAMCHOICE.METAPARAMCODE =
$metaParamCode");

return $query->result_array();
}

public function get_metaArcParamChoices($arcMetaParamCode)
{
$query = $this->db->query("SELECT * FROM
`METAPARAMCHOICEARC` WHERE
METAPARAMCHOICEARC.METAPARAMARCCODE =
$arcMetaParamCode");

return $query->result_array();
}

public function getNextStations($barCode,$stationCode) {
$query = $this->db->query("SELECT * FROM
`DOCUMETAPATHCON` WHERE
DOCUMETAPATHCON.BARCODE = '$barCode' AND
DOCUMETAPATHCON.METACONSOURCE = '$stationCode'
;");

return $query->result_array();
}

public function getStationName($stationCode) {
$query = $this->db->query("SELECT STATIONNAME FROM
`STATION` WHERE STATION.STATIONCODE =
'$stationCode';");

return $query->row_array();
}

//edit

public function
edit_user($userid,$username,$usertype,$password,$stationcode,$fi
rstname,$lastname,$email,$phone,$address){
$query = $this->db->query("SELECT * FROM
`USERACCOUNT` WHERE `USERID` = $userid;");

$query1 = $query->row_array();
$pwd = $query1['PASSWORD'];

if($pwd==$password){
$queryString = "UPDATE `USERACCOUNT` SET
`USERNAME` = '$username', `USERTYPE` = '$usertype',
`STATIONCODE`=$stationcode', `FIRSTNAME`=$firstname',
`LASTNAME`=$lastname', `EMAIL`=$email',
`PHONE`=$phone', `ADDRESS`=$address' WHERE `USERID`
= $userid;";
}else{
$queryString = "UPDATE `USERACCOUNT` SET
`USERNAME` = '$username', `USERTYPE` = '$usertype',
`PASSWORD` = '".md5($password)."',
`STATIONCODE`=$stationcode', `FIRSTNAME`=$firstname',
`LASTNAME`=$lastname', `EMAIL`=$email',
`PHONE`=$phone', `ADDRESS`=$address' WHERE `USERID`
= $userid;";
}

}

}

$query = $this->db->query($queryString);

if (!$query) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color='\"#ff0000\"' size='\"1\"'>$errnum =
$errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>User saved</font>";
}
}

public function edit_typePath($typeCode,$connections,$stations) {

$query1 = $this->db->query("DELETE FROM
DOCUPATHSTATION WHERE TYPECODE = $typeCode;");
$query2 = $this->db->query("DELETE FROM DOCUPATHCON
WHERE TYPECODE = $typeCode;");

if(!$query1||!$query2){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {

if (isset($stations)&&is_array($stations)) {
foreach ($stations as $stations_item):
$query4 = $this->db->query("INSERT INTO
`DOCUPATHSTATION`
(`TYPECODE`,`STATIONCODE`,`LEFTPOS`,`TOPPOS`)
VALUES
('$typeCode','.$stations_item[0].','.$stations_item[1].','.$statio
ns_item[2].'");");
endforeach;
}
if (isset($connections)||is_array($connections)) {
foreach ($connections as $connections_item):
$query3 = $this->db->query("INSERT INTO `DOCUPATHCON`
(`TYPECODE`,`CONSOURCE`,`CONTARGET`,`CONDESCR`)
VALUES
('$typeCode','.$connections_item[0].','.$connections_item[1].','.$
connections_item[2].'");");
endforeach;
}

if((isset($query4)&&!$query4)||isset($query3)&&!$query3){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Path Saved</font>";
}
}

public function edit_typeMeta($name,$desc,$typeCode){
$query = $this->db->query("UPDATE `DOCUTYPE` SET
`TYPENAME` = '$name', `TYPEDESC` = '$desc' WHERE
`TYPECODE` = $typeCode;");

if (!$query) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
}
}

```

```

} else {
return "<font color=#1c9c00" size=2">Name and description
saved</font>";
}
}

public function
edit_parameter($name,$options="", $default="", $description="", $st
ypeCode,$paramcode,$order=0,$colSize="", $inputtype=""){
$query = $this->db->query("SELECT `PARAMORDER` FROM
`PARAM` WHERE `PARAMCODE` = $paramcode;");
$query1 = $query->row_array();
$orderOrg = $query1['PARAMORDER'];

$query = $this->db->query("SELECT MAX(`PARAMORDER`)
AS MAXNUM FROM `PARAM` WHERE
`TYPECODE`=$typeCode;");
$query1 = $query->row_array();
$orderMax = $query1['MAXNUM'];

if($order=="||$order==0||$order==null){
$order=$orderOrg;
}

if($orderMax<$order){
$order = $orderMax;
}

if($orderOrg>$order){
$query = $this->db->query("UPDATE `PARAM` SET
`PARAMORDER`=`PARAMORDER`+1 WHERE
`TYPECODE`=$typeCode AND `PARAMORDER` >= $order
AND `PARAMORDER` < $orderOrg");
} else if ($orderOrg<$order){
$query = $this->db->query("UPDATE `PARAM` SET
`PARAMORDER`=`PARAMORDER`-1 WHERE
`TYPECODE`=$typeCode AND `PARAMORDER` <= $order
AND `PARAMORDER` > $orderOrg");
}

$query = $this->db->query("UPDATE `PARAM` SET
`PARAMNAME` = '$name', `INPUTTYPE`=$inputtype,
`COLSIZE` = '$colSize', `DEFAULTVAL` = '$default',
`PARAMDESC` = '$description', `PARAMORDER` = $order
WHERE `PARAMCODE` = $paramcode;");

if (!$query) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
$query2 = $this->db->query("DELETE FROM
`PARAMCHOICE` WHERE `PARAMCODE`=$paramcode;");

if (!$query2) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
$query3 = " ";
if($options!=""){
$token = strtok($options, '&');
while ($token) {
$query3 = $this->db->query("INSERT INTO `PARAMCHOICE`
(`PARAMCODE`, `CHOICENAME`) VALUES
('$paramcode', '$token');");
}
}
}
}
}

```

```

$token = strtok('&');
}
}
if(!$query3){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
return "<font color=#1c9c00" size=2">Parameter
Edited!</font>";
}
}
}

public function edit_station($stationCode,$name,$desc){
$query = $this->db->query("UPDATE `STATION` SET
`STATIONNAME` = '$name', `STATIONDESC` = '$desc'
WHERE `STATIONCODE` = $stationCode;");

if(!$query){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
return "<font color=#1c9c00" size=2">Station
Edited!</font>";
}
}

public function
edit_company($companyCode,$name,$email,$phone,$add,$desc){
$query = $this->db->query("UPDATE `COMPANY` SET
`COMPANYNAME` = '$name', `EMAIL`=$email ,
`PHONE`=$phone', `ADDRESS`=$add', `COMPANYDESC` =
'$desc' WHERE `COMPANYCODE` = $companyCode;");

if(!$query){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
return "<font color=#1c9c00" size=2">Company
Edited!</font>";
}
}

public function edit_dataType($dataTypeCode,$name){
$query = $this->db->query("UPDATE `DATATYPE` SET
`DATATYPENAME` = '$name' WHERE `DATATYPECODE` =
$dataTypeCode;");

if(!$query){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color=#ff0000">$errnum = $errmsg</font>";
} else {
return "<font color=#1c9c00" size=2">Data Type
Edited!</font>";
}
}

public function
edit_profile($username,$password,$firstname,$lastname,$email,$p
hone,$address,$theme){
$userid = $this->session->userdata('userid');

$query = $this->db->query("SELECT * FROM
`USERACCOUNT` WHERE `USERID` = $userid;");

```

```

$query1 = $query->row_array();
$pwd = $query1['PASSWORD'];

$queryString = "";
if($pwd==$password){
$queryString = "UPDATE `USERACCOUNT` SET
`USERNAME` = '$username', `FIRSTNAME`=' $firstname',
`LASTNAME`=' $lastname', `EMAIL`=' $email',
`PHONE`=' $phone', `ADDRESS`=' $address', `THEME`=' $theme'
WHERE `USERID` = $userid;";
}else{
$queryString = "UPDATE `USERACCOUNT` SET
`USERNAME` = '$username', `PASSWORD` =
''.md5($password).'', `FIRSTNAME`=' $firstname',
`LASTNAME`=' $lastname', `EMAIL`=' $email',
`PHONE`=' $phone', `ADDRESS`=' $address', `THEME`=' $theme'
WHERE `USERID` = $userid;";
}

$query = $this->db->query($queryString);

if(!$query){
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();
return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Profile Saved.<br/>Changes will be seen after next log</font>";
}

public function
editCreatePermission($typeCode,$stationCode,$value) {
$query = $this->db->query("UPDATE `CREATEPERMISSION`
SET `PERTYPE` = '$value' WHERE `TYPECODE` = $typeCode
AND `STATIONCODE` = $stationCode;");

if(!$query){
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();
return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Create Permission
Saved</font>";
}

public function editDLock($barCode,$usr,$value) {
$query = $this->db->query("UPDATE `DOCLOCK` SET
`STATUS` = '$value' WHERE `USERID` = $usr AND
`BARCODE` = $barCode;");

if(!$query){
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();
return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {
if ( $value == "y" || $value == 'y' ) {
return "<font color='\"#1c9c00\"' size='\"2\"'>Document $barCode
Locked</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Document $barCode
Unlocked</font>";
}
}
}

//add

```

```

public function
add_user($username,$usertype,$password,$stationcode,$firstname
,$lastname,$email,$phone,$address){
$querystring = "INSERT INTO `USERACCOUNT`
(`USERNAME`,`USERTYPE`,`PASSWORD`,`STATIONCODE`,`
`FIRSTNAME`,`LASTNAME`,`EMAIL`,`PHONE`,`ADDRESS`)
VALUES
('$username','$usertype',''.md5($password).'', $stationcode,' $firstn
ame', $lastname, $email, $phone, $address);";
$query = $this->db->query($querystring);
if (!$query) {
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>User Added</font>";
}

public function add_docuTypeMetaData ($name,$description){
$query = $this->db->query("INSERT INTO `DOCUTYPE`
(`TYPENAME`,`TYPEDESC`) VALUES
('$name','$description');");

if (!$query) {
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();

return "fail";
} else {
$query = $this->db->query("SELECT MAX(`TYPECODE`) AS
MAXNUM FROM `DOCUTYPE`;");
$query1 = $query->row_array();
$typeCode = $query1['MAXNUM'];

return $typeCode;
}

public function save_form ($queryDataContainer){
foreach ($queryDataContainer as $key => $value) {
$query = $this->db->query("UPDATE `METAPARAM` SET
STRINGVAL = '$value' WHERE METAPARAMCODE =
'$key';");
}

if (!$query) {
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Document
Saved</font>";
}

public function
add_permission($stationCode,$paramCode,$perType){
$query = $this->db->query("INSERT INTO `PERMISSION`
(`PARAMCODE`,`STATIONCODE`,`PERTYPE`) VALUES
('$paramCode','$stationCode','$perType');");

if (!$query) {
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errornum = $errmsg</font>";
} else {

```

```

return "<font color=#1c9c00 size=2>Permission
Added</font>";
}
}

public function
add_parameter($name,$options="", $default="", $description="", $typeCode,$order=1,$colSize=100,$inputtype="text"){

$query = $this->db->query("SELECT MAX(`PARAMORDER`)
AS MAXNUM FROM `PARAM` WHERE
`TYPECODE` = $typeCode;");
$query1 = $query->row_array();
$orderMax = $query1['MAXNUM'];

if($order==0){
$order=1;
}

if($orderMax!=null){
if($order=="||$order==0||$order==null){
$order=$orderMax+1;
}
if($orderMax<$order){
$order = $orderMax+1;
}
$query = $this->db->query("UPDATE `PARAM` SET
`PARAMORDER` = `PARAMORDER` + 1 WHERE
`TYPECODE` = $typeCode AND `PARAMORDER` >= $order;");
}

$query0 = $this->db->query("INSERT INTO `PARAM`
(`PARAMORDER`,`TYPECODE`,`INPUTTYPE`,`PARAMNAME`,`COLSIZE`,`DEFAULTVAL`,`PARAMDESC`) VALUES
($order,$typeCode,$inputtype,$name,$colSize,$default,$description);");
if (!$query0) {
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();

return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
$query = $this->db->query("SELECT MAX(`PARAMCODE`) AS
MAXNUM FROM `PARAM`;");
$query1 = $query->row_array();
$max = $query1['MAXNUM'];
$query2 = " ";
$token = strtok($options, '&');
while ($token) {
$query2 = $this->db->query("INSERT INTO `PARAMCHOICE`
(`PARAMCODE`,`CHOICENAME`) VALUES
($max,$token);");
$token = strtok('&');
}

if (!$query2){
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();
return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
return "<font color=#1c9c00 size=2>Parameter
Added</font>";
}
}

public function add_station($name,$desc){

```

```

$query = $this->db->query("INSERT INTO `STATION`
(`STATIONNAME`,`STATIONDESC`) VALUES
($name,$desc);");

if (!$query) {
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();

return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
return "<font color=#1c9c00 size=2>Station
Added</font>";
}

public function
add_company($companyCode,$name,$email,$phone,$add,$desc){
$query = $this->db->query("INSERT INTO `COMPANY`
(`COMPANYNAME`,`EMAIL`,`PHONE`,`ADDRESS`,`COMPANYDESC`) VALUES ($name,$email,$phone,$add,$desc);");

if (!$query) {
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();

return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
return "<font color=#1c9c00 size=2>Company
Added</font>";
}

public function add_dataType($name){
$query = $this->db->query("INSERT INTO `DATATYPE`
(`DATATYPENAME`) VALUES ($name);");

if (!$query) {
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();

return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
return "<font color=#1c9c00 size=2>Data Type
Added</font>";
}

public function
saveCreatePermission($typeCode,$stationCode,$value) {
$query = $this->db->query("INSERT INTO
`CREATEPERMISSION`
(`TYPECODE`,`STATIONCODE`,`PERTYPE`) VALUES
($typeCode,$stationCode,$value);");

if (!$query){
$errorNum = $this->db->_error_number();
$errorMsg = $this->db->_error_message();
return "<font color=#ff0000>$errorNum = $errorMsg</font>";
} else {
return "<font color=#1c9c00 size=2>Create Permission
Saved</font>";
}

public function saveDLock($barCode,$usr,$value) {
$query0 = $this->db->query("SELECT * FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode;");

if (count($query0->result_array())>0){

```

```

$query1 = $this->db->query("DELETE FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode;");
}

$query = $this->db->query("INSERT INTO `DOCLOCK`
(`USERID`,`BARCODE`,`STATUS`) VALUES
('$usr','$barCode','$value');");

if(!$query){
    $errnum = $this->db->_error_number();
    $errmsg = $this->db->_error_message();
    return "<font color=#ff0000>$errnum = $errmsg</font>";
} else {
    return "<font color=#1c9c00" size=2>Doc $barCode is
locked to you.</font>";
}

public function assignApp($userId,$companies) {

    $deleteCompaniesQuery = $this->db->query("DELETE FROM
`COMPANYUSER` WHERE COMPANYUSER.USERID =
'$userId';");
    //companies
    foreach ( $companies as $company ) {
        $addCompaniesQuery = $this->db->query("INSERT INTO
`COMPANYUSER` (`USERID`,`COMPANYCODE`) VALUES
('$userId','$company');");
    }

    if(!$addCompaniesQuery||!$deleteCompaniesQuery){
        return "failed";
    } else {
        return "<font color=#1c9c00" size=2>Assigned</font>";
    }
}

public function add_document
($title,$description,$typeCode,$userId,$companies) {
    $addToDocumetaQuery = $this->db->query("INSERT INTO
`DOCUCOREMETADATA`
(`TITLE`,`TYPECODE`,`USERID`,`DOCDESCR`) VALUES
('$title','$typeCode','$userId','$description');");

    $barCodeQ = $this->db->query("SELECT MAX(BARCODE) as
maxbar FROM DOCUCOREMETADATA;");
    $barCodeArr = $barCodeQ->row_array();
    $barCode = $barCodeArr['maxbar'];

    //companies
    foreach ( $companies as $company ) {
        $addCompaniesQuery = $this->db->query("INSERT INTO
`COMPANYDOC` (`BARCODE`,`COMPANYCODE`)
VALUES ('$barCode','$company');");
    }
    //stations
    $docStationQuery = $this->db->query("SELECT * FROM
DOCUPATHSTATION WHERE
DOCUPATHSTATION.TYPECODE = $typeCode;");
    foreach ( $docStationQuery->result_array() as $docStation ) {
        $addStationsQuery = $this->db->query("INSERT INTO
`DOCUMETAPATHSTATION`
(`BARCODE`,`STATIONCODE`,`METALEFTPOS`,
`METATOPPOS`) VALUES
('$barCode','.$docStation['STATIONCODE'].','.$docStation['LE
FTPOS'].','.$docStation['TOPPOS'].');");
    }
    //paths
    $docPathQuery = $this->db->query("SELECT * FROM
DOCUPATHCON WHERE DOCUPATHCON.TYPECODE =
$typeCode;");
    foreach ( $docPathQuery->result_array() as $docPath ) {
        $addPathQuery = $this->db->query("INSERT INTO
`DOCUMETAPATHCON`
(`BARCODE`,`METACONSOURCE`,`METACONTARGET`,
`METACONDESCR`) VALUES
('$barCode','.$docPath['CONSOURCE'].','.$docPath['CONTAR
GET'].','.$docPath['CONDESCR'].');");
    }
    //params
    $docParamQuery = $this->db->query("SELECT * FROM
PARAM WHERE PARAM.TYPECODE = $typeCode;");
    foreach ( $docParamQuery->result_array() as $docParam ) {
        $addParamsQuery = $this->db->query("INSERT INTO
`METAPARAM` (`BARCODE`,`INPUTTYPE`,`DATATYPE`,
`PARAMNAME`,`PARAMORDER`,`STRINGVAL`,`COLSIZE`,
`PARAMDESC`) VALUES
('$barCode','.$docParam['INPUTTYPE'].','.$docParam['DATAT
YPE'].','.$docParam['PARAMNAME'].','.$docParam['PARAM
ORDER'].','.$docParam['DEFAULTVAL'].','.$docParam['COL
SIZE'].','.$docParam['PARAMDESC'].');");
    }
    //choice of params
    $paramQ = $this->db->query("SELECT
MAX(METAPARAMCODE) as maxpar FROM
METAPARAM;");
    $paramArr = $paramQ->row_array();
    $param = $paramArr['maxpar'];

    $docChoicesQuery = $this->db->query("SELECT * FROM
PARAMCHOICE WHERE PARAMCHOICE.PARAMCODE =
'.$docParam['PARAMCODE'].');");

    foreach ( $docChoicesQuery->result_array() as $docChoices ) {
        $addChoicesQuery = $this->db->query("INSERT INTO
`METAPARAMCHOICE`
(`METAPARAMCODE`,`METACHOICENAME`) VALUES
('$param','.$docChoices['CHOICENAME'].');");
    }

    //permissions
    $docPermissionQuery = $this->db->query("SELECT * FROM
PERMISSION WHERE PERMISSION.PARAMCODE =
'.$docParam['PARAMCODE'].');");
    foreach ( $docPermissionQuery->result_array() as $docPermission ) {
        $addPermissionQuery = $this->db->query("INSERT INTO
`METAPERMISSION`
(`METAPARAMCODE`,`STATIONCODE`,`PERTYPE`)
VALUES
('$param','.$docPermission['STATIONCODE'].','.$docPermissi
on['PERTYPE'].');");
    }
    //place the document to start position
    //set docloc
    $docLocQuery = $this->db->query("SELECT * FROM
DOCUMETAPATHCON WHERE
DOCUMETAPATHCON.BARCODE = $barCode AND
DOCUMETAPATHCON.METACONSOURCE = '-1';");
    $docLocArr = $docLocQuery->row_array();
    $docLoc = $docLocArr['METACONTARGET'];
    $date = date('Y-m-d H:i:s');
    $dateZero = date();
    $addLocQuery = $this->db->query("INSERT INTO `DOCULOC`
(`BARCODE`,`STATIONCODE`,`STEPORDER`,`TIMEIN`,`TI
MEOUT`) VALUES
('$barCode','$docLoc','1','$date','$dateZero');");
}

```

```

if (!$addToDocumetaQuery || !$addCompaniesQuery || !$addStations
Query || !$addPathQuery || !$addParamsQuery || !$addChoicesQuery) {
return "failed";
} else {
return "<font color=#1c9c00 size=2>New Document
Saved</font>";
}
}

```

```

public function
transfer_form($barCode,$stationCode,$stationCode2) {

```

```

$query00 = $this->db->query("SELECT * FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode;");

```

```

if (count($query00->result_array())>0){
$query01 = $this->db->query("DELETE FROM `DOCLOCK`
WHERE DOCLOCK.BARCODE = $barCode;");
}

```

```

$query0 = $this->db->query("SELECT MAX(STEPORDER) as
maxstep FROM DOCULOC WHERE DOCULOC.BARCODE =
'$barCode';");
$steporderarr = $query0->row_array();
$steporder = $steporderarr['maxstep'];
$date = date('Y-m-d H:i:s');
$query1 = $this->db->query("UPDATE `DOCULOC` SET
DOCULOC.TIMEOUT = '$date' WHERE
DOCULOC.STATIONCODE = '$stationCode2' AND
DOCULOC.BARCODE = '$barCode' AND
DOCULOC.STEPORDER = '$steporder';");
$steporder = $steporder+1;
$dateZero = date();
$query2 = $this->db->query("INSERT INTO `DOCULOC`
(` BARCODE`, `STATIONCODE`, `STEPORDER`, `TIMEIN`, `TI
MEOUT`) VALUES
('$barCode','$stationCode','$steporder','$date','$dateZero');");

```

```

if (!$query0 || !$query1 || !$query2) {
$errornum = $this->db->_error_number();
$errormsg = $this->db->_error_message();
return "<font color=#ff0000>$errornum = $errmsg</font>";
} else {
return "<font color=#1c9c00 size=2>Sent</font>";
}
}

```

```

public function archiveDoc($barCode,$stationCode) {

```

```

$queryA = $this->db->query("SELECT MAX(STEPORDER) as
maxstep FROM DOCULOC WHERE DOCULOC.BARCODE =
'$barCode';");
$steporderarr = $queryA->row_array();
$steporder = $steporderarr['maxstep'];
$date = date('Y-m-d H:i:s');
$update0 = $this->db->query("UPDATE `DOCULOC` SET
DOCULOC.TIMEOUT = '$date' WHERE
DOCULOC.STATIONCODE = '$stationCode' AND
DOCULOC.BARCODE = '$barCode' AND
DOCULOC.STEPORDER = '$steporder';");

```

```

$query0 = $this->db->query("SELECT * FROM
DOCUCOREMETADATA, DOCUTYPE WHERE
DOCUCOREMETADATA.TYPECODE =
DOCUTYPE.TYPECODE AND
DOCUCOREMETADATA.BARCODE = $barCode;");
$query1 = $this->db->query("SELECT * FROM
COMPANYDOCU, COMPANY WHERE

```

```

COMPANY.COMPANYCODE =
COMPANYDOCU.COMPANYCODE AND
COMPANYDOCU.BARCODE = $barCode;");
$query2 = $this->db->query("SELECT * FROM
DOCUMETAPATHSTATION, STATION WHERE
STATION.STATIONCODE =
DOCUMETAPATHSTATION.STATIONCODE AND
DOCUMETAPATHSTATION.BARCODE = $barCode;");
$query3 = $this->db->query("SELECT * FROM
DOCUMETAPATHCON WHERE
DOCUMETAPATHCON.BARCODE = $barCode;");
$query4 = $this->db->query("SELECT * FROM METAPARAM
WHERE METAPARAM.BARCODE = $barCode;");
$query5 = $this->db->query("SELECT * FROM DOCULOC,
STATION WHERE STATION.STATIONCODE =
DOCULOC.STATIONCODE AND DOCULOC.BARCODE =
$barCode;");

```

```

$docuMeta = $query0->row_array();
$companyDoc = $query1->result_array();
$station = $query2->result_array();
$conn = $query3->result_array();
$param = $query4->result_array();
$doculoc = $query5->result_array();

```

```

$insert0 = $this->db->query("INSERT INTO
`DOCUCOREMETADATAARC`
(` ARCCODE`, `USERID`, `TITLE`, `TYPENAME`, `DOCDESCR`)
VALUES
('".$docuMeta['BARCODE']. "','".$docuMeta['USERID']. "','".$docu
Meta['TITLE']. "','".$docuMeta['TYPENAME']. "','".$docuMeta['D
OCDESCR']. "','");
foreach ( $station as $station_elmt ) {
$insert1 = $this->db->query("INSERT INTO
`DOCUMETAPATHSTATIONARC`
(` ARCCODE`, `STATIONARCCODE`, `STATIONNAME`, `MET
ALEFTPOS`, `METATOPPOS`) VALUES
('".$station_elmt['BARCODE']. "','".$station_elmt['STATIONCOD
E']. "','".$station_elmt['STATIONNAME']. "','".$station_elmt['MET
ALEFTPOS']. "','".$station_elmt['METATOPPOS']. "','");
}
foreach ( $conn as $conn_elmt ) {
$insert2 = $this->db->query("INSERT INTO
`DOCUMETAPATHCONARC`
(` ARCCODE`, `METACONSOURCE`, `METACONTARGET`, `M
ETACONDESCR`) VALUES
('".$conn_elmt['BARCODE']. "','".$conn_elmt['METACONSOUR
CE']. "','".$conn_elmt['METACONTARGET']. "','".$conn_elmt['M
ETACONDESCR']. "','");
}
foreach ( $companyDoc as $companyDoc_elmt ) {
$insert3 = $this->db->query("INSERT INTO
`COMPANYDOCUARC` (` ARCCODE`, `COMPANYNAME`)
VALUES
('$barCode', '".$companyDoc_elmt['COMPANYNAME']. "','");
}
foreach ( $param as $param_elmt ) {
$insert4 = $this->db->query("INSERT INTO
`METAPARAMARC`
(` METAPARAMARCCODE`, `ARCCODE`, `INPUTTYPE`, `DAT
ATYPE`, `PARAMNAME`, `PARAMORDER`, `COLSIZE`, `STRI
NGVAL`, `PARAMDESC`) VALUES
('".$param_elmt['METAPARAMCODE']. "','".$param_elmt['BAR
CODE']. "','".$param_elmt['INPUTTYPE']. "','".$param_elmt['DAT
ATYPE']. "','".$param_elmt['PARAMNAME']. "','".$param_elmt['P
ARAMORDER']. "','".$param_elmt['COLSIZE']. "','".$param_elmt['
STRINGVAL']. "','".$param_elmt['PARAMDESC']. "','");
$query6 = $this->db->query("SELECT * FROM
METAPARAMCHOICE WHERE

```



```

METAPARAMCHOICE.METAPARAMCODE =
$.param_elmt['METAPARAMCODE']."";");
$choices = $query6->result_array();
foreach ( $choices as $choices_elmt ) {
$insert5 = $this->db->query("INSERT INTO
`METAPARAMCHOICEARC`
(`METACHOICECODEARC`,`METAPARAMARCCODE`,`ME
TACHOICENAME`) VALUES
('".$choices_elmt['METACHOICECODE']. "','".$choices_elmt['ME
TAPARAMCODE']. "','".$choices_elmt['METACHOICENAME'].
'");");
}

```

```

$delete1 = $this->db->query("DELETE FROM
`METAPERMISSION` WHERE
METAPERMISSION.METAPARAMCODE='".$param_elmt['ME
TAPARAMCODE']. "','");
$delete2 = $this->db->query("DELETE FROM
`METAPARAMCHOICE` WHERE
METAPARAMCHOICE.METAPARAMCODE='".$param_elmt['
METAPARAMCODE']. "','");
}
foreach ( $doculoc as $doculoc_elmt ) {
$insert6 = $this->db->query("INSERT INTO `DOCULOCAARC`
(` ARCCODE`,`STATIONNAME`,`STEPORDER`,`TIMEIN`,`TI
MEOUT`) VALUES
('".$doculoc_elmt['BARCODE']. "','".$doculoc_elmt['STATIONN
AME']. "','".$doculoc_elmt['STEPORDER']. "','".$doculoc_elmt['TI
MEIN']. "','".$doculoc_elmt['TIMEOUT']. "','");");
}

```

```

$delete0 = $this->db->query("DELETE FROM `DOCULOC`
WHERE DOCULOC.BARCODE=' $barCode;");
$delete3 = $this->db->query("DELETE FROM `METAPARAM`
WHERE METAPARAM.BARCODE=' $barCode;");
$delete4 = $this->db->query("DELETE FROM
`DOCUMENTAPATHCON` WHERE
DOCUMENTAPATHCON.BARCODE=' $barCode;");
$delete5 = $this->db->query("DELETE FROM
`DOCUMENTAPATHSTATION` WHERE
DOCUMENTAPATHSTATION.BARCODE=' $barCode;");
$delete6 = $this->db->query("DELETE FROM
`COMPANYDOCU` WHERE
COMPANYDOCU.BARCODE=' $barCode;");
$delete7 = $this->db->query("DELETE FROM
`DOCUCOREMETADATA` WHERE
DOCUCOREMETADATA.BARCODE=' $barCode;");

```

```

if(!$query0||$query1||$query2||$query3||$query4||$query5||$qu
ery6||$insert0||$insert1||$insert2||$insert3||$insert4||$insert5||$
insert6||$delete0||$delete1||$delete2||$delete3||$delete4||$delete
5||$delete6||$delete7){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color=#ff0000>$errnum = $errmsg</font>";
} else {
return "<font color=#1c9c00" size=2>Finished and
Archived</font>";
}
}

```

```
//delete
```

```
public function deleteDoc($barCode,$stationCode) {
```

```

$queryA = $this->db->query("SELECT MAX(STEPORDER) as
maxstep FROM DOCULOC WHERE DOCULOC.BARCODE =
'$barCode;");

```

```

$steporderarr = $queryA->row_array();
$steporder = $steporderarr['maxstep'];
$date = date('Y-m-d H:i:s');
$update0 = $this->db->query("UPDATE `DOCULOC` SET
DOCULOC.TIMEOUT = '$date' WHERE
DOCULOC.STATIONCODE = '$stationCode' AND
DOCULOC.BARCODE = '$barCode' AND
DOCULOC.STEPORDER = '$steporder;");

```

```

$query0 = $this->db->query("SELECT * FROM
DOCUCOREMETADATA, DOCUTYPE WHERE
DOCUCOREMETADATA.TYPECODE =
DOCUTYPE.TYPECODE AND
DOCUCOREMETADATA.BARCODE = $barCode;");
$query1 = $this->db->query("SELECT * FROM
COMPANYDOCU, COMPANY WHERE
COMPANY.COMPANYCODE =
COMPANYDOCU.COMPANYCODE AND
COMPANYDOCU.BARCODE = $barCode;");
$query2 = $this->db->query("SELECT * FROM
DOCUMENTAPATHSTATION, STATION WHERE
STATION.STATIONCODE =
DOCUMENTAPATHSTATION.STATIONCODE AND
DOCUMENTAPATHSTATION.BARCODE = $barCode;");
$query3 = $this->db->query("SELECT * FROM
DOCUMENTAPATHCON WHERE
DOCUMENTAPATHCON.BARCODE = $barCode;");
$query4 = $this->db->query("SELECT * FROM METAPARAM
WHERE METAPARAM.BARCODE = $barCode;");
$query5 = $this->db->query("SELECT * FROM DOCULOC,
STATION WHERE STATION.STATIONCODE =
DOCULOC.STATIONCODE AND DOCULOC.BARCODE =
$barCode;");

```

```

$docuMeta = $query0->row_array();
$companyDoc = $query1->result_array();
$station = $query2->result_array();
$conn = $query3->result_array();
$param = $query4->result_array();
$doculoc = $query5->result_array();

```

```

$insert0 = $this->db->query("INSERT INTO
`DOCUCOREMETADATAARC`
(` ARCCODE`,`USERID`,`TITLE`,`TYPENAME`,`DOCDESCR`)
VALUES
('".$docuMeta['BARCODE']. "','".$docuMeta['USERID']. "','".$docu
Meta['TITLE']. "','".$docuMeta['TYPENAME']. "','".$docuMeta['D
OCDESCR']. "','");");
foreach ( $station as $station_elmt ) {
$insert1 = $this->db->query("INSERT INTO
`DOCUMENTAPATHSTATIONARC`
(` ARCCODE`,`STATIONARCCODE`,`STATIONNAME`,`MET
ALEFTPOS`,`METATOPPOS`) VALUES
('".$station_elmt['BARCODE']. "','".$station_elmt['STATIONCOD
E']. "','".$station_elmt['STATIONNAME']. "','".$station_elmt['MET
ALEFTPOS']. "','".$station_elmt['METATOPPOS']. "','");");
}
foreach ( $conn as $conn_elmt ) {
$insert2 = $this->db->query("INSERT INTO
`DOCUMENTAPATHCONARC`
(` ARCCODE`,`METACONSOURCE`,`METACONTARGET`,`M
ETACONDESCR`) VALUES
('".$conn_elmt['BARCODE']. "','".$conn_elmt['METACONSOUR
CE']. "','".$conn_elmt['METACONTARGET']. "','".$conn_elmt['M
ETACONDESCR']. "','");");
}
foreach ( $companyDoc as $companyDoc_elmt ) {
$insert3 = $this->db->query("INSERT INTO
`COMPANYDOCUARC` (` ARCCODE`,`COMPANYNAME`)

```

```

VALUES
($barCode,'"$.companyDoc_elmt['COMPANYNAME'].");");
}
foreach ($param as $param_elmt) {
$insert4 = $this->db->query("INSERT INTO
`METAPARAMARC`
(`METAPARAMARCCODE`,`ARCCODE`,`INPUTTYPE`,`DAT
ATYPE`,`PARAMNAME`,`PARAMORDER`,`COLSIZE`,`STRI
NGVAL`,`PARAMDESC`) VALUES
('".$param_elmt['METAPARAMARCCODE'].",".$param_elmt['BAR
CODE'].",".$param_elmt['INPUTTYPE'].",".$param_elmt['DAT
ATYPE'].",".$param_elmt['PARAMNAME'].",".$param_elmt['P
ARAMORDER'].",".$param_elmt['COLSIZE'].",".$param_elmt['P
ARAMORDER'].",".$param_elmt['COLSIZE'].",".$param_elmt['P
ARAMORDER'].",".$param_elmt['PARAMDESC'].");");
$query6 = $this->db->query("SELECT * FROM
METAPARAMCHOICE WHERE
METAPARAMCHOICE.METAPARAMARCCODE =
'".$param_elmt['METAPARAMARCCODE'].";");
$choices = $query6->result_array();
foreach ($choices as $choices_elmt) {
$insert5 = $this->db->query("INSERT INTO
`METAPARAMCHOICEARC`
(`METACHOICECCODEARC`,`METAPARAMARCCODE`,`ME
TACHOICENAME`) VALUES
('".$choices_elmt['METACHOICECCODE'].",".$choices_elmt['ME
TAPARAMARCCODE'].",".$choices_elmt['METACHOICENAME'].
");");
}

$delete1 = $this->db->query("DELETE FROM
`METAPERMISSION` WHERE
METAPERMISSION.METAPARAMARCCODE='".$param_elmt['ME
TAPARAMARCCODE'].";");
$delete2 = $this->db->query("DELETE FROM
`METAPARAMCHOICE` WHERE
METAPARAMCHOICE.METAPARAMARCCODE='".$param_elmt['
METAPARAMARCCODE'].";");
}
foreach ($doculoc as $doculoc_elmt) {
$insert6 = $this->db->query("INSERT INTO `DOCULOCAARC`
(`ARCCODE`,`STATIONNAME`,`STEPORDER`,`TIMEIN`,`TI
MEOUT`) VALUES
('".$doculoc_elmt['BARCODE'].",".$doculoc_elmt['STATIONN
AME'].",".$doculoc_elmt['STEPORDER'].",".$doculoc_elmt['TI
MEIN'].",".$doculoc_elmt['TIMEOUT'].");");
}

$delete0 = $this->db->query("DELETE FROM `DOCULOC`
WHERE DOCULOC.BARCODE='".$barCode.'");
$delete3 = $this->db->query("DELETE FROM `METAPARAM`
WHERE METAPARAM.BARCODE='".$barCode.'");
$delete4 = $this->db->query("DELETE FROM
`DOCUMETAPATHCON` WHERE
DOCUMETAPATHCON.BARCODE='".$barCode.'");
$delete5 = $this->db->query("DELETE FROM
`DOCUMETAPATHSTATION` WHERE
DOCUMETAPATHSTATION.BARCODE='".$barCode.'");
$delete6 = $this->db->query("DELETE FROM
`COMPANYDOCU` WHERE
COMPANYDOCU.BARCODE='".$barCode.'");
$delete7 = $this->db->query("DELETE FROM
`DOCUCOREMETADATA` WHERE
DOCUCOREMETADATA.BARCODE='".$barCode.'");

if(!$query0||$query1||$query2||$query3||$query4||$query5||$qu
ery6||$insert0||$insert1||$insert2||$insert3||$insert4||$insert5||$
insert6||$delete0||$delete1||$delete2||$delete3||$delete4||$delete
5||$delete6||$delete7){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

```

```

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Deleted and
Archived</font>";
}
}

```

```

public function delete_user($userid){
$query = $this->db->query("DELETE FROM `USERACCOUNT`
WHERE `USERID`='".$userid.'");

if (!$query) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

```

```

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>User Deleted</font>";
}
}

```

```

public function delete_parameter($paramcode,$typeCode,$order){

$query = $this->db->query("DELETE FROM `PARAM` WHERE
`PARAMCODE`='".$paramcode.'");
$query = $this->db->query("UPDATE `PARAM` SET
`PARAMORDER`=`PARAMORDER`-1 WHERE
`TYPECODE`='".$typeCode.' AND `PARAMORDER` > $order;");

```

```

if (!$query) {
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();

return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {

```

```

$query = $this->db->query("DELETE FROM `PARAMCHOICE`
WHERE `PARAMCODE`='".$paramcode.'");
if (!$query){
$errnum = $this->db->_error_number();
$errmsg = $this->db->_error_message();
return "<font color='\"#ff0000\"'>$errnum = $errmsg</font>";
} else {
return "<font color='\"#1c9c00\"' size='\"2\"'>Parameter
Deleted</font>";
}
}
}

```

```

public function delete_station($stationCode){

$query = $this->db->query("SELECT * FROM
`DOCUPATHSTATION`,`DOCUMETAPATHSTATION`,`
DOCULOC` WHERE
DOCUPATHSTATION.STATIONCODE='".$stationCode.' OR
DOCUMETAPATHSTATION.STATIONCODE='".$stationCode.'
OR DOCULOC.STATIONCODE='".$stationCode.'");

$count = count ($query->result_array());

if($count>0){
return "<font color='\"#ff0000\"'>Please make sure this station is no
longer used</font>";
}
}

```

```

$query = $this->db->query("DELETE FROM `STATION`
WHERE `STATIONCODE`='".$stationCode.'");

```

```

if (!$query) {
    $errnum = $this->db->_error_number();
    $errmsg = $this->db->_error_message();

    return "<font color=#ff0000>$errnum = $errmsg</font>";
} else {
    return "<font color=#1c9c00 size=2>Station Deleted</font>";
}

public function delete_company($companyCode){
    $query = $this->db->query("SELECT * FROM `COMPANYDOCU`,`DOCUCOREMETADATA` WHERE DOCUCOREMETADATA.BARCODE = COMPANYDOCU.BARCODE AND COMPANYDOCU.COMPANYCODE=$companyCode");
    $docComp = $query->result_array();
    $count = count($docComp);

    if($count>0){
        $retStr = "<font color=#ff0000 size=2>Company still has unfinished documents: <br />";
        foreach ($docComp as $docCompentry) {
            $retStr .= $docCompentry['TITLE']."<br />";
        }
        $retStr .= "</font>";
        return $retStr;
    }

    $query = $this->db->query("DELETE FROM `COMPANY` WHERE `COMPANYCODE`=$companyCode");
    $query = $this->db->query("DELETE FROM `COMPANYUSER` WHERE COMPANYUSER.COMPANYCODE = '$companyCode'");

    if (!$query) {
        $errnum = $this->db->_error_number();
        $errmsg = $this->db->_error_message();

        return "<font color=#ff0000>$errnum = $errmsg</font>";
    } else {
        return "<font color=#1c9c00 size=2>Company Deleted</font>";
    }
}

public function delete_dataType($dataTypeCode){

    $query = $this->db->query("DELETE FROM `DATATYPE` WHERE `DATATYPECODE`=$dataTypeCode");

    if (!$query) {
        $errnum = $this->db->_error_number();
        $errmsg = $this->db->_error_message();

        return "<font color=#ff0000>$errnum = $errmsg</font>";
    } else {
        return "<font color=#1c9c00 size=2>Data type Deleted</font>";
    }
}

public function delete_type($typeCode){

    $query = $this->db->query("SELECT * FROM `PARAM` WHERE `TYPECODE` = '$typeCode'");

```

```

    $temp = $query->result_array();

    foreach ($temp as $temp_item):
        $query = $this->db->query("DELETE FROM `PARAMCHOICE` WHERE `PARAMCODE`='.$temp_item['PARAMCODE']. '");
    endforeach;

    if (!$query) {
        $errnum = $this->db->_error_number();
        $errmsg = $this->db->_error_message();

        return "<font color=#ff0000>$errnum = $errmsg</font>";
    } else {
        $query = $this->db->query("DELETE FROM DOCUPATHSTATION WHERE TYPECODE = $typeCode");
        $query = $this->db->query("DELETE FROM DOCUPATHCON WHERE TYPECODE = $typeCode");

        if (!$query) {
            $errnum = $this->db->_error_number();
            $errmsg = $this->db->_error_message();

            return "<font color=#ff0000>$errnum = $errmsg</font>";
        } else {
            $query = $this->db->query("DELETE FROM `PARAM` WHERE `TYPECODE`=$typeCode");

            if (!$query){
                $errnum = $this->db->_error_number();
                $errmsg = $this->db->_error_message();
                return "<font color=#ff0000>$errnum = $errmsg</font>";
            } else {
                $query = $this->db->query("DELETE FROM `DOCUTYPE` WHERE `TYPECODE`=$typeCode");
                if (!$query){
                    $errnum = $this->db->_error_number();
                    $errmsg = $this->db->_error_message();
                    return "<font color=#ff0000>$errnum = $errmsg</font>";
                } else {
                    return "<font color=#1c9c00 size=2>Parameter Deleted</font>";
                }
            }
        }
    }

    public function delete_permission($paramCode,$stationCode){

        $query = $this->db->query("DELETE FROM `PERMISSION` WHERE `PARAMCODE`=$paramCode AND `STATIONCODE`=$stationCode");

        if (!$query) {
            $errnum = $this->db->_error_number();
            $errmsg = $this->db->_error_message();

            return "<font color=#ff0000>$errnum = $errmsg</font>";
        } else {
            return "<font color=#1c9c00 size=2>Permission Deleted</font>";
        }
    }

    //end for document type
}
?>

```

ii. controllers/Document.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```
class Document extends CI_Controller {

public function __construct(){
parent::__construct();
// $this->load->helper('form');
$this->load->helper('url');
$this->load->model('document_model');
}

public function index(){
$data['title'] = 'Login';
$data['status'] = '';
if($this->session->userdata('login_state') == FALSE){
$this->load->view('panels/header',$data);
$this->load->view('panels/sidebar');
$this->load->view('login',$data);
$this->load->view('panels/footer');
} else {
} else {
$this->home();
}
}

public function home(){
if ( $this->session->userdata('login_state') == FALSE) {
redirect('/');
}
$this->load->view('panels/header');
$this->load->view('panels/sidebar');
if($this->session->userdata('usertype')=="admin"){
$this->load->view('welcome_message');
} else if ($this->session->userdata('usertype')=="head"){
$this->load->view('welcome_message');
} else if ($this->session->userdata('usertype')=="assistant"){
$this->load->view('welcome_message');
} else if ($this->session->userdata('usertype')=="applicant"){
$this->load->view('welcome_message');
} else {
$this->load->view('welcome_message');
}
$this->load->view('panels/footer');
}

public function logout(){
$sesdata = array(
'userid' => "",
'username' => "",
'firstname' => "",
'lastname' => "",
'stationcode' => "",
'usertype' => "",
'email' => "",
'phone' => "",
'address' => "",
'theme' => ""
);
$this->session->unset_userdata($sesdata);
$this->session->set_userdata('login_state', FALSE);

$data['title'] = 'Login';
$data['status'] = '';
```

```
$this->load->view('panels/header',$data);
$this->load->view('panels/sidebar');
$this->load->view('login',$data);
$this->load->view('panels/footer');
}

public function setupworkflow($typeCode) {
if ( $this->session->userdata('login_state') == FALSE || $this->session->userdata('usertype') != "admin") {
redirect('/');
}
$data['documentTypeStation'] = $this->document_model->get_documentTypeStation($typeCode);
$data['documentTypePath'] = $this->document_model->get_documentTypePath($typeCode);
$data['stationsToAdd'] = $this->document_model->get_otherStations($typeCode);
$data['typeCode'] = $typeCode;

$data['title'] = 'Edit Workflow';
$data['status'] = '';

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/setup',$data);
$this->load->view('panels/footer');
}

public function getTypePathInstances($typeCode) {
$page = isset($_POST['page'])? $this->input->post('page'):1; // get the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get how many rows we want to have into the grid
$idx = isset($_POST['idx'])? $this->input->post('idx'):'TITLE'; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])? $this->input->post('sord'):'ASC'; // get the direction

$queryString = "1";

if ( $idx == 'ELAPSED' ) {

$count = count ($this->document_model->get_pathInstancesCount($typeCode));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;

$start = 0;

if ($count > 0) {
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)
}

$typeList = $this->document_model->get_pathInstances($typeCode,"TITLE", "", "0" , $count, "1");

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;
```

```

$i=0;
$j=0;
$dataRows;
$docStatus_holder = array();
$prev = -1;

foreach ($typeList as $typeList_item):

    $connections = $this->document_model-
    >getMetaConnections($typeList_item['BARCODE']);
    $startDate = $this->document_model-
    >getDocLocStartDate($typeList_item['BARCODE']);
    $docStatus = $this-
    >getDocStatus($typeList_item['STATIONCODE'],$connections);
    $dateRnge = $this->getDateDiff($startDate['MINDATE']);

    if ( $prev == -1 ) {
        $prev = $docStatus;
    }
    if ( $prev != $docStatus ) {
        $prev = $docStatus;
        if ( $this->isArray($docStatus_holder,$docStatus) ) {
            $j=$docStatus_holder[$docStatus];
        } else {
            $j=0;
        }
    }

    $dataRows[$dateRnge.".".$j]['id'] = $typeList_item['BARCODE'];
    $dataRows[$dateRnge.".".$j]['cell'] =
    array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
    _item['STATIONNAME'],$dateRnge,$docStatus);
    $i++;
    $j++;
    $docStatus_holder[$docStatus] = $j;
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
    krsort($dataRows);
} else {
    ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
    if ( $i >= $start && $i < $limit*$page ) {
        $data->rows[$j] = $dataRow;
        $j++;
    }
    $i++;
}
} else if ( $sidx == 'PROGRESS' ) {

    $count = count ($this->document_model-
    >get_pathInstancesCount($typeCode));
    if( $count >0 ) {
        $total_pages = ceil($count/$limit);
    } else {
        $total_pages = 0;
    }
    if ( $page > $total_pages ) $page=$total_pages;
    $start = 0;

    if ( $count >0 ) {
        $start = $limit*$page - $limit; // do not put $limit*( $page - 1)
    }

    $typeList = $this->document_model-
    >get_pathInstances($typeCode,"TITLE", "", "0" , $count, "1");
    //$typeList = $this->document_model->get_docs("TITLE", "", "0" ,
    $count, "1",null);

    $data->total = $total_pages;
    $data->page = $page;
    $data->records = $count;

    $i=0;
    $j=0;
    $dataRows;
    $docStatus_holder = array();
    $prev = -1;
    foreach ($typeList as $typeList_item):

        $connections = $this->document_model-
        >getMetaConnections($typeList_item['BARCODE']);
        $startDate = $this->document_model-
        >getDocLocStartDate($typeList_item['BARCODE']);
        $docStatus = $this-
        >getDocStatus($typeList_item['STATIONCODE'],$connections);
        $dateRnge = $this->getDateDiff($startDate['MINDATE']);

        if ( $prev == -1 ) {
            $prev = $docStatus;
        }
        if ( $prev != $docStatus ) {
            $prev = $docStatus;
            if ( $this->isArray($docStatus_holder,$docStatus) ) {
                $j=$docStatus_holder[$docStatus];
            } else {
                $j=0;
            }
        }

        $dataRows[$docStatus.".".$j]['id'] = $typeList_item['BARCODE'];
        $dataRows[$docStatus.".".$j]['cell'] =
        array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
        _item['STATIONNAME'],$dateRnge,$docStatus);
        $i++;
        $j++;
        $docStatus_holder[$docStatus] = $j;
    endforeach;

    if ( $sord == 'ASC' || $sord == 'asc' ) {
        krsort($dataRows);
    } else {
        ksort($dataRows);
    }

    $i=0;
    $j=0;
    foreach ($dataRows as $dataRow){
        if ( $i >= $start && $i < $limit*$page ) {
            if($ _POST[' _search']=='true'){
                $query = json_decode($this->input->post('filters')); // get the
                query

                switch ($this->input->post("searchOper")) {
                    case "eq":
                        if ( $this->input->post("searchString") == $dataRow['cell'][5] ) {
                            $data->rows[$j] = $dataRow;
                        }
                    }
                }
            }
        }
    }
}

```

```

}
break;
case "ne":
if ( $this->input->post("searchString") != $dataRow['cell'][5] ) {
$data->rows[$j] = $dataRow;
}
break;
case "lt":
if ( $dataRow['cell'][5] < $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "le":
if ( $dataRow['cell'][5] <= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "gt":
if ( $dataRow['cell'][5] > $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "ge":
if ( $dataRow['cell'][5] >= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "nu":
if ( !isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
break;
case "nn":
if ( isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
break;
case "cn":
$data->rows[$j] = $dataRow;
break;
} else {
$data->rows[$j] = $dataRow;
}
$j++;
}
} else {

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ( $this->input->post("searchOper") ) {
case "eq":
$operation = " = ". $this->input->post("searchString")."";
break;
case "ne":
$operation = " <> ". $this->input->post("searchString")."";
break;
case "lt":
$operation = " < ". $this->input->post("searchString")."";

```

```

break;
case "le":
$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $this->input->post("searchString")."%";
break;
}

$queryString = "(" . $this->input->
post("searchField").$operation."";
}

if(!$sidx) $sidx = 'TITLE';

$count = count ( $this->document_model-
>get_pathInstancesCount($typeCode));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ( $page > $total_pages ) $page=$total_pages;
$start = 0;

if ( $count > 0 ) {
$start = $limit*$page - $limit; // do not put $limit*( $page - 1
)

$typeList = $this->document_model-
>get_pathInstances($typeCode,$sidx, $sord, $start , $limit,
$queryString);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

foreach ( $typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model-
>getCompanies($typeList_item['BARCODE']));

```

```

$data->rows[$i]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['STATIONNAME'],$dateRnge,$docStatus);
$i++;
endforeach;
}

echo json_encode($data);
}

```

```

public function viewDocType($typeCode) {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype') != "admin" ) {
redirect('/');
}
}

```

```
$max = 100;
```

```

$stationData = $this->document_model-
>get_documentTypeStation($typeCode);

```

```

for ( $i=0 ; $i < count($stationData) ; $i++ ){
$dccnt = $this->document_model-
>get_stationdoccount($stationData[$i]['STATIONCODE']);
$cnt = $dccnt['DOCCOUNT'];
$stationData[$i] =
array_merge($stationData[$i],array('docCount'=>$cnt,
'color'=>$this->rgbToHex($cnt,$max-$cnt,0,$max)));
}

```

```
$data['documentTypeStation'] = $stationData;
```

```

$data['documentTypePath'] = $this->document_model-
>get_documentTypePath($typeCode);
$data['stationsToAdd'] = $this->document_model-
>get_otherStations($typeCode);
//$data['pathInstances'] = $this->document_model-
>get_pathInstances($typeCode);
$data['typeCode'] = $typeCode;
$data['documentTypeMeta'] = $this->document_model-
>get_documentTypeMeta($typeCode);

```

```

$data['title'] = 'View Document Type';
$data['status'] = '';

```

```

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewDocType',$data);
$this->load->view('panels/footer');
}

```

```

public function savePathCont() {
$connetions=$this->input->post('connetions');
$stations=$this->input->post('stations');
$typeCode=$this->input->post('typeCode');
}

```

```

echo $this->document_model-
>edit_typePath($typeCode,$connetions,$stations);
}

```

```

public function getStationCreatePermission() {
$typeCode=$this->input->post('typeCode');
$stationCode=$this->input->post('stationCode');
}

```

```

$createpermission = $this->document_model-
>getCreatePermission($typeCode,$stationCode);

```

```

if (count($createpermission)<=0) {
echo "n";
} else {
echo $createpermission['PERTYPE'];
}
}

```

```

public function getDocuTypeForm() {
$typeCode=$this->input->post('typeCode');
}

```

```

$params = $this->document_model-
>get_documentTypeParaWOLimiters($typeCode);

```

```

$echoText = "<table>";
foreach ($params as $param) {
$inputType = $param['INPUTTYPE'];
if ($inputType=='text') {
$echoText .= "<tr><td style=\"padding:5px 10px 5px
10px;\>". $param['PARAMNAME']."</td><td>";
$echoText .= "INPUT ";
$echoText .= "TYPE=\"TEXT\"
NAME=\"\". $param['PARAMCODE']."\"
ID=\"\". $param['PARAMCODE']."\"
VALUE=\"\". $param['DEFAULTVAL']."\" READONLY ";
$echoText .= "</td></tr>";
}
}

```

```

} else if ($inputType=='select') {
$echoText .= "<tr><td style=\"padding:5px 10px 5px
10px;\>". $param['PARAMNAME']."</td><td>";
$echoText = "<select ";
$echoText = "name=\"\". $param['PARAMCODE']."\"
id=\"\". $param['PARAMCODE']."\"
VALUE=\"\". $param['DEFAULTVAL']."\">";
$selectOptions = $this->document_model-
>get_paramChoices($param['PARAMCODE']);
foreach ($selectOptions as $selectOption) {
$echoText .= "<option
value=\"\". $selectOption['CHOICENAME']."\" ";
if ( $selectOption['CHOICENAME'] == $param['DEFAULTVAL'] ) {
$echoText = "selected";
}
$echoText = " >". $selectOption['CHOICENAME']."</option>";
}
}

```

```

$echoText = "</select>";
$echoText = "</td></tr>";
} else if ($inputType=='checkbox') {
$echoText .= "<tr><td style=\"padding:5px 10px 5px
10px;\>". $param['PARAMNAME']."</td><td>";
$echoText = "<input type=\"checkbox\" id=\"check1\"
disabled=\"disabled\" ";
if ( isset($param['DEFAULTVAL']) ) {
$echoText = "checked=\"checked\"";
}
$echoText = " />";
$echoText = "</td></tr>";
} else if ($inputType=='radio') {
$echoText .= "<tr><td style=\"padding:5px 10px 5px
10px;\>". $param['PARAMNAME']."</td><td>";
$selectOptions = $this->document_model-
>get_paramChoices($param['PARAMCODE']);
}
}

```



```

public function assignApplicant($userId) {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}

$data['companies'] = $this->document_model-
>getUserCompaniesChoice($userId);

$data['usercompanies'] = $this->document_model-
>getUserCompanies($userId);

$data['userMetaData'] = $this->document_model-
>getUserMeta($userId);
$data['userid'] = $userId;

$data['title'] = 'Assign Applicant';
$data['status'] = '';

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/assignApplicant',$data);
$this->load->view('panels/footer');
}

public function saveTypeMeta() {
$name = $this->input->post('name');
$desc=$this->input->post('desc');
$typeCode=$this->input->post('typeCode');

echo $this->document_model-
>edit_typeMeta($name,$desc,$typeCode);
}

public function typeEditParam($typeCode) {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype') != "admin" ) {
redirect('/');
}
$data['typeCode'] = $typeCode;
$data['typeData'] = $this->document_model-
>get_documentTypeMeta($typeCode);
$data['typeParamChoice'] = $this->document_model-
>get_documentTypeParamChoice();

$data['title'] = 'Edit Parameters';
$data['status'] = '';

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/typeEditParam',$data);
$this->load->view('panels/footer');
}

public function adminDocs() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}
$data['title'] = 'Documents';
$data['status'] = '';
$data['stations'] = $this->document_model-
>get_stationRowCount();

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/adminDoc',$data);
}

```

```

$this->load->view('panels/footer');
}

public function viewType() {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype')!="admin" ) {
redirect('/');
}
$data['title'] = 'Document Types';
$data['status'] = '';
$data['stations'] = $this->document_model-
>get_stationRowCount();

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewAllDoctype',$data);
$this->load->view('panels/footer');
}

public function viewDocs() {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype')!="admin" && $this->session-
>userdata('usertype')!="head" ) {
redirect('/');
}
$data['title'] = 'Documents';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewAdminDocs',$data);
$this->load->view('panels/footer');
}

public function viewCompanies() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}
$data['title'] = 'Documents';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewCompanies',$data);
$this->load->view('panels/footer');
}

public function viewStationDocs() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}
$data['title'] = 'Documents';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewStationDocs',$data);
$this->load->view('panels/footer');
}

public function viewApplicantDocs() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}

```

```

}
$data['title'] = 'Documents';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('doctype/viewApplicantDocs', $data);
$this->load->view('panels/footer');
}

public function viewArcDocs() {
    $data['title'] = 'Archived Documents';
    $data['status'] = '';
    //$data['stations'] = $this->document_model->get_docs(null);

    $this->load->view('panels/header', $data);
    $this->load->view('panels/sidebar');
    $this->load->view('doctype/viewArcDocs', $data);
    $this->load->view('panels/footer');
}

public function viewArcDoc($arcCode) {
    if ( $this->session->userdata('login_state') == FALSE) {
        redirect('/');
    }

    $data['company'] = $this->getCompanyStr($this->document_model->getArcDocCompanies($arcCode));

    $data['arcDocumentMeta'] = $this->document_model->getArcDocumentMeta($arcCode);
    $data['connections'] = $this->document_model->getMetaArcConnections($arcCode);

    $max = 100;

    //get stations
    //add a row for number of docs per station
    $data['stations'] = $this->document_model->getMetaArcStations($arcCode);

    $startDate = $this->document_model->getDocLocArcStartDate($arcCode);
    $endDate = $this->document_model->getDocLocArcEndDate($arcCode);

    $data['dateRnge'] = $this->getDateDiff2($startDate['MINDATE'], $endDate['MAXDATE']);

    $data['arcCode'] = $arcCode;

    $data['title'] = 'Document';
    $data['status'] = '';
    //$data['stations'] = $this->document_model->get_docs(null);

    $this->load->view('panels/header', $data);
    $this->load->view('panels/sidebar');
    $this->load->view('doctype/viewArcDoc', $data);
    $this->load->view('panels/footer');
}

```

```

public function viewDoc($barCode) {
    if ( $this->session->userdata('login_state') == FALSE) {
        redirect('/');
    }

    $data['documentMeta'] = $this->document_model->getDocumentMeta($barCode);
    //$data['pathdata'] = $this->document_model->getDocumentMeta($barCode);
    $data['connections'] = $this->document_model->getMetaConnections($barCode);

    $max = 100;

    //get stations
    //add a row for number of docs per station
    $stationData = $this->document_model->getMetaStations($barCode);

    for ( $i=0; $i < count($stationData); $i++){
        $doccnt = $this->document_model->get_stationdoccount($stationData[$i]['STATIONCODE']);
        $cnt = $doccnt['DOCCOUNT'];
        $stationData[$i] = array_merge($stationData[$i], array('docCount'=>$cnt, 'color'=>$this->rgbToHex($cnt, $max-$cnt, 0, $max)));
    }

    $data['stations'] = $stationData;

    $data['lock'] = "n";

    $lock = $this->document_model->getDocLock($barCode, $this->session->userdata('userid'));

    if (count($lock)>0) {
        $data['lock'] = $lock[0]['STATUS'];
    } else {
        $lockExists = $this->document_model->getDocLockExists($barCode, $this->session->userdata('userid'));
        if (count($lockExists)<=0) {
            $data['lock'] = "n";
        }
    }

    $startDate = $this->document_model->getDocLocStartDate($barCode);
    $data['docLoc'] = $this->document_model->getDocLoc($barCode);

    $data['docStatus'] = $this->getDocStatus($data['documentMeta']['STATIONCODE'], $data['connections']);
    $data['dateRnge'] = $this->getDateDiff($startDate['MINDATE']);

    $data['barCode'] = $barCode;

    $data['title'] = 'Document';
    $data['status'] = '';
    //$data['stations'] = $this->document_model->get_docs(null);

```

```

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewAdminDoc',$data);
$this->load->view('panels/footer');
}

private function rgbToHex($r, $g, $b,$max) {
return "#".$this->componentToHex($r,$max).$this->
componentToHex($g,$max).$this->componentToHex($b,$max);
}

private function componentToHex($c,$max) {
if ( $c < 0 ) {
$c = 0;
} else if ( $c > $max ) {
$c = 255;
} else {
$c = floor(($c/$max)*255);
}
$numClr = base_convert($c, 10, 16);
return strlen($numClr) == 1 ? "0".$numClr : $numClr;
}

public function formview($barCode) {
if ( $this->session->userdata('login_state') == FALSE) {
redirect('/');
}
$data['documentMeta'] = $this->document_model->
getDocumentMetaData($barCode);

$selectOptions = array();

if( $this->session->userdata('usertype') == "admin" ) {
$data['params'] = $this->document_model->
get_documentParaWOLimiters($barCode);
} else {
$data['params'] = $this->document_model->
get_documentParaWOLimiters($barCode,$this->session->
userdata('stationcode'));
}

foreach ($data['params'] as $param) {
$selectOptions[$param['METAPARAMCODE']] = $this->
document_model->
get_metaParamChoices($param['METAPARAMCODE']);
}

$data['selectOptions'] = $selectOptions;

$data['title'] = 'Document Form';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewForm',$data);
$this->load->view('panels/footer');
}

public function arcformview($arCcode) {
if ( $this->session->userdata('login_state') == FALSE) {
redirect('/');
}

```

```

$data['documentMeta'] = $this->document_model->
getArcDocumentMetaData($arCcode);

$selectOptions = array();

$data['params'] = $this->document_model->
get_documentMetaPara($arCcode);

foreach ($data['params'] as $param) {
$selectOptions[$param['METAPARAMARCCODE']] = $this->
document_model->
get_metaArcParamChoices($param['METAPARAMARCCODE']);
}

$data['selectOptions'] = $selectOptions;

$data['title'] = 'Document Form';
$data['status'] = '';
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewArcForm',$data);
$this->load->view('panels/footer');
}

public function formedit($barCode) {
if ( $this->session->userdata('login_state') == FALSE) {
redirect('/');
}
$data['documentMeta'] = $this->document_model->
getDocumentMetaData($barCode);

$selectOptions = array();

if( $this->session->userdata('usertype') == "admin" ) {
$data['params'] = $this->document_model->
get_documentParaWOLimiters($barCode);
} else {
$data['params'] = $this->document_model->
get_documentParaWOLimiters($barCode,$this->session->
userdata('stationcode'));
}

foreach ($data['params'] as $param) {
$selectOptions[$param['METAPARAMCODE']] = $this->
document_model->
get_metaParamChoices($param['METAPARAMCODE']);
}

if( $this->session->userdata('usertype') == "admin" ) {
$nxt = $this->document_model->
getNextStations($barCode,$data['documentMeta']['STATIONCO
DE']);
} else {
$nxt = $this->document_model->getNextStations($barCode,$this->
session->userdata('stationcode'));
}

$nxtval = array();
$i=0;
foreach ($nxt as $elem) {
$nxtval[$i]['METACONTARGET'] = $elem['METACONTARGET'];
$nxtval[$i]['METACONDESCR'] = $elem['METACONDESCR'];

```

```

if ( $elem['METACONTARGET'] == 0 || $elem['METACONTARGET']
== '0' ) {
$nextval[$i]['STATIONNAME'] = "Finish Doc";
} else {
$stnme = $this->document_model-
>getStationName($elem['METACONTARGET']);
$nextval[$i]['STATIONNAME'] = $stnme['STATIONNAME'];
}
$i++;
}

$data['nextstations'] = $nextval;

$data['selectOptions'] = $selectOptions;

$data['mailRecipients'] = $this->document_model-
>getMailTo($barCode);

$data['title'] = 'Document Form';
$data['status'] = "";
//$data['stations'] = $this->document_model->get_docs(null);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
if ( $this->session->userdata('usertype') == "admin" ) {
$this->load->view('docutype/adminEditForm',$data);
} else {
$this->load->view('docutype/editForm',$data);
}
$this->load->view('panels/footer');
}

public function viewStations() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}
$data['title'] = 'Stations';
$data['status'] = "";

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewAllStations',$data);
$this->load->view('panels/footer');
}

public function viewStation($stationCode) {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}
$data['title'] = 'Station';
$data['status'] = "";

$data['stationMetaData'] = $this->document_model-
>get_stationMetaData($stationCode);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewStation',$data);
$this->load->view('panels/footer');
}

public function myProfile() {
if ( $this->session->userdata('login_state') == FALSE ) {
redirect('/');
}

$data['userData'] = $this->document_model->get_profile();
$data['title'] = 'Stations';
$data['status'] = "";

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/userAccount',$data);
$this->load->view('panels/footer');
}

public function viewDataTypes() {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype') != "admin" ) {
redirect('/');
}
$data['title'] = 'Data Types';
$data['status'] = "";

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewDataType',$data);
$this->load->view('panels/footer');
}

public function users() {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype') != "admin" && $this->session-
>userdata('usertype') != "head" ) {
redirect('/');
}
}
$data['title'] = 'Users';
$data['status'] = "";
$data['stations'] = $this->document_model-
>get_stationRowCount();

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/viewUsers',$data);
$this->load->view('panels/footer');
}

public function editPermission($stationCode) {
if ( $this->session->userdata('login_state') == FALSE || $this-
>session->userdata('usertype') != "admin" ) {
redirect('/');
}
}
$data['title'] = 'Edit Permission';
$data['status'] = "";

$data['stationCode'] = $stationCode;
$data['stationMetaData'] = $this->document_model-
>get_stationMetaData($stationCode);

$this->load->view('panels/header', $data);
$this->load->view('panels/sidebar');
$this->load->view('docutype/editPermission',$data);
$this->load->view('panels/footer');
}

public function loginEval() {
$username = $this->input->post('username');
$password = $this->input->post('password');

if($this->document_model->isUser($username,$password)){
$this->home();
}
}

```

```

} else {
if($this->session->userdata('login_state') == FALSE){
$data['title'] = 'Login';
$data['status'] = "Username or password incorrect";
$this->load->view('panels/header',$data);
$this->load->view('panels/sidebar');
$this->load->view('login',$data);
$this->load->view('panels/footer');
} else {
$this->home();
}
}
}

public function getStationDocElements($stationCode) {

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input->post('idx'):'TITLE'; //
get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if ( $idx == 'ELAPSED' || $_this->input->post("searchField") ==
'ELAPSED' ) {

$count = count ($this->document_model-
>get_docuRowCount($stationCode));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_docs("TITLE", "", "0",
$count, "1",$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

$rows;

foreach ($typeList as $typeList_item):

$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getStationStartDate($typeList_item['BARCODE'],$stationCode);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);

$rows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];

$rows[$dateRnge.$i]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['TYPENAME'],$dateRnge,$docStatus);
$i++;
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($rows);
} else {
ksort($rows);
}

$i=0;
$j=0;
foreach ($rows as $row){
if ( $i >= $start && $i < $limit*$page ) {
$data->rows[$j] = $row;
$j++;
}
}
} else if ( $idx == 'PROGRESS' || $_this->input->post("searchField")
== 'PROGRESS' ) {

$count = count ($this->document_model-
>get_docuRowCount($stationCode));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_docs("TITLE", "", "0" ,
$count, "1",$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
$j=0;
$rows;
$docStatus_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):

$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getStationStartDate($typeList_item['BARCODE'],$stationCode);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);

if ( $prev == -1 ) {
$prev = $docStatus;
}
if ( $prev != $docStatus ) {
$prev = $docStatus;
if ( $this->isArray($docStatus_holder,$docStatus) ) {
$j=$docStatus_holder[$docStatus];
} else {

```

```

$j=0;
}
}

$dataRows[$docStatus." ".$j]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus." ".$j]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['TYPENAME'],$dateRnge,$docStatus);
$i++;
$j++;
$docStatus_holder[$docStatus] = $j;
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($dataRows);
} else {
ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
if ( $i >= $start && $i < $limit*$page ) {
if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

switch ($this->input->post("searchOper")) {
case "eq":
if ( $this->input->post("searchString") == $dataRow['cell'][5] ) {
$data->rows[$j] = $dataRow;
}
break;
case "ne":
if ( $this->input->post("searchString") != $dataRow['cell'][5] ) {
$data->rows[$j] = $dataRow;
}
break;
case "lt":
if ( $dataRow['cell'][5] < $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "le":
if ( $dataRow['cell'][5] <= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "gt":
if ( $dataRow['cell'][5] > $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "ge":
if ( $dataRow['cell'][5] >= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "nu":
if ( !isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
break;
case "nn":
if ( isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
}

} else {
break;
case "cn":
$data->rows[$j] = $dataRow;
break;
} else {
$data->rows[$j] = $dataRow;
}
}
}

} else {
if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = '". $this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%";
break;
}

$queryString = "(".$this->input->
post("searchField").$operation.")";
}

if(!$sidx) $sidx = 'TITLE';

$count = count ($this->document_model-
>get_docuRowCount($stationCode));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;

```

```

$start = 0;
if ($page>0)
$start = $limit*$page - $limit; // do not put $limit*($page - 1)

$typeList = $this->document_model->get_docs($sidx, $sord,
$start , $limit, $queryString,$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getStationStartDate($typeList_item['BARCODE'],$stationCode);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);

$data->rows[$i]['cell'] = $typeList_item['BARCODE'];

$data->rows[$i]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['TYPENAME'],$dateRnge,$docStatus);
$i++;
endforeach;
}

echo json_encode($data);
}

public function getDocApplicantElements() {

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])?$_this->input->post('sidx'):'ELAPSED';
// get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):'DESC'; //
get the direction

$queryString = "1";

if ( $sidx == 'ELAPSED' || $_this->input->post("searchField") ==
'ELAPSED' ) {

$count = count ($this->document_model-
>get_applicantDocuRowCount($this->session-
>userdata('userid')));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*($page - 1)

```

```

$typeList = $this->document_model-
>get_applicantDocuRowCount($this->session->userdata('userid'));

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

$dataRows;

foreach ($typeList as $typeList_item):

$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model-
>getCompanies($typeList_item['BARCODE']));

$dataRows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];

$dataRows[$dateRnge.$i]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$
docStatus,$company);
$i++;
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($dataRows);
} else {
ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
if ( $i >= $start && $i < $limit*$page ) {
$data->rows[$j] = $dataRow;
$j++;
}
}
} else if ( $sidx == 'PROGRESS' || $_this->input->post("searchField")
== 'PROGRESS' ) {

$count = count ($this->document_model-
>get_applicantDocuRowCount($this->session-
>userdata('userid')));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*($page - 1)

$typeList = $this->document_model-
>get_applicantDocuRowCount($this->session->userdata('userid'));

```

```

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
$j=0;
$rows;
$docStatus_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):

$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model-
>getCompanies($typeList_item['BARCODE']));

if ( $prev == -1 ) {
$prev = $docStatus;
}
if ( $prev != $docStatus ) {
$prev = $docStatus;
if ( $this->isArray($docStatus_holder,$docStatus) ) {
$j=$docStatus_holder[$docStatus];
} else {
$j=0;
}
}

$rows[$docStatus.".".$j]['id'] = $typeList_item['BARCODE'];
$rows[$docStatus.".".$j]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList
_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$
docStatus,$company);
$i++;
$j++;
$docStatus_holder[$docStatus] = $j;
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
ksort($rows);
} else {
ksort($rows);
}

$i=0;
$j=0;
foreach ($rows as $row){
if ( $i >= $start && $i < $limit*$page ) {
if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

switch ($this->input->post("searchOper")) {
case "eq":
if ( $this->input->post("searchString") == $row['cell'][5] ) {
$rows[$j] = $row;
}
break;
case "ne":
if ( $this->input->post("searchString") != $row['cell'][5] ) {

```

```

$rows[$j] = $row;
}
break;
case "lt":
if ( $row['cell'][5] < $this->input->post("searchString") ) {
$rows[$j] = $row;
}
break;
case "le":
if ( $row['cell'][5] <= $this->input->post("searchString") ) {
$rows[$j] = $row;
}
break;
case "gt":
if ( $row['cell'][5] > $this->input->post("searchString") ) {
$rows[$j] = $row;
}
break;
case "ge":
if ( $row['cell'][5] >= $this->input->post("searchString") ) {
$rows[$j] = $row;
}
break;
case "nu":
if ( !isset($row['cell'][5]) ) {
$rows[$j] = $row;
}
break;
case "nn":
if ( isset($row['cell'][5]) ) {
$rows[$j] = $row;
}
break;
case "cn":
$rows[$j] = $row;
break;
}
} else {
$rows[$j] = $row;
}
}
}
}
}
}
}

} else {
if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = ".$this->input->post("searchString")."";
break;
case "ne":
$operation = " <> ".$this->input->post("searchString")."";
break;
case "lt":
$operation = " < ".$this->input->post("searchString")."";
break;
case "le":
$operation = " <= ".$this->input->post("searchString")."";
break;

```



```

case "gt":
$operation = ">" . $this->input->post("searchString")."";
break;
case "ge":
$operation = ">=" . $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $this->input->post("searchString"). "%'";
break;
}

$queryString = "(" . $this->input->post("searchField"). $operation. ")";
}

if(!$sidx) $sidx = 'TITLE';

$count = count ($this->document_model->get_applicantDocuRowCount($this->session->userdata('userid')));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_applicantDocuRowCount($this->session->userdata('userid'));

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$connections = $this->document_model->getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model->getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this->getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model->getCompanies($typeList_item['BARCODE']));

$data->rows[$i]['cell'] = $typeList_item['BARCODE'];

$data->rows[$i]['cell'] =
array($typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
endforeach;
}

```

```

echo json_encode($data);
}

public function getDocStationElements($stationCode) {

$page = isset($_POST['page'])? $this->input->post('page'):1; // get the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])? $this->input->post('sidx'):'ELAPSED'; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])? $this->input->post('sord'):'DESC'; // get the direction

$queryString = "1";

if ( $sidx == 'ELAPSED' || $this->input->post("searchField") == 'ELAPSED' ) {

$count = count ($this->document_model->get_docuRowCount($stationCode));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_docs("TITLE", "", "0", $count, "1",$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

$dataRows;

foreach ($typeList as $typeList_item):

$connections = $this->document_model->getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model->getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this->getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model->getCompanies($typeList_item['BARCODE']));

$lock = $this->document_model->getDocLock($typeList_item['BARCODE'],$this->session->userdata('userid'));

if (count($lock)>0) {
$dataRows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$dataRows[$dateRnge.$i]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
}
}

```

```

$i++;
} else {
$lockExists = $this->document_model-
>getDocLockExists($typeList_item['BARCODE'],$this->session-
>userdata('userid'));
if (count($lockExists)<=0) {
$dataRows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$dataRows[$dateRnge.$i]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$
typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$date
eRnge,$docStatus,$company);
$i++;
}
}
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($dataRows);
} else {
ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
if ( $i >= $start && $i < $limit*$page ) {
$data->rows[$j] = $dataRow;
$j++;
}
}
} else if ( $sidx == 'PROGRESS' || $this->input->post("searchField")
== 'PROGRESS' ) {

$count = count ($this->document_model-
>get_docuRowCount($stationCode));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*($page - 1)

$typeList = $this->document_model->get_docs("TITLE", "", "0" ,
$count, "1",$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
$j=0;
$dataRows;
$docStatus_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):

$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);

```

```

$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model-
>getCompanies($typeList_item['BARCODE']));

if ( $prev == -1 ) {
$prev = $docStatus;
}
if ( $prev != $docStatus ) {
$prev = $docStatus;
if ( $this->isArray($docStatus_holder,$docStatus) ) {
$j=$docStatus_holder[$docStatus];
} else {
$j=0;
}
}

$lock = $this->document_model-
>getDocLock($typeList_item['BARCODE'],$this->session-
>userdata('userid'));

if (count($lock)>0) {
$dataRows[$docStatus.".".$j]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus.".".$j]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARC
ODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeLI
st_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
$j++;
$docStatus_holder[$docStatus] = $j;
} else {
$lockExists = $this->document_model-
>getDocLockExists($typeList_item['BARCODE'],$this->session-
>userdata('userid'));
if (count($lockExists)<=0) {
$dataRows[$docStatus.".".$j]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus.".".$j]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$
typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$date
eRnge,$docStatus,$company);
$i++;
$j++;
$docStatus_holder[$docStatus] = $j;
}
}
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($dataRows);
} else {
ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
if ( $i >= $start && $i < $limit*$page ) {
if ($_POST['_search']=='true'){
$query = json_decode($this->input->post('filters')); // get the
query

switch ($this->input->post("searchOper")) {
case "eq":
if ( $this->input->post("searchString") == $dataRow['cell'][5] ) {
$data->rows[$j] = $dataRow;
}
}
}
}
}
}

```

```

break;
case "ne":
if ( $this->input->post("searchString") != $dataRow['cell'][5] ) {
$data->rows[$j] = $dataRow;
}
break;
case "lt":
if ( $dataRow['cell'][5] < $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "le":
if ( $dataRow['cell'][5] <= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "gt":
if ( $dataRow['cell'][5] > $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "ge":
if ( $dataRow['cell'][5] >= $this->input->post("searchString") ) {
$data->rows[$j] = $dataRow;
}
break;
case "nu":
if ( !isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
break;
case "nn":
if ( isset($dataRow['cell'][5]) ) {
$data->rows[$j] = $dataRow;
}
break;
case "cn":
$data->rows[$j] = $dataRow;
break;
} else {
$data->rows[$j] = $dataRow;
}
$j++;
}
}
} else {

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = '". $this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $this->input->post("searchString"). "%'";
break;
}

$queryString = "(" . $this->input->
post("searchField"). $operation . ")";

if(!$sidx) $sidx = 'TITLE';

$count = count ($this->document_model-
>get_docuRowCount($stationCode));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1

$typeList = $this->document_model->get_docs($sidx, $sord,
$start , $limit, $queryString,$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$connections = $this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model-
>getCompanies($typeList_item['BARCODE']));

$lock = $this->document_model-
>getDocLock($typeList_item['BARCODE'],$this->session-
>userdata('userid'));

if (count($lock)>0) {

```

```

$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$data->rows[$i]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARCODE'],
$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],
$dateRnge,$docStatus,$company);
$i++;
} else {
$lockExists = $this->document_model->getDocLockExists($typeList_item['BARCODE'],$this->session->userdata('userid'));
if (count($lockExists)<=0) {
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$data->rows[$i]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],
$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
}
}
endforeach;
}

echo json_encode($data);
}

```

```

public function getArcDocElements() {

```

```

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input->post('idx'):'TITLE'; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; // get the direction

```

```

$queryString = "1";

```

```

if($_POST['search']=='true'){
$query = json_decode($_this->input->post('filters')); // get the query

```

```

$operation = "";

```

```

switch ($_this->input->post("searchOper")) {
case "eq":
$operation = " = '". $_this->input->post("searchString"). "'";
break;
case "ne":
$operation = " <> '". $_this->input->post("searchString"). "'";
break;
case "lt":
$operation = " < '". $_this->input->post("searchString"). "'";
break;
case "le":
$operation = " <= '". $_this->input->post("searchString"). "'";
break;
case "gt":
$operation = " > '". $_this->input->post("searchString"). "'";
break;
case "ge":
$operation = " >= '". $_this->input->post("searchString"). "'";
break;
case "nu":
$operation = " IS NULL";

```

```

break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $_this->input->post("searchString"). "%'";
break;
}

```

```

$queryString = "(" . $_this->input->post("searchField"). $operation . ")";
}

```

```

if(!$idx) $idx = 'TITLE';

```

```

$count = count ($this->document_model->get_arcdoccount());
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

```

```

$typeList = $this->document_model->get_arcdocs($idx, $sord, $start, $limit, $queryString);

```

```

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

```

```

$i=0;

```

```

foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['ARCCODE'];
$startDate = $this->document_model->getDocLocArcStartDate($typeList_item['ARCCODE']);
$endDate = $this->document_model->getDocLocArcEndDate($typeList_item['ARCCODE']);
$dateRnge = $this->getDateDiff2($startDate['MINDATE'],$endDate['MAXDATE']);
$company = $this->getCompanyStr($this->document_model->getArcDocCompanies($typeList_item['ARCCODE']));

```

```

$data->rows[$i]['cell'] = $typeList_item['ARCCODE'];

```

```

$data->rows[$i]['cell'] =
array($typeList_item['ARCCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],
$dateRnge,$company);
$i++;
endforeach;

```

```

echo json_encode($data);
}

```

```

public function getDocElements() {

```

```

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input->post('idx'):'TITLE'; // get index row - i.e. user click to sort

```

```

$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if ( $sidx == 'ELAPSED' || $_this->input->post("searchField") ==
'ELAPSED' ) {

$count = count ($_this->document_model-
>get_docuRowCount(null));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $_this->document_model->get_docs("TITLE", "", "0" ,
$count, "1",null);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

$rows;

foreach ($typeList as $typeList_item):

$connections = $_this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $_this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $_this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $_this->getDateDiff($startDate['MINDATE']);
$company = $_this->getCompanyStr($_this->document_model-
>getCompanies($typeList_item['BARCODE']));

$lock = $_this->document_model-
>getDocLock($typeList_item['BARCODE'],$this->session-
>userdata('userid'));

if (count($lock)>0) {
$rows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$rows[$dateRnge.$i]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARC
ODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeLi
st_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
} else {
$lockExists = $_this->document_model-
>getDocLockExists($typeList_item['BARCODE'],$this->session-
>userdata('userid'));
if (count($lockExists)<=0) {
$rows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$rows[$dateRnge.$i]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$
typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dat
eRnge,$docStatus,$company);
$i++;
}
}
} else {
$rows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$rows[$dateRnge.$i]['cell'] =
array("r","r",$typeList_item['BARCODE'],$typeList_item['TITLE'],$t
ypeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$date
Rnge,$docStatus,$company);
$i++;
}
}
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($rows);
} else {
ksort($rows);
}

$i=0;
$j=0;
foreach ($rows as $row){
if ( $i >= $start && $i < $limit*$page ) {
$rows[$j] = $row;
$j++;
}
}
} else if ( $sidx == 'PROGRESS' || $_this->input->post("searchField")
== 'PROGRESS' ) {

$count = count ($_this->document_model-
>get_docuRowCount(null));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $_this->document_model->get_docs("TITLE", "", "0" ,
$count, "1",null);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
$j=0;
$rows;
$docStatus_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):

$connections = $_this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $_this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $_this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $_this->getDateDiff($startDate['MINDATE']);
$company = $_this->getCompanyStr($_this->document_model-
>getCompanies($typeList_item['BARCODE']));

if ( $prev == -1 ) {

```

```

} else {
$rows[$dateRnge.$i]['id'] = $typeList_item['BARCODE'];
$rows[$dateRnge.$i]['cell'] =
array("r","r",$typeList_item['BARCODE'],$typeList_item['TITLE'],$t
ypeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$date
Rnge,$docStatus,$company);
$i++;
}
}
endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($rows);
} else {
ksort($rows);
}

$i=0;
$j=0;
foreach ($rows as $row){
if ( $i >= $start && $i < $limit*$page ) {
$rows[$j] = $row;
$j++;
}
}
} else if ( $sidx == 'PROGRESS' || $_this->input->post("searchField")
== 'PROGRESS' ) {

$count = count ($_this->document_model-
>get_docuRowCount(null));
if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $_this->document_model->get_docs("TITLE", "", "0" ,
$count, "1",null);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
$j=0;
$rows;
$docStatus_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):

$connections = $_this->document_model-
>getMetaConnections($typeList_item['BARCODE']);
$startDate = $_this->document_model-
>getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $_this-
>getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $_this->getDateDiff($startDate['MINDATE']);
$company = $_this->getCompanyStr($_this->document_model-
>getCompanies($typeList_item['BARCODE']));

if ( $prev == -1 ) {

```

```

$prev = $docStatus;
}
if ( $prev != $docStatus ) {
$prev = $docStatus;
if ( $this->isArray($docStatus_holder,$docStatus) ) {
$ج=$docStatus_holder[$docStatus];
} else {
$ج=0;
}
}

$lock = $this->document_model-
>getDocLock($typeList_item['BARCODE'],$this->session-
>userdata('userid'));

if (count($lock)>0) {
$dataRows[$docStatus.".".$ج]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus.".".$ج]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARC
ODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeLi
st_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$ج++;
$ج++;
$docStatus_holder[$docStatus] = $ج;
} else {
$lockExists = $this->document_model-
>getDocLockExists($typeList_item['BARCODE'],$this->session-
>userdata('userid'));
if (count($lockExists)<=0) {
$dataRows[$docStatus.".".$ج]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus.".".$ج]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$
typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dat
eRnge,$docStatus,$company);
$ج++;
$ج++;
$docStatus_holder[$docStatus] = $ج;
} else {
$dataRows[$docStatus.".".$ج]['id'] = $typeList_item['BARCODE'];
$dataRows[$docStatus.".".$ج]['cell'] =
array("r","r",$typeList_item['BARCODE'],$typeList_item['TITLE'],$t
ypeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$date
Rnge,$docStatus,$company);
$ج++;
$ج++;
$docStatus_holder[$docStatus] = $ج;
}
}

endforeach;

if ( $sord == 'ASC' || $sord == 'asc' ) {
krsort($dataRows);
} else {
ksort($dataRows);
}

$ج=0;
$ج=0;
foreach ($dataRows as $dataRow){
if ( $ج >= $start && $ج < $limit*$page ) {
if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

```

```

switch ($this->input->post("searchOper")) {
case "eq":
if ( $this->input->post("searchString") == $dataRow['cell'][5] ) {
$data->rows[$ج] = $dataRow;
}
break;
case "ne":
if ( $this->input->post("searchString") != $dataRow['cell'][5] ) {
$data->rows[$ج] = $dataRow;
}
break;
case "lt":
if ( $dataRow['cell'][5] < $this->input->post("searchString") ) {
$data->rows[$ج] = $dataRow;
}
break;
case "le":
if ( $dataRow['cell'][5] <= $this->input->post("searchString") ) {
$data->rows[$ج] = $dataRow;
}
break;
case "gt":
if ( $dataRow['cell'][5] > $this->input->post("searchString") ) {
$data->rows[$ج] = $dataRow;
}
break;
case "ge":
if ( $dataRow['cell'][5] >= $this->input->post("searchString") ) {
$data->rows[$ج] = $dataRow;
}
break;
case "nu":
if ( !isset($dataRow['cell'][5]) ) {
$data->rows[$ج] = $dataRow;
}
break;
case "nn":
if ( isset($dataRow['cell'][5]) ) {
$data->rows[$ج] = $dataRow;
}
break;
case "cn":
$data->rows[$ج] = $dataRow;
break;
} else {
$data->rows[$ج] = $dataRow;
}
$ج++;
}
$ج++;
}

} else {

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = "".$this->input->post("searchString")."";
break;
case "ne":

```

```

$operation = " <> ". $this->input->post("searchString")."";
break;
case "lt":
$operation = " < ". $this->input->post("searchString")."";
break;
case "le":
$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%";
break;
}

$queryString = "(".$this->input->post("searchField").$operation.")";
}

if(!$sidx) $sidx = 'TITLE';

$count = count ($this->document_model->get_docuRowCount(null));
if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_docs($sidx, $sord,
$start , $limit, $queryString,null);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['BARCODE'];
$connections = $this->document_model->getMetaConnections($typeList_item['BARCODE']);
$startDate = $this->document_model->getDocLocStartDate($typeList_item['BARCODE']);
$docStatus = $this->getDocStatus($typeList_item['STATIONCODE'],$connections);
$dateRnge = $this->getDateDiff($startDate['MINDATE']);
$company = $this->getCompanyStr($this->document_model->getCompanies($typeList_item['BARCODE']));

```

```

$lock = $this->document_model->getDocLock($typeList_item['BARCODE'],$this->session->userdata('userid'));
if (count($lock)>0) {
$data->rows[$i]['cell'] = $typeList_item['BARCODE'];
$data->rows[$i]['cell'] =
array($lock[0]['STATUS'],$lock[0]['STATUS'],$typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
} else {
$lockExists = $this->document_model->getDocLockExists($typeList_item['BARCODE'],$this->session->userdata('userid'));
if (count($lockExists)<=0) {
$data->rows[$i]['cell'] = $typeList_item['BARCODE'];
$data->rows[$i]['cell'] =
array("n","n",$typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
} else {
$data->rows[$i]['cell'] = $typeList_item['BARCODE'];
$data->rows[$i]['cell'] =
array("r","r",$typeList_item['BARCODE'],$typeList_item['TITLE'],$typeList_item['TYPENAME'],$typeList_item['STATIONNAME'],$dateRnge,$docStatus,$company);
$i++;
}
}
endforeach;

echo json_encode($data);
}

private function getCompanyStr($companies) {
$retstr = "";
foreach ( $companies as $company ) {
$retstr.= "[".$company['COMPANYNAME']."] ";
}
return $retstr;
}

private function getDateDiff($mindate) {
$time = strtotime($mindate);
$datediff = time() - $time;

if ( $datediff < 0 ) {
$datediff = 0;
}

$year = floor($datediff/(60*60*24*30*12));
$month = floor($datediff/(60*60*24*30)) - $year*(12);
$day = floor($datediff/(60*60*24)) - $month*(30) - $year*(12*30);
$hrs = floor($datediff/(60*60)) - $day*(24) - $month*(30*24) - $year*(12*30*24);

$year = strlen($year) == 1 ? "0".$year : $year;
$month = strlen($month) == 1 ? "0".$month : $month;
$day = strlen($day) == 1 ? "0".$day : $day;
$hrs = strlen($hrs) == 1 ? "0".$hrs : $hrs;

return $year.".".$month.".".$day.".".$hrs;
}

```

```

private function getDateDiff2($mindate,$maxdate) {
    $datediff = strtotime($maxdate) - strtotime($mindate);

    if ( $datediff < 0 ) {
        $datediff = 0;
    }

    $year = floor($datediff/(60*60*24*30*12));
    $month = floor($datediff/(60*60*24*30)) - $year*(12);
    $day = floor($datediff/(60*60*24)) - $month*(30) -
        $year*(12*30);
    $hrs = floor($datediff/(60*60)) - $day*(24) - $month*(30*24) -
        $year*(12*30*24);

    $year = strlen($year) == 1 ? "0".$year : $year;
    $month = strlen($month) == 1 ? "0".$month : $month;
    $day = strlen($day) == 1 ? "0".$day : $day;
    $hrs = strlen($hrs) == 1 ? "0".$hrs : $hrs;

    return $year."-".$month."-".$day."-".$hrs;
}

private function getDocStatus($stationcode,$connections) {
    if ( $stationcode == "0" ) {
        return 100;
    }

    if ( $stationcode == "-1" ) {
        return 0;
    }

    $conn = $this->getArray($connections);
    $stationsArr = $this->getStations($connections);
    $fromStart = $this->getDocPos("-1",$stationcode,$conn,0)-1;
    $toEnd = $this->getDocPos($stationcode,"0",$conn,0);

    //return $fromStart."->".$toEnd." ".$stationcode;

    if ( $fromStart <= 0 || $toEnd <= 0 ) {
        return 0;
    }

    return round(($fromStart/($fromStart + $toEnd))*100);
}

private function getArray($connections) {
    $returnArray = array();
    $i = 0;
    foreach ($connections as $connection) {
        $returnArray[$i] =
            array($connection['METACONSOURCE'],$connection['METACONT
                ARGET'],0);
        $i++;
    }

    return $returnArray;
}

private function getStations($connections) {
    $returnArray = array();

    $count = count($connections);
    foreach ($connections as $connection) {

```

```

        if (
            !array_key_exists($connection['METACONSOURCE'],$returnArray)
        ) {
            $returnArray[$connection['METACONSOURCE']] = 0;
        }

        if (
            !array_key_exists($connection['METACONTARGET'],$returnArray)
        ) {
            $returnArray[$connection['METACONTARGET']] = 0;
        }
    }

    return $returnArray;
}

//gets the number of step from source node to target node
private function getDocPos($source,$target,$connections,$index)
{
    $ret = 0;
    for ($i = 0; $i<count($connections); $i++) {
        if ( $connections[$i][0] == $source && $connections[$i][2] == 0 ) {
            if ( $connections[$i][1] == $target ) {
                return 1;
            }
            $connections[$i][2] = 1;
            $sval = $this-
                >getDocPos($connections[$i][1],$target,$connections,$i);
            if ( $sval < $ret || ($ret == 0 && $sval > 0) ) {
                $ret = $sval+1;
            }
        }
    }
    return $ret;
}

public function getDocArcLocation($arcCode) {

    $page = isset($_POST['page'])?$_this->input->post('page'):1; // get
    the requested page
    $limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
    how many rows we want to have into the grid
    $sidx = isset($_POST['sidx'])?$_this->input->post('sidx'):'TIMEIN'; //
    get index row - i.e. user click to sort
    $sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
    get the direction

    $queryString = "1";

    if($_POST['_search']=="true"){
        $query = json_decode($_this->input->post('filters')); // get the
        query

        $operation = "";

        switch ($this->input->post("searchOper")) {
            case "eq":
                $operation = "=". $_this->input->post("searchString")."";
                break;
            case "ne":
                $operation = "<>". $_this->input->post("searchString")."";
                break;
            case "lt":
                $operation = "<". $_this->input->post("searchString")."";
                break;
            case "le":

```



```

$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%";
break;
}

$queryString = "(".$this->input->
post("searchField").$operation."";
}

if(!$sidx) $sidx = 'TIMEIN';

$count = count ($this->document_model-
>getDocArcLoc($sarcCode));

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model-
>getDocArcLocForTable($sidx, $sord, $start, $limit,
$queryString,$sarcCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $i;
$data->rows[$i]['cell'] =
array($typeList_item['STATIONNAME'],$typeList_item['TIMEIN'],$
typeList_item['TIMEOUT']);
$i++;
endforeach;

echo json_encode($data);
}

public function getDocLocation($barcode) {

$page = isset($_POST['page'])? $this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get
how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])? $this->input->post('sidx'):'TIMEIN'; //
get index row - i.e. user click to sort

```

```

$sord = isset($_POST['sord'])? $this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if($_POST['_search']=='true'){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = ". $this->input->post("searchString")."";
break;
case "ne":
$operation = " <> ". $this->input->post("searchString")."";
break;
case "lt":
$operation = " < ". $this->input->post("searchString")."";
break;
case "le":
$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%";
break;
}

$queryString = "(".$this->input->
post("searchField").$operation."";
}

if(!$sidx) $sidx = 'TIMEIN';

$count = count ($this->document_model->getDocLoc($barcode));

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->getDocLocForTable($sidx,
$sord, $start, $limit, $queryString,$barcode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

```

```

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $i;
$data->rows[$i]['cell'] =
array($typeList_item['STATIONCODE'],$typeList_item['STATIONN
AME'],$typeList_item['TIMEIN'],$typeList_item['TIMEOUT']);
$i++;
endforeach;

echo json_encode($data);
}

```

```

public function getTypeElements() {

```

```

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input-
>post('idx'):TYPENAME; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):ASC; //
get the direction

```

```

$queryString = "1";

```

```

if($_POST['_search']=="true"){
$query = json_decode($_this->input->post('filters')); // get the
query

```

```

$operation = "";

```

```

switch ($_this->input->post("searchOper")) {
case "eq":
$operation = " = '". $_this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $_this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $_this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $_this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $_this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $_this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $_this->input->post("searchString")."%";
break;
}

```

```

$queryString = "(" . $_this->input-
>post("searchField").$operation . ")";

```

```

}

if(!$sidx) $sidx = 'TYPENAME';

```

```

$count = count ($_this->document_model-
>get_docuTypeRowCount());

```

```

if ($count > 0) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

```

```

$typeList = $_this->document_model->get_docuType($sidx, $sord,
$start, $limit, $queryString);

```

```

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

```

```

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['TYPECODE'];
$data->rows[$i]['cell'] =
array($typeList_item['TYPECODE'],$typeList_item['TYPENAME'],$t
ypeList_item['TYPEDESC']);
$i++;
endforeach;

```

```

echo json_encode($data);
}

```

```

public function getParamElements($typeCode) {

```

```

$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input-
>post('idx'):TYPENAME; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):ASC; //
get the direction

```

```

$queryString = "1";

```

```

if($_POST['_search']=="true"){
$query = json_decode($_this->input->post('filters')); // get the
query

```

```

$operation = "";

```

```

switch ($_this->input->post("searchOper")) {
case "eq":
$operation = " = '". $_this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $_this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $_this->input->post("searchString")."'";
break;
case "le":

```

```

$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $this->input->post("searchString"). "%'";
break;
}

```

```

$queryString = "(" . $this->input->
post("searchField"). $operation. ")";
}

```

```

if(!$sidx) $sidx = 'TYPENAME';

```

```

$count = count ($this->document_model-
>get_documentTypeParam($typeCode));

```

```

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}

```

```

if ($page > $total_pages)
$page=$total_pages;

```

```

$start = 0;
if ($page > 0)
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

```

```

$typeParams = $this->document_model-
>get_documentTypeParam($typeCode, $sidx, $sord, $start,
$limit, $queryString);
$typeParamChoice = $this->document_model-
>get_documentTypeParamChoice();

```

```

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

```

```

$i=0;
foreach ($typeParams as $typeParams_item):
$temp = "";

```

```

foreach ($typeParamChoice as $typeParamChoice_item):
if($typeParamChoice_item['PARAMCODE'] ==
$typeParams_item['PARAMCODE']){
$temp .= $typeParamChoice_item['CHOICENAME']. "&";
}
endforeach;
if($temp!=""){
$temp = substr($temp, 0, -1);
}

```

```

$data->rows[$i]['id'] = $typeParams_item['PARAMORDER'];
$data->rows[$i]['cell'] =
array($typeParams_item['PARAMCODE'],$typeParams_item['PAR
AMORDER'],$typeParams_item['PARAMNAME'],$typeParams_ite
m['INPUTTYPE'],$temp,
$typeParams_item['COLSIZE'],$typeParams_item['DEFAULTVAL'],
$typeParams_item['PARAMDESC']);
$i++;
endforeach;

```

```

echo json_encode($data);
}

```

```

public function
getParamElementsPermission($stationCode,$typeCode) {

```

```

$page = isset($_POST['page'])? $this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get
how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])? $this->input-
>post('sidx'):'TYPENAME'; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])? $this->input->post('sord'):'ASC'; //
get the direction

```

```

$queryString = "1";

```

```

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

```

```

$operation = "";

```

```

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = ". $this->input->post("searchString")."";
break;
case "ne":
$operation = " <> ". $this->input->post("searchString")."";
break;
case "lt":
$operation = " < ". $this->input->post("searchString")."";
break;
case "le":
$operation = " <= ". $this->input->post("searchString")."";
break;
case "gt":
$operation = " > ". $this->input->post("searchString")."";
break;
case "ge":
$operation = " >= ". $this->input->post("searchString")."";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $this->input->post("searchString"). "%'";
break;
}

```

```

$queryString = "(" . $this->input-
>post("searchField"). $operation. ")";

```

```

}

if(!$sidx) $sidx = 'TYPENAME';

$count = count ($this->document_model-
>get_documentTypeParamStation($typeCode,$stationCode));

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages)
$page=$total_pages;

$start = 0;
if ($page > 0)
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeParams = $this->document_model-
>get_documentTypeParamStation($typeCode, $stationCode,
$sidx, $sord, $start , $limit, $queryString);
$typeParamChoice = $this->document_model-
>get_documentTypeParamChoice();

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeParams as $typeParams_item):
$temp = "";

foreach ($typeParamChoice as $typeParamChoice_item):
if($typeParamChoice_item['PARAMCODE'] ==
$typeParams_item['PARAMCODE']){
$temp .= $typeParamChoice_item['CHOICENAME']."&";
}
endforeach;
if($temp!=""){
$temp = substr($temp, 0, -1);
}

$data->rows[$i]['id'] = $typeParams_item['PARAMORDER'];
$data->rows[$i]['cell'] =
array($typeParams_item['PARAMCODE'],$typeParams_item['PAR
AMORDER'],$typeParams_item['PARAMNAME'],$typeParams_ite
m['INPUTTYPE'],$typeParams_item['DATATYPE'], $temp ,
$typeParams_item['COLSIZE'],$typeParams_item['DEFAULTVAL'],
$typeParams_item['PARAMDESC']);
$i++;
endforeach;

echo json_encode($data);
}

public function isInArray($arr,$elm) {
if (count($arr)>0) {
foreach ( $arr as $key=>$value ) {
if ( $key == $elm ) {
return true;
}
}
}
}

```

```

}
return false;
}

public function getStationElements() {
$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])?$_this->input-
>post('sidx'):'STATIONNAME'; // get index row - i.e. user click to
sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if ( $sidx == 'CONGEST' ) {
$count = count ($this->document_model-
>get_stationRowCount());

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model-
>get_stations("STATIONNAME", "", "0" , $count, "1");

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;

$dataRows;
$j=0;
$dccnt_holder = array();
$prev = -1;
foreach ($typeList as $typeList_item):
$typeList = $this->document_model-
>get_stationdoaccount($typeList_item['STATIONCODE']);
$dccnt = $typeList['DOCCOUNT'];
if ( $prev == -1 ) {
$prev = $dccnt;
}
if ( $prev != $dccnt ) {
$prev = $dccnt;
if ( $this->isInArray($dccnt_holder,$dccnt) ) {
$j=$dccnt_holder[$dccnt];
} else {
$j=0;
}
}
$dataRows[$dccnt.".".$j]['id'] = $typeList_item['STATIONCODE'];
$dataRows[$dccnt.".".$j]['cell'] =
array($typeList_item['STATIONCODE'],$typeList_item['STATIONN
AME'],$typeList['DOCCOUNT'],$typeList_item['STATIONDESC']);
$i++;
$j++;
$dccnt_holder[$dccnt] = $j;
endforeach;

```

```

if ( $sord == 'ASC' || $sord == 'asc' ) {
    krsort($dataRows);
} else {
    ksort($dataRows);
}

$i=0;
$j=0;
foreach ($dataRows as $dataRow){
    if ( $i >= $start && $i < $limit*$page ) {
        if($_POST['_search']=="true"){
            $query = json_decode($this->input->post('filters')); // get the query

            switch ($this->input->post("searchOper")) {
                case "eq":
                    if ( $this->input->post("searchString") == $dataRow['cell'][5] ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "ne":
                    if ( $this->input->post("searchString") != $dataRow['cell'][5] ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "lt":
                    if ( $dataRow['cell'][5] < $this->input->post("searchString") ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "le":
                    if ( $dataRow['cell'][5] <= $this->input->post("searchString") ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "gt":
                    if ( $dataRow['cell'][5] > $this->input->post("searchString") ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "ge":
                    if ( $dataRow['cell'][5] >= $this->input->post("searchString") ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "nu":
                    if ( !isset($dataRow['cell'][5]) ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "nn":
                    if ( isset($dataRow['cell'][5]) ) {
                        $data->rows[$j] = $dataRow;
                    }
                    break;
                case "cn":
                    $data->rows[$j] = $dataRow;
                    break;
            }
        } else {
            $data->rows[$j] = $dataRow;
        }
        $j++;
    }
    $i++;
}

} else {
    if($_POST['_search']=="true"){
        $query = json_decode($this->input->post('filters')); // get the query

        $operation = "";

        switch ($this->input->post("searchOper")) {
            case "eq":
                $operation = " = '". $this->input->post("searchString")."'";
                break;
            case "ne":
                $operation = " <> '". $this->input->post("searchString")."'";
                break;
            case "lt":
                $operation = " < '". $this->input->post("searchString")."'";
                break;
            case "le":
                $operation = " <= '". $this->input->post("searchString")."'";
                break;
            case "gt":
                $operation = " > '". $this->input->post("searchString")."'";
                break;
            case "ge":
                $operation = " >= '". $this->input->post("searchString")."'";
                break;
            case "nu":
                $operation = " IS NULL";
                break;
            case "nn":
                $operation = " IS NOT NULL";
                break;
            case "cn":
                $operation = " LIKE '%" . $this->input->post("searchString"). "%'";
                break;
        }

        $queryString = "(" . $this->input->post("searchField"). $operation . ")";
    }

    if(!$sidx) $sidx = 'STATIONNAME';

    $count = count ($this->document_model->get_stationRowCount());

    if( $count > 0 ) {
        $total_pages = ceil($count/$limit);
    } else {
        $total_pages = 0;
    }
    if ($page > $total_pages) $page=$total_pages;
    $start = $limit*$page - $limit; // do not put $limit*( $page - 1)

    $typeList = $this->document_model->get_stations($sidx, $sord, $start , $limit, $queryString);

    $data->total = $total_pages;
    $data->page = $page;
    $data->records = $count;

    $i=0;
    foreach ($typeList as $typeList_item):
        $data->rows[$i]['id'] = $typeList_item['STATIONCODE'];

```

```

$typeList = $this->document_model-
>get_stationdoccount($typeList_item['STATIONCODE']);

$data->rows[$i]['cell'] =
array($typeList_item['STATIONCODE'],$typeList_item['STATIONN
AME'],$typeList['DOCCOUNT'],$typeList_item['STATIONDESC']);
$i++;
endforeach;
}

echo json_encode($data);
}

public function getCompanyElements() {
$page = isset($_POST['page'])? $this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['sidx'])? $this->input-
>post('sidx'):COMPANYNAME; // get index row - i.e. user click to
sort
$sord = isset($_POST['sord'])? $this->input->post('sord'):ASC; //
get the direction

$queryString = "1";

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = '". $this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%";
break;
}

$queryString = "(".$this->input-
>post("searchField").$operation.")";
}

```

```

if(!$idx) $idx = 'COMPANYNAME';

$count = count ($this->document_model->get_companies());

if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model-
>get_companies_with_limits($idx, $sord, $start, $limit,
$queryString);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['COMPANYNAME'];
$data->rows[$i]['cell'] =
array($typeList_item['COMPANYCODE'],$typeList_item['COMPAN
YNAME'],$typeList_item['EMAIL'],$typeList_item['PHONE'],$typeL
ist_item['ADDRESS'],$typeList_item['COMPANYDESC']);
$i++;
endforeach;

echo json_encode($data);
}

public function getDataElements() {
$page = isset($_POST['page'])? $this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['sidx'])? $this->input-
>post('sidx'):DATATYPENAME; // get index row - i.e. user click to
sort
$sord = isset($_POST['sord'])? $this->input->post('sord'):ASC; //
get the direction

$queryString = "1";

if($_POST['_search']=="true"){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = '". $this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $this->input->post("searchString")."'";
break;

```

```

case "le":
$operation = " <= '". $this->input->post("searchString")."''";
break;
case "gt":
$operation = " > '". $this->input->post("searchString")."''";
break;
case "ge":
$operation = " >= '". $this->input->post("searchString")."''";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%'";
break;
}

$queryString = "(".$this->input-
->post("searchField").$operation.")";
}

if(!$sidx) $sidx = 'DATATYPENAME';

$count = count ($this->document_model->get_dataTypeCount());

if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model->get_dataTypeList($sidx,
$sord, $start , $limit, $queryString);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['DATATYPECODE'];
$data->rows[$i]['cell'] =
array($typeList_item['DATATYPECODE'],$typeList_item['DATATYP
ENAME']);
$i++;
endforeach;

echo json_encode($data);
}

public function getPermissions($stationCode,$typeCode) {
$page = isset($_POST['page'])? $this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])? $this->input->post('rows'):20; // get
how many rows we want to have into the grid
$sidx = isset($_POST['sidx'])? $this->input-
->post('sidx'):'PARAMNAME'; // get index row - i.e. user click to
sort

```

```

$sord = isset($_POST['sord'])? $this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if($_POST['_search']=='true'){
$query = json_decode($this->input->post('filters')); // get the
query

$operation = "";

switch ($this->input->post("searchOper")) {
case "eq":
$operation = " = '". $this->input->post("searchString")."''";
break;
case "ne":
$operation = " <> '". $this->input->post("searchString")."''";
break;
case "lt":
$operation = " < '". $this->input->post("searchString")."''";
break;
case "le":
$operation = " <= '". $this->input->post("searchString")."''";
break;
case "gt":
$operation = " > '". $this->input->post("searchString")."''";
break;
case "ge":
$operation = " >= '". $this->input->post("searchString")."''";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%". $this->input->post("searchString")."%'";
break;
}

$queryString = "(".$this->input-
->post("searchField").$operation.")";
}

if(!$sidx) $sidx = 'PARAMNAME';

$count = count ($this->document_model-
->get_PermissionCount($stationCode,$typeCode));

if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;

$start = 0;
if ($page > 0)
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $this->document_model-
->get_permissionList($stationCode,$typeCode,$sidx, $sord, $start ,
$limit, $queryString);

```

```

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['PARAMCODE'];
$data->rows[$i]['cell'] =
array($typeList_item['PARAMCODE'],$typeList_item['PARAMNAM
E'],$typeList_item['PERTYPE']);
$i++;
endforeach;

echo json_encode($data);
}

public function getUserList() {
$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input-
>post('idx'):'USERNAME'; // get index row - i.e. user click to sort
$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
get the direction

if ( $idx == "STATIONCODE" ) {
$idx = "STATION.STATIONCODE";
}

$queryString = "1";

if($_POST['_search']=="true"){
$query = json_decode($_this->input->post('filters')); // get the
query

$operation = "";

switch ($_this->input->post("searchOper")) {
case "eq":
$operation = " = '". $_this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $_this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $_this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $_this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $_this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $_this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $_this->input->post("searchString")."%";

```

```

break;
}

$queryString = "(" . $_this->input-
>post("searchField").$operation."";
}

if(!$idx) $idx = 'USERNAME';

if($_this->session->userdata('usertype') == "admin"){
$count = count ($_this->document_model->get_usersRowCount());
} else {
$count = count ($_this->document_model-
>get_usersRowCountStation($_this->session-
>userdata('stationcode')));
}
$count = $count - 1;
if ( $count < 0 ) {
$count = 0;
}

if( $count > 0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

if($_this->session->userdata('usertype') == "admin"){
$typeList = $_this->document_model->get_users($idx, $sord,
$start, $limit, $queryString);
} else {
$typeList = $_this->document_model->get_users_station($idx,
$sord, $start, $limit, $queryString, $_this->session-
>userdata('stationcode'));
}
$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

$i=0;
foreach ($typeList as $typeList_item):
if ($typeList_item['USERID']!= $_this->session->userdata('userid')) {
$data->rows[$i]['id'] = $typeList_item['USERID'];
$data->rows[$i]['cell'] =
array($typeList_item['USERID'],$typeList_item['USERNAME'],$typ
eList_item['PASSWORD'],$typeList_item['USERTYPE'],$typeList_ite
m['STATIONNAME'],$typeList_item['FIRSTNAME'],$typeList_item['
LASTNAME'],$typeList_item['EMAIL'],$typeList_item['PHONE'],$ty
peList_item['ADDRESS']);
$i++;
}
endforeach;

echo json_encode($data);
}

public function getUserListStation($stationCode) {
$page = isset($_POST['page'])?$_this->input->post('page'):1; // get
the requested page
$limit = isset($_POST['rows'])?$_this->input->post('rows'):20; // get
how many rows we want to have into the grid
$idx = isset($_POST['idx'])?$_this->input-
>post('idx'):'USERNAME'; // get index row - i.e. user click to sort

```



```

$sord = isset($_POST['sord'])?$_this->input->post('sord'):'ASC'; //
get the direction

$queryString = "1";

if($_POST['_search']=="true"){
$query = json_decode($_this->input->post('filters')); // get the
query

$operation = "";

switch ($_this->input->post("searchOper")) {
case "eq":
$operation = " = '". $_this->input->post("searchString")."'";
break;
case "ne":
$operation = " <> '". $_this->input->post("searchString")."'";
break;
case "lt":
$operation = " < '". $_this->input->post("searchString")."'";
break;
case "le":
$operation = " <= '". $_this->input->post("searchString")."'";
break;
case "gt":
$operation = " > '". $_this->input->post("searchString")."'";
break;
case "ge":
$operation = " >= '". $_this->input->post("searchString")."'";
break;
case "nu":
$operation = " IS NULL";
break;
case "nn":
$operation = " IS NOT NULL";
break;
case "cn":
$operation = " LIKE '%" . $_this->input->post("searchString")."%";
break;
}

$queryString = "(" . $_this->input-
>post("searchField").$operation . ")";
}

if(!isset($sidx) || $sidx != 'USERNAME');

$count = count ($_this->document_model-
>get_usersRowCountStation($stationCode));

if( $count >0 ) {
$total_pages = ceil($count/$limit);
} else {
$total_pages = 0;
}
if ($page > $total_pages) $page=$total_pages;
$start = $limit*$page - $limit; // do not put $limit*( $page - 1)

$typeList = $_this->document_model->get_stationUsers($sidx,
$sord, $start , $limit, $queryString,$stationCode);

$data->total = $total_pages;
$data->page = $page;
$data->records = $count;

```

```

$i=0;
foreach ($typeList as $typeList_item):
$data->rows[$i]['id'] = $typeList_item['USERID'];
$data->rows[$i]['cell'] =
array($typeList_item['USERID'],$typeList_item['USERNAME'],$typ
eList_item['USERTYPE'],$typeList_item['FIRSTNAME'],$typeList_it
em['LASTNAME']);
$i++;
endforeach;

echo json_encode($data);
}

public function addUser() {
$username=$_this->input->post('USERNAME');
$userType=$_this->input->post('USERTYPE');
$password=$_this->input->post('PASSWORD');
$stationCode=$_this->input->post('STATIONCODE');
$firstName=$_this->input->post('FIRSTNAME');
$lastName=$_this->input->post('LASTNAME');
$email=$_this->input->post('EMAIL');
$phone=$_this->input->post('PHONE');
$address=$_this->input->post('ADDRESS');

echo $_this->document_model-
>add_user($username,$userType,$password,$stationCode,$firstn
ame,$lastName,$email,$phone,$address);
}

public function saveNewDocuType() {
$name = $_this->input->post('name');
$description = $_this->input->post('desc');

echo $_this->document_model-
>add_docuTypeMetaData($name,$description);
}

public function saveForm() {
$barCode = $_this->input->post('barCode');
if( $_this->session->userdata('userType') == "admin" ) {
$params = $_this->document_model-
>get_documentParaWOLimiters($barCode);
} else {
$params = $_this->document_model-
>get_documentParaWOLimiters($barCode,$_this->session-
>userdata('stationcode'));
}
$queryDataContainer = array();
foreach ($params as $param) {
if( $_this->session->userdata('userType') != "admin" ) {
if ( $param['PERTYPE']=="rw" ) {
$queryDataContainer[$param['METAPARAMCODE']] = $_this-
>input->post($param['METAPARAMCODE']);
}
} else {
$queryDataContainer[$param['METAPARAMCODE']] = $_this-
>input->post($param['METAPARAMCODE']);
}
}

echo $_this->document_model-
>save_form($queryDataContainer,$_this->session-
>userdata('stationcode'));
}

```

```

public function transferForm() {
$barCode = $this->input->post('barCode');
$stationCode = $this->input->post('stationCode');

if ( $stationCode == 0 || $stationCode == '0' ) {
echo $this->document_model->archiveDoc($barCode,$this->session->userdata('stationcode'));

$reclist = $this->input->post('reclist');
$this->load->library('email');
foreach ( $reclist as $rec ) {
$this->email->from($this->session->userdata('email'), $this->session->userdata('username'));
$this->email->to($rec);
$this->email->subject($subj);
$this->email->message($msg);
$this->email->send();
echo $this->email->print_debugger();
}
} else {
echo $this->document_model->transfer_form($barCode,$stationCode,$this->session->userdata('stationcode'));
}
}

public function deleteDoc() {
$barCode = $this->input->post('barcode');
echo $this->document_model->deleteDoc($barCode,$this->session->userdata('stationcode'));
}

public function saveNewDoc() {
$title = $this->input->post('title');
$description = $this->input->post('desc');
$typeCode = $this->input->post('typeCode');
$companies = $this->input->post('companylistvals');

echo $this->document_model->add_document($title,$description,$typeCode,$this->session->userdata('userid'),$companies);
}

public function sendMail() {
$reclist = $this->input->post('reclist');
$subj = $this->input->post('subj');
$msg = $this->input->post('msg');
$this->load->library('email');
foreach ( $reclist as $rec ) {
$this->email->from($this->session->userdata('email'), $this->session->userdata('username'));
$this->email->to($rec);
$this->email->subject($subj);
$this->email->message($msg);
$this->email->send();
echo $this->email->print_debugger();
}
}

public function assignAppSave() {
$userId = $this->input->post('userId');
$companies = $this->input->post('companylistvals');

echo $this->document_model->assignApp($userId,$companies);
}

public function saveProfile() {
$username = $this->input->post('username');
$password = $this->input->post('password');
$password2 = $this->input->post('password2');
$firstname = $this->input->post('firstname');
$lastname = $this->input->post('lastname');
$email = $this->input->post('email');
$phone = $this->input->post('phone');
$address = $this->input->post('address');
$theme = $this->input->post('theme');
$isComplete = true;
$errorMsg = "Fill up: ";
if($username==""){
$errorMsg .= "User Name, ";
$isComplete = false;
}
if($password==""){
$errorMsg .= "Password, ";
$isComplete = false;
}
if($password2==""){
$errorMsg .= "Retyped Password, ";
$isComplete = false;
}
if($firstname==""){
$errorMsg .= "First Name, ";
$isComplete = false;
}
if($lastname==""){
$errorMsg .= "Last Name, ";
$isComplete = false;
}
if($isComplete){
if($password!=$password2){
echo "<font color='\"#ff0000\"'>Passwords did not match<br /></font>";
} else {
echo $this->document_model->edit_profile($username,$password,$firstname,$lastname,$email,$phone,$address,$theme);
}
} else {
echo "<font color='\"#ff0000\"'>\".errorMsg.\"</font>";
}
}

public function saveNewParameter() {
$name = $this->input->post('PARAMNAME');
$inputtype=$this->input->post('INPUTTYPE');
$options=$this->input->post('options');
$default=$this->input->post('DEFAULTVAL');
$description=$this->input->post('description');
$typeCode=$this->input->post('typecode');
$order=$this->input->post('PARAMORDER');
$colSize=$this->input->post('COLSIZE');

echo $this->document_model->add_parameter($name,$options,$default,$description,$typeCode,$order,$colSize,$inputtype);
}

public function saveNewStation() {
$name=$this->input->post('STATIONNAME');
$desc=$this->input->post('STATIONDESC');
}

```

```

echo $this->document_model->add_station($name,$desc);
}

public function saveNewCompany() {
$companyCode=$this->input->post('COMPANYCODE');
$name=$this->input->post('COMPANYNAME');
$email=$this->input->post('EMAIL');
$phone=$this->input->post('PHONE');
$add=$this->input->post('ADDRESS');
$desc=$this->input->post('COMPANYDESC');

echo $this->document_model-
>add_company($companyCode,$name,$email,$phone,$add,$desc);
}

public function saveNewDataType() {
$name=$this->input->post('DATATYPENAME');

echo $this->document_model->add_dataType($name);
}

public function savePermission() {
$stationCode=$this->input->post('stationCode');
$paramsCode=$this->input->post('paramCode');
$perType=$this->input->post('PERTYPE');

echo $this->document_model-
>add_permission($stationCode,$paramsCode,$perType);
}

public function editUser() {
$userid=$this->input->post('userid');
$username=$this->input->post('USERNAME');
$userType=$this->input->post('USERTYPE');
$password=$this->input->post('PASSWORD');
$stationCode=$this->input->post('STATIONCODE');
$firstName=$this->input->post('FIRSTNAME');
$lastName=$this->input->post('LASTNAME');
$email=$this->input->post('EMAIL');
$phone=$this->input->post('PHONE');
$address=$this->input->post('ADDRESS');

echo $this->document_model-
>edit_user($userid,$username,$userType,$password,$stationCode,$firstName,$lastName,$email,$phone,$address);
}

public function editParameter() {
$name = $this->input->post('PARAMNAME');
$inputType=$this->input->post('INPUTTYPE');
$options=$this->input->post('options');
$default=$this->input->post('DEFAULTVAL');
$description=$this->input->post('description');
$typeCode=$this->input->post('typecode');
$paramsCode=$this->input->post('PARAMCODE');
$order=$this->input->post('PARAMORDER');
$colSize=$this->input->post('COLSIZE');

echo $this->document_model-
>edit_parameter($name,$options,$default,$description,$typeCode,$paramsCode,$order,$colSize,$inputType);
}

public function editStation() {
$stationCode=$this->input->post('stationCode');

```

```

$name=$this->input->post('stationName');
$name=$this->input->post('STATIONNAME');
$desc=$this->input->post('STATIONDESC');

echo $this->document_model-
>edit_station($stationCode,$name,$desc);
}

public function editCompany() {
$companyCode=$this->input->post('COMPANYCODE');
$name=$this->input->post('COMPANYNAME');
$email=$this->input->post('EMAIL');
$phone=$this->input->post('PHONE');
$add=$this->input->post('ADDRESS');
$desc=$this->input->post('COMPANYDESC');

echo $this->document_model-
>edit_company($companyCode,$name,$email,$phone,$add,$desc);
}

public function editDataType() {
$dataTypeCode=$this->input->post('datatypecode');
$name=$this->input->post('DATATYPENAME');

echo $this->document_model-
>edit_dataType($dataTypeCode,$name);
}

public function deleteParameter() {
$typeCode=$this->input->post('typecode');
$paramsCode=$this->input->post('paramcode');
$order=$this->input->post('order');

echo $this->document_model-
>delete_parameter($paramsCode,$typeCode,$order);
}

public function deleteUser() {
$userid=$this->input->post('userid');

echo $this->document_model->delete_user($userid);
}

public function deleteType() {
$typeCode=$this->input->post('typecode');

echo $this->document_model->delete_type($typeCode);
}

public function deleteStation() {
$stationCode=$this->input->post('stationcode');

echo $this->document_model->delete_station($stationCode);
}

public function deleteCompany() {
$companyCode=$this->input->post('companycode');

echo $this->document_model->delete_company($companyCode);
}

public function deleteDataType() {
$dataTypeCode=$this->input->post('datatypecode');

```

```

echo $this->document_model-
>delete_dataType($dataTypeCode);
}

public function deletePermission() {
$paramsCode=$this->input->post('paramCode');
$stationCode=$this->input->post('stationCode');

echo $this->document_model-
>delete_permission($paramCode,$stationCode);
}
}

```

iii. views/document/newDoc.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:20px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:100px;">Title</TD><TD><INPUT TYPE="TEXT"
NAME="TYPENAME" ID="TYPENAME" VALUE="" size=42></TD>
<TD rowspan="3" style="padding:0px 0px 0px 5px;">
<select id="companylist" size="4" style="font-
size:12px;width:300px;">
<?php
foreach ($companies as $company):
echo "<option
value=\"\". $company['COMPANYCODE'].\"\">\". $company['COMPA
NYNAME'].\"\"</option>\n\";
endforeach;
?>
</select>
<center style="padding:2px;">
<input type="button" value="<<"
onclick="transfer('companylistvals','companylist');"/>
<input type="button" value=">>"
onclick="transfer('companylist','companylistvals');"/>
</center>
<select id="companylistvals" size="4" style="font-
size:12px;width:300px;">

</select>
</TD>
</TR>
<TR>
<TD style="width:100px;">Description</TD><TD><TEXTAREA
TYPE="TEXT" NAME="TYPEDESC" ID="TYPEDESC" cols="33"
rows="3"></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" style="padding:0px 0px 0px 0px;">
<select id="typeList" size="5" style="font-size:12px;width:386px;"
onchange="showDetails($('#typeList').val());">
<?php
foreach ($docTypes as $docType):
echo "<option
value=\"\". $docType['TYPECODE'].\"\">\". $docType['TYPENAME'].\"\"<
/option>\n\";
endforeach;
?>
</select>
</TD>
</TR>

```

```

<TR>
<TD colspan="3" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>

<div name="typeDetails" id="typeDetails" class="exp"
style="left:220px;top:260px;width:720px;">
<div id="details">
<ul>
<li><a href="#tab1" style="font-size: 12px;">format</a></li>
<li><a href="#tab2" style="font-size: 12px;">instances</a></li>
</ul>
<div id="tab1" style="padding:5px 40px 10px 10px">
<div id="form" style="font-size:12px;"></div>
</div>
<div id="tab2" style="padding:2px 0px 0px 0px">
<table id="instance"></table>
<div id="pager2"></div>
</div>
</div>
</div>
</div>

<script type="text/javascript">

$(function() {
$("#typeMeta, #typeDetails").css("display", "none");
$("#typeMeta, #typeDetails").fadeIn(1000);
$("input:submit, button").button();

$("#typeSaveMeta").button().click(function() {
if ($("#TYPENAME").val() == null || ($("#TYPENAME").val() == "" ||
!(/^[a-zA-Z0-9]+$/).test($("#TYPENAME").val())) {
alert("Invalid title");
} else if ($("#TYPEDESC").val() == null || ($("#TYPEDESC").val() ==
"" ) {
alert("Invalid description");
} else if ($("#typeList").val() == null || ($("#typeList").val() == "" ) {
alert("Please select a type");
} else {
if(confirm('Are you sure you want to save?')){
var complist = [];
$("#companylistvals option").each(function() {
complist.push($(this).val());
})

$.ajax({
type: "POST",
url: "<?php echo
\"\".base_url('index.php/document/saveNewDoc').\"\""; ?>,
data: { title : ($("#TYPENAME").val() ), desc : ($("#TYPEDESC").val() ),
typeCode : ($("#typeList").val()),
companylistvals:complist
},
success: function(msg){
if( msg=="failed" || msg==" " || msg==null ){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").append("Add Failed!<hr /><br />");
} else {
alert("Document Saved");
window.location.href = "<?php echo
base_url('index.php/document/viewDocs');?>";
}
}
}

```

```

}
});
}
});
});

function showDetails(selected) {
$("#instance").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getTypePathInstances")."^\\"
;?>+selected}).trigger("reloadGrid");

$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/getDocuTypeForm")."^\\";?>,
data: { typeCode : selected },
success: function(msg){
//alert(msg);
$("#form").empty();
$("#form").show();
$("#form").append(msg);
}
});
};

$("#details").tabs({
collapsible: true,
selected: -1
});

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#instance").jqGrid({

```

```

url:<?php echo
"\\".base_url("index.php/document/getTypePathInstances")."^\\"
;?>,
datatype: "json",
mtype : "post",
colNames:['Barcode','Title', 'Location','Duration','Status'],
colModel:[
{name:'BARCODE',index:'BARCODE', width:0,hidden:true},
{name:'TITLE',index:'TITLE', width:215,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url("index.php/document/viewDoc");?>/"+rowObject[0]+\"
>" + cellvalue + "</a>";
}
},
{name:'STATIONNAME',index:'STATIONNAME',
width:150,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:140,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:185,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
],
rowNum: 10,
height: 200,
rowList:[10,20,30],
pager: '#pager2',
sortname: 'TITLE',
viewrecords: true,
sortorder: "desc",
caption: "Type Instances"
});

jQuery("#instance").jqGrid('navGrid','#pager2',
{
edit:false,add:false,del:false
});//options
{
};// edit options
{
};// add options
{
};// del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
};// search options
});

$("#ui-jqgrid-titlebar").hide();

function transfer(sel1,sel2){
var elSel = document.getElementById(sel1);
var i;
var slctd;
for (i = elSel.length - 1; i>=0; i--) {
if (elSel.options[i].selected) {
slctd = i;
}
}
var elSel2 = document.getElementById(sel2);
var elOptNew = document.createElement('option');

```

```

elOptNew.text = elSel.options[slctd].text;
elOptNew.value = elSel.options[slctd].value;
try {
elSel2.add(elOptNew,elSel2.options[0]); // standards compliant;
doesn't work in IE
}
catch(ex) {
elSel2.add(elOptNew,0); // IE only
}
elSel.remove(slctd);
}
</script>

```

iv. views/docuype/adminDoc.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:80px;padding:5px;width:6
45px;background-color:white;border:2px solid #ddd; ;">
<table id="docuype"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

```

```

<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:33em">
<div id="docViewSel">
<input type="radio" id="radio1" name="radio"
checked="checked"/><label for="radio1">Monitoring</label>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">Edit</label>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">Parameters</label>
</div>
</div>

```

```

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});

$("#docViewSel").buttonset();

$("#radio1").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditMeta").\\"";?>
});
$("#radio2").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/setupworkflow").\\"";?>
});
$("#radio3").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditParam").\\"";?>
});

```

```

jQuery("#docuype").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getDocuments").\\"";?>,
datatype: "json",
mtype : "post",

```

```

colNames:['Barcode','Title','Type','Location','Progress','Date
Recieved','Days in Progress'],
colModel:[
{name:'BARCODE',index:'BARCODE', width:70},
{name:'TYPENAME',index:'TYPENAME', width:200},
{name:'LOCATION',index:'LOCATION', width:430,sortable:false},
{name:'PROGRESS',index:'PROGRESS', width:430,sortable:false},
{name:'DATERECIEVED',index:'DATERECIEVED',
width:430,sortable:false},
{name:'DAYSINPROG',index:'DAYSINPROG',
width:430,sortable:false}
],
rowNum: 20,
height: 300,
rowList:[10,20,30],
pager: '#pager',
sortname: 'TYPENAME',
viewrecords: true,
sortorder: "desc",
caption: "Documents",
});

```

```

jQuery("#docuype").jqGrid('navGrid','#pager',
{
editfunc: function (rowid) {
window.location.href = "<?php echo
base_url("index.php/document/typeEditMeta");?>"+"/"+rowid;
},
addfunc: function () {
window.location.href = "<?php echo
base_url("index.php/document/typeNewMeta");?>";
}
}, //options
{
}, // edit options
{
}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteType").\\"";?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete this type?",
delData: {
typecode:
function() {
var sel_id = jQuery("#docuype").jqGrid('getGridParam', 'selrow');
var value = jQuery("#docuype").jqGrid('getCell', sel_id,
'TYPECODE');
return value;
}
},
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{

```

```

sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
</script>

```

v. views/docutype/adminEditForm.php

```

<div id="transfer" class="exp"
style="left:220px;top:20px;padding:5px;width:700px;">
<font size="6" color="#b8b8b7"><?php echo
$documentMeta['TITLE'];?></font><br /><br />
<table>
<tr>
<td style="padding:20px;width:100px;">
<button id="transferdoc" style="font-size: 12px;">send</button>
</td>
<td>
<SELECT ID="stationOptions" style="width:530px;" size=4 />
<?php
foreach ( $nextstations as $nextstation ) {
echo "<option value=\"\". $nextstation['METACONTARGET'].\" \"\
>\". $nextstation['STATIONNAME'].\" -
\". $nextstation['METACONDESCR'].\"</option>\";
}
?>
</SELECT>
</td>
</tr>
</table>
</div>
<div id="email" class="exp"
style="left:220px;top:185px;padding:5px;width:700px;">
<table>
<tr>
<td style="padding:20px;width:100px;" rowspan=6>
<button id="sendMail" style="font-size: 12px;">send
mail</button>
</td>
<td style="padding:5px 5px 5px 5px;width:100px;">
<tr>
<td>To</td>
<td>
<INPUT TYPE="TEXT" NAME="TO" ID="TO" VALUE="<?php
foreach ($mailRecipients as $mailRecipient) {
echo " ". $mailRecipient['EMAIL'].\";\";
}
?>" size=64 READONLY>
</td>
</tr>
</td>
<td style="padding:0px 5px 5px 5px;width:100px;">
<tr>
<td>Subject</td>
<td>
<INPUT TYPE="TEXT" NAME="SUBJECT" ID="SUBJECT" VALUE=""
size=64>
</td>
</tr>
</td>

```

```

<td style="padding:0px 5px 5px 5px;width:100px;">
<tr>
<td>Message</td>
<td>
<TEXTAREA TYPE="TEXT" NAME="MESSAGE" ID="MESSAGE"
cols="50" rows="2"></TEXTAREA>
</td>
</tr>
</td>
</tr>
</table>
</div>
<div id="typeMeta" class="exp"
style="left:220px;top:330px;padding:5px;width:700px;">
<div id="saveMetaStatus"></div>
<div style="padding:10px;">
<table>
<?php
foreach ($params as $param) {
?>
<tr>
<td style="padding:5px 10px 5px 10px;">
<?php
echo $param['PARAMNAME'];
?>
</td>
<td>
<?php
$inputType = $param['INPUTTYPE'];
if ($inputType=='text') {
?>
<INPUT TYPE="TEXT" NAME="<?php echo
$param['METAPARAMCODE'];?>" ID="<?php echo
$param['METAPARAMCODE'];?>" VALUE="<?php echo
$param['STRINGVAL'];?>" />
<?php
} else if ($inputType=='select') {
?>
<SELECT NAME="<?php echo $param['METAPARAMCODE'];?>"
ID="<?php echo $param['METAPARAMCODE'];?>" />
<?php
foreach ($selectOptions[$param['METAPARAMCODE']] as
$selectOption) {
echo "<option value=\"\". $selectOption['METACHOICENAME'].\" \"\
\";
if ( $selectOption['METACHOICENAME'] == $param['STRINGVAL'] )
{
echo "selected";
}
echo " >\". $selectOption['METACHOICENAME'].\"</option>\";
}
?>
</SELECT>
<?php
} else if ($inputType=='checkbox') {
?>
<INPUT TYPE="checkbox" NAME="<?php echo
$param['METAPARAMCODE'];?>" ID="<?php echo
$param['METAPARAMCODE'];?>" <?php if (
isset($param['STRINGVAL']) && $param['STRINGVAL'] == 'true' ) {
echo "checked=\"checked\"";?> } ?> />
<?php
} else if ($inputType=='radio') {
foreach ($selectOptions[$param['METAPARAMCODE']] as
$selectOption) {

```



```

var elSel = document.getElementById(sel1);
var i;
var slctd;
for (i = elSel.length - 1; i>=0; i--) {
if (elSel.options[i].selected) {
slctd = i;
}
}
var elSel2 = document.getElementById(sel2);
var elOptNew = document.createElement('option');
elOptNew.text = elSel.options[slctd].text;
elOptNew.value = elSel.options[slctd].value;
try {
elSel2.add(elOptNew,elSel2.options[0]); // standards compliant;
doesn't work in IE
}
catch(ex) {
elSel2.add(elOptNew,0); // IE only
}
elSel.remove(slctd);
}
</script>

```

vii. views/docutype/editForm.php

```

<div id="transfer" class="exp"
style="left:220px;top:20px;padding:5px;width:700px;">
<font size="6" color="#b8b8b7"><?php echo
$documentMeta['TITLE'];?></font><br /><br />
<table>
<tr>
<td style="padding:20px;width:100px;">
<button id="transferdoc" style="font-size: 12px;">send</button>
</td>
<td>
<SELECT ID="stationOptions" style="width:530px;" size=4 />
<?php
foreach ( $nextstations as $nextstation ) {
echo "<option value=\"\" . $nextstation['METACONTARGET'].\"\"
>\" . $nextstation['STATIONNAME'].\" -
\" . $nextstation['METACONDESCR'].\"</option>\";
}
?>
</SELECT>
</td>
</tr>
</table>
</div>
<div id="email" class="exp"
style="left:220px;top:185px;padding:5px;width:700px;">
<table>
<tr>
<td style="padding:20px;width:100px;" rowspan=6>
<button id="sendMail" style="font-size: 12px;">send
mail</button>
</td>
<td style="padding:5px 5px 5px 5px;width:100px;">
<tr>
<td>To</td>
<td>
<INPUT TYPE="TEXT" NAME="TO" ID="TO" VALUE="<?php
foreach ($mailRecipients as $mailRecipient) {
echo " \" . $mailRecipient['EMAIL'].\";\";
}

```

```

?>" size=64 READONLY>
</td>
</tr>
</td>
<td style="padding:0px 5px 5px 5px;width:100px;">
<tr>
<td>Subject</td>
<td>
<INPUT TYPE="TEXT" NAME="SUBJECT" ID="SUBJECT" VALUE=""
size=64>
</td>
</tr>
</td>
<td style="padding:0px 5px 5px 5px;width:100px;">
<tr>
<td>Message</td>
<td>
<TEXTAREA TYPE="TEXT" NAME="MESSAGE" ID="MESSAGE"
cols="50" rows="2"></TEXTAREA>
</td>
</tr>
</td>
</tr>
</table>
</div>
<div id="typeMeta" class="exp"
style="left:220px;top:330px;padding:5px;width:700px;">
<div id="saveMetaStatus"></div>
<div style="padding:10px;">
<table>
<?php
foreach ($params as $param) {
$isDisabled = ($param['PERTYPE']=="r");
?>
<tr>
<td style="padding:5px 10px 5px 10px;">
<?php
echo $param['PARAMNAME'];
if (!$isDisabled) {
echo "<font color='red'> *</font>\";
}
?>
</td>
<td>
<?php
$inputType = $param['INPUTTYPE'];
if ($inputType=="text") {
?>
<INPUT TYPE="TEXT" NAME="<?php echo
$param['METAPARAMCODE'];?>" ID="<?php echo
$param['METAPARAMCODE'];?>" VALUE="<?php echo
$param['STRINGVAL'];?>" <?php if ($isDisabled){ echo
"READONLY";}?> />
<?php
} else if ($inputType=="select") {
?>
<SELECT NAME="<?php echo $param['METAPARAMCODE'];?>"
ID="<?php echo $param['METAPARAMCODE'];?>" <?php if
($isDisabled){ echo "disabled";}?> />
<?php
foreach ($selectOptions[$param['METAPARAMCODE']] as
$selectOption) {
echo "<option value=\"\" . $selectOption['METACHOICENAME'].\"\"
\";
if ( $selectOption['METACHOICENAME'] == $param['STRINGVAL'] )
{

```



```

<div id="one"
style="position:absolute;left:220px;top:60px;padding:5px;width:4
15px;background-color:white;border:2px solid #ddd; ;">
<table id="permissions"></table>
<div id="pager1"></div>
<div id="saveParamStatus"></div>
</div>

<div id="two"
style="position:absolute;left:655px;top:10px;padding:5px;width:4
20px;background-color:white;border:2px solid #ddd; ;">
<table id="parameters"></table>
<div id="pager2"></div>
</div>

<div id="three"
style="position:absolute;left:655px;top:280px;padding:5px;width:
420px;background-color:white;border:2px solid #ddd; ;">
<table id="docutypes"></table>
<div id="pager3"></div>
</div>

<script type="text/javascript">
$(function() {
$("#one,#two,#three,#four").css("display", "none");
$("#one,#two,#three,#four").fadeIn(1000);
$("input:submit, button").button();
});

jQuery("#permissions").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getPermissions/$stationCode
")."/0\\"", ?>,
datatype: "json",
mtype : "post",
colNames:['parameter code', 'Parameter Name', 'Permission
Type'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:true,hidden:true},
{name:'PARAMNAME',index:'PARAMNAME', width:200,
sorttype:"text",editable:false},
{name:'PERTYPE',index:'PERTYPE', width:200,
sorttype:"text",editable:true,edittype:"select",editoptions:{value:
"r:read,rw:read/write"}}
],
rowNum:20,
height: 400,
rowList:[10,20,30],
pager: '#pager1',
sortname: 'PARAMNAME',
viewrecords: true,
sortorder: "asc",
caption: "Permissions"
});

jQuery("#permissions").jqGrid('navGrid','#pager1',
{edit:false,add:false,del:true}, //options
{}), // edit options
{}), // add options
{
url: <?php echo
"\\".base_url("index.php/document/deletePermission")."\\"", ?>,
reloadAfterSubmit:true,
type: "POST",

```

```

width:300,
msg: "Are you sure you want to delete data permission?",
delData: {
paramCode: function() {
var sel_id = jQuery("#permissions").jqGrid('getGridParam',
'selrow');
var value = jQuery("#permissions").jqGrid('getCell', sel_id,
'PARAMCODE');
return value;
},
stationCode: <?php echo "\\".$stationCode.\\"";?>
},
afterComplete:function(msg){
$("#parameters").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getParamElementsPermissio
n/$stationCode")."/\\"", ?>+jQuery("#docutypes").jqGrid('getCell',
jQuery("#docutypes").jqGrid('getGridParam','selrow'),
'TYPECODE').toString()}).trigger("reloadGrid");
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
});

jQuery("#parameters").jqGrid({
url:"",
datatype: "json",
mtype : "post",
colNames:['paramcode','Order','Name','Type','Options','Column
Size','Default Value','Description'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:false,hidden:true},
{name:'PARAMORDER',index:'PARAMORDER', width:0,
sorttype:"int",editable:false,hidden:true},
{name:'PARAMNAME',index:'PARAMNAME', width:125,
sorttype:"text",editable:false},
{name:'DATATYPE',index:'DATATYPE', width:125,
sorttype:"text",editable:false, edittype: 'select', editrules: {
required: true},editoptions:{value: ""}},
{name:'options',index:'options', width:0,
align:"right",hidden:true,editable:false},
{name:'COLSIZE',index:'COLSIZE', width:0,
align:"right",hidden:true,editable:false},
{name:'DEFAULTVAL',index:'DEFAULTVAL', width:0,
hidden:true,editable:false},
{name:'description',index:'description', width:150,
align:"right",sortable:false,editable:false,editrules: {
},editoptions:{size:70}}
],
rowNum:20,
height: 180,
rowList:[10,20,30],
pager: '#pager2',
sortname: 'PARAMORDER',
viewrecords: true,
sortorder: "asc",
caption: "Parameters"
}).jqGrid('navGrid','#pager2',

```



```

$("#check1").click(function() {
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveStationCreatePermission
").\\";?>,
data: { typeCode : jQuery("#docutypes").jqGrid('getCell', id,
'TYPECODE'), stationCode: <?php echo $stationCode?>, value:
jQuery("#check1").attr("checked") },
success: function(msg){
$( "#saveParamStatus" ).empty();
$( "#saveParamStatus" ).show();
$( "#saveParamStatus" ).append(msg);
$( "#saveParamStatus" ).fadeOut(6000);
}
});
});
}
});
jqGrid('navGrid','#pager3',
{edit:false,add:false,del:false}, //options
{, // edit options
{, // add options
{, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
});
</script>

```

ix. views/docutype/newApplicant.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:20px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:130px;">Username</TD><TD><INPUT
TYPE="TEXT" NAME="TYPENAME" ID="TYPENAME" VALUE=""
size=42></TD>
<TD rowspan="3" style="padding:0px 0px 0px 5px;">
<select id="companylist" size="4" style="font-
size:12px;width:300px;">
<?php
foreach ($companies as $company):
echo "<option
value=\"\".$company['COMPANYCODE'].\">\".$company['COMPA
NYNAME'].\"</option>\n";
endforeach;
?>
</select>
<center style="padding:2px;">
<input type="button" value="<<"
onclick="transfer('companylistvals','companylist');"/>
<input type="button" value=">>"
onclick="transfer('companylist','companylistvals');"/>

```

```

</center>
<select id="companylistvals" size="4" style="font-
size:12px;width:300px;">
</select>
</TD>
</TR>
<TR>
<TD style="width:100px;">Password</TD><TD><INPUT
TYPE="PASSWORD" NAME="PASSWORD" ID="PASSWORD"
VALUE="" size=42></TD>
</TR>
<TR>
<TD style="width:100px;">Retype Password</TD><TD><INPUT
TYPE="PASSWORD" NAME="RPASSWORD" ID="RPASSWORD"
VALUE="" size=42></TD>
</TR>
<TR>
<TD style="width:100px;">First Name</TD><TD><INPUT
TYPE="TEXT" NAME="FNAME" ID="FNAME" VALUE=""
size=42></TD>
</TR>
<TR>
<TD style="width:100px;">Last Name</TD><TD><INPUT
TYPE="TEXT" NAME="LNAME" ID="LNAME" VALUE=""
size=42></TD>
</TR>
<TR>
<TD style="width:100px;">E-Mail</TD><TD><INPUT TYPE="TEXT"
NAME="EMAIL" ID="EMAIL" VALUE="" size=42></TD>
</TR>
<TR>
<TD style="width:100px;">Phone</TD><TD><INPUT TYPE="TEXT"
NAME="PHONE" ID="PHONE" VALUE="" size=42></TD>
</TR>
<TR>
<TD style="width:100px;">Description</TD><TD><TEXTAREA
TYPE="TEXT" NAME="TYPEDESC" ID="TYPEDESC" cols="33"
rows="3"></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" style="padding:0px 0px 0px 0px;">
<select id="typeList" size="5" style="font-size:12px;width:386px;"
onchange="showDetails($('#typeList').val());">
<?php
foreach ($docTypes as $docType):
echo "<option
value=\"\".$docType['TYPECODE'].\">\".$docType['TYPENAME'].\"<
/option>\n";
endforeach;
?>
</select>
</TD>
</TR>
<TR>
<TD colspan="3" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>
<script type="text/javascript">
$(function() {

```

```

$("#typeMeta").css("display", "none");
$("#typeMeta").fadeOut(1000);
$("input:submit, button").button();

$("#typeSaveMeta").button().click(function() {
if ($("#TYPENAME").val() == null || ($("#TYPENAME").val() == "" ||
!/^[a-zA-Z0-9]+$/ .test($("#TYPENAME").val())) {
alert("Invalid title");
} else if ($("#TYPEDESC").val() == null || ($("#TYPEDESC").val() ==
"" ) {
alert("Invalid description");
} else if ($("#typeList").val() == null || ($("#typeList").val() == "" ) {
alert("Please select a type");
} else {
if(confirm('Are you sure you want to save?')){
var complist = [];
$("#companylistvals option").each(function() {
complist.push($("#this").val());
})

$.ajax({
type: "POST",
url: "<?php echo
"\".base_url('index.php/document/saveNewDoc')."\"";?>,
data: { title : $("#TYPENAME").val() , desc : ($("#TYPEDESC").val() ,
typeCode : ($("#typeList").val(),
companylistvals:complist
},
success: function(msg){
if( msg=="failed" || msg==" " || msg==null){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").append("Add Failed!<hr /><br />");
} else {
alert("Document Saved");
window.location.href = "<?php echo
base_url('index.php/document/viewDocs');?>";
}
}
});
}
});
});

function showDetails(selected) {
$("#instance").jqGrid('setGridParam',{ url:<?php echo
"\".base_url('index.php/document/getTypePathInstances')."\"/^\"
; ?>+selected}).trigger("reloadGrid");

$.ajax({
type: "POST",
url: "<?php echo
"\".base_url('index.php/document/getDocuTypeForm')."\"";?>,
data: { typeCode : selected },
success: function(msg){
//alert(msg);
$("#form").empty();
$("#form").show();
$("#form").append(msg);
}
});
});

$("#details").tabs({
collapsible: true,
selected: -1
});

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)/(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>;
};

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#instance").jqGrid({
url:<?php echo
"\".base_url('index.php/document/getTypePathInstances')."\"/^\"
; ?>,
datatype: "json",
mtype : "post",
colNames:['Barcode','Title', 'Location','Duration','Status'],
colModel:[
{name:'BARCODE',index:'BARCODE', width:0 ,hidden:true},
{name:'TITLE',index:'TITLE', width:215,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url('index.php/document/viewDoc');?>/\"+rowObject[0]+\"\"
>\" + cellvalue + \"</a>\";
}
},
{name:'STATIONNAME',index:'STATIONNAME',
width:150,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:140,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:185,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
],
rowNum: 10,
height: 200,
rowList:[10,20,30],

```



```

pager: '#pager2',
sortname: 'TITLE',
viewrecords: true,
sortorder: "desc",
caption: "Type Instances"
});

jQuery("#instance").jqGrid('navGrid', '#pager2',
{
edit:false,add:false,del:false
}, //options
{
}, // edit options
{
}, // add options
{
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);

$(".ui-jqgrid-titlebar").hide();

function transfer(sel1,sel2){
var elSel = document.getElementById(sel1);
var i;
var slctd;
for (i = elSel.length - 1; i>=0; i--) {
if (elSel.options[i].selected) {
slctd = i;
}
}
var elSel2 = document.getElementById(sel2);
var elOptNew = document.createElement('option');
elOptNew.text = elSel.options[slctd].text;
elOptNew.value = elSel.options[slctd].value;
try {
elSel2.add(elOptNew,elSel2.options[0]); // standards compliant;
doesn't work in IE
}
catch(ex) {
elSel2.add(elOptNew,0); // IE only
}
elSel.remove(slctd);
}
}
</script>

```

x. views/docutype/newDocuType.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:100px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:100px;">Type Name</TD><TD><INPUT
TYPE="TEXT" NAME="TYPENAME" ID="TYPENAME" VALUE=""
size=90</TD>
</TR>
<TR>

```

```

<TD style="width:100px;">Type Desc</TD><TD><TEXTAREA
TYPE="TEXT" NAME="TYPEDESC" ID="TYPEDESC" cols="72"
rows="4"></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>

```

```

<script type="text/javascript">

$(function() {
$("#typeMeta").css("display", "none");
$("#typeMeta").fadeIn(1000);
$("input:submit, button").button();

$("#typeSaveMeta").button().click(function() {
if ($("#TYPEDESC").val() == null){
$("#TYPEDESC").val() = "";
}
if(confirm('Are you sure you want to save?')){
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveNewDocuType")."\\"; ?>,
data: { name : $("#TYPENAME").val() , desc: $("#TYPEDESC").val()
},
success: function(msg){
if( msg=="fail" || msg==" " || msg==null){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").append("Add Failed!<hr /><br />");
} else {
window.location.href = "<?php echo
base_url("index.php/document/typeEditMeta");?>"+"/"+msg;
}
}
});
}
});
});
});
</script>

```

xiii. views/docutype/setup.php

```

<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:20em">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio" /><label
for="radio1">METADATA</label>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">PARAMETERS</label>
<input type="radio" id="radio2" name="radio"
checked="checked" /><label for="radio2">PATH</label>
</div>
</div>

<div id="permNavBar1" name="pathGroup" class="exp"
style="right:15px;top:80px;padding:1px;width:400px">
<table id="permissions"></table>

```



```

//}
}

for (var i = 0; i < errMsgs1.length; i++) {
//if ($.inArray(errMsgs1[i], exemptions1) < 0) {
err+= "Cannot reach end node from node "+errMsgs1[i]+"\n";
//}
}

if (err != "") {
return err;
} else {
return "save";
}
}

function recursiveCheck(conn,index) {
//console.log(conn);
var to = conn[index][1];
if ( to == 0 || to == "0" ) {
return true;
} else {
for (var i = 0; i < conn.length; i++) {
//console.log("perm: "+conn[i][2]+ " to: "+to+ " | "+conn[i][1]);
if ( i != index && to == conn[i][0] && conn[i][2] == 0 ) {
//console.log("to: "+to+ " | "+conn[i][1]);
conn[index][2] = 1;
if( recursiveCheck(conn,i) ) {
return true;
}
}
}
return false;
}
}

function recursiveCheck2(conn,index) {
//console.log(conn);
var from = conn[index][0];
if ( from == -1 || from == "-1" ) {
return true;
} else {
for (var i = 0; i < conn.length; i++) {
//console.log("perm: "+conn[i][2]+ " from: "+from+ " | "+conn[i][1]);
if ( i != index && from == conn[i][1] && conn[i][2] == 0 ) {
//console.log("from: "+from+ " | "+conn[i][1]);
conn[index][2] = 1;
if( recursiveCheck2(conn,i) ) {
return true;
}
}
}
return false;
}
}

function saveStationPath() {
var startConn = 0;
var endConn = 0;

var temp = jsPlumb.getConnections();

var statusMsg = checkIfConnected(temp);
if ( statusMsg == "save" ) {
if(confirm('Are you sure you want to save changes?')){

```

```

var temp = jsPlumb.getConnections();

var temp1 = [];
for(var i=0;i<temp.length;i++){
temp1.push([temp[i].sourceId,temp[i].targetId,temp[i].getOverlay("label").getLabel()]);
}

var temp2 = [];

$(".w").each(function(i,e) { if ( e.id != -1 && e.id != 0 ) {
temp2.push([e.id,$(e).css("left").slice(0,-2),$(e).css("top").slice(0,-2)]); }})

$.ajax({
type: "POST",
url: <?php echo "\".base_url("index.php/document/savePathCont")."\";?>,
data: { typeCode : <?php echo "\".$.typeCode."\";?>, connetions : temp1 , stations: temp2 ,
success: function(msg){
$("#rep" ).empty();
$("#rep" ).append(msg);
}
});
} else {
alert("Error.\n"+statusMsg);
}
}

function deleteStation(stationCode,stationName) {
var temp = jsPlumb.getConnections();

var go = 1;

for(var i=0;i<temp.length;i++){
if(temp[i].sourceId == stationCode || temp[i].targetId == stationCode ){
go = 0;
}
}

if( go == 1 ){
if(confirm('Are you sure you want to delete '+stationName+'?')){
$("#"+stationCode).remove();
$("#stationsToBeAdded").append("<option value=\""+ stationCode + "\">"+ stationName + "</option>");
}
} else {
alert("Please remove connections first.");
}
}

$(function() {
$("#editDocTypeSel" ).buttonset();
$("#radio1" ).click(function() {
window.location = <?php echo "\".base_url("index.php/document/typeEditMeta/".$typeCode)."\";?>;
});
$("#radio2" ).click(function() {
window.location = <?php echo "\".base_url("index.php/document/setupworkflow/".$typeCode)."\";?>;
}

```

```

});
$("#radio3").click(function() {
window.location = <?php echo
"\".base_url("index.php/document/typeEditParam/\".$typeCode)
.\"\".\"?>;
});
});
});

```

```
</script>
```

```
<!-- end js codes -->
```

```

<div name="pathGroup" id="stationFormat" class="exp"
style="right:1em;top:380px;">
<div id="station">
<ul>
<li><a href="#tabs-1" style="font-size: 12px;">add
station</a></li>
<li><a href="#tabs-2" style="font-size: 12px;">view path
instances</a></li>
<li><a href="#tabs-3" style="font-size: 12px;">save path</a></li>
</ul>
<div id="tabs-1">
<select id="stationsToBeAdded" style="font-size:12px;">
<?php
foreach ($stationsToAdd as $stationsToAdd_item):
echo "<option
value=\"\". $stationsToAdd_item['STATIONCODE'].\"\">\".$stationsT
oAdd_item['STATIONNAME'].\"</option>\n";
endforeach;
?>
</select>
<button id="addStation" style="font-size: 12px;">add</button>
</div>
<div id="tabs-2" style="padding:3px">
<table id="instance"></table>
<div id="pager"></div>
</div>
<div id="tabs-3">
<button id="savePath" onClick="saveStationPath()" style="font-
size: 12px;">save</button>
</div>
</div>
<div id="rep"></div>
</div>

```

```

<div id="dialogCon" title="Connection" style="font-size: 14px;">
<p id="validateTips"></p>
<fieldset id="conField">
<label for="conname">&nbsp;Connection
Name:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</label>
<input type="text" name="conname" id="conname" class="text
ui-widget-content ui-corner-all" />
</fieldset>
</div>

```

```
<!-- more js codes for grid -->
```

```
<script type="text/javascript">
```

```

function updateTips( t ) {
$( "#validateTips" ).text( t ).addClass( "ui-state-highlight" );
setTimeout(function() {
$( "#validateTips" ).removeClass( "ui-state-highlight", 1500 );
}, 500 );

```

```
}
```

```

function checkRegexp( o, regexp, n ) {
if ( !( regexp.test( o.val() ) ) ) {
o.addClass( "ui-state-error" );
updateTips( n );
return false;
} else {
return true;
}
}

```

```

var gradientNumberFormat = function ( cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue ) {
var dataAsNumber = parseFloat( cellvalue ); /* parseInt( cellvalue,
10 ); */
if ( dataAsNumber > maxDataValue ) {
dataAsNumber = maxDataValue;
}
if ( dataAsNumber < minDataValue ) {
dataAsNumber = minDataValue;
}

```

```
var colour = rgbToHex( cellvalue, 100 - cellvalue, 0 );
```

```

var prozentVal = minDisplayValue + ( dataAsNumber -
minDataValue ) / ( maxDataValue - minDataValue );
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top: -5px; right: 2px; height: 22px; float: left; width: '+
prozentVal + '%; background: '+ colour + ';"></div><div
class="cellTextRight">'+ cellvalue + '%</div></div>';
};

```

```

function componentToHex( c ) {
c = Math.floor( ( c / 100 ) * 255 );
var numClr = c.toString( 16 );
return numClr.length == 1 ? "0" + numClr : numClr;
}

```

```

function rgbToHex( r, g, b ) {
return "#" + componentToHex( r ) + componentToHex( g ) +
componentToHex( b );
}

```

```

jQuery( "#instance" ).jqGrid( {
url: <?php echo
"\".base_url("index.php/document/getTypePathInstances").\" /\".$
typeCode.\"\". \"?>,
datatype: "json",
mtype : "post",
colNames: [ 'Barcode', 'Title', 'Location', 'Duration', 'Status' ],
colModel: [
{ name: 'BARCODE', index: 'BARCODE', width: 0, hidden: true },
{ name: 'TITLE', index: 'TITLE', width: 115, sortable: true, editable: true,
formatter: function ( cellvalue, options, rowObject ) {
return "<a href=\"\"<?php echo
base_url("index.php/document/viewDoc");?>\" + rowObject[ 0 ] + \"\"
>\" + cellvalue + "</a>";
}
},
{ name: 'STATIONNAME', index: 'STATIONNAME',
width: 100, sortable: true },
{ name: 'ELAPSED', index: 'ELAPSED',
width: 140, sortable: true, searchoptions: { sopt: [ ] } },

```

```

{name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
],
rowNum: 10,
height: 100,
rowList:[10,20,30],
pager: '#pager',
sortname: 'TITLE',
viewrecords: true,
sortorder: "desc",
caption: "Type Instances"
});

```

```

jQuery("#instance").jqGrid('navGrid','#pager',
{
edit:false,add:false,del:false
});//options
{
};// edit options
{
};// add options
{
};// del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
};// search options
);

```

```

jQuery("#permissions").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getPermissions/0")."/0\\"",
?>,
datatype: "json",
mtype: "post",
colNames:['parameter code', 'Parameter Name', 'Permission
Type'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:true,hidden:true},
{name:'PARAMNAME',index:'PARAMNAME', width:200,
sorttype:"text",editable:false},
{name:'PERTYPE',index:'PERTYPE', width:189,
sorttype:"text",editable:true,edittype:"select",editoptions:{value:
"r:read;rw:read/write"}}
],
rowNum:20,
height: 80,
rowList:[10,20,30],
pager: '#pager1',
sortname: 'PARAMNAME',
viewrecords: true,
sortorder: "asc",
caption: "Permissions"
}).jqGrid('navGrid','#pager1',
{edit:false,add:false,del:true},//options
{
};// edit options
{
};// add options
{
url: <?php echo
"\\".base_url("index.php/document/deletePermission")."\\"; ?>,

```

```

reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete data permission?",
delData: {
paramCode: function() {
var sel_id = jQuery("#permissions").jqGrid('getGridParam',
'selrow');
var value = jQuery("#permissions").jqGrid('getCell', sel_id,
'PARAMCODE');
return value;
},
stationCode: function() {
return stationCode;
}
},
afterComplete:function(msg){
$("#parameters").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getParamElementsPermissio
n/")."/\\"",
?>+stationCode+"/"+"<?php echo
$typeCode;?>").trigger("reloadGrid");
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
});// del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
};// search options
);

```

```

jQuery("#parameters").jqGrid({
url:"",
datatype: "json",
mtype: "post",
colNames:['paramcode','Order','Name','Type', 'Options', 'Column
Size', 'Default Value', 'Description'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:false,hidden:true},
{name:'PARAMORDER',index:'PARAMORDER', width:0,
sorttype:"int",editable:false,hidden:true},
{name:'PARAMNAME',index:'PARAMNAME', width:200,
sorttype:"text",editable:false},
{name:'DATATYPE',index:'DATATYPE', width:189,
sorttype:"text",editable:false, edittype: 'select', editrules: {
required: true},editoptions:{value: ""}},
{name:'options',index:'options', width:0,
align:"right",hidden:true,editable:false},
{name:'COLSIZE',index:'COLSIZE', width:0,
align:"right",hidden:true,editable:false},
{name:'DEFAULTVAL',index:'DEFAULTVAL', width:0,
hidden:true,editable:false},
{name:'description',index:'description', width:0, hidden:true,
align:"right",sortable:false,editable:false,editrules: {
},editoptions:{size:70}}
],
rowNum:20,
height: 80,
rowList:[10,20,30],
pager: '#pager2',
sortname: 'PARAMORDER',
viewrecords: true,

```

```

sortorder: "asc",
caption: "Parameters"
}).jqGrid('navGrid','#pager2',
{edit:false,add:false,del:false}, //options
{}), // edit options
{}), // add options
{}), // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
).jqGrid('navButtonAdd','#pager2',{
caption:"ADD",
buttonicon:"ui-icon-newwin",
onClickButton:function(){
var sel_id = jQuery("#parameters").jqGrid('getGridParam',
'selrow');
if(sel_id==null){
alert("Please select a parameter.");
} else {
var value = jQuery("#parameters").jqGrid('getCell', sel_id,
'PARAMCODE');
jQuery("#permissions").jqGrid('editGridRow',"new",{
url: <?php echo
"\\".base_url("index.php/document/savePermission")."\""; ?>,
height:100,
width:450,
reloadAfterSubmit:true,
addCaption: 'Add Permission',
type: "POST",
closeAfterAdd: true,
editData: { stationCode:stationCode, paramCode: value },
afterComplete:function(msg){
$("#parameters").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getParamElementsPermissio
n/")."\""; ?>+stationCode+"/"+"<?php echo
$typeCode;?>}).trigger("reloadGrid");
$("#permissions").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getPermissions/")."\"";
?>+stationCode+"/"+"<?php echo
$typeCode;?>}).trigger("reloadGrid");
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
});
}
});

function selectStation(stnCode) {
stationCode = stnCode;
$("#parameters").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getParamElementsPermissio
n/")."\""; ?>+stationCode+"/"+"<?php echo
$typeCode;?>}).trigger("reloadGrid");
$("#permissions").jqGrid('setGridParam',{ url:<?php echo
"\\".base_url("index.php/document/getPermissions/")."\"";
?>+stationCode+"/"+"<?php echo
$typeCode;?>}).trigger("reloadGrid");

$("#create").empty();
$("#create").show();

$("#create").append("<font color='\"#818181\"' size=2><input
type='\"checkbox\"' id='\"check1\"' /><label for='\"check1\"'>Create
Permission</label></font>");
$("#create").buttonset();

$("#check1").click(function() {
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveStationCreatePermission
")."\""; ?>,
data: { typeCode : <?php echo $typeCode; ?>, stationCode:
function () { return stationCode; }, value:
jQuery("#check1").attr("checked") },
success: function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg);
$("#saveParamStatus").fadeOut(6000);
}
});
});

$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/getStationCreatePermission
")."\""; ?>,
data: { typeCode : <?php echo $typeCode; ?>, stationCode:
function () { return stationCode; }, value:
jQuery("#check1").attr("checked") },
success: function(msg){
if (msg=="y") {
//alert(msg);
$("#create").empty();
$("#create").show();
$("#create").append("<font color='\"#818181\"' size=2><input
type='\"checkbox\"' id='\"check1\"' checked='\"checked\"' /><label
for='\"check1\"'>Create Permission</label></font>");
$("#create").buttonset();

$("#check1").click(function() {
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveStationCreatePermission
")."\""; ?>,
data: { typeCode : <?php echo $typeCode; ?>, stationCode:
function () { return stationCode; }, value:
jQuery("#check1").attr("checked") },
success: function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg);
$("#saveParamStatus").fadeOut(6000);
}
});
});
}
});

$("#ui-jqgrid-titlebar").hide();
</script>

```

```
<!-- end js codes for grid -->
```

xii. views/docutype/stationEdit.php

```
<div id="typeMeta" class="exp"
style="left:220px;top:100px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:100px;">Type Name</TD><TD><INPUT
TYPE="TEXT" NAME="STATIONNAME" ID="STATIONNAME"
VALUE="<?php echo $documentTypeMeta['TYPENAME']; ?>"
size=90></TD>
</TR>
<TR>
<TD style="width:100px;">Type Desc</TD><TD><TEXTAREA
TYPE="TEXT" NAME="DESC" ID="DESC" cols="72"
rows="4"><?php echo $stationMeta['TYPEDESC'];
?></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>
```

```
<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:20em">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio"
checked="checked"/><label for="radio1">METADATA</label>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">PATH</label>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">PARAMETERS</label>
</div>
</div>
```

```
<script type="text/javascript">
$(function() {
$("#typeMeta").css("display", "none");
$("#typeMeta").fadeIn(1000);
$("input.submit, button").button();

$("#typeSaveMeta").button().click(function() {
$.ajax({
type: "POST",
url: <?php echo
"\".base_url("index.php/document/saveTypeMeta").\""; ?>,
data: { typeCode : <?php echo "\".base_url("index.php/document/saveTypeMeta").\""; ?>, name :
$("#TYPENAME").val() , desc: ($("#TYPEDESC").val() ) ,
success: function(msg){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").append(msg+"<hr /><br />");
}
});
});
});
```

```
$("#editDocTypeSel").buttonset();
```

```
$("#radio1").click(function() {
```

```
window.location = <?php echo
"\".base_url("index.php/document/typeEditMeta/\".$typeCode).\"";
?>;
});
$("#radio2").click(function() {
window.location = <?php echo
"\".base_url("index.php/document/setupworkflow/\".$typeCode)
.\""; ?>;
});
$("#radio3").click(function() {
window.location = <?php echo
"\".base_url("index.php/document/typeEditParam/\".$typeCode)
.\""; ?>;
});
});
</script>
```

xiii. views/docutype/typeEditMeta.php

```
<div id="typeMeta" class="exp"
style="left:220px;top:100px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:100px;">Type Name</TD><TD><INPUT
TYPE="TEXT" NAME="TYPENAME" ID="TYPENAME"
VALUE="<?php echo $documentTypeMeta['TYPENAME']; ?>"
size=90></TD>
</TR>
<TR>
<TD style="width:100px;">Type Desc</TD><TD><TEXTAREA
TYPE="TEXT" NAME="TYPEDESC" ID="TYPEDESC" cols="72"
rows="4"><?php echo $documentTypeMeta['TYPEDESC'];
?></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>
```

```
<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:20em">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio"
checked="checked"/><label for="radio1">METADATA</label>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">PARAMETERS</label>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">PATH</label>
</div>
</div>
```

```
<script type="text/javascript">
$(function() {
$("#typeMeta").css("display", "none");
$("#typeMeta").fadeIn(1000);
$("input.submit, button").button();

$("#typeSaveMeta").button().click(function() {
$.ajax({
```

```

type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveTypeMeta")."\\";?>,
data: { typeCode : <?php echo "\\".$typeCode."\\";?>, name :
$("#TYPENAME").val() , desc: $("#TYPEDESC").val() },
success: function(msg){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").append(msg+"<hr /><br />");
}
});
});
});

$("#editDocTypeSel").buttonset();

$("#radio1").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditMeta/".$typeCode)."\\";?>;
});
$("#radio2").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/setupworkflow/".$typeCode)."\\";?>;
});
$("#radio3").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditParam/".$typeCode)."\\";?>;
});
});
});
</script>

```

xiv. views/docutype/typeEditParam.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:70px;padding:5px;background
und-color:white;border:2px solid #ddd; ;">
<div style="padding:10px;"><font size="6"
color="#b8b8b7"><?php echo $typeData['TYPENAME'];?>
Parameters</font></div>
<table id="parameters"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:20em">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio" /><label
for="radio1">METADATA</label>
<input type="radio" id="radio3" name="radio"
checked="checked" /><label for="radio3">PARAMETERS</label>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">PATH</label>
</div>

</div>

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit,button").button();

```

```

$("#editDocTypeSel").buttonset());

$("#radio1").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditMeta/".$typeCode)."\\";?>;
});
$("#radio2").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/setupworkflow/".$typeCode)."\\";?>;
});
$("#radio3").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/typeEditParam/".$typeCode)."\\";?>;
});
});

jQuery("#parameters").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getParamElements/$typeCod
e")."\\";?>,
datatype: "json",
mtype: "post",
colNames:['paramcode','Order','Name','Input Type','Options','Col
Size','Default Value','Description'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:true,hidden:true},
{name:'PARAMORDER',index:'PARAMORDER', width:50,
sorttype:"int",editable:true,editrules: { integer:true, minValue:1}},
{name:'PARAMNAME',index:'PARAMNAME', width:100,
sorttype:"text",editable:true,editrules: { required:
true},editoptions:{size:20}},
{name:'INPUTTYPE',index:'INPUTTYPE', width:70,editable:true,
editttype: 'select', editrules: { required:
true},editoptions:{value:"text;text;select;checkbox;checkboxo
x;radio;radio;textarea:textarea"}},
{name:'options',index:'options', width:200,
align:"right",sortable:false, editable:true, editrules: {
},editoptions:{size:70}},
{name:'COLSIZE',index:'COLSIZE', width:70,
align:"right",sortable:false, editable:true, editrules: { required:
true},editoptions:{size:70}},
{name:'DEFAULTVAL',index:'DEFAULTVAL', width:100,
align:"right",sorttype:"text",editable:true,editrules: {
},editoptions:{size:20}},
{name:'description',index:'description', width:200,
align:"right",sortable:false,editable:true,editrules: {
},editoptions:{size:70}}
],
rowNum:20,
height: 300,
rowList:[10,20,30],
pager: '#pager',
sortname: 'PARAMORDER',
viewrecords: true,
sortorder: "asc",
caption: "Parameters"
});

jQuery("#parameters").jqGrid('navGrid','#pager',

```



```

}, //options
{
url: <?php echo
"\\".base_url("index.php/document/editParameter")."\\"; ?>,
height:280,
width:750,
reloadAfterSubmit:true,
editCaption: 'Edit Parameter',
type: "POST",
closeAfterEdit: true,
editData: { typecode: <?php echo "\\".$typeCode."\\";?>, orderid:
jQuery("#parameters").jqGrid('getGridParam','selrow')},
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // edit options
{
url: <?php echo
"\\".base_url("index.php/document/saveNewParameter")."\\"; ?>,
height:280,
width:750,
reloadAfterSubmit:true,
addCaption: 'New Parameter',
type: "POST",
closeAfterAdd: true,
editData: { typecode: <?php echo "\\".$typeCode."\\";?>,
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteParameter")."\\"; ?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete parameter?",
delData: {
typecode: <?php echo "\\".$typeCode."\\";?>,
order: function() {
var sel_id = jQuery("#parameters").jqGrid('getGridParam',
'selrow');
var value = jQuery("#parameters").jqGrid('getCell', sel_id,
'PARAMORDER');
return value;
},
paramcode: function() {
var sel_id = jQuery("#parameters").jqGrid('getGridParam',
'selrow');
var value = jQuery("#parameters").jqGrid('getCell', sel_id,
'PARAMCODE');
return value;
}
},
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
}
}
}

```

```

$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
$("#ui-jqgrid-titlebar").hide();

// for(var i=0;i<=mydata.length;i++)
//
jQuery("#parameters").jqGrid('addRowData',i+1,mydat
a[i]);
</script>

```

xv. views/docutype/userAccount.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:20px;padding:10px;width:750px;color:#828
282;">
<div id="saveMetaStatus"></div>
<table>
<tr>
<td style="width:200px; padding:30px 10px 10px 10px;">User
Name</td><td style="width:200px; padding:30px 10px 10px 10px
0px;"><input type="text" name="username"
id="username" value="<?php echo $userData['username'];
?>" size=90></td>
</tr>
<tr>
<td style="width:200px;
padding:10px;">Password</td><td><input type="password"
name="password" id="password" value="<?php echo
$userData['password']; ?>" size=90></td>
</tr>
<tr>
<td style="width:200px; padding:10px;">Retype
Password</td><td><input type="password"
name="password2" id="password2" value="<?php echo
$userData['password']; ?>" size=90></td>
</tr>
<tr>
<td style="width:200px; padding:10px;">User
Type</td><td><input type="text" name="usertype"
id="usertype" value="<?php echo $userData['usertype']; ?>"
size=90 readonly></td>
</tr>
<tr>
<td style="width:200px;
padding:10px;">Department</td><td><input type="text"
name="stationcode" id="stationcode" value="<?php
if($userid = $this->session->userdata('usertype')=="admin"){
echo $userData['stationcode'];
} else {
echo $userData['stationname'];
}
?>" size=90 readonly></td>
</tr>
<tr>
<td style="width:200px; padding:10px;">First
Name</td><td><input type="text" name="firstname"

```

```

ID="FIRSTNAME" VALUE="<?php echo $userData['FIRSTNAME'];
?>" size=90></TD>
</TR>
<TR>
<TD style="width:200px; padding:10px;">Last
Name</TD><TD><INPUT TYPE="TEXT" NAME="LASTNAME"
ID="LASTNAME" VALUE="<?php echo $userData['LASTNAME'];
?>" size=90></TD>
</TR>
<TR>
<TD style="width:200px; padding:10px;">E-
mail</TD><TD><INPUT TYPE="TEXT" NAME="EMAIL" ID="EMAIL"
VALUE="<?php echo $userData['EMAIL']; ?>" size=90></TD>
</TR>
<TR>
<TD style="width:200px;
padding:10px;">Phone</TD><TD><INPUT TYPE="TEXT"
NAME="PHONE" ID="PHONE" VALUE="<?php echo
$userData['PHONE']; ?>" size=90></TD>
</TR>
<TR>
<TD style="width:200px;
padding:10px;">Address</TD><TD><INPUT TYPE="TEXT"
NAME="ADDRESS" ID="ADDRESS" VALUE="<?php echo
$userData['ADDRESS']; ?>" size=90></TD>
</TR>
<TR>
<TD style="width:200px; padding:30px 10px 10px
10px;">Theme</TD><TD style="width:200px; padding:30px 10px
10px 0px;">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio"
value="smoothness" <?php
if($userData['THEME']=='smoothness'){ echo
"checked=\"checked\""; } ?>><label
for="radio1">Smoothness</label>
<input type="radio" id="radio2" name="radio" value="ui-
lightness" <?php if($userData['THEME']=='ui-lightness'){ echo
"checked=\"checked\""; } ?>><label
for="radio2">Lightness</label>
<input type="radio" id="radio3" name="radio" value="start"
<?php if($userData['THEME']=='start'){ echo
"checked=\"checked\""; } ?>><label for="radio3">Nature</label>
<input type="radio" id="radio4" name="radio" value="cute"
<?php if($userData['THEME']=='cute'){ echo
"checked=\"checked\""; } ?>><label for="radio4">Red</label>
</div>
</TD>
</TR>
<TR>
<TD colspan="2" align="right">
<button id="saveUser" style="font-size: 12px;">save</button>
</TD>
</TR>
</table>
</div>

<script type="text/javascript">
$(function() {
$("#typeMeta").css("display", "none");
$("#typeMeta").fadeIn(1000);
$("input:submit, button").button();

$("#saveUser").button().click(function() {
$.ajax({
type: "POST",

```

```

url: <?php echo
"\".base_url("index.php/document/saveProfile").\"";?>,
data: { username : $("#USERNAME").val(), password :
$("#PASSWORD").val(), password2: $("#PASSWORD2").val(),
firstname: $("#FIRSTNAME").val(), lastname:
$("#LASTNAME").val(), email: $("#EMAIL").val(), phone:
$("#PHONE").val(), address: $("#ADDRESS").val(),
theme:$("#input:radio[name=radio]:checked").val() },
success: function(msg){
$("#saveMetaStatus").empty();
$("#saveMetaStatus").show();
$("#saveMetaStatus").append(msg+"<hr />");
$("#saveMetaStatus").fadeOut(15000);
}
});
});
$("#editDocTypeSel").buttonset();
});
</script>

```

xvi. views/docutype/viewAdminDoc.php

```

<script type="text/javascript" src="<?php echo
base_url("development-
bundle/ui/jquery.ui.tabs.js");?>"></script>
<script type="text/javascript" src="<?php echo
base_url("js/list.js");?>"></script>
<script type="text/javascript" src="<?php echo
base_url("js/helper-jquery.js");?>"></script>
<script type="text/javascript">
;(function() {

window.jsPlumbFDA = {

init :function() {

jsPlumb.importDefaults({
Endpoint : ["Dot", {radius:2}],
HoverPaintStyle : {strokeStyle:"#6b6b6b", lineWidth:2 },
ConnectionOverlays : [
["Arrow", {
location:1,
id:"arrow",
length:14,
foldback:0.8
}],
["Label", { id:"label", cssClass:"aLabel" }]
]
});

$("#ep").each(function(i,e) {
var p = $(e).parent();
var pid = p[0].id;
console.log(pid+"="+i);

jsPlumb.makeSource($(e), {
parent:p,
//anchor:"BottomCenter",
anchor:"Continuous",
//paintStyle:{ fillStyle:"ffffff",radius:0 },
connector:["StateMachine", { curviness:20 } ],

```



```

formatter:function (cellvalue, options, rowObject) {
return "<a href=\<?php echo
base_url("index.php/document/viewStation");?>/"+rowObject[0]
+ "\>" + cellvalue + "</a>";
}
},
{name:'TIMEIN',index:'TIMEIN', width:117,
align:"right",sorttype:"text"},
{name:'TIMEOUT',index:'TIMEOUT', width:117,
align:"right",sorttype:"text"}
],
rowNum:20,
rowList:[10,20,30],
pager: '#pager',
sortname: 'TIMEIN',
viewrecords: true,
sortorder: "asc",
caption: "Document History"
});

jQuery("#docloctable").jqGrid('navGrid','#pager',
{
edit:false,add:false,del:false
}, //options
{
}, // edit options
{
}, // add options
{
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
</script>

```

xvii. views/docutype/viewAdminDocs.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:48
48px;background-color:white;border:2px solid #ddd; ;">
<div id="docSel" style="position:absolute;left:10px;top:10px;">
<?php
if ($this->session->userdata('usertype') != "assistant" ) {
?>
<input type="radio" id="radio2" name="radio"
checked="checked"/><label for="radio2">ALL ACTIVE</label>
<?php
}
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

<input type="radio" id="radio1" name="radio" /><label
for="radio1">STATION</label>

<?php
}
?>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">ARCHIVED</label>

```

```

</div>
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Documents</font><br /></div>
<table id="docutype"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});

$(function() {
$("#docSel").buttonset();
<?php
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

$("#radio1").click(function() {
window.location = <?php echo
"\>".base_url("index.php/document/viewStationDocs")."\>";?>
});
<?php
}
if ($this->session->userdata('usertype') != "assistant" ) {
?>
$("#radio2").click(function() {
window.location = <?php echo
"\>".base_url("index.php/document/viewDocs")."\>";?>
});
<?php
}
?>
$("#radio3").click(function() {
window.location = <?php echo
"\>".base_url("index.php/document/viewArcDocs")."\>";?>
});
});

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

```

```

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#docutype").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getDocElements").\\"";?>,
datatype: "json",
mtype : "post",
colNames:['Lock0','Lock','Barcode','Title','Type','Location','Time
Elapsed','Progress','Companies'],
colModel:[
{name:'LOCK0',index:'LOCK0', width:0,hidden:true},
{name:'LOCK',index:'LOCK', <?php if ($this->session-
>userdata('usertype') == "head") { echo "width:0,hidden:true";}
else {echo "width:30";}?>,sortable:false,
formatter:function (cellvalue, options, rowObject) {
if ( cellvalue == "y" || cellvalue == 'y' ) {
return "<input id=\"checkbox\" + rowObject[2] + \"\
type=\"checkbox\" onclick=\"toggleCheckbox('checkbox\" +
rowObject[2] + \"\", \"\" + rowObject[2] + \"\")\" value=\" + cellvalue +
checked=\"checked\">";
} else if ( cellvalue == "n" || cellvalue == 'n' ) {
return "<input id=\"checkbox\" + rowObject[2] + \"\
type=\"checkbox\" onclick=\"toggleCheckbox('checkbox\" +
rowObject[2] + \"\", \"\" + rowObject[2] + \"\")\" value=\" + cellvalue +
\">";
} else {
return "<input id=\"checkbox\" + rowObject[2] + \"\
type=\"checkbox\" onclick=\"toggleCheckbox('checkbox\" +
rowObject[2] + \"\", \"\" + rowObject[2] + \"\")\" value=\" + cellvalue +
disabled=\"true\">";
}
},
unformat:function (cellvalue, options, rowObject) {
return cellvalue;
}
},
{name:'DOCUD.BARCODE',index:'DOCUD.BARCODE', width:50
,sortable:true},
{name:'TITLE',index:'TITLE', width:125,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url("index.php/document/viewDoc");?>\"+rowObject[2]+\"\"
>\" + cellvalue + \"</a>";
}
},
{name:'TYPENAME',index:'TYPENAME',
width:100,editable:true,sortable:true},
{name:'STATIONNAME',index:'STATIONNAME',
width:100,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:105,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
}
}

```

```

},
{name:'COMPANIES',index:'COMPANIES',
width:200,sortable:false}
],
rowNum: 20,
height: 400,
rowList:[10,20,30],
pager: '#pager',
sortname: 'ELAPSED',
viewrecords: true,
sortorder: "desc",
caption: "Documents"
});

function toggleCheckbox(cbid,barcd){
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveLock").\\"";?>,
data: { barCode : barcd, value: jQuery("#"+cbid).attr("checked") },
success: function(msg){
$( "#saveParamStatus" ).empty();
$( "#saveParamStatus" ).show();
$( "#saveParamStatus" ).append(msg);
$( "#saveParamStatus" ).fadeOut(6000);
}
});
}

jQuery("#docutype").jqGrid('navGrid','#pager',
{
<?php if ($this->session->userdata('usertype') == "admin" ) {?>
editfunc: function (rowid) {
var lk = jQuery("#docutype").jqGrid('getCell',
jQuery("#docutype").jqGrid('getGridParam', 'selrow'), 'LOCK0');
if( lk == "y" || lk == 'y' ){
window.location.href = "<?php echo
base_url("index.php/document/formedit");?>\"+\"/\"+jQuery("#doc
utype").jqGrid('getCell',
jQuery("#docutype").jqGrid('getGridParam', 'selrow'),
'DOCUD.BARCODE');
} else {
alert("Lock document first.");
}
},
<?php } else { ?>
edit:false,
<?php } ?>
addfunc: function () {
window.location.href = "<?php echo
base_url("index.php/document/newDoc");?>";
}
}, //options
{
}, // edit options
{
}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteDoc").\\"";?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete this document?",
delData: {

```

```

barcode:
function() {
var sel_id = jQuery("#docutype").jqGrid('getGridParam', 'selrow');
var value = jQuery("#docutype").jqGrid('getCell', sel_id,
'DOCUD.BARCODE');
return value;
}
},

afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
});

$(".ui-jqgrid-titlebar").hide();
</script>

```

xviii. views/docutype/viewDocType.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:6
45px;background-color:white;border:2px solid #ddd; ;">
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Document Types</font><br /></div>
<table id="docutype"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});

jQuery("#docutype").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getTypeElements").\\""; ?>,
datatype: "json",
mtype: "post",
colNames:['Typecode','Name','Description'],
colModel:[
{name:'TYPECODE',index:'TYPECODE', width:0 ,hidden:true},
{name:'TYPENAME',index:'TYPENAME', width:200,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\\"<?php echo
base_url("index.php/document/viewDocType");?>/\"+rowObject[0
]+\"\\>\" + cellvalue + \"</a>\";
}
},
{name:'TYPEDESC',index:'TYPEDESC', width:430,sortable:false}
],

```

```

rowNum: 20,
height: 300,
rowList:[10,20,30],
pager: '#pager',
sortname: 'TYPENAME',
viewrecords: true,
sortorder: "desc",
caption: "Document Types"
});

jQuery("#docutype").jqGrid('navGrid','#pager',
{
editfunc: function (rowid) {
window.location.href = "<?php echo
base_url("index.php/document/typeEditMeta");?>"+"/"+rowid;
},
addfunc: function () {
window.location.href = "<?php echo
base_url("index.php/document/typeNewMeta");?>";
}
}, //options
{
}, // edit options
{
}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteType").\\""; ?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete this type?",
delData: {
typecode:
function() {
var sel_id = jQuery("#docutype").jqGrid('getGridParam', 'selrow');
var value = jQuery("#docutype").jqGrid('getCell', sel_id,
'TYPECODE');
return value;
}
},

afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
});

$(".ui-jqgrid-titlebar").hide();
</script>

```

xix. views/docutype/viewAllStations.php

```
<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:6
45px;background-color:white;border:2px solid #ddd; ;">
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Departments</font><br /></div>
<table id="stations"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>
```

```
<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input.submit,button").button();
});
```

```
jQuery("#stations").jqGrid({
url:<?php echo
"\".base_url("index.php/document/getStationElements").\"";
?>,
datatype: "json",
mtype: "post",
colNames:['stationCode', 'Name', 'Doc Num', 'Description'],
colModel:[
{name:'STATIONCODE',index:'STATIONCODE', width:0,
sorttype:"int",editable:true,hidden:true},
{name:'STATIONNAME',index:'STATIONNAME',
width:200,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url("index.php/document/viewStation");?>/\"+rowObject[0]
+\"\\>\" + cellvalue + \"</a>\";
},
unformat:function (cellvalue, options, rowObject) {
return cellvalue;
}
},
{name:'CONGEST',index:'CONGEST', width:100, editable:false,
formatter: function (cellvalue) {
var max = 100;
var colour = rgbToHex(cellvalue,max-cellvalue,0,max);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top: -5px; right: 2px; height:22px; float:left; width:'+
'100%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
}
},
{name:'STATIONDESC',index:'STATIONDESC', width:330,
align:"right",sortable:false,editable:true,editrules: {
},editoptions:{size:70}}
],
rowNum:20,
height: 300,
rowList:[10,20,30],
pager: '#pager',
sortname: 'STATIONNAME',
viewrecords: true,
sortorder: "asc",
caption: "Stations"
});
```

```
function componentToHex(c,max) {
if ( c < 0 ) {
c = 0;
} else if ( c > max ) {
c = 255;
} else {
c = Math.floor((c/max)*255);
}
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}
```

```
function rgbToHex(r, g, b ,max) {
return "#" + componentToHex(r,max) + componentToHex(g,max)
+ componentToHex(b,max);
}
```

```
jQuery("#stations").jqGrid('navGrid','#pager',
{
<?php
if ($this->session->userdata('usertype') != "admin") {
?>
edit:false,add:false,del:false
<?php
}
?>
}, //options
{
url: <?php echo
"\".base_url("index.php/document/editStation").\""; ?>,
height:180,
width:650,
reloadAfterSubmit:true,
editCaption: 'Edit Station',
type: "POST",
closeAfterEdit: true,
editData: { stationCode: function() {
var sel_id = jQuery("#stations").jqGrid('getGridParam', 'selrow');
var value = jQuery("#stations").jqGrid('getCell', sel_id,
'STATIONCODE');
return value;
}},
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // edit options
{
url: <?php echo
"\".base_url("index.php/document/saveNewStation").\""; ?>,
height:180,
width:650,
reloadAfterSubmit:true,
addCaption: 'New Station',
type: "POST",
closeAfterAdd: true,
editData: { },
afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}
```



```

}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteStation")."\\"; ?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete station?",
delData: {
stationcode: function() {
var sel_id = jQuery("#stations").jqGrid('getGridParam', 'selrow');
var value = jQuery("#stations").jqGrid('getCell', sel_id,
'STATIONCODE');
return value;
}
},

afterComplete:function(msg){
$("#saveParamStatus").empty();
$("#saveParamStatus").show();
$("#saveParamStatus").append(msg.responseText);
$("#saveParamStatus").fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
)<?php
if ($this->session->userdata('usertype') == "admin") {
?>.jqGrid('navButtonAdd','#pager',{
caption:"Edit Permission",
buttonicon:"ui-icon-newwin",
onClickButton:function(lol){
var sel_id = jQuery("#stations").jqGrid('getGridParam', 'selrow');
if(sel_id==null){
alert("Please select a row.");
} else {
var value = jQuery("#stations").jqGrid('getCell', sel_id,
'STATIONCODE');
window.location.href = "<?php echo
base_url("index.php/document/editPermission");?>"+"/"+value;
}
},
position: "last",
title:"",
cursor: "pointer"
})<?php
}
?>

$(".ui-jqgrid-titlebar").hide();
</script>

```

xx. views/docutype/ViewApplicantDocs.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:48px;background-color:white;border:2px solid #ddd; ">
<div id="docSel" style="position:absolute;left:10px;top:10px;">
</div>
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Documents</font><br /></div>

```

```

<table id="docutype"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

```

```

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});
var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}
}

```

```
var colour = rgbToHex(cellvalue,100-cellvalue,0);
```

```

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width: '+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

```

```

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

```

```

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

```

```

jQuery("#docutype").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getDocApplicantElements")."
\\"; ?>,
datatype: "json",
mtype: "post",
colNames:['Barcode','Title','Type','Location','Time
Elapsed','Progress','Companies'],
colModel:[
{name:'DOCUD.BARCODE',index:'DOCUD.BARCODE', width:50
,sortable:true},
{name:'TITLE',index:'TITLE', width:125,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url("index.php/document/viewDoc");?>/"+rowObject[0]+\"
\">\" + cellvalue + "</a>";
}
},
{name:'TYPENAME',index:'TYPENAME',
width:150,editable:true,sortable:true},

```

```

{name:'STATIONNAME',index:'STATIONNAME',
width:100,sortable:true,hidden:true},
{name:'ELAPSED',index:'ELAPSED',
width:140,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
{name:'COMPANIES',index:'COMPANIES',
width:250,sortable:false}
],
rowNum: 20,
height: 400,
rowList:[10,20,30],
pager: '#pager',
sortname: 'ELAPSED',
viewrecords: true,
sortorder: "desc",
caption: "Documents"
});

jQuery("#doctype").jqGrid('navGrid','#pager',
{
edit:false,add:false,del:false
}), //options
{
}, // edit options
{
}, // add options
{
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);

$(".ui-jqgrid-titlebar").hide();
</script>

```

xxi. views/docuype/viewArcDoc.php

```

<script type="text/javascript" src="<?php echo
base_url("development-
bundle/ui/jquery.ui.tabs.js");?>"></script>
<script type="text/javascript" src="<?php echo
base_url("js/list.js");?>"></script>
<script type="text/javascript" src="<?php echo
base_url("js/helper-jquery.js");?>"></script>
<script type="text/javascript">
;(function() {

window.jsPlumbFDA = {

init :function() {

jsPlumb.importDefaults({
Endpoint : ["Dot", {radius:2}],

```

```

HoverPaintStyle : {strokeStyle:"#6b6b6b", lineWidth:2 },
ConnectionOverlays : [
[ "Arrow", {
location:1,
id:"arrow",
length:14,
foldback:0.8
} ],
[ "Label", { id:"label", cssClass:"aLabel" } ]
]
});

$(".ep").each(function(i,e) {
var p = $(e).parent();
var pid = p[0].id;
console.log(pid+"="+i);

jsPlumb.makeSource($(e), {
parent:p,
//anchor:"BottomCenter",
anchor:"Continuous",
//paintStyle:{ fillStyle:"#ffffff",radius:0 },
connector:[ "StateMachine", { curviness:20 } ],
//connectorStyle:{ strokeStyle:conColor, lineWidth:2 },
maxConnections:5,
onMaxConnections:function(info, e) {
alert("Maximum connections (" + info.maxConnections + ")
reached");
}
});
});

jsPlumb.bind("beforeDrop", function(c) {
return false;
});

jsPlumb.makeTarget(jsPlumb.getSelector(".w"), {
dropOptions:{ hoverClass:"dragHover" },
paintStyle:{ fillStyle:"#ffffff",radius:0 },
anchor:"Continuous"
//anchor:"TopCenter"
});
});
});
</script>
<script type="text/javascript" src="<?php echo
base_url("js/html2canvas.js");?>"></script>
<div id="typeNavBar3" name="pathGroup" class="exp"
style="right:25px;top:300px;padding:10px;width:25em">
<table id="docloctable"></table>
<div id="pager"></div>
</div>

<div id="typeNavBar1" name="pathGroup" class="exp"
style="right:25px;top:5px;padding:10px;width:25em">
<font color="#aaa" size="6px" style="padding:10px"><?php echo
$arcDocumentMeta['TITLE']; ?></font><font color="#aaa"
size="3px" style="padding:10px"> (Archived)</font>
<table>
<TR>
<TD
style="width:100px;color:#555555;">Barcode</TD><TD><INPUT
TYPE="TEXT" NAME="BARCODE" ID="BARCODE" VALUE="<?php

```



```

echo "i".$cntr.".setPaintStyle({strokeStyle:\\". $conColor.\",
lineWidth:2},false);\n";
$cntr++;
endforeach;
?>
pop = true;
});

jQuery("#docloctable").jqGrid({
url:<?php
$doclocurl =
"index.php/document/getDocArcLocation/".$$arcDocumentMeta['
ARCCODE'];
echo "\\".base_url($doclocurl).\\"";
?>,
datatype: "json",
mtype : "post",
colNames:['Station', 'Time in','Time out'],
colModel:[
{name:'STATIONNAME',index:'STATIONNAME', width:110,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\\"<?php echo
base_url("index.php/document/viewStation");?>/\"+rowObject[0]
+\\">\" + cellvalue + \\"</a>\";
}
},
{name:'TIMEIN',index:'TIMEIN', width:117,
align:"right",sorttype:"text"},
{name:'TIMEOUT',index:'TIMEOUT', width:117,
align:"right",sorttype:"text"}
],
rowNum:20,
rowList:[10,20,30],
pager: "#pager",
sortname: 'TIMEIN',
viewrecords: true,
sortorder: "asc",
caption: "Document History"
});

jQuery("#docloctable").jqGrid('navGrid', '#pager',
{
edit:false,add:false,del:false
}, //options
{
}, // edit options
{
}, // add options
{
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
</script>

```

xxii. views/docutype/viewArcDocs.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:8
48px;background-color:white;border:2px solid #ddd; ;">
<div id="docSel" style="position:absolute;left:10px;top:10px;">
<?php
if ($this->session->userdata('usertype') != "assistant" ) {
?>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">ALL ACTIVE</label>
<?php
}
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

<input type="radio" id="radio1" name="radio" /><label
for="radio1">STATION</label>

<?php
}
?>
<input type="radio" id="radio3" name="radio"
checked="checked" /><label for="radio3">ARCHIVED</label>
</div>
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Documents</font><br /></div>
<table id="docutype"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});
$(function() {
$("#docSel").buttonset();
<?php
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

$("#radio1").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewStationDocs").\\"";?>;
});
<?php
}
if ($this->session->userdata('usertype') != "assistant" ) {
?>
$("#radio2").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewDocs").\\"";?>;
});
<?php
}
?>
$("#radio3").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewArcDocs").\\"";?>;
});
});

```

```

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left:-
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#docutype").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getArcDocElements")."\\";
?>,
datatype: "json",
mtype : "post",
colNames:['ARCCODE','Title','Type','Time Elapsed','Companies'],
colModel:[
{name:'ARCCODE',index:'ARCCODE', width:150 ,sortable:true},
{name:'TITLE',index:'TITLE', width:175,sortable:true,editable:true},
formatter:function (cellvalue, options, rowObject) {
return "<a href=\\"<?php echo
base_url("index.php/document/viewArcDoc");?>/\"+rowObject[0]
+\\">\" + cellvalue + "\</a>";
}
},
{name:'TYPENAME',index:'TYPENAME',
width:150,editable:true,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:105,sortable:true,searchoptions: { sopt: [] }},
{name:'COMPANIES',index:'COMPANIES',
width:236,sortable:false}
],
rowNum: 20,
height: 400,
rowList:[10,20,30],
pager: '#pager',
sortname: 'ARCCODE',
viewrecords: true,
sortorder: "desc",
caption: "Documents"
});

```

```

jQuery("#docutype").jqGrid('navGrid','#pager',
{
edit:false, del:false,
addfunc: function () {
window.location.href = "<?php echo
base_url("index.php/document/newDoc");?>";
}
}, //options
{
}, // edit options
{
}, // add options
{
url: <?php echo
"\\".base_url("index.php/document/deleteType")."\\"; ?>,
reloadAfterSubmit:true,
type: "POST",
width:300,
msg: "Are you sure you want to delete this type?",
delData: {
typecode:
function() {
var sel_id = jQuery("#docutype").jqGrid('getGridParam', 'selrow');
var value = jQuery("#docutype").jqGrid('getCell', sel_id,
'TYPECODE');
return value;
}
},
afterComplete:function(msg){
$("#saveParamStatus" ).empty();
$("#saveParamStatus" ).show();
$("#saveParamStatus" ).append(msg.responseText);
$("#saveParamStatus" ).fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);

$("#ui-jqgrid-titlebar").hide();
</script>

```

xxiii. views/docutype/viewArcForm.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:20px;padding:5px;width:700px;">
<div style="padding:10px;"><font size="6"
color="#b8b8b7"><?php echo
$documentMeta['TITLE']?></font><br /><br />
<table>
<?php
foreach ($params as $param) {
?>
<tr>
<td style="padding:5px 10px 5px 10px;">

```



```

<script type="text/javascript" src="<?php echo
base_url("js/setup.js");?>"></script>

<script type="text/javascript">
jsPlumb.bind("ready", function() {
<?php
foreach ($documentTypePath as $documentTypePath_item):
echo "jsPlumb.connect({
source:\"\".$documentTypePath_item['CONSOURCE'].\"\",
target:\"\".$documentTypePath_item['CONTARGET']
.\"\"}).getOverlay(\"label\").setLabel(
\"\".$documentTypePath_item['CONDESCR'].\" \");\n";
endforeach;
?>
pop = true;
});

function checkIfConnected(temp) {
var conn = new Array(temp.length);
var conn2 = new Array(temp.length);
var err = "";
var errMsgs1 = new Array();
var errMsgs2 = new Array();
var exemptions1 = new Array();
var exemptions2 = new Array();
var ind;
var startIndex = 0;

for (var j = 0; j < temp.length; j++) {
conn[j] = [temp[j].sourceId,temp[j].targetId,0];
conn2[j] = conn[j];
}

for (var i = 0; i < temp.length; i++) {
//console.log("index: "+i);
for (var j = 0; j < temp.length; j++) {
conn[j][2] = 0;
conn2[j][2] = 0;
}

if (!recursiveCheck2(conn2,i)) {
if ($.inArray(temp[i].sourceId, errMsgs2) < 0) {
errMsgs2.push(temp[i].sourceId);
}
} else {
if ($.inArray(temp[i].sourceId, exemptions2) < 0) {
exemptions2.push(temp[i].sourceId);
}
}
}

if (!recursiveCheck(conn,i)) {
if ($.inArray(temp[i].sourceId, errMsgs1) < 0) {
errMsgs1.push(temp[i].sourceId);
}
} else {
if ($.inArray(temp[i].sourceId, exemptions1) < 0) {
exemptions1.push(temp[i].sourceId);
}
}
}

for (var i = 0; i < errMsgs2.length; i++) {
if ($.inArray(errMsgs2[i], exemptions2) < 0) {
err+= "Cannot reach node "+errMsgs2[i]+" from start node.\n";
}
}
}

```

```

for (var i = 0; i < errMsgs1.length; i++) {
if ($.inArray(errMsgs1[i], exemptions1) < 0) {
err+= "Cannot reach end node from node "+errMsgs1[i]+".\n";
}
}

if (err != "") {
return err;
} else {
return "save";
}
}

function recursiveCheck(conn,index) {
//console.log(conn);
var to = conn[index][1];
if ( to == 0 || to == "0" ) {
return true;
} else {
for (var i = 0; i < conn.length; i++) {
//console.log("perm: "+conn[i][2]+" to: "+to+" | "+conn[i][1]);
if ( i != index && to == conn[i][0] && conn[i][2] == 0 ) {
//console.log("to: "+to+" | "+conn[i][1]);
conn[index][2] = 1;
if( recursiveCheck(conn,i) ) {
return true;
}
}
}
return false;
}
}

function recursiveCheck2(conn,index) {
//console.log(conn);
var from = conn[index][0];
if ( from == -1 || from == "-1" ) {
return true;
} else {
for (var i = 0; i < conn.length; i++) {
//console.log("perm: "+conn[i][2]+" from: "+from+" | "+conn[i][1]);
if ( i != index && from == conn[i][1] && conn[i][2] == 0 ) {
//console.log("from: "+from+" | "+conn[i][1]);
conn[index][2] = 1;
if( recursiveCheck2(conn,i) ) {
return true;
}
}
}
return false;
}
}

function saveStationPath() {
var startConn = 0;
var endConn = 0;

var temp = jsPlumb.getConnections();

var statusMsg = checkIfConnected(temp);
if ( statusMsg == "save" ) {
if(confirm("Are you sure you want to save changes?")){
var temp = jsPlumb.getConnections();
}
}
}

```



```

<!-- more js codes for grid -->

<script type="text/javascript">

function updateTips( t ) {
$( "#validateTips" ).text( t ).addClass( "ui-state-highlight" );
setTimeout(function() {
$( "#validateTips" ).removeClass( "ui-state-highlight", 1500 );
}, 500 );
}

function checkRegexp( o, regexp, n ) {
if ( !( regexp.test( o.val() ) ) ) {
o.addClass( "ui-state-error" );
updateTips( n );
return false;
} else {
return true;
}
}

var mydata = [
<?php

foreach ( $pathInstances as $pathInstances_item ):
echo "{barcode:\"" . $pathInstances_item['BARCODE'] . "\",
title:\"" . $pathInstances_item['TITLE'] . "\", loc: \""
.$pathInstances_item['STATIONNAME'] . "\",
desc:\"" . $pathInstances_item['DOCDESCR'] . "\",";
echo "\n";
endforeach;

?>
];

jQuery("#instance").jqGrid({
datatype: "local",
colNames:['Barcode','Title', 'Location','Description'],
colModel:[
{name:'barcode',index:'barcode', width:60, sorttype:"int"},
{name:'title',index:'title', width:90, sorttype:"text"},
{name:'loc',index:'loc', width:150, align:"right",sorttype:"text"},
{name:'desc',index:'desc', width:155, align:"right",sorttype:false}
],
rowNum:20,
rowList:[10,20,30],
pager: '#pager',
sortname: 'barcode',
viewrecords: true,
sortorder: "desc",
loadonce: true,
caption: "Instances"
});

for(var i=0;i<=mydata.length;i++)
jQuery("#instance").jqGrid('addRowData',i+1,mydata[i]);
</script>

<!-- end js codes for grid -->

```

xxvi. views/docutype/viewDocType.php

```

<div name="pathGroup" id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:467px">
<font size="6" color="#b8b8b7"><?php echo
$documentTypeMeta['TYPENAME']; ?></font><button
id="docTypeEdit" style="font-size: 12px; position:absolute;
right:15px;">Edit</button>
<br /><br />
<font size="2" color="#000000"><?php echo
$documentTypeMeta['TYPEDESC']; ?></font>
</div>

<div name="pathGroup" style="position:absolute">
<div id="boxes">
<?php
//print start node
echo "<div id=\"-1\" class=\"w\" style=\"left:180px; top:0px;
background-color:#baff80;\">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
color=#000000>start</font><div class=\"ep\"
style=\"width:0em;height:0em;\"></div></div>\n";

foreach ( $documentTypeStation as $documentTypeStation_item ):
echo "<div
id=\"" . $documentTypeStation_item['STATIONCODE'] . "\"
class=\"w\" style=\"left:\"" . $documentTypeStation_item['LEFTPOS']
."px;
top:\"" . $documentTypeStation_item['TOPPOS'] . "\"px;\">\". $documen
tTypeStation_item['STATIONCODE'];
echo "<a
href=\"" . base_url("index.php/document/viewStation")."/\". $docu
mentTypeStation_item['STATIONCODE'] . "\"><font
color=#000000'
size=\"2\">\". $documentTypeStation_item['STATIONNAME'] .\"</fo
nt></a>";
echo "<div class=\"ep\"
style=\"width:0em;height:0em;\"></div><div style=\"background-
color: ";
echo $documentTypeStation_item['color'];
echo ";float:right;width:55px;height:20px;text-
align:center;\">\". $documentTypeStation_item['docCount'] .\"</div
></div>\n";
endforeach;

//print end node
echo "<div id=\"0\" class=\"w\" style=\"left:550px; top:0px;
background-color:#feb3b3;\">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<font
color=#000000>end</font></div>\n";
?>
</div>
</div>

<!-- js codes -->

<script type="text/javascript">
var pop = false;
</script>
<script type="text/javascript" src="<?php echo
base_url("development-
bundle/ui/jquery.ui.tabs.js");?>"></script>
<script type="text/javascript">
;(function() {
var curColourIndex = 1, maxColourIndex = 24, nextColour =
function() {
var R,G,B;
R = parseInt(128+Math.sin((curColourIndex*3+0)*1.3)*128);
G = parseInt(128+Math.sin((curColourIndex*3+1)*1.3)*128);
B = parseInt(128+Math.sin((curColourIndex*3+2)*1.3)*128);

```



```

<input type="text" name="conname" id="conname" class="text
ui-widget-content ui-corner-all" />
</fieldset>
</div>

<!-- more js codes for grid -->

<script type="text/javascript">

function updateTips( t ) {
$( "#validateTips" ).text( t ).addClass( "ui-state-highlight" );
setTimeout(function() {
$( "#validateTips" ).removeClass( "ui-state-highlight", 1500 );
}, 500 );
}

function checkRegexp( o, regexp, n ) {
if ( !( regexp.test( o.val() ) ) ) {
o.addClass( "ui-state-error" );
updateTips( n );
return false;
} else {
return true;
}
}

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width: '+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#instance").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getTypePathInstances")."/".$
typeCode."\"";?>,
datatype: "json",
mtype : "post",

```

```

colNames:['Barcode','Title', 'Location','Duration','Status'],
colModel:[
{name:'BARCODE',index:'BARCODE', width:0 ,hidden:true},
{name:'TITLE',index:'TITLE', width:115,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url('index.php/document/viewDoc');?>/\"+rowObject[0]+\"
>\" + cellvalue + \"</a>\";
}
},
{name:'STATIONNAME',index:'STATIONNAME',
width:100,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:140,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
],
rowNum: 10,
height: 200,
rowList:[10,20,30],
pager: '#pager2',
sortname: 'TITLE',
viewrecords: true,
sortorder: "desc",
caption: "Type Instances"
});

jQuery("#instance").jqGrid('navGrid','#pager2',
{
edit:false,add:false,del:false
}, //options
{
}, // edit options
{
}, // add options
{
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);

jQuery("#parameters").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getParamElements/$typeCod
e").\"";?>,
datatype: "json",
mtype : "post",
colNames:['paramcode','Order','Name','Input Type', 'Options',
'Col Size', 'Default Value', 'Description'],
colModel:[
{name:'PARAMCODE',index:'PARAMCODE', width:0,
sorttype:"int",editable:true,hidden:true},
{name:'PARAMORDER',index:'PARAMORDER', width:50,
sorttype:"int",editable:true},
{name:'PARAMNAME',index:'PARAMNAME', width:100,
sorttype:"text",editable:true},
{name:'INPUTTYPE',index:'INPUTTYPE',
width:100,sorttype:"text",editable:true},

```



```

<div id="one" class="exp"
style="position:absolute;left:220px;top:10px;padding:20px;height
:80px;width:792px;background-color:white;border:2px solid
#ddd;">
<div id="metadata">
<font size="6" color="#b8b8b7"><?php echo
$stationMetaData['STATIONNAME']; ?></font>
<br>
<font size="2" color="#000000"><?php echo
$stationMetaData['STATIONDESC']; ?></font>
</div>
</div>

<div id="two"
style="position:absolute;left:220px;top:160px;padding:5px;width:
382px;background-color:white;border:2px solid #ddd;">
<table id="users"></table>
<div id="pager2"></div>
</div>

<?php
if ($this->session->userdata('usertype') != "assistant") {
?>
<div id="three"
style="position:absolute;left:625px;top:160px;padding:5px;width:
420px;background-color:white;border:2px solid #ddd;">
<table id="documents"></table>
<div id="pager3"></div>
</div>
<?php
}
?>

<script type="text/javascript">
$(function() {
$("#one,#two,#three").css("display", "none");
$("#one,#two,#three").fadeIn(1000);
$("input:submit, button").button();
});

<?php
if ($this->session->userdata('usertype') != "assistant") {
?>

var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

```

```

function componentToHex(c) {
c = Math.floor((c/100)*255);
var numClr = c.toString(16);
return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
return "#" + componentToHex(r) + componentToHex(g) +
componentToHex(b);
}

jQuery("#documents").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getStationDocElements")."/".
$stationMetaData['STATIONCODE']."\\"; ?>,
datatype: "json",
mtype: "post",
colNames:['Barcode','Title','Type','Time Elapsed','Progress'],
colModel:[
{name:'BARCODE',index:'BARCODE', width:0 ,hidden:true},
{name:'TITLE',index:'TITLE', width:105,sortable:true,editable:true,
formatter:function (cellvalue, options, rowObject) {
return "<a href=\"<?php echo
base_url("index.php/document/viewDoc");?>/\"+rowObject[0]+\"\\
>\" + cellvalue + \"</a>\";
}
},
{name:'TYPENAME',index:'TYPENAME',
width:93,editable:true,sortable:true},
{name:'ELAPSED',index:'ELAPSED',
width:100,sortable:true,searchoptions: { sopt: [] }},
{name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
formatter: function (cellvalue) {
return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
}
},
rowNum: 30,
height: 300,
rowList:[10,20,30],
pager: '#pager3',
sortname: 'TITLE',
viewrecords: true,
sortorder: "desc",
caption: "Documents"
});

jQuery("#documents").jqGrid('navGrid','#pager3',
{edit:false,add:false,del:false}, //options
{}, // edit options
{}, // add options
{}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
<?php
?>

jQuery("#users").jqGrid({

```

```

url:<?php echo
"\\".base_url("index.php/document/getUserListStation")."/".$stat
ionMetaData['STATIONCODE']."\\";?>,
datatype: "json",
mtype : "post",
colNames:['UserID','Username','Type','First Name','Last Name'],
colModel:[
{name:'USERID',index:'USERID', width:0,hidden:true},
{name:'USERNAME',index:'USERNAME',
width:90,editable:true,editrules: { required: true}},
{name:'USERTYPE',index:'USERTYPE',
width:90,editable:true,editrules: { required: true}},
{name:'FIRSTNAME',index:'FIRSTNAME',
width:90,editable:true,editrules: { required: true}},
{name:'LASTNAME',index:'LASTNAME',
width:90,editable:true,editrules: { required: true}},
],
rowNum: 20,
height: 300,
rowList:[10,20,30],
pager: '#pager2',
sortname: 'USERNAME',
viewrecords: true,
sortorder: "desc",
caption: "Users"
});

jQuery("#users").jqGrid('navGrid','#pager2',
{edit:false,add:false,del:false}, //options
{}, // edit options
{}, // add options
{}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);
</script>

```

xxix. views/docuType/viewStationDocs.php

```

<div id="typeParam"
style="position:absolute;left:220px;top:10px;padding:5px;width:8
48px;background-color:white;border:2px solid #ddd; ">
<div id="docSel" style="position:absolute;left:10px;top:10px;">
<?php
if ($this->session->userdata('usertype') != "assistant" ) {
?>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">ALL ACTIVE</label>
<?php
}
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

<input type="radio" id="radio1" name="radio"
checked="checked"/><label for="radio1">STATION</label>

<?php
}
?>

```

```

<input type="radio" id="radio3" name="radio" /><label
for="radio3">ARCHIVED</label>
</div>
<div style="padding:10px;" align="right"><font size="6"
color="#b8b8b7">Documents</font><br /></div>
<table id="docuType"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

<script type="text/javascript">
$(function() {
$("#typeParam").css("display", "none");
$("#typeParam").fadeIn(1000);
$("input:submit, button").button();
});
$(function() {
$("#docSel").buttonset();
<?php
if ($this->session->userdata('usertype') == "head" || $this-
>session->userdata('usertype') == "assistant" ) {
?>

$("#radio1").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewStationDocs")."\\";?>
});
<?php
}
if ($this->session->userdata('usertype') != "assistant" ) {
?>
$("#radio2").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewDocs")."\\";?>
});
<?php
}
?>
$("#radio3").click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewArcDocs")."\\";?>
});
});
var gradientNumberFormat = function (cellvalue, gradientClass,
minDataValue,
maxDataValue, minDisplayValue, maxDisplayValue) {
var dataAsNumber = parseFloat(cellvalue); /* parseInt(cellvalue,
10);*/
if (dataAsNumber > maxDataValue) {
dataAsNumber = maxDataValue;
}
if (dataAsNumber < minDataValue) {
dataAsNumber = minDataValue;
}

var colour = rgbToHex(cellvalue,100-cellvalue,0);

var prozentVal = minDisplayValue+(dataAsNumber-
minDataValue)*(maxDisplayValue-
minDisplayValue)/(maxDataValue-minDataValue);
return '<div class="cellDiv"><div style="position: absolute; left: -
2px; top:-5px; right: 2px; height:22px; float:left; width:'+
prozentVal+'%; background: '+colour+'; "></div><div
class="cellTextRight">'+cellvalue +'</div></div>';
};

```

```

function componentToHex(c) {
    c = Math.floor((c/100)*255);
    var numClr = c.toString(16);
    return numClr.length == 1 ? "0" + numClr : numClr;
}

function rgbToHex(r, g, b) {
    return "#" + componentToHex(r) + componentToHex(g) +
    componentToHex(b);
}

jQuery("#doctype").jqGrid({
    url:<?php echo
    "\".base_url("index.php/document/getDocStationElements")."/".
    $this->session->userdata('stationcode')."\""; ?>,
    datatype: "json",
    mtype: "post",
    colNames:['Lock','Lock0','Barcode','Title','Type','Location','Time
    Elapsed','Progress','Companies'],
    colModel:[
    {name:'LOCK0',index:'LOCK0', width:0,hidden:true},
    {name:'DOCUD.LOCK',index:'DOCUD.LOCK', width:30
    ,sortable:false,
    formatter:function (cellvalue, options, rowObject) {
    if ( cellvalue == "y" || cellvalue == 'y' ) {
    return "<input id=\"checkbox\" + rowObject[2] + \"\
    type=\"checkbox\" onclick=\"toggleCheckbox('checkbox\" +
    rowObject[2] + \"\",\"\" + rowObject[2] + \"\")\" value=\" + cellvalue +
    \"checked\" checked\">";
    } else {
    return "<input id=\"checkbox\" + rowObject[2] + \"\
    type=\"checkbox\" onclick=\"toggleCheckbox('checkbox\" +
    rowObject[2] + \"\",\"\" + rowObject[2] + \"\")\" value=\" + cellvalue +
    \">";
    }
    }
    },
    {name:'DOCUD.BARCODE',index:'DOCUD.BARCODE', width:50
    ,sortable:true},
    {name:'TITLE',index:'TITLE', width:125,sortable:true,editable:true,
    formatter:function (cellvalue, options, rowObject) {
    return "<a href=\"<?php echo
    base_url(\"index.php/document/viewDoc\");?>/\"+rowObject[2]+\"\"
    >\" + cellvalue + "</a>";
    }
    },
    {name:'TYPENAME',index:'TYPENAME',
    width:150,editable:true,sortable:true},
    {name:'STATIONNAME',index:'STATIONNAME',
    width:100,sortable:true,hidden:true},
    {name:'ELAPSED',index:'ELAPSED',
    width:140,sortable:true,searchoptions: { sopt: [] }},
    {name:'PROGRESS',index:'PROGRESS', width:100,sortable:true,
    searchoptions: { sopt: ['eq','ne','lt','le','gt','ge','nu','nn'] },
    formatter: function (cellvalue) {
    return gradientNumberFormat(cellvalue,"gradient1",0,100,0,100);
    }
    },
    {name:'COMPANIES',index:'COMPANIES',
    width:220,sortable:false}
    ],
    rowNum: 20,
    height: 400,
    rowList:[10,20,30],
    pager: '#pager',

```

```

    sortname: 'ELAPSED',
    viewrecords: true,
    sortOrder: "desc",
    caption: "Documents"
    });

function toggleCheckbox(cbid,barcd){
$.ajax({
    type: "POST",
    url: <?php echo
    "\".base_url("index.php/document/saveLock").\"\""; ?>,
    data: { barCode : barcd, value: jQuery("#"+cbid).attr("checked") },
    success: function(msg){
    $( "#saveParamStatus" ).empty();
    $( "#saveParamStatus" ).show();
    $( "#saveParamStatus" ).append(msg);
    $( "#saveParamStatus" ).fadeOut(6000);
    }
    });
}

jQuery("#doctype").jqGrid('navGrid','#pager',
{
    editfunc: function (rowid) {
    var lk = jQuery("#doctype").jqGrid('getCell',
    jQuery("#doctype").jqGrid('getGridParam', 'selrow'), 'LOCK0');
    if( lk == "y" || lk == 'y' ){
    window.location.href = "<?php echo
    base_url(\"index.php/document/formedit\");?>\"+\"/\"+jQuery("#doc
    utable").jqGrid('getCell',
    jQuery("#doctype").jqGrid('getGridParam', 'selrow'),
    'DOCUD.BARCODE');
    } else {
    alert("Lock document first.");
    }
    },
    addfunc: function () {
    window.location.href = "<?php echo
    base_url(\"index.php/document/newDoc\");?>";
    }
    }, //options
    {
    }, // edit options
    {
    }, // add options
    {
    url: <?php echo
    "\".base_url("index.php/document/deleteDoc").\"\""; ?>,
    reloadAfterSubmit:true,
    type: "POST",
    width:300,
    msg: "Are you sure you want to delete this document?",
    delData: {
    barcode:
    function() {
    var sel_id = jQuery("#doctype").jqGrid('getGridParam', 'selrow');
    var value = jQuery("#doctype").jqGrid('getCell', sel_id,
    'DOCUD.BARCODE');
    return value;
    }
    }
    },

    afterComplete:function(msg){
    $( "#saveParamStatus" ).empty();
    $( "#saveParamStatus" ).show();

```



```

$( "#saveParamStatus" ).append(msg.responseText);
$( "#saveParamStatus" ).fadeOut(6000);
}
}, // del options
{
sopt:['cn','eq','ne','lt','le','gt','ge','nu','nn'],
reloadAfterSubmit: true,
closeAfterSearch: true
} // search options
);

$( ".ui-jqgrid-titlebar" ).hide();
</script>

```

xxx. views/docutype/viewTypeMeta.php

```

<div id="typeMeta" class="exp"
style="left:220px;top:100px;padding:10px;width:700px;">
<div id="saveMetaStatus"></div>
<table>
<TR>
<TD style="width:100px;">Type Name</TD><TD><INPUT
TYPE="TEXT" NAME="TYPENAME" ID="TYPENAME"
VALUE="<?php echo $documentTypeMeta['TYPENAME']; ?>"
READONLY size=90></TD>
</TR>
<TR>
<TD style="width:100px;">Type Desc</TD><TD><TEXTAREA
TYPE="TEXT" NAME="TYPEDESC" ID="TYPEDESC" cols="72"
rows="4" READONLY><?php echo
$documentTypeMeta['TYPEDESC']; ?></TEXTAREA></TD>
</TR>
<TR>
<TD colspan="2" align="right">
<button id="typeSaveMeta" style="font-size:
12px;">save</button>
</TD>
</TR>
</table>
</div>

```

```

<div id="typeNavBar" class="exp"
style="right:15px;top:5px;padding:10px;width:33em">
<div id="editDocTypeSel">
<input type="radio" id="radio1" name="radio"
checked="checked"/><label for="radio1">METADATA</label>
<input type="radio" id="radio2" name="radio" /><label
for="radio2">PATH</label>
<input type="radio" id="radio3" name="radio" /><label
for="radio3">PARAMETERS</label>
</div>
</div>

```

```

<script type="text/javascript">
$(function() {
$( "#typeMeta" ).css("display", "none");
$( "#typeMeta" ).fadeIn(1000);
$( "input:submit, button" ).button();

```

```

$( "#typeSaveMeta" ).button().click(function() {
$.ajax({
type: "POST",
url: <?php echo
"\\".base_url("index.php/document/saveTypeMeta").\\""; ?>,

```

```

data: { typeCode : <?php echo "\\".typeCode.\\"";?>, name :
$( "#TYPENAME" ).val() , desc: $( "#TYPEDESC" ).text() },
success: function(msg){
$( "#saveMetaStatus" ).empty();
$( "#saveMetaStatus" ).append(msg+"<hr /><br />");
}
});
});

$( "#editDocTypeSel" ).buttonset();

```

```

$( "#radio1" ).click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewTypeMeta"/).\\".typeCode)
.\\"";?>;
});
$( "#radio2" ).click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewsetupworkflow"/).\\".typeC
ode).\\"";?>;
});
$( "#radio3" ).click(function() {
window.location = <?php echo
"\\".base_url("index.php/document/viewTypeParam"/).\\".typeCode)
.\\"";?>;
});
});
</script>

```

xxxi. views/docutype/viewUsers.php

```

<div id="userList"
style="position:absolute;left:220px;top:10px;padding:5px;width:9
45px;background-color:white;border:2px solid #ddd; ;">
<div style="padding:10px;" align="right"><div id="buttoncont"
style="position:absolute; left:15px;"><button id="applicantEdit"
style="font-size: 12px;">Assign</button></div><font size="6"
color="#b8b8b7">Users</font><br /></div>
<table id="users"></table>
<div id="pager"></div>
<div id="saveParamStatus"></div>
</div>

```

```

<script type="text/javascript">
$(function() {
$( "#userList" ).css("display", "none");
$( "#userList" ).fadeIn(1000);
$( "input:submit, button" ).button();
jQuery("#buttoncont").hide();
});

$( "#applicantEdit" ).button().click(function() {
window.location.href = "<?php echo
base_url("index.php/document/assignApplicant");?>"+"/"+jQuery
("#users").jqGrid('getCell',
jQuery("#users").jqGrid('getGridParam', 'selrow'), 'USERID');
});

```

```

jQuery("#users").jqGrid({
url:<?php echo
"\\".base_url("index.php/document/getUserList").\\""; ?>,
datatype: "json",
mtype : "post",

```


xxxii. Sql/core.sql

```
CREATE DATABASE DOCUTRACKER2;
USE DOCUTRACKER2;
```

```
CREATE TABLE IF NOT EXISTS `ci_sessions` (
  session_id varchar(40) DEFAULT '0' NOT NULL,
  ip_address varchar(45) DEFAULT '0' NOT NULL,
  user_agent varchar(120) NOT NULL,
  last_activity int(10) unsigned DEFAULT 0 NOT NULL,
  user_data text NOT NULL,
  PRIMARY KEY (session_id),
  KEY `last_activity_idx` (`last_activity`)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUTYPE (
  TYPECODE INT NOT NULL
  AUTO_INCREMENT,
  TYPENAME VARCHAR(40) NOT NULL,
  TYPEDESC VARCHAR(500) NOT NULL,
  UNIQUE (TYPENAME),
  PRIMARY KEY (TYPECODE)
);
```

```
CREATE TABLE IF NOT EXISTS DATATYPE (
  DATATYPECODE INT NOT NULL AUTO_INCREMENT,
  DATATYPENAME VARCHAR(40) NOT NULL,
  UNIQUE (DATATYPENAME),
  PRIMARY KEY (DATATYPECODE)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUCOREMETADATA (
  BARCODE INT NOT NULL AUTO_INCREMENT,
  USERID INT
  NOT NULL,
  TITLE VARCHAR(40) NOT NULL,
  TYPECODE INT NOT NULL,
  STEP INT NOT NULL,
  DOCDESCR VARCHAR(500) NOT NULL,
  UNIQUE (TITLE),
  PRIMARY KEY (BARCODE),
  FOREIGN KEY (TYPECODE) REFERENCES DOCUTYPE (TYPECODE),
  FOREIGN KEY (USERID) REFERENCES USERACCOUNT (USERID)
);
```

```
CREATE TABLE IF NOT EXISTS COMPANYDOCU (
  BARCODE INT NOT NULL,
  COMPANYCODE INT
  NOT NULL,
  PRIMARY KEY (BARCODE,COMPANYCODE),
  FOREIGN KEY (BARCODE) REFERENCES
  DOCUMENTCOREMETADATA (BARCODE),
  FOREIGN KEY (COMPANYCODE) REFERENCES
  COMPANYCONTACTDETAILS (COMPANYCODE)
);
```

```
);
```

```
CREATE TABLE IF NOT EXISTS COMPANY (
  COMPANYCODE INT
  NOT NULL AUTO_INCREMENT,
  COMPANYNAME VARCHAR(40) NOT NULL,
  COMPANYDESC VARCHAR(500) NOT NULL,
  EMAIL VARCHAR(40) NOT NULL,
  PHONE VARCHAR(40) NOT NULL,
  ADDRESS VARCHAR(100) NOT NULL,
  UNIQUE (COMPANYNAME),
  PRIMARY KEY (COMPANYCODE)
);
```

```
CREATE TABLE IF NOT EXISTS STATION (
  STATIONCODE INT
  NOT NULL AUTO_INCREMENT,
  STATIONNAME VARCHAR(40) NOT NULL,
  STATIONDESC VARCHAR(500) NOT NULL,
  UNIQUE (STATIONNAME),
  PRIMARY KEY (STATIONCODE)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUPATHSTATION (
  TYPECODE INT NOT NULL,
  STATIONCODE INT
  NOT NULL,
  LEFTPOS INT
  NOT NULL,
  TOPPOS INT
  NOT NULL,
  PRIMARY KEY (TYPECODE,STATIONCODE),
  FOREIGN KEY (TYPECODE) REFERENCES DOCUTYPE (TYPECODE),
  FOREIGN KEY (STATIONCODE) REFERENCES STATION
  (STATIONCODE)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUPATHCON (
  TYPECODE INT NOT NULL,
  CONSOURCE INT
  NOT NULL,
  CONTARGET INT
  NOT NULL,
  CONDESCR VARCHAR(80) NOT NULL,
  PRIMARY KEY (TYPECODE,CONSOURCE,CONTARGET),
  FOREIGN KEY (TYPECODE) REFERENCES DOCUTYPE (TYPECODE)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUMETAPATHSTATION (
  BARCODE INT NOT NULL,
  STATIONCODE INT
  NOT NULL,
  METALEFTPOS INT
  NOT NULL,
  METATOPPOS INT
  NOT NULL,
  PRIMARY KEY (BARCODE,STATIONCODE),
  FOREIGN KEY (BARCODE) REFERENCES DOCUCOREMETADATA
  (BARCODE),
  FOREIGN KEY (STATIONCODE) REFERENCES STATION
  (STATIONCODE)
);
```

```
CREATE TABLE IF NOT EXISTS DOCUMETAPATHCON (
  BARCODE INT NOT NULL,
```

```

METACONSOURCE INT NOT
NULL,
METACONTARGET INT NOT
NULL,
METACONDESCR VARCHAR(80) NOT NULL,
PRIMARY KEY (BARCODE,METACONSOURCE,METACONTARGET),
FOREIGN KEY (BARCODE) REFERENCES DOCUCOREMETADATA
(BARCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCULOC (
BARCODE INT NOT NULL,
STATIONCODE INT
NOT NULL,
STEPORDER INT
NOT NULL,
TIMEIN DATETIME NOT NULL default
'0000-00-00 00:00:00',
TIMEOUT DATETIME NOT NULL default
'0000-00-00 00:00:00',
PRIMARY KEY (BARCODE,STATIONCODE,STEPORDER),
FOREIGN KEY (BARCODE) REFERENCES DOCUCOREMETADATA
(BARCODE),
FOREIGN KEY (STATIONCODE) REFERENCES STATION
(STATIONCODE)
);

```

```

CREATE TABLE IF NOT EXISTS PARAM (
PARAMCODE INT AUTO_INCREMENT,
TYPECODE INT NOT NULL,
INPUTTYPE VARCHAR(10)
NOT NULL default 'text',
DATATYPE VARCHAR(80) NOT NULL,
PARAMNAME VARCHAR(80)
NOT NULL,
PARAMORDER INT
NOT NULL,
DEFAULTVAL VARCHAR(80) NOT NULL,
COLSIZE INT
NOT NULL,
PARAMDESC VARCHAR(200) NOT NULL,
PRIMARY KEY (PARAMCODE,TYPECODE),
FOREIGN KEY (TYPECODE) REFERENCES DOCUTYPE (TYPECODE)
);

```

```

CREATE TABLE IF NOT EXISTS METAPARAM (
METAPARAMCODE INT AUTO_INCREMENT,
BARCODE INT NOT NULL,
INPUTTYPE VARCHAR(10)
NOT NULL default 'text',
DATATYPE VARCHAR(80) NOT NULL,
PARAMNAME VARCHAR(80)
NOT NULL,
PARAMORDER INT
NOT NULL,
COLSIZE INT
NOT NULL,
STRINGVAL VARCHAR(80) NOT NULL,
PARAMDESC VARCHAR(200) NOT NULL,
PRIMARY KEY (METAPARAMCODE),
FOREIGN KEY (BARCODE) REFERENCES DOCUCOREMETADATA
(BARCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCLOCK (
USERID INT NOT NULL,

```

```

BARCODE INT NOT NULL,
STATUS VARCHAR(80)
NOT NULL,
PRIMARY KEY (BARCODE,USERID),
FOREIGN KEY (BARCODE) REFERENCES DOCUCOREMETADATA
(BARCODE),
FOREIGN KEY (USERID) REFERENCES USERACCOUNT (USERID)
);

```

```

CREATE TABLE IF NOT EXISTS PARAMCHOICE (
CHOICECODE INT AUTO_INCREMENT,
PARAMCODE INT NOT NULL,
CHOICENAME VARCHAR(80)
NOT NULL,
PRIMARY KEY (CHOICECODE),
FOREIGN KEY (PARAMCODE) REFERENCES PARAM (PARAMCODE)
);

```

```

CREATE TABLE IF NOT EXISTS METAPARAMCHOICE (
METACHOICECODE INT AUTO_INCREMENT,
METAPARAMCODE INT NOT NULL,
METACHOICENAME VARCHAR(80) NOT NULL,
PRIMARY KEY (METACHOICECODE),
FOREIGN KEY (METAPARAMCODE) REFERENCES METAPARAM
(METAPARAMCODE)
);

```

```

CREATE TABLE IF NOT EXISTS PERMISSION (
PARAMCODE INT NOT NULL,
STATIONCODE INT NOT NULL,
PERTYPE VARCHAR(80)
NOT NULL,
PRIMARY KEY (PARAMCODE,STATIONCODE),
FOREIGN KEY (STATIONCODE) REFERENCES STATION
(STATIONCODE),
FOREIGN KEY (PARAMCODE) REFERENCES PARAM (PARAMCODE)
);

```

```

CREATE TABLE IF NOT EXISTS CREATEPERMISSION (
TYPECODE INT NOT NULL,
STATIONCODE INT NOT NULL,
PERTYPE VARCHAR(80)
NOT NULL,
PRIMARY KEY (TYPECODE,STATIONCODE),
FOREIGN KEY (STATIONCODE) REFERENCES STATION
(STATIONCODE),
FOREIGN KEY (TYPECODE) REFERENCES DOCUTYPE (TYPECODE)
);

```

```

CREATE TABLE IF NOT EXISTS METAPERMISSION (
METAPARAMCODE INT NOT NULL,
STATIONCODE INT NOT NULL,
PERTYPE VARCHAR(80)
NOT NULL,
PRIMARY KEY (METAPARAMCODE,STATIONCODE),
FOREIGN KEY (STATIONCODE) REFERENCES STATION
(STATIONCODE),
FOREIGN KEY (METAPARAMCODE) REFERENCES METAPARAM
(METAPARAMCODE)
);

```

```

CREATE TABLE IF NOT EXISTS USERACCOUNT (
USERID INT AUTO_INCREMENT,
USERTYPE VARCHAR(20) NOT NULL,
USERNAME VARCHAR(80)
NOT NULL,

```

```

PASSWORD          VARCHAR(80)
                NOT NULL,
STATIONCODE       INT      NOT NULL,
FIRSTNAME         VARCHAR(20) NOT NULL,
LASTNAME         VARCHAR(20) NOT NULL,
EMAIL            VARCHAR(40) NOT NULL,
PHONE            VARCHAR(20) NOT NULL,
ADDRESS          VARCHAR(80) NOT NULL,
THEME            VARCHAR(40)
                NOT NULL default 'smoothness',
UNIQUE (USERNAME),
PRIMARY KEY (USERID),
FOREIGN KEY (STATIONCODE) REFERENCES STATION
(STATIONCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCUCOREMETADATAARC (
ARCCODE          INT
                NOT NULL,
USERID           INT
                NOT NULL,
TITLE            VARCHAR(40)      NOT NULL,
TYPENAME        VARCHAR(40) NOT NULL,
STEP            INT      NOT NULL,
DOCDESCR       VARCHAR(500) NOT NULL,
PRIMARY KEY (ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCUMETAPATHSTATIONARC (
ARCCODE          INT      NOT NULL,
STATIONARCCODE  INT              NOT
NULL,
STATIONNAME     VARCHAR(40)      NOT
NULL,
METALEFTPOS     INT
                NOT NULL,
METATOPPOS      INT
                NOT NULL,
PRIMARY KEY (ARCCODE,STATIONARCCODE),
FOREIGN KEY (ARCCODE) REFERENCES DOCUCOREMETADATAARC
(ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCUMETAPATHCONARC (
ARCCODE          INT      NOT NULL,
METACONSOURCE   INT              NOT
NULL,
METACONTARGET   INT              NOT
NULL,
METACONDESCR    VARCHAR(80)      NOT NULL,
PRIMARY KEY (ARCCODE,METACONSOURCE,METACONTARGET),
FOREIGN KEY (ARCCODE) REFERENCES DOCUCOREMETADATAARC
(ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS DOCULOCAARC (
ARCCODE          INT
                NOT NULL,
STATIONNAME     VARCHAR(40)      NOT
NULL,
STEPORDER       INT
                NOT NULL,
TIMEIN          DATETIME NOT NULL default
'0000-00-00 00:00:00',
TIMEOUT         DATETIME NOT NULL default
'0000-00-00 00:00:00',

```

```

PRIMARY KEY (ARCCODE,STEPORDER),
FOREIGN KEY (ARCCODE) REFERENCES DOCUCOREMETADATAARC
(ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS COMPANYDOCUARC (
ARCCODE          INT      NOT NULL,
COMPANYNAME     VARCHAR(40)      NOT
NULL,
PRIMARY KEY (ARCCODE,COMPANYNAME),
FOREIGN KEY (ARCCODE) REFERENCES DOCUCOREMETADATAARC
(ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS METAPARAMARC (
METAPARAMARCCODE INT      NOT NULL,
ARCCODE          INT      NOT NULL,
INPUTTYPE       VARCHAR(10)
                NOT NULL default 'text',
DATATYPE        VARCHAR(80)      NOT NULL,
PARAMNAME       VARCHAR(80)
                NOT NULL,
PARAMORDER      INT
                NOT NULL,
COLSIZE         INT
                NOT NULL,
STRINGVAL       VARCHAR(80)      NOT NULL,
PARAMDESC      VARCHAR(200)      NOT NULL,
PRIMARY KEY (METAPARAMARCCODE),
FOREIGN KEY (ARCCODE) REFERENCES DOCUCOREMETADATAARC
(ARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS METAPARAMCHOICEARC (
METACHOICECODEARC INT      NOT NULL,
METAPARAMARCCODE  INT      NOT NULL,
METACHOICENAME    VARCHAR(80)      NOT NULL,
PRIMARY KEY (METACHOICECODEARC),
FOREIGN KEY (METAPARAMARCCODE) REFERENCES
METAPARAMARC (METAPARAMARCCODE)
);

```

```

CREATE TABLE IF NOT EXISTS COMPANYUSER (
COMPANYCODE      INT      NOT NULL,
USERID           INT      NOT NULL,
PRIMARY KEY (COMPANYCODE,USERID),
FOREIGN KEY (COMPANYCODE) REFERENCES COMPANY
(COMPANYCODE),
FOREIGN KEY (USERID) REFERENCES USERACCOUNT (USERID)
);

```

```

INSERT INTO USERACCOUNT ('USERTYPE', 'USERNAME',
'PASSWORD', 'STATIONCODE', 'FIRSTNAME', 'LASTNAME',
'EMAIL', 'PHONE', 'ADDRESS') VALUES
('admin', 'admin', MD5('admin' ), 0, 'Louie', 'Ano',
'jose_louie_mark@yahoo.com', '09229726892', '1048, Escoda St.
Paco, Manila');

```

xxxiii. js/list.js

```
;(function() {
```

```

var entries = [
  { id:"demo", name:"jsPlumb Home", about:"Main jsPlumb demo
page. Contains a bit of everything" },
  { id:"flowchartConnectorsDemo", name:"Flowchart Connectors",
  about:"Demonstration of the Flowchart connectors" },
  { id:"draggableConnectorsDemo", name:"Draggable Connections",
  about:"Demonstration of drag and drop connections" },
  { id:"perimeterAnchorsDemo", name:"Perimeter Anchors",
  about:"Demonstration of perimeter anchors" },
  { id:"stateMachineDemo", name:"State Machine",
  about:"Demonstration of the State Machine Connectors and
Continuous Anchors" },
  { id:"chartDemo", name:"Chart Demonstration", about:"Simple
Chart Demo" },
  { id:"anchorDemo", name:"Sources and Targets",
  about:"Demonstration of how to make whole elements
Connection sources and targets" },
  { id:"dynamicAnchorsDemo", name:"Dynamic Anchors",
  about:"Demonstration of Dynamic Anchors" },
  { id:"dragAnimDemo", name:"Animation", about:"Drag/drop
demo with animation" }
],
libraries = [
  {id:"jquery", name:"jQuery"},
  {id:"mootools", name:"MooTools"},
  {id:"yui3", name:"YUI3"}
],
prepareOtherLibraryString = function(demold, library) {
  var s = "", demoldInfo = jsPlumb.DemoList.find(demold);
  for (var i = 0; i < libraries.length; i++) {
    var c = libraries[i].id == library ? "selected" : "";
    s += '<a class="' + c + "' href=" + libraries[i].id + '/' + demold +
'.html"' + libraries[i].name + '</a>&nbsp;&nbsp;&nbsp;';
  }
  return s;
},
demoSelectorString = (function() {
  var s = '<h5>Select Demo:</h5><select id="demoSelector"
class="demoSelector">';
  for (var i = 0; i < entries.length; i++) {
    s += '<option value="' + entries[i].id + '.html"' + entries[i].name +
'</option>';
  }
  s += '</select>';
  return s;
})();

jsPlumb.DemoList = {
  find:function(id) {
    for (var i = 0; i < entries.length; i++) {
      if (entries[i].id === id) {
        var next = i < entries.length - 1 ? i + 1 : 0,
        prev = i > 0 ? i - 1 : entries.length - 1;

        return {
          current:entries[i],
          prev:entries[prev],
          next:entries[next],
          idx:i
        };
      }
    }
  },
  init : function() {
    var bod = document.body,

```

```

  demold = bod.getAttribute("data-demo-id"),
  library = bod.getAttribute("data-library"),
  renderModeString = '<h5>Render Mode:</h5><a href="#"
class="rmode" mode="svg">SVG</a>&nbsp;&nbsp;&nbsp;<a href="#"
class="selected rmode"
mode="canvas">Canvas</a>&nbsp;&nbsp;&nbsp;<a href="#"
class="rmode" mode="vml">VML</a>',
  libraryString = '<h5>Library:</h5><div
class="otherLibraries"></div>' +
  prepareOtherLibraryString(demold, library),
  demoldInfo = jsPlumb.DemoList.find(demold);

  if (demoldInfo) {
    var prevString = '<h5>Previous:</h5><a href="' +
demoldInfo.prev.id + '.html"' + demoldInfo.prev.name + '</a>',
    nextString = '<h5>Next:</h5><a href="' + demoldInfo.next.id +
'.html"' + demoldInfo.next.name + '</a>',
    menuString = '<div class="menu"><a href="..</doc>/usage.html"
class="mmlink">view documentation</a>' +
'&nbsp;&nbsp;&nbsp;<a href="..</apidocs">view api docs</a>' +
'&nbsp;&nbsp;&nbsp;<a href="..</tests/qunit-all.html">qUnit tests</a>' +
'&nbsp;&nbsp;&nbsp;<a href="mailto:simon.porritt@gmail.com"
class="mmlink">contact me</a>' +
'&nbsp;&nbsp;&nbsp;<a href="http://github.com/sporritt/jsplumb/"
class="mmlink">jsPlumb on GitHub</a>' +
'&nbsp;&nbsp;&nbsp;<a href="http://code.google.com/p/jsplumb/"
class="mmlink">jsPlumb on Google code</a>' +
'&nbsp;&nbsp;&nbsp;<a href="http://jsplumb.tumblr.com"
class="mmlink">jsPlumb on Tumblr</a>';

    var d = document.createElement("div");
    d.className = "renderMode";
    d.innerHTML = renderModeString + libraryString + prevString +
"<br/>" + nextString + demoSelectorString;
    bod.appendChild(d);

    var m = document.createElement("div");
    m.className = "menu";
    m.innerHTML = menuString;
    bod.appendChild(m);

    var ds = document.getElementById("demoSelector");
    ds.selectedIndex = demoldInfo.idx;
    ds.onchange = function() {
      window.location.href = ds.options[ds.selectedIndex].value;
    }
  }
})();

```

xxxiv. js/setup.js

```

$(function() {
  $("#dialog:ui-dialog").dialog("destroy");
  $("#dialogCon").dialog({autoOpen: false});

  $("div[name=pathGroup]").css("display", "none");
  $("div[name=pathGroup]").fadeIn(1000);
  $("input.submit, button").button();

  $("#addStation").click(function() {

```



```

$("#conname").remove();
$("#conField").append("<input type='text' name='conname'
id='conname' value='' + c.getOverlay("label").getLabel()
+'\" class='text ui-widget-content ui-corner-all' />");

$("#dialogCon" ).dialog({
autoOpen: false,
height: 170,
width: 380,
modal: true,
buttons: {
"Save": function() {
//if(confirm("Are you sure you want to save?")) {
var bValid = true;

//bValid = checkRegexp( $("#conname" ), /^[0-9a-zA-Z]+$/ ,
"Connection name only allow : a-z 0-9" );

if ( bValid ) {
c.getOverlay("label").setLabel( $("#conname" ).val() );
$( this ).dialog( "close" );
}
//}
},
"Delete": function() {
//if(confirm("Are you sure you want to delete?")) {
jsPlumb.detach(c);
$( this ).dialog( "close" );
//}
},
Cancel: function() {
$( this ).dialog( "close" );
}
},
close: function() {
//allFields.val( "" ).removeClass( "ui-state-error" );
}
});

$("#dialogCon" ).dialog( "open" );
});

// hand off to the library specific demo code here. not my ideal,
but to write common code
// is less helpful for everyone, because all developers just like to
copy stuff, right?
// make each ".ep" div a source and give it some parameters to
work with. here we tell it
// to use a Continuous anchor and the StateMachine connectors,
and also we give it the
// connector's paint style. note that in this demo the strokeStyle
is dynamically generated,
// which prevents us from just setting a
jsPlumb.Defaults.PaintStyle. but that is what i
// would recommend you do.
jsPlumbFDA.initEndpoints(nextColour);

jsPlumb.bind("jsPlumbConnection", function(conn) {

conn.connection.setPaintStyle({strokeStyle:nextColour});

if(pop){

$("#dialog:ui-dialog" ).dialog( "destroy" );

```

```

$("#validateTips" ).empty();
$("#conname").remove();
$("#conField").append("<input type='text' name='conname'
id='conname' value=''
+'\" class='text ui-widget-content ui-corner-all' />");

$("#dialogCon" ).dialog({
autoOpen: false,
height: 170,
width: 380,
modal: true,
buttons: {
"Save": function() {
//if(confirm("Are you sure you want to add?")) {
var bValid = true;

//bValid = checkRegexp( $("#conname" ), /^[0-9a-zA-Z]+$/ ,
"Connection name only allow : a-z 0-9" );

if ( bValid ) {
conn.connection.getOverlay("label").setLabel( $("#conname"
).val() );
$( this ).dialog( "close" );
}
//}
},
Cancel: function() {
jsPlumb.detach(conn.connection);
$( this ).dialog( "close" );
}
},
close: function() {
//allFields.val( "" ).removeClass( "ui-state-error" );
}
});

$("#dialogCon" ).dialog( "open" );
}

var temp = jsPlumb.getConnections();
var temp1 = "";
for(var i=0;i<temp.length;i++){
temp1 += temp[i].sourceId + '=' + temp[i].targetId + " ";
}
//alert(temp1);
});

jsPlumb.makeTarget(jsPlumb.getSelector(".w"), {
dropOptions:{ hoverClass:"dragHover" },
paintStyle:{ fillStyle:"#ffffff",radius:0 },
anchor:"Continuous"
//anchor:"TopCenter"
});
}
};
})();

xxxvi. js/stateMachiene-jquery.js

;(function() {

jsPlumbFDA.initEndpoints = function(nextColour) {
$("#.ep").each(function(i,e) {

```



```
var p = $(e).parent();
jsPlumb.makeSource$(e, {
  parent:p,
  //anchor:"BottomCenter",
  anchor:"Continuous",
  paintStyle:{ fillStyle:"#ffffff",radius:0 },
  connector:[ "StateMachine", { curviness:20 } ],
  connectorStyle:{ strokeStyle:nextColour(), lineWidth:2 },
  maxConnections:5,
  onMaxConnections:function(info, e) {
```

```
  alert("Maximum connections (" + info.maxConnections + ")
  reached");
  }
});
};
});
```

VI. Acknowledgement

I feel very fortunate and blessed to have done this Special Project (SP). This gave me a rare chance to make a contribution to our country through information technology. This is why “thank you” might not be enough to express my appreciation to everyone that helped me in every step of the way through the completion of this project. I owe my immense gratitude to the following.

First, to my adviser, Sir Geoffrey Solano, who took the word “adviser” into a whole new level. He believed in me and gave me the confidence no other mentor has ever given me before. That confidence fuels my drive to do this project. I’m really grateful for him for always finding time if I needed advice. Even if sometimes I haven’t set any schedule for advising, he is kind and patient enough to lend me his time if I approach him.

To Dr, Alvin Marcelo for giving me this project and connecting me to FDA. To former FDA Director, Dr. Suzette Lazo, who gave me the support I needed and the permission to go around FDA for numerous interviews. To FDA’s head in Information Technology, Mr. Roger Manapat, for finding time and personally working with me

To my ever supportive Mama and Papa who endured a lot of frustration because of me. Thank you for supporting me on everything I love to do. Thank you for being the home I could turn to when I’m down. Thank you for all the love, moral, spiritual and financial support, patience and daily reminders that I have an SP to finish.

To my friend, Renz, for lending me a hand on my SP defense. To the rest or the DOTA boys who are always ready to help me and providing me tips. To all my blockmates for adopting me to the block, and being supportive all on my academics which gave way to the completion of this SP.

To all my professors and lecturers, who provided me essential knowledge and skills needed for this SP.

Most of all, to God for giving me all the pieces to complete this SP. Thank you for answering my prayers and giving me more than what I have asked for.