

University of the Philippines Manila
College of Arts and Sciences
Department of Computer Science and Mathematics

**PregSS:
Pregnancy Surveillance System
Using Ontology for Interoperability**

A Special Problem
In Partial Fulfillment of Requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:
Darryl John P. Ong
2007-02375

April 2013

ACCEPTANCE SHEET

The Special Problem entitled “*PregSS: Pregnancy Surveillance System Using Ontology for Interoperability*” prepared and submitted by *Darryl John P. Ong* in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

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

EXAMINERS:

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

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

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

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

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

Abstract

PregSS: Pregnancy Surveillance System is an information system catering to the maternal care of women in barangay health centers in the Philippines. Since the maternal mortality rate of the country just increased prior the execution of this study, this aims to increase health care coverage by: being web-based, providing alerts on the patient's health risks while recommending the level of care, also, generating monthly reports on maternal health indicators, and finally, incorporating an ontology for interoperability with other possible health services. This is achieved by mapping an existing medical ontology, named OpenGALEN, to the local database through the use of an RDF (Resource Description Framework)-to-RDB (Relational Database) mapping tool called D2RQ.

Keywords: pregnancy, clinical information system, maternal care, ontology, database, mapping

Table of Contents

I.	Introduction	1
A.	Background of the Study	1
B.	Statement of the Problem	2
C.	Objectives	4
D.	Significance of the Study	7
E.	Scope and Limitations	8
F.	Assumptions.....	8
II.	Review of Related Literature	9
A.	Medical Ontologies.....	9
B.	Healthcare Information Systems using Ontologies	10
C.	Implementations of Ontology in Health Care	11
D.	Relational Database – Ontology Mapping.....	12
III.	Theoretical Framework	15
A.	Pregnancy and Maternal Health	15
B.	Home Based Mother’s Record.....	15
C.	FSHIS Monthly Form – Program Report (M1)	18
D.	Millennium Development Goals	20
E.	Information System	21
F.	Ontology	22
G.	OWL	24
H.	RDF and SPARQL, Linked Data.....	24
I.	Protégé	28
J.	OpenGALEN	29
K.	D2RQ Platform	30
L.	JavaServer Faces	31
IV.	Design and Implementation	32
A.	PregSS.....	32
1.	System Architecture	32
2.	Context Diagram	33
3.	Use-Case Diagram.....	34

4.	Entity Relationship Diagram.....	50
5.	Data Dictionary	52
V.	Technical Architecture	60
VI.	Results.....	61
A.	PregSS.....	61
B.	OpenGALEN to PregSS Mapping.....	80
C.	The D2RQ Endpoint	83
D.	Querying with SPARQL	87
VII.	Discussion	90
VIII.	Conclusion.....	91
IX.	Recommendation.....	92
X.	Bibliography	93
XI.	Appendix	96
	Source Code.....	99
	Source Package.....	99
	Web Content	162
	Mapping File.....	196
XII.	Acknowledgement.....	205

I. Introduction

A. Background of the Study

Advances in technology have provided us with more and more ways of communication. Now, almost every household has access to the internet, and that's why our healthcare services is utilizing this to increase its coverage.

An alarming statistic from the Department of Health (DOH) shows that the maternal mortality rate for the Philippines has increased from 162 per 100,000 live births in 2011, to 221 per 100,000 live births in 2012. This is evidently much farther from the 52 per 100,000 live births goal of the DOH, that is, included in the maternal health portion of the Millennium Development Goals (MDGs) for 2015. The MDGs are international goals which aim to eradicate poverty by achieving a set of targets, which are measurable, and usually include health or economic indicators [1]. Health Secretary Enrique Ona has noted that while this outcome was highly preventable through effective family planning, access to health services also plays a huge factor for decreasing this rate [2].

As the most convenient health service available for pregnant patients in rural areas are barangay health centers, a clinical application, which is highly available through the web, enables health care practitioners such as obstetricians or midwives to provide quality maternal care.

B. Statement of the Problem

In the barangay setting, a specialist or midwife may not be available every time. Regardless of that, a certain level of care (e.g. hospital care, close observation by physician) is identifiable based on the pregnant patient's symptoms. That's why a clinical decision-support system that alerts health personnel about the patient's health risks is needed. Not to mention that records on paper are nearly obsolete, since transmission of these data for statistical purposes is difficult, along with the increasing demand for sharing and re-usage of patient data [3].

As a result, maternal care provided in these areas increases, by enabling health personnel to give fast and sound actions. Also, by being able to generate maternal health indicator reports per barangay, the corresponding health sectors gets information on which areas need more or less support. Overall, maternal mortality rates cumulatively decreases and thus is nearing one of the MDGs for maternal care, which is to reduce the maternal mortality ratio by three quarters.

While an information system is sufficient in a local aspect, there is also the difficulty on how to transmit data to other systems for re-usage or demographic analyses. Locally, we have the Department of Health (DOH), which governs a wide array of Information and Communication Technology (ICT) services for health care. For technical interoperability and data integration, they have implemented the National Health Data Dictionary, which promotes data uniformity and consistency. Also, they already have plans for an Enterprise Architecture (EA), while currently evaluating both the SNOMED terminology and HL7 standards [4]. Despite that, most of their planned solutions are still for implementation and is focused towards technical interoperability. On the other hand, semantic interoperability

is also needed because of the seemingly arbitrary meaning of data across different health sectors, which may result to classification errors when collecting data. For example, DOH has an aggregate data collector called the electronic Field Health Services Information System (eFHSIS), which collects information on public health (e.g. maternal and child health, nutrition, family planning) from Rural Health Units (RHUs) and Barangay Health Stations. Since it collects data from different areas, there are standards and transmission protocols that are being followed. This restricts the flexibility of the systems that are able to provide data for the service. This is called the data evolution problem of information systems, where it's not easy to modify the database to reflect new business requirements. One way to solve this, which is Canlas recommends in his paper 'The Philippine Health Information System: Critical Challenges and Solutions', is by having a shared knowledge of the concept, such as an ontology, specifically, for pregnancy. By providing metadata for the fields in the system, data is more meaningful, and thus, can easily be inferred with by computers [5]. Also, a major benefit of this is, by only being a translation table, it allows different systems to easily exchange data even if they are using different coding standards [6].

C. Objectives

The Pregnancy Surveillance System (PregSS) gathers information from the pregnant patient based on the Department of Health (DOH) Home Based Mother's Record, and then evaluates those using risk factors so that it can recommend a mode of action. Also, a report system based on DOH's Field Health Service Information System (FSHIS) Monthly Form for barangay health centers is available for generation of statistical reports on maternal health indicators. In addition, it incorporates the OpenGALEN ontology (a medical ontology) by mapping it to the database, effectively annotating the fields, while also enabling external agents to query its data through another endpoint, the D2RQ server (relational database to ontology tool), with the ontology's or the local namespace.

Formally, PregSS main functionalities are as follows:

1. Allows all Registered Users to:

View and Edit Account Details

2. Allows Health Practitioners to:

- a. Do what Registered Users can do

- b. Manage patient records

- i. View, add, edit, or delete the following information: patient personal information, obstetrical history and present health problems, vaccine history

- ii. Search a patient

- c. Manage pregnancy consultation records

- i. View, add, edit or delete pregnancy consultation records: monthly prenatal checkups, referrals, labor and delivery, postpartum care and family planning
 - ii. Provide alerts on danger signs and recommend level of care
 - d. Transfer a patient record to the current health center
- 3. Allows the Barangay Health Center administrators to:
 - a. Perform the tasks Health Practitioners can do
 - b. Perform user account management on the Barangay Health Center
 - i. View, add, edit, or delete users
 - c. View and edit Barangay Health Center details
 - d. Generate an FSHIS monthly report on maternal care for the barangay in Excel Format
 - i. Pregnant with 3 or more Prenatal Visits
 - ii. Pregnant given Complete Iron dosage
 - iii. Postpartum (PP) w/ at least one PP visit
 - iv. PP mothers given Complete Iron dosage
 - v. PP women initiated breastfeeding
 - vi. Breastfeeding mothers given Vitamin A
 - vii. Women 15-49 given iodized oil capsules
- 4. Allows the PregSS administrator to:
 - a. Do what registered users can do
 - b. Approve and reject new Barangay Health Center requests
 - c. View, edit, or delete Barangay Health Centers
 - d. Search for Barangay Health Centers

- e. Generate an FSHIS monthly report like what the Barangay Health Center administrator can do, but for any region, province, city or municipality, and health center.
5. Allows Unregistered Users to:
- Register their Barangay Health Center and himself as its administrator

And to achieve semantic interoperability:

1. Map the local database schema to the OpenGALEN ontology
 - a. Write a mapping file that depicts the local database schema
 - b. Include the mappings between database fields and ontology classes with the same meaning
 - c. Use the D2RQ platform to link those two by using the mapping file
2. Set up another endpoint that will enable External Agents to:
 - a. Browse PregSS data as RDF Data
 - b. Search PregSS data by a SPARQL query through the URL, or through the interface, using the OpenGALEN or local namespace

D. Significance of the Study

PregSS is a clinical decision-support system that provides information and recommends a sound action for the wellness of the pregnant patient. It aims to provide assistance to health care practitioners in barangay health centers and also serves as a surveillance tool to monitor the current status of maternal health in rural areas.

By mapping the ontology to the database, other systems has the privileges to extract maternal-related data without difficulty since the database fields are more defined. In addition, it provides publicly accessible data for research and development in the field of pregnancy.

With the lack of currently available standards of DOH in dealing with semantic interoperability, I acknowledge the use of ontologies in line with the vision and goals of the semantic web. Since the OpenGALEN ontology is publicly available on the web, it has potential to serve as a foundation for building a local medical ontology for the Philippines, or by extending the said ontology, pave way for interoperability between global systems.

While semantic interoperability, along with the use of web services and ontologies, is still a young field, information on how to incorporate ontologies are thus scarce. This paper could serve as a demonstration on how to use ontologies for making the data in the system more accessible, while still retaining the relational structure of the system.

E. Scope and Limitations

The system incorporates a single, existing ontology. Therefore, it only uses the available terminology mappable to the relational database. To extend the ontology, ontology engineering methodologies are needed, but not included in the scope of this study. The main reason for this is that we want to be in line with the original ontology distributed, which could be used by other systems; including additional classes locally is not helpful to those external systems that may query PregSS.

F. Assumptions

Barangay Health Centers are assumed to have an internet connection to be able to access PregSS. Also, health personnel should have basic knowledge of using computers, especially browsers, since PregSS is web-based. To accommodate that, the Barangay Health Center administrator or representative is expected to orient the users first on how the system is used.

For barangay health center workers, it is assumed that each is associated mainly with one barangay health center only. For this reason, the health worker account is tied to the barangay center, and to work around with multiple barangays, he may request for another user account from the other health center.

II. Review of Related Literature

A. Medical Ontologies

While hospitals and other health care services may have good facilities and services internally, there is the problem of sharing information to other hospitals. That's why ontologies such as Smart Hospital (SH) ontology [7] were developed to serve as an intermediary between other information systems. The SH ontology was created using an iterative approach based on knowledge engineering methodology. Then, the resulting ontology was mapped for unified querying and presentation of information. Similar to this, Yao et al. proposed Context-embedded Intelligent Hospital Ontology (CIHO) [8] that caters to "intelligent" hospital environments. One good example where it can be used is in a smart hospital which employs RFID (Radio-frequency Identification) technology. The utilization of RFID technology gives way for a sensor-based and highly dynamic environment, and support for context awareness is needed for it to be able to adapt to changing contexts, and also, interoperability between different agents. The solution for this was the CIHO model, which represents important hospital entities, and also their relationships and properties. It resolves the issue of semantic interoperability using ontology or rule-based reasoning.

Another one is for Tropical Diseases Management [3], which primary goal is to represent the domain of tropical diseases, especially those occurring in South East Asia. It enables the use of an organized information support system that is accessible online, and thus, users are able to search easily for information.

B. Healthcare Information Systems using Ontologies

Ontologies are now being incorporated in Health Information Systems. For example, it is being used in decision support systems such as a Psychiatric Disorder Prevention System [9]. Lin's ontology-based system aims to associate the meridian system with modern medicine in detecting psychiatric disorders. He designed the ontology-based system using both domain ontology and task ontology to be able to describe the meridian domain knowledge, while also defining the rules in meridian diagnoses. This results to a decision support tree that is the basis of calculations for detecting sicknesses. While the system already contributes an effective framework for disorder detection, the ontology-based approach also provides more comprehensive results.

Ontology also has its role on public health. The public health domain caters more on the effects of diseases on the whole population rather than individuals, thus certainty of terms and processes are of more importance. A case study on the use of ontological principles in public health, specifically for tuberculosis care, was done by Kostkova and Kumar [10]. It defined a model around a tuberculosis care pathway, which is a step by step decision path, so that the best evidence available could be retrieved from medical databases.

There are large biomedical ontologies that have been developed already, such as SNOMED CT. While creating a new ontology requires domain expertise, one can already use such pre-existing ontologies in their applications. One example is an ontology-based drug recommendations discovery using GalenOWL [11]. The GalenOWL ontology provided a rich knowledge base for drug-drug and drug-disease interactions, and thus the system was able to provide good matches for drug recommendations. It was reported in this paper

that while the drawbacks of using the semantic web approach are higher memory consumption and longer setup times, it is still the better alternative for integrating knowledge in the system rather than simple business logic.

Ontology also has its usage in the Telehealth industry. One of which is OntoPsic [12], a Telepsychiatry Domain ontology that aims to assist professionals (psychiatrists, psychoanalysts and psychologists) in the field of mental identification and treatment. Along with OntoConsult, the health information system that uses the ontology as its knowledge base, it enabled the professionals to make correct identifications on disorders and their corresponding treatments. All in all, the incorporation of ontology in the Telehealth field resulted in a ubiquitous system that provides very expressive data.

C. Implementations of Ontology in Health Care

Incorporating ontology into an information system gives more means of storing, processing and retrieving of data. For building one, Buranarach's paper discussed an Integrated Ontology Management Framework, which has a four-step approach in developing the ontology [13]:

1. Ontology Creation – the creation of the ontology, which includes concepts and their properties, and also their relationships to one another.
2. Ontology Interpretation – how the ontology is stored and processed in a knowledge base. It also contains the logic defined by a set of sentences in a formal language.
3. Ontology Manipulation – the mechanisms that enable the users to access and update the data within the system.

4. **Ontology Distribution** – provides means for sharing and re-usage of the ontology.

The aim of this simplified approach is to introduce semantic web technologies more into the field, and for developers to be able to integrate the ontology with the system.

Most concepts used in health data registries may also be found in other medical fields, especially in interwoven health systems. Some of them are assumed to be semantically equivalent, when in fact, they are not. Thus, a method to map heterogeneity between these similar but non-equivalent semantic attributes [14] was proposed by Schuurman and Leszczynski. It aims to formalize the differences between such concepts using OWL ontology, and employs ontology-based metadata as extended context for data-specific attributes.

Another approach for semantic interoperability between such health care information systems (HCIS) could also be attained by archetype patterns which provide meta-models, and in turn serve as bridges between internal ontology and external standards [15].

D. Relational Database – Ontology Mapping

For interoperability between applications, one must ensure that the information is fully accessible and easily processed or interpreted by another system. Specifically, semantic interoperability should involve different systems having the same meaning of distributed information. One approach of achieving this semantic heterogeneity is by the use of ontologies. Wache summarizes three architectures in employing ontologies [16]:

1. Single Ontology – all information sources are using a shared vocabulary from one global ontology
2. Multiple Ontologies – each information source is described by its own ontology, not necessarily sharing the same vocabulary
3. Hybrid – same as the multiple ontology approach, but built upon a shared vocabulary

	Single Ontology Approaches	Multiple Ontology Approaches	Hybrid Ontology Approaches
implementation effort	straight-forward	costly	reasonable
semantic heterogeneity	similar view of a domain	supports heterogeneous views	supports heterogeneous views
adding/removing of sources	need for some adaptation in the global ontology	providing a new source ontology; relating to other ontologies	providing a new source ontology;
comparing of multiple ontologies	---	difficult because of a lack of common vocabulary	simple because ontologies use a common vocabulary

Table 2.1: Benefits and Drawbacks of the different ontology-based integration approaches

There are tools [17] [18] available in employing these approaches, by mapping ontologies to the information sources such as databases. DB2OWL [17] is an automated tool that creates an ontology expressed in OWL-DL language from a relational database. It works by converting database components to their corresponding ontology components, while taking into account several database cases. R2O [18], on the other hand, defines a language to describe mappings between relational DB schemas and ontologies. The resulting mappings, in turn, could then be processed by other tools to generate triples that

correspond to the database. A much simpler way is provided by RDB2Onto [19], which build OWL templates from SQL queries.

III. Theoretical Framework

A. Pregnancy and Maternal Health

Pregnancy, according to the World Health Organization (WHO), is the state when the woman carries a developing embryo and fetus in her womb, which usually lasts for 40 weeks. It is typically divided into three periods, called trimesters, each having unique growth patterns. Accordingly, they have their own levels of care, where each age of gestation has its own risk factors.

Accordingly, The DOH provides a Home Based Mother's Record (HBMR), which not only covers the span of the patient's pregnancy, but of her overall maternal health. This includes the prenatal (during pregnancy), natal (childbirth), and postpartum (after childbirth) periods. The HBMR aims to help in detecting the risks, and also the necessary actions needed to be done, since the baby is likely to develop problems if the mother is at risk [20].

B. Home Based Mother's Record

To be able to illustrate how maternal care is handled at barangay health centers, we'll look into what makes up the HBMR, as stated in the Midwives' Manual on Maternal Care:

1. Panel 1 – contains general information, obstetrical history, present health problems, and the risks associated with them. This could easily be filled up by any Barangay Health Worker.

Risk Factors for Pregnant Women

If a patient expresses one or more of these factors, she should be closely monitored by a midwife, or referred to a physician, or must have hospital delivery

- An age less than 18 or greater than 35
- Being less than 145 cm (4'9") tall
- Having a fourth (or more) baby
- Having had one or more of the following:
 - A previous caesarean section
 - 3 consecutive miscarriages or a stillborn baby
 - Postpartum Hemorrhage
- Having one or more of the following medical conditions:
 - Tuberculosis
 - Heart Disease
 - Diabetes
 - Bronchial Asthma
 - Goiter

Danger Signs

All patients are advised to report to available health personnel immediately if they experience the following symptoms during pregnancy:

1. Any type of vaginal bleeding
2. Headache, dizziness, blurred vision
3. Puffiness of face and hands

4. Being pale or anemic

Obstetrical History

- Number of previous pregnancies
- Previous Caesarean Section
- 3 Consecutive Abortion / Miscarriages
- Stillbirth
- Postpartum Hemorrhage

Present Health Problems

- Tuberculosis
- Heart Disease
- Diabetes
- Bronchial Asthma
- Goiter

2. Panel 2 – contains information about the present pregnancy, which includes a table on prenatal checkups that are usually conducted by midwives, nurses or a doctor. Next we have the labor and delivery section which includes information at the time of birth such as date, place and type of delivery, etc.

- LMP – Last Menstrual Period
- EDC – Expected Date of Confinement
- AOG – Age of Gestation

General Examination and Actions

- Concerns on evaluating physical symptoms. *See Appendix.*

Labor and Delivery

- Concerns on the date, place and type of delivery, and also the baby's health.

See Appendix.

3. Panel 3 – contains the referrals made by the personnel and the follow-up results.
4. Panel 4 – contains postpartum care information and also the family planning sessions.

Postpartum Care

- Concerns on postpartum symptoms and if the mother breastfeeds exclusively.

See Appendix.

Family Planning

- Explanation of the benefits of family planning and the risks of having a baby too soon
- Choices of family planning methods such as contraceptives

C. FSHIS Monthly Form – Program Report (M1)

The Field Health Service Information System (FHSIS) is developed by DOH to collect aggregate statistics on its different health programs such as Maternal Care, Family Planning, Child Care and Disease Control. In fact, it is the only information system of the DOH so far that is able to reach down to the barangay levels in the hierarchy.

For PregSS, the specific report it generates is the Maternal Care part of the FSHIS Monthly Form - Program Report (M1). The fields included, and its corresponding specifications are based from the 'Electronic Field Health Service Information System: Manual of Operations:

1. Pregnant with 3 or more Prenatal visits (w/ at least 1 visit per trimester) - the total number of pregnant women who had 3 or more prenatal visits during the month such that at least one visit occurs during the first trimester, one during the second trimester and one during the third trimester
2. Pregnant given TT2 plus - the total number of pregnant women given TT2 plus during the month. TT2 plus includes 2nd, 3rd, 4th and 5th doses of Tetanus Toxoid given to pregnant women
3. Pregnant given Complete Iron Dosage - the total number of pregnant women given complete tablet of 60 mg of Fe with 400 mcg Folic acid, once a day for 6 months or 180 tablets.
4. Postpartum (PP) w/ at least 1 PP visit - the total number of post-partum women who were seen by the midwife/PHN/MHO at home or at the clinic once or more than once after delivery such that first visit should be within 24 hours upon delivery
5. PP mothers given Complete Iron Dosage - the total number of post-partum women given complete tablet of 60 mcg of Fe with 400 mcg Folic acid, once a day for 3 months or a total of 90 tablets.
6. PP women initiated breastfeeding - the total number of post-partum or lactating women given 200,000 I.U. of Vitamin A capsule within 4 weeks after delivery.

7. Breastfeeding mothers given Vitamin A - Postpartum women which are breastfeeding and given vitamin A during the month
8. Women 15-49 given iodized oil capsules – Women between the age of 15 and 49 given iodized oil capsules during the month

D. Millennium Development Goals

According to WHO, as of this time, the United Nations Millennium Development Goals are eight goals that are agreed upon by all the 191 UN States, including the Philippines, to be achieved by 2015. These are benchmarks that aim to combat poverty, hunger, disease, illiteracy, environmental degradation, and discrimination against women. Specifically, the eight ones are, in verbatim:

1. to eradicate extreme poverty and hunger;
2. to achieve universal primary education;
3. to promote gender equality and empower women;
4. to reduce child mortality;
5. to improve maternal health;
6. to combat HIV/AIDS, malaria, and other diseases;
7. to ensure environmental sustainability; and
8. to develop a global partnership for development

This study caters to satisfying the improvement of maternal health in the country. To expand, the targets of improving maternal health are:

1. Reduce the maternal mortality ratio by three quarters

- As stated in the background, the concrete goal of the Philippines for 2015 is 52 per 100,000 live births. As of this time, the ratio is still higher, but increasing information system coverage could alleviate this situation.
 - Also included here is that there must be a high proportion of births attended by skilled health personnel
2. Achieve universal access to reproductive health
- This is achievable through a higher coverage of antenatal care, which is one of the goals of this study
 - Also, contraceptive usage and the needs for family planning must be met

What is convenient here is that these goals are said to be inter-dependent. For example, improving the maternal health in the country could also improve resistance to diseases for children, which in turn, helps with the combat of other diseases goal.

E. Information System

To provide assistance to health practitioners, while also managing barangay health centers, an information system is deemed feasible. An information system is formally defined as an “integrated set of components for collecting, storing, and processing data and for delivering information and knowledge” [21]. In particular, PregSS is considered as both a clinical decision-support system and surveillance tool. Clinical, in terms of the environment where it’s used, and a decision-support system because it analyzes patient data to provide a recommendation on the level of care required for the patient. Also, it serves as a surveillance tool, because it monitors maternal health per location by collecting statistical data on maternal indicators and then generating reports.

F. Ontology

The term ‘ontology’ has its roots in philosophy, where it usually involves the nature of being and the kinds of existence [22]. It was then adopted in computer science, where Gruber formally defined it as a “specification of a conceptualization”, which includes the terms being used and the relations between them [23] [24]. Basically, it is an approach to represent a specific domain or knowledge base by using entities as classes, each with their according properties or attributes, and also the relationships existing among its members. This solves many problems in the integration of heterogeneous databases, and thus, enables interoperability among different systems [25].

In the field of health care and medicine, ontologies can be used to organize and classify medical terminologies to prevent ambiguity. For example, SNOMED-CT (Systematized Nomenclature of Medicine – Clinical Terms) ¹, is a very large and comprehensive clinical terminology available in most languages, especially English. It is under the UMLS, or Unified Medical Language System², which is a set of tools and knowledge sources that provide biomedical vocabularies for interoperability between health systems.

To illustrate, here is a diagram from Amanda Ryan’s paper on SNOMED-CT that greatly describes how an ontology, specifically a medical one, looks like [26]:

¹http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html

²<http://www.nlm.nih.gov/research/umls/quickstart.html>

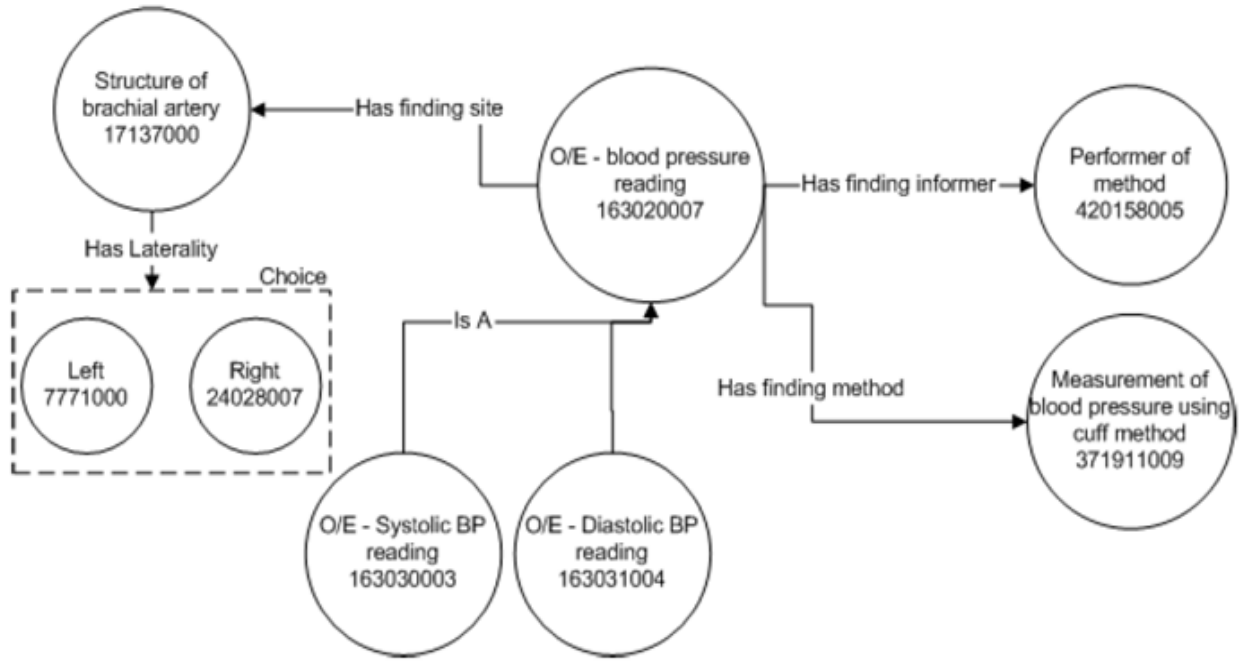


Figure 1: Blood Pressure Reading Concepts and Attributes in SNOMED-CT

Concept	Attribute	Concept
O/E - blood pressure reading	Finding site	Structure of brachial artery
Structure of brachial artery	Laterality	Left/Right
O/E - blood pressure reading	Finding informer	Performer of method
O/E - blood pressure reading	Finding method	Measurement of blood pressure using cuff method
O/E - systolic BP reading	Is a	O/E - blood pressure reading
O/E - diastolic BP reading	Is a	O/E - blood pressure reading

Figure 3.1: Visualization of a part of SNOMED-CT and also its relationship table

As seen in the image, an ontology can be viewed in graph form. Each concept (in circles) could be identified semantically through its relationships (in arrows) with other concepts.

Granted that, ontologies are included in the World Wide Web Consortium (W3C) standards for the semantic web. The semantic web is an effort to link data on the web through the use of shared vocabularies and rules for handling these data, so as to enable interactions between different sources [27].

G. OWL

For representing ontologies on the web, it is W3C's recommendation since 2004 to use OWL. The Web Ontology Language (OWL) is intended to be used by applications for processing data, and not on representing information to humans [28]. It's basically an ontology language that can formally define the terminology used in web applications.

One of the needs of the semantic web is that information must be easily processed by other machines, and a solution for this is to add information about the resources that describe the web content [29]. That is, including an ontology possibly in OWL form, and also allowing for information to be gathered from different resources, if importing from other ontologies is done.

OWL comes in different sublanguages, each with differing expressiveness:

- OWL Lite is the least complex of the three, and supports classification hierarchy and simple constraints
- OWL DL supports maximum expressiveness, while still having guaranteed computable conclusions
- OWL Full also supports maximum expressiveness, with added syntactic freedom of RDF, but has no computational guarantee

H. RDF and SPARQL, Linked Data

OWL combines XML and RDF. It uses XML (Extensible Markup Language) as its syntax, while utilizing the Resource Description Framework (RDF)'s way of representing data. Delving deeper, RDF is a framework representing information on the web [30]. It could be visualized as a simple graph-based data model with additional formal semantics,

thus enabling for sound deductions to be made on RDF data. As seen in the ontology concept earlier, ontology in RDF is a collection of triples, each consisting of a subject, a predicate and an object. What makes OWL different from RDF is that OWL adds more depth in describing properties and classes, such as cardinality, equality, characteristics of properties, and enumerated classes, among many others [28].

RDF identifies resources through Uniform Resource Identifiers (URIs), and describes them in terms of properties and property values [31]. To exemplify (here we use vCard (an electronic business card profile) descriptions of people) [32]:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Figure 3.2: An RDF describing a person named Eric Miller in RDF/XML syntax.

The above example describes an individual Eric Miller of type Person, with properties fullName (String), mailbox (another RDF resource) and personalTitle (String), all with their corresponding values. It used the ‘contact’ namespace defined by the URI <http://www.w3.org/2000/10/swap/pim/contact#>.

On the other hand, its corresponding graph form is as follows:

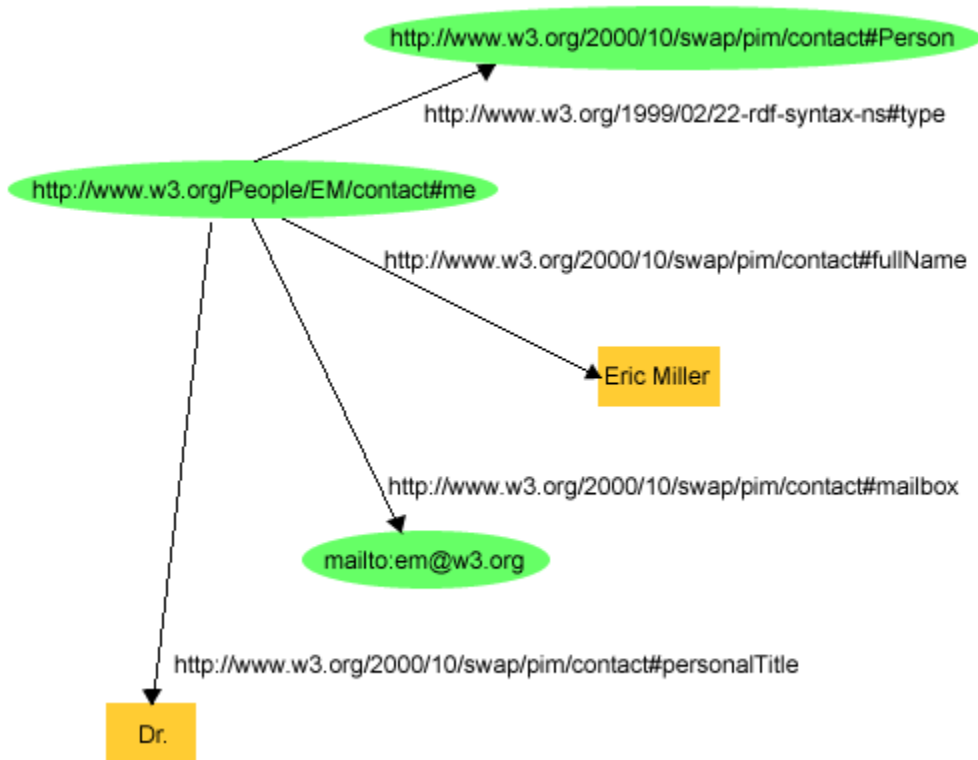


Figure 3.3: RDF Graph describing Eric Miller.

Here, you can clearly see the URIs used to identify the resources. Notice that each property has their own URIs, which makes relationships between linked items precisely identifiable. With that, like in HTML, URIs can be used to link pieces of information across the web. And that is one of the goals of the semantic web made possible by RDF, namely, linked data. These are data which could be interlinked with other related data, referable by user agents through URIs, and should be easily understood by computers by standards of RDF and its query language.

A query language for RDF is SPARQL. A SPARQL query usually contains a set of triple patterns called a basic graph pattern. Triple patterns are, in the same sense, RDF triples, but the subject, predicate and object could be variables. To demonstrate:

Consider the following RDF data [32]:

```
@prefix vCard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://somewhere/MattJones/>vCard:FN    "Matt Jones" .
<http://somewhere/MattJones/>vCard:N    _:b0 .
_:b0 vCard:Family "Jones" .
_:b0 vCard:Given  "Matthew" .

<http://somewhere/RebeccaSmith/>vCard:FN    "Becky Smith" .
<http://somewhere/RebeccaSmith/>vCard:N    _:b1 .
_:b1 vCard:Family "Smith" .
_:b1 vCard:Given  "Rebecca" .

<http://somewhere/JohnSmith/>vCard:FN    "John Smith" .
<http://somewhere/JohnSmith/>vCard:N    _:b2 .
_:b2 vCard:Family "Smith" .
_:b2 vCard:Given  "John" .

<http://somewhere/SarahJones/>vCard:FN    "Sarah Jones" .
<http://somewhere/SarahJones/>vCard:N    _:b3 .
_:b3 vCard:Family "Jones" .
_:b3 vCard:Given  "Sarah" .
```

And this basic pattern:

```
SELECT?givenName
WHERE
{?y<http://www.w3.org/2001/vcard-rdf/3.0#Family>"Smith".
?y<http://www.w3.org/2001/vcard-rdf/3.0#Given>?givenName.
}
```


The one above looks similar to a standard SQL SELECT query, that consists of the field name to be retrieved and a WHERE clause, whereas the ones preceded with “?” are variables.

Running the query (basic pattern) results to:

```
-----  
|givenName|  
=====  
|"John"|  
|"Rebecca"|  
-----
```

As seen from the above example, it matches data triples where all the pattern triples are satisfied.

I. Protégé

To view the ontology in both tree and graph format, we have Protégé. Protégé³ is a tool, an ontology editor that is both free and open-source. It is written in Java and was developed by the Stanford Center for Biomedical Informatics Research. It enables ontology creation and manipulation using visual representations, such as trees. Then the ontology can be exported into various formats, including OWL, RDF(s) and XML Schema.

Protégé also has two main ways of modeling ontologies, one of which is the Protégé-OWL editor that is intended to support OWL. Protégé-OWL can work hand-in-hand with Jena (an ontology API for Java), and in fact, is closely integrated with the said API.

³<http://protege.stanford.edu/>

J. OpenGALEN

OpenGALEN⁴ is a publicly available knowledge repository. It provides an open-source medical terminology that encompasses many different fields, including obstetrics. It is available as ontology in OWL format and has one of the most, if not the most, comprehensive information on obstetrics that is freely available on the web.

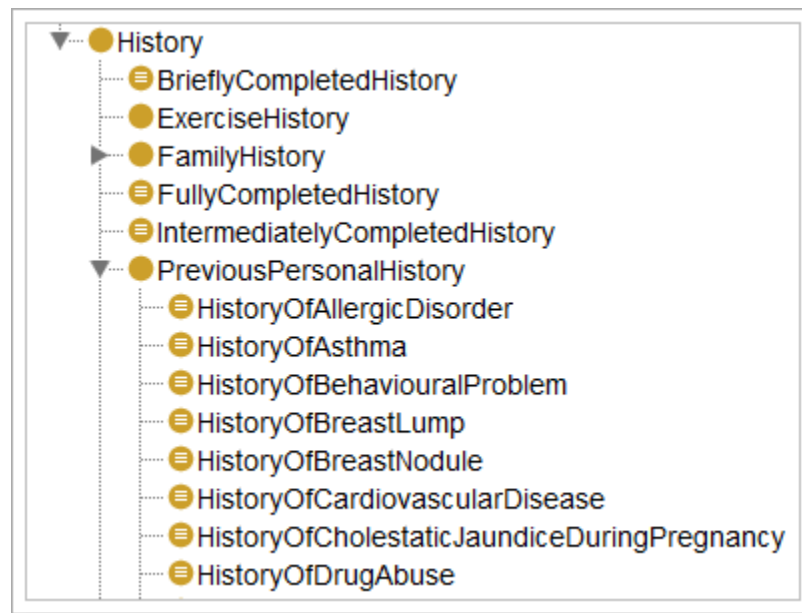


Figure 3.4: Part of OpenGALEN – Obstetrics tree viewed through the Protégé browser



Figure 3.5: Protégé Graph View: Sample relationship of OpenGALEN concepts

⁴<http://www.opengalen.org/>

K. D2RQ Platform

Since the OpenGALEN ontology is written in RDF format, we have a tool for mapping RDF to a relational database. It's called D2RQ [33], which is an open-source platform for accessing relational databases (e.g. Oracle, MySQL) as RDF graphs. One of the strengths of the platform is that the content from the database is not replicated; it is accessed virtually, hence lower space requirements. And in summary, it enables the users to:

- Query a non-RDF database using SPARQL
- View the database content as Linked Data
- Replicate data from the database into RDF format

The D2RQ Platform components that are utilized in this study are:

1. D2RQ Mapping Language – a mapping language for describing the relation between an ontology and a relational data model. D2RQ is able to automatically generate a default mapping file based on the database schema, and then by relating the classes and properties with the ones from an existing ontology, we have linked the two together.
2. D2RQ Server – an HTTP server that provides a Linked Data view, and also serves as a SPARQL protocol endpoint for querying the database

L. JavaServer Faces

JavaServer Faces (JSF) [34] is a Java-based, component-oriented user interface (UI) web framework. JSF follows the Model-View-Controller (MVC) design pattern, while further separating behavior and presentation unlike traditional client-side UI architectures. It also includes many features, such as server-side validation, event-handling and navigation.

Usually, a JSF application is composed of [35]:

- A set of JSP/XHTML pages
- A set of backing beans, which are reusable objects in the form of JavaBeans for the UI components on a page
- An application configuration resource file (faces-config.xml), which configures beans and sets navigation rules
- A deployment descriptor (web.xml)
- A set of custom tags for custom objects

IV. Design and Implementation

A. PregSS

1. System Architecture

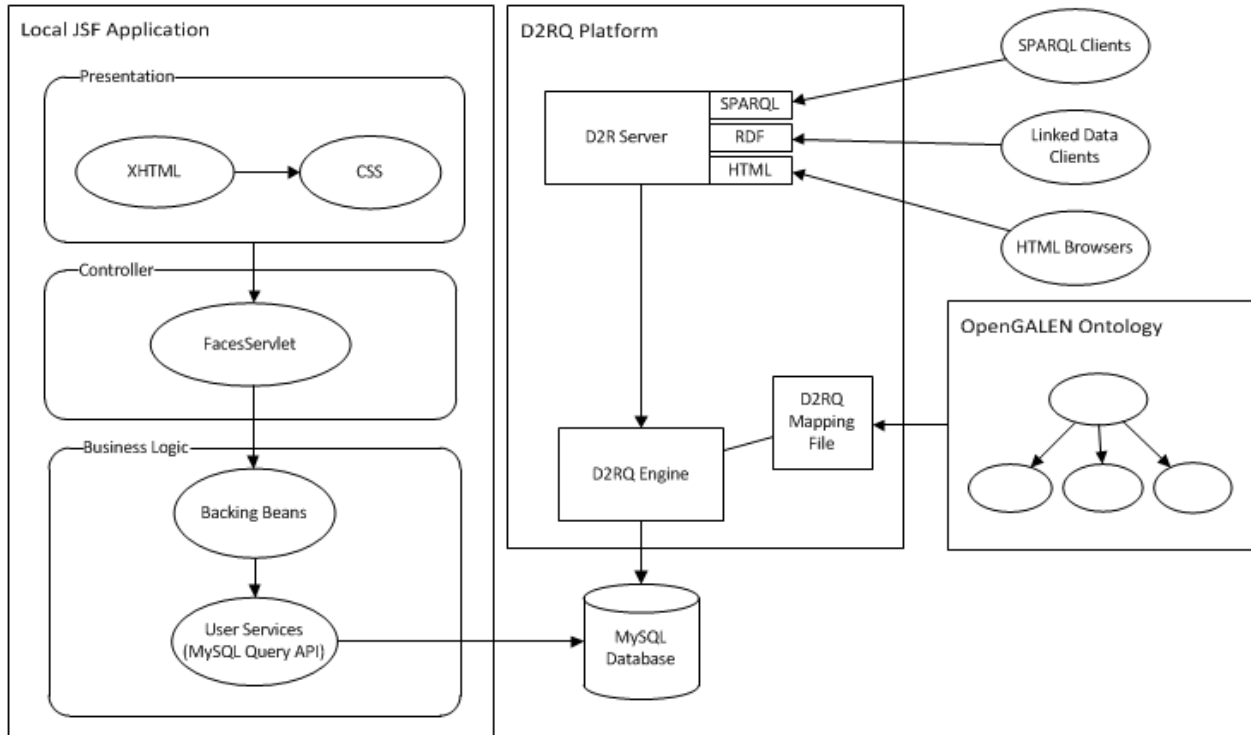


Figure 4.1: Architecture of PregSS

PregSS stores its data in a relational MySQL Database, and can be queried from two endpoints. One is from the local web application in JSF, seen above following a Model-View-Controller paradigm, and queries the database using SQL queries. Another endpoint is the D2RQ server, which is accessible from a different URL, wherein passing SPARQL queries through the D2RQ Engine leads to a corresponding call to the MySQL database and returns back data in RDF format. By the use of a D2RQ mapping file, some fields in the database are annotated and mapped with the OpenGALEN ontology, and thus can be queried using the OpenGALEN namespace.

2. Context Diagram

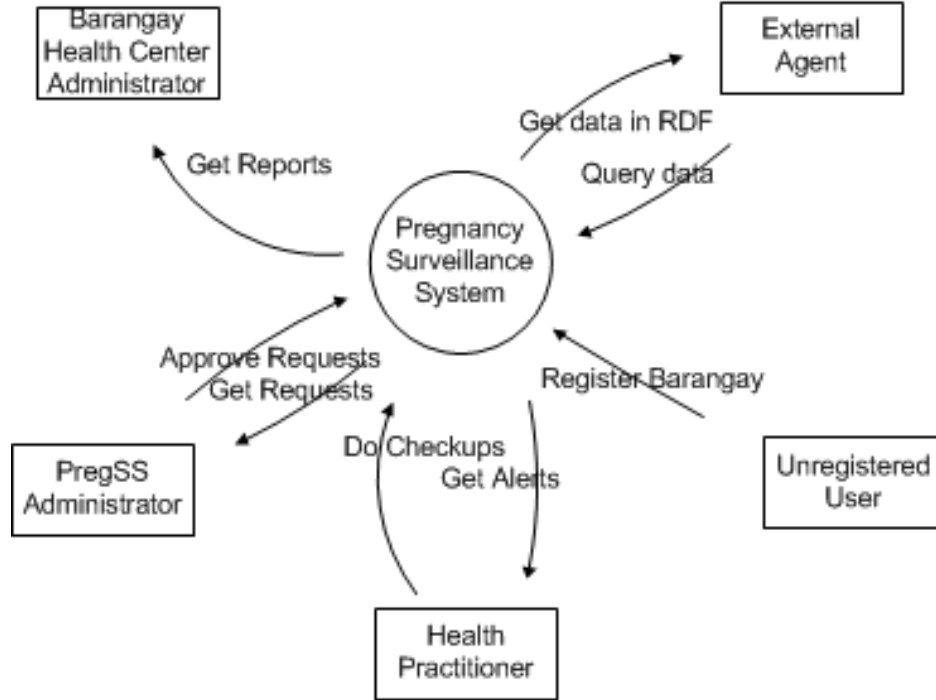


Figure 4.2: Context Diagram for PregSS

The context diagram above represents the Pregnancy Surveillance System. The unregistered user registers their Barangay Health Center, and then the PregSS administrator receives the requests, and approves/declines the requests. Health practitioners input patient data through checkups, and receives corresponding alerts from the system. PregSS does an aggregation of patient data through reports, which can be downloaded by the Barangay Health Center administrator. Also, external agents from the D2RQ endpoint runs queries and get RDF data in return.

3. Use-Case Diagram

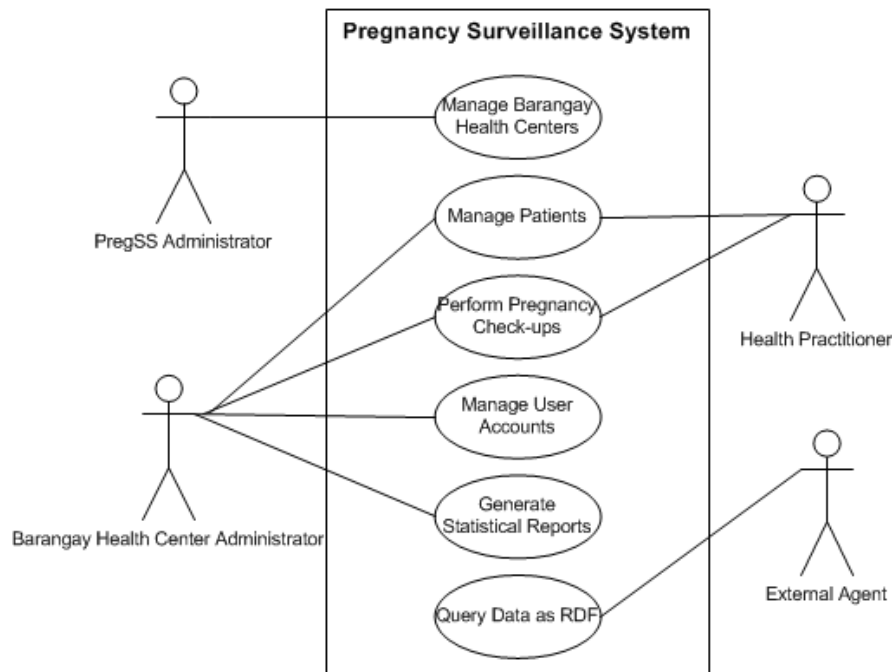


Figure 4.3: Top-level use case diagram of PregSS

As seen from above, the PregSS Administrator's main role is to manage barangay health centers. Health Practitioners, on the other hand, handles patient information and so is able to perform pregnancy check-ups. Barangay Health Center Administrators also has the same capabilities as a Health Practitioner, but he is also responsible for handling users and generating the monthly report for the barangay. Finally, we have the external agent that queries PregSS data as RDF through the D2RQ endpoint.

1. Manage Barangay Health Centers

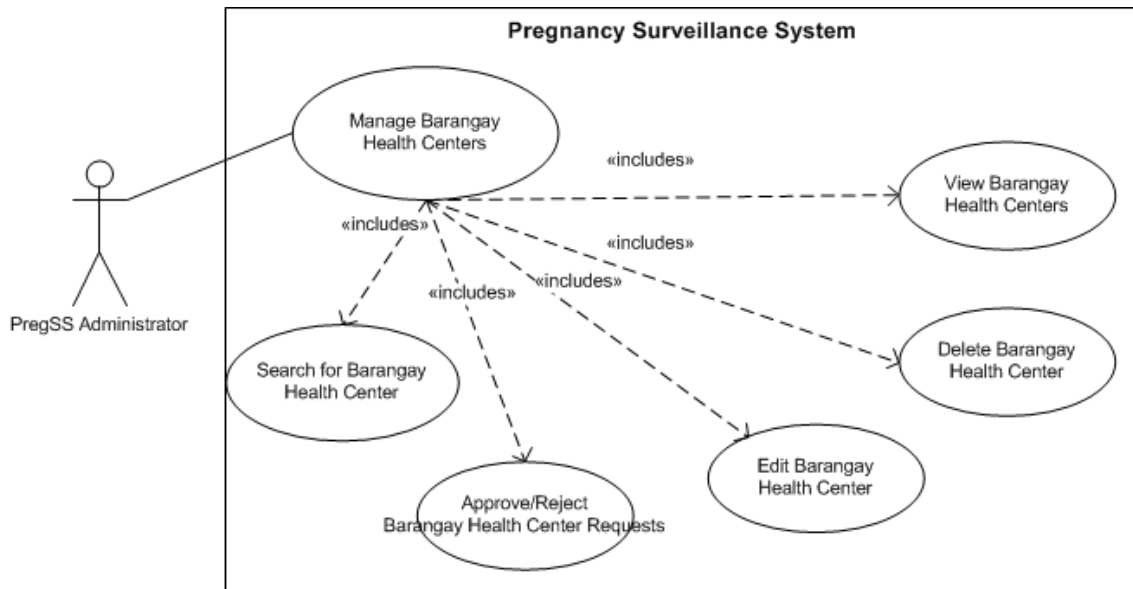


Figure 4.4: Manage Barangay Health Centers use case diagram of PregSS

The PregSS Administrator's main role is to Manage Barangay Health Centers, which includes approving health center requests, viewing, editing, deleting and searching for barangay health centers.

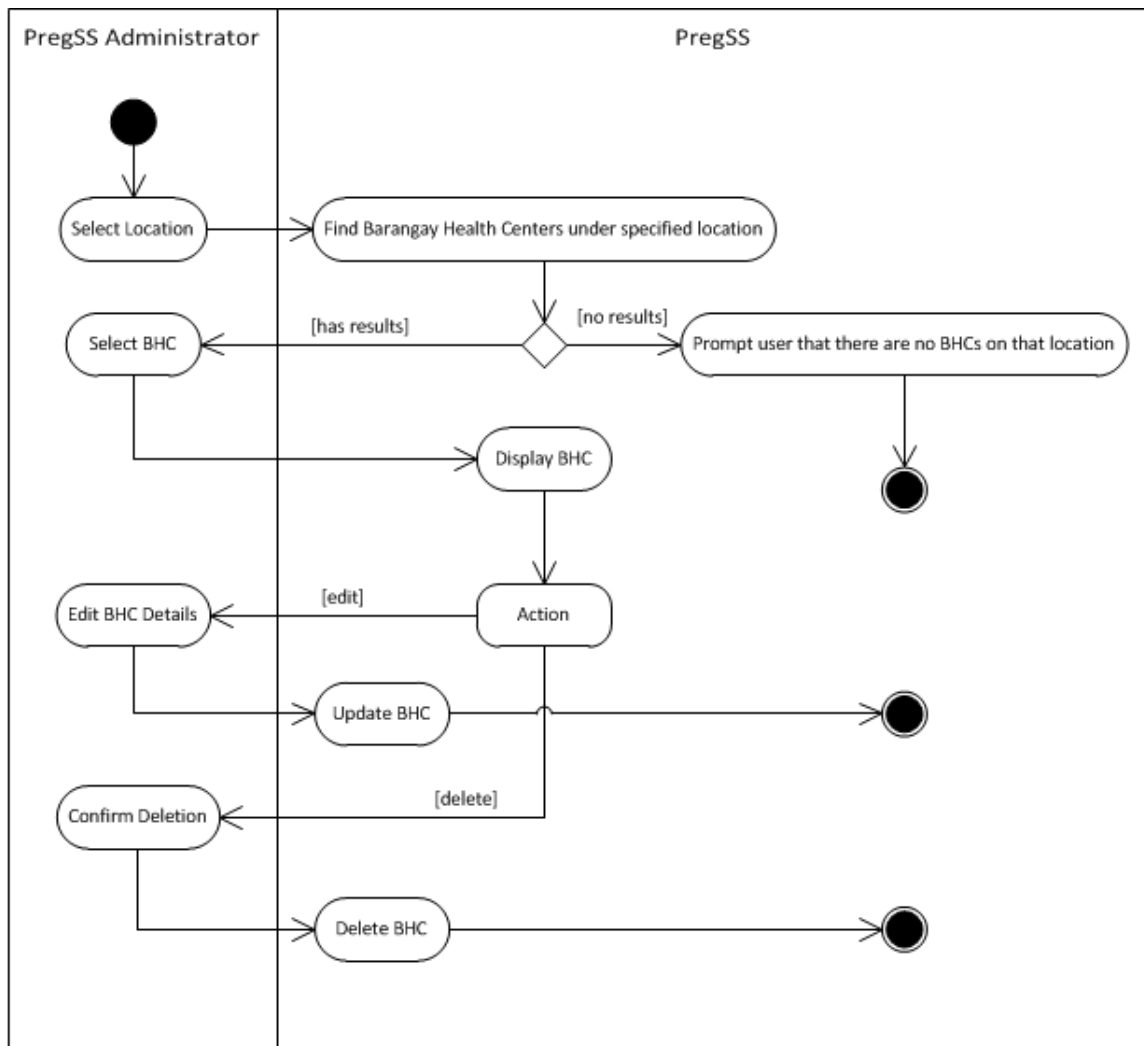


Figure 4.5: View, Edit and Delete Barangay Health Centers Activity Diagram

The PregSS Administrator in the PregSS administrator control panel first specifies a location (region, province, and city), then PregSS shows the barangay health centers situated on that location, or none otherwise. If a health center is selected, PregSS display its details and also the option to edit or delete the health center. To update, he must first input the new BHC details. To delete, PregSS asks for confirmation before deleting the BHC.

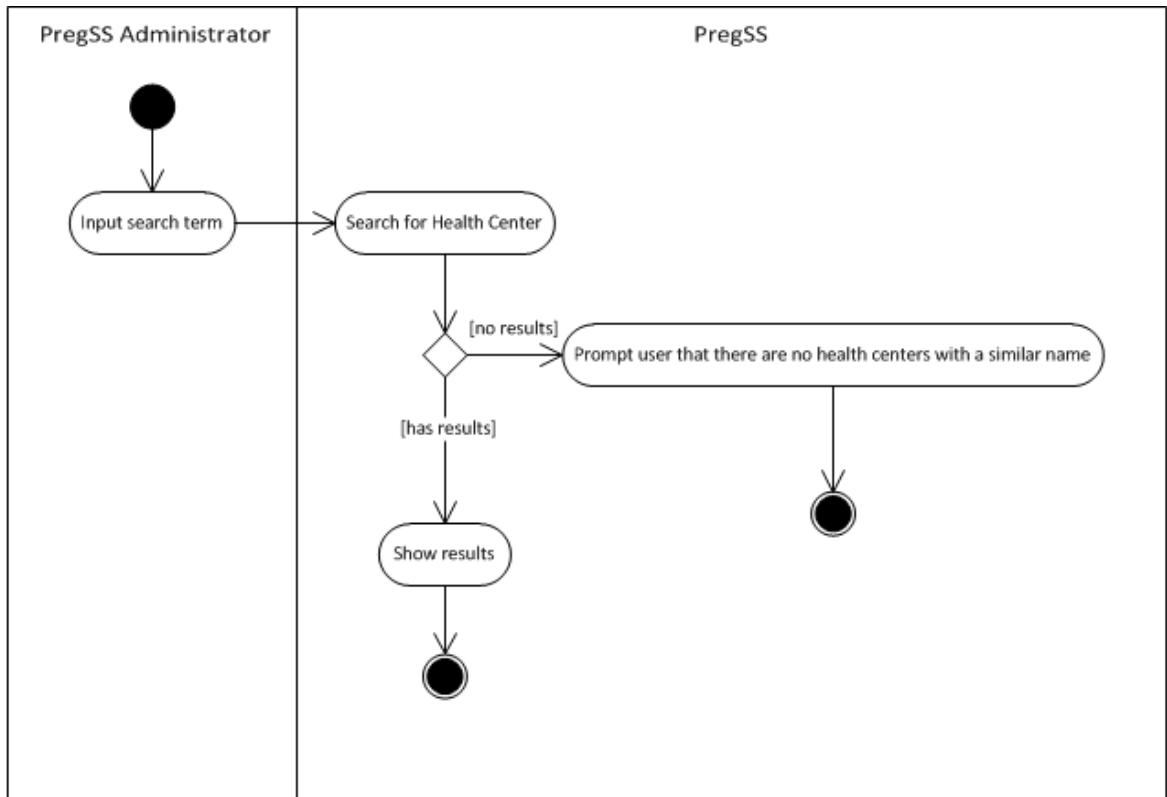


Figure 4.6: Search for Barangay Health Center Activity Diagram

In the select by location list in the PregSS Administrator control panel, the user may input a filter text for barangay to limit the results and easily search for the target BHC.

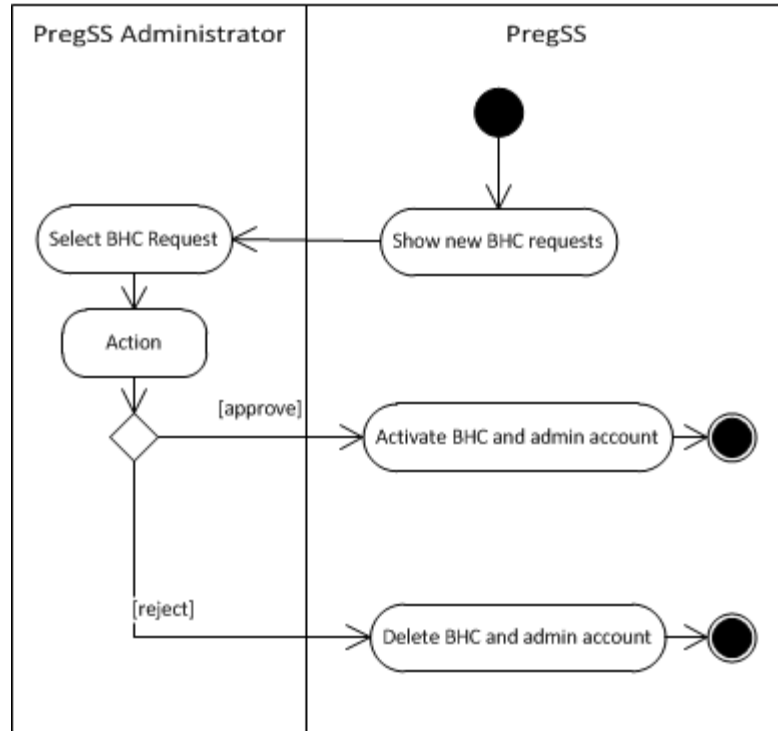


Figure 4.7: Approve/Reject Barangay Health Center Activity Diagram

In the Approve Health Center Requests page, the PregSS Admin sees a list of requests with a choice of approving or rejecting them. Choosing the ‘approve’ one activates both the barangay health center and admin account. On the other hand, rejecting one deletes both the BHC and admin account requests.

2. Generate Statistical Reports

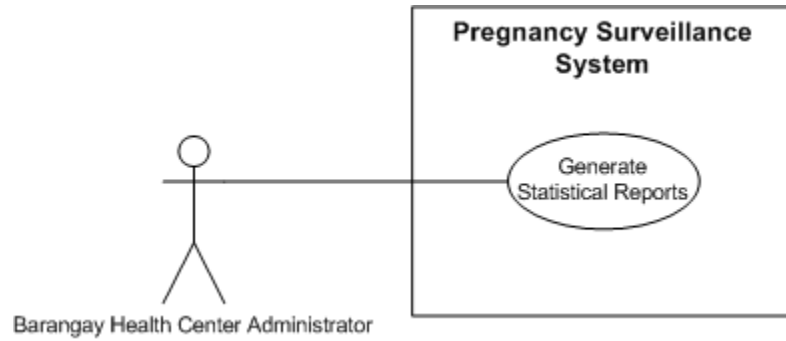


Figure 4.8: Generate Statistical Reports use case diagram of PregSS

The Barangay Health Center Administrator is able to generate monthly reports based on the FHSIS Monthly Report Form.

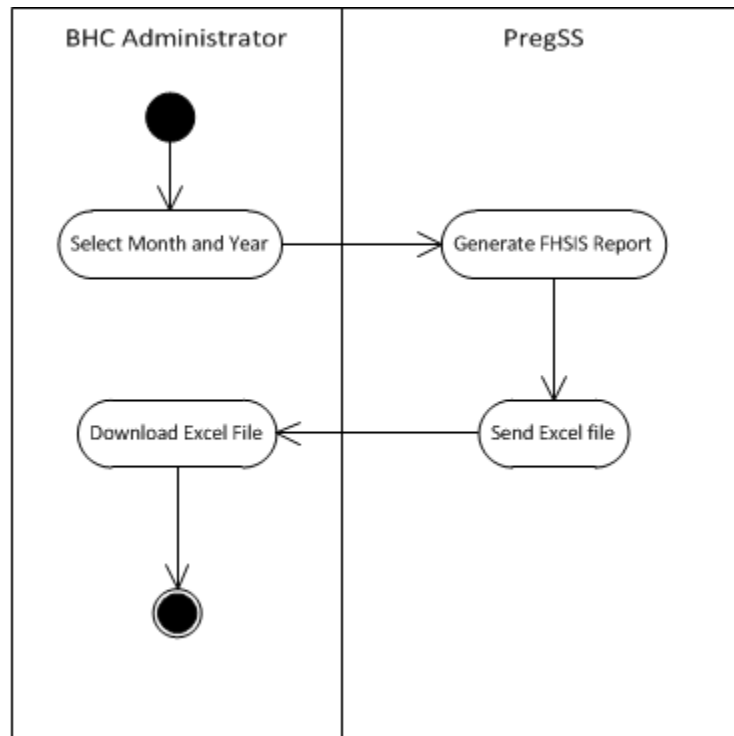


Figure 4.10: Generate Statistical Reports Activity Diagram

To generate an FHSIS report, the BHC admin specifies the month and year for the report. Then, PregSS sends the report for download.

3. Manage User Accounts

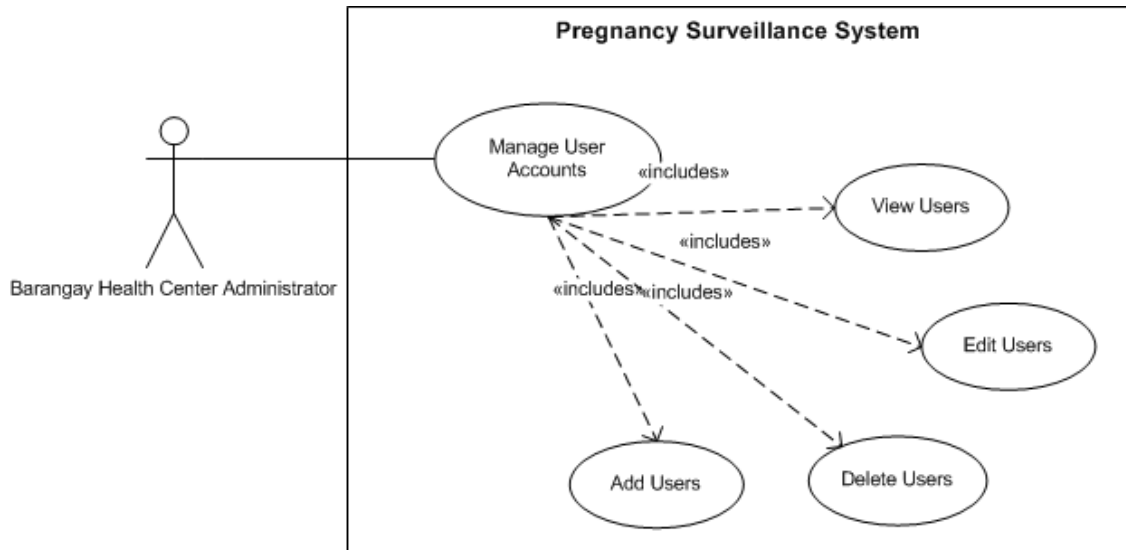


Figure 4.11: Manage User Accounts use case diagram of PregSS

One capability of the BHC administrator is to manage the user accounts under the barangay health center. This includes viewing, adding, editing and deleting users.

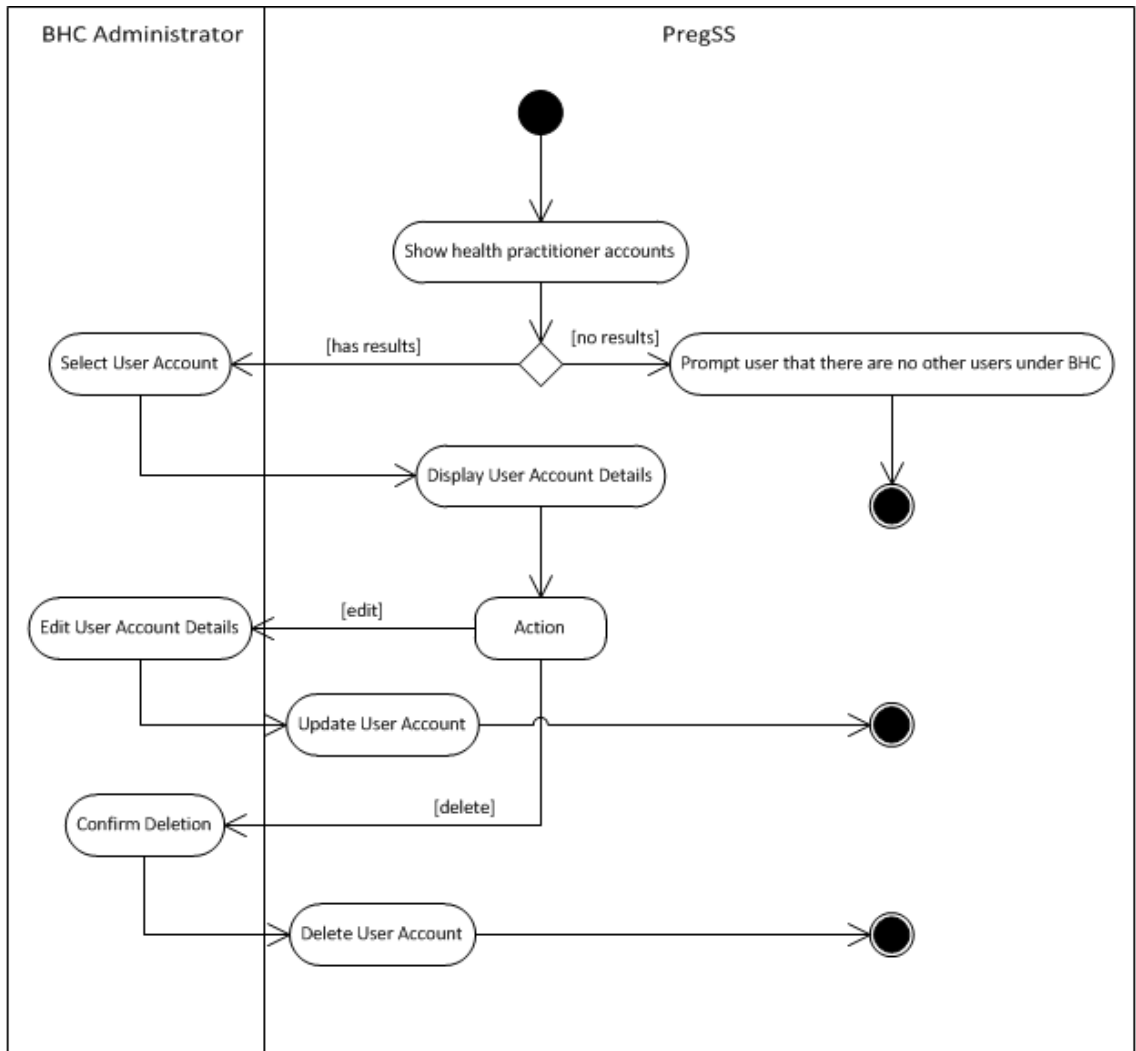


Figure 4.12: View, Edit, and Delete Users Activity Diagram

The BHC Administrator is first presented with a list of user accounts under the barangay health center, otherwise, none. He then selects a user account and PregSS displays the user details. There is the choice of editing or deleting the user account. If he chooses edit, he must then input the changes and then it will propagate to the system. Otherwise, choosing delete asks for confirmation before the user is deleted from the system.

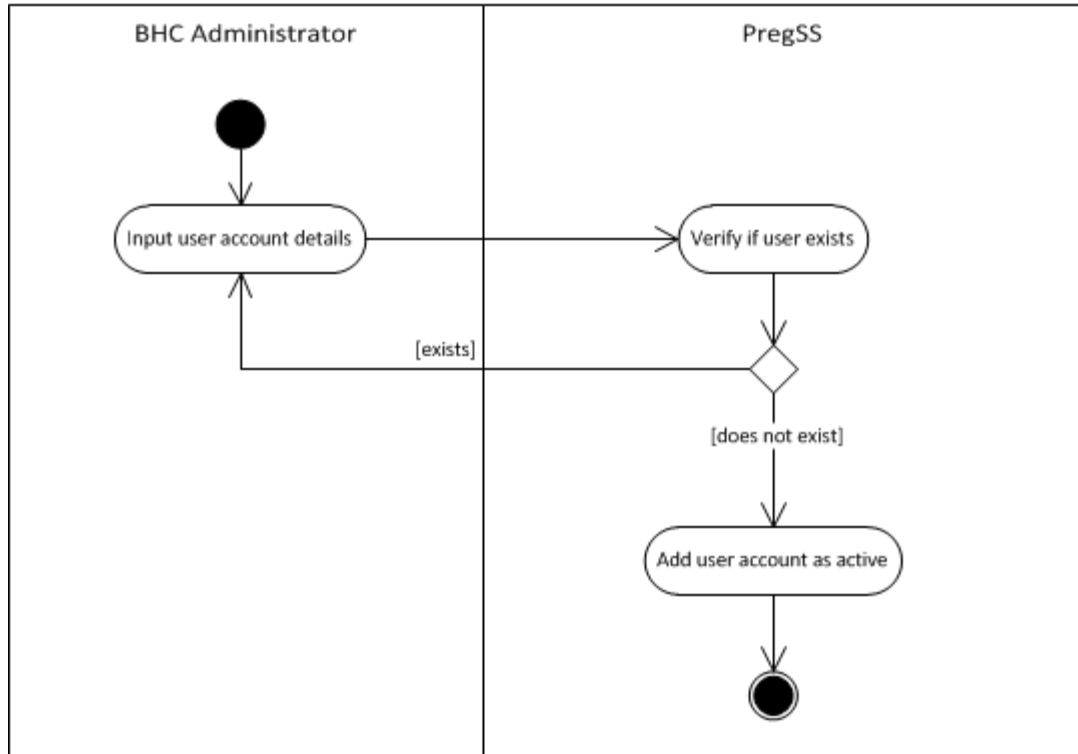


Figure 4.13: Add User Activity Diagram

The user inputs the new account details, and will be added to the system as long as the username specified does not exist yet.

4. Manage Patients

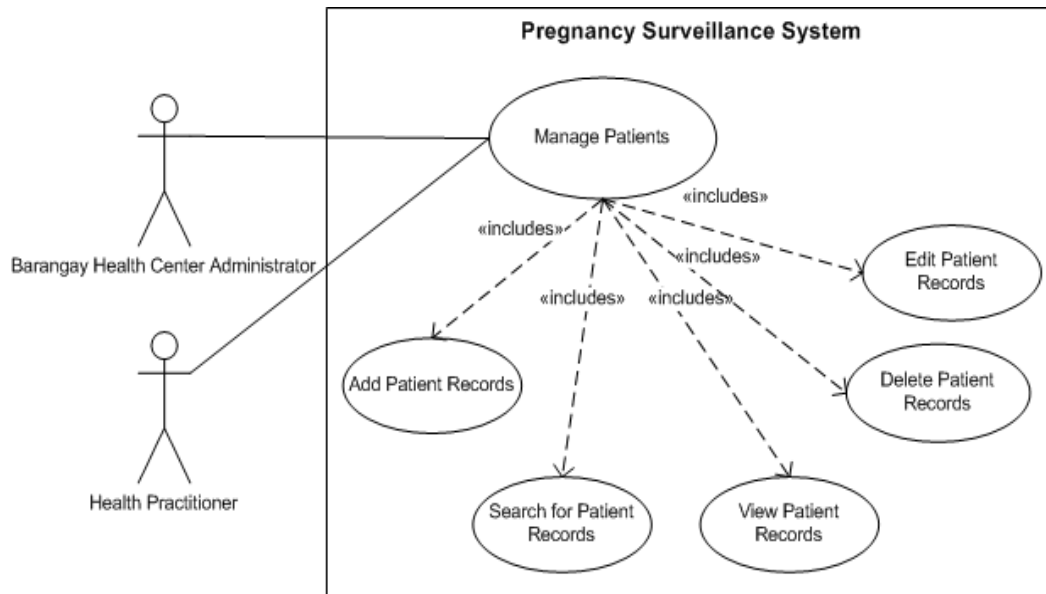


Figure 4.14: Manage Patients use case diagram of PregSS

Both the BHC Admin and Health Practitioner can manage the patients of PregSS. This includes viewing, editing, adding, deleting and searching for patients.

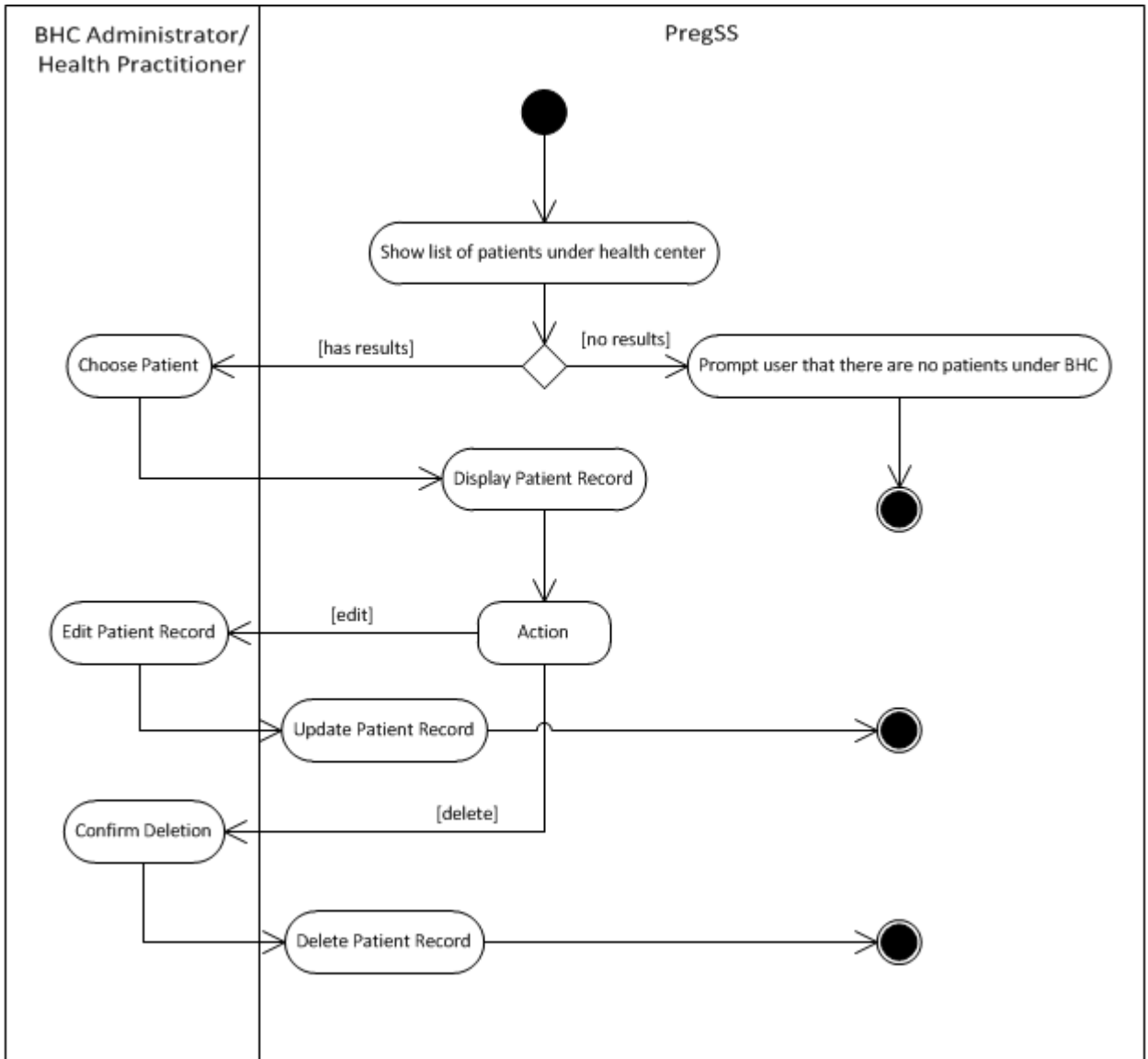


Figure 4.15: View, Edit, and Delete Patient Records Activity Diagram

The BHC Administrator or Health Practitioner is first presented with a list of patients under the barangay health center, otherwise, none. He then selects a patient and PregSS displays the patient information. There is the choice of editing or deleting the patient. He can choose a part to edit, and then the changes should be inputted first before the specified information can be updated. Otherwise, choosing delete asks for confirmation before the patient is deleted from the system.

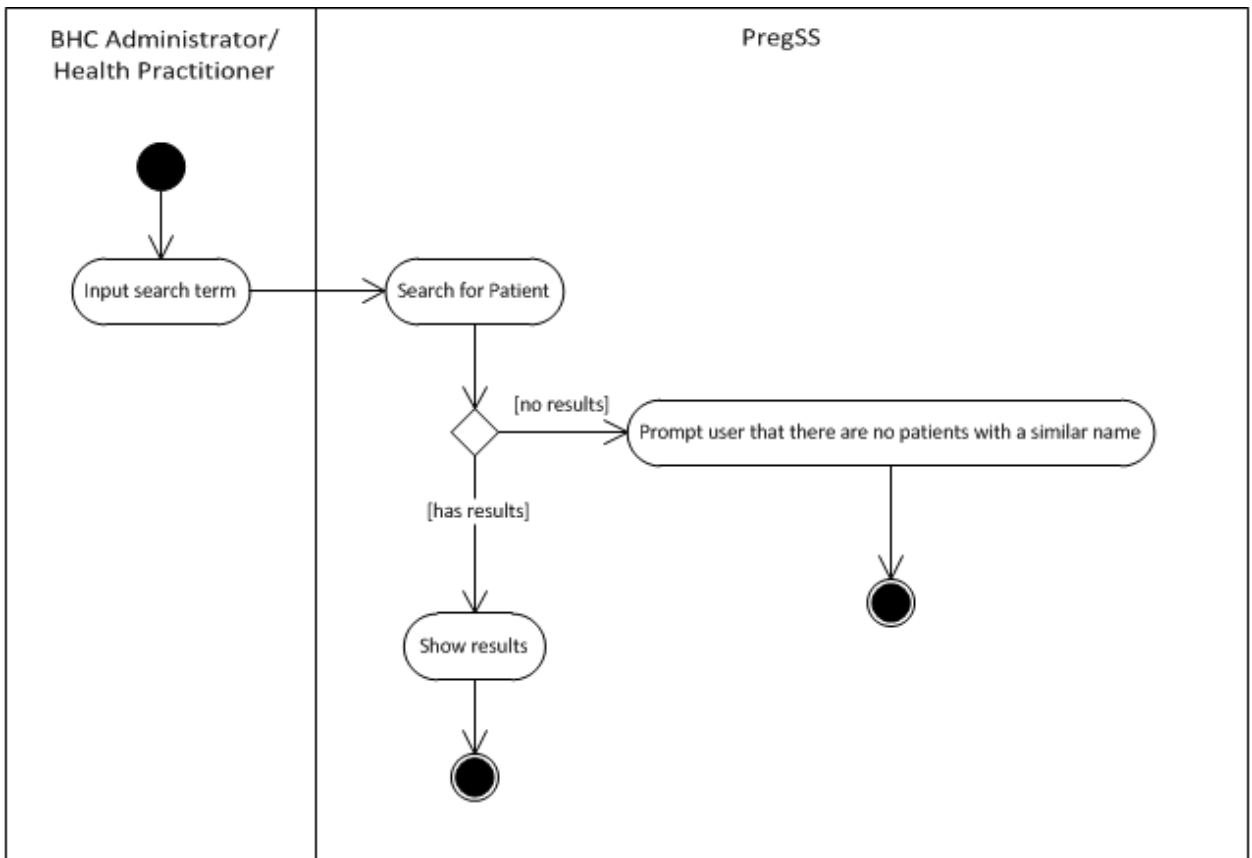


Figure 4.16: Search for Patient Activity Diagram

The user first inputs the search term and chooses the scope for the search. If local, PregSS shows results of patients under the barangay health center. If all, it searches the whole system for the specified name.

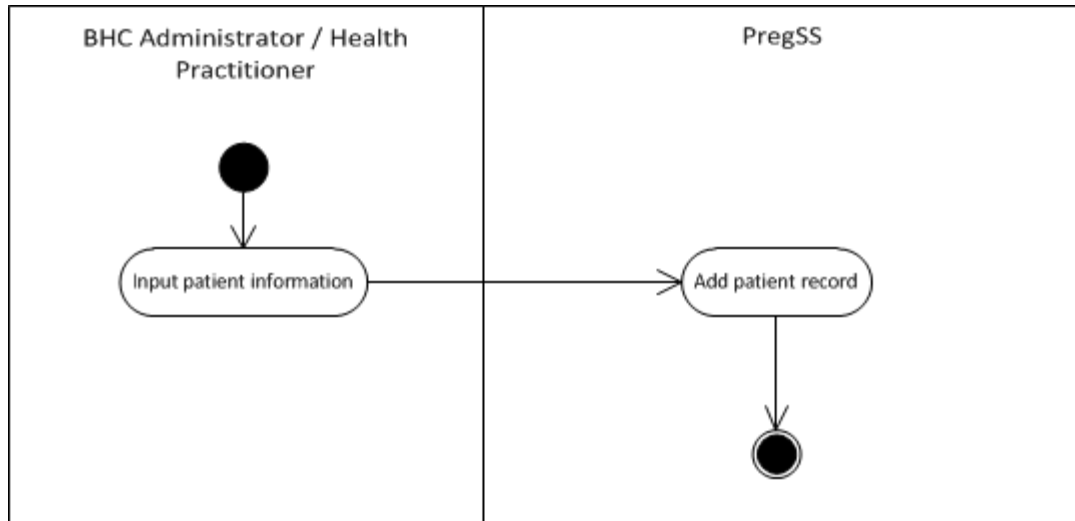


Figure 4.17: Add Patient Record Activity Diagram

First input the patient information, after that, PregSS then adds the patient to the system.

5. Perform Pregnancy Check-ups

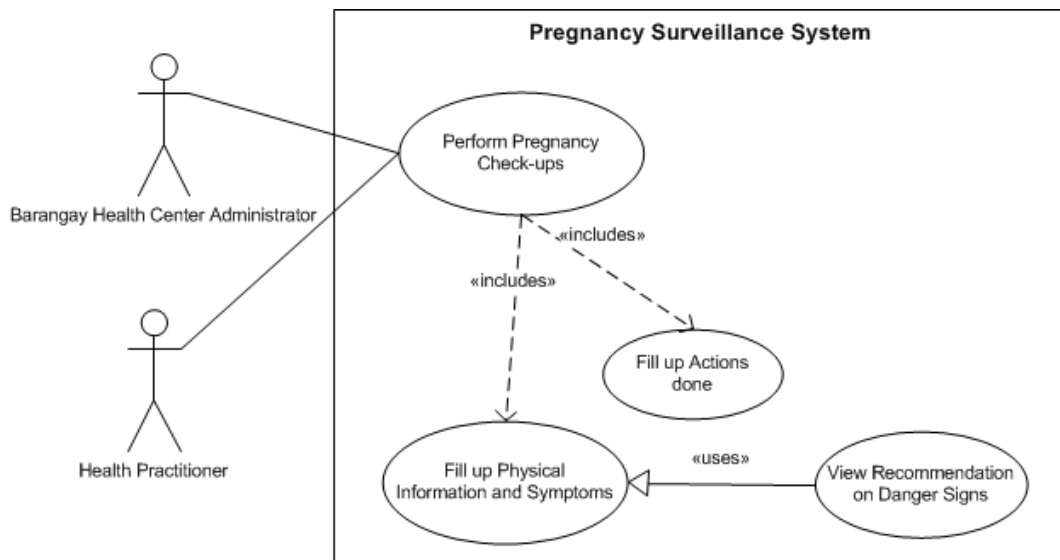


Figure 4.18: Perform Pregnancy Check-ups use case diagram of PregSS

Both the BHC Admin and Health Practitioner can perform pregnancy checkups and view alerts.

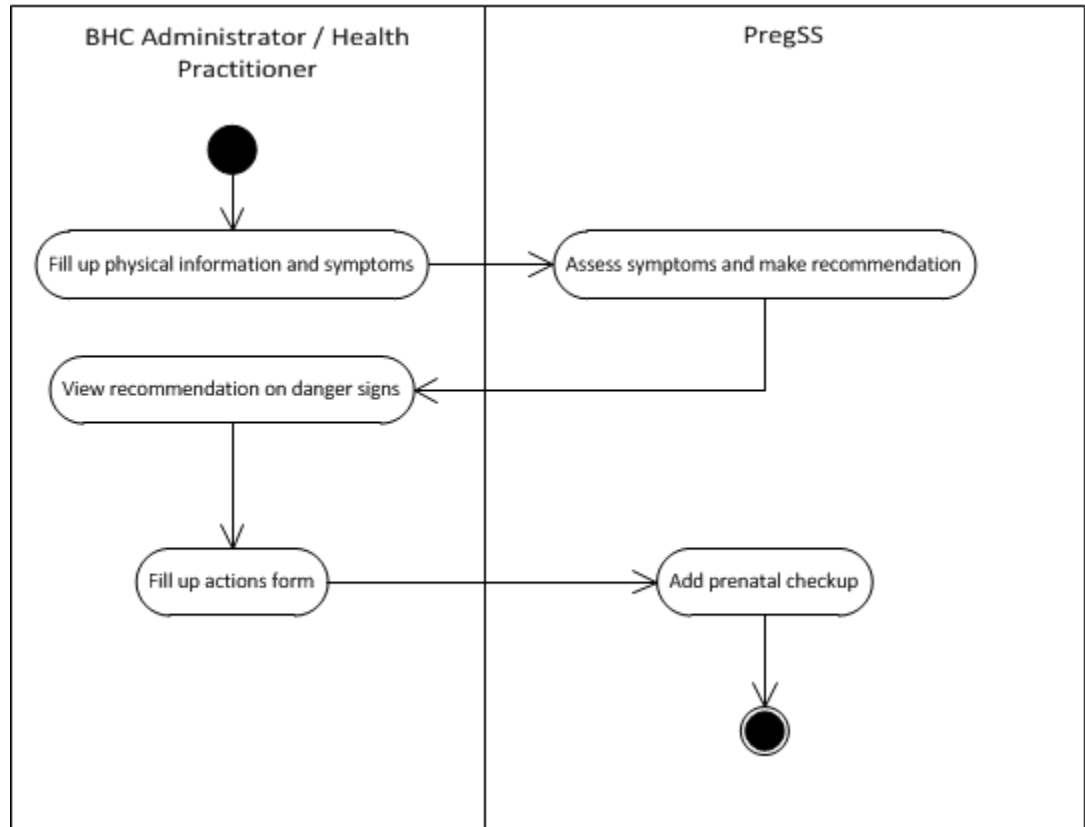


Figure 4.19: Perform Pregnancy Checkup Activity Diagram

The user first inputs the symptoms of the patient, after that, PregSS evaluates these symptoms based on the HBMR form, and then it outputs alerts on risks. Next, the user is required to input the mode of action done. Following that, the checkup information is added to the system.

6. Query Data as RDF

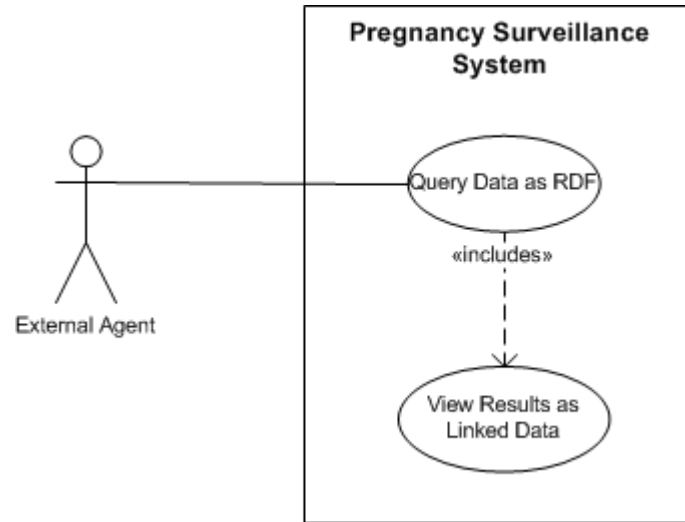


Figure 4.20: Query Data as RDF use case diagram of PregSS

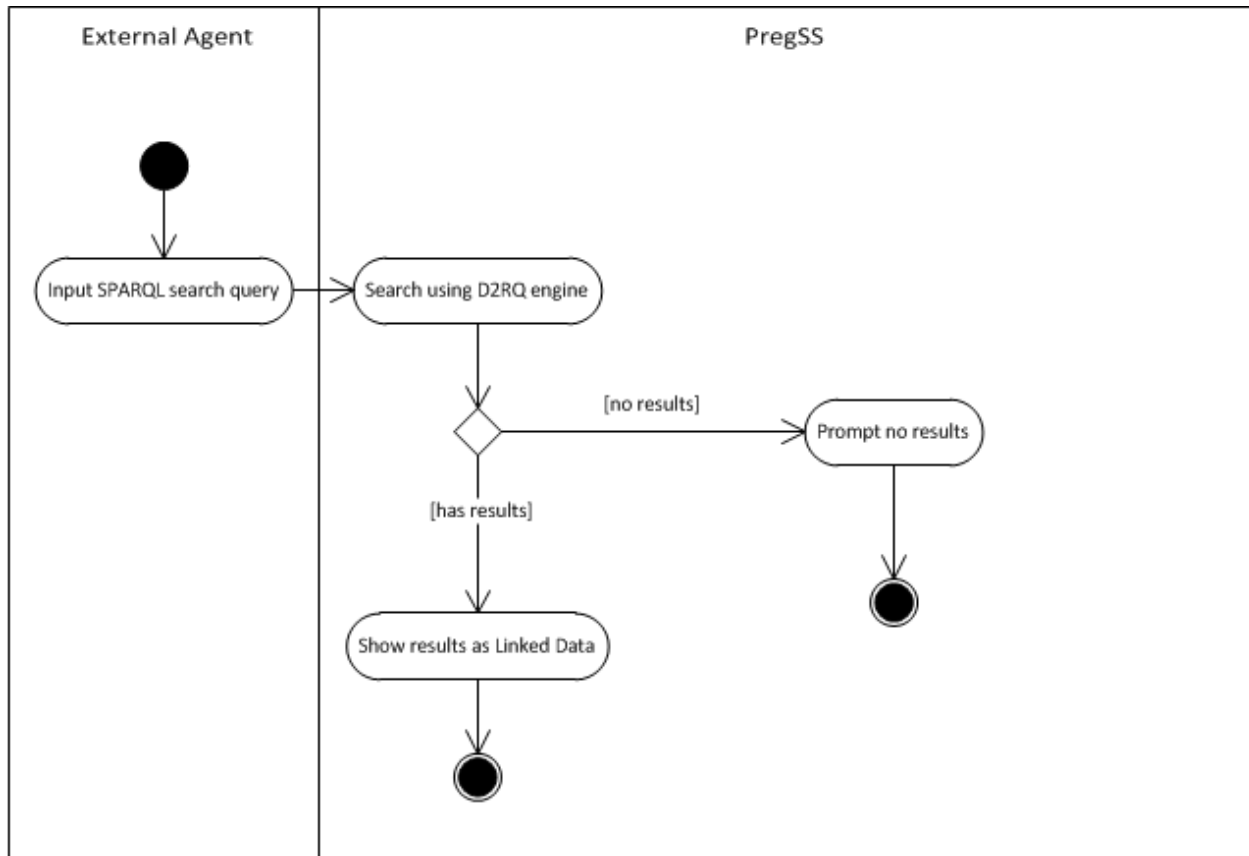


Figure 4.21: Query Data as RDF Activity Diagram

First an external agent accesses the D2RQ server URL, and from there he can use the AJAX-based SPARQL querier and then inputting a SPARQL query. The system then shows if there are results or none.

4. Entity Relationship Diagram

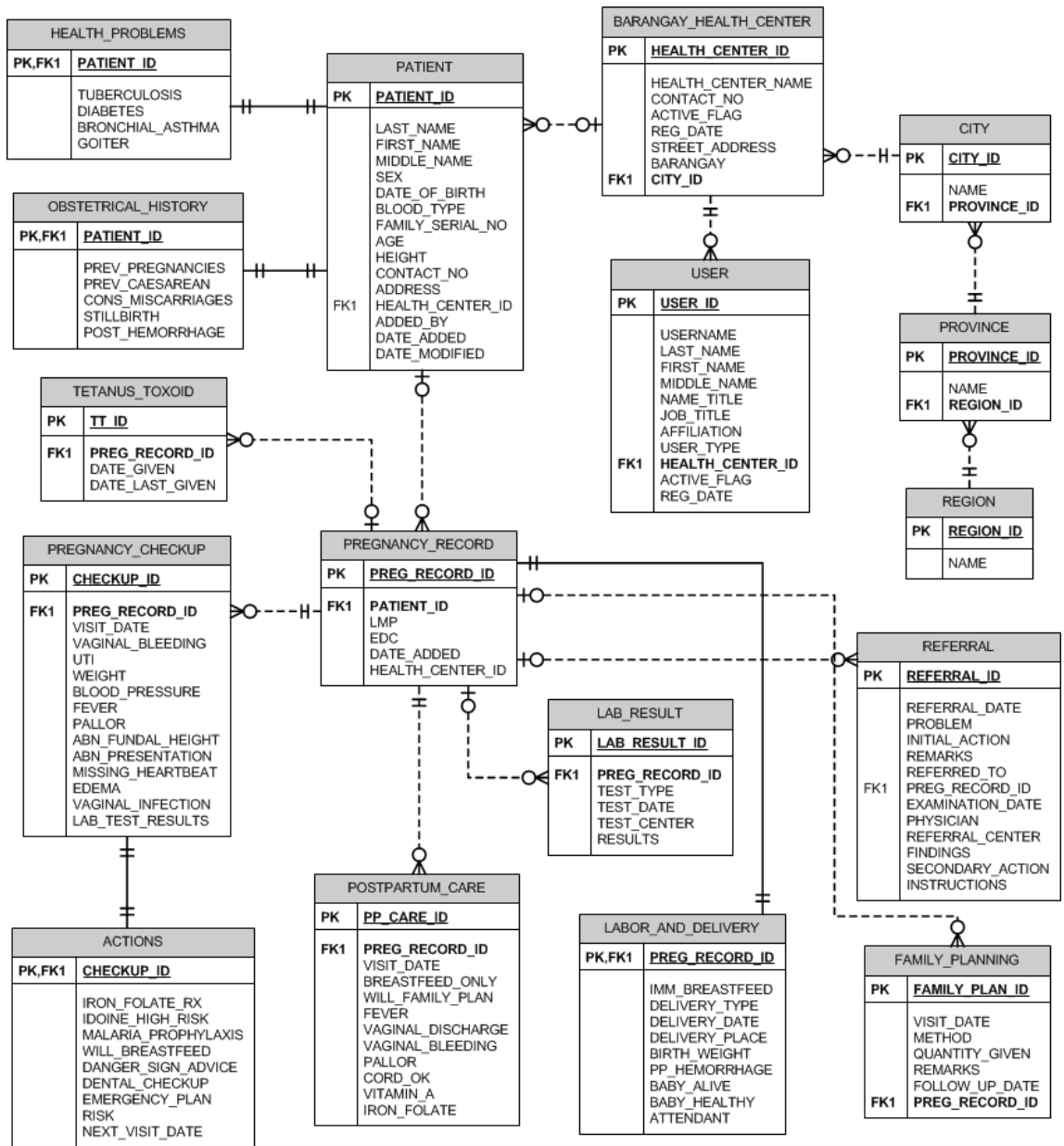


Figure 4.22: Entity Relationship Diagram for PregSS

PregSS is comprised of many tables, many of which revolve around the pregnancy record. There are the location identifiers, province, region, and city, which group the barangays and barangay health centers. Each barangay health center can have many

users, and also many patients. A patient, on the other hand, treats its health_center_id field as the current health center, since it's able to transfer to other health centers whenever needed. Linked to a patient is her health and obstetrical problems. A patient too, can have many pregnancy records, where each record has corresponding subparts. These include pregnancy checkups and actions, referrals, family planning sessions, tetanus toxoid history, postpartum cares and labor and delivery.

5. Data Dictionary

Attribute	Attribute type	Description
checkup_id	integer	primary and foreign key from the pregnancy_checkup table that identifies the checkup where this action corresponds
iron_folate_rx	integer	number of iron supplements given to the patient
iodine_high_risk	Integer	number of iodine supplements given to the patient
malaria_prophylaxis	varchar(1)	determines whether malaria prophylaxis was given or not
will_breastfeed	varchar(1)	determines whether the patient planned to breastfeed or not
danger_sign_advice	varchar(1)	determines whether advice on danger signs was given or not
dental_checkup	varchar(1)	determines whether dental checkup was recommended or not
emergency_plan	varchar(1)	determines whether plan for delivery and emergency was set or not
risk	varchar(1)	determines whether the patient was at risk or not
next_visit_date	date	date when the patient was to visit again

Table 4.1: Detailed description of the actions table

Attribute	Attribute type	Description
health_center_id	integer	primary key of the health center
health_center_name	varchar(100)	name of the health center
contact_no	varchar(40)	contact number of the health center
active_flag	varchar(1)	determines whether the health_center is active or not
reg_date	timestamp	date and time when the barangay health center is registered
street_address	varchar(120)	the house/block/building number with the street name
barangay	varchar(70)	the barangay name
city_id	integer	foreign key from the city table that determines what city this barangay belongs to

Table 4.2: Detailed description of barangay_health_center table

Attribute	Attribute type	Description
city_id	integer	primary key of the city
name	varchar(50)	name of the city
province_id	integer	foreign key from the province table that determines what province is this city located

Table 4.3: Detailed description of city_table

Attribute	Attribute type	Description
family_planning_id	integer	primary key of the family planning entry
visit_date	date	date when the patient visited
method	varchar(50)	family planning method given
quantity_given	varchar(100)	the quantity of medication given
remarks	text	additional remarks
follow_up_date	date	date when patient should follow up
preg_record_id	integer	foreign key from the pregnancy_record table that determines what record this family planning entry belongs to

Table 4.4: Detailed description of family_planning table

Attribute	Attribute type	Description
patient_id	integer	primary and foreign key from the patient table that identifies the patient who has these health problems
tuberculosis	varchar (1)	determines whether the patient has tuberculosis or not
diabetes	varchar(1)	determines whether the patient has diabetes or not
bronchial_asthma	varchar(1)	determines whether the patient has bronchial asthma or not
goiter	varchar(1)	determines whether the patient has goiter or not

Table 4.5: Detailed prescription of health_problems table

Attribute	Attribute type	Description
preg_record_id	integer	primary and foreign key from the pregnancy_record table that identifies which record has this labor and delivery details
imm_breastfeed	varchar(1)	determines whether the patient done breastfeeding immediately after delivery
delivery_type	varchar (40)	the type of delivery
delivery_date	date	date when the baby was delivered
delivery_place	varchar(70)	place where the baby was delivered
birth_weight	integer	the weight of the baby
post_hemorrhage	varchar(1)	determines whether postpartum hemorrhage occurred or not
baby_alive	varchar(1)	determines whether the baby was alive or not
baby_healthy	varchar(1)	determines whether the baby was healthy or not

Table 4.6: Detailed description of labor_and_delivery table.

Attribute	Attribute type	Description
lab_result_id	integer	primary key of the lab result
test_type	varchar (50)	the test type/name
date	date	date when the test was conducted
test_center	varchar(70)	laboratory/institution where the test was done
results	text	the results of the lab test
checkup_id	integer	foreign key from the pregnancy_checkup table that identifies what checkup this result corresponds to

Table 4.7: Detailed description of lab_result table

Attribute	Attribute type	Description
preg_record_id	integer	primary and foreign key from the pregnancy_record table that identifies which record has this obstetrical history
prev_pregnancies	integer	the number of pregnancies the patient has already had
prev_caesarean	varchar(1)	determines whether the last delivery

		type the patient had was caesarean or not
cons_miscarriages	varchar(1)	determines whether the patient had three consecutive miscarriages already or not
stillbirth	varchar(1)	determines whether the patient had stillbirth or not
post_hemorrhage	varchar(1)	determines whether the patient had postpartum hemorrhage or not

Table 4.8: Detailed description of obstetrical_history table

Attribute	Attribute type	Description
patient_id	integer	the id of the patient
last_name	varchar(40)	last name of the patient
first_name	varchar(40)	first name of the patient
middle_name	varchar(40)	middle name of the patient
sex	varchar(1)	sex of the patient
date_of_birth	date	the birth date of the patient
blood_type	varchar(3)	the blood type of the patient
family_serial_no	varchar(30)	the family serial number of the patient
address	varchar(255)	the address of the patient
contact_no	varchar(50)	the contact number of the patient
age	integer	age of the patient
height	integer	height of the patient
health_center_id	integer	foreign key from the barangay_health_center table that determines where this patient is currently under
added_by	varchar(32)	username of the user who added this patient record
date_added	timestamp	date and time this record was added
date_modified	timestamp	date and time this record was modified

Table 4.9: Detailed description of patient table

Attribute	Attribute type	Description
pp_care_id	integer	primary key of the postpartum care entry
preg_record_id	integer	foreign key from the pregnancy_record that determines what pregnancy record this entry

		corresponds to
visit_date	date	date of visit
breastfeed_only	varchar(1)	determines whether the patient will breastfeed exclusively or not
will_family_plan	varchar(1)	determines whether the patient will do family planning or not
fever	varchar(1)	determines whether the patient had fever or not
vaginal_discharge	varchar(1)	determines whether the patient was experiencing vaginal discharge or not
vaginal_bleeding	varchar(1)	determines whether the patient was experiencing vaginal bleeding or not
pallor	varchar(1)	determines whether the patient showed pallor or not
cord_ok	varchar(1)	determines whether the cord of the patient was ok or not

Table 4.10: Detailed description of postpartum_care table

Attribute	Attribute type	Description
checkup_id	integer	primary key of the pregnancy checkup
preg_record_id	integer	foreign key from the pregnancy_record table that determines what pregnancy record is this checkup under
visit_date	date	the date when the patient visited
vaginal_bleeding	varchar(1)	determines whether the patient was experiencing vaginal bleeding or not
uti	varchar(1)	determines whether the patient had urinary tract infection or not
weight	integer	the weight of the patient in kilograms
blood_pressure	varchar(10)	blood pressure reading of the patient
fever	varchar(1)	determines whether the patient had fever or not
pallor	varchar(1)	determines whether the patient showed pallor or not
abn_fundal_height	varchar(1)	determines whether the patient had abnormal fundal height or not
abn_presentation	varchar(1)	determines whether the patient had an abnormal presentation or not
missing_heartbeat	varchar(1)	determines whether the fetal heartbeat was missing or not
edema	varchar(1)	determines whether the patient had edema or not
vaginal_infection	varchar(1)	determines whether the patient had a vaginal infection or not

Table 4.11: Detailed description of pregnancy_checkup table

Attribute	Attribute type	Description
preg_record_id	integer	primary key of the pregnancy record
patient_id	integer	foreign key from the patient table that determines which patient has this pregnancy record
Imp	date	date of last menstrual period
edc	date	date of estimated confinement
date_added	timestamp	date this pregnancy record was added

Table 4.12: Detailed description of pregnancy_record table

Attribute	Attribute type	Description
province_id	integer	primary key of the province
name	varchar(50)	name of the province
region_id	integer	foreign key from the region table that determines where this province is located

Table 4.13: Detailed description of province table

Attribute	Attribute type	Description
referral_id	integer	primary key of the referral
referral_date	date	date when the referral is made
problem	text	problems identified by the doctor or midwife
action_taken	text	preliminary actions taken by the health care practitioner
referred_to	varchar(100)	where the patient is referred to by the health care practitioner
preg_record_id	integer	foreign key from the pregnancy_record table that identifies where this referral belongs to

Table 4.14: Detailed description of referral table

Attribute	Attribute type	Description
referral_id	integer	primary and foreign key from the referral table that determines which referral this result corresponds to
appointment_date	date	date of appointment on referral center
physician	integer	health care practitioner that conducted the examination
referral_center	varchar(100)	institution or clinic where the examination was conducted
findings	text	the findings from the examination
action_taken	text	actions taken by the physician
instructions	text	instructions given by the physician

Table 4.15: Detailed description of referral_result table

Attribute	Attribute type	Description
region_id	integer	primary key of the region
name	varchar(50)	name of the region

Table 4.16: Detailed description of region table

Attribute	Attribute type	Description
user_id	integer	the id of the user
username	varchar(32)	username of the user
password	varchar(64)	password of the user
last_name	varchar(40)	last name of the user
first_name	varchar(40)	first name of the user
middle_name	varchar (40)	middle name of the user
name_title	varchar(20)	title of the user
job_title	varchar(70)	the job of the user
affiliation	varchar(70)	institution where the user is under
user_type	varchar(40)	the type of the user (master, admin, or practitioner)
email_address	varchar(254)	the email address of the user
active_flag	varchar(1)	determines whether the user account is active or not
health_center_id	integer	foreign key from the barangay_health_center table that identifies which health center this user is under
reg_date	timestamp	date and time when the user registered

Table 4.17: Detailed description of user table

Attribute	Attribute type	Description
tt_id	integer	primary key of the tetanus toxoid vaccination
date_given	date	date when tetanus toxoid was taken
patient_id	integer	foreign key from the patient table that determines which patient had this vaccination

Table 4.18: Detailed description of tetanus_toxoid table

V. Technical Architecture

To run PregSS, the following are required:

- Apache Tomcat 6.0
- Java Standard Edition Runtime Environment (JRE) or JDK 5.0 or later
(required by Tomcat 6)
- MySQL 5
- Any operating system that supports the said software

The server has been tested to run on a machine with the following specifications:

Operating System: Windows 7

CPU: Quad-core processor @ 2.27 GHz

Memory: 8.00 GB RAM

To access the system, only a web client is required. Javascript should also be supported. Tested on Chrome.

VI. Results

A. PregSS

The following screenshots shows PregSS' pages, and what all the users (unregistered users, health practitioners, barangay health center administrators, PregSS administrator) are allowed to do in the system.



Figure 6.1: PregSS Homepage



Figure 6.2: Login Page, if the user decided not to log in from the Home Page

Register Barangay Health Center

Please enter your barangay health center information and user details. Field names with asterisks(*) are required.

Region*:	<input type="text" value="NCR"/>
Province*:	<input type="text" value="Metro Manila"/>
City*:	<input type="text" value="-----"/>
Barangay*:	<input type="text"/>
Street Address*:	<input type="text"/>
Health Center Name*:	<input type="text"/>
Health Center Contact Number:	<input type="text"/>

Administrator Account

Name*:	<input type="text"/>	<input type="text"/>	<input type="text"/>
	First Name	Middle Name	Last Name
Job Title*:	<input type="text" value="General Practitioner"/>		
Affiliation:	<input type="text"/>		
Username*:	<input type="text"/>		
Password*:	<input type="text"/>		
Repeat Password*:	<input type="text"/>		
Email Address:	<input type="text"/>		
Contact Number:	<input type="text"/>		

Figure 6.3: Register a Barangay Health Center and also a corresponding Administrator Account

Search Patient: Last Name , First Name Middle Name Scope: Local

You are logged in as [daweee](#). [Logout](#)

Account Details

Name*:
First Name **Middle Name** **Last Name**

Job Title*:

Affiliation:

Username:

New Password:

Repeat New Password:

Email Address:

Contact Number:

Figure 6.4: Edit Account Details

PregSS

Pregnancy Surveillance System

Home Patient List New Patient Generate Report Health Center Administrator

Search Patient: Last Name , First Name Middle Name Scope: Local


You are logged in as [daweee](#). [Logout](#)

Patient List - Moonwalk Health Center

Name	Address
First, Eve The	Garden of Eden
Patient, Sample Pregnant	Sample St., Sample Avenue, Sample City

« First Prev 1 Next » End »

Figure 6.5: Logging in as a Health Center Administrator or Health Practitioner leads to the Patient List page

Personal Information 

Name: First, Eve The
Sex: F
Blood Type: AB
Family Serial No.: 123
Age: 23
Height: 153
Address: Garden of Eden
Contact No.: 1234567

Close observation or action is recommended due to the following:
 ◦ The patient has tuberculosis.



 Perform Pregnancy Check-up  Create New Pregnancy Record

Figure 6.6: The top part of the View Patient page, containing the personal information, and the alerts depending on the patient’s health problems

History Present Pregnancy Labor and Delivery Family Planning Postpartum Care Referrals

Obstetrical History 

Previous Pregnancies: 2
 Previous Caesarean Section: N
 3 Consecutive Miscarriages: N
 Stillbirth: N
 Post-Partum Hemorrhage: N

Health Problems 

Tuberculosis (14 Days+ of Cough): Y
 Heart Disease: N
 Diabetes: N
 Brochial Asthma: N
 Goiter: N

Tetanus Toxoid 

Tetanus Toxoid No.	Date Given
1	04/11/2013

Figure 6.7: The History tab of the Patient Record, containing Obstetrical History, Health Problems, and Tetanus Toxoid History

[< Back to Patient](#)

Obstetrical History

Previous Pregnancies:

Previous Caesarean Section: No Yes

3 Consecutive Miscarriages: No Yes

Stillbirth: No Yes

Post-Partum Hemorrhage: No Yes

Figure 6.8: Edit Obstetrical History

[< Back to Patient](#)

Health Problems

Tuberculosis (14 Days+ of Cough): No Yes

Heart Disease: No Yes

Diabetes: No Yes

Brochial Asthma: No Yes

Goiter: No Yes

Figure 6.9: Edit Health Problems

[< Back to Patient](#)

Tetanus Toxoid

Tetanus Toxoid No.	Date Given
1	04/11/2013

Figure 6.10: Edit Tetanus Toxoid History

History Present Pregnancy **Labor and Delivery** Family Planning Postpartum Care Referrals

Labor and Delivery

Immediate Breastfeeding: Y
Type of Delivery: Caesarean
Date of Delivery: 04/03/2013
Place of Delivery: Paranaque Medical Center
Birth Weight in Grams: 2340
Post-Partum Hemorrhage 500 CC+: N
Baby Alive: Y
Baby Healthy: Y
Birth Attendant:

Figure 6.11: Labor and Delivery Tab of the Patient Record

[< Back to Patient](#)

Labor and Delivery

Immediate Breastfeeding: No Yes

Type of Delivery: Caesarean ▾

Date of Delivery: 04/03/2013 

Place of Delivery: Paranaque Medical C

Birth Weight in Grams: 2340

Post-Partum Hemorrhage 500 CC+: No Yes

Baby Alive: No Yes

Baby Healthy: No Yes

Birth Attendant:

Figure 6.12: Edit Labor and Delivery

[History](#) [Present Pregnancy](#) [Labor and Delivery](#) [Family Planning](#) [Postpartum Care](#) [Referrals](#)

Family Planning

Date of Visit	Method	Quantity Given	Remarks	Date of Follow-up	
2013-04-03	Contraceptives	28 tablets	Drink tablets once a day.	05/03/2013	
2013-04-18	Contraceptives	28 tablets	Drink thrice a week.	05/03/2013	

[Add Family Planning](#)

Figure 6.13: Family Planning Tab of the Patient Record

[< Back to Patient](#)

Family Planning

Date of Visit: 

Method:

Quantity Given:

Remarks:

Date of Follow-up: 

Figure 6.14: Add Family Planning

History Present Pregnancy Labor and Delivery **Family Planning** Postpartum Care Referrals

Postpartum Care


Date of Visit	04/11/2013	04/12/2013
Exclusive Breastfeeding	N	Y
Intends to use Family Planning	N	N
Fever 38°C and Above	N	N
Foul-smelling Vaginal Discharge	N	N
Excessive Vaginal Bleeding	N	N
Pallor	N	N
Cord OK?	N	N
Vitamin A 200,000 IU	N	Y
Iron/Folate #/RX	45	50
	Edit	Edit

[Add Postpartum Care](#)

Figure 6.15: Postpartum Care Tab of the Patient Record

[< Back to Patient](#)

Postpartum Care

Date of Visit: 

Exclusive Breastfeeding: No Yes

Intends to use Family Planning: No Yes

Fever 38°C and Above: No Yes

Foul-smelling Vaginal Discharge: No Yes

Excessive Vaginal Bleeding: No Yes

Pallor: No Yes

Cord OK?: No Yes

Vitamin A 200,000 IU: No Yes

Iron/Folate #/RX: tablets

Figure 6.16: Add Postpartum Care

History Present Pregnancy Labor and Delivery Family Planning Postpartum Care Referrals


Referrals

This patient has no referral records yet. [Add Referral](#)

Figure 6.17: Referrals Tab of the Patient Record, in this case, it's still empty

[< Back to Patient](#)

Referral

Referral Date: 

Problem:

Action Taken:

Remarks:

Referred To:

Figure 6.18: Add Referral Form

Present Pregnancy

LMP:04/02/2013 EDC:04/02/2013


TRIMESTER	1ST	2ND	3RD
AOG in Months	0	0	
Date of Visit	04/02/13	04/04/13	
Vaginal Bleeding	N	N	
Urinary Tract Infection	N	N	
Weight in Kg.	0	0	
Blood Pressure	0/0	0/0	
BP 140/90 and Above	N	N	
Fever 38°C and Above	N	N	
Pallor	N	N	
Abnormal Fundal Height	N	N	
Abnormal Presentation	N	N	
Missing Fetal Heartbeat	N	N	
Edema	N	N	
Vaginal Infection	N	N	
ACTION:	Edit		
Iron/Folate #/RX		0	
Iodine Supplementation		Y	
Malaria Prophylaxis		N	
Intends to Breastfeed?		Y	
Advice on 4 Danger Signs		Y	
Dental Check-up?		N	
Emergency Plans		Y	
Risk?		N	
Date of Next Visit		04/09/13	
Action		Edit	

[Perform Pregnancy Check-up](#)

Figure 6.19: Present Pregnancy Tab of the Patient Record, containing the Prenatal Checkups information

[< Back to Patient](#)

Pregnancy Checkup

Date of this Visit: 

Vaginal Bleeding: No Yes

Urinary Track Infection: No Yes

Weight in kg.:

Blood Pressure: /

Fever 38°C and above: No Yes

Pallor: No Yes

Abnormal Fundal Height: No Yes

Abnormal Presentation: No Yes

Missing Fetal Heartbeat: No Yes

Edema: No Yes

Vaginal Infection: No Yes

Figure 6.20: Perform Pregnancy Checkup

Patient should be referred to a physician/RHU due to the following symptoms:

- o Fever
- o Vaginal Infection

Figure 6.21: Alerts Based on Symptoms

Action

Iron/Folate #/RX: tablets

Iodine Supplementation in High Risk Areas: No Yes

Malaria Prophylaxis: No Yes

Mother intends to breastfeed? No Yes

Advice on 4 Danger Signs: No Yes

Dental Check-up? No Yes

Emergency Plans and Place of Delivery: No Yes

Risk? No Yes


Date of Next Visit: 

Figure 6.22: Corresponding Action Form

Home Patient List New Patient Generate Report Health Center Administrator

Search Patient: Last Name Middle Name Scope: Local

You are logged in as [daweee](#). [Logout](#)

Search Results

Name	Address
First, Eve The	Garden of Eden
Patient, Sample Pregnant	Sample St., Sample Avenue, Sample City

<< First < Prev 1 Next > End >>

Figure 6.23: Local Patient Search

Home Patient List New Patient Generate Report Health Center Administrator

Search Patient: Last Name , e Middle Name Scope: All

You are logged in as [daweee](#). [Logout](#)

Search Results

Name	Address	Health Center
First, Eve The	Garden of Eden	Moonwalk Health Center
Patient, Sample Pregnant	Sample St., Sample Avenue, Sample City	Moonwalk Health Center
Test, Test Patient Test	157 Judea St.	LP Health Center

<< First < Prev 1 Next > End >>

Figure 6.24: Search all of patients under PregSS

Search Patient: Last Name , First Name Middle Name Scope: Local

You are logged in as [daweee](#). [Logout](#)

Personal Information

This patient is currently under a different health center. Do you want to transfer the patient? [Transfer](#). 

Name: Test, Test Patient Test
Sex: F
Blood Type: AB
Family Serial No. 0123
Age: 25
Height: 124
Address: 157 Judea St.
Contact No.: 8695645

Hospital delivery and referral is recommended due to the following:

- o The patient has goiter.
- o The patient has tuberculosis.
- o The patient had previously undergone Caesarean section.
- o The patient's height is below 145 cm.

History Present Pregnancy Labor and Delivery Family Planning Postpartum Care Referrals

Figure 6.25: If patient is from a different health center, it asks if you want to transfer the patient to your health center

Transfer Patient

You are about to transfer patient 'Test, Test Patient Test' to your health center, *Moonwalk Health Center*.

[Continue](#)

Figure 6.26: Transfer Confirmation Screen

The screenshot displays the PregSS (Pregnancy Surveillance System) interface. At the top, the title 'PregSS' is prominently displayed in a large, stylized font, with the subtitle 'Pregnancy Surveillance System' below it. A navigation bar includes links for 'Home', 'Patient List', 'New Patient', 'Generate Report', and 'Health Center Administrator'. Below the navigation bar is a search area with input fields for 'Last Name', 'First Name', and 'Middle Name', along with a 'Scope' dropdown menu set to 'Local' and a 'Search' button. A status message indicates the user is logged in as 'daweee' with a 'Logout' link. The main content area is titled 'Control Panel' and 'Manage Users', featuring a 'New User' button. A table lists the current users, and navigation controls are provided at the bottom of the table.

Username	Last Name	First Name	Job Title
doctor	Where	Who	General Practitioner
testdoctor	Doom	Test Doctor	General Practitioner

<< First < Prev 1 Next > End >>

Figure 6.27: Manage Users in Health Center Administrator Control Panel

Search Patient: , Scope:

You are logged in as [daweee](#). [Logout](#)

Generate Report

(HIS)FHSIS Monthly Report

Month:

Year:

Test Footer

Figure 6.28: Generate FHSIS Monthly Report

Moonwalk Health Center: HIS (FHSIS) Report for April 2013	
MATERNAL CARE	NO.
Pregnant with 3 or more Prenatal visits (w/ at least 1 visit per trimester)	0
Pregnant given TT2 plus	1
Pregnant given Complete Iron dosage	0
Postpartum (PP) w/ at least 1 PP visit	0
PP mothers given Complete Iron dosage	1
PP women initiated breastfeeding	1
Breastfeeding women given Vitamin A	1
Women 15-49 given iodized oil capsules	1

Figure 6.29: Sample FHSIS Report in Excel

You are logged in as [claweee](#). [Logout](#)

Control Panel

[Manage Users](#)
[Update Health Center Details](#)

User Details

Name*:

First Name
Middle Name
Last Name

Job Title*:

Affiliation:

Username*:

New Password:

Repeat Password:

Email Address:

Contact Number:

Figure 6.30: Edit User Details Form

Home
PregSS Administrator

You are logged in as [pregssadmin](#). [Logout](#)

Control Panel

[Manage Health Centers](#)
[Health Center Requests \(0\)](#)

Select Location

Region:

Province:

City:

Barangay Filter:

Barangay	Health Center Name	Street Address
Barangay Moonwalk Health Center	Moonwalk Health Center	Armstrong Ave.
Multinational Village	MultiVille Health Center	123 John St.
Sto. Nino	Barangay Sto. Nino Health Center	123 Block 1 Lot 5 Fast Track Ave.

<< First < Prev 1 Next > End >>

Figure 6.31: Manage Health Centers on PregSS Administrator Control Panel

Home | PregSS Administrator

You are logged in as [pregssadmin](#). [Logout](#)

Control Panel

[Manage Health Centers](#)
[Health Center Requests \(1\)](#)

Health Center Requests

Health Center Name	Street Address	Barangay	City	Province	Region	Registration Date	Action
Legaspi Village Health Center	12 Rada St.	Legaspi Village	Makati	Metro Manila	NCR	2013-04-10 12:48:47.0	Approve Reject

<< First < Prev 1 Next > End >>

Figure 6.32: Health Center Requests on PregSS Administrator Control Panel

Control Panel

[Manage Health Centers](#)
[Health Center Requests \(1\)](#)

Health Center Details

Region*:

Province*:

City*:

Barangay*:

Street Address*:

Health Center Name*:

Health Center Contact Number:

Figure 6.33: Edit Barangay Health Center Details

B. OpenGALEN to PregSS Mapping

The distributed OpenGALEN ontology is available both in OWL-F and RDF. In this case, I used the RDF format so it is compatible with D2RQ. The package is composed of different medical fields (referred in the ontology as chapters), including ones for Nutrition, Respiratory, Musculoskeletal, etc. The main one used here and to be loaded in Protégé (along with its dependencies) is the Obstetrics chapter.

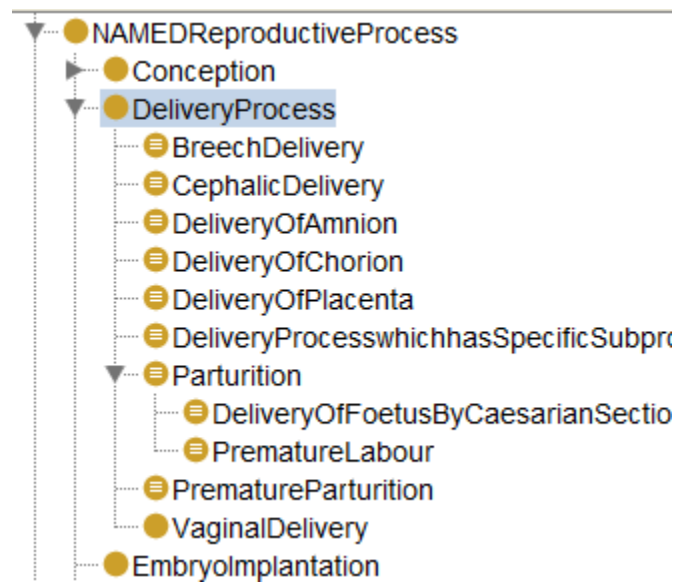


Figure 6.34: OpenGALEN Obstetrics as viewed through Protégé

OpenGALEN is a very large ontology, although we cannot say it can completely represent all the possible concepts in the field. But, to be able to map the terms from PregSS' relational database to the OpenGALEN ontology with as much accuracy and consistency possible, I enlisted the help of Dr. Anonuevo, who specializes in Obstetrics and Gynecology, in validating the fields matched from the database to the available classes in OpenGALEN.

First, with the use of a D2RQ tool called generate-mapping, we automatically get an RDF representation of the local database. Next step is to remove the tables or columns that we don't want exposed or are unnecessary. Example of which are patients' personal data. Afterwards, we attempt to find a possible match for every table and field to an ontology class, using Protégé to find similar terms and equivalent classes. Finally, the ones left with no mapping will use the PregSS namespace.

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/config.rdf#> .
@prefix meta: <http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/metadata#> .
```

Figure 6.35: The header of a mapping file, which contains all the namespace prefixes along with their equivalent URIs

```
<> a d2r:Server;
  rdfs:label "PregSS D2R Server";
  d2r:baseURI <http://agila.upm.edu.ph:8083/pregssd2rq/>;
  d2r:port 2020;
  d2r:vocabularyIncludeInstances true;
```

Figure 6.36: Following that is an excerpt of the configuration block, which defines connection details if D2RQ is to be deployed as a web application

```
map:postpartum_care_pallor a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:postpartum_care;
  d2rq:property OpenGALEN:PallorSymptom;
  d2rq:propertyDefinitionLabel "postpartum_care pallor";
  d2rq:column "postpartum_care.pallor";
  .
```

Figure 6.37: A sample mapping block which maps the class OpenGALEN:PallorSymptom to the database field pregnancy_checkup.pallor

Consequently, not all of the fields in the PregSS system were mapped to the ontology. Most are similar, but some are not quite exact, inconsistent or are of different contexts. That leaves us with mixed namespaces to query PregSS data through D2RQ.

C. The D2RQ Endpoint

Visiting the URL of the D2RQ endpoint, we'll see the generated homepage:



PregSS D2RQ Server
Running at <http://agila.upm.edu.ph:8083/pregssd2rq/>

Home | [action](#) [barangay_health_center](#) [city](#) [family_planning](#) [health_problems](#) [lab_result](#) [labor_and_delivery](#) [obstetrical_history](#) [patient](#) [postpartum_care](#) [pregnancy_checkup](#) [pregnancy_record](#) [province](#) [referral](#) [region](#) [tetanus_toxoid](#)

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

1. HTML View

You can use the navigation links at the top of this page to explore the database.

2. RDF View

You can also explore this database with **Semantic Web browsers** like [Disco](#) or [Marbles](#). To start browsing, open this entry point URL in your Semantic Web browser:

<http://agila.upm.edu.ph:8083/pregssd2rq/all>

3. SPARQL Endpoint

SPARQL clients can query the database at this SPARQL endpoint:

<http://agila.upm.edu.ph:8083/pregssd2rq/sparql>

The database can also be explored using [this AJAX-based SPARQL Explorer](#).

Figure 6.38: The D2RQ Server Homepage for PregSS

You can browse using the table names at the top, and see different resources as linked data.

Property	Value
PregSS:AbnormalFundalHeight	N
OpenGALEN:BodyWeightSymptom	0 (xsd:integer)
OpenGALEN:CurrentBloodPressureLevel	150/90
OpenGALEN:Fever	N
PregSS:MissingFetalHeartbeat	N
OpenGALEN:OedemaSymptom	N
OpenGALEN:PallorSymptom	N
OpenGALEN:UrinaryTractInfection	N
OpenGALEN:VaginalHaemorrhageSymptom	N
PregSS:VaginalInfection	N
PregSS:VisitDate	2013-05-22 (xsd:date)
is PregSS:checkup_id of	<http://agila.upm.edu.ph:8083/pregssd2rq/resource/action/1>
PregSS:checkup_id	1 (xsd:integer)
rdfs:label	pregnancy_checkup #1
PregSS:preg_record_id	<http://agila.upm.edu.ph:8083/pregssd2rq/resource/pregnancy>
OpenGALEN:presencewhichisExistenceOfFetalMalpresentation	N
rdf:type	PregSS:PregnancyCheckup

Figure 6.39: a pregnancy_checkup entry viewed through D2RQ, notice that some of the properties are named using the OpenGALEN namespace

Asides from viewing the data in PregSS as linked data, one can query the database through SPARQL. Clicking AJAX-based SPARQL Explorer on the homepage leads to:

SPARQL:

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX db: <http://agila.upm.edu.ph:8083/pregssd2rq/resource/>
PREFIX OpenGALEN: <http://www.openGALEN.org/owl/openGALEN.owl#>
PREFIX d2r: <http://sites.wiwiwss.eu-berlin.de/suhl/bizer/d2r-server/config.rdf#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX map: <http://agila.upm.edu.ph:8083/pregssd2rq/resource/#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vocab: <http://agila.upm.edu.ph:8083/pregssd2rq/resource/vocab/>
PREFIX meta: <http://www4.wiwiwss.eu-berlin.de/bizer/d2r-server/metadata#>
PREFIX PregSS: <http://www.pregss.com/owl/pregss.owl#>

SELECT DISTINCT * WHERE {
  ?s ?p ?o
}
LIMIT 10

```

Browse:

- [Classes](#)
- [Properties](#)

Figure 6.40: Snorql, a SPARQL explorer similar to SQL query boxes

One seemingly big problem when querying the database here is the unfamiliarity on the classes and properties used in the system. However, as seen from the above image, at the upper right, there are Browse Classes and Properties links. To show:

class
OpenGALEN:Patient
PregSS:Action
PregSS:BarangayHealthCenter
PregSS:City
PregSS:FamilyPlanning
PregSS:HealthProblems
PregSS:LaborAndDelivery
PregSS:ObstetricalHistory
PregSS:PostPartumCare
PregSS:PregnancyCheckup
PregSS:PregnancyRecord
PregSS:Province
PregSS:Referral
PregSS:Region
PregSS:TetanusToxoid

Figure 6.41: List of RDF Classes in PregSS

List of all properties:

property
OpenGALEN:Asthma
OpenGALEN:BodyWeightSymptom
OpenGALEN:CholoroquinePhosphate
OpenGALEN:CurrentBloodPressureLevel
OpenGALEN:DeliveryProcess
OpenGALEN:DiabetesMellitus
OpenGALEN:Fever
OpenGALEN:Goiter
OpenGALEN:HeartDisease
OpenGALEN:IodineAsDrug
OpenGALEN:IronDrugCompound
OpenGALEN:OedemaSymptom
OpenGALEN:PallorSymptom
OpenGALEN:PreviousPersonalHistorywhichisHistoryofCaesareanSection
OpenGALEN:Tuberculosis
OpenGALEN:UrinaryTractInfection
OpenGALEN:VaginalDischargeSymptom
OpenGALEN:VaginalHaemorrhageSymptom
OpenGALEN:VitaminAAsDrug
OpenGALEN:presencewhichisExistenceOfBreastFeeding
OpenGALEN:presencewhichisExistenceOfFetalMalpresentation
OpenGALEN:presencewhichisExistenceOfPostpartumHaemorrhage
PregSS:AbnormalFundalHeight
PregSS:AdvisedDangerSigns

Figure 6.42: An excerpt from the list of all properties available in PregSS

D. Querying with SPARQL

Here, I will show sample queries that may give you an idea on how other systems might query PregSS through SPARQL.

To find a patient object which has Goiter as one of her present health problems, we call this:

```
SELECT ?patient WHERE {  
  ?hProbs OpenGALEN:Goiter 'Y' .  
  ?hProbs PregSS:patient_id ?patient .  
}
```

As seen from above, we want to retrieve the ?patient variable. From the first line it means, get the ?hProbs object (referred to by its id, in the RDB it's patient_id as a foreign and primary key from patient) which has OpenGALEN:Goiter property equal to Y. This retrieves all health_problems resources with goiter = 'Y'. Then, to get the patient object, we have the second line, because the health_problems class has the said property PregSS:patient_id. This results to the patient, defined by its patient_id, satisfying the conditions.

```
SELECT ?patient WHERE {
  ?hProbs OpenGALEN:Goiter 'Y' .
  ?hProbs PregSS:patient_id ?patient .
}
```

Results:

SPARQL results:

patient
db:patient/7 [↗](#)

Figure 6.43: Running a simple SPARQL query

```
SELECT ?patient WHERE {
  ?laborDel OpenGALEN:DeliveryProcess 'Caesarean' .
  ?laborDel PregSS:preg_record_id ?pregRecord .
  ?pregRecord PregSS:pregnancy_record_patient_id ?patient .
}
LIMIT 10
```

Results:

SPARQL results:

patient
db:patient/3 [↗](#)

Figure 6.44: This time, finding patients which had Caesarean as her delivery type

Noticing the URL, we see the same query, but with escaped characters:

```
http://agila.upm.edu.ph:8083/pregssd2rq/snorql/?query=SELECT+%3Fpatient+WHERE+%  
7B%0D%0A++%3FlaborDel+OpenGALEN%3ADeliveryProcess+%27Caesarean%27+.%0D%  
0A++%3FlaborDel+PregSS%3Apreg_record_id+%3FpregRecord+.%0D%0A++%3FpregRecor  
d+PregSS%3Apregnancy_record_patient_id+%3Fpatient+.%0D%0A%7D%0D%0ALIMIT+1  
0
```

This means that PregSS data could be queried directly and easily through the URL.

VII. Discussion

The alert system implemented shows the highest danger level recommended for the patient. For example, if one risk exists that recommends the patient to go to the hospital, all other lower recommendations such as close observation by midwife are overridden. If the patient transfers residence, PregSS allows the health practitioners or barangay health center administrators to transfer the patient record under the current barangay health center. For this reason, pregnancy records are thus tied to the barangay health center, so that even if the patients changed residences, reports for archived records still reflect the original one.

For the FSHIS reports, the computation for the indicators was based upon the conditions stated in the eFHSIS manual. It is computed from the fields in the Home Based Mother's Record without requiring additional fields.

For the semantic interoperability part, the result is that not all of the PregSS relational database fields have direct mappings with the OpenGALEN ontology. However, it covered most of the pregnancy_checkup and health_problem tables, which is one of the more significant tables in PregSS. Despite that, it was still able to represent the data in PregSS as another format, which makes it more open to mining. In the event that the ontology and database are changed or updated, the mapping file should be updated directly by the system or ontology expert. Also, because of confidentiality purposes, we completely removed the patient information from the mappings except its id.

It is evidently possible for semantic interoperability on systems without sticking to a coding standard. But it must be kept in mind that the ontology to be used must be highly

sufficient in describing the context. To solve that, the ontology could be extended or utilizing multiple ontologies could be more feasible.

VIII. Conclusion

PregSS is a web-based system that can be very helpful to rural health centers, especially for health care workers in barangay health centers, because it acts as a clinical decision-support system that helps in alerting the health risks of pregnant patients. The system is also able to provide maternal health indicator reports for every barangay, and since it is equivalent to the FHSIS monthly form, it is easily understandable by the DOH, making it possible to determine which areas need attention.

The system is also able to address the issue of semantic interoperability by making the use of an ontology by mapping it to the relational database, thus specifying the meaning of the data used in the system. While not fully complete, it could still be improved by extending the ontology with the use of ontology engineering methodologies. This makes it easier for other systems that have access to the same ontology to interpret the maternal-related data whenever data is exchanged. Also, it is shown here that using an ontology makes the system structure more flexible compared to using coding standards. Finally, it allows research in the pregnancy field by making the data publicly available, while demonstrating one of the ways ontologies can be incorporated into a system.

IX. Recommendation

The OpenGALEN is extendable or modifiable depending on the needs of the system, or based on the current trends in health, in general. Probably, a multiple ontology model may be considered. Better yet, developing an ontology for the use of most Philippine Health Services is recommended. For the PregSS system, it's better if it is able support uploading of files such as images for laboratory results. To improve reports, it's more understandable if there's a choice to provide visual representations such as graphs. Moreover, since it already has information distributed into barangays, cities or municipalities, provinces and regions, introducing geographical information technologies potentially increases the system's usability.

X. Bibliography

1. United Nations Millennium Development Goals. *United Nations Millennium Development Goals*. [Online] [Cited: April 4, 2013.] <http://www.un.org/millenniumgoals/maternal.shtml>.
2. **Alave, Kristine L.** Maternal mortality rate rose in 2011, says DOH. *INQUIRER.net*. [Online] June 18, 2012. [Cited: October 3, 2012.] <http://newsinfo.inquirer.net/214829/maternal-mortality-rate-rose-in-2011-says-doh>.
3. *Building and Reusing Medical Ontology for Tropical Diseases Management*. **Haikal Satria, Radha S. Priya, Lukman H. Ismail, Eko Supriyanto**. 1, s.l. : International Journal of Education and Information Technologies, 2012, Vol. 6.
4. *Enterprise Architecture Version 1.0*. [PDF File] s.l. : Department of Health - Information Management Service, 2011.
5. **Gagnon, Michel**. *Ontology-Based Integration of Data Sources*. [PDF Document] Québec, Canada : Defence R&D Canada - Valcartier.
6. **Canlas, Ruben D.** *PHIS: The Philippine Health Information System (Critical Challenges and Solutions)*. [PDF File] 2009.
7. *Development of Ontology for Smart Hospital and Implementation using UML and RDF*. **Sanjay Anand, Akshat Verma**. 5, s.l. : IJCSI International Journal of Computer Science Issues, 2010, Vol. 7.
8. **Wen Yao, Chao-Hsien Chu, Akhil Kumar, Zang Li**. Using Ontology to Support Context Awareness in Healthcare. *Penn State Personal Web Server*. [Online] [Cited: 10 21, 2012.] http://www.personal.psu.edu/wxy119/pub/WITS09-Wen-Camera_Ready_final.pdf.
9. *Design and Implementation of an Ontology-Based Psychiatric Disorder Detection System*. **Lin, Y. C.** 1, s.l. : WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, 2010, Vol. 7.
10. **Patty Kostkova, Anand Kumar, et al.** Ontological Principles of Disease Management from Public Health Perspective: a Tuberculosis Case Study. *Universitat Leipzig*. [Online] [Cited: October 20, 2012.] http://www.uni-leipzig.de/~akumar/TB_Onto.pdf.
11. *GalenOWL: Ontology based drug recommendations discovery*. **Charalampos Doulaverakis, George Nikolaidis, Athanasios Kleontas, Ioannis Kompatsiaris**. s.l. : International Conference on Biomedical Ontology, 2012.
12. *A Telepsychiatry Domain Ontology Applied to Healthcare Information Systems*. **Fabricio da Costa Dias, Ryan Ribeiro de Azevedo, Cleyton Rodrigues, et. al.** s.l. : Euro American Association on Telematics and Information Systems (EATIS), 2010.

13. **Marut Buranarach, Pornpimon Palingoon.** Development of an Ontology-based Health Information System using Integrated Ontology Management Framework. *National Electronics and Computer Technology Center: NECTEC*. [Online] [Cited: November 2, 2012.] http://text.hlt.nectec.or.th/marut/papers/SWWS2007_marut.pdf.
14. *A method to map heterogeneity between near but non-equivalent semantic attributes in multiple health data registries.* **Nadine Schuurman, Agnieszka Leszczynski.** 1, s.l. : SAGE Publications, 2008, Vol. 14.
15. *An Archetype Pattern Driven Approach to Ontology Development in Health Care Information Systems.* **Wullianallur Raghupathi, Amjad Umar.** s.l. : Decision Sciences Institute, 2008.
16. *Ontology-Based Integration of Information - A Survey of Existing Approaches.* **Wache, H., et al.** Seattle, USA : s.n., 2001.
17. **Nadine Cullot, Raji Ghawi, and Kokou Yétongnon.** *DB2OWL: A Tool for Automatic Database-to-Ontology Mapping.* [PDF Document] Dijon, FRANCE : Université de Bourgogne, 2010.
18. **Jesús Barrasa, Óscar Corcho, Asunción Gómez-Pérez.** *R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language.* [PDF Document] s.l. : Universidad Politécnica de Madrid, Spain, 2004.
19. **Laclavik, Michal.** *RDB2Onto: Relational Database Data to Ontology Individuals Mapping.* [PDF Document] Bratislava, Slovakia : Slovak Academy of Sciences, 2008.
20. *Midwives Manual on Maternal Care.* San Lazaro, Manil : Maternal and Child Health Service - Department of Health.
21. information system -- Britannica Online Encyclopedia. *Britannica Online Encyclopedia.* [Online] Britannica. [Cited: January 21, 2013.] <http://global.britannica.com/EBchecked/topic/287895/information-system>.
22. OpenClinical: Ontologies. *OpenClinical.* [Online] [Cited: August 14, 2012.] <http://www.openclinical.org/ontologies.html>.
23. **Gruber, Tom.** What is an ontology? *Stanford Knowledge Systems, AI Laboratory.* [Online] [Cited: August 12, 2012.] <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
24. **Natalya F. Noy, Deborah L. McGuinness.** What is an ontology and why we need it. *The Protege Ontology Editor and Knowledge Acquisition System.* [Online] [Cited: August 13, 2012.] http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

25. **Gruber, Tom.** Ontology (Computer Science) - definition in Encyclopedia of Database Systems. *Tom Gruber.* [Online] 2007. [Cited: August 14, 2012.] <http://tomgruber.org/writing/ontology-definition-2007.htm>.
26. *Towards semantic interoperability in healthcare: ontology mapping from SNOMED-CT to HL7 version 3.* **Ryan, Amanda.** Darlinghurst, Australia : Australian Computer Society, Inc., 2006.
27. Ontologies - W3C. *World Wide Web Consortium (W3C).* [Online] [Cited: October 12, 2012.] <http://www.w3.org/standards/semanticweb/ontology>.
28. **Deborah L. McGuinness, Frank van Harmelen.** OWL Web Ontology Language Overview. *World Wide Web Consortium (W3C).* [Online] February 10, 2004. [Cited: August 14, 2012.] <http://www.w3.org/TR/2004/REC-owl-features-20040210/>.
29. **Michael K. Smith, Chris Welty, Deborah L. McGuinness.** OWL Web Ontology Language Guide. *World Wide Web Consortium (W3C).* [Online] February 10, 2004. [Cited: August 14, 2012.] <http://www.w3.org/TR/owl-guide/>.
30. **Graham Klyne, Jeremy J. Carroll.** Resource Description Framework (RDF): Concepts and Abstract Syntax. *World Wide Web Consortium (W3C).* [Online] November 8, 2002. [Cited: August 15, 2012.] <http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/>.
31. **Frank Manola, Eric Miller.** RDF Primer. *World Wide Web Consortium (W3C).* [Online] February 10, 2004. [Cited: January 13, 2013.] <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
32. Apache Jena - SPARQL Tutorial. *Apache Jena.* [Online] Apache. [Cited: January 20, 2013.] <http://jena.apache.org/tutorials/sparql.html>.
33. The D2RQ Platform - Accessing Relational Databases as Virtual RDF Graphs. *The D2RQ Platform.* [Online] [Cited: January 12, 2013.] <http://d2rq.org/>.
34. *JavaServer Faces.org.* [Online] [Cited: September 23, 2012.] <http://www.javaserverfaces.org/>.
35. What is a JavaServer Faces Application? - The Java EE 5 Tutorial. *Oracle Documentation.* [Online] September 2010. [Cited: September 23, 2012.] <http://docs.oracle.com/javaee/5/tutorial/doc/bnapk.html>.

XI. Appendix

REFERRAL

PROBLEMS IDENTIFIED AND ACTION TAKEN BY THE MIDWIFE (INDICATED DATE)

(FINDINGS, ACTIONS TAKEN, AND INSTRUCTIONS FROM REFERRAL CENTER)

PANEL 2

POSTPARTUM CARE

Y = YES N = NO

REFER TO HOSPITAL
 REFER TO PHYSICIAN/RHU

TIMING OF POST-PARTUM VISIT	HOME VISIT			CLINIC VISIT
	24 HOURS	1 WEEK	2-4 WEEKS	
DATE OF VISIT				
EXCLUSIVE BREASTFEEDING (Y/N)				
INTENDS TO USE FAMILY PLANNING (Y/N)				
FEVER 38°C AND ABOVE (Y/N)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FOUL SMELLING VAGINAL DISCHARGE (Y/N)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EXCESSIVE VAGINAL BLEEDING (Y/N)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PALLOR (Y/N)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CORD OK? (Y/N)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

VITAMIN A 200,000 IU Y/N:

IRON/FOLATE DATE/#:

BREASTFEEDING IS BEST FOR YOUR BABY. SPACING BIRTHS MORE THAN 24 MONTHS APART WILL ENSURE YOUR HEALTH AND THE HEALTH OF YOUR CHILDREN.

FAMILY PLANNING

DATE OF FOLLOWUP	DATE OF VISIT	METHOD	QUANTITY GIVEN	REMARKS

PANEL 3

HOME BASED MOTHER'S RECORD

ALWAYS BRING THIS CARD WHEN YOU VISIT A HEALTH FACILITY

BLOOD TYPE: _____ FAMILY SERIAL NO. _____

NAME: _____

ADDRESS: _____

DATE TETANUS TOXOID GIVEN: 1. _____ 2. _____ 3. _____

AGE: _____ yr. below 16 16-34

HEIGHT: _____ cm. below 145 cm 145 cm - 160 cm

OBSTETRICAL HISTORY

NUMBER OF PREVIOUS PREGNANCIES: 0 1 2 3 4 5 6 7 8 9 10

PREVIOUS CAESAREAN SECTION	NO	YES
3 CONSECUTIVE MISCARRIAGES	NO	YES
STILLBIRTH	NO	YES
POST-PARTUM HEMORRHAGE	NO	YES

PRESENT HEALTH PROBLEMS

TUBERCULOSIS (14 DAYS + OF COUGH)	NO	YES
HEART DISEASE	NO	YES
DIABETES	NO	YES
BRONCHIAL ASTHMA	NO	YES
GOUT	NO	YES

= REFER TO PHYSICIAN/RHU (AND FOLLOW UP)
 = CLOSE OBSERVATION OR ACTION BY MIDWIFE
 (H) = HOSPITAL DELIVERY RECOMMENDED

Appendix 1: Home Based Mother's Record – Front

PRESENT PREGNANCY

LMP: month _____ day _____ year EDC: month _____ day _____ year

 = IMMEDIATE REFERRAL TO HOSPITAL
 = REFERRAL TO PHYSICIAN

TRIMESTER	1ST			2ND				3RD	
	2 OR 3	4	5	6	7	8	9		
AGE IN MONTHS									
DATE OF THIS VISIT									
VAGINAL BLEEDING (Y/N)									
GENITAL TRACT INFECTION (Y/N)									
WEIGHT IN KG									
BLOOD PRESSURE									
BP > 140/90 AND ABOVE (Y/N)									
FEVER 38°C AND ABOVE (Y/N)									
PALPITATION (Y/N)									
ABNORMAL FUNDAL HEIGHT (Y/N)									
ABNORMAL PRESENTATION (Y/N)									
MISSING FETAL HEARTBEAT (Y/N)									
DEMA (Y/N)									
VAGINAL INFECTION (Y/N)									
LAB TEST RESULTS									
Hb, HGB, URINE, VDRL									

ACTION:

IRON/FOLATE #/RX								
DIET SUPPLEMENTATION IN HIGH RISK AREAS (Y/N)								
MALARIA PROPHYLAXIS (Y/N)								
MOTHER INTENDS TO BREASTFEED? (Y/N)								
ADVISE ON 4 DANGER SIGNS (Y/N)								
FENTAL CHECK UP? (Y/N)								
EMERGENCY PLANS AND PLACE								
DELIVERY (Y/N)								
SMK? (Y/N)								
DATE OF NEXT VISIT								

LABOR & DELIVERY:

IMMEDIATE BREASTFEEDING (Y/N)		BIRTH WEIGHT IN GRAMS	
TYPE OF DELIVERY		POST-PARTUM HEMORRHAGE 500 CC + (Y/N)	
TYPE OF DELIVERY		BABY ALIVE (Y/N)	
TYPE OF DELIVERY		BABY HEALTHY (Y/N)	
ATTENDANT (MD, NURSE, MIDWIFE, UNTRAINED HILOT, TRAINED HILOT, OTHERS)			



SIGNS OF DANGER DURING PREGNANCY:

- TYPE OF VAGINAL BLEEDING
- Nausea, DIZZINESS, BLURRED VISION
- SWELLING OF THE FACE AND HANDS
- WEIGHT GAIN OR ANEMIC
- Notes or No in appropriate spaces

MALARIA PROPHYLAXIS
 CHLOROQUINE 150 mg base
 per tablet, 2 tablets per week

IRON FOLATE
 60 mg tablet, 2 tablets daily
 for 125 days

Appendix 2: Home Based Mother's Record – Back

		FHS (FHSIS) REPORT for the MONTH _____, YEAR: _____ Name of BHS: _____ City/Municipality of: _____ Project Population of the Year: _____			FHS (FHSIS) Monthly Form 			
MATERNAL CARE		NO.	CHILD CARE		NO.	DISEASE CONTROL		NO.
Pregnant with 3 or more Prenatal visits (w/ at least 1 visit per trimester)			Fully immunized Children (9-11 mos.)			LEPROSY: New cases diagnosed (dx)		
Pregnant given TT2 plus			Infant given 3rd dose of Hepa B			Completed treatment (tx)		
Pregnant given Complete Iron dosage			Infants seen at 4th month			Continuing tx		
Postpartum (PP) w/ at least 1 PP visit			Infants exclusively breastfed up to 4th month			MALARIA: Confirmed		
PP mothers given Complete Iron dosage			Diarrhea cases given ORS (0-59 mos.)			Clinically dx		
PP women initiated breastfeeding			Pneumonia cases seen (0-59 mos.)			Given tx		
Breastfeeding mothers given Vitamin A			Pneumonia cases given treatment (0-59 mos)			TB: Symptomatics with sputum exam		
Women 15-49 given iodized oil capsules			Children (9-11) mos.) given Vit. A capsules			New sputum(+) initiated tx		
			Children (12-59 mos.) given Vit A capsules			Old sputum(+) being re-treated		
			Moderately underweight children (6-59 mos.)			X-ray(+) initiated tx		
			- Given food supplementation (New)			SCHISTOSOMIASIS: Examined		
			- Receiving food supplementation			Positive		
			- Rehabilitated			Given tx		
A. Condom			Severely underweight children (6-59 mos.)			RABIES: Animal bites seen		
B. Injection			- Given food supplementation (New)			Given post exposure immunization		
C. IUD			- Receiving food supplementation			FILARIASIS: Cases		
D. LAM			- Rehabilitated			Given tx		
E. NFP			Number of infants given:			STD: w/Vaginal Discharge		
F. Pills			DPT1	BCG		w/Urethral Discharge (Male)		
G. Male Ster.			DPT2	Measles		w/Genital Ulcers (Male/Female)		
H. Fem. Ster.			DPT3	HepB1				
			OPV1	HepB2				
			OPV2	HepB3				
			OPV3					
Prepared By: _____		Date: _____				Received By: _____		Date: _____

Appendix 3: FHSIS Monthly Report (M1)

Source Code

Source Package

AccountDetailsBean.java

```
package com.pregss.account.management;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.user.UserWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class AccountDetailsBean {
    private User user;
    private String[] title_list = PregssConstants.JOB_TITLES;

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public String[] getTitle_list() {
        return title_list;
    }

    public void setTitle_list(String[] title_list) {
        this.title_list = title_list;
    }

    public AccountDetailsBean(){
        UserBean userBean = HttpUtil.findBean("userBean");
        this.user = userBean.getUser();
    }

    public String update(){
        HttpServletRequest request = HttpUtil.getRequest();
        int ret = UserWorker.updateUser(this.user);
        if(ret == 1){

            request.setAttribute(PregssConstants.SUCCESS_MESSAGE, "Your account was updated successfully.");
            return "";
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE, "Update failed.");
            return "";
        }
    }
}
```

RegisterHealthCenterBean.java

```
package com.pregss.account.management;

import java.util.LinkedHashMap;
import javax.servlet.http.HttpServletRequest;
```

```
import com.pregss.healthcenter.HealthCenter;
import com.pregss.healthcenter.HealthCenterWorker;
import com.pregss.location.LocationWorker;
import com.pregss.user.User;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class RegisterHealthCenterBean {
    private HealthCenter healthCenter = new HealthCenter();
    private User user = new User();
    private String[] title_list = PregssConstants.JOB_TITLES;

    private int regionId =
PregssConstants.DEFAULT_REGION;
    private int provinceId =
PregssConstants.DEFAULT_PROVINCE;
    private LinkedHashMap<String,Integer> regionMap =
LocationWorker.getRegionMap();
    private LinkedHashMap<String,Integer> provinceMap =
LocationWorker.getProvinceMap(this.regionId);
    private LinkedHashMap<String,Integer> cityMap =
LocationWorker.getCityMap(this.provinceId);

    public int getRegionId() {
        return regionId;
    }

    public void setRegionId(int regionId) {
        this.regionId = regionId;
    }

    public int getProvinceId() {
        return provinceId;
    }

    public void setProvinceId(int provinceId) {
        this.provinceId = provinceId;
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public HealthCenter getHealthCenter() {
        return healthCenter;
    }

    public void setHealthCenter(HealthCenter healthCenter) {
        this.healthCenter = healthCenter;
    }

    public String[] getTitle_list() {
        return title_list;
    }

    public void setTitle_list(String[] title_list) {
        this.title_list = title_list;
    }
}
```



```

}

public LinkedHashMap<String, Integer> getRegionMap() {
    return this.regionMap;
}

public LinkedHashMap<String, Integer> getProvinceMap()
{
    return this.provinceMap;
}

public LinkedHashMap<String, Integer> getCityMap() {
    return this.cityMap;
}

public void changeRegion(){
    this.provinceMap =
LocationWorker.getProvinceMap(this.regionId);
    this.provinceId = -1;
    changeProvince();
}

public void changeProvince(){
    this.cityMap =
LocationWorker.getCityMap(this.provinceId);
}

public String register(){
    HttpServletRequest request = HttpUtil.getRequest();

    int ret =
HealthCenterWorker.addHealthCenterAndAdmin(healthCe
nter, user);
    if(ret > 0){
        return PregssConstants.REGISTER_SUCCESS;
    }else if(ret == -3){

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Health Center Name already exists. Please try a different
one.");
    }else if(ret == -2){

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Username already exists. Please try a different one.");
    }else{

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Unable to register.");
    }
    return "";
}
}

```

Action.java

```

package com.pregss.action;

import java.util.Date;

public class Action {
    private int checkup_id = -1;
    private int iron_folate_rx;
    private String iodine_high_risk = "N";
    private String malaria_prophylaxis = "N";

```

```

private String will_breastfeed = "Y";
private String danger_sign_advice = "Y";
private String dental_checkup = "N";
private String emergency_plan = "Y";
private String risk = "N";
private Date next_visit_date;

public int getCheckup_id() {
    return checkup_id;
}

public void setCheckup_id(int checkup_id) {
    this.checkup_id = checkup_id;
}

public int getIron_folate_rx() {
    return iron_folate_rx;
}

public void setIron_folate_rx(int iron_folate_rx) {
    this.iron_folate_rx = iron_folate_rx;
}

public String getIodine_high_risk() {
    return iodine_high_risk;
}

public void setIodine_high_risk(String iodine_high_risk) {
    this.iodine_high_risk = iodine_high_risk;
}

public String getMalaria_prophylaxis() {
    return malaria_prophylaxis;
}

public void setMalaria_prophylaxis(String
malaria_prophylaxis) {
    this.malaria_prophylaxis = malaria_prophylaxis;
}

public String getWill_breastfeed() {
    return will_breastfeed;
}

public void setWill_breastfeed(String will_breastfeed) {
    this.will_breastfeed = will_breastfeed;
}

public String getDanger_sign_advice() {
    return danger_sign_advice;
}

public void setDanger_sign_advice(String
danger_sign_advice) {
    this.danger_sign_advice = danger_sign_advice;
}

public String getDental_checkup() {
    return dental_checkup;
}

public void setDental_checkup(String dental_checkup) {
    this.dental_checkup = dental_checkup;
}

public String getEmergency_plan() {
    return emergency_plan;
}

```

```

}

public void setEmergency_plan(String emergency_plan) {
    this.emergency_plan = emergency_plan;
}

public String getRisk() {
    return risk;
}

public void setRisk(String risk) {
    this.risk = risk;
}

public Date getNext_visit_date() {
    return next_visit_date;
}

public void setNext_visit_date(Date next_visit_date) {
    this.next_visit_date = next_visit_date;
}
}

```

ActionBean.java

```

package com.pregss.action;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.evaluator.Evaluator;
import com.pregss.pregnancy.checkup.PregnancyCheckup;
import com.pregss.pregnancy.checkup.PregnancyCheckupWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class ActionBean {
    private PregnancyCheckup checkup = new
    PregnancyCheckup();
    private Action action = new Action();
    private int checkupId = -1;
    private int patientId = -1;
    private int bp1;
    private int bp2;

    private List<String> dangerList;
    private int dangerLevel = 0;

    public ActionBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
                Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("cid")!=null){
            try{

```

```

                this.checkupId =
                Integer.parseInt(request.getParameter("cid"));
                this.checkup =
                PregnancyCheckupWorker.findCheckup(this.checkupId);

                HashMap<String,Integer> dangerMap =
                Evaluator.evaluateCheckup(checkup);
                if(dangerMap != null){
                    this.dangerList = new
                    ArrayList<String>(dangerMap.keySet());
                    this.dangerLevel =
                    Evaluator.getMaxLevel(dangerMap);
                }

                this.action =
                ActionWorker.findAction(this.checkupId);
                if(this.action == null){
                    this.action = new Action();
                }
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public PregnancyCheckup getCheckup() {
        return checkup;
    }

    public void setCheckup(PregnancyCheckup checkup) {
        this.checkup = checkup;
    }

    public Action getAction() {
        return action;
    }

    public void setAction(Action action) {
        this.action = action;
    }

    public int getCheckupId() {
        return checkupId;
    }

    public void setCheckupId(int checkupId) {
        this.checkupId = checkupId;
    }

    public int getDangerLevel() {
        return dangerLevel;
    }

    public void setDangerLevel(int dangerLevel) {
        this.dangerLevel = dangerLevel;
    }

    public int getBp1() {
        String blood_pressure =
        this.checkup.getBlood_pressure();
        int indexOfSlash = blood_pressure.indexOf("/");
        String systolic =
        blood_pressure.substring(0,indexOfSlash);
        try{
            int sysInt = Integer.parseInt(systolic);
            return sysInt;
        }catch(NumberFormatException nfe){

```

```

        nfe.printStackTrace();
    }
    return 0;
}

public void setBp1(int bp1) {
    this.bp1 = bp1;
}

public int getBp2() {
    String blood_pressure =
this.checkup.getBlood_pressure();
    int indexOfSlash = blood_pressure.indexOf("/");
    String diastolic =
blood_pressure.substring(indexOfSlash+1,blood_pressure.leng
th());
    try{
        int diaInt = Integer.parseInt(diastolic);
        return diaInt;
    }catch(NumberFormatException nfe){
        nfe.printStackTrace();
    }
    return 0;
}

public void setBp2(int bp2) {
    this.bp2 = bp2;
}

public int getPatientId() {
    return patientId;
}

public void setPatientId(int patientId) {
    this.patientId = patientId;
}

public List<String> getDangerList() {
    return dangerList;
}

public void setDangerList(List<String> dangerList) {
    this.dangerList = dangerList;
}

public String addAction(){
    int ret = ActionWorker.addAction(this.action,
checkupId);
    if(ret>0){
        try {

            HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#present_pregnancy");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            HttpUtil.getRequest().setAttribute(PregssConstants.ERROR
R_MESSAGE, "Failed adding pregnancy checkup.");
        }
        return "";
    }
}

public String update(){
    HttpServletRequest request = HttpUtil.getRequest();
    this.checkup.setBlood_pressure(this.bp1 + "/" + this.bp2);

```

```

    int ret1 =
PregnancyCheckupWorker.updateCheckup(this.checkup);
    int ret2 = ActionWorker.updateAction(this.action);

    if((ret1 > 0) && (ret2 > 0)){

        request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
    }else{

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
    }
    return "";
}

public String delete(){
    int ret =
PregnancyCheckupWorker.deleteCheckup(this.checkup.getC
heckup_id());
    if(ret>0){
        try {

            HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#present_pregnancy");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            HttpUtil.getRequest().setAttribute(PregssConstants.ERROR
R_MESSAGE, "Failed deleting pregnancy checkup.");
        }
        return "";
    }
}

```

ActionWorker.java

```

package com.pregss.action;

import java.sql.Connection;
import java.sql.SQLException;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class ActionWorker {
    public static int addAction(Action action, int checkup_id){
        String addAction = "INSERT into action " +

            "(checkup_id,iron_folate_rx,iodine_high_risk,malaria_prop
hylaxis,will_breastfeed," +

            "danger_sign_advice,dental_checkup,emergency_plan,risk,
next_visit_date) " +
            "VALUES (?,?,?,?,?,?,?,?)";

        Object[] params = {
            checkup_id,
            action.getIron_folate_rx(),
            action.getIodine_high_risk(),
            action.getMalaria_prophylaxis(),

```

```

        action.getWill_breastfeed(),
        action.getDanger_sign_advice(),
        action.getDental_checkup(),
        action.getEmergency_plan(),
        action.getRisk(),

    PregssUtil.utilDateToSQLDate(action.getNext_visit_date()
    )
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, addAction, params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static Action findAction(int checkupId){
    String findActionSQL = "SELECT * FROM action
    WHERE checkup_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<Action> rsh = new
    BeanHandler<Action>(Action.class);
            Action action = run.query(conn, findActionSQL, rsh,
    checkupId);

            return action;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

public static int updateAction(Action action){
    String updateActionSQL = "UPDATE action SET " +
        "iron_folate_rx = ?, iodine_high_risk = ?,
    malaria_prophylaxis = ?, will_breastfeed = ?, " +
        "danger_sign_advice = ?, dental_checkup = ?,
    emergency_plan = ?, " +
        "risk = ?, next_visit_date = ? " +
        "WHERE checkup_id = ?";

    Object[] params = {

```

```

        action.getIron_folate_rx(),
        action.getIodine_high_risk(),
        action.getMalaria_prophylaxis(),
        action.getWill_breastfeed(),
        action.getDanger_sign_advice(),
        action.getDental_checkup(),
        action.getEmergency_plan(),
        action.getRisk(),
        action.getNext_visit_date(),
        action.getCheckup_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updateActionSQL,
    params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

HealthCenterRequestsBean.java

```

package com.pregss.administrator;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.healthcenter.HealthCenter;
import com.pregss.healthcenter.HealthCenterWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PagingHelper;
import com.pregss.util.PregssConstants;

public class HealthCenterRequestsBean {
    private static final int MAX_RESULTS = 20;
    private PagingHelper paging = new
    PagingHelper(HealthCenterWorker.getTotalHealthCenterRe
    quests(),MAX_RESULTS);

    private List<HealthCenter> healthCenters = new
    ArrayList<HealthCenter>();

    public void initialize(){
        HttpServletRequest request = HttpUtil.getRequest();
        if(request.getParameter("page") != null){
            try{
                int page =
                Integer.parseInt(request.getParameter("page"));
                this.paging.setPage(page);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }
}

```

```

    }
    this.healthCenters =
HealthCenterWorker.getHealthCenterRequests((this.paging.
getPage()-1)*paging.getNumRows(), paging.getNumRows());
}

public PagingHelper getPaging() {
    return paging;
}

public void setPaging(PagingHelper paging) {
    this.paging = paging;
}

public List<HealthCenter> getHealthCenters() {
    return healthCenters;
}

public void setHealthCenters(List<HealthCenter>
healthCenters) {
    this.healthCenters = healthCenters;
}

public String approve(){
    HttpServletRequest request = HttpUtil.getRequest();

    if(request.getParameter("hcid") != null){
        try{
            int healthCenterId =
Integer.parseInt(request.getParameter("hcid"));
            int ret =
HealthCenterWorker.approveHealthCenter(healthCenterId);
            if(ret > 0){

                request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Barangay Health Center approved.");
                }else{

                request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed activating Barangay Health Center.");
                }
                }catch(NumberFormatException nfe){
                nfe.printStackTrace();
                }
            }

            return
PregssConstants.HEALTH_CENTER_REQUESTS;
        }

        public String reject(){
            HttpServletRequest request = HttpUtil.getRequest();

            if(request.getParameter("hcid") != null){
                try{
                    int healthCenterId =
Integer.parseInt(request.getParameter("hcid"));
                    int ret =
HealthCenterWorker.deleteHealthCenter(healthCenterId);
                    if(ret > 0){

                        request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Barangay Health Center request rejected and
deleted.");
                        }else{

```

```

                request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed rejecting Barangay Health Center request.");
                }
                }catch(NumberFormatException nfe){
                nfe.printStackTrace();
                }
            }

            return
PregssConstants.HEALTH_CENTER_REQUESTS;
        }
    }
}

```

ManageHealthCentersBean.java

```

package com.pregss.administrator;

import java.util.ArrayList;
import java.util.LinkedHashMap;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.healthcenter.HealthCenter;
import com.pregss.healthcenter.HealthCenterWorker;
import com.pregss.location.LocationWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PagingHelper;
import com.pregss.util.PregssConstants;

public class ManageHealthCentersBean {
    private static final int MAX_RESULTS = 20;
    private PagingHelper paging = new PagingHelper();

    private int regionId =
PregssConstants.DEFAULT_REGION;
    private int provinceId =
PregssConstants.DEFAULT_PROVINCE;
    private int cityId = -1;

    private LinkedHashMap<String,Integer> regionMap =
LocationWorker.getRegionMap();
    private LinkedHashMap<String,Integer> provinceMap =
LocationWorker.getProvinceMap(this.regionId);
    private LinkedHashMap<String,Integer> cityMap =
LocationWorker.getCityMap(this.provinceId);

    private String filter;
    private List<HealthCenter> healthCenters = new
ArrayList<HealthCenter>();

    public void initialize(){
        HttpServletRequest request = HttpUtil.getRequest();
        if(request.getParameter("page") != null){
            try{
                int page =
Integer.parseInt(request.getParameter("page"));
                this.paging.setPage(page);
                this.healthCenters =
HealthCenterWorker.getHealthCentersInCity(this.cityId,
this.filter, (this.paging.getPage()-1)*paging.getNumRows(),
paging.getNumRows());
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }
}

```

```

public PagingHelper getPaging() {
    return paging;
}

public void setPaging(PagingHelper paging) {
    this.paging = paging;
}

public int getRegionId() {
    return regionId;
}

public void setRegionId(int regionId) {
    this.regionId = regionId;
}

public int getProvinceId() {
    return provinceId;
}

public void setProvinceId(int provinceId) {
    this.provinceId = provinceId;
}

public int getCityId() {
    return cityId;
}

public void setCityId(int cityId) {
    this.cityId = cityId;
}

public LinkedHashMap<String, Integer> getRegionMap() {
    return regionMap;
}

public LinkedHashMap<String, Integer> getProvinceMap()
{
    return provinceMap;
}

public LinkedHashMap<String, Integer> getCityMap() {
    return cityMap;
}

public List<HealthCenter> getHealthCenters() {
    return healthCenters;
}

public void setHealthCenters(List<HealthCenter>
healthCenters) {
    this.healthCenters = healthCenters;
}

public String getFilter() {
    return filter;
}

public void setFilter(String filter) {
    this.filter = filter;
}

public void changeRegion(){
    this.provinceMap =
LocationWorker.getProvinceMap(this.regionId);
    this.provinceId = -1;

```

```

    changeProvince();
}

public void changeProvince(){
    this.cityMap =
LocationWorker.getCityMap(this.provinceId);
    this.cityId = -1;
    changeCity();
}

public void changeCity(){
    int totalHC =
HealthCenterWorker.getTotalHealthCentersInCity(this.cityI
d, this.filter);
    this.paging = new
PagingHelper(totalHC,MAX_RESULTS);
    this.healthCenters =
HealthCenterWorker.getHealthCentersInCity(this.cityId,
this.filter, 0, paging.getNumRows());
}

public String goToPage(){
    return "";
}

public String filterBarangay(){
    //refresh list
    changeCity();
    return "";
}
}

```

ManageUsersBean.java

```
package com.pregss.administrator;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.user.UserWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PagingHelper;
import com.pregss.util.PregssConstants;
```

```
public class ManageUsersBean {
    private static final int MAX_RESULTS = 20;
    private PagingHelper paging = new PagingHelper();
```

```
    private int healthCenterId = -1;
    private List<User> users = new ArrayList<User>();
```

```
    private String filter;
```

```
    public ManageUsersBean(){
        UserBean userBean = HttpUtil.findBean("userBean");
        this.healthCenterId =
userBean.getUser().getHealth_center_id();
        paging = new
PagingHelper(UserWorker.getTotalUsersUnderHC(this.heal
thCenterId),MAX_RESULTS);
    }

```

```
    public void initialize(){
        HttpServletRequest request = HttpUtil.getRequest();

```

```

        if(request.getParameter("page") != null){
            try{
                int page =
Integer.parseInt(request.getParameter("page"));
                this.paging.setPage(page);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
        this.users =
UserWorker.getUsersUnderHC(this.healthCenterId,
(this.paging.getPage()-1)*paging.getNumRows(),
paging.getNumRows());
    }

    public PagingHelper getPaging() {
        return paging;
    }

    public void setPaging(PagingHelper paging) {
        this.paging = paging;
    }

    public int getHealthCenterId() {
        return healthCenterId;
    }

    public void setHealthCenterId(int healthCenterId) {
        this.healthCenterId = healthCenterId;
    }

    public String getFilter() {
        return filter;
    }

    public void setFilter(String filter) {
        this.filter = filter;
    }

    public List<User> getUsers() {
        return users;
    }

    public void setUsers(List<User> users) {
        this.users = users;
    }

    public String goToNewUser(){
        return PregssConstants.USER_DETAILS;
    }
}

```

PregssAdminBean.java

```

package com.pregss.administrator;

import com.pregss.healthcenter.HealthCenterWorker;

public class PregssAdminBean {
    private int num_requests =
HealthCenterWorker.getTotalHealthCenterRequests();

    public int getNum_requests() {
        return num_requests;
    }

    public void setNum_requests(int num_requests) {
        this.num_requests = num_requests;
    }
}

```

```

    }
}

```

UserDetailsBean.java

```

package com.pregss.administrator;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.user.UserWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class UserDetailsBean {
    private User user = new User();
    private String[] title_list = PregssConstants.JOB_TITLES;

    public UserDetailsBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("uid")!=null){
            try{
                int userId =
Integer.parseInt(request.getParameter("uid"));
                this.user = UserWorker.findUser(userId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public String[] getTitle_list() {
        return title_list;
    }

    public void setTitle_list(String[] title_list) {
        this.title_list = title_list;
    }

    public String update(){
        HttpServletRequest request = HttpUtil.getRequest();
        int ret = UserWorker.updateUser(this.user);
        if(ret == 1){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
        }

        return "";
    }

    public String delete(){
        HttpServletRequest request = HttpUtil.getRequest();
        int ret = UserWorker.deleteUser(this.user.getUser_id());
    }
}

```

```

        if(ret == 1){

            request.setAttribute(PregssConstants.SUCCESS_MESSAGE, "User successfully deleted.");
            return PregssConstants.MANAGE_USERS;
        }else{

            request.setAttribute(PregssConstants.SUCCESS_MESSAGE, "Failed to delete user.");
        }
        return "";
    }

    public String add(){
        HttpServletRequest request = HttpUtil.getRequest();
        UserBean userBean = HttpUtil.findBean("userBean");
        int healthCenterId =
        userBean.getUser().getHealth_center_id();

        this.user.setUser_type("RU");
        this.user.setActive_flag("Y");
        int ret = UserWorker.addUser(this.user,
        healthCenterId);
        if(ret == 1){

            request.setAttribute(PregssConstants.SUCCESS_MESSAGE, "User successfully created.");
            return PregssConstants.MANAGE_USERS;
        }else if(ret == -2){

            request.setAttribute(PregssConstants.ERROR_MESSAGE, "Username already exists. Please try a different one.");
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE, "Unable to create user.");
        }
        return "";
    }
}

```

LoginBean.java

```

package com.pregss.authentication;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.user.UserWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class LoginBean {
    private String username;
    private String password;

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }
}

```

```

    public void setPassword(String password) {
        this.password = password;
    }

    public String login(){
        HttpServletRequest request = HttpUtil.getRequest();
        User user = UserWorker.findUser(this.username,
        this.password);
        if(user!=null){
            if("Y".equals(user.getActive_flag())){
                UserBean userBean =
                HttpUtil.findBean("userBean");
                userBean.setUser(user);

                if("HCA".equals(user.getUser_type()) ||
                "RU".equals(user.getUser_type())){
                    return PregssConstants.PATIENT_LIST;
                }else if("PA".equals(user.getUser_type())){
                    return
                    PregssConstants.MANAGE_HEALTH_CENTERS;
                }else{
                    return PregssConstants.HOME;
                }
            }else{
                request.setAttribute("errorMsg", "Your account has
                not been approved yet.");
                return null;
            }
        }else{
            request.setAttribute("errorMsg", "Wrong
            username/password. Please try again.");
            return null;
        }
    }
}

```

LogoutBean.java

```

package com.pregss.authentication;

import javax.servlet.http.HttpSession;

import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class LogoutBean{

    public String logout(){
        HttpSession session = HttpUtil.getSession(false);

        if(session != null){
            session.invalidate();
        }

        return PregssConstants.HOME;
    }
}

```

ConnectionManager.java

```

package com.pregss.db;

import java.sql.Connection;
import java.sql.DriverManager;

```



```

import java.sql.SQLException;

import com.pregss.util.PropertiesUtil;
/**
 * This class provides a jdbc connection.
 */
public class ConnectionManager {
    public static String url =
        PropertiesUtil.getProperty("jdbc.url");
    public static String user =
        PropertiesUtil.getProperty("jdbc.user");
    public static String pass =
        PropertiesUtil.getProperty("jdbc.pass");

    /** this method returns a connection for the specified url,
    user, and
    password.
    */
    public static Connection getConnection() throws
    SQLException {
        try {
            Class.forName("com.mysql.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return DriverManager.getConnection(url, user, pass);
    }
}

```

DangerLevel.java

```
package com.pregss.evaluator;
```

```

public class DangerLevel {
    public static final int HIGH = 3;
    public static final int MEDIUM = 2;
    public static final int LOW = 1;
}

```

Evaluator.java

```
package com.pregss.evaluator;
```

```

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map.Entry;

```

```

import com.pregss.health.history.HealthProblems;
import com.pregss.obstetrical.ObstetricalHistory;
import com.pregss.patient.Patient;
import com.pregss.postpartum.PostpartumCare;
import com.pregss.pregnancy.checkup.PregnancyCheckup;

```

```

public class Evaluator {
    public static HashMap<String, Integer>
    evaluateCheckup(PregnancyCheckup checkup){
        HashMap<String,Integer> dangerList = new
        HashMap<String,Integer>();

        if("Y".equals(checkup.getVaginal_bleeding())){
            dangerList.put("Vaginal Bleeding",
            DangerLevel.HIGH);
        }
        if("Y".equals(checkup.getHighBP())){
            dangerList.put("High Blood Pressure",
            DangerLevel.HIGH);
        }
        if("Y".equals(checkup.getUti())){

```

```

            dangerList.put("Urinary Tract Infection",
            DangerLevel.MEDIUM);
        }
        if("Y".equals(checkup.getFever())){
            dangerList.put("Fever", DangerLevel.LOW);
        }
        if("Y".equals(checkup.getPallor())){
            dangerList.put("Pallor", DangerLevel.LOW);
        }
        if("Y".equals(checkup.getVaginal_infection())){
            dangerList.put("Vaginal Infection",
            DangerLevel.LOW);
        }
        if((checkup.getAog() >= 5 && checkup.getAog() <= 9) &&
        "Y".equals(checkup.getAbn_fundal_height())){
            dangerList.put("Abnormal Fundal Height",
            DangerLevel.MEDIUM);
        }
        if((checkup.getAog() >= 8 && checkup.getAog() <= 9) &&
        "Y".equals(checkup.getAbn_presentation())){
            dangerList.put("Abnormal Presentation",
            DangerLevel.MEDIUM);
        }
        if((checkup.getAog() >= 6 && checkup.getAog() <= 9) &&
        "Y".equals(checkup.getAbn_presentation())){
            dangerList.put("Missing Fetal Heartbeat",
            DangerLevel.MEDIUM);
        }
        if((checkup.getAog() >= 2 && checkup.getAog() <= 8) &&
        "Y".equals(checkup.getEdema())){
            dangerList.put("Edema", DangerLevel.LOW);
        }
    }

    return dangerList;
}

    public static HashMap<String, Integer>
    evaluatePatientHistory(Patient patient, ObstetricalHistory
    obsHistory, HealthProblems healthProbs){
        HashMap<String,Integer> dangerList = new
        HashMap<String,Integer>();

        //Patient
        if(patient != null){
            if(patient.getAge() < 18){
                dangerList.put("The patient is less than 18 years of
                age.", DangerLevel.LOW);
            }else if(patient.getAge() >= 35){
                dangerList.put("The patient is of 35 years of age or
                greater.", DangerLevel.LOW);
            }
        }

        if(patient.getHeight() < 145){
            dangerList.put("The patient's height is below 145
            cm.", DangerLevel.LOW);
        }
    }

    //Obstetrical History
    if(obsHistory != null){
        if(obsHistory.getPrev_pregnancies() >= 4){
            dangerList.put("The patient already had 4 or greater
            pregnancies.", DangerLevel.HIGH);
        }
        if("Y".equals(obsHistory.getPrev_caesarean())){
            dangerList.put("The patient had previously
            undergone Caesarean section.", DangerLevel.HIGH);

```

```

    }
    if("Y".equals(obsHistory.getCons_miscarriages())){
        dangerList.put("The patient had 3 consecutive
miscarriages.", DangerLevel.MEDIUM);
    }
    if("Y".equals(obsHistory.getCons_miscarriages())){
        dangerList.put("The patient had stillbirth.",
DangerLevel.MEDIUM);
    }
    if("Y".equals(obsHistory.getCons_miscarriages())){
        dangerList.put("The patient had post-partum
hemorrhage.", DangerLevel.HIGH);
    }
}

//Health Problems
if(healthProbs != null){
    if("Y".equals(healthProbs.getTuberculosis())){
        dangerList.put("The patient has tuberculosis.",
DangerLevel.LOW);
    }
    if("Y".equals(healthProbs.getHeart_disease())){
        dangerList.put("The patient has heart disease.",
DangerLevel.HIGH);
    }
    if("Y".equals(healthProbs.getDiabetes())){
        dangerList.put("The patient has tuberculosis.",
DangerLevel.HIGH);
    }
    if("Y".equals(healthProbs.getBronchial_asthma())){
        dangerList.put("The patient has bronchial asthma.",
DangerLevel.HIGH);
    }
    if("Y".equals(healthProbs.getGoiter())){
        dangerList.put("The patient has goiter.",
DangerLevel.MEDIUM);
    }
}

return dangerList;
}

public static HashMap<String, Integer>
evaluatePpCare(PostpartumCare ppCare){
    HashMap<String,Integer> dangerList = new
HashMap<String,Integer>();

    if("Y".equals(ppCare.getFever())){
        dangerList.put("Fever", DangerLevel.HIGH);
    }
    if("Y".equals(ppCare.getVaginal_discharge())){
        dangerList.put("Foul-smelling Vaginal Discharge",
DangerLevel.MEDIUM);
    }
    if("Y".equals(ppCare.getVaginal_bleeding())){
        dangerList.put("Excessive Vaginal Bleeding",
DangerLevel.HIGH);
    }
    if("Y".equals(ppCare.getPallor())){
        dangerList.put("Pallor", DangerLevel.MEDIUM);
    }
}

return dangerList;
}

public static int getMaxLevel(HashMap<String, Integer>
dangerMap){

```

```

    int maxLevel = 0;

    Iterator it = dangerMap.entrySet().iterator();
    while (it.hasNext()) {
        Entry<String, Integer> entry = (Entry<String,
Integer>) it.next();
        if(entry.getValue() > maxLevel){
            maxLevel = entry.getValue();
        }
        it.remove(); // avoids a
ConcurrentModificationException
    }

    return maxLevel;
}
}

```

HealthProblems.java

```

package com.pregss.health.history;

public class HealthProblems {
    private int patient_id;
    private String tuberculosis = "N";
    private String heart_disease = "N";
    private String diabetes = "N";
    private String bronchial_asthma = "N";
    private String goiter = "N";

    public int getPatient_id() {
        return patient_id;
    }

    public void setPatient_id(int patient_id) {
        this.patient_id = patient_id;
    }

    public String getTuberculosis() {
        return tuberculosis;
    }

    public void setTuberculosis(String tuberculosis) {
        this.tuberculosis = tuberculosis;
    }

    public String getHeart_disease() {
        return heart_disease;
    }

    public void setHeart_disease(String heart_disease) {
        this.heart_disease = heart_disease;
    }

    public String getDiabetes() {
        return diabetes;
    }

    public void setDiabetes(String diabetes) {
        this.diabetes = diabetes;
    }

    public String getBronchial_asthma() {
        return bronchial_asthma;
    }

    public void setBronchial_asthma(String bronchial_asthma)
    {
        this.bronchial_asthma = bronchial_asthma;
    }
}

```

```

    }

    public String getGoiter() {
        return goiter;
    }

    public void setGoiter(String goiter) {
        this.goiter = goiter;
    }
}

HealthProblemsBean.java
package com.pregss.health.history;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class HealthProblemsBean {
    private HealthProblems healthProbs;
    private int patientId = -1;

    public HealthProblemsBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
Integer.parseInt(request.getParameter("pid"));
                this.healthProbs =
HealthProblemsWorker.findHealthProblems(patientId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public HealthProblems getHealthProbs() {
        return healthProbs;
    }

    public void setHealthProbs(HealthProblems healthProbs) {
        this.healthProbs = healthProbs;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public String update(){
        HttpServletRequest request = HttpUtil.getRequest();
        int ret =
HealthProblemsWorker.updateHealthProblems(this.healthP
robs);
        if(ret > 0){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
        }else{

```

```

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
        }
        return "";
    }
}

```

HealthProblemsWorker.java

```

package com.pregss.health.history;

import java.sql.Connection;
import java.sql.SQLException;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;

import com.pregss.db.ConnectionManager;

public class HealthProblemsWorker {
    public static HealthProblems findHealthProblems(int
patient_id){
        String findHealthProbsSQL = "SELECT * FROM
health_problems WHERE patient_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<HealthProblems> rsh = new
BeanHandler<HealthProblems>(HealthProblems.class);
                HealthProblems healthProbs = run.query(conn,
findHealthProbsSQL, rsh, patient_id);

                return healthProbs;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }

        return null;
    }

    public static int addHealthProblems(HealthProblems
healthProbs, int patient_id){
        String addHealthProbsSQL = "INSERT into
health_problems " +

        "(patient_id,tuberculosis,heart_disease,diabetes,bronchial_
asthma,goiter) " +
        "VALUES (?,?,?,?,?)";
        Object[] params = {
            patient_id,
            healthProbs.getTuberculosis(),
            healthProbs.getHeart_disease(),
            healthProbs.getDiabetes(),
            healthProbs.getBronchial_asthma(),
            healthProbs.getGoiter()
        };
    }
}

```

```

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, addHealthProbsSQL,
params);

        return ret;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}

public static int updateHealthProblems(HealthProblems
healthProbs){
    String updateHealthProbsSQL = "UPDATE
health_problems SET " +
        "tuberculosis = ?, heart_disease = ?, diabetes = ?, " +
        "bronchial_asthma = ?, goiter = ? " +
        "WHERE patient_id = ? ";

    Object[] params = {
        healthProbs.getTuberculosis(),
        healthProbs.getHeart_disease(),
        healthProbs.getDiabetes(),
        healthProbs.getBronchial_asthma(),
        healthProbs.getGoiter(),
        healthProbs.getPatient_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updateHealthProbsSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

HealthCenter.java

```

package com.pregss.healthcenter;

import java.sql.Timestamp;

public class HealthCenter {
    private int health_center_id;

```

```

    private String health_center_name;
    private String contact_no;
    private String active_flag;
    private Timestamp reg_date;
    private String street_address;
    private String barangay;
    private int city_id;

    //implied, not a field in table
    private String city_name;
    private String province_name;
    private String region_name;

    public int getHealth_center_id() {
        return health_center_id;
    }

    public void setHealth_center_id(int health_center_id) {
        this.health_center_id = health_center_id;
    }

    public String getHealth_center_name() {
        return health_center_name;
    }

    public void setHealth_center_name(String
health_center_name) {
        this.health_center_name = health_center_name;
    }

    public String getContact_no() {
        return contact_no;
    }

    public void setContact_no(String contact_no) {
        this.contact_no = contact_no;
    }

    public String getActive_flag() {
        return active_flag;
    }

    public void setActive_flag(String active_flag) {
        this.active_flag = active_flag;
    }

    public Timestamp getReg_date() {
        return reg_date;
    }

    public void setReg_date(Timestamp reg_date) {
        this.reg_date = reg_date;
    }

    public String getStreet_address() {
        return street_address;
    }

    public void setStreet_address(String street_address) {
        this.street_address = street_address;
    }

    public String getBarangay() {
        return barangay;
    }

    public void setBarangay(String barangay) {

```

```

    this.barangay = barangay;
}

public int getCity_id() {
    return city_id;
}

public void setCity_id(int city_id) {
    this.city_id = city_id;
}

public String getCity_name() {
    return city_name;
}

public void setCity_name(String city_name) {
    this.city_name = city_name;
}

public String getProvince_name() {
    return province_name;
}

public void setProvince_name(String province_name) {
    this.province_name = province_name;
}

public String getRegion_name() {
    return region_name;
}

public void setRegion_name(String region_name) {
    this.region_name = region_name;
}
}

```

HealthCenterBean.java

```

package com.pregss.healthcenter;

import java.util.LinkedHashMap;

import javax.servlet.http.HttpServletRequest;

import com.pregss.location.LocationWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class HealthCenterBean {
    private HealthCenter healthCenter = new HealthCenter();

    private int provinceId;
    private int regionId;
    private LinkedHashMap<String,Integer> regionMap =
LocationWorker.getRegionMap();
    private LinkedHashMap<String,Integer> provinceMap;
    private LinkedHashMap<String,Integer> cityMap;

    public HealthCenterBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("hcid")!=null){
            try{
                int hcId =
Integer.parseInt(request.getParameter("hcid"));
                this.healthCenter =
HealthCenterWorker.findHealthCenter(hcId);

```

```

        this.provinceId =
LocationWorker.getProvinceId(this.healthCenter.getCity_id(
));
        this.regionId =
LocationWorker.getRegionId(this.provinceId);

        this.provinceMap =
LocationWorker.getProvinceMap(this.regionId);
        this.cityMap =
LocationWorker.getCityMap(this.provinceId);
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }
}

public HealthCenter getHealthCenter() {
    return healthCenter;
}

public void setHealthCenter(HealthCenter healthCenter) {
    this.healthCenter = healthCenter;
}

public int getProvinceId() {
    return provinceId;
}

public void setProvinceId(int provinceId) {
    this.provinceId = provinceId;
}

public int getRegionId() {
    return regionId;
}

public void setRegionId(int regionId) {
    this.regionId = regionId;
}

public LinkedHashMap<String, Integer> getRegionMap() {
    return this.regionMap;
}

public LinkedHashMap<String, Integer> getProvinceMap()
{
    return this.provinceMap;
}

public LinkedHashMap<String, Integer> getCityMap() {
    return this.cityMap;
}

public void changeRegion(){
    this.provinceMap =
LocationWorker.getProvinceMap(this.regionId);
    this.provinceId = -1;
    changeProvince();
}

public void changeProvince(){
    this.cityMap =
LocationWorker.getCityMap(this.provinceId);
}

public String update(){
    HttpServletRequest request = HttpUtil.getRequest();

```

```

        int ret =
HealthCenterWorker.updateHealthCenter(this.healthCenter
);
        if(ret > 0){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Barangay Health Center details updated.");
            }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed updating Barangay Health Center details.");
            }
            return "";
        }

        public String delete(){
            HttpServletRequest request = HttpUtil.getRequest();
            int ret =
HealthCenterWorker.deleteHealthCenter(this.healthCenter.
getHealth_center_id());
            if(ret > 0){

                request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Barangay Health Center deleted.");
                return
PregssConstants.MANAGE_HEALTH_CENTERS;
            }else{

                request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed deleting Barangay Health Center.");
                }
                return "";
            }
        }

```

HealthCenterWorker.java

```

package com.pregss.healthcenter;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.LinkedHashMap;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;
import org.apache.commons.lang3.StringUtils;

import com.pregss.db.ConnectionManager;
import com.pregss.user.User;
import com.pregss.user.UserWorker;
import com.pregss.util.PregssUtil;

public class HealthCenterWorker {

    public static HealthCenter findHealthCenter(int hcId){
        String findHCSQL = "SELECT * FROM
barangay_health_center where health_center_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();

```

```

            ResultSetHandler<HealthCenter> rsh = new
BeanHandler<HealthCenter>(HealthCenter.class);
            HealthCenter healthCenter = run.query(conn,
findHCSQL, rsh, hcId);

```

```

            return healthCenter;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
        }catch(SQLException e){
            return null;
        }
    }

```

```

        return null;
    }

```

```

    public static HealthCenter findHealthCenter(String
name){
        String findHCSQL = "SELECT * FROM
barangay_health_center where health_center_name = ?";

```

```

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<HealthCenter> rsh = new
BeanHandler<HealthCenter>(HealthCenter.class);
                HealthCenter healthCenter = run.query(conn,
findHCSQL, rsh, name);

```

```

                return healthCenter;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }
    }

```

```

        return null;
    }

```

```

    public static int addHealthCenterAndAdmin(HealthCenter
healthCenter, User user){
        String addBarangaySQL = "INSERT into
barangay_health_center " +
            "(health_center_name,street_address,barangay," +
            "city_id,contact_no,active_flag) " +
            "VALUES (?,?,?,?,'N)";
        String generatedKeySQL = "SELECT
LAST_INSERT_ID()";

```

```

        Object[] params = {
            healthCenter.getHealth_center_name(),
            healthCenter.getStreet_address(),
            healthCenter.getBarangay(),
            healthCenter.getCity_id(),
            healthCenter.getContact_no()
        };

```

```

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();

```

```

try{
    conn = ConnectionManager.getConnection();

    if(null !=
findHealthCenter(healthCenter.getHealth_center_name())){
        return -3;
    }else if (null !=
UserWorker.findUser(user.getUsername())){
        return -2;
    }
    run.update(conn, addBarangaySQL, params);

    int generatedKey = run.query(conn,
generatedKeySQL, PregssUtil.firstIntRsh);

    user.setUser_type("HCA");
    user.setActive_flag("N");
    int ret = UserWorker.addUser(user, generatedKey);

    return ret;
}catch(SQLException e){
    e.printStackTrace();
}finally{
    DbUtils.close(conn);
}
}catch(SQLException e){
    return -1;
}

return -1;
}

public static List<HealthCenter>
getHealthCentersInCity(int cityId, String filter, int offset,
int numRows){
    String getHealthCentersSQL = "SELECT * FROM
barangay_health_center " +
        "WHERE city_id = ? AND active_flag='Y' ";
    if(StringUtils.isEmpty(filter)){
        getHealthCentersSQL += "AND barangay LIKE '%"
+ filter + "%' ";
    }
    getHealthCentersSQL += " ORDER BY barangay,
health_center_name " +
        "LIMIT " + offset + "," + numRows;

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<HealthCenter>> rsh = new
BeanListHandler<HealthCenter>(HealthCenter.class);
            List<HealthCenter> healthCenters =
run.query(conn, getHealthCentersSQL, rsh, cityId);

            return healthCenters;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }
}

return null;

```

```

}

public static int getTotalHealthCentersInCity(int cityId,
String filter){
    String getTotalHealthCentersSQL = "SELECT count(*)
FROM barangay_health_center " +
        "WHERE city_id = ? AND active_flag='Y' ";
    if(StringUtils.isEmpty(filter)){
        getTotalHealthCentersSQL += "AND barangay LIKE
 '%" + filter + "%' ";
    }
}

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();

        int total = run.query(conn,
getTotalHealthCentersSQL, PregssUtil.firstIntRsh, cityId);

        return total;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return 0;
}

return 0;
}

public static int updateHealthCenter(HealthCenter
healthCenter){
    String updateHCSQL = "UPDATE
barangay_health_center SET " +
        "health_center_name = ?, street_address = ?,
barangay = ?, " +
        "city_id = ?, contact_no = ? " +
        "WHERE health_center_id = ? ";

    Object[] params = {
        healthCenter.getHealth_center_name(),
        healthCenter.getStreet_address(),
        healthCenter.getBarangay(),
        healthCenter.getCity_id(),
        healthCenter.getContact_no(),
        healthCenter.getHealth_center_id()
    };
}

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, updateHCSQL, params);

        return ret;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

```

```

    }

    return -1;
}

public static int deleteHealthCenter(int health_center_id){
    if(health_center_id == 0){ //prevent from deleting Health
Center #0 reserved for Pregss Admins
        return -1;
    }

    String deleteHCSQL = "DELETE from
barangay_health_center WHERE health_center_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteHCSQL,
health_center_id);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}

public static List<HealthCenter>
getHealthCenterRequests(int offset, int numRows){
    String getHealthCentersSQL =
        "SELECT bhc.health_center_id,
bhc.health_center_name, " +
        "bhc.street_address, bhc.barangay, bhc.reg_date, " +
        "c.name as city_name, " +
        "p.name as province_name, " +
        "r.name as region_name " +
        "FROM " +
        "(SELECT * FROM barangay_health_center where
active_flag='N') bhc " +
        "LEFT JOIN city c " +
        "ON bhc.city_id = c.city_id " +
        "LEFT JOIN province p " +
        "ON c.province_id = p.province_id " +
        "LEFT JOIN region r " +
        "ON p.region_id = r.region_id " +
        "ORDER BY bhc.reg_date " +
        "LIMIT " + offset + " " + numRows;

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<HealthCenter>> rsh = new
BeanListHandler<HealthCenter>(HealthCenter.class);
            List<HealthCenter> healthCenters =
run.query(conn, getHealthCentersSQL, rsh);

            return healthCenters;

```

```

        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }
}catch(SQLException e){
    return null;
}

return null;
}

public static int getTotalHealthCenterRequests(){
    String getTotalHealthCentersSQL = "SELECT count(*)
FROM barangay_health_center " +
        "WHERE active_flag='N' ";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();

            int total = run.query(conn,
getTotalHealthCentersSQL, PregssUtil.firstIntRsh);

            return total;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return 0;
    }
}

return 0;
}

public static int approveHealthCenter(int
health_center_id){
    String approveHCSQL = "UPDATE
barangay_health_center SET " +
        "active_flag = 'Y' " +
        "WHERE health_center_id = ? ";
    String activateMembersSQL = "UPDATE user SET " +
        "active_flag = 'Y' " +
        "WHERE health_center_id = ? ";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret1 = run.update(conn, approveHCSQL,
health_center_id);
            run.update(conn, activateMembersSQL,
health_center_id);

            return ret1;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;

```



```

    }

    return -1;
}

public static LinkedHashMap<String,Integer>
getHealthCenterMap(int cityId){
    LinkedHashMap<String,Integer> healthCenterMap =
new LinkedHashMap<String,Integer>();

    String getCentersSQL = "SELECT * FROM
barangay_health_center WHERE city_id = ? AND
health_center_id != 0";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<HealthCenter>> rsh = new
BeanListHandler<HealthCenter>(HealthCenter.class);
            List<HealthCenter> centers = run.query(conn,
getCentersSQL, rsh, cityId);

            for(HealthCenter center: centers){

                healthCenterMap.put(center.getHealth_center_name(),cen
ter.getHealth_center_id());
            }
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        healthCenterMap = null;
    }
}

return healthCenterMap;
}
}

```

LaborAndDelivery.java

```

package com.pregss.labor;

import java.util.Date;

public class LaborAndDelivery {
    private int preg_record_id = -1;
    private String imm_breastfeed = "Y";
    private String delivery_type;
    private Date delivery_date;
    private String delivery_place;
    private int birth_weight;
    private String pp_hemorrhage = "N";
    private String baby_alive = "Y";
    private String baby_healthy = "Y";
    private String attendant;

    public int getPreg_record_id() {
        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }
}

```

```

public String getImm_breastfeed() {
    return imm_breastfeed;
}

public void setImm_breastfeed(String imm_breastfeed) {
    this.imm_breastfeed = imm_breastfeed;
}

public String getDelivery_type() {
    return delivery_type;
}

public void setDelivery_type(String delivery_type) {
    this.delivery_type = delivery_type;
}

public Date getDelivery_date() {
    return delivery_date;
}

public void setDelivery_date(Date delivery_date) {
    this.delivery_date = delivery_date;
}

public String getDelivery_place() {
    return delivery_place;
}

public void setDelivery_place(String delivery_place) {
    this.delivery_place = delivery_place;
}

public int getBirth_weight() {
    return birth_weight;
}

public void setBirth_weight(int birth_weight) {
    this.birth_weight = birth_weight;
}

public String getPp_hemorrhage() {
    return pp_hemorrhage;
}

public void setPp_hemorrhage(String pp_hemorrhage) {
    this.pp_hemorrhage = pp_hemorrhage;
}

public String getBaby_alive() {
    return baby_alive;
}

public void setBaby_alive(String baby_alive) {
    this.baby_alive = baby_alive;
}

public String getBaby_healthy() {
    return baby_healthy;
}

public void setBaby_healthy(String baby_healthy) {
    this.baby_healthy = baby_healthy;
}

public String getAttendant() {
    return attendant;
}
}

```

```

    public void setAttendant(String attendant) {
        this.attendant = attendant;
    }
}

LaborAndDeliveryBean.java
package com.pregss.labor;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class LaborAndDeliveryBean {
    private int pregRecordId = -1;
    private int patientId = -1;
    private LaborAndDelivery laborDel;
    private String[] deliveryTypes =
PregssConstants.DELIVERY_TYPES;

    public LaborAndDeliveryBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("prid")!=null){
            try{
                this.pregRecordId =
Integer.parseInt(request.getParameter("prid"));
                this.laborDel =
LaborAndDeliveryWorker.findLaborAndDelivery(this.pregR
ecordId);
                if(this.laborDel == null){
                    laborDel = new LaborAndDelivery();
                }
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public String[] getDeliveryTypes() {
        return deliveryTypes;
    }

    public LaborAndDelivery getLaborDel() {
        return laborDel;
    }
}

```

```

    public void setLaborDel(LaborAndDelivery laborDel) {
        this.laborDel = laborDel;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public String addLaborDel(){
        if(this.pregRecordId!=-1){
            int ret =
LaborAndDeliveryWorker.addLaborAndDelivery(this.laborD
el, pregRecordId);
            if(ret>0){
                try {

                    HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#labor_delivery");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }else{

                HttpUtil.getRequest().setAttribute(PregssConstants.ERROR_
MESSAGE, "Failed adding labor and delivery
information.");
            }

            return "";
        }

        public String update(){
            int ret =
LaborAndDeliveryWorker.updateLaborDel(this.laborDel);
            HttpServletRequest request = HttpUtil.getRequest();
            if(ret > 0){

                request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
            }else{

                request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
            }
            return null;
        }
    }

LaborAndDeliveryWorker.java
package com.pregss.labor;

import java.sql.Connection;
import java.sql.SQLException;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

```

```

public class LaborAndDeliveryWorker {
    public static int addLaborAndDelivery(LaborAndDelivery
laborDel, int preg_record_id){
        String addAction = "INSERT into labor_and_delivery " +

"(preg_record_id,imm_breastfeed,delivery_type,delivery_da
te," +

"delivery_place,birth_weight,pp_hemorrhage,baby_alive,ba
by_healthy,attendant) " +
        "VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

        Object[] params = {
            preg_record_id,
            laborDel.getImm_breastfeed(),
            laborDel.getDelivery_type(),

            PregssUtil.utilDateToSQLDate(laborDel.getDelivery_date(
)),
            laborDel.getDelivery_place(),
            laborDel.getBirth_weight(),
            laborDel.getPp_hemorrhage(),
            laborDel.getBaby_alive(),
            laborDel.getBaby_healthy(),
            laborDel.getAttendant()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, addAction, params);

                return ret;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }

        return -1;
    }

    public static LaborAndDelivery findLaborAndDelivery(int
pregRecordId){
        String findLaborDelSQL = "SELECT * FROM
labor_and_delivery WHERE preg_record_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<LaborAndDelivery> rsh = new
BeanHandler<LaborAndDelivery>(LaborAndDelivery.class);
                LaborAndDelivery laborDel = run.query(conn,
findLaborDelSQL, rsh, pregRecordId);

                return laborDel;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{

```

```

                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }

        return null;
    }

    public static int updateLaborDel(LaborAndDelivery
laborDel){
        String updateLaborDelSQL = "UPDATE
labor_and_delivery SET " +
            "imm_breastfeed = ?, delivery_type = ?, delivery_date
= ?, " +
            "delivery_place = ?, birth_weight = ?, pp_hemorrhage
= ?, " +
            "baby_alive = ?, baby_healthy = ?, attendant = ? " +
            "WHERE preg_record_id = ? ";

        Object[] params = {
            laborDel.getImm_breastfeed(),
            laborDel.getDelivery_type(),

            PregssUtil.utilDateToSQLDate(laborDel.getDelivery_date(
)),
            laborDel.getDelivery_place(),
            laborDel.getBirth_weight(),
            laborDel.getPp_hemorrhage(),
            laborDel.getBaby_alive(),
            laborDel.getBaby_healthy(),
            laborDel.getAttendant(),
            laborDel.getPreg_record_id()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, updateLaborDelSQL,
params);

                return ret;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }

        return -1;
    }
}

```

City.java

```

package com.pregss.location;

public class City {
    private int city_id;
    private String name;
    private int province_id;

    public int getCity_id() {
        return city_id;
    }
}

```

```

    }
    public String getName() {
        return name;
    }
    public int getProvince_id() {
        return province_id;
    }
    public void setCity_id(int city_id) {
        this.city_id = city_id;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setProvince_id(int province_id) {
        this.province_id = province_id;
    }
}

```

LocationWorker.java

```
package com.pregss.location;
```

```
import java.sql.Connection;
import java.sql.SQLException;
import java.util.LinkedHashMap;
import java.util.List;
```

```
import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;
```

```
import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;
```

```
public class LocationWorker {
    private static final String GET_ALL_REGIONS =
        "SELECT * from region";

    private static final String
GET_PROVINCES_BY_REGION =
        "SELECT * from province where region_id = ?";

    private static final String GET_CITIES_BY_PROVINCE =
        "SELECT * from city where province_id = ?";

    private static final String GET_PROVINCE_OF_CITY =
        "SELECT province_id from city where city_id = ?";

    private static final String GET_REGION_OF_PROVINCE
=
        "SELECT region_id from province where province_id =
?";

    public static LinkedHashMap<String,Integer>
getRegionMap(){
        LinkedHashMap<String,Integer> regionMap = new
LinkedHashMap<String,Integer>();

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();

```

```

            ResultSetHandler<List<Region>> rsh = new
BeanListHandler<Region>(Region.class);
            List<Region> regions = run.query(conn,
GET_ALL_REGIONS, rsh);

```

```

                for(Region region: regions){

                    regionMap.put(region.getName(),region.getRegion_id());
                }
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            regionMap = null;
        }
    }

```

```

        return regionMap;
    }

```

```

    public static LinkedHashMap<String,Integer>
getProvinceMap(int regionId){
        LinkedHashMap<String,Integer> provinceMap = new
LinkedHashMap<String,Integer>();

```

```

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<Province>> rsh = new
BeanListHandler<Province>(Province.class);
                List<Province> provinces = run.query(conn,
GET_PROVINCES_BY_REGION, rsh, regionId);

```

```

                for(Province province: provinces){

                    provinceMap.put(province.getName(),province.getProvince
_id());
                }
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            provinceMap = null;
        }
    }

    return provinceMap;
}

```

```

    public static LinkedHashMap<String,Integer>
getCityMap(int provinceId){
        LinkedHashMap<String,Integer> cityMap = new
LinkedHashMap<String,Integer>();

```

```

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<City>> rsh = new
BeanListHandler<City>(City.class);
                List<City> cities = run.query(conn,
GET_CITIES_BY_PROVINCE, rsh, provinceId);

```

```

        for(City city: cities){
            cityMap.put(city.getName(),city.getCity_id());
        }
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    cityMap = null;
}
}

return cityMap;
}

public static int getProvinceId(int city_id){
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int provinceId = run.query(conn,
GET_PROVINCE_OF_CITY, PregssUtil.firstIntRsh, city_id);

            return provinceId;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}

public static int getRegionId(int province_id){
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int regionId = run.query(conn,
GET_REGION_OF_PROVINCE, PregssUtil.firstIntRsh,
province_id);

            return regionId;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}

public static City findCity(int cityId){
    String findCitySQL = "SELECT * FROM city WHERE
city_id = ?";

    try{
        Connection conn = null;

```

```

        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<City> rsh = new
BeanHandler<City>(City.class);
            City city = run.query(conn, findCitySQL, rsh, cityId);

            return city;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }
}

return null;
}

public static Province findProvince(int provinceId){
    String findProvinceSQL = "SELECT * FROM province
WHERE province_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<Province> rsh = new
BeanHandler<Province>(Province.class);
            Province province = run.query(conn,
findProvinceSQL, rsh, provinceId);

            return province;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }
}

return null;
}

public static Region findRegion(int regionId){
    String findRegionSQL = "SELECT * FROM region
WHERE region_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<Region> rsh = new
BeanHandler<Region>(Region.class);
            Region region = run.query(conn, findRegionSQL, rsh,
regionId);

            return region;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }
}

```

```

    }catch(SQLException e){
        return null;
    }

    return null;
}
}

```

Province.java

```
package com.pregss.location;
```

```

public class Province {
    private int province_id;
    private String name;
    private int region_id;

    public int getProvince_id() {
        return province_id;
    }
    public String getName() {
        return name;
    }
    public int getRegion_id() {
        return region_id;
    }
    public void setProvince_id(int province_id) {
        this.province_id = province_id;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setRegion_id(int region_id) {
        this.region_id = region_id;
    }
}

```

Region.java

```
package com.pregss.location;
```

```

public class Region {
    private int region_id;
    private String name;

    public int getRegion_id() {
        return region_id;
    }

    public String getName() {
        return name;
    }

    public void setRegion_id(int region_id) {
        this.region_id = region_id;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

ObstetricalHistory.java

```
package com.pregss.obstetrical;
```

```

public class ObstetricalHistory {
    private int patient_id;
    private int prev_pregnancies = 0;

```

```

    private String prev_caesarean = "N";
    private String cons_miscarriages = "N";
    private String stillbirth = "N";
    private String pp_hemorrhage = "N";

```

```

    public int getPatient_id() {
        return patient_id;
    }

```

```

    public void setPatient_id(int patient_id) {
        this.patient_id = patient_id;
    }

```

```

    public int getPrev_pregnancies() {
        return prev_pregnancies;
    }

```

```

    public void setPrev_pregnancies(int prev_pregnancies) {
        this.prev_pregnancies = prev_pregnancies;
    }

```

```

    public String getPrev_caesarean() {
        return prev_caesarean;
    }

```

```

    public void setPrev_caesarean(String prev_caesarean) {
        this.prev_caesarean = prev_caesarean;
    }

```

```

    public String getCons_miscarriages() {
        return cons_miscarriages;
    }

```

```

    public void setCons_miscarriages(String
cons_miscarriages) {
        this.cons_miscarriages = cons_miscarriages;
    }

```

```

    public String getStillbirth() {
        return stillbirth;
    }

```

```

    public void setStillbirth(String stillbirth) {
        this.stillbirth = stillbirth;
    }

```

```

    public String getPp_hemorrhage() {
        return pp_hemorrhage;
    }

```

```

    public void setPp_hemorrhage(String pp_hemorrhage) {
        this.pp_hemorrhage = pp_hemorrhage;
    }

```

```
}
```

ObstetricalHistoryBean.java

```
package com.pregss.obstetrical;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import com.pregss.util.HttpUtil;
```

```
import com.pregss.util.PregssConstants;
```

```

public class ObstetricalHistoryBean {
    private ObstetricalHistory obsHistory;
    private int patientId = -1;

```

```

public ObstetricalHistoryBean(){
    HttpServletRequest request = HttpUtil.getRequest();

    if(request.getParameter("pid")!=null){
        try{
            this.patientId =
Integer.parseInt(request.getParameter("pid"));
            this.obsHistory =
ObstetricalWorker.findObstetricalHistory(patientId);
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }
}

public int getPatientId() {
    return patientId;
}

public void setPatientId(int patientId) {
    this.patientId = patientId;
}

public ObstetricalHistory getObsHistory() {
    return obsHistory;
}

public void setObsHistory(ObstetricalHistory obsHistory) {
    this.obsHistory = obsHistory;
}

public String update(){
    HttpServletRequest request = HttpUtil.getRequest();
    int ret =
ObstetricalWorker.updateObstetricalHistory(this.obsHistory
);
    if(ret > 0){

        request.setAttribute(PregsssConstants.SUCCESS_MESSA
GE, "Update successful.");
    }else{

        request.setAttribute(PregsssConstants.ERROR_MESSAGE,
"Update failed.");
    }
    return null;
}
}

```

ObstetricalWorker.java

```

package com.pregsss.obstetrical;

import java.sql.Connection;
import java.sql.SQLException;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;

import com.pregsss.db.ConnectionManager;

public class ObstetricalWorker {
    public static ObstetricalHistory findObstetricalHistory(int
patient_id){

```

```

String findObsHistorySQL = "SELECT * FROM
obstetrical_history WHERE patient_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<ObstetricalHistory> rsh = new
BeanHandler<ObstetricalHistory>(ObstetricalHistory.class);
            ObstetricalHistory obsHistory = run.query(conn,
findObsHistorySQL, rsh, patient_id);

            return obsHistory;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

    public static int addObstetricalHistory(ObstetricalHistory
obsHistory, int patient_id){
        String addObsHistorySQL = "INSERT into
obstetrical_history " +

        "(patient_id,prev_pregnancies,prev_caesarean,cons_miscar
riages,stillbirth,pp_hemorrhage) " +
        "VALUES (?,?,?,?,?)";
        Object[] params = {
            patient_id,
            obsHistory.getPrev_pregnancies(),
            obsHistory.getPrev_caesarean(),
            obsHistory.getCons_miscarriages(),
            obsHistory.getStillbirth(),
            obsHistory.getPp_hemorrhage()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, addObsHistorySQL,
params);

                return ret;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }

        return -1;
    }

    public static int
updateObstetricalHistory(ObstetricalHistory obsHistory){

```

```

String updateObsHistorySQL = "UPDATE
obstretical_history SET " +
    "prev_pregnancies = ?, prev_caesarean = ?,
cons_miscarriages = ?, " +
    "stillbirth = ?, pp_hemorrhage = ? " +
    "WHERE patient_id = ? ";

Object[] params = {
    obsHistory.getPrev_pregnancies(),
    obsHistory.getPrev_caesarean(),
    obsHistory.getCons_miscarriages(),
    obsHistory.getStillbirth(),
    obsHistory.getPp_hemorrhage(),
    obsHistory.getPatient_id()
};

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, updateObsHistorySQL,
params);

        return ret;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}
}

```

Patient.java

```

package com.pregss.patient;

public class Patient {
    private int patient_id = -1;
    private String first_name;
    private String middle_name;
    private String last_name;
    private String sex;
    private int age;
    private String blood_type;
    private String family_serial_no;
    private int height;
    private String address;
    private String contact_no;
    private String added_by;
    private int health_center_id;

    //not in table
    private String health_center_name;

    public int getPatient_id() {
        return patient_id;
    }

    public void setPatient_id(int patient_id) {
        this.patient_id = patient_id;
    }
}

```

```

public String getFirst_name() {
    return first_name;
}

public void setFirst_name(String first_name) {
    this.first_name = first_name;
}

public String getMiddle_name() {
    return middle_name;
}

public void setMiddle_name(String middle_name) {
    this.middle_name = middle_name;
}

public String getLast_name() {
    return last_name;
}

public void setLast_name(String last_name) {
    this.last_name = last_name;
}

public String getSex() {
    return sex;
}

public void setSex(String sex) {
    this.sex = sex;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

public String getBlood_type() {
    return blood_type;
}

public void setBlood_type(String blood_type) {
    this.blood_type = blood_type;
}

public String getFamily_serial_no() {
    return family_serial_no;
}

public void setFamily_serial_no(String family_serial_no) {
    this.family_serial_no = family_serial_no;
}

public int getHeight() {
    return height;
}

public void setHeight(int height) {
    this.height = height;
}

public String getAddress() {
    return address;
}
}

```



```

public void setAddress(String address) {
    this.address = address;
}

public String getContact_no() {
    return contact_no;
}

public void setContact_no(String contact_no) {
    this.contact_no = contact_no;
}

public String getAdded_by() {
    return added_by;
}

public void setAdded_by(String added_by) {
    this.added_by = added_by;
}

public int getHealth_center_id() {
    return health_center_id;
}

public void setHealth_center_id(int health_center_id) {
    this.health_center_id = health_center_id;
}

public String getHealth_center_name() {
    return health_center_name;
}

public void setHealth_center_name(String
health_center_name) {
    this.health_center_name = health_center_name;
}
}

```

PatientBean.java

```

package com.pregss.patient;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.evaluator.Evaluator;
import com.pregss.health.history.HealthProblems;
import com.pregss.health.history.HealthProblemsWorker;
import com.pregss.obstetrical.ObstetricalHistory;
import com.pregss.obstetrical.ObstetricalWorker;
import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class PatientBean {
    private Patient patient = new Patient();
    private ObstetricalHistory obsHistory = new
ObstetricalHistory();
    private HealthProblems healthProbs = new
HealthProblems();

```

```

private List<String> dangerList;
private int dangerLevel = 0;

public PatientBean(){
    HttpServletRequest request = HttpUtil.getRequest();

    if(request.getParameter("pid")!=null){
        try{
            int patientId =
Integer.parseInt(request.getParameter("pid"));
            this.patient = PatientWorker.findPatient(patientId);
            this.obsHistory =
ObstetricalWorker.findObstetricalHistory(patientId);
            this.healthProbs =
HealthProblemsWorker.findHealthProblems(patientId);

            HashMap<String,Integer> dangerMap =
Evaluator.evaluatePatientHistory(this.patient,
this.obsHistory, this.healthProbs);
            if(dangerMap != null){
                this.dangerList = new
ArrayList<String>(dangerMap.keySet());
                this.dangerLevel =
Evaluator.getMaxLevel(dangerMap);
            }
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    public Patient getPatient() {
        return patient;
    }

    public void setPatient(Patient patient) {
        this.patient = patient;
    }

    public ObstetricalHistory getObsHistory() {
        return obsHistory;
    }

    public void setObsHistory(ObstetricalHistory obsHistory) {
        this.obsHistory = obsHistory;
    }

    public HealthProblems getHealthProbs() {
        return healthProbs;
    }

    public void setHealthProbs(HealthProblems healthProbs) {
        this.healthProbs = healthProbs;
    }

    public List<String> getDangerList() {
        return dangerList;
    }

    public void setDangerList(List<String> dangerList) {
        this.dangerList = dangerList;
    }

    public int getDangerLevel() {
        return dangerLevel;
    }
}

```

```

public void setDangerLevel(int dangerLevel) {
    this.dangerLevel = dangerLevel;
}

public String addPatient(){
    UserBean userBean = HttpUtil.findBean("userBean");
    User user = userBean.getUser();

    this.patient.setSex("F");
    this.patient.setAdded_by(user.getUsername());

    this.patient.setHealth_center_id(user.getHealth_center_id(
));
    int key = PatientWorker.addPatient(this.patient,
this.obsHistory, this.healthProbs);

    if(key >= 0){
        String url = "view_patient.xhtml?pid=" + key;
        try {
            HttpUtil.redirect(url);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    return "";
}

public String updatePatient(){
    HttpServletRequest request = HttpUtil.getRequest();
    int ret = PatientWorker.updatePatient(this.patient);
    if(ret == 1){
        if(ret > 0){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
        }
    }
    return "";
}

public String deletePatient(){
    int ret =
PatientWorker.deletePatient(this.patient.getPatient_id());
    HttpServletRequest request = HttpUtil.getRequest();
    if(ret > 0){

        request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Patient deleted.");
        return PregssConstants.PATIENT_LIST;
    }else{

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed deleting patient.");
    }
    return "";
}
}

```

PatientListBean.java
package com.pregss.patient;

```

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.UserBean;
import com.pregss.util.HttpUtil;
import com.pregss.util.PagingHelper;

public class PatientListBean {
    private static final int MAX_RESULTS = 20;
    private PagingHelper paging = new PagingHelper();

    private int healthCenterId = -1;
    private List<Patient> patients = new
ArrayList<Patient>();

    public PatientListBean(){
        UserBean userBean = HttpUtil.findBean("userBean");
        this.healthCenterId =
userBean.getUser().getHealth_center_id();
        paging = new
PagingHelper(PatientWorker.getTotalPatientsUnderHC(this
.healthCenterId),MAX_RESULTS);
    }

    public void initialize(){
        HttpServletRequest request = HttpUtil.getRequest();
        if(request.getParameter("page") != null){
            try{
                int page =
Integer.parseInt(request.getParameter("page"));
                this.paging.setPage(page);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
        this.patients =
PatientWorker.getPatientsUnderHC(this.healthCenterId,
(this.paging.getPage()-1)*paging.getNumRows(),
paging.getNumRows());
    }

    public PagingHelper getPaging() {
        return paging;
    }

    public void setPaging(PagingHelper paging) {
        this.paging = paging;
    }

    public List<Patient> getPatients() {
        return patients;
    }

    public void setPatients(List<Patient> patients) {
        this.patients = patients;
    }
}

```

PatientWorker.java
package com.pregss.patient;
import java.sql.Connection;

```

import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.health.history.HealthProblems;
import com.pregss.health.history.HealthProblemsWorker;
import com.pregss.obstetrical.ObstetricalHistory;
import com.pregss.obstetrical.ObstetricalWorker;
import com.pregss.util.PregssUtil;

public class PatientWorker {

    public static int addPatient(Patient patient,
ObstetricalHistory obsHistory, HealthProblems
healthProbs){
        String addPatientSQL = "INSERT into patient
(first_name,middle_name,last_name," +

        "sex,blood_type,family_serial_no,age,height,address,contac
t_no,added_by,health_center_id,date_added) " +
        "VALUES (?,?,?,?,?,?,?,?,?,?,NOW()";

        String generatedKeySQL = "SELECT
LAST_INSERT_ID()";

        Object[] params = {
            patient.getFirst_name(),
            patient.getMiddle_name(),
            patient.getLast_name(),
            patient.getSex(),
            patient.getBlood_type(),
            patient.getFamily_serial_no(),
            patient.getAge(),
            patient.getHeight(),
            patient.getAddress(),
            patient.getContact_no(),
            patient.getAdded_by(),
            patient.getHealth_center_id()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                run.update(conn, addPatientSQL, params);

                int patientId = run.query(conn, generatedKeySQL,
PregssUtil.firstIntRsh);

                ObstetricalWorker.addObstetricalHistory(obsHistory,
patientId);

                HealthProblemsWorker.addHealthProblems(healthProbs,
patientId);

                return patientId;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{

```

```

                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }
    }

    return -1;
}

public static int updatePatient(Patient patient){
    String updatePatientSQL = "UPDATE patient SET " +
        "first_name = ?, middle_name = ?, last_name = ?, " +
        "sex = ?, blood_type = ?, family_serial_no = ?, " +
        "age = ?, height = ?, address = ?, " +
        "contact_no = ? " +
        "WHERE patient_id = ? ";

    Object[] params = {
        patient.getFirst_name(),
        patient.getMiddle_name(),
        patient.getLast_name(),
        patient.getSex(),
        patient.getBlood_type(),
        patient.getFamily_serial_no(),
        patient.getAge(),
        patient.getHeight(),
        patient.getAddress(),
        patient.getContact_no(),
        patient.getPatient_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updatePatientSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static Patient findPatient(int id){
    String findPatientSQL = "SELECT * FROM patient
WHERE patient_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<Patient> rsh = new
BeanHandler<Patient>(Patient.class);
            Patient patient = run.query(conn, findPatientSQL,
rsh, id);

            return patient;

```

```

    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return null;
}

return null;
}

public static int deletePatient(int patient_id){
    String deletePatientSQL = "DELETE from patient
WHERE patient_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deletePatientSQL,
patient_id);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static List<Patient> getPatientsUnderHC(int
health_center_id, int offset, int numRows){
    String getPatientsUnderHCSQL = "SELECT * FROM
patient WHERE health_center_id = ? " +
    "ORDER BY last_name, first_name, middle_name "
+
    "LIMIT " + offset + "," + numRows;
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<Patient>> rsh = new
BeanListHandler<Patient>(Patient.class);
            List<Patient> patients = run.query(conn,
getPatientsUnderHCSQL, rsh, health_center_id);

            return patients;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

```

```

    public static int getTotalPatientsUnderHC(int
health_center_id){
        String getTotalPatientsSQL = "SELECT count(*) FROM
patient " +
            "WHERE health_center_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();

                int total = run.query(conn, getTotalPatientsSQL,
PregssUtil.firstIntRsh, health_center_id);

                return total;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return 0;
        }

        return 0;
    }

    public static List<Patient> searchPatients(String
lastName, String firstName, String middleName, int
health_center_id, int offset, int numRows){
        String searchPatientsSQL = "SELECT patient.*,
bhc.health_center_name FROM patient,
barangay_health_center bhc " +
            "WHERE last_name LIKE ? " +
            "AND first_name LIKE ? " +
            "AND middle_name LIKE ? " +
            "AND patient.health_center_id =
bhc.health_center_id ";
        if(health_center_id != -1){
            searchPatientsSQL += "AND
patient.health_center_id = " + health_center_id;
        }
        searchPatientsSQL += " ORDER BY last_name,
first_name, middle_name " +
            " LIMIT " + offset + "," + numRows;

        Object[] params = {
            "%" + lastName + "%",
            "%" + firstName + "%",
            "%" + middleName + "%",
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<Patient>> rsh = new
BeanListHandler<Patient>(Patient.class);
                List<Patient> patients = run.query(conn,
searchPatientsSQL, rsh, params);

                return patients;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{

```

```

        DbUtils.close(conn);
    }
} catch (SQLException e) {
    return null;
}

return null;
}

public static int countPatientSearchResults(String
lastName, String firstName, String middleName, int
health_center_id) {
    String totalSearchPatientsSQL = "SELECT count(*)
FROM patient " +
        "WHERE last_name LIKE ? " +
        "AND first_name LIKE ? " +
        "AND middle_name LIKE ? ";
    if (health_center_id != -1) {
        totalSearchPatientsSQL += "AND health_center_id = "
+ health_center_id;
    }

    Object[] params = {
        "%" + lastName + "%",
        "%" + firstName + "%",
        "%" + middleName + "%"
    };

    try {
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try {
            conn = ConnectionManager.getConnection();

            int total = run.query(conn, totalSearchPatientsSQL,
PregssUtil.firstIntRsh, params);

            return total;
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DbUtils.close(conn);
        }
    } catch (SQLException e) {
        return 0;
    }

    return 0;
}

public static int transferPatient(Patient patient, int
newHealthCenterId) {
    String updatePatientSQL = "UPDATE patient SET " +
        "health_center_id = ? " +
        "WHERE patient_id = ? ";

    Object[] params = {
        newHealthCenterId,
        patient.getPatient_id()
    };

    try {
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try {
            conn = ConnectionManager.getConnection();

```

```

        int ret = run.update(conn, updatePatientSQL,
params);

        return ret;
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        DbUtils.close(conn);
    }
} catch (SQLException e) {
    return -1;
}

return -1;
}
}

```

SearchPatientBean.java

```

package com.pregss.patient;

import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.UserBean;
import com.pregss.util.HttpUtil;
import com.pregss.util.PagingHelper;
import com.pregss.util.PregssConstants;

public class SearchPatientBean {
    private String lastName = "";
    private String firstName = "";
    private String middleName = "";
    private String scope;

    private static final int MAX_RESULTS = 20;
    private PagingHelper paging = new PagingHelper();

    private int healthCenterId = -1;
    private List<Patient> patients = new
ArrayList<Patient>();

    public SearchPatientBean() {
        UserBean userBean = HttpUtil.findBean("userBean");
        this.healthCenterId =
userBean.getUser().getHealth_center_id();
    }

    public void initialize() {
        HttpServletRequest request = HttpUtil.getRequest();
        if ("local".equals(this.scope)) {
            paging = new
PagingHelper(PatientWorker.countPatientSearchResults(las
tName.trim(), firstName.trim(), middleName.trim(),
healthCenterId), MAX_RESULTS);
        } else {
            paging = new
PagingHelper(PatientWorker.countPatientSearchResults(las
tName.trim(), firstName.trim(), middleName.trim(), -
1), MAX_RESULTS);
        }
        if (request.getParameter("page") != null) {
            try {
                int page =
Integer.parseInt(request.getParameter("page"));
                this.paging.setPage(page);
            }

```

```

        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }
    if("local".equals(this.scope)){
        this.patients =
        PatientWorker.searchPatients(lastName.trim(),
        firstName.trim(), middleName.trim(), healthCenterId,
        (this.paging.getPage()-1)*paging.getNumRows(),
        paging.getNumRows());
    }else{
        this.patients =
        PatientWorker.searchPatients(lastName.trim(),
        firstName.trim(), middleName.trim(), -1,
        (this.paging.getPage()-1)*paging.getNumRows(),
        paging.getNumRows());
    }
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getMiddleName() {
    return middleName;
}

public void setMiddleName(String middleName) {
    this.middleName = middleName;
}

public String getScope() {
    return scope;
}

public void setScope(String scope) {
    this.scope = scope;
}

public PagingHelper getPaging() {
    return paging;
}

public void setPaging(PagingHelper paging) {
    this.paging = paging;
}

public List<Patient> getPatients() {
    return patients;
}

public void setPatients(List<Patient> patients) {
    this.patients = patients;
}

```

```

public String search(){
    return "/" + PregssConstants.SEARCH_PATIENT +
    "?faces-redirect=true&includeViewParams=true";
}
}

```

TransferPatientBean.java

```

package com.pregss.patient;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class TransferPatientBean {
    private Patient patient = new Patient();

    public TransferPatientBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                int patientId =
                Integer.parseInt(request.getParameter("pid"));
                this.patient = PatientWorker.findPatient(patientId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public Patient getPatient() {
        return patient;
    }

    public void setPatient(Patient patient) {
        this.patient = patient;
    }

    public String transfer(){
        UserBean userBean = HttpUtil.findBean("userBean");
        User user = userBean.getUser();

        int ret = PatientWorker.transferPatient(patient,
        user.getHealth_center_id());
        HttpServletRequest request = HttpUtil.getRequest();
        if(ret > 0){
            try {

                HttpUtil.redirect("view_patient.xhtml?pid="+this.patient.g
                etPatient_id());
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
            "Transfer failed.");
        }
        return "";
    }
}

```

FamilyPlanBean.java

```
package com.pregss.planning;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class FamilyPlanBean {
    private FamilyPlanning famPlan = new FamilyPlanning();
    private int pregRecordId;
    private int patientId = -1;

    public FamilyPlanBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
                Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("prid")!=null){
            try{
                this.pregRecordId =
                Integer.parseInt(request.getParameter("prid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("fpid")!=null){
            try{
                int familyPlanId =
                Integer.parseInt(request.getParameter("fpid"));
                this.famPlan =
                FamilyPlanWorker.findFamilyPlanning(familyPlanId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public FamilyPlanning getFamPlan() {
        return famPlan;
    }

    public void setFamPlan(FamilyPlanning famPlan) {
        this.famPlan = famPlan;
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public int getPatientId() {
        return patientId;
    }
}
```

```
public void setPatientId(int patientId) {
    this.patientId = patientId;
}

public String addFamilyPlan(){

    if(this.pregRecordId!=-1){
        int ret = FamilyPlanWorker.addFamilyPlan(famPlan,
        pregRecordId);
        if(ret>0){
            try {

                HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
                d+"#family_planning");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            HttpUtil.getRequest().setAttribute(PregssConstants.ERRO
            R_MESSAGE, "Failed adding a family planning entry.");
        }

        return "";
    }

    public String update(){
        int ret =
        FamilyPlanWorker.updateFamilyPlan(this.famPlan);
        HttpServletRequest request = HttpUtil.getRequest();
        if(ret > 0){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
            GE, "Update successful.");
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
            "Update failed.");
        }
        return "";
    }

    public String delete(){
        int ret =
        FamilyPlanWorker.deleteFamilyPlan(this.famPlan.getFamIl
        y_plan_id());
        HttpServletRequest request = HttpUtil.getRequest();
        if(ret > 0){
            try {

                HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
                d+"#family_planning");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
            "Delete failed.");
        }
        return null;
    }
}
```

FamilyPlanning.java

```
package com.pregss.planning;

import java.util.Date;

public class FamilyPlanning {
    private int family_plan_id = -1;
    private int preg_record_id;
    private Date visit_date;
    private String method;
    private String quantity_given;
    private String remarks;
    private Date follow_up_date;

    public int getFamily_plan_id() {
        return family_plan_id;
    }

    public void setFamily_plan_id(int family_plan_id) {
        this.family_plan_id = family_plan_id;
    }

    public int getPreg_record_id() {
        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }

    public Date getVisit_date() {
        return visit_date;
    }

    public void setVisit_date(Date visit_date) {
        this.visit_date = visit_date;
    }

    public String getMethod() {
        return method;
    }

    public void setMethod(String method) {
        this.method = method;
    }

    public String getQuantity_given() {
        return quantity_given;
    }

    public void setQuantity_given(String quantity_given) {
        this.quantity_given = quantity_given;
    }

    public String getRemarks() {
        return remarks;
    }

    public void setRemarks(String remarks) {
        this.remarks = remarks;
    }

    public Date getFollow_up_date() {
        return follow_up_date;
    }

    public void setFollow_up_date(Date follow_up_date) {
```

```
        this.follow_up_date = follow_up_date;
    }
}
```

FamilyPlanWorker.java

```
package com.pregss.planning;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class FamilyPlanWorker {
    public static FamilyPlanning findFamilyPlanning(int
familyPlanId){
        String findFamilyPlanSQL = "SELECT * FROM
family_planning WHERE family_plan_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<FamilyPlanning> rsh = new
BeanHandler<FamilyPlanning>(FamilyPlanning.class);
                FamilyPlanning famPlan = run.query(conn,
findFamilyPlanSQL, rsh, familyPlanId);

                return famPlan;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }

        return null;
    }

    public static int addFamilyPlan(FamilyPlanning famPlan,
int preg_record_id){
        String addFamPlanSQL = "INSERT into
family_planning " +
            "(preg_record_id,visit_date,method,quantity_given,"
+
            "remarks, follow_up_date) " +
            "VALUES (?,?,?,?,?)";

        Object[] params = {
            preg_record_id,
            PregssUtil.utilDateToSQLDate(famPlan.getVisit_date()),
            famPlan.getMethod(),
            famPlan.getQuantity_given(),
```



```

        famPlan.getRemarks(),
    PregssUtil.utilDateToSQLDate(famPlan.getFollow_up_date()),
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, addFamPlanSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static List<FamilyPlanning> getFamilyPlans(int
pregRecordId){
    String getFamilyPlansSQL = "SELECT * from
family_planning " +
        "WHERE preg_record_id = ? "+
        "ORDER BY visit_date ";
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<FamilyPlanning>> rsh =
new
BeanListHandler<FamilyPlanning>(FamilyPlanning.class);
            List<FamilyPlanning> checkups = run.query(conn,
getFamilyPlansSQL, rsh, pregRecordId);

            return checkups;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

public static int updateFamilyPlan(FamilyPlanning
famPlan){
    String updateFamilyPlanSQL = "UPDATE
family_planning SET " +
        "visit_date = ?, method = ?, quantity_given = ?, " +
        "remarks = ?, follow_up_date = ? " +
        "WHERE family_plan_id = ? ";

    Object[] params = {

```

```

        PregssUtil.utilDateToSQLDate(famPlan.getVisit_date()),
        famPlan.getMethod(),
        famPlan.getQuantity_given(),
        famPlan.getRemarks(),

        PregssUtil.utilDateToSQLDate(famPlan.getFollow_up_date()),
        famPlan.getFamily_plan_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updateFamilyPlanSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static int deleteFamilyPlan(int familyPlanId){
    String deleteFamilyPlanSQL = "DELETE from
family_planning WHERE family_plan_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteFamilyPlanSQL,
familyPlanId);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

PostpartumCare.java

```

package com.pregss.postpartum;

import java.util.Date;

public class PostpartumCare {
    private int pp_care_id = -1;
    private int preg_record_id;
    private Date visit_date;

```

```

private String breastfeed_only = "N";
private String will_family_plan = "N";
private String fever = "N";
private String vaginal_discharge = "N";
private String vaginal_bleeding = "N";
private String pallor = "N";
private String cord_ok = "N";
private String vitamin_a = "N";
private int iron_folate_rx;

public int getPp_care_id() {
    return pp_care_id;
}

public void setPp_care_id(int pp_care_id) {
    this.pp_care_id = pp_care_id;
}

public int getPreg_record_id() {
    return preg_record_id;
}

public void setPreg_record_id(int preg_record_id) {
    this.preg_record_id = preg_record_id;
}

public Date getVisit_date() {
    return visit_date;
}

public void setVisit_date(Date visit_date) {
    this.visit_date = visit_date;
}

public String getBreastfeed_only() {
    return breastfeed_only;
}

public void setBreastfeed_only(String breastfeed_only) {
    this.breastfeed_only = breastfeed_only;
}

public String getWill_family_plan() {
    return will_family_plan;
}

public void setWill_family_plan(String will_family_plan) {
    this.will_family_plan = will_family_plan;
}

public String getFever() {
    return fever;
}

public void setFever(String fever) {
    this.fever = fever;
}

public String getVaginal_discharge() {
    return vaginal_discharge;
}

public void setVaginal_discharge(String vaginal_discharge)
{
    this.vaginal_discharge = vaginal_discharge;
}

```

```

public String getVaginal_bleeding() {
    return vaginal_bleeding;
}

public void setVaginal_bleeding(String vaginal_bleeding) {
    this.vaginal_bleeding = vaginal_bleeding;
}

public String getPallor() {
    return pallor;
}

public void setPallor(String pallor) {
    this.pallor = pallor;
}

public String getCord_ok() {
    return cord_ok;
}

public void setCord_ok(String cord_ok) {
    this.cord_ok = cord_ok;
}

public String getVitamin_a() {
    return vitamin_a;
}

public void setVitamin_a(String vitamin_a) {
    this.vitamin_a = vitamin_a;
}

public int getIron_folate_rx() {
    return iron_folate_rx;
}

public void setIron_folate_rx(int iron_folate_rx) {
    this.iron_folate_rx = iron_folate_rx;
}
}

```

PostpartumCareBean.java

```

package com.pregss.postpartum;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.evaluator.Evaluator;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class PostpartumCareBean {
    private PostpartumCare ppCare = new PostpartumCare();
    private int pregRecordId;
    private int patientId;

    private List<String> dangerList;
    private int dangerLevel = 0;

    public PostpartumCareBean(
        HttpServletRequest request = HttpUtil.getRequest();

```

```

    if(request.getParameter("pid")!=null){
        try{
            this.patientId =
Integer.parseInt(request.getParameter("pid"));
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    if(request.getParameter("prid")!=null){
        try{
            this.pregRecordId =
Integer.parseInt(request.getParameter("prid"));
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    if(request.getParameter("ppid")!=null){
        try{
            int ppCareId =
Integer.parseInt(request.getParameter("ppid"));
            this.ppCare =
PostpartumCareWorker.findPpCare(ppCareId);

            HashMap<String,Integer> dangerMap =
Evaluator.evaluatePpCare(this.ppCare);
            if(dangerMap != null){
                this.dangerList = new
ArrayList<String>(dangerMap.keySet());
                this.dangerLevel =
Evaluator.getMaxLevel(dangerMap);
            }
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    public PostpartumCare getPpCare() {
        return ppCare;
    }

    public void setPpCare(PostpartumCare ppCare) {
        this.ppCare = ppCare;
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public List<String> getDangerList() {
        return dangerList;
    }

    public void setDangerList(List<String> dangerList) {
        this.dangerList = dangerList;
    }

    public int getDangerLevel() {
        return dangerLevel;
    }

```

```

    public void setDangerLevel(int dangerLevel) {
        this.dangerLevel = dangerLevel;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public String addPpCare(){
        if(this.pregRecordId!=-1){
            int ret =
PostpartumCareWorker.addPostpartumCare(ppCare,
pregRecordId);
            if(ret>0){
                try {

                    HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#postpartum_care");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }else{

                HttpUtil.getRequest().setAttribute(PregssConstants.ERRO
R_MESSAGE, "Failed adding a postpartum care entry.");
            }
        }
        return "";
    }

    public String update(){
        int ret =
PostpartumCareWorker.updatePpCare(this.ppCare);
        HttpServletRequest request = HttpUtil.getRequest();
        if(ret > 0){

            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
        }else{

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
        }
        return "";
    }

    public String delete(){
        int ret =
PostpartumCareWorker.deletePpCare(this.ppCare.getPp_car
e_id());
        if(ret>0){
            try {

                HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#postpartum_care");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{

            HttpUtil.getRequest().setAttribute(PregssConstants.ERRO
R_MESSAGE, "Delete failed.");
        }
    }

```

```

        return "";
    }
}

PostpartumCareWorker.java
package com.pregss.postpartum;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class PostpartumCareWorker {
    public static PostpartumCare findPpCare(int ppCareId){
        String findFamilyPlanSQL = "SELECT * FROM
postpartum_care WHERE pp_care_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<PostpartumCare> rsh = new
BeanHandler<PostpartumCare>(PostpartumCare.class);
                PostpartumCare ppCare = run.query(conn,
findFamilyPlanSQL, rsh, ppCareId);

                return ppCare;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }
    }

    return null;
}

    public static int addPostpartumCare(PostpartumCare
ppCare, int preg_record_id){
        String addPpCareSQL = "INSERT into postpartum_care
" +

        "(preg_record_id,visit_date,breastfeed_only,will_family_pla
n," +

        "fever,vaginal_discharge,vaginal_bleeding,pallor,cord_ok,"
+

        "vitamin_a,iron_folate_rx) " +
        "VALUES (?,?,?,?,?,?,?,?,?)";

        Object[] params = {
            preg_record_id,

            PregssUtil.utilDateToSQLDate(ppCare.getVisit_date()),

```

```

ppCare.getBreastfeed_only(),
ppCare.getWill_family_plan(),
ppCare.getFever(),
ppCare.getVaginal_discharge(),
ppCare.getVaginal_bleeding(),
ppCare.getPallor(),
ppCare.getCord_ok(),
ppCare.getVitamin_a(),
ppCare.getIron_folate_rx()
};

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, addPpCareSQL, params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

    return -1;
}

    public static List<PostpartumCare> getPostCares(int
pregRecordId){
        String getPostCaresSQL = "SELECT * from
postpartum_care " +
        "WHERE preg_record_id = ? " +
        "ORDER BY visit_date ";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<PostpartumCare>> rsh =
new
BeanListHandler<PostpartumCare>(PostpartumCare.class);
                List<PostpartumCare> checkups = run.query(conn,
getPostCaresSQL, rsh, pregRecordId);

                return checkups;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }
    }

    return null;
}

    public static int updatePpCare(PostpartumCare ppCare){
        String updatePpCareSQL = "UPDATE postpartum_care
SET " +
        "visit_date = ?, breastfeed_only = ?, will_family_plan
= ?, " +

```

```

        "fever = ?, vaginal_discharge = ?, vaginal_bleeding =
?, " +
        "pallor = ?, cord_ok = ?, vitamin_a = ?, iron_folate_rx
= ? " +
        "WHERE pp_care_id = ? ";

    Object[] params = {

PregssUtil.utilDateToSQLDate(ppCare.getVisit_date()),
        ppCare.getBreastfeed_only(),
        ppCare.getWill_family_plan(),
        ppCare.getFever(),
        ppCare.getVaginal_discharge(),
        ppCare.getVaginal_bleeding(),
        ppCare.getPallor(),
        ppCare.getCord_ok(),
        ppCare.getVitamin_a(),
        ppCare.getIron_folate_rx(),
        ppCare.getPp_care_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updatePpCareSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}

public static int deletePpCare(int ppCareId){
    String deleteFamilyPlanSQL = "DELETE from
postpartum_care WHERE pp_care_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteFamilyPlanSQL,
ppCareId);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}
}

```

PregnancyCheckup.java

```

package com.pregsss.pregnancy.checkup;

import java.util.Date;

import org.apache.commons.lang3.StringUtils;

import com.pregsss.action.Action;

public class PregnancyCheckup {
    private int checkup_id = -1;
    private int preg_record_id;
    private Date visit_date;
    private String vaginal_bleeding = "N";
    private String uti = "N";
    private int weight;
    private String blood_pressure;
    private String fever = "N";
    private String pallor = "N";
    private String abn_fundal_height = "N";
    private String abn_presentation = "N";
    private String missing_heartbeat = "N";
    private String edema = "N";
    private String vaginal_infection = "N";

    //not in table
    private Action action;
    private int aog = 1;

    public int getCheckup_id() {
        return checkup_id;
    }

    public void setCheckup_id(int checkup_id) {
        this.checkup_id = checkup_id;
    }

    public int getPreg_record_id() {
        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }

    public Date getVisit_date() {
        return visit_date;
    }

    public void setVisit_date(Date visit_date) {
        this.visit_date = visit_date;
    }

    public String getVaginal_bleeding() {
        return vaginal_bleeding;
    }

    public void setVaginal_bleeding(String vaginal_bleeding) {
        this.vaginal_bleeding = vaginal_bleeding;
    }

    public String getUti() {
        return uti;
    }

    public void setUti(String uti) {

```

```

    this.uti = uti;
}

public int getWeight() {
    return weight;
}

public void setWeight(int weight) {
    this.weight = weight;
}

public String getBlood_pressure() {
    return blood_pressure;
}

public void setBlood_pressure(String blood_pressure) {
    this.blood_pressure = blood_pressure;
}

public String getFever() {
    return fever;
}

public void setFever(String fever) {
    this.fever = fever;
}

public String getPallor() {
    return pallor;
}

public void setPallor(String pallor) {
    this.pallor = pallor;
}

public String getAbn_fundal_height() {
    return abn_fundal_height;
}

public void setAbn_fundal_height(String
abn_fundal_height) {
    this.abn_fundal_height = abn_fundal_height;
}

public String getAbn_presentation() {
    return abn_presentation;
}

public void setAbn_presentation(String abn_presentation) {
    this.abn_presentation = abn_presentation;
}

public String getMissing_heartbeat() {
    return missing_heartbeat;
}

public void setMissing_heartbeat(String
missing_heartbeat) {
    this.missing_heartbeat = missing_heartbeat;
}

public String getEdema() {
    return edema;
}

public void setEdema(String edema) {
    this.edema = edema;
}

```

```

}

public String getVaginal_infection() {
    return vaginal_infection;
}

public void setVaginal_infection(String vaginal_infection) {
    this.vaginal_infection = vaginal_infection;
}

public Action getAction() {
    return action;
}

public void setAction(Action action) {
    this.action = action;
}

public int getAog() {
    return aog;
}

public void setAog(int aog) {
    this.aog = aog;
}

public String getHighBP(){
    if(StringUtils.isEmpty(this.blood_pressure)){
        int indexOfSlash = this.blood_pressure.indexOf("/");
        String systolic =
this.blood_pressure.substring(0,indexOfSlash);
        String diastolic =
this.blood_pressure.substring(indexOfSlash+1,this.blood_pre
ssure.length());
        try{
            int sysInt = Integer.parseInt(systolic);
            int diaInt = Integer.parseInt(diastolic);
            return (sysInt > 140 || diaInt > 90) ? "Y":"N";
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }
    return "?";
}
}

```

PregnancyCheckupBean.java

```

package com.pregss.pregnancy.checkup;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;

public class PregnancyCheckupBean {
    private PregnancyCheckup pregCheckup = new
PregnancyCheckup();
    private int patientId = -1;
    private int pregRecordId = -1;
    private int bp1;
    private int bp2;

    public PregnancyCheckupBean(){
        HttpServletRequest request = HttpUtil.getRequest();
    }
}

```

```

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("prid")!=null){
            try{
                this.pregRecordId =
Integer.parseInt(request.getParameter("prid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public PregnancyCheckup getPregCheckup() {
        return pregCheckup;
    }

    public void setPregCheckup(PregnancyCheckup
pregCheckup) {
        this.pregCheckup = pregCheckup;
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public int getBp1() {
        return bp1;
    }

    public void setBp1(int bp1) {
        this.bp1 = bp1;
    }

    public int getBp2() {
        return bp2;
    }

    public void setBp2(int bp2) {
        this.bp2 = bp2;
    }

    public String addCheckup(){

        if(this.pregRecordId!=-1){
            this.pregCheckup.setBlood_pressure(this.bp1 + "/" +
this.bp2);

```

```

        int newCheckup =
PregnancyCheckupWorker.addPregnancyCheckup(this.preg
Checkup, pregRecordId);
        if(newCheckup >= 0){
            String url = "action.xhtml?pid="+ this.patientId
+"&cid=" + newCheckup;
            try {
                HttpUtil.redirect(url);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{
            return "fail";
        }
    }

    return "";
}
}

```

PregnancyCheckupWorker.java

```

package com.pregss.pregnancy.checkup;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;

import com.pregss.action.ActionWorker;
import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class PregnancyCheckupWorker {

    public static PregnancyCheckup findCheckup(int
checkupId){
        String findCheckupSQL = "SELECT pc.*, "+
            "PERIOD_DIFF(date_format(pc.visit_date, '%Y%m'),
date_format(pr.lmp, '%Y%m')) as aog "+
            "FROM pregnancy_checkup pc, pregnancy_record pr
"+
            "WHERE pc.preg_record_id = pr.preg_record_id "+
            "AND pc.checkup_id = ? "+
            "ORDER BY visit_date ";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<PregnancyCheckup> rsh = new
BeanHandler<PregnancyCheckup>(PregnancyCheckup.class
);
                PregnancyCheckup checkup = run.query(conn,
findCheckupSQL, rsh, checkupId);

                return checkup;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{

```

```

        DbUtils.close(conn);
    }
} catch (SQLException e) {
    return null;
}

return null;
}

public static List<PregnancyCheckup>
getPregnancyCheckups(int pregRecordId) {
    String getPregnancyCheckupsHCSQL = "SELECT pc.*,
"+
    "PERIOD_DIFF(date_format(pc.visit_date, '%Y%m'),
date_format(pr.lmp, '%Y%m')) as aog "+
    "FROM pregnancy_checkup pc, pregnancy_record pr
"+
    "WHERE pc.preg_record_id = pr.preg_record_id "+
    "AND pc.preg_record_id = ? "+
    "ORDER BY visit_date ";
    try {
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try {
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<PregnancyCheckup>> rsh =
new
BeanListHandler<PregnancyCheckup>(PregnancyCheckup.c
lass);
            List<PregnancyCheckup> checkups =
run.query(conn, getPregnancyCheckupsHCSQL, rsh,
pregRecordId);

            if (checkups != null) {
                for (PregnancyCheckup checkup : checkups) {

                    checkup.setAction(ActionWorker.findAction(checkup.getCh
eckup_id()));
                }
            }
            return checkups;
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DbUtils.close(conn);
        }
    } catch (SQLException e) {
        return null;
    }

    return null;
}

public static int
addPregnancyCheckup(PregnancyCheckup pregCheckup, int
preg_record_id) {
    String addPregCheckupSQL = "INSERT into
pregnancy_checkup "+

    "(preg_record_id,visit_date,vaginal_bleeding,uti,weight,bl
ood_pressure," +

    "fever,pallor,abn_fundal_height,abn_presentation,missing_
heartbeat," +

    "edema,vaginal_infection) "+
    "VALUES (?,?,?,?,?,?,?,?,?,?)";

```

```

Object[] params = {
    preg_record_id,

    PregssUtil.utilDateToSQLDate(pregCheckup.getVisit_date
    )),
    pregCheckup.getVaginal_bleeding(),
    pregCheckup.getUti(),
    pregCheckup.getWeight(),
    pregCheckup.getBlood_pressure(),
    pregCheckup.getFever(),
    pregCheckup.getPallor(),
    pregCheckup.getAbn_fundal_height(),
    pregCheckup.getAbn_presentation(),
    pregCheckup.getMissing_heartbeat(),
    pregCheckup.getEdema(),
    pregCheckup.getVaginal_infection()
};

String generatedKeySQL = "SELECT
LAST_INSERT_ID()";

try {
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try {
        conn = ConnectionManager.getConnection();
        run.update(conn, addPregCheckupSQL, params);

        int checkupId = run.query(conn, generatedKeySQL,
PregssUtil.firstIntRsh);
        return checkupId;
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        DbUtils.close(conn);
    }
} catch (SQLException e) {
    return -1;
}

return -1;
}

public static int updateCheckup(PregnancyCheckup
checkup) {
    String updateCheckupSQL = "UPDATE
pregnancy_checkup SET "+
    "visit_date = ?, vaginal_bleeding = ?, uti = ?, "+
    "weight = ?, blood_pressure = ?, fever = ?, "+
    "pallor = ?, abn_fundal_height = ?, abn_presentation
= ?, "+
    "missing_heartbeat = ?, edema = ?, vaginal_infection
= ? "+
    "WHERE checkup_id = ?";

    Object[] params = {

    PregssUtil.utilDateToSQLDate(checkup.getVisit_date()),
    checkup.getVaginal_bleeding(),
    checkup.getUti(),
    checkup.getWeight(),
    checkup.getBlood_pressure(),
    checkup.getFever(),
    checkup.getPallor(),
    checkup.getAbn_fundal_height(),
    checkup.getAbn_presentation(),
    checkup.getMissing_heartbeat(),

```



```

        checkup.getEdema(),
        checkup.getVaginal_infection(),
        checkup.getCheckup_id()
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updateCheckupSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}

public static int deleteCheckup(int checkupId){
    String deleteCheckupSQL = "DELETE from
pregnancy_checkup WHERE checkup_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteCheckupSQL,
checkupId);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }
}

return -1;
}
}

```

PregnancyRecord.java

```

package com.pregss.pregnancy.record;

import java.sql.Timestamp;
import java.util.Date;

public class PregnancyRecord {
    private int preg_record_id = -1;
    private int patient_id;
    private Date lmp;
    private Date edc;
    private Timestamp date_added;
    private int health_center_id;

    public int getPreg_record_id() {

```

```

        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }

    public int getPatient_id() {
        return patient_id;
    }

    public void setPatient_id(int patient_id) {
        this.patient_id = patient_id;
    }

    public Date getLmp() {
        return lmp;
    }

    public void setLmp(Date lmp) {
        this.lmp = lmp;
    }

    public Date getEdc() {
        return edc;
    }

    public void setEdc(Date edc) {
        this.edc = edc;
    }

    public Timestamp getDate_added() {
        return date_added;
    }

    public void setDate_added(Timestamp date_added) {
        this.date_added = date_added;
    }

    public int getHealth_center_id() {
        return health_center_id;
    }

    public void setHealth_center_id(int health_center_id) {
        this.health_center_id = health_center_id;
    }
}

```

PregnancyRecordBean.java

```

package com.pregss.pregnancy.record;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.patient.Patient;
import com.pregss.patient.PatientWorker;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class PregnancyRecordBean {
    private PregnancyRecord pregRecord = new
PregnancyRecord();
    private int patientId = -1;

    public PregnancyRecordBean(){

```

```

    HttpServletRequest request = HttpUtil.getRequest();

    if(request.getParameter("pid")!=null){
        try{
            this.patientId =
Integer.parseInt(request.getParameter("pid"));
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    if(request.getParameter("prid")!=null){
        try{
            int pregRecordId =
Integer.parseInt(request.getParameter("prid"));
            this.pregRecord =
PregnancyRecordWorker.findRecord(pregRecordId);
        }catch(NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }
}

public PregnancyRecord getPregRecord() {
    return pregRecord;
}

public void setPregRecord(PregnancyRecord pregRecord) {
    this.pregRecord = pregRecord;
}

public int getPatientId() {
    return patientId;
}

public void setPatientId(int patientId) {
    this.patientId = patientId;
}

public String addRecord(){
    Patient patient =
PatientWorker.findPatient(this.patientId);
    HttpServletRequest request = HttpUtil.getRequest();

    if(this.patientId!=-1){
        int ret =
PregnancyRecordWorker.addPregnancyRecord(this.pregRecord, this.patientId,patient.getHealth_center_id());
        if(ret == -1){

            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed adding new pregnancy record.");
        }else{
            try {
                HttpUtil.redirect("view_patient.xhtml?pid=" +
this.patientId + "#present_pregnancy");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    return "";
}

public String update(){

```

```

    int ret =
PregnancyRecordWorker.updateRecord(this.pregRecord);
    HttpServletRequest request = HttpUtil.getRequest();
    if(ret > 0){

        request.setAttribute(PregssConstants.SUCCESS_MESSAGE, "Update successful.");
    }else{

        request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
    }
    return null;
}
}

```

PregnancyRecordWorker.java

```
package com.pregss.pregnancy.record;
```

```
import java.sql.Connection;
import java.sql.SQLException;
```

```
import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
```

```
import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;
```

```
public class PregnancyRecordWorker {
```

```
    public static PregnancyRecord findRecord(int
pregRecordId){
        String findRecordSQL = "SELECT * FROM
pregnancy_record WHERE preg_record_id = ?";
```

```
        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<PregnancyRecord> rsh = new
BeanHandler<PregnancyRecord>(PregnancyRecord.class);
                PregnancyRecord pregRecord = run.query(conn,
findRecordSQL, rsh, pregRecordId);
```

```
                return pregRecord;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }
    }

```

```
    return null;
}

```

```
    public static int getLatestPregnancyRecord(int patient_id){
        String hasRecordsSQL = "SELECT * FROM
pregnancy_record WHERE patient_id = ? ORDER BY
date_added DESC";
```

```
        try{
            Connection conn = null;
```

```

    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int latestRecord = run.query(conn, hasRecordsSQL,
PregssUtil.firstIntRsh, patient_id);

        return latestRecord;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}

public static int addPregnancyRecord(PregnancyRecord
pregRecord, int patient_id, int health_center_id){
    String addPregRecordSQL = "INSERT into
pregnancy_record " +
        "(patient_id,lmp.edc,health_center_id) " +
        "VALUES (?, ?, ?, ?)";

    Object[] params = {
        patient_id,
        PregssUtil.utilDateToSQLDate(pregRecord.getLmp()),
        PregssUtil.utilDateToSQLDate(pregRecord.getEdc()),
        health_center_id
    };

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, addPregRecordSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}

public static int updateRecord(PregnancyRecord
pregRecord){
    String updateRecordSQL = "UPDATE pregnancy_record
SET " +
        "lmp = ?, edc = ? " +
        "WHERE preg_record_id = ? ";

    Object[] params = {
        pregRecord.getLmp(),
        pregRecord.getEdc(),
        pregRecord.getPreg_record_id()
    };

```

```

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, updateRecordSQL,
params);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

PresentPregnancyBean.java

```
package com.pregss.pregnancy.record;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import com.pregss.labor.LaborAndDelivery;
import com.pregss.labor.LaborAndDeliveryWorker;
import com.pregss.planning.FamilyPlanWorker;
import com.pregss.planning.FamilyPlanning;
import com.pregss.postpartum.PostpartumCare;
import com.pregss.postpartum.PostpartumCareWorker;
import com.pregss.pregnancy.checkup.PregnancyCheckup;
import
com.pregss.pregnancy.checkup.PregnancyCheckupWorker;
import com.pregss.referral.Referral;
import com.pregss.referral.ReferralWorker;
import com.pregss.tetanus_toxoid.TetanusToxoid;
import com.pregss.tetanus_toxoid.TetanusToxoidWorker;
import com.pregss.util.HttpUtil;
```

```
public class PresentPregnancyBean {
    private PregnancyRecord pregRecord;
    private List<TetanusToxoid> vaccines;
    private List<PregnancyCheckup> checkups = new
ArrayList<PregnancyCheckup>();
    private LaborAndDelivery laborDel;
    private List<FamilyPlanning> familyPlans;
    private List<PostpartumCare> postCares;
    private List<Referral> referrals;

```

```

    public PresentPregnancyBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                int patientId =
Integer.parseInt(request.getParameter("pid"));
                int latestRecord =
PregnancyRecordWorker.getLatestPregnancyRecord(patient
Id);
                if(latestRecord != -1){

```

```

        this.pregRecord =
PregnancyRecordWorker.findRecord(latestRecord);
        if(this.pregRecord != null){
            this.vaccines =
TetanusToxoidWorker.getVaccines(latestRecord);
            this.checkups =
PregnancyCheckupWorker.getPregnancyCheckups(latestRecord);
            this.laborDel =
LaborAndDeliveryWorker.findLaborAndDelivery(latestRecord);
            this.familyPlans =
FamilyPlanWorker.getFamilyPlans(latestRecord);
            this.postCares =
PostpartumCareWorker.getPostCares(latestRecord);
            this.referrals =
ReferralWorker.getReferrals(latestRecord);
        }
    }
} catch(NumberFormatException nfe){
    nfe.printStackTrace();
}
}
}

public PregnancyRecord getPregRecord() {
    return pregRecord;
}

public void setPregRecord(PregnancyRecord pregRecord) {
    this.pregRecord = pregRecord;
}

public List<PregnancyCheckup> getCheckups() {
    return checkups;
}

public void setCheckups(List<PregnancyCheckup>
checkups) {
    this.checkups = checkups;
}

public LaborAndDelivery getLaborDel() {
    return laborDel;
}

public void setLaborDel(LaborAndDelivery laborDel) {
    this.laborDel = laborDel;
}

public List<FamilyPlanning> getFamilyPlans() {
    return familyPlans;
}

public void setFamilyPlans(List<FamilyPlanning>
familyPlans) {
    this.familyPlans = familyPlans;
}

public List<PostpartumCare> getPostCares() {
    return postCares;
}

public void setPostCares(List<PostpartumCare>
postCares) {

```

```

        this.postCares = postCares;
    }

public List<Referral> getReferrals() {
    return referrals;
}

public void setReferrals(List<Referral> referrals) {
    this.referrals = referrals;
}

public List<TetanusToxoid> getVaccines() {
    return vaccines;
}

public void setVaccines(List<TetanusToxoid> vaccines) {
    this.vaccines = vaccines;
}

public List<PregnancyCheckup> getFirstSem(){
    List<PregnancyCheckup> firstSem = new
ArrayList<PregnancyCheckup>();
    for(PregnancyCheckup checkup: this.checkups){
        if(checkup.getAog() < 4){
            firstSem.add(checkup);
        }
    }
    if(firstSem.size() == 0){ //add one empty row for display
purposes
        firstSem.add(null);
    }
    return firstSem;
}

public List<PregnancyCheckup> getSecondSem(){
    List<PregnancyCheckup> secondSem = new
ArrayList<PregnancyCheckup>();
    for(PregnancyCheckup checkup: this.checkups){
        if(checkup.getAog() >= 4 && checkup.getAog() <= 6){
            secondSem.add(checkup);
        }
    }
    if(secondSem.size() == 0){ //add one empty row for display
purposes
        secondSem.add(null);
    }
    return secondSem;
}

public List<PregnancyCheckup> getThirdSem(){
    List<PregnancyCheckup> thirdSem = new
ArrayList<PregnancyCheckup>();
    for(PregnancyCheckup checkup: this.checkups){
        if(checkup.getAog() > 6){
            thirdSem.add(checkup);
        }
    }
    if(thirdSem.size() == 0){ //add one empty row for display
purposes
        thirdSem.add(null);
    }
    return thirdSem;
}
}
}

```

Referral.java

```
package com.pregss.referral;

import java.util.Date;

public class Referral {
    private int referral_id;
    private int preg_record_id;
    private Date referral_date;
    private String problem;
    private String initial_action;
    private String remarks;
    private String referred_to;
    private Date examination_date;
    private String physician;
    private String referral_center;
    private String findings;
    private String secondary_action;
    private String instructions;

    public int getReferral_id() {
        return referral_id;
    }

    public void setReferral_id(int referral_id) {
        this.referral_id = referral_id;
    }

    public int getPreg_record_id() {
        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }

    public Date getReferral_date() {
        return referral_date;
    }

    public void setReferral_date(Date referral_date) {
        this.referral_date = referral_date;
    }

    public String getProblem() {
        return problem;
    }

    public void setProblem(String problem) {
        this.problem = problem;
    }

    public String getInitial_action() {
        return initial_action;
    }

    public void setInitial_action(String initial_action) {
        this.initial_action = initial_action;
    }

    public String getRemarks() {
        return remarks;
    }

    public void setRemarks(String remarks) {
        this.remarks = remarks;
    }
}
```

```
public String getReferred_to() {
    return referred_to;
}

public void setReferred_to(String referred_to) {
    this.referred_to = referred_to;
}

public Date getExamination_date() {
    return examination_date;
}

public void setExamination_date(Date examination_date) {
    this.examination_date = examination_date;
}

public String getPhysician() {
    return physician;
}

public void setPhysician(String physician) {
    this.physician = physician;
}

public String getReferral_center() {
    return referral_center;
}

public void setReferral_center(String referral_center) {
    this.referral_center = referral_center;
}

public String getFindings() {
    return findings;
}

public void setFindings(String findings) {
    this.findings = findings;
}

public String getSecondary_action() {
    return secondary_action;
}

public void setSecondary_action(String secondary_action) {
    this.secondary_action = secondary_action;
}

public String getInstructions() {
    return instructions;
}

public void setInstructions(String instructions) {
    this.instructions = instructions;
}
}
```

ReferralBean.java

```
package com.pregss.referral;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;
```

```

import com.pregss.util.PregssConstants;

public class ReferralBean {
    private Referral referral = new Referral();
    private int pregRecordId = -1;
    private int patientId = -1;

    public ReferralBean(){
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{
                this.patientId =
Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("prid")!=null){
            try{
                this.pregRecordId =
Integer.parseInt(request.getParameter("prid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("rid")!=null){
            try{
                int referralId =
Integer.parseInt(request.getParameter("rid"));
                this.referral =
ReferralWorker.findReferral(referralId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }
    }

    public Referral getReferral() {
        return referral;
    }

    public void setReferral(Referral referral) {
        this.referral = referral;
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public String addReferral(){
        if(this.pregRecordId!=-1){

```

```

            int ret=ReferralWorker.addReferral(this.referral,
pregRecordId);
            if(ret>0){
                try {
                    HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#referrals");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }else{
                HttpUtil.getRequest().setAttribute(PregssConstants.ERRO
R_MESSAGE, "Failed adding a referral entry.");
            }
        }
        return "";
    }

    public String update(){
        int ret = ReferralWorker.updateReferral(this.referral);
        HttpServletRequest request = HttpUtil.getRequest();
        if(ret > 0){
            request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Update successful.");
        }else{
            request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Update failed.");
        }
        return "";
    }

    public String delete(){
        int ret =
ReferralWorker.deleteReferral(this.referral.getReferral_id());
        if(ret>0){
            try {
                HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#referrals");
            } catch (IOException e) {
                e.printStackTrace();
            }
        }else{
            HttpUtil.getRequest().setAttribute(PregssConstants.ERRO
R_MESSAGE, "Delete failed.");
        }
        return "";
    }
}

```

ReferralWorker.java

```
package com.pregss.referral;
```

```
import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;
```

```
import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;
```

```

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class ReferralWorker {
    public static int addReferral(Referral referral, int
preg_record_id){
        String addRefSQL = "INSERT into referral " +

"(preg_record_id,referral_date,problem,initial_action," +
        "remarks,referred_to) " +
        "VALUES (?, ?, ?, ?, ?)";

        Object[] params = {
            preg_record_id,

PregssUtil.utilDateToSQLDate(referral.getReferral_date())
,
            referral.getProblem(),
            referral.getInitial_action(),
            referral.getRemarks(),
            referral.getReferred_to()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, addRefSQL, params);

                return ret;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }

        return -1;
    }

    public static int updateReferral(Referral referral){
        String addRefSQL = "UPDATE referral SET " +
            "referral_date = ?, problem = ?, initial_action = ?, " +
            "remarks = ?, referred_to = ?, examination_date = ?, "
+
            "physician = ?, referral_center = ?, findings = ?, " +
            "secondary_action = ?, instructions = ? " +
            "WHERE referral_id = ?";

        Object[] params = {

PregssUtil.utilDateToSQLDate(referral.getReferral_date())
,
            referral.getProblem(),
            referral.getInitial_action(),
            referral.getRemarks(),
            referral.getReferred_to(),

PregssUtil.utilDateToSQLDate(referral.getExamination_d
ate()),
            referral.getPhysician(),
            referral.getReferral_center(),
            referral.getFindings(),

```

```

            referral.getSecondary_action(),
            referral.getInstructions(),
            referral.getReferral_id()
        };

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, addRefSQL, params);

                return ret;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return -1;
        }

        return -1;
    }

    public static Referral findReferral(int id){
        String findReferralSQL = "SELECT * FROM referral
WHERE referral_id = ?";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<Referral> rsh = new
BeanHandler<Referral>(Referral.class);
                Referral referral = run.query(conn, findReferralSQL,
rsh, id);

                return referral;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }catch(SQLException e){
            return null;
        }

        return null;
    }

    public static List<Referral> getReferrals(int
pregRecordId){
        String getReferralsSQL = "SELECT * from referral " +
            "WHERE preg_record_id = ? " +
            "ORDER BY referral_date ";

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<Referral>> rsh = new
BeanListHandler<Referral>(Referral.class);
                List<Referral> checkups = run.query(conn,
getReferralsSQL, rsh, pregRecordId);

```

```

        return checkups;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return null;
}

return null;
}

public static int deleteReferral(int referralId){
    String deleteReferralSQL = "DELETE from referral
WHERE referral_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteReferralSQL,
referralId);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

GenerateReportBean.java

```

package com.pregss.report;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.LinkedHashMap;
import java.util.List;

import com.pregss.healthcenter.HealthCenterWorker;
import com.pregss.location.LocationWorker;
import com.pregss.user.User;
import com.pregss.user.UserBean;
import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class GenerateReportBean {
    private int regionId =
PregssConstants.DEFAULT_REGION;
    private int provinceId =
PregssConstants.DEFAULT_PROVINCE;
    private int cityId = 0;
    private int hcId = 0;

    private LinkedHashMap<String,Integer> regionMap =
LocationWorker.getRegionMap();
    private LinkedHashMap<String,Integer> provinceMap =
LocationWorker.getProvinceMap(this.regionId);

```

```

    private LinkedHashMap<String,Integer> cityMap =
LocationWorker.getCityMap(this.provinceId);
    private LinkedHashMap<String,Integer>
healthCenterMap =
HealthCenterWorker.getHealthCenterMap(this.cityId);

    private String year;
    private String month;

    public int getCityId() {
        return cityId;
    }

    public void setCityId(int cityId) {
        this.cityId = cityId;
    }

    public int getRegionId() {
        return regionId;
    }

    public void setRegionId(int regionId) {
        this.regionId = regionId;
    }

    public int getProvinceId() {
        return provinceId;
    }

    public void setProvinceId(int provinceId) {
        this.provinceId = provinceId;
    }

    public int getHcId() {
        return hcId;
    }

    public void setHcId(int hcId) {
        this.hcId = hcId;
    }

    public LinkedHashMap<String, Integer> getRegionMap() {
        return regionMap;
    }

    public LinkedHashMap<String, Integer> getProvinceMap()
{
        return provinceMap;
    }

    public LinkedHashMap<String, Integer> getCityMap() {
        return cityMap;
    }

    public LinkedHashMap<String, Integer>
getHealthCenterMap() {
        return healthCenterMap;
    }

    public void changeRegion(){
        this.provinceMap =
LocationWorker.getProvinceMap(this.regionId);
        this.provinceId = 0;
        changeProvince();
    }

    public void changeProvince(){

```



```

        this.cityMap =
LocationWorker.getCityMap(this.provinceId);
        this.cityId = 0;
        changeCity();
    }

    public void changeCity(){
        this.healthCenterMap =
HealthCenterWorker.getHealthCenterMap(this.cityId);
        this.hcId = 0;
    }

    public String getYear() {
        return year;
    }

    public void setYear(String year) {
        this.year = year;
    }

    public String getMonth() {
        //set current month as default value
        Calendar cal = Calendar.getInstance();
        cal.setTime(new Date());
        int month = cal.get(Calendar.MONTH) + 1;
        String monthString = String.format("%02d", month);

        return monthString;
    }

    public void setMonth(String month) {
        this.month = month;
    }

    public List<String> getYears(){
        List<String> years = new ArrayList<String>();
        int currentYear =
Calendar.getInstance().get(Calendar.YEAR);

        for(int year = PregssConstants.START_YEAR; year <=
currentYear; year++){
            years.add(Integer.toString(year));
        }

        return years;
    }

    public LinkedHashMap<String,String> getMonths(){
        return PregssConstants.MONTHS;
    }

    public void getReport(){
        UserBean userBean = HttpUtil.findBean("userBean");
        User user = userBean.getUser();

        String granularity = "barangay";
        int id = 0;

        if("HCA".equals(user.getUser_type())){
            granularity = "barangay";
            id = user.getHealth_center_id();
        }else if("PA".equals(user.getUser_type())){
            if(this.hcId != 0){
                granularity = "barangay";
                id = this.hcId;
            }else if(this.cityId != 0){
                granularity = "city";

```

```

            id = this.cityId;
        }else if(this.provinceId != 0){
            granularity = "province";
            id = this.provinceId;
        }else{
            granularity = "region";
            id = this.regionId;
        }
    }

    ReportGenerator.createExcelReport(this.month,
this.year, granularity, id);
}
}

ReportGenerator.java
package com.pregss.report;

import java.io.IOException;

import javax.faces.context.ExternalContext;
import javax.faces.context.FacesContext;

import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellStyle;
import org.apache.poi.ss.usermodel.Font;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.util.CellRangeAddress;

import com.pregss.healthcenter.HealthCenterWorker;
import com.pregss.location.LocationWorker;
import com.pregss.util.PregssUtil;

public class ReportGenerator {
    public static void createExcelReport(String month, String
year, String granularity, int id){
        HSSFWorkbook workbook = new HSSFWorkbook();
        HSSFSheet sheet = workbook.createSheet(month);

        CellStyle percentStyle = workbook.createCellStyle();

        percentStyle.setDataFormat(workbook.createDataFormat(
).getFormat("0.00%"));

        CellStyle boldStyle = workbook.createCellStyle();
        Font boldFont = workbook.createFont();
        boldFont.setBoldweight(Font.BOLDWEIGHT_BOLD);
        boldStyle.setFont(boldFont);

        //Create a new row in current sheet
        Row row = sheet.createRow(0);
        //Create a new cell in current row
        Cell cell = row.createCell(0);
        cell.setCellStyle(boldStyle);

        String title = "";
        if("barangay".equals(granularity)){
            title +=
HealthCenterWorker.findHealthCenter(id).getHealth_center
_name();
        }else if("city".equals(granularity)){
            title += LocationWorker.findCity(id).getName();
        }else if("province".equals(granularity)){
            title += LocationWorker.findProvince(id).getName();
        }else if("region".equals(granularity)){

```

```

        title += LocationWorker.findRegion(id).getName();
    }
    title += ": HIS (FSHIS) Report for " +
PregssUtil.getMonth(month) + " " + year;
    cell.setCellValue(title);

    sheet.addMergedRegion(new CellRangeAddress(
        0, //first row (0-based)
        0, //last row (0-based)
        0, //first column (0-based)
        4 //last column (0-based)
    ));

    row = sheet.createRow(1);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("MATERNAL CARE");
    cell = row.createCell(1);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("NO.");

    row = sheet.createRow(2);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("Pregnant with 3 or more Prenatal
visits (w/ at least 1 visit per trimester)");

    row.createCell(1).setCellValue(ReportsWorker.getPreg3or
MoreVisits(month, year, granularity, id));

    row = sheet.createRow(3);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("Pregnant given TT2 plus");

    row.createCell(1).setCellValue(ReportsWorker.getPregGiv
enTT2plus(month, year, granularity, id));

    row = sheet.createRow(4);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("Pregnant given Complete Iron
dosage");

    row.createCell(1).setCellValue(ReportsWorker.getPregGiv
enCompleteIron(month, year, granularity, id));

    row = sheet.createRow(5);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("Postpartum (PP) w/ at least 1 PP
visit");

    row.createCell(1).setCellValue(ReportsWorker.getPPatLea
st1Visit(month, year, granularity, id));

    row = sheet.createRow(6);
    cell = row.createCell(0);
    cell.setCellStyle(boldStyle);
    cell.setCellValue("PP mothers given Complete Iron
dosage");

    row.createCell(1).setCellValue(ReportsWorker.getPPGiven
CompleteIron(month, year, granularity, id));

    row = sheet.createRow(7);
    cell = row.createCell(0);

```

```

        cell.setCellStyle(boldStyle);
        cell.setCellValue("PP women initiated breastfeeding");

        row.createCell(1).setCellValue(ReportsWorker.getPPinitiat
eBreastfeed(month, year, granularity, id));

        row = sheet.createRow(8);
        cell = row.createCell(0);
        cell.setCellStyle(boldStyle);
        cell.setCellValue("Breastfeeding women given Vitamin
A");

        row.createCell(1).setCellValue(ReportsWorker.getBreastfe
edGivenVitaminA(month, year, granularity, id));

        row = sheet.createRow(9);
        cell = row.createCell(0);
        cell.setCellStyle(boldStyle);
        cell.setCellValue("Women 15-49 given iodized oil
capsules");

        row.createCell(1).setCellValue(ReportsWorker.getWomenG
ivenIodizedCapsules(month, year, granularity, id));

        sheet.autoSizeColumn(0);
        sheet.autoSizeColumn(1);

        FacesContext facesContext =
FacesContext.getCurrentInstance();
        ExternalContext externalContext =
facesContext.getExternalContext();

        externalContext.setContentType("application/vnd.
ms-excel");
        externalContext.setResponseHeader("Content-
Disposition", "attachment; filename=\"\" + \"FSHIS_Report\"
+ month + \"\" + year + \").xls\"");

        try {

            workbook.write(externalContext.getResponseOutputStrea
m());
        } catch (IOException e) {
            e.printStackTrace();
        }

        facesContext.responseComplete();
    }
}

```

ReportsWorker.java

```

package com.pregss.report;

import java.sql.Connection;
import java.sql.SQLException;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class ReportsWorker {
    public static String generateLocationBasedQuery(String
subQuery, String granularity, int id){
        StringBuilder sb = new StringBuilder();

```

```

sb.append("SELECT count(*) FROM ( ");
sb.append("SELECT pr.* FROM pregnancy_record pr ");
sb.append("JOIN barangay_health_center bhc ");
sb.append("ON bhc.health_center_id =
pr.health_center_id ");
if("barangay".equals(granularity)){
sb.append("WHERE bhc.health_center_id = " + id + "
");
}else if("city".equals(granularity)){
sb.append("JOIN city c ");
sb.append("ON bhc.city_id = c.city_id ");
sb.append("WHERE c.city_id = " + id + " ");
}else if("province".equals(granularity)){
sb.append("JOIN city c ");
sb.append("ON bhc.city_id = c.city_id ");
sb.append("JOIN province p ");
sb.append("ON c.province_id = p.province_id ");
sb.append("WHERE p.province_id = " + id + " ");
}else if("region".equals(granularity)){
sb.append("JOIN city c ");
sb.append("ON bhc.city_id = c.city_id ");
sb.append("JOIN province p ");
sb.append("ON c.province_id = p.province_id ");
sb.append("JOIN region r ");
sb.append("ON p.region_id = r.region_id ");
sb.append("WHERE r.region_id = " + id + " ");
}
sb.append(") pr ");
sb.append(subQuery);
return sb.toString();
}

```

```

public static int getCount(String mainSQL, String
granularity, int id){
String getCountSQL =
generateLocationBasedQuery(mainSQL, granularity, id);
try{
Connection conn = null;
QueryRunner run = new QueryRunner();
try{
conn = ConnectionManager.getConnection();
int count = run.query(conn, getCountSQL,
PregUtil.firstIntRsh);

return count;
}catch(SQLException e){
e.printStackTrace();
}finally{
DbUtils.close(conn);
}
}catch(SQLException e){
return 0;
}

return 0;
}

```

```

public static int getPreg3orMoreVisits(String month,
String year, String granularity, int id){
String subSQL =
"WHERE EXISTS " +
"(" +
" SELECT * " +
" FROM pregnancy_checkup pc " +
" WHERE pc.preg_record_id = pr.preg_record_id " +

```

```

" AND DATE_FORMAT(pc.visit_date, '%Y%m') = " +
year + month + " " +
" ) " +
"AND EXISTS " +
"(" +
" SELECT * " +
" FROM pregnancy_checkup pc " +
" WHERE pc.preg_record_id = pr.preg_record_id " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m')," + year + month + " ) <= 0 " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m'), date_format(pr.lmp, '%Y%m')) between 0 and 3 " +
" ) " +
"AND EXISTS " +
"(" +
" SELECT * " +
" FROM pregnancy_checkup pc " +
" WHERE pc.preg_record_id = pr.preg_record_id " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m')," + year + month + " ) <= 0 " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m'), date_format(pr.lmp, '%Y%m')) between 4 and 6 " +
" ) " +
"AND EXISTS " +
"(" +
" SELECT * " +
" FROM pregnancy_checkup pc " +
" WHERE pc.preg_record_id = pr.preg_record_id " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m')," + year + month + " ) <= 0 " +
" AND PERIOD_DIFF(date_format(pc.visit_date,
'%Y%m'), date_format(pr.lmp, '%Y%m')) between 7 and 9 " +
" ) ";
return getCount(subSQL, granularity, id);
}

```

```

public static int getPregGivenTT2plus(String month,
String year, String granularity, int id){
String subSQL =
"JOIN " +
"(SELECT preg_record_id, count FROM " +
" (SELECT preg_record_id, count(*) as count " +
" from tetanus_toxoid tt " +
" WHERE PERIOD_DIFF(date_format(date_given,
'%Y%m')," + year + month + " ) <= 0 " +
" group by preg_record_id) tt " +
"WHERE count > 1 " +
"AND EXISTS(" +
" SELECT x.preg_record_id from tetanus_toxoid x "
+
" where DATE_FORMAT(date_given, '%Y%m') = " +
+ year + month + " " +
" AND x.preg_record_id = tt.preg_record_id " +
" )" +
" ) tt " +
"ON pr.preg_record_id = tt.preg_record_id ";
return getCount(subSQL, granularity, id);
}

```

```

public static int getPregGivenCompleteIron(String month,
String year, String granularity, int id){
String subSQL =
"JOIN " +
"(SELECT preg_record_id, sum FROM " +
" (SELECT cu.preg_record_id,
sum(a.iron_folate_rx) as sum " +
" from pregnancy_checkup cu, action a " +

```

```

        " WHERE a.checkup_id = cu.checkup_id " +
        " AND PERIOD_DIFF(date_format(cu.visit_date,
"%Y%m")," + year + month + ") <= 0 " +
        " group by cu.preg_record_id) a " +
        "WHERE EXISTS( " +
        " SELECT cu.preg_record_id from
pregnancy_checkup cu " +
        " WHERE DATE_FORMAT(visit_date, '%Y%m') =
" + year + month + " " +
        " AND cu.preg_record_id = a.preg_record_id " +
        " )" +
        "AND sum >= 180 " +
        ") a " +
        "ON pr.preg_record_id = a.preg_record_id";
return getCount(subSQL, granularity, id);
}

public static int getPPatLeast1Visit(String month, String
year, String granularity, int id){
String subSQL =
    "WHERE EXISTS " +
    "( " +
    " SELECT * " +
    " FROM postpartum_care pp, labor_and_delivery ld
" +
    " WHERE pp.preg_record_id = pr.preg_record_id "
+
    " AND ld.preg_record_id = pp.preg_record_id " +
    " AND DATEDIFF(pp.visit_date, ld.delivery_date)
between 0 and 1 " +
    ")AND EXISTS " +
    "( " +
    " SELECT * " +
    " FROM postpartum_care pp " +
    " WHERE pp.preg_record_id = pr.preg_record_id "
+
    " AND DATE_FORMAT(pp.visit_date, '%Y%m') = " +
year + month + " " +
    ")";

return getCount(subSQL, granularity, id);
}

public static int getPPGivenCompleteIron(String month,
String year, String granularity, int id){
String subSQL =
    "JOIN " +
    "(SELECT preg_record_id, sum FROM " +
    " (SELECT preg_record_id, sum(iron_folate_rx) as
sum " +
    " from postpartum_care " +
    " WHERE PERIOD_DIFF(date_format(visit_date,
"%Y%m")," + year + month + ") <= 0 " +
    " group by preg_record_id) a " +
    "WHERE EXISTS( " +
    " SELECT * from postpartum_care pp " +
    " WHERE DATE_FORMAT(visit_date, '%Y%m') =
" + year + month + " " +
    " AND pp.preg_record_id = a.preg_record_id " +
    " )" +
    "AND sum >= 90 " +
    ") a " +
    "ON pr.preg_record_id = a.preg_record_id";
return getCount(subSQL, granularity, id);
}

```

```

public static int getPPInitiateBreastfeed(String month,
String year, String granularity, int id){
String subSQL =
    "WHERE EXISTS " +
    "( " +
    " SELECT * " +
    " FROM labor_and_delivery ld " +
    " WHERE ld.preg_record_id = pr.preg_record_id " +
    " AND ld.imm_breastfeed = 'Y' " +
    ")AND EXISTS " +
    "( " +
    " SELECT * " +
    " FROM postpartum_care pp " +
    " WHERE pp.preg_record_id = pr.preg_record_id "
+
    " AND DATE_FORMAT(pp.visit_date, '%Y%m') = " +
year + month + " " +
    ")";
return getCount(subSQL, granularity, id);
}

public static int getBreastfeedGivenVitaminA(String
month, String year, String granularity, int id){
String subSQL =
    "WHERE EXISTS " +
    "( " +
    " SELECT * " +
    " FROM postpartum_care pp, labor_and_delivery ld
" +
    " WHERE pp.preg_record_id = pr.preg_record_id "
+
    " AND ld.preg_record_id = pp.preg_record_id " +
    " AND pp.vitamin_a = 'Y' " +
    " AND pp.breastfeed_only = 'Y' " +
    " AND DATEDIFF(pp.visit_date, ld.delivery_date)
between 0 and 28 " +
    ")AND EXISTS " +
    "( " +
    " SELECT * " +
    " FROM postpartum_care pp " +
    " WHERE pp.preg_record_id = pr.preg_record_id "
+
    " AND DATE_FORMAT(pp.visit_date, '%Y%m') = " +
year + month + " " +
    ")";
return getCount(subSQL, granularity, id);
}

public static int getWomenGivenIodizedCapsules(String
month, String year, String granularity, int id){
String subSQL =
    "WHERE EXISTS " +
    "( " +
    " SELECT * " +
    " FROM pregnancy_checkup cu, action a " +
    " WHERE a.checkup_id = cu.checkup_id " +
    " AND cu.preg_record_id = pr.preg_record_id " +
    " AND a.iodine_high_risk = 'Y' " +
    " AND DATE_FORMAT(cu.visit_date, '%Y%m') = " +
year + month + " " +
    ")AND EXISTS " +
    "( " +
    " SELECT * " +
    " FROM patient p " +
    " WHERE p.patient_id = pr.patient_id " +
    " AND p.age between 15 and 49 " +
    ")";
}

```

```

        return getCount(subSQL, granularity, id);
    }
}

TetanusToxoid.java
package com.pregss.tetanus_toxoid;

import java.util.Date;

public class TetanusToxoid {
    private int tt_id = -1;
    private Date date_given;
    private int preg_record_id;

    public int getTt_id() {
        return tt_id;
    }

    public void setTt_id(int tt_id) {
        this.tt_id = tt_id;
    }

    public int getPreg_record_id() {
        return preg_record_id;
    }

    public void setPreg_record_id(int preg_record_id) {
        this.preg_record_id = preg_record_id;
    }

    public Date getDate_given() {
        return date_given;
    }

    public void setDate_given(Date date_given) {
        this.date_given = date_given;
    }
}

TetanusToxoidBean.java
package com.pregss.tetanus_toxoid;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;

import com.pregss.util.HttpUtil;
import com.pregss.util.PregssConstants;

public class TetanusToxoidBean {
    private List<TetanusToxoid> vaccines;
    private List<TetanusToxoid> newVaccines = new
    ArrayList<TetanusToxoid>();

    private int patientId = -1;
    private int pregRecordId = -1;

    public TetanusToxoidBean() {
        HttpServletRequest request = HttpUtil.getRequest();

        if(request.getParameter("pid")!=null){
            try{

```

```

                this.patientId =
                Integer.parseInt(request.getParameter("pid"));
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

        if(request.getParameter("prid")!=null){
            try{
                this.pregRecordId =
                Integer.parseInt(request.getParameter("prid"));
                this.vaccines =
                TetanusToxoidWorker.getVaccines(this.pregRecordId);
            }catch(NumberFormatException nfe){
                nfe.printStackTrace();
            }
        }

    }

    public List<TetanusToxoid> getVaccines() {
        return vaccines;
    }

    public void setVaccines(List<TetanusToxoid> vaccines) {
        this.vaccines = vaccines;
    }

    public int getPatientId() {
        return patientId;
    }

    public void setPatientId(int patientId) {
        this.patientId = patientId;
    }

    public int getPregRecordId() {
        return pregRecordId;
    }

    public void setPregRecordId(int pregRecordId) {
        this.pregRecordId = pregRecordId;
    }

    public List<TetanusToxoid> getNewVaccines() {
        return newVaccines;
    }

    public void setNewVaccines(List<TetanusToxoid>
    newVaccines) {
        this.newVaccines = newVaccines;
    }

    public void addNewVaccine(){
        this.newVaccines.add(new TetanusToxoid());
    }

    public void removeNewVaccine(){
        if(this.newVaccines.size() > 0){
            this.newVaccines.remove(this.newVaccines.size()-1);
        }
    }

    public String addVaccines(){
        if(this.patientId != -1){
            int ret =
            TetanusToxoidWorker.addVaccines(this.pregRecordId,
            this.newVaccines);
            if(ret>0){
                try {

```

```

    HttpUtil.redirect("view_patient.xhtml?pid="+this.patientI
d+"#patient_history");
    } catch (IOException e) {
        e.printStackTrace();
    }
} else {

    HttpUtil.getRequest().setAttribute(PregssConstants.ERROR
MESSAGE, "Failed adding Tetanus Toxoid history
information.");
    }
}
return "";
}

public String deleteVaccine(){
    HttpServletRequest request = HttpUtil.getRequest();

    if(request.getParameter("ttid")!=null){
        try{
            int tt_id =
Integer.parseInt(request.getParameter("ttid"));
            int ret = TetanusToxoidWorker.deleteVaccine(tt_id);
            if(ret > 0){

                request.setAttribute(PregssConstants.SUCCESS_MESSA
GE, "Vaccine deleted.");

                //refresh vaccine list
                this.vaccines =
TetanusToxoidWorker.getVaccines(this.pregRecordId);
            } else {

                request.setAttribute(PregssConstants.ERROR_MESSAGE,
"Failed to remove vaccine.");
            }
        } catch (NumberFormatException nfe){
            nfe.printStackTrace();
        }
    }

    return "";
}
}
}

```

TetanusToxoidWorker.java

```

package com.pregss.tetanus_toxoid;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import
org.apache.commons.dbutils.handlers.BeanListHandler;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class TetanusToxoidWorker {
    public static List<TetanusToxoid> getVaccines(int
pregRecordId){
        String getVaccinesSQL = "SELECT * from
tetanus_toxoid " +

```

```

        "WHERE preg_record_id = ? "+
        "ORDER BY date_given ";
        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<List<TetanusToxoid>> rsh = new
BeanListHandler<TetanusToxoid>(TetanusToxoid.class);
                List<TetanusToxoid> checkups = run.query(conn,
getVaccinesSQL, rsh, pregRecordId);

                return checkups;
            } catch (SQLException e){
                e.printStackTrace();
            } finally {
                DbUtils.close(conn);
            }
        } catch (SQLException e){
            return null;
        }

        return null;
    }

    public static int addVaccines(int pregRecordId,
List<TetanusToxoid> vaccines){
        StringBuilder sb = new StringBuilder();
        sb.append("INSERT into tetanus_toxoid (date_given,
preg_record_id) VALUES ");

        int count = 0;
        for(TetanusToxoid vaccine: vaccines){
            if(count > 0){
                sb.append(", ");
            }
            sb.append("(" +
PregssUtil.utilDateToSQLDate(vaccine.getDate_given()) +
""");
            sb.append(", " + pregRecordId);
            sb.append(")");
            count++;
        }

        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                int ret = run.update(conn, sb.toString());

                return ret;
            } catch (SQLException e){
                e.printStackTrace();
            } finally {
                DbUtils.close(conn);
            }
        } catch (SQLException e){
            return -1;
        }

        return -1;
    }

    public static int deleteVaccine(int tt_id){
        String deleteVaccineSQL = "DELETE from
tetanus_toxoid WHERE tt_id = ?";

```

```

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, deleteVaccineSQL, tt_id);

        return ret;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}
}

```

User.java

```

package com.pregss.user;

public class User {
    private int user_id = -1;
    private String username;
    private String password;
    private String first_name;
    private String middle_name;
    private String last_name;
    private String job_title;
    private String affiliation;
    private String email_address;
    private String contact_no;
    private String user_type;
    private String active_flag;
    private int health_center_id;

    //not in table
    private String health_center_name;
    private String new_password;

    public int getUser_id() {
        return user_id;
    }
    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getFirst_name() {
        return first_name;
    }
    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }

```

```

}
    public String getMiddle_name() {
        return middle_name;
    }
    public void setMiddle_name(String middle_name) {
        this.middle_name = middle_name;
    }
    public String getLast_name() {
        return last_name;
    }
    public void setLast_name(String last_name) {
        this.last_name = last_name;
    }
    public String getJob_title() {
        return job_title;
    }
    public void setJob_title(String job_title) {
        this.job_title = job_title;
    }
    public String getAffiliation() {
        return affiliation;
    }
    public void setAffiliation(String affiliation) {
        this.affiliation = affiliation;
    }
    public String getEmail_address() {
        return email_address;
    }
    public void setEmail_address(String email_address) {
        this.email_address = email_address;
    }
    public String getContact_no() {
        return contact_no;
    }
    public void setContact_no(String contact_no) {
        this.contact_no = contact_no;
    }
    public String getUser_type() {
        return user_type;
    }
    public void setUser_type(String user_type) {
        this.user_type = user_type;
    }
    public String getActive_flag() {
        return active_flag;
    }
    public void setActive_flag(String active_flag) {
        this.active_flag = active_flag;
    }
    public int getHealth_center_id() {
        return health_center_id;
    }
    public void setHealth_center_id(int health_center_id) {
        this.health_center_id = health_center_id;
    }
    public String getHealth_center_name() {
        return health_center_name;
    }
    public void setHealth_center_name(String
health_center_name) {
        this.health_center_name = health_center_name;
    }
    public String getNew_password() {
        return new_password;
    }
    public void setNew_password(String new_password) {
        this.new_password = new_password;
    }

```

```

    }
}

```

UserBean.java

```

package com.pregss.user;

public class UserBean {
    private User user;

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public boolean isLoggedIn() {
        return user != null;
    }
}

```

UserWorker.java

```

package com.pregss.user;

import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;

import org.apache.commons.dbutils.DbUtils;
import org.apache.commons.dbutils.QueryRunner;
import org.apache.commons.dbutils.ResultSetHandler;
import org.apache.commons.dbutils.handlers.BeanHandler;
import org.apache.commons.dbutils.handlers.BeanListHandler;
import org.apache.commons.lang3.StringUtils;

import com.pregss.db.ConnectionManager;
import com.pregss.util.PregssUtil;

public class UserWorker {
    public static User findUser(String username, String password){
        String findUserSQL = "SELECT user.*,
bhc.health_center_name FROM user,
barangay_health_center bhc " +
            " WHERE username='" + username + "'" +
            " AND password=md5('" + password + "') AND
user.health_center_id = bhc.health_center_id";
        try{
            Connection conn = null;
            QueryRunner run = new QueryRunner();
            try{
                conn = ConnectionManager.getConnection();
                ResultSetHandler<User> rsh = new
BeanHandler<User>(User.class);
                User user = run.query(conn, findUserSQL, rsh);

                return user;
            }catch(SQLException e){
                e.printStackTrace();
            }finally{
                DbUtils.close(conn);
            }
        }
    }
}

```

```

    }catch(SQLException e){
        return null;
    }

    return null;
}

public static User findUser(int userId){
    String findUserSQL = "SELECT * FROM user WHERE
user_id = ? ";
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<User> rsh = new
BeanHandler<User>(User.class);
            User user = run.query(conn, findUserSQL, rsh,
userId);

            return user;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

public static User findUser(String username){
    String findUserSQL = "SELECT * FROM user WHERE
username = ? ";
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<User> rsh = new
BeanHandler<User>(User.class);
            User user = run.query(conn, findUserSQL, rsh,
username);

            return user;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

public static int addUser(User user, int health_center_id){
    String addUserSQL = "INSERT into user " +
+
"(username,password,first_name,middle_name,last_name,"
+
"job_title,affiliation,user_type,email_address,contact_no,"
+

```



```

        "active_flag,health_center_id" +
        "VALUES (?,md5(?,?,?,?,?,?,?,?));";
Object[] params = {
    user.getUsername(),
    user.getPassword(),
    user.getFirst_name(),
    user.getMiddle_name(),
    user.getLast_name(),
    user.getJob_title(),
    user.getAffiliation(),
    user.getUser_type(),
    user.getEmail_address(),
    user.getContact_no(),
    user.getActive_flag(),
    health_center_id
};

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, addUserSQL, params);

        return ret;
    }catch(SQLException e){
        if(e.getMessage().contains("Duplicate entry")){
            return -2;
        }
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}

public static int updateUser(User user){
    if(StringUtils.isEmpty(user.getNew_password())){
        user.setPassword(user.getNew_password());
    }

    String updateUserSQL = "UPDATE user SET
first_name=?, middle_name=?, last_name=?, password=";

    if(StringUtils.isEmpty(user.getNew_password())){
        user.setPassword(user.getNew_password());
        updateUserSQL += "md5(?)";
    }else{
        updateUserSQL += "?";
    }

    updateUserSQL += ", job_title=?, affiliation=?,
email_address=?, " +
        "contact_no=? WHERE user_id=?";
Object[] params = {
    user.getFirst_name(),
    user.getMiddle_name(),
    user.getLast_name(),
    user.getPassword(),
    user.getJob_title(),
    user.getAffiliation(),
    user.getEmail_address(),
    user.getContact_no(),

```

```

    user.getUser_id()
};

try{
    Connection conn = null;
    QueryRunner run = new QueryRunner();
    try{
        conn = ConnectionManager.getConnection();
        int ret = run.update(conn, updateUserSQL, params);

        return ret;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
}catch(SQLException e){
    return -1;
}

return -1;
}

public static List<User> getUsersUnderHC(int
health_center_id, int offset, int numRows){
    String getUsersUnderHCSQL = "SELECT * FROM user
WHERE health_center_id = ? " +
        "AND user_type='RU' AND active_flag='Y'" +
        "LIMIT " + offset + "," + numRows;
    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            ResultSetHandler<List<User>> rsh = new
BeanListHandler<User>(User.class);
            List<User> users = run.query(conn,
getUsersUnderHCSQL, rsh, health_center_id);

            return users;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return null;
    }

    return null;
}

public static int getTotalUsersUnderHC(int
health_center_id){
    String getTotalUsersSQL = "SELECT count(*) FROM
user " +
        "WHERE health_center_id = ? AND user_type='RU'
AND active_flag='Y';

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();

            int total = run.query(conn, getTotalUsersSQL,
PreggsUtil.firstIntRsh, health_center_id);

```

```

        return total;
    }catch(SQLException e){
        e.printStackTrace();
    }finally{
        DbUtils.close(conn);
    }
    }catch(SQLException e){
        return 0;
    }

    return 0;
}

public static int deleteUser(int user_id){
    String deleteUserSQL = "DELETE from user WHERE
user_id = ?";

    try{
        Connection conn = null;
        QueryRunner run = new QueryRunner();
        try{
            conn = ConnectionManager.getConnection();
            int ret = run.update(conn, deleteUserSQL, user_id);

            return ret;
        }catch(SQLException e){
            e.printStackTrace();
        }finally{
            DbUtils.close(conn);
        }
    }catch(SQLException e){
        return -1;
    }

    return -1;
}
}

```

HttpUtil.java

```

package com.pregss.util;

import java.io.IOException;

import javax.faces.context.FacesContext;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class HttpUtil {

    public static HttpServletRequest getRequest(){
        return (HttpServletRequest)
FacesContext.getCurrentInstance().getExternalContext().get
Request();
    }

    public static HttpServletResponse getResponse(){
        return (HttpServletResponse)
FacesContext.getCurrentInstance().getExternalContext().get
Response();
    }

    public static HttpSession getSession(boolean createNew){
        return (HttpSession)
FacesContext.getCurrentInstance().getExternalContext().get
Session(createNew);
    }
}

```

```

    }

    public static void redirect(String url) throws IOException{
        FacesContext.getCurrentInstance().getExternalContext().r
edirect(url);
    }

    public static void dispatch(String url) throws IOException{
        FacesContext.getCurrentInstance().getExternalContext().d
ispatch(url);
    }

    public static <T> T findBean(String beanName) {
        FacesContext context =
FacesContext.getCurrentInstance();
        return (T)
FacesContext.getCurrentInstance().getApplication().evaluate
ExpressionGet(context, "#{"+ beanName + "}", Object.class);
    }
}

```

PagingHelper.java

```

package com.pregss.util;

import java.util.ArrayList;
import java.util.List;

public class PagingHelper {
    private int page = 1;
    private int numRows = 1;
    private int totalPages = 0;
    private List<Integer> pageList = new
ArrayList<Integer>();
    private String sortBy;

    public PagingHelper(){

    }

    public PagingHelper(int totalResults, int numRows){
        this.numRows = numRows;
        this.totalPages = (int) Math.ceil(totalResults * 1.0 /
numRows);
        for(int i = 0; i < totalPages; i++){
            this.pageList.add(i+1);
        }
    }

    public int getPage() {
        return page;
    }

    public void setPage(int page) {
        this.page = page;
    }

    public int getTotalPages() {
        return totalPages;
    }

    public void setTotalPages(int totalPages) {
        this.totalPages = totalPages;
    }

    public int getNumRows() {

```

```

    return numRows;
}

public void setNumRows(int numRows) {
    this.numRows = numRows;
}

public List<Integer> getPageList() {
    return pageList;
}

public void setPageList(List<Integer> pageList) {
    this.pageList = pageList;
}

public String getSortBy() {
    return sortBy;
}

public void setSortBy(String sortBy) {
    this.sortBy = sortBy;
}
}

```

PregssConstants.java

```

package com.pregss.util;

import java.util.LinkedHashMap;

public class PregssConstants {
    //Data Constants
    public static final int DEFAULT_REGION = 3; //NCR
    public static final int DEFAULT_PROVINCE = 1; //Metro
    Manila
    public static final String[] JOB_TITLES = new
    String[]{"General
    Practitioner","Obstetrician","Midwife","Others"};
    public static final String[] DELIVERY_TYPES = new
    String[]{"Normal","Caesarean"};
    public static final String TETANUS_TOXOID = "Tetanus
    Toxoid";

    //Request,Response,Session Constants
    public static final String ERROR_MESSAGE = "errorMsg";
    public static final String SUCCESS_MESSAGE =
    "successMsg";

    //Navigation Constants
    public static final String HOME = "home";
    public static final String MANAGE_HEALTH_CENTERS
    = "manage_hc";
    public static final String HEALTH_CENTER_REQUESTS
    = "hc_requests";
    public static final String MANAGE_USERS =
    "manage_users";
    public static final String USER_DETAILS = "user_details";
    public static final String SEARCH_PATIENT =
    "search_patient";
    public static final String PATIENT_LIST = "patient_list";
    public static final String VIEW_PATIENT =
    "view_patient";
    public static final String REGISTER_SUCCESS =
    "register_success";

    //Reports

```

```

    public static final String WOMEN_POPULATION =
    "Population of Women";
    public static final String LIVE_BIRTHS = "Live Births";
    public static final String LOW_BIRTH_WEIGHT = "Low
    Birth Weight";
    public static final String PERINATAL_MORTALITY =
    "Perinatal Mortality";
    public static final String C_SECTIONS = "Caesarean
    Sections";
    public static final String FOLIC_ACID = "Folic Acid
    Vitamin Usage";
    public static final int START_YEAR = 2013;

    public static final LinkedHashMap<String,String>
    MONTHS = new LinkedHashMap<String,String>(){
        {
            put("January","01");
            put("February","02");
            put("March","03");
            put("April","04");
            put("May","05");
            put("June","06");
            put("July","07");
            put("August","08");
            put("September","09");
            put("October","10");
            put("November","11");
            put("December","12");
        }
    };
}

```

PregssUtil.java

```

package com.pregss.util;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.apache.commons.dbutils.ResultSetHandler;

public class PregssUtil {
    //ResultSetHandler
    public static ResultSetHandler<Integer> firstIntRsh = new
    ResultSetHandler<Integer>() {
        public Integer handle(ResultSet rs) throws
    SQLException {
            if (!rs.next()) {
                return -1;
            }else{
                return rs.getInt(1);
            }
        }
    };

    public static java.sql.Date
    utilDateToSQLDate(java.util.Date date){
        return new java.sql.Date(date.getTime());
    }

    public static String getMonth(String monthNum){
        if("01".equals(monthNum)){
            return "January";
        }else if("02".equals(monthNum)){
            return "February";
        }else if("03".equals(monthNum)){
            return "March";
        }else if("04".equals(monthNum)){

```

```

        return "April";
    }else if("05".equals(monthNum)){
        return "May";
    }else if("06".equals(monthNum)){
        return "June";
    }else if("07".equals(monthNum)){
        return "July";
    }else if("08".equals(monthNum)){
        return "August";
    }else if("09".equals(monthNum)){
        return "September";
    }else if("10".equals(monthNum)){
        return "October";
    }else if("11".equals(monthNum)){
        return "November";
    }else if("12".equals(monthNum)){
        return "December";
    }
    }
    return null;
}
}

```

PropertiesUtil.java

```
package com.pregss.util;
```

```
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;
```

```
public class PropertiesUtil {
    private static Properties properties = new Properties();

    static{
        try {
            //load a properties file
            ClassLoader classLoader =
Thread.currentThread().getContextClassLoader();
            InputStream stream =
classLoader.getResourceAsStream("pregss.properties");
            properties.load(stream);
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    public static String getProperty(String propertyName){
        return properties.getProperty(propertyName);
    }
}

```

ConfirmPasswordValidator.java

```
package com.pregss.validator;
```

```
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.component.UIInput;
import javax.faces.context.FacesContext;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;
```

```
public class ConfirmPasswordValidator implements
Validator {
```

```
    @Override
```

```

        public void validate(FacesContext context, UIComponent
component, Object value) throws ValidatorException {
            UIInput passwordComponent = (UIInput)
component.getAttributes().get("passwordComponent");
            String password = (String)
passwordComponent.getValue();
            String confirmPassword = (String) value;

            if (confirmPassword != null &&
!confirmPassword.equals(password)) {
                passwordComponent.setValid(false);
                FacesMessage message = new FacesMessage("Repeat
password is not the same as password");

                message.setSeverity(FacesMessage.SEVERITY_ERROR);
                throw new ValidatorException(message);
            }
        }
    }
}

```

```
pregss.properties
```

```
#jdbc.url=jdbc:mysql://localhost/pregss
#jdbc.user=root
#jdbc.pass=
```

```
jdbc.url=jdbc:mysql://localhost/PregnancyMonitor
jdbc.user=PregnancyMonitor
jdbc.pass=r6dB837nuNXcRZDP
```

```
PreggsMessages.properties
```

```

#
# DO NOT ALTER OR REMOVE COPYRIGHT NOTICES
OR THIS HEADER.
#
# Copyright (c) 2010 Oracle and/or its affiliates. All rights
reserved.
#
# The contents of this file are subject to the terms of either
the GNU
# General Public License Version 2 only ("GPL") or the
Common Development
# and Distribution License("CDDL") (collectively, the
"License"). You
# may not use this file except in compliance with the License.
You can
# obtain a copy of the License at
# https://glassfish.dev.java.net/public/CDDL+GPL_1_1.html
# or packager/legal/LICENSE.txt. See the License for the
specific
# language governing permissions and limitations under the
License.
#
# When distributing the software, include this License
Header Notice in each
# file and include the License file at
packager/legal/LICENSE.txt.
#
# GPL Classpath Exception:
# Oracle designates this particular file as subject to the
"Classpath"
# exception as provided by Oracle in the GPL Version 2
section of the License
# file that accompanied this code.
#

```

```

# Modifications:
# If applicable, add the following below the License Header,
with the fields
# enclosed by brackets [] replaced by your own identifying
information:
# "Portions Copyright [year] [name of copyright owner]"
#
# Contributor(s):
# If you wish your version of this file to be governed by only
the CDDL or
# only the GPL Version 2, indicate your decision by adding
"[Contributor]
# elects to include this software in this distribution under
the [CDDL or GPL
# Version 2] license." If you don't indicate a single choice of
license, a
# recipient has the option to distribute your version of this
file under
# either the CDDL, the GPL Version 2 or to extend the
choice of license to
# its licensees as provided above. However, if you add GPL
Version 2 code
# and therefore, elected the GPL Version 2 license, then the
option applies
# only if the new code is made subject to such option by the
copyright
# holder.
#

# DO NOT ALTER OR REMOVE COPYRIGHT NOTICES
OR THIS HEADER.
#
# Copyright 1997-2007 Sun Microsystems, Inc. All rights
reserved.
#
# The contents of this file are subject to the terms of either
the GNU
# General Public License Version 2 only ("GPL") or the
Common Development
# and Distribution License("CDDL") (collectively, the
"License"). You
# may not use this file except in compliance with the
License. You can obtain
# a copy of the License at
https://glassfish.dev.java.net/public/CDDL+GPL.html
# or glassfish/bootstrap/legal/LICENSE.txt. See the License
for the specific
# language governing permissions and limitations under the
License.
#
# When distributing the software, include this License
Header Notice in each
# file and include the License file at
glassfish/bootstrap/legal/LICENSE.txt.
# Sun designates this particular file as subject to the
"Classpath" exception
# as provided by Sun in the GPL Version 2 section of the
License file that
# accompanied this code. If applicable, add the following
below the License
# Header, with the fields enclosed by brackets [] replaced by
your own
# identifying information: "Portions Copyrighted [year]
# [name of copyright owner]"
#
# Contributor(s):
#

```

```

# If you wish your version of this file to be governed by only
the CDDL or
# only the GPL Version 2, indicate your decision by adding
"[Contributor]
# elects to include this software in this distribution under
the [CDDL or GPL
# Version 2] license." If you don't indicate a single choice of
license, a
# recipient has the option to distribute your version of this
file under
# either the CDDL, the GPL Version 2 or to extend the
choice of license to
# its licensees as provided above. However, if you add GPL
Version 2 code
# and therefore, elected the GPL Version 2 license, then the
option applies
# only if the new code is made subject to such option by the
copyright
# holder.
#

#
=====
# Component Errors
#
=====
javax.faces.component.UIInput.CONVERSION={0}:
Conversion error occurred.
javax.faces.component.UIInput.REQUIRED={0} is required.
javax.faces.component.UIInput.UPDATE={0}: An error
occurred when processing your submitted information.
javax.faces.component.UISelectOne.INVALID={0} is not
valid
javax.faces.component.UISelectMany.INVALID={0} is not
valid

#
=====
# Converter Errors
#
=====
javax.faces.converter.BigDecimalConverter.DECIMAL={2}:
"{0}" must be a signed decimal number.
javax.faces.converter.BigDecimalConverter.DECIMAL_detail={2}:
"{0}" must be a signed decimal number consisting of
zero or more digits, that may be followed by a decimal point
and fraction. Example: {1}
javax.faces.converter.BigIntegerConverter.BIGINTEGER={2}:
"{0}" must be a number consisting of one or more digits.
javax.faces.converter.BigIntegerConverter.BIGINTEGER_detail={2}:
"{0}" must be a number consisting of one or more
digits. Example: {1}
javax.faces.converter.BooleanConverter.BOOLEAN={1}: "{0}"
must be 'true' or 'false'.
javax.faces.converter.BooleanConverter.BOOLEAN_detail={1}:
"{0}" must be 'true' or 'false'. Any value other than 'true'
will evaluate to 'false'.
javax.faces.converter.ByteConverter.BYTE={2}: "{0}" must be
a number between 0 and 255.
javax.faces.converter.ByteConverter.BYTE_detail={2}: "{0}"
must be a number between 0 and 255. Example: {1}
javax.faces.converter.CharacterConverter.CHARACTER={1}:
"{0}" must be a valid character.

```

javax.faces.converter.CharacterConverter.CHARACTER_detail={1}: "{0}" must be a valid ASCII character.

javax.faces.converter.DateTimeConverter.DATE="{0}" could not be understood as a date.

javax.faces.converter.DateTimeConverter.DATE_detail="{0}" could not be understood as a date. Example: {1}

javax.faces.converter.DateTimeConverter.TIME="{0}" could not be understood as a time.

javax.faces.converter.DateTimeConverter.TIME_detail={2}: "{0}" could not be understood as a time. Example: {1}

javax.faces.converter.DateTimeConverter.DATETIME="{0}" could not be understood as a date and time.

javax.faces.converter.DateTimeConverter.DATETIME_detail={2}: "{0}" could not be understood as a date and time. Example: {1}

javax.faces.converter.DateTimeConverter.PATTERN_TYPE={1}: A 'pattern' or 'type' attribute must be specified to convert the value "{0}".

javax.faces.converter.DoubleConverter.DOUBLE={2}: "{0}" must be a number consisting of one or more digits.

javax.faces.converter.DoubleConverter.DOUBLE_detail={2}: "{0}" must be a number between 4.9E-324 and 1.7976931348623157E308 Example: {1}

javax.faces.converter.EnumConverter.ENUM={2}: "{0}" must be convertible to an enum.

javax.faces.converter.EnumConverter.ENUM_detail={2}: "{0}" must be convertible to an enum from the enum that contains the constant "{1}".

javax.faces.converter.EnumConverter.ENUM_NO_CLASS={1}: "{0}" must be convertible to an enum from the enum, but no enum class provided.

javax.faces.converter.EnumConverter.ENUM_NO_CLASS_detail={1}: "{0}" must be convertible to an enum from the enum, but no enum class provided.

javax.faces.converter.FloatConverter.FLOAT={2}: "{0}" must be a number consisting of one or more digits.

javax.faces.converter.FloatConverter.FLOAT_detail={2}: "{0}" must be a number between 1.4E-45 and 3.4028235E38 Example: {1}

javax.faces.converter.IntegerConverter.INTEGER={2} must be a number.

javax.faces.converter.IntegerConverter.INTEGER_detail={2} must be a number.

javax.faces.converter.LongConverter.LONG={2}: "{0}" must be a number consisting of one or more digits.

javax.faces.converter.LongConverter.LONG_detail={2}: "{0}" must be a number between -9223372036854775808 to 9223372036854775807 Example: {1}

javax.faces.converter.NumberConverter.CURRENCY={2}: "{0}" could not be understood as a currency value.

javax.faces.converter.NumberConverter.CURRENCY_detail={2}: "{0}" could not be understood as a currency value. Example: {1}

javax.faces.converter.NumberConverter.PERCENT={2}: "{0}" could not be understood as a percentage.

javax.faces.converter.NumberConverter.PERCENT_detail={2}: "{0}" could not be understood as a percentage. Example: {1}

javax.faces.converter.NumberConverter.NUMBER={2}: "{0}" is not a number.

javax.faces.converter.NumberConverter.NUMBER_detail={2}: "{0}" is not a number. Example: {1}

javax.faces.converter.NumberConverter.PATTERN={2}: "{0}" is not a number pattern.

javax.faces.converter.NumberConverter.PATTERN_detail={2}: "{0}" is not a number pattern. Example: {1}

javax.faces.converter.ShortConverter.SHORT={2}: "{0}" must be a number consisting of one or more digits.

javax.faces.converter.ShortConverter.SHORT_detail={2}: "{0}" must be a number between -32768 and 32767 Example: {1}

javax.faces.converter.STRING={1}: Could not convert "{0}" to a string.

#

=====

=====

Validator Errors

#

=====

=====

javax.faces.validator.DoubleRangeValidator.MAXIMUM={1}: Validation Error: Value is greater than allowable maximum of "{0}"

javax.faces.validator.DoubleRangeValidator.MINIMUM={1}: Validation Error: Value is less than allowable minimum of "{0}"

javax.faces.validator.DoubleRangeValidator.NOT_IN_RANGE={2}: Validation Error: Specified attribute is not between the expected values of {0} and {1}.

javax.faces.validator.DoubleRangeValidator.TYPE={0}: Validation Error: Value is not of the correct type

javax.faces.validator.LengthValidator.MAXIMUM={1} must be less than {0} characters.

javax.faces.validator.LengthValidator.MINIMUM={1} must be greater than {0} characters.

javax.faces.validator.LongRangeValidator.MAXIMUM={1}: Validation Error: Value is greater than allowable maximum of "{0}"

javax.faces.validator.LongRangeValidator.MINIMUM={1}: Validation Error: Value is less than allowable minimum of "{0}"

javax.faces.validator.LongRangeValidator.NOT_IN_RANGE={2}: Validation Error: Specified attribute is not between the expected values of {0} and {1}.

javax.faces.validator.LongRangeValidator.TYPE={0}: Validation Error: Value is not of the correct type.

javax.faces.validator.NOT_IN_RANGE=Validation Error: Specified attribute is not between the expected values of {0} and {1}.

javax.faces.validator.RegexValidator.PATTERN_NOT_SET=Regex pattern must be set.

javax.faces.validator.RegexValidator.PATTERN_NOT_SET_detail=Regex pattern must be set to non-empty value.

javax.faces.validator.RegexValidator.NOT_MATCHED=Input value is not valid.

javax.faces.validator.RegexValidator.NOT_MATCHED_detail=Input value is not valid.

javax.faces.validator.RegexValidator.MATCH_EXCEPTION=Error in regular expression.

javax.faces.validator.RegexValidator.MATCH_EXCEPTION_detail=Error in regular expression, "{0}"

javax.faces.validator.BeanValidator.MESSAGE={0}

Web Content

/admin/health_center.xhtml

```
<ui:composition
template="/templates/admin_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="admin_content">
<h:form id="registerHC" prependId="false">
<h2>Health Center Details</h2>
<h:panelGrid columns="3">
<h:outputLabel value="Region*:" />
<h:selectOneMenu id="regionMap"
value="#{hCenterBean.regionId}" label="Region"
required="true">
<f:selectItems value="#{hCenterBean.regionMap}" />
<f:ajax listener="#{hCenterBean.changeRegion}"
render="provinceMap cityMap" />
</h:selectOneMenu>
<h:message errorClass="error" for="regionMap" />
<h:outputLabel value="Province*:" />
<h:selectOneMenu id="provinceMap" label="Province"
value="#{hCenterBean.provinceId}" required="true">
<f:selectItem itemLabel="-----" itemValue="" />
<f:selectItems value="#{hCenterBean.provinceMap}"
/>
<f:ajax listener="#{hCenterBean.changeProvince}"
render="cityMap" />
</h:selectOneMenu>
<h:message errorClass="error" for="provinceMap" />
<h:outputLabel value="City*:" />
<h:selectOneMenu id="cityMap" label="City"
value="#{hCenterBean.healthCenter.city_id}"
required="true">
<f:selectItem itemLabel="-----" itemValue="" />
<f:selectItems value="#{hCenterBean.cityMap}" />
</h:selectOneMenu>
<h:message errorClass="error" for="cityMap" />
<h:outputLabel value="Barangay*:" />
<h:inputText id="barangay" label="Barangay"
value="#{hCenterBean.healthCenter.barangay}"
required="true" />
<h:message errorClass="error" for="barangay" />
<h:outputLabel value="Street Address*:" />
<h:inputText id="streetAddress" label="Street
Address"
value="#{hCenterBean.healthCenter.street_address}"
required="true" />
<h:message errorClass="error" for="streetAddress" />
<h:outputLabel value="Health Center Name*:" />
<h:inputText id="hcName" label="Health Center
Name"
value="#{hCenterBean.healthCenter.health_center_name}"
required="true" />
<h:message errorClass="error" for="hcName" />
<h:outputLabel value="Health Center Contact
Number:" />
<h:inputText id="hcContact" label="Contact Number"
value="#{hCenterBean.healthCenter.contact_no}" />
<h:message errorClass="error" for="hcContact" />
</h:panelGrid>
<h:commandButton type="submit" value="Update"
action="#{hCenterBean.update}" />
</ui:define>
</ui:composition>
```

```
<h:commandButton type="reset" value="Reset" />
<h:commandButton type="submit" value="Delete"
action="#{hCenterBean.delete}" immediate="true"
rendered="#{userBean.user.user_type eq 'PA'"
onclick="return confirm('Are you sure you want to delete this
health center?')"/>
</h:form>
</ui:define>
</ui:composition>
```

/admin/health_center_requests.xhtml

```
<ui:composition
template="/templates/admin_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="admin_content">
<f:metadata>
<f:event type="preRenderView"
listener="#{hCenterRequestsBean.initialize}" />
</f:metadata>
<h:form prependId="false">
<h2>Health Center Requests</h2>
<h:outputText value="There are currently no health
center requests." rendered="#{empty
hCenterRequestsBean.healthCenters}" />
<h:dataTable id="health_centers"
value="#{hCenterRequestsBean.healthCenters}"
var="healthCenter" border="1" rendered="#{not
empty hCenterRequestsBean.healthCenters}">
<h:column>
<f:facet name="header">Health Center Name</f:facet>
#{healthCenter.health_center_name}
</h:column>
<h:column>
<f:facet name="header">Street Address</f:facet>
#{healthCenter.street_address}
</h:column>
<h:column>
<f:facet name="header">Barangay</f:facet>
#{healthCenter.barangay}
</h:column>
<h:column>
<f:facet name="header">City</f:facet>
#{healthCenter.city_name}
</h:column>
<h:column>
<f:facet name="header">Province</f:facet>
#{healthCenter.province_name}
</h:column>
<h:column>
<f:facet name="header">Region</f:facet>
#{healthCenter.region_name}
</h:column>
<h:column>
<f:facet name="header">Registration Date</f:facet>
#{healthCenter.reg_date}
</h:column>
</ui:define>
</ui:composition>
```

```

        <h:column>
            <f:facet name="header">Action</f:facet>
            <h:commandLink value="Approve"
action="#{hCenterRequestsBean.approve}">
                <f:param name="hcid"
value="#{healthCenter.health_center_id}" />
            </h:commandLink>
            <h:outputText value="#160; | &#160;" />
            <h:commandLink value="Reject"
action="#{hCenterRequestsBean.reject}">
                <f:param name="hcid"
value="#{healthCenter.health_center_id}" />
            </h:commandLink>
        </h:column>

    </h:dataTable>
    <h:panelGroup
rendered="#{hCenterRequestsBean.paging.totalPages > 0}">
        <h:commandLink value="&lt;&lt; First"

            disabled="#{hCenterRequestsBean.paging.page eq 1}">
                <f:param name="page" value="1"/>
            </h:commandLink>
            <h:commandLink value=" &lt; Prev"

            disabled="#{hCenterRequestsBean.paging.page eq 1}">
                <f:param name="page"
value="#{pageIndex - 1}"/>
            </h:commandLink>
            <ui:repeat var="pageIndex"
value="#{hCenterRequestsBean.paging.pageList}"
varStatus="count">
                <h:commandLink value=" #{pageIndex} "

                    disabled="#{hCenterRequestsBean.paging.page eq
pageIndex}">
                        <f:param name="page"
value="#{pageIndex}"/>
                    </h:commandLink>
                </ui:repeat>
                <h:commandLink value="Next &gt; "

                    disabled="#{hCenterRequestsBean.paging.page eq
hCenterRequestsBean.paging.totalPages}">
                        <f:param name="page"
value="#{pageIndex+1}"/>
                    </h:commandLink>
                <h:commandLink value="End &gt;&gt;"

                    disabled="#{hCenterRequestsBean.paging.page eq
hCenterRequestsBean.paging.totalPages}">
                        <f:param name="page"
value="#{hCenterRequestsBean.paging.totalPages}"/>
                    </h:commandLink>
            </h:panelGroup>
    </h:form>
</ui:define>
</ui:composition>

/admin/manage_health_centers.xhtml
<ui:composition
template="/templates/admin_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facets">
    <ui:define name="admin_content">

```

```

        <f:metadata>
            <f:event type="preRenderView"
listener="#{manageHCBean.initialize}"/>
        </f:metadata>
        <h:form prependId="false">
            <h2>Select Location</h2>
            <h:panelGrid columns="2">
                <h:outputLabel value="Region:" />
                <h:selectOneMenu id="regionMap"
value="#{manageHCBean.regionId}" label="Region"
required="true">
                    <f:selectItems value="#{manageHCBean.regionMap}"
/>
                <f:ajax listener="#{manageHCBean.changeRegion}"
render="provinceMap cityMap health_centers" />
            </h:selectOneMenu>
            <h:outputLabel value="Province:" />
            <h:selectOneMenu id="provinceMap"
value="#{manageHCBean.provinceId}" required="true">
                <f:selectItem itemLabel="-----" itemValue="" />
                <f:selectItems
value="#{manageHCBean.provinceMap}" />
                <f:ajax listener="#{manageHCBean.changeProvince}"
render="cityMap health_centers" />
            </h:selectOneMenu>
            <h:outputLabel value="City:" />
            <h:selectOneMenu id="cityMap"
value="#{manageHCBean.cityId}" required="true">
                <f:selectItem itemLabel="-----" itemValue="" />
                <f:selectItems value="#{manageHCBean.cityMap}" />
                <f:ajax listener="#{manageHCBean.changeCity}"
render="health_center_table" />
            </h:selectOneMenu>
            <h:outputLabel value="Barangay Filter:" />
            <h:panelGroup>
                <h:inputText value="#{manageHCBean.filter}" />
                <h:commandButton value="Apply Filter"
action="#{manageHCBean.filterBarangay}"/>
            </h:panelGroup>
        </h:panelGrid>

        <h:panelGrid id="health_center_table" >
            <h:outputText value="There are no health centers
under the specified location." rendered="#{empty
manageHCBean.healthCenters}"/>
            <h:dataTable id="health_centers"
value="#{manageHCBean.healthCenters}"
var="healthCenter" border="1" rendered="#{not
empty manageHCBean.healthCenters}">

                <h:column>
                    <f:facet
name="header">Barangay</f:facet>
                    #{healthCenter.barangay}
                </h:column>

                <h:column>
                    <f:facet name="header">Health Center
Name</f:facet>
                    <h:link
value="#{healthCenter.health_center_name}"
outcome="edit_hc">
                        <f:param name="hcid"
value="#{healthCenter.health_center_id}" />
                    </h:link>
                </h:column>
            </h:column>

```



```

        </h:column>
        <f:facet name="header">Street
Address</f:facet>
        #{healthCenter.street_address}
        </h:column>

    </h:dataTable>
    <h:panelGroup
rendered="{#{manageHCBean.paging.totalPages > 0}}">
        <h:commandLink value="&lt;&lt; First"
            disabled="{#{manageHCBean.paging.page
eq 1}}">
            <f:param name="page" value="1"/>
        </h:commandLink>
        <h:commandLink value=" &lt; Prev"
            disabled="{#{manageHCBean.paging.page
eq 1}}">
            <f:param name="page"
value="{#{pageIndex - 1}}"/>
        </h:commandLink>
        <ui:repeat var="pageIndex"
value="{#{manageHCBean.paging.pageList}"
varStatus="count">
            <h:commandLink value=" #{pageIndex} "
                disabled="{#{manageHCBean.paging.page
eq pageIndex}}">
                <f:param name="page"
value="{#{pageIndex}}"/>
            </h:commandLink>
            </ui:repeat>
            <h:commandLink value="Next &gt; "
                disabled="{#{manageHCBean.paging.page
eq manageHCBean.paging.totalPages}}">
                <f:param name="page"
value="{#{pageIndex+1}}"/>
            </h:commandLink>
            <h:commandLink value="End &gt;&gt;"
                disabled="{#{manageHCBean.paging.page
eq manageHCBean.paging.totalPages}}">
                <f:param name="page"
value="{#{manageHCBean.paging.totalPages}}"/>
            </h:commandLink>
        </h:panelGroup>
    </h:panelGrid>
</h:form>
</ui:define>
</ui:composition>

```

/admin/manage_users.xhtml

```

<ui:composition
template="/templates/admin_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="admin_content">
    <f:metadata>
        <f:event type="preRenderView"
listener="{#{manageUsersBean.initialize}}"/>
    </f:metadata>
    <h:form prependId="false">
        <h:panelGrid columns="2">
            <h2>Manage Users</h2>
            <h:commandLink id="new_user" value="New User"
action="{#{manageUsersBean.goToNewUser}}">
                <f:param name="action" value="new"/>
            </h:commandLink>

```

```

    </h:panelGrid>
    <h:outputText value="There are no other users under
the barangay health center."
rendered="{#{empty manageUsersBean.users}}"/>
    <h:dataTable id="user_list"
value="{#{manageUsersBean.users}"
var="user" border="1" rendered="{#{not empty
manageUsersBean.users}}">
        <h:column>
            <f:facet name="header">Username</f:facet>
            <h:link value="{#{user.username}"
outcome="user_details">
                <f:param name="uid" value="{#{user.user_id}"/>
            </h:link>
        </h:column>
        <h:column>
            <f:facet name="header">Last Name</f:facet>
            #{user.last_name}
        </h:column>
        <h:column>
            <f:facet name="header">First Name</f:facet>
            #{user.first_name}
        </h:column>
        <h:column>
            <f:facet name="header">Job Title</f:facet>
            #{user.job_title}
        </h:column>
    </h:dataTable>
    <h:panelGroup
rendered="{#{manageUsersBean.paging.totalPages > 0}}">
        <h:commandLink value="&lt;&lt; First"
            disabled="{#{manageUsersBean.paging.page eq 1}}">
            <f:param name="page" value="1"/>
        </h:commandLink>
        <h:commandLink value=" &lt; Prev"
            disabled="{#{manageUsersBean.paging.page eq 1}}">
            <f:param name="page"
value="{#{pageIndex - 1}}"/>
        </h:commandLink>
        <ui:repeat var="pageIndex"
value="{#{manageUsersBean.paging.pageList}"
varStatus="count">
            <h:commandLink value=" #{pageIndex} "
                disabled="{#{manageUsersBean.paging.page eq
pageIndex}}">
                <f:param name="page"
value="{#{pageIndex}}"/>
            </h:commandLink>
            </ui:repeat>
            <h:commandLink value="Next &gt; "
                disabled="{#{manageUsersBean.paging.page eq
manageUsersBean.paging.totalPages}}">
                <f:param name="page"
value="{#{pageIndex+1}}"/>
            </h:commandLink>
            <h:commandLink value="End &gt;&gt;"

```

```

disabled="#{manageUsersBean.paging.page eq
manageUsersBean.paging.totalPages}">
    <f:param name="page"
value="#{manageUsersBean.paging.totalPages}" />
    </h:commandLink>
</h:panelGroup>
</h:form>
<script type="text/javascript">
$(function() {
    $("#new_user").button({ icons: { primary: "ui-icon-
person" } });
});
</script>
</ui:define>
</ui:composition>

```

/admin/user_details.xhtml

```

<ui:composition
template="/templates/admin_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="admin_content">
<h:form prependId="false">
<h2>User Details</h2>
<h:panelGrid columns="2">
<h:outputLabel value="Name*:" />
<h:panelGroup>
<h:panelGrid columns="3">
<h:inputText id="firstName" label="First name"
value="#{userDetailsBean.user.first_name}" required="true"
/>
<h:inputText id="middleName" label="Middle name"
value="#{userDetailsBean.user.middle_name}"
required="true" />
<h:inputText id="lastName" label="Last name"
value="#{userDetailsBean.user.last_name}" required="true"
/>
<h:outputLabel value="First Name" />
<h:outputLabel value="Middle Name" />
<h:outputLabel value="Last Name" />
<h:message errorClass="error" for="firstName" />
<h:message errorClass="error" for="middleName" />
<h:message errorClass="error" for="lastName" />
</h:panelGrid>
</h:panelGroup>
<h:outputLabel value="Job Title*:" />
<h:selectOneMenu
value="#{userDetailsBean.user.job_title}" required="true">
<f:selectItems value="#{userDetailsBean.title_list}"
/>
</h:selectOneMenu>
<h:outputLabel value="Affiliation:" />
<h:inputText
value="#{userDetailsBean.user.affiliation}" />
<h:outputLabel value="Username*:" />
<h:panelGroup>
<h:inputText id="username" label="Username"
value="#{userDetailsBean.user.username}" required="true"
disabled="#{userDetailsBean.user.user_id != -1}">
<f:validateLength minimum="6" maximum="24"/>
</h:inputText>
<h:message errorClass="error" for="username" />
</h:panelGroup>

```

```

<h:outputLabel value="Password*:"
rendered="#{userDetailsBean.user.user_id == -1}" />
<h:panelGroup
rendered="#{userDetailsBean.user.user_id == -1}">
<h:inputSecret id="password" label="Password"
binding="#{passwordComponent}"
value="#{userDetailsBean.user.password}"
required="true">
<f:validateLength minimum="6" maximum="24"/>
</h:inputSecret>
<h:message errorClass="error" for="password" />
</h:panelGroup>
<h:outputLabel value="New Password:"
rendered="#{userDetailsBean.user.user_id != -1}" />
<h:panelGroup
rendered="#{userDetailsBean.user.user_id != -1}">
<h:inputSecret id="new_password" label="New
Password" binding="#{passwordComponent2}"
value="#{userDetailsBean.user.new_password}">
<f:validateLength minimum="6" maximum="24"/>
</h:inputSecret>
<h:message errorClass="error" for="new_password"
/>
</h:panelGroup>
<h:outputLabel value="Repeat Password*:"
rendered="#{userDetailsBean.user.user_id == -1}" />
<h:panelGroup
rendered="#{userDetailsBean.user.user_id == -1}">
<h:inputSecret id="repeatPassword" label="Repeat
Password" required="#{not empty
passwordComponent.value}">
<f:validator
validatorId="confirmPasswordValidator" />
<f:attribute name="passwordComponent"
value="#{passwordComponent}" />
</h:inputSecret>
<h:message errorClass="error" for="repeatPassword"
/>
</h:panelGroup>
<h:outputLabel value="Repeat Password:"
rendered="#{userDetailsBean.user.user_id != -1}" />
<h:panelGroup
rendered="#{userDetailsBean.user.user_id != -1}">
<h:inputSecret id="repeatPassword2" label="Repeat
Password" required="#{not empty
passwordComponent2.value}">
<f:validator
validatorId="confirmPasswordValidator" />
<f:attribute name="passwordComponent"
value="#{passwordComponent2}" />
</h:inputSecret>
<h:message errorClass="error"
for="repeatPassword2" />
</h:panelGroup>
<h:outputLabel value="Email Address:" />
<h:panelGroup>
<h:inputText id="email" label="Email address"
value="#{userDetailsBean.user.email_address}">
<f:validateRegex
pattern="([^\s@]+)(\.[^\s@]+)*@([^\s@]+\.)+([^\s@]+)" />
</h:inputText>
<h:message errorClass="error" for="email" />
</h:panelGroup>
<h:outputLabel value="Contact Number:" />

```

```

    <h:inputText
value="#{userDetailsBean.user.contact_no}"/>
</h:panelGrid>
    <h:commandButton type="submit" value="Add"
action="#{userDetailsBean.add}"
rendered="#{userDetailsBean.user.user_id == -1}"/>
    <h:panelGroup
rendered="#{userDetailsBean.user.user_id != -1}">
    <h:commandButton type="submit" value="Update"
action="#{userDetailsBean.update}"/>
    <h:commandButton type="submit" value="Delete"
action="#{userDetailsBean.delete}" immediate="true"/>
    <h:commandButton type="reset" value="Reset" />
    </h:panelGroup>
</h:form>
</ui:define>
</ui:composition>

```

/components/banner.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<h:graphicImage library="images" name="banner.jpg" />
</ui:composition>

```

/components/common.js.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">

<h:outputStylesheet name="css/humanity/jquery-ui-
1.10.2.custom.css"></h:outputStylesheet>

<h:outputScript library="js" name="jquery-1.9.1.js" />
<h:outputScript library="js" name="jquery-ui-
1.10.2.custom.min.js" />
<h:outputScript library="js" name="jquery.jlabel-1.3.min.js"
/>

<h:outputScript library="js" name="common.js" />

</ui:composition>

```

/components/top_menu.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<h:outputStylesheet library="css"
name="main.css"></h:outputStylesheet>
<ui:include page="banner.xhtml" />
<ui:include page="common.js.xhtml" />
<div class="main_nav">
    <ul>
        <li><h:link outcome="home">Home</h:link></li>
        <li><h:link outcome="register_barangay"
rendered="#{not userBean.loggedIn}">Register
Barangay</h:link></li>
        <li><h:link outcome="patient_list"
rendered="#{(userBean.user.user_type eq 'RU') or
(userBean.user.user_type eq 'HCA')}">Patient
List</h:link></li>
        <li><h:link outcome="new_patient"

```

```

rendered="#{(userBean.user.user_type eq 'RU') or
(userBean.user.user_type eq 'HCA')}">New
Patient</h:link></li>
        <li><h:link outcome="generate_report"
rendered="#{userBean.user.user_type eq 'HCA' or
userBean.user.user_type eq 'PA'}">Generate
Report</h:link></li>
        <li><h:link outcome="manage_hc"
rendered="#{userBean.user.user_type eq
'PA'}">PregSS Administrator</h:link></li>
        <li><h:link outcome="manage_users"
rendered="#{userBean.user.user_type eq
'HCA'}">Health Center Administrator</h:link></li>
        <li><h:link rendered="#{!userBean.loggedIn}"
outcome="login">Login</h:link></li>
        <!-- <li><a href="new_patient.xhtml">New
Patient</a></li>
        <li><a href="checkup.xhtml">Check-up</a></li>
        <li><a href="full_report.xhtml">View Full
Report</a></li> -->
    </ul>
</div>
<div id="top_bar">
    <h:form id="searchPatient" prependId="false"
rendered="#{(userBean.user.user_type eq 'RU') or
(userBean.user.user_type eq 'HCA')}">
        <div id="search_box">
            <h:outputText value="Search Patient:" />
            <h:inputText id="searchLastName"
value="#{searchPatientBean.lastName}" title="Last
Name"/>&#160;
            <h:inputText id="searchFirstName"
value="#{searchPatientBean.firstName}" title="First
Name"/>&#160;
            <h:inputText id="searchMiddleName"
value="#{searchPatientBean.middleName}" title="Middle
Name"/>

            <h:outputText value="Scope:" />
            <h:selectOneMenu
value="#{searchPatientBean.scope}">
                <f:selectItem itemLabel="Local" itemValue="local"
/>

                <f:selectItem itemLabel="All" itemValue="all"/>
            </h:selectOneMenu>
            <h:commandButton type="submit" value="Search"
action="#{searchPatientBean.search}" />
        </div>
    </h:form>
    <h:form prependId="false">
        <h:panelGroup rendered="#{userBean.loggedIn}">
            <h:outputText
value="#{userBean.user.health_center_name} - "
rendered="#{userBean.user.health_center_id ne 0}" />
            You are logged in as <h:link
value="#{userBean.user.username}"
outcome="account_details"/>.
            <h:commandLink action="#{logoutBean.logout}"
value="Logout"/>
        </h:panelGroup>
    </h:form>
</div>
</ui:composition>

```

META-INF/MANIFEST.MF

```

Manifest-Version: 1.0
Class-Path:

```

/record/family_planning.xhtml

```
<ui:composition
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets">

  <div id="family_planning">
    <h2>Family Planning</h2>
    <h:panelGroup rendered="#{empty
  presentPregnancyBean.familyPlans}">
      This patient has no family planning records yet.
    </h:panelGroup>
    <h:panelGroup rendered="#{not empty
  presentPregnancyBean.familyPlans}">
      <h:dataTable id="family_planning_list"
  value="#{presentPregnancyBean.familyPlans}"
  var="famPlan" border="1">

        <h:column>
          <f:facet name="header">Date of Visit</f:facet>
          #{famPlan.visit_date}
        </h:column>

        <h:column>
          <f:facet name="header">Method</f:facet>
          #{famPlan.method}
        </h:column>

        <h:column>
          <f:facet name="header">Quantity Given</f:facet>
          #{famPlan.quantity_given}
        </h:column>

        <h:column>
          <f:facet name="header">Remarks</f:facet>
          #{famPlan.remarks}
        </h:column>

        <h:column>
          <f:facet name="header">Date of Follow-up</f:facet>
          <h:outputText value="#{famPlan.follow_up_date}">
            <f:convertDateTime
  pattern="MM/dd/yyyy" />
          </h:outputText>
        </h:column>

        <h:column>
          <h:link value="Edit" outcome="family_planning"
  styleClass="icon edit_icon">
            <f:param name="pid"
  value="#{patientBean.patient.patient_id}"></f:param>
            <f:param name="fpid"
  value="#{famPlan.family_plan_id}"></f:param>
          </h:link>
        </h:column>

      </h:dataTable>
    </h:panelGroup>
    <h:link value="Add Family Planning"
  outcome="family_planning">
      <f:param name="pid"
  value="#{patientBean.patient.patient_id}" />
      <f:param name="prid"
  value="#{presentPregnancyBean.pregRecord.preg_record_id}" />
    </h:link>
  </div>
```

```
</h:link>
</div>
</ui:composition>
```

/record/labor_delivery.xhtml

```
<ui:composition
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets">
  <div id="labor_delivery">
    <h:panelGroup
  rendered="#{(presentPregnancyBean.laborDel eq null)}">
      There is no labor and delivery information for this
  patient yet.
      <h:link outcome="labor_delivery" value="Add Labor
  and Delivery">
        <f:param name="pid"
  value="#{patientBean.patient.patient_id}" />
        <f:param name="prid"
  value="#{presentPregnancyBean.pregRecord.preg_record_id}" />
      </h:link>
    </h:panelGroup>
    <h:panelGroup
  rendered="#{(presentPregnancyBean.laborDel ne null)}">
      <h:form prependId="false">
        <h2>Labor and Delivery
          <h:link value="[Edit]" outcome="labor_delivery"
  styleClass="icon edit_icon">
            <f:param name="pid"
  value="#{patientBean.patient.patient_id}" />
            <f:param name="prid"
  value="#{presentPregnancyBean.pregRecord.preg_record_id}" />
          </h:link>
        </h2>
        <h:panelGrid columns="2">
          <h:outputLabel value="Immediate Breastfeeding:"
  />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.imm_breastfeed}" />
          <h:outputLabel value="Type of Delivery:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.delivery_type}" />
          <h:outputLabel value="Date of Delivery:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.delivery_date}">
            <f:convertDateTime
  pattern="MM/dd/yyyy" />
          </h:outputText>
          <h:outputLabel value="Place of Delivery:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.delivery_place}" />
          <h:outputLabel value="Birth Weight in Grams:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.birth_weight}" />
          <h:outputLabel value="Post-Partum Hemorrhage
  500 CC+:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.pp_hemorrhage}" />
          <h:outputLabel value="Baby Alive:" />
          <h:outputText
  value="#{presentPregnancyBean.laborDel.baby_alive}" />
          <h:outputLabel value="Baby Healthy:" />
        </h:panelGrid>
      </h:form>
    </h:panelGroup>
  </div>
```

```

        <h:outputText
value="#{presentPregnancyBean.laborDel.baby_healthy}" />
        <h:outputLabel value="Birth Attendant:" />
        <h:outputText
value="#{presentPregnancyBean.laborDel.attendant}" />
        </h:panelGrid>
    </h:form>
</h:panelGroup>
</div>
</ui:composition>

```

/record/postpartum_care.xhtml

```

<ui:composition
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<div id="postpartum_care">
<h:form prependId="false">
<h2>Postpartum Care</h2>
<h:panelGroup rendered="#{empty
presentPregnancyBean.postCares}">
    This patient has no postpartum care records yet.
</h:panelGroup>
<h:panelGroup rendered="#{not empty
presentPregnancyBean.postCares}">
<table border="1">
<tr>
<td>
                <h:panelGroup>
                    <ul
class="floating_list">
<li><h:outputLabel value="Date of Visit" /></li>
<li><h:outputLabel value="Exclusive Breastfeeding"
/></li>
<li><h:outputLabel value="Intends to use Family
Planning" /></li>
<li><h:outputLabel value="Fever 38&#176;C and Above"
/></li>
<li><h:outputLabel value="Foul-smelling Vaginal
Discharge" /></li>
<li><h:outputLabel value="Excessive Vaginal Bleeding"
/></li>
<li><h:outputLabel value="Pallor" /></li>
<li><h:outputLabel value="Cord OK?" /></li>
<li><h:outputLabel value="Vitamin A 200,000 IU" /></li>
<li><h:outputLabel value="Iron/Folate #RX" /></li>
<li><h:outputLabel value="&#160;" /></li>
                    </ul>
                </h:panelGroup>
            </td>
            <td align="center">
                <ul
class="floating_list">

```

```

<li>
<h:outputText value="#{ppCare.visit_date}">
    <f:convertDateTime pattern="MM/dd/yyyy" />
</h:outputText>
</li>
<li><h:outputText
value="#{ppCare.breastfeed_only}" /></li>
<li><h:outputText
value="#{ppCare.will_family_plan}" /></li>
<li><h:outputText value="#{ppCare.fever}" /></li>
<li><h:outputText value="#{ppCare.vaginal_discharge}"
/></li>
<li><h:outputText value="#{ppCare.vaginal_bleeding}"
/></li>
<li><h:outputText value="#{ppCare.pallor}" /></li>
<li><h:outputText value="#{ppCare.cord_ok}" /></li>
<li><h:outputText value="#{ppCare.vitamin_a}" /></li>
<li><h:outputText value="#{ppCare.iron_folate_rx}" /></li>
<li>
<h:link value="Edit" outcome="postpartum_care">
    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
    <f:param name="ppid" value="#{ppCare.pp_care_id}" />
</h:link>
</li>
                </td>
            </td>
        </tr>
    </table>
</h:panelGroup>
<h:link value="Add Postpartum Care"
outcome="postpartum_care">
    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
    <f:param name="prid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}"
/>
</h:link>
</h:form>
</div>
</ui:composition>
</ui:composition xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"

```

```

    xmlns:ui="http://java.sun.com/jsf/facelets">
    <div id="present_pregnancy">
    <h:panelGroup
rendered="#{presentPregnancyBean.pregRecord eq null}">
    There is no pregnancy record for this patient yet.
    <h:link outcome="pregnancy_record" value="Add New
Pregnancy Record">
    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
    </h:link>
    </h:panelGroup>
    <h:panelGroup
rendered="#{presentPregnancyBean.pregRecord ne null}">
    <h2>Present Pregnancy
    <h:link value="[Edit]" outcome="pregnancy_record"
styleClass="icon edit_icon">
    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
    <f:param name="prid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}
" />
    </h:link>
    </h2>
    <h:outputLabel value="LMP:"/>
    <h:outputText
value="#{presentPregnancyBean.pregRecord.lmp}">
    <f:convertDateTime pattern="MM/dd/yyyy" />
    </h:outputText>
    &#160;
    <h:outputLabel value="EDC:"/>
    <h:outputText
value="#{presentPregnancyBean.pregRecord.edc}">
    <f:convertDateTime pattern="MM/dd/yyyy" />
    </h:outputText>
    <table border="1" id="checkup_table">
    <tr>
    <th>TRIMESTER</th>
    <th
colspan="#{fn:length(presentPregnancyBean.firstSem)}"
align="center">1ST</th>
    <th
colspan="#{fn:length(presentPregnancyBean.secondSem)}"
align="center">2ND</th>
    <th
colspan="#{fn:length(presentPregnancyBean.thirdSem)}"
align="center">3RD</th>
    </tr>
    <tr>
    <td>
    <h:panelGroup>
    <ul class="floating_list">
    Months</li>
    Visit</li>
    Bleeding</li>
    Tract Infection</li>
    in Kg.</li>
    Pressure</li>
    140/90 and Above</li>
    38&#176;C and Above</li>

```

```

    </li>Pallor</li>
    </li>Abnormal Fundal Height</li>
    </li>Abnormal Presentation</li>
    </li>Missing
    Fetal Heartbeat</li>
    </li>Edema</li>
    </li>Vaginal
    Infection</li>
    </li><b>ACTION:</b></li>
    </li>Iron/Folate #/RX</li>
    </li>Iodine
    Supplementation</li>
    </li>Malaria
    Prophylaxis</li>
    </li>Intends
    to Breastfeed?</li>
    </li>Advice
    on 4 Danger Signs</li>
    </li>Dental
    Check-up?</li>
    </li>Emergency Plans</li>
    </li>Risk?</li>
    </li>Date of
    Next Visit</li>
    </li>Action</li>
    </ul>
    </h:panelGroup>
    </td>
    <ui:repeat
value="#{presentPregnancyBean.firstSem}" var="checkup">
    <td align="center">
    <ul class="floating_list">
    </li><h:outputText value="#{checkup.aog}"/></li>
    </li>
    <h:outputText value="#{checkup.visit_date}">
    <f:convertDateTime pattern="MM/dd/yy" />
    </h:outputText>
    </li>
    </li><h:outputText
value="#{checkup.vaginal_bleeding}"/></li>
    </li><h:outputText value="#{checkup.uti}" /></li>
    </li><h:outputText value="#{checkup.weight}" /></li>
    </li><h:outputText value="#{checkup.blood_pressure}"
/></li>
    </li><h:outputText value="#{checkup.highBP}" /></li>
    </li><h:outputText value="#{checkup.fever}" /></li>
    </li><h:outputText value="#{checkup.pallor}" /></li>

```

```

    <li><h:outputText value="#{checkup.abn_fundal_height}"
/></li>

    <li><h:outputText value="#{checkup.abn_presentation}"
/></li>

    <li><h:outputText value="#{checkup.missing_heartbeat}"
/></li>

    <li><h:outputText value="#{checkup.edema}" /></li>

    <li><h:outputText value="#{checkup.vaginal_infection}"
/></li>

    <li><h:outputText value="#160;"
rendered="#{checkup.action ne null}" /></li>

    <li><h:outputText
value="#{checkup.action.iron_folate_rx}" /></li>

    <li><h:outputText
value="#{checkup.action.iodine_high_risk}" /></li>

    <li><h:outputText
value="#{checkup.action.malaria_prophylaxis}" /></li>

    <li><h:outputText
value="#{checkup.action.will_breastfeed}" /></li>

    <li><h:outputText
value="#{checkup.action.danger_sign_advice}" /></li>

    <li><h:outputText
value="#{checkup.action.dental_checkup}" /></li>

    <li><h:outputText
value="#{checkup.action.emergency_plan}" /></li>

    <li><h:outputText value="#{checkup.action.risk}" /></li>
    <li>

    <h:outputText value="#{checkup.action.next_visit_date}">

    <f:convertDateTime pattern="MM/dd/yy" />

    </h:outputText>

    </li>
    <li>

    <h:link value="Edit" outcome="action"
rendered="#{checkup ne null}">

    <f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>

    <f:param name="cid"
value="#{checkup.checkup_id}"></f:param>

    </h:link>

    </li>
    </ul>
    </td>
    </ui:repeat>
    <ui:repeat
value="#{presentPregnancyBean.secondSem}"
var="checkup">

```

```

        <td align="center">
            <ul class="floating_list">
                <li><h:outputText value="#{checkup.aog}" /></li>
                <li>

                <h:outputText value="#{checkup.visit_date}">

                <f:convertDateTime pattern="MM/dd/yy" />

                </h:outputText>

                </li>

                <li><h:outputText
value="#{checkup.vaginal_bleeding}" /></li>

                <li><h:outputText value="#{checkup.uti}" /></li>

                <li><h:outputText value="#{checkup.weight}" /></li>

                <li><h:outputText value="#{checkup.blood_pressure}"
/></li>

                <li><h:outputText value="#{checkup.highBP}" /></li>

                <li><h:outputText value="#{checkup.fever}" /></li>

                <li><h:outputText value="#{checkup.pallor}" /></li>

                <li><h:outputText value="#{checkup.abn_fundal_height}"
/></li>

                <li><h:outputText value="#{checkup.abn_presentation}"
/></li>

                <li><h:outputText value="#{checkup.missing_heartbeat}"
/></li>

                <li><h:outputText value="#160;"
rendered="#{checkup.action ne null}" /></li>

                <li><h:outputText
value="#{checkup.action.iron_folate_rx}" /></li>

                <li><h:outputText
value="#{checkup.action.iodine_high_risk}" /></li>

                <li><h:outputText
value="#{checkup.action.malaria_prophylaxis}" /></li>

                <li><h:outputText
value="#{checkup.action.will_breastfeed}" /></li>

                <li><h:outputText
value="#{checkup.action.danger_sign_advice}" /></li>

                <li><h:outputText
value="#{checkup.action.dental_checkup}" /></li>

                <li><h:outputText
value="#{checkup.action.emergency_plan}" /></li>
            </ul>

```

```

<li><h:outputText value="#{checkup.action.risk}" /></li>
</li>

<h:outputText value="#{checkup.action.next_visit_date}">

<f:convertDateTime pattern="MM/dd/yy" />

</h:outputText>

</li>
</li>

<h:link value="Edit" outcome="action"
rendered="#{checkup ne null}">

<f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>

<f:param name="cid"
value="#{checkup.checkup_id}"></f:param>

</h:link>

</li>

</ul>
</td>
</ui:repeat>
</ui:repeat>
value="#{presentPregnancyBean.thirdSem}" var="checkup">
<td align="center">
<ul class="floating_list">

<li><h:outputText value="#{checkup.aog}" /></li>
</li>

<h:outputText value="#{checkup.visit_date}">

<f:convertDateTime pattern="MM/dd/yy" />

</h:outputText>

</li>

<li><h:outputText
value="#{checkup.vaginal_bleeding}" /></li>

<li><h:outputText value="#{checkup.uti}" /></li>

<li><h:outputText value="#{checkup.weight}" /></li>

<li><h:outputText value="#{checkup.blood_pressure}"
/></li>

<li><h:outputText value="#{checkup.highBP}" /></li>

<li><h:outputText value="#{checkup.fever}" /></li>

<li><h:outputText value="#{checkup.pallor}" /></li>

<li><h:outputText value="#{checkup.abn_fundal_height}"
/></li>

<li><h:outputText value="#{checkup.abn_presentation}"
/></li>

<li><h:outputText value="#{checkup.missing_heartbeat}"
/></li>

<li><h:outputText value="#{checkup.edema}" /></li>

```

```

<li><h:outputText value="#{checkup.vaginal_infection}"
/></li>

<li><h:outputText value="#160;"
rendered="#{checkup.action ne null}" /></li>

<li><h:outputText
value="#{checkup.action.iron_folate_rx}" /></li>

<li><h:outputText
value="#{checkup.action.iodine_high_risk}" /></li>

<li><h:outputText
value="#{checkup.action.malaria_prophylaxis}" /></li>

<li><h:outputText
value="#{checkup.action.will_breastfeed}" /></li>

<li><h:outputText
value="#{checkup.action.danger_sign_advice}" /></li>

<li><h:outputText
value="#{checkup.action.dental_checkup}" /></li>

<li><h:outputText
value="#{checkup.action.emergency_plan}" /></li>

<li><h:outputText value="#{checkup.action.risk}" /></li>
</li>

<h:outputText value="#{checkup.action.next_visit_date}">

<f:convertDateTime pattern="MM/dd/yy" />

</h:outputText>

</li>

<h:link value="Edit" outcome="action"
rendered="#{checkup ne null}">

<f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>

<f:param name="cid"
value="#{checkup.checkup_id}"></f:param>

</h:link>

</li>

</ul>
</td>
</tr>
</ui:repeat>
</table>
<h:link value="Perform Pregnancy Check-up"
outcome="pregnancy_checkup">
<f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>
<f:param name="pid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}"
"/>
</h:link>
</h:panelGroup>
</div>
</ui:composition>

```


/record/referrals.xhtml

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:ui="http://java.sun.com/jsf/facelets">
  <div id="referrals">
    <h2>Referrals</h2>
    <h:panelGroup rendered="#{empty
  presentPregnancyBean.referrals}">
      This patient has no referral records yet.
    </h:panelGroup>
    <h:panelGroup rendered="#{not empty
  presentPregnancyBean.referrals}">
      <h:dataTable id="referrals"
  value="#{presentPregnancyBean.referrals}"
  var="ref" binding="#{table}" border="1">

        <h:column>
          <f:facet name="header">#</f:facet>
          #{table.rowIndex + 1}
        </h:column>

        <h:column headerClass="referral_col1">
          <f:facet name="header">Referral</f:facet>
          <h:panelGrid columns="2"
  columnClasses="nowrap" >
            <h:outputLabel value="Referral
  Date:"/>
            <h:outputText
  value="#{ref.referral_date}">
              <f:convertDateTime
  pattern="MM/dd/yyyy" />
            </h:outputText>
            <h:outputLabel
  value="Problem:"/>
            <h:outputText
  value="#{ref.problem}" />
            <h:outputLabel value="Action
  Taken:"/>
            <h:outputText
  value="#{ref.initial_action}" />
            <h:outputLabel
  value="Remarks:"/>
            <h:outputText
  value="#{ref.remarks}" />
            <h:outputLabel
  value="Referred To:"/>
            <h:outputText
  value="#{ref.referred_to}" />
              </h:panelGrid>
            </h:column>

        <h:column headerClass="referral_col2">
          <f:facet name="header">Referral Result</f:facet>
          <h:panelGrid columns="2"
  columnClasses="nowrap">
            <h:outputLabel
  value="Examination Date:"/>
            <h:outputText
  id="examination_date" value="#{ref.examination_date}" >
              <f:convertDateTime
  pattern="MM/dd/yyyy"/>
            </h:outputText>
            <h:outputLabel
  value="Physician:"/>
            <h:outputText
  value="#{ref.physician}" />

```

```

            <h:outputLabel value="Referral
  Center:"/>
            <h:outputText
  value="#{ref.referral_center}" />
            <h:outputLabel
  value="Findings:"/>
            <h:outputText
  value="#{ref.findings}" />
            <h:outputLabel value="Action
  Taken:"/>
            <h:outputText
  value="#{ref.secondary_action}" />
            <h:outputLabel
  value="Instructions:"/>
            <h:outputText
  value="#{ref.instructions}" />
              </h:panelGrid>
            </h:column>

        <h:column>
          <h:link value="Update" outcome="update_referral"
  styleClass="icon edit_icon">
            <f:param name="pid"
  value="#{patientBean.patient.patient_id}" />
            <f:param name="rid"
  value="#{ref.referral_id}" />
            </h:link>
          </h:column>
        </h:dataTable>
      </h:panelGroup>
      <h:link value="Add Referral" outcome="referral">
        <f:param name="pid"
  value="#{patientBean.patient.patient_id}" />
        <f:param name="prid"
  value="#{presentPregnancyBean.pregRecord.preg_record_id}"
  />
      </h:link>
    </div>
  </ui:composition>

```

/resources/css/main.css

```
/*
  Reset CSS
*/
html, body, div, span, applet, object, iframe,
p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
/* table, caption, tbody, tfoot, thead, tr, th, td,*/
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  font: inherit;
  vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {

```

```

display: block;
}
body {
}
ol, ul {
list-style: none;
}
blockquote, q {
quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
content: "";
content: none;
}
table {
border-collapse: collapse;
border-spacing: 0;
}

/* End Reset */

/* Start Global */
* {
font-family: Century Gothic, sans-serif !important; /* <-
fonts */
line-height: 22px;
}
label {
font-weight: bold;
}
b {
font-weight: bold;
}
i {
font-style: italic;
}
body {
background-color: #785D56;
}
h1, h2, h3, h4, h5, h6 {
color: #BE4C54;
}
table {
margin: 5px 0px;
}
td {
vertical-align: baseline;
}
td.center_text {
text-align: center;
}
td.nowrap {
white-space: nowrap;
}
label.textarea {
display: inline-block;
vertical-align: top;
padding-top: 2px;
}
a:link {color:#BE4C54 !important;} /* unvisited link */
a:visited {color:#BE4C54 !important;} /* visited link */
a:hover {color:#785D56 !important;} /* mouse over link */
a:active {color:#785D56 !important;} /* selected link */

.error {
color: #B22222;

```

```

margin-left: 5px;
}
.success {
color: green;
margin-left: 5px;
}

ul.floating_list {
display: table;
float: left;
}
ul.floating_list li {
display: table-row;
}
/* End Global */

/* Main Template */
#wrapper {
width: 960px;
min-height: 100%;
margin-left: auto;
margin-right: auto;
background-color: #FFFFFF;
-moz-box-shadow: 0px 4px 15px #000;
-webkit-box-shadow: 0px 4px 15px #000;
box-shadow: 0px 4px 15px #000;
position: relative;
}

#content {
background-color: #FFF4E3;
padding: 0.5em 1em 5em;
text-align: left;
}

#footer {
background-color: #E6D5C1;
position: absolute;
bottom: 0;
width: 100%;
color: #BE4C54;
}

#messages {
display: block;
clear: both;
}
/* Main Template */

/* Start Main Navigation */
.main_nav {
background-color: #E6D5C1;
border-top: 1px solid #BE4C54;
border-bottom: 1px solid #BE4C54;
width: 100%;
height: 40px;
}
.main_nav ul {
list-style-type: none;
}
.main_nav ul li {
float: left;
}
.main_nav li + li {
border-left: 1px solid #BE4C54;
}
.main_nav ul a {
background-repeat: no-repeat;

```

```
padding-right: 20px;
padding-left: 20px;
display: block;
line-height: 40px;
text-decoration: none;
font-family: Tahoma, Geneva, sans-serif;
font-size: 125%;
color: #BE4C54;
}
```

```
/* End Main Navigation */
```

```
/* Start Top Bar */
```

```
#top_bar{
background-color: #FFF4E3;
width: 100%;
text-align:right;
}
```

```
#search_box{
background-color:#E6D5C1;
display:inline-block;
padding:5px;
border-left:1px solid #BE4C54;
border-bottom:1px solid #BE4C54;
border-bottom-left-radius: 20px;
}
```

```
#searchPatient label {
color: #785D56 !important;
z-index: 0 !important;
}
```

```
/* End Top Bar */
```

```
/*Start General Forms*/
```

```
form input, select, textarea {
font-size:100%;
}
form input[type="text"] {
padding-left:2px;
}
```

```
/*End General Forms*/
```

```
/* Start Home */
```

```
#login_box{
width:270px;
padding:10px;
border:1px solid #BE4C54;
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
margin-left:auto;
margin-right:auto;
margin-top:1em;
}
```

```
#welcome{
padding-left:0.5em;
}
```

```
#welcome p{
text-indent:50px;
}
```

```
#login_button{
display:block;
margin-left:auto;
margin-right:auto;
}
```

```
/* End Patient List */
```

```
/* Start Patient List */
```

```
table#patient_list{ width:100%; }
#patient_list .name_col{ width:40%; }
/* End Patient List */
```

```
/* Start Referrals */
```

```
.referral_col1 { width: 40%; }
.referral_col2 { width: 55%; }
/* End Referrals */
```

```
/* Start Alert Boxes */
```

```
.danger_box{
display:block;
padding:10px;
max-width:550px;
-webkit-border-radius: 5px;
-moz-border-radius: 5px;
border-radius: 5px;
margin-top:10px;
}
```

```
.danger_box ul{
list-style:circle;
list-style-position:inside;
}
```

```
.danger1_box{
border: 1px dashed #FFFF99;
background-color:rgba(255,255,153,0.2);
}
```

```
.danger2_box{
border: 1px dashed #FFCC66;
background-color:rgba(255,204,102,0.2);
}
```

```
.danger3_box{
border: 1px dashed #F75D59;
background-color:rgba(247,93,89,0.1);
}
```

```
/* End Alert Boxes */
```

```
/* Start Patient Record */
```

```
#patient_info td{
vertical-align: top;
}
```

```
#patient_info .col1{
width:55%;
}
```

```
#patient_info .col2{
width:45%;
}
```

```
#checkup_table td{
vertical-align: top;
}
```

```
#checkup_table li{
border-bottom: 1px dashed gray;
}
```

```
#perform_checkup{
margin-left: 5em;
margin-bottom:0.5em;
}
```

```
#new_record{
position:absolute;
margin-bottom:0.5em;
right:1em;
}
```

```
#delete_patient{
```

```

    position:absolute;
    right:1em;
}
#new_patient{
    margin-left:1em;
}

#postpartum_care table li{
    border-bottom: 1px dashed gray;
}

#family_planning_list{
    width:100%;
}
/* End Patient Record */

/* Start Admin */
#admin_panels{
    width:100%;
}

#admin_panels .col1 {width:25%;}
#admin_panels .col2 {width:75%;}

#user_list{
    width:100%;
}
#new_user{
    margin-left:1em;
}
/* End Admin */

/* Start icons */
.icon{
    display:inline-block;
    padding: 24px 0 0 0;
    background-repeat: no-repeat;
    overflow: hidden;
    width:24px;
    height: 0px !important;
    height /**/:24px;
    margin-left: 5px;
}
.edit_icon{
    background-image: url(images/iconza/edit_24x24.png);
}
.add_icon{
    background-image: url(images/iconza/add_24x24.png);
}
.remove_icon{
    background-image: url(images/iconza/remove_24x24.png);
}
.delete_icon{
    background-image: url(images/iconza/delete_24x24.png);
}
.trash_icon{
    background-image: url(images/iconza/trash_24x24.png);
}
/* End icons */

/resources/js/common.js
$(function() {
    $("button, input:submit, input:button,
input:reset").button();

    $("#datepicker").datepicker({
        showOn: "button",

```

```

        buttonImage: "images/calendar.gif",
        buttonImageOnly: true
    });
    $("#datepicker").datepicker('setDate', new Date());

    $("#searchLastName").jLabel();
    $("#searchFirstName").jLabel();
    $("#searchMiddleName").jLabel();
});

```

/resources/js/record.js

```

$(function() {
    $("#record_tabs").tabs();
    $("#perform_checkup").button({ icons: { primary: "ui-icon-
clipboard" } });
    $("#new_record").button({ icons: { primary: "ui-icon-
plusthick" } });
});

```

/templates/admin_template.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:panelGrid columns="2" id="admin_panels"
columnClasses="col1,col2">
        <h:panelGroup>
            <h:form prependId="false">
                <h2>Control Panel</h2>
                <ul>
                    <h:panelGroup
rendered="#{userBean.user.user_type eq 'PA'}">
                        <li><h:link value="Manage
Health Centers" outcome="manage_hc"/></li>
                        <li>
                            <h:link
outcome="hc_requests">
                                Health
Center Requests (<h:outputText
value="#{pAdminBean.num_requests}'"/>)
                            </h:link>
                        </li>
                    </h:panelGroup>
                    <h:panelGroup
rendered="#{userBean.user.user_type eq 'HCA'}">
                        <li><h:link value="Manage
Users" outcome="manage_users"/></li>
                        <li>
                            <h:link
value="Update Health Center Details" outcome="edit_hc">
                                <f:param
name="hcid" value="#{userBean.user.health_center_id}'"/>
                            </h:link>
                        </li>
                    </h:panelGroup>
                </ul>
            </h:form>
        </h:panelGroup>
        <h:panelGroup>
            <ui:insert name="admin_content">Admin
content</ui:insert>
        </h:panelGroup>
    </h:panelGrid>

```

```
</ui:define>
</ui:composition>
```

/templates/main_template.xhtml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<h:head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title><ui:insert name="title">PregSS - Pregnancy
Surveillance System</ui:insert></title>
</h:head>
<h:body>
<div id="wrapper" align="center">
<div id="header">
<ui:include page="/components/top_menu.xhtml" />
</div>
<div id="content">
<div id="messages">
<h:outputText styleClass="error" rendered="#{not
empty errorMsg}" value="#{errorMsg}"/>
<h:outputText styleClass="success" rendered="#{not
empty successMsg}" value="#{successMsg}"/>
</div>
<ui:insert name="content">Default content</ui:insert>
</div>
<div id="footer">
Copyright &#169; 2013 University of the Philippines
Manila. All Rights Reserved.
</div>
</div>
</h:body>
</html>
```

/WEB-INF/faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-facesconfig_2_1.xsd"
version="2.1">
<application>
<message-bundle>
PregssMessages
</message-bundle>
</application>
<managed-bean>
<managed-bean-name>registerHCBean</managed-bean-
name>
<managed-bean-
class>com.pregss.account.management.RegisterHealthCente
rBean</managed-bean-class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<description>The currently logged-in user</description>
```

```
<managed-bean-name>userBean</managed-bean-name>
<managed-bean-
class>com.pregss.user.UserBean</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>loginBean</managed-bean-
name>
<managed-bean-
class>com.pregss.authentication.LoginBean</managed-bean-
class>
<managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>logoutBean</managed-bean-
name>
<managed-bean-
class>com.pregss.authentication.LogoutBean</managed-
bean-class>
<managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>patientBean</managed-bean-
name>
<managed-bean-
class>com.pregss.patient.PatientBean</managed-bean-
class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>pregRecordBean</managed-bean-
name>
<managed-bean-
class>com.pregss.pregnancy.record.PregnancyRecordBean</
managed-bean-class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>pregCheckupBean</managed-
bean-name>
<managed-bean-
class>com.pregss.pregnancy.checkup.PregnancyCheckupBea
n</managed-bean-class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>actionBean</managed-bean-
name>
<managed-bean-
class>com.pregss.action.ActionBean</managed-bean-class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>laborDelBean</managed-bean-
name>
<managed-bean-
class>com.pregss.labor.LaborAndDeliveryBean</managed-
bean-class>
<managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
<managed-bean-name>ppCareBean</managed-bean-
name>
<managed-bean-
class>com.pregss.postpartum.PostpartumCareBean</manag
ed-bean-class>
<managed-bean-scope>view</managed-bean-scope>
```

```

</managed-bean>
<managed-bean>
  <managed-bean-name>famPlanBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.planning.FamilyPlanBean</managed-
bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>refBean</managed-bean-name>
<managed-bean-
class>com.pregss.referral.ReferralBean</managed-bean-
class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>pAdminBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.administrator.PregssAdminBean</manage
d-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>manageHCBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.administrator.ManageHealthCentersBean
</managed-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>hCenterBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.healthcenter.HealthCenterBean</managed
-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-
name>hCenterRequestsBean</managed-bean-name>
  <managed-bean-
class>com.pregss.administrator.HealthCenterRequestsBean
</managed-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>accountDetailsBean</managed-
bean-name>
  <managed-bean-
class>com.pregss.account.management.AccountDetailsBean
</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>manageUsersBean</managed-
bean-name>
  <managed-bean-
class>com.pregss.administrator.ManageUsersBean</manage
d-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>userDetailsBean</managed-
bean-name>

```

```

  <managed-bean-
class>com.pregss.administrator.UserDetailsBean</managed-
bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>patientListBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.patient.PatientListBean</managed-bean-
class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>searchPatientBean</managed-
bean-name>
  <managed-bean-
class>com.pregss.patient.SearchPatientBean</managed-
bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>genReportBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.report.GenerateReportBean</managed-
bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-
name>presentPregnancyBean</managed-bean-name>
  <managed-bean-
class>com.pregss.pregnancy.record.PresentPregnancyBean</
managed-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>obsHistoryBean</managed-bean-
name>
  <managed-bean-
class>com.pregss.obstetrical.ObstetricalHistoryBean</mana
ged-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>healthProbsBean</managed-
bean-name>
  <managed-bean-
class>com.pregss.health.history.HealthProblemsBean</man
aged-bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>transferPatientBean</managed-
bean-name>
  <managed-bean-
class>com.pregss.patient.TransferPatientBean</managed-
bean-class>
  <managed-bean-scope>view</managed-bean-scope>
</managed-bean>
<managed-bean>
  <managed-bean-name>ttBean</managed-bean-name>
  <managed-bean-
class>com.pregss.tetanus_toxoid.TetanusToxoidBean</mana
ged-bean-class>
  <managed-bean-scope>view</managed-bean-scope>

```

```

</managed-bean>

<navigation-rule>
  <from-view-id>*</from-view-id>
  <navigation-case>
    <from-outcome>home</from-outcome>
    <to-view-id>/home.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
  <navigation-case>
    <from-outcome>register_barangay</from-outcome>
    <to-view-id>/register_barangay.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>register_success</from-outcome>
    <to-view-id>/register_success.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>new_patient</from-outcome>
    <to-view-id>/new_patient.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>patient_list</from-outcome>
    <to-view-id>/patient_list.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>search_patient</from-outcome>
    <to-view-id>/search_patient.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>view_patient</from-outcome>
    <to-view-id>/view_patient.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>login</from-outcome>
    <to-view-id>/login.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>account_details</from-outcome>
    <to-view-id>/account_details.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>generate_report</from-outcome>
    <to-view-id>/generate_report.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>manage_hc</from-outcome>
    <to-view-id>/admin/manage_health_centers.xhtml</to-
view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>manage_users</from-outcome>
    <to-view-id>/admin/manage_users.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>edit_hc</from-outcome>
    <to-view-id>/admin/edit_health_center.xhtml</to-view-
id>
  </navigation-case>
  <navigation-case>
    <from-outcome>user_details</from-outcome>
    <to-view-id>/admin/user_details.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>hc_requests</from-outcome>
    <to-view-id>/admin/health_center_requests.xhtml</to-
view-id>

```

```

  </navigation-case>
</navigation-rule>
<navigation-rule>
  <display-name>view_patient.xhtml</display-name>
  <from-view-id>/view_patient.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>update_patient</from-outcome>
    <to-view-id>/update_patient.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>pregnancy_record</from-outcome>
    <to-view-id>/pregnancy_record.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>pregnancy_checkup</from-outcome>
    <to-view-id>/pregnancy_checkup.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>labor_delivery</from-outcome>
    <to-view-id>/labor_delivery.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>postpartum_care</from-outcome>
    <to-view-id>/postpartum_care.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>family_planning</from-outcome>
    <to-view-id>/family_planning.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>referral</from-outcome>
    <to-view-id>/referral.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>tetanus_toxoid</from-outcome>
    <to-view-id>/tetanus_toxoid.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>obstetrical_history</from-outcome>
    <to-view-id>/obstetrical_history.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>health_problems</from-outcome>
    <to-view-id>/health_problems.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>action</from-outcome>
    <to-view-id>/action.xhtml</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>transfer_patient</from-outcome>
    <to-view-id>/transfer_patient.xhtml</to-view-id>
  </navigation-case>
</navigation-rule>
<validator>
  <validator-id>confirmPasswordValidator</validator-id>
  <validator-
class>com.pregss.validator.ConfirmPasswordValidator</vali
dator-class>
  </validator>
</faces-config>

```

/WEB-INF/web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-

```

```

app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
id="WebApp_ID" version="2.5">
  <display-name>pregss</display-name>

  <welcome-file-list>
    <welcome-file>faces/home.xhtml</welcome-file>
  </welcome-file-list>

  <servlet>
    <description></description>
    <display-name>LoginServlet</display-name>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-
class>com.pregss.authentication.LoginServlet</servlet-
class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/login</url-pattern>
  </servlet-mapping>
  <servlet>
    <description></description>
    <display-name>LogoutServlet</display-name>
    <servlet-name>LogoutServlet</servlet-name>
    <servlet-
class>com.pregss.authentication.LogoutServlet</servlet-
class>
  </servlet>
  <servlet-mapping>
    <servlet-name>LogoutServlet</servlet-name>
    <url-pattern>/logout</url-pattern>
  </servlet-mapping>
  <servlet>
    <description></description>
    <display-name>BarangayRegistrationServlet</display-
name>
    <servlet-name>BarangayRegistrationServlet</servlet-
name>
    <servlet-
class>com.pregss.registration.BarangayRegistrationServlet<
/servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>BarangayRegistrationServlet</servlet-
name>
    <url-pattern>/registerBarangay</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-
class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.xhtml</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>

```

```

    <url-pattern>*.faces</url-pattern>
  </servlet-mapping>

  <context-param>
    <description>State saving method: 'client' or 'server'
(=default). See JSF Specification 2.5.2</description>
    <param-
name>javax.faces.STATE_SAVING_METHOD</param-
name>
    <param-value>client</param-value>
  </context-param>
  <context-param>
    <param-
name>javax.servlet.jsp.jstl.fmt.localizationContext</param-
name>
    <param-value>resources.application</param-value>
  </context-param>
  <context-param>
    <param-
name>javax.faces.DATETIMECONVERTER_DEFAULT_TI
MEZONE_IS_SYSTEM_TIMEZONE</param-name>
    <param-value>true</param-value>
  </context-param>
  <listener>
    <listener-
class>com.sun.faces.config.ConfigureListener</listener-
class>
  </listener>
</web-app>

```

/account_details.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
  <ui:define name="content">
    <h:form prependId="false">
      <h1>Account Details</h1>
      <h:panelGrid columns="2">
        <h:outputLabel value="Name: *" />
        <h:panelGroup>
          <h:panelGrid columns="3">
            <h:inputText id="firstName" label="First name"
value="#{accountDetailsBean.user.first_name}"
required="true" />
            <h:inputText id="middleName" label="Middle name"
value="#{accountDetailsBean.user.middle_name}"
required="true" />
            <h:inputText id="lastName" label="Last name"
value="#{accountDetailsBean.user.last_name}"
required="true" />
          </h:panelGrid>
          <h:outputLabel value="First Name" />
          <h:outputLabel value="Middle Name" />
          <h:outputLabel value="Last Name" />
          <h:message errorClass="error" for="firstName" />
          <h:message errorClass="error" for="middleName" />
          <h:message errorClass="error" for="lastName" />
        </h:panelGrid>
      </h:panelGroup>
      <h:outputLabel value="Job Title: *" />
      <h:selectOneMenu
value="#{accountDetailsBean.user.job_title}"
required="true">

```



```

        <f:selectItems
value="#{accountDetailsBean.title_list}" />
        </h:selectOneMenu>
        <h:outputLabel value="Affiliation:" />
        <h:inputText
value="#{accountDetailsBean.user.affiliation}" />
        <h:outputLabel value="Username:" />
        <h:inputText
value="#{accountDetailsBean.user.username}"
disabled="true"/>
        <h:outputLabel value="New Password:" />
        <h:panelGroup>
        <h:inputSecret id="new_password" label="New
Password" binding="#{passwordComponent}"
value="#{accountDetailsBean.user.new_password}">
        <f:validateLength minimum="6" maximum="24"/>
        </h:inputSecret>
        <h:message errorClass="error" for="new_password"
/>
        </h:panelGroup>
        <h:outputLabel value="Repeat New Password:" />
        <h:panelGroup>
        <h:inputSecret id="repeatNewPassword"
label="Repeat New Password" required="#{not empty
passwordComponent.value}">
        <f:validator
validatorId="confirmPasswordValidator" />
        <f:attribute name="passwordComponent"
value="#{passwordComponent}" />
        </h:inputSecret>
        <h:message errorClass="error"
for="repeatNewPassword" />
        </h:panelGroup>
        <h:outputLabel value="Email Address:" />
        <h:panelGroup>
        <h:inputText id="email" label="Email address"
value="#{accountDetailsBean.user.email_address}">
        <f:validateRegex
pattern="([\^.\@]+\)(\.[\^.\@]+)*@([\^.\@]+\.)+([\^.\@]+)" />
        </h:inputText>
        <h:message errorClass="error" for="email" />
        </h:panelGroup>
        <h:outputLabel value="Contact Number:" />
        <h:inputText
value="#{accountDetailsBean.user.contact_no}" />
        </h:panelGrid>
        <h:commandButton type="submit" value="Update"
action="#{accountDetailsBean.update}" />
        </h:form>
</ui:define>
</ui:composition>

</action.xhtml>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{actionBean.patientId ne -1}">
        <f:param name="pid"
value="#{ppCareBean.patientId}" />
        </h:link>
        <h:panelGroup rendered="#{actionBean.dangerList ne
null}">

```

```

        <div class="danger_box
danger#{actionBean.dangerLevel}_box">
        <h:panelGroup rendered="#{actionBean.dangerLevel
== 3}">
                Patient should be immediately referred to the
hospital due to the following symptoms:
        </h:panelGroup>
        <h:panelGroup rendered="#{actionBean.dangerLevel
== 2}">
                Patient should be referred to a physician/RHU, and
hospital delivery
                is recommended due to the following symptoms:
        </h:panelGroup>
        <h:panelGroup rendered="#{actionBean.dangerLevel
== 1}">
                Patient should be referred to a physician/RHU due to
the following symptoms:
        </h:panelGroup>
        <ul>
        <ui:repeat value="#{actionBean.dangerList}"
var="danger">
                <li>#{danger}</li>
        </ui:repeat>
        </ul>
        </div>
</h:panelGroup>
<h:form prependId="false">
        <h2>Pregnancy Checkup</h2>
        <h:panelGrid columns="2"
rendered="#{actionBean.action.checkup_id == -1}">
        <h:outputLabel value="Date of this Visit:" />
        <h:outputText
value="#{actionBean.checkup.visit_date}">
        <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:outputText>
        <h:outputLabel value="Vaginal Bleeding:" />
        <h:outputText
value="#{actionBean.checkup.vaginal_bleeding}" />
        <h:outputLabel value="Urinary Track Infection:" />
        <h:outputText value="#{actionBean.checkup.uti}" />
        <h:outputLabel value="Weight in kg.:" />
        <h:outputText
value="#{actionBean.checkup.weight}" />
        <h:outputLabel value="Blood Pressure:" />
        <h:outputText
value="#{actionBean.checkup.blood_pressure}" />
        <h:outputLabel value="BP 140/90 and Above:" />
        <h:outputText
value="#{actionBean.checkup.highBP}" />
        <h:outputLabel value="Fever 38&#176;C and above:"
/>
        <h:outputText value="#{actionBean.checkup.fever}"
/>
        <h:outputLabel value="Pallor:" />
        <h:outputText value="#{actionBean.checkup.pallor}"
/>
        <h:outputLabel value="Abnormal Fundal Height:" />
        <h:outputText
value="#{actionBean.checkup.abn_fundal_height}" />
        <h:outputLabel value="Abnormal Presentation:" />
        <h:outputText
value="#{actionBean.checkup.abn_presentation}" />
        <h:outputLabel value="Missing Fetal Heartbeat:" />
        <h:outputText
value="#{actionBean.checkup.missing_heartbeat}" />
        <h:outputLabel value="Edema:" />

```

```

        <h:outputText value="#{actionBean.checkup.edema}"
/>
        <h:outputLabel value="Vaginal Infection:" />
        <h:outputText
value="#{actionBean.checkup.vaginal_infection}" />
        </h:panelGrid>
        <h:panelGrid columns="2"
rendered="#{actionBean.action.checkup_id != -1}">
            <h:outputLabel value="Date of this Visit:" />
            <h:panelGroup>
                <h:inputText id="visit_date" label="Visit Date"
value="#{actionBean.checkup.visit_date}" required="true"
size="10">
                    <f:convertDateTime
pattern="MM/dd/yyyy" />
                    </h:inputText>
                    <h:message errorClass="error" for="visit_date" />
                </h:panelGroup>
                <h:outputLabel value="Vaginal Bleeding:" />
                <h:selectOneRadio
value="#{actionBean.checkup.vaginal_bleeding}">
                    <f:selectItem itemLabel="No" itemValue="N"/>
                    <f:selectItem itemLabel="Yes" itemValue="Y"/>
                </h:selectOneRadio>
                <h:outputLabel value="Urinary Track Infection:" />
                <h:selectOneRadio
value="#{actionBean.checkup.uti}">
                    <f:selectItem itemLabel="No" itemValue="N"/>
                    <f:selectItem itemLabel="Yes" itemValue="Y"/>
                </h:selectOneRadio>
                <h:outputLabel value="Weight in kg.:" />
                <h:panelGroup>
                    <h:inputText id="weight" label="Weight"
value="#{actionBean.checkup.weight}" maxLength="3"
size="3" required="true">
                        <h:message errorClass="error" for="weight" />
                    </h:panelGroup>
                    <h:outputLabel value="Blood Pressure:" />
                    <h:panelGroup>
                        <h:inputText id="bp1" label="BP-1"
value="#{actionBean.bp1}" maxLength="3" size="3"
required="true" />
                        <h:inputText id="bp2" label="BP-2"
value="#{actionBean.bp2}" maxLength="3" size="3"
required="true" />
                        <h:message errorClass="error" for="bp1" />
                    </h:panelGroup>
                    <h:message errorClass="error" for="bp2" />
                </h:panelGroup>
                <h:outputLabel value="Fever 38&#176;C and above:"
/>
                    <h:selectOneRadio
value="#{actionBean.checkup.fever}">
                        <f:selectItem itemLabel="No" itemValue="N"/>
                        <f:selectItem itemLabel="Yes" itemValue="Y"/>
                    </h:selectOneRadio>
                    <h:outputLabel value="Pallor:" />
                    <h:selectOneRadio
value="#{actionBean.checkup.pallor}">
                        <f:selectItem itemLabel="No" itemValue="N"/>
                        <f:selectItem itemLabel="Yes" itemValue="Y"/>
                    </h:selectOneRadio>
                    <h:outputLabel value="Abnormal Fundal Height:" />
                    <h:selectOneRadio
value="#{actionBean.checkup.abn_fundal_height}">
                        <f:selectItem itemLabel="No" itemValue="N"/>
                        <f:selectItem itemLabel="Yes" itemValue="Y"/>
                    </h:selectOneRadio>

```

```

        <h:outputLabel value="Abnormal Presentation:" />
        <h:selectOneRadio
value="#{actionBean.checkup.abn_presentation}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Missing Fetal Heartbeat:" />
        <h:selectOneRadio
value="#{actionBean.checkup.missing_heartbeat}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Edema:" />
        <h:selectOneRadio
value="#{actionBean.checkup.edema}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Vaginal Infection:" />
        <h:selectOneRadio
value="#{actionBean.checkup.vaginal_infection}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        </h:panelGrid>
        <h2>Action</h2>
        <h:panelGrid columns="2">
            <h:outputLabel value="Iron/Folate #/RX:" />
            <h:panelGroup>
                <h:inputText id="iron_folate" label="Iron Folate"
value="#{actionBean.action.iron_folate_rx}" size="4"
required="true" />
                <h:outputText value="tablets"/>
                <h:message errorClass="error" for="iron_folate" />
            </h:panelGroup>
            <h:outputLabel value="Iodine Supplementation in
High Risk Areas:" />
            <h:selectOneRadio
value="#{actionBean.action.iodine_high_risk}">
                <f:selectItem itemLabel="No" itemValue="N"/>
                <f:selectItem itemLabel="Yes" itemValue="Y"/>
            </h:selectOneRadio>
            <h:outputLabel value="Malaria Prophylaxis:" />
            <h:selectOneRadio
value="#{actionBean.action.malaria_prophylaxis}">
                <f:selectItem itemLabel="No" itemValue="N"/>
                <f:selectItem itemLabel="Yes" itemValue="Y"/>
            </h:selectOneRadio>
            <h:outputLabel value="Mother intends to breastfeed?"
/>
                <h:selectOneRadio
value="#{actionBean.action.will_breastfeed}">
                    <f:selectItem itemLabel="No" itemValue="N"/>
                    <f:selectItem itemLabel="Yes" itemValue="Y"/>
                </h:selectOneRadio>
                <h:outputLabel value="Advice on 4 Danger Signs:" />
                <h:selectOneRadio
value="#{actionBean.action.danger_sign_advice}">
                    <f:selectItem itemLabel="No" itemValue="N"/>
                    <f:selectItem itemLabel="Yes" itemValue="Y"/>
                </h:selectOneRadio>
                <h:outputLabel value="Dental Check-up?" />
                <h:selectOneRadio
value="#{actionBean.action.dental_checkup}">
                    <f:selectItem itemLabel="No" itemValue="N"/>
                    <f:selectItem itemLabel="Yes" itemValue="Y"/>
                </h:selectOneRadio>

```

```

    <h:outputLabel value="Emergency Plans and Place of
Delivery:" />
    <h:selectOneRadio
value="#{actionBean.action.emergency_plan}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
    </h:selectOneRadio>
    <h:outputLabel value="Risk?" />
    <h:selectOneRadio value="#{actionBean.action.risk}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
    </h:selectOneRadio>
    <h:outputLabel value="Date of Next Visit:" />
    <h:panelGroup>
        <h:inputText id="next_visit" label="Next Visit Date"
value="#{actionBean.action.next_visit_date}" size="10"
required="true">
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:inputText>
        <h:message errorClass="error" for="next_visit" />
    </h:panelGroup>
</h:panelGrid>
    <h:commandButton value="Submit"
action="#{actionBean.addAction}"
rendered="#{(actionBean.action eq null) ||
(actionBean.action.checkup_id == -1)}" />
    <h:panelGroup rendered="#{(actionBean.action ne null)
and (actionBean.action.checkup_id != -1)}">
        <h:commandButton value="Update"
action="#{actionBean.update}" />
        <h:commandButton value="Delete"
action="#{actionBean.delete}" styleClass="delete_button"/>
        <h:commandButton type="reset" value="Reset" />
    </h:panelGroup>
</h:form>
<script type="text/javascript">
$(function() {
    $("#visit_date").datepicker({
        showOn: "button",
        buttonImage: "resources/images/calendar.gif",
        buttonImageOnly: true
    });
    $("#next_visit").datepicker({
        showOn: "button",
        buttonImage: "resources/images/calendar.gif",
        buttonImageOnly: true
    });
});
</script>
</ui:define>
</ui:composition>

<b>/family_planning.xhtml</b>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
    <ui:define name="content">
        <h:form prependId="false">
            <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{famPlanBean.patientId ne -1}">
                <f:param name="pid"
value="#{famPlanBean.patientId}" />
            </h:link>
            <h2>Family Planning</h2>
            <h:panelGrid columns="3">

```

```

                <h:outputLabel value="Date of Visit:" />
                <h:inputText id="visit_date" label="Visit Date"
value="#{famPlanBean.famPlan.visit_date}" size="10"
required="true">
                    <f:convertDateTime pattern="MM/dd/yyyy" />
                </h:inputText>
                <h:message errorClass="error" for="visit_date" />
                <h:outputLabel value="Method:" />
                <h:inputText id="method" label="Method"
value="#{famPlanBean.famPlan.method}" required="true" />
                <h:message errorClass="error" for="method" />
                <h:outputLabel value="Quantity Given:" />
                <h:inputText id="quantity" label="Quantity Given"
value="#{famPlanBean.famPlan.quantity_given}" />
                <h:message errorClass="error" for="quantity" />
                <h:outputLabel value="Remarks:" />
                <h:inputTextarea id="remarks" label="Remarks"
value="#{famPlanBean.famPlan.remarks}" rows="3"
cols="25"/>
                <h:message errorClass="error" for="remarks" />
                <h:outputLabel value="Date of Follow-up:" />
                <h:inputText id="follow_up_date" label="Follow-up
Date" value="#{famPlanBean.famPlan.follow_up_date}"
size="10" required="true">
                    <f:convertDateTime pattern="MM/dd/yyyy" />
                </h:inputText>
                <h:message errorClass="error" for="follow_up_date"
/ >
            </h:panelGrid>
            <h:panelGroup
rendered="#{famPlanBean.famPlan.family_plan_id == -1}">
                <h:commandButton value="Submit"
action="#{famPlanBean.addFamilyPlan}" />
            </h:panelGroup>
            <h:panelGroup
rendered="#{famPlanBean.famPlan.family_plan_id != -1}">
                <h:commandButton value="Update"
action="#{famPlanBean.update}" />
                <h:commandButton value="Delete"
action="#{famPlanBean.delete}" onclick="return confirm('Are
you sure you want to delete this family planning?')"/>
                <h:commandButton type="reset" value="Reset" />
            </h:panelGroup>
        </h:form>
        <script type="text/javascript">
$(function() {
    $("#visit_date").datepicker({
        showOn: "button",
        buttonImage: "resources/images/calendar.gif",
        buttonImageOnly: true
    });
    $("#follow_up_date").datepicker({
        showOn: "button",
        buttonImage: "resources/images/calendar.gif",
        buttonImageOnly: true
    });
});
</script>
</ui:define>
</ui:composition>

<b>/generate_report.xhtml</b>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">

```

```

<ui:define name="content">
  <h:form prependId="false">
    <h1>Generate Report</h1>
    <h2>(HIS)FHSIS Monthly Report</h2>
    <h:panelGrid columns="2">
rendered="#{userBean.user.user_type eq 'PA'}">
      <h:outputLabel value="Region:" />
      <h:selectOneMenu id="regionMap"
value="#{genReportBean.regionId}" label="Region"
required="true">
        <f:selectItems
value="#{genReportBean.regionMap}" />
        <f:ajax
listener="#{genReportBean.changeRegion}"
render="provinceMap cityMap healthCenterMap" />
      </h:selectOneMenu>
      <h:outputLabel value="Province:" />
      <h:selectOneMenu id="provinceMap"
value="#{genReportBean.provinceId}" required="true">
        <f:selectItem itemLabel="All Provinces"
itemValue="0"/>
        <f:selectItems
value="#{genReportBean.provinceMap}" />
        <f:ajax
listener="#{genReportBean.changeProvince}"
render="cityMap healthCenterMap" />
      </h:selectOneMenu>
      <h:outputLabel value="City:" />
      <h:selectOneMenu id="cityMap"
value="#{genReportBean.cityId}" required="true">
        <f:selectItem itemLabel="All Cities"
itemValue="0"/>
        <f:selectItems
value="#{genReportBean.cityMap}" />
        <f:ajax
listener="#{genReportBean.changeCity}"
render="healthCenterMap" />
      </h:selectOneMenu>
      <h:outputLabel value="Barangay:" />
      <h:selectOneMenu id="healthCenterMap"
value="#{genReportBean.hcId}" required="true">
        <f:selectItem itemLabel="All Health
Centers" itemValue="0"/>
        <f:selectItems
value="#{genReportBean.healthCenterMap}" />
      </h:selectOneMenu>
      <h:panelGrid>
      <h:panelGrid columns="2">
      <h:outputLabel value="Month:" />
      <h:selectOneMenu value="#{genReportBean.month}"
required="true">
        <f:selectItems value="#{genReportBean.months}"
/ >
      </h:selectOneMenu>
      <h:outputLabel value="Year:" />
      <h:selectOneMenu value="#{genReportBean.year}"
required="true">
        <f:selectItems value="#{genReportBean.years}" />
      </h:selectOneMenu>
      </h:panelGrid>
      <h:commandButton value="Generate Report"
action="#{genReportBean.getReport}" />
    </h:form>
  </ui:define>
</ui:composition>

```

/health_problems.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
  <ui:define name="content">
    <h:form prependId="false">
      <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{healthProbsBean.patientId ne -1}">
        <f:param name="pid"
value="#{laborDelBean.patientId}" />
      </h:link>
      <h2>Health Problems</h2>
      <h:panelGrid columns="2">
      <h:outputLabel value="Tuberculosis (14 Days+ of
Cough):" />
      <h:selectOneRadio
value="#{healthProbsBean.healthProbs.tuberculosis}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
      </h:selectOneRadio>
      <h:outputLabel value="Heart Disease:" />
      <h:selectOneRadio
value="#{healthProbsBean.healthProbs.heart_disease}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
      </h:selectOneRadio>
      <h:outputLabel value="Diabetes:" />
      <h:selectOneRadio
value="#{healthProbsBean.healthProbs.diabetes}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
      </h:selectOneRadio>
      <h:outputLabel value="Brochial Asthma:" />
      <h:selectOneRadio
value="#{healthProbsBean.healthProbs.bronchial_asthma}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
      </h:selectOneRadio>
      <h:outputLabel value="Goiter:" />
      <h:selectOneRadio
value="#{healthProbsBean.healthProbs.goiter}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
      </h:selectOneRadio>
      </h:panelGrid>
      <h:commandButton value="Update"
action="#{healthProbsBean.update}" />
    </h:form>
  </ui:define>
</ui:composition>

```

/home.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
  <ui:define name="content">
    <h:panelGrid columns="2">
      <h:panelGroup rendered="#{!userBean.loggedIn}">
      <div id="login_box">
      <h:form prependId="false">
      <h:panelGrid columns="2">
      <h:outputLabel value="Username:">

```

```

        <h:inputText id="username"
label="Username" value="#{loginBean.username}"
required="true"/>
        <h:outputLabel value="Password:"/>
        <h:inputSecret id="password"
label="Password" value="#{loginBean.password}"
required="true"/>
        </h:panelGrid>
        <h:messages errorClass="error"/>
        <h:commandButton id="login_button"
type="submit" value="Login" action="#{loginBean.login}"/>
        </h:form>
    </div>
</h:panelGroup>
<h:panelGroup>
<div id="welcome">
    <h1>Welcome to PregSS!</h1>
    <p>PregSS is a clinical decision-support system that
provides information
    and recommends a sound action for the wellness of the
pregnant patient.
    It aims to provide assistance to health care
practitioners in barangay health centers
    and also serves as a surveillance tool to monitor the
current status of maternal health
    in rural areas.</p>
    <h:panelGroup rendered="#{userBean.loggedIn}"/>
    <br />
    <p>
        PregSS is also available in RDF-based Linked Data
form! You can access some parts of the database as RDF
through
        <a
href="http://agila.upm.edu.ph:8083/pregssd2rq/">here</a>.
    </p>
</div>
</h:panelGroup>
</div>
</h:panelGrid>
</ui:define>
</ui:composition>

```

/labor_delivery.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{laborDelBean.patientId ne -1}">
            <f:param name="pid"
value="#{laborDelBean.patientId}"/>
        </h:link>
        <h1>Labor and Delivery</h1>
        <h:panelGrid columns="2">
            <h:outputLabel value="Immediate Breastfeeding:" />
            <h:selectOneRadio
value="#{laborDelBean.laborDel.imm_breastfeed}">
                <f:selectItem itemLabel="No" itemValue="N"/>
                <f:selectItem itemLabel="Yes" itemValue="Y"/>
            </h:selectOneRadio>
            <h:outputLabel value="Type of Delivery:" />
            <h:selectOneMenu
value="#{laborDelBean.laborDel.delivery_type}">

```

```

        <f:selectItems
value="#{laborDelBean.deliveryTypes}"/>
        </h:selectOneMenu>
        <h:outputLabel value="Date of Delivery:" />
        <h:panelGroup>
            <h:inputText id="delivery_date" label="Delivery
Date" value="#{laborDelBean.laborDel.delivery_date}"
size="10" required="true">
                <f:convertDateTime
pattern="MM/dd/yyyy" />
            </h:inputText>
            <h:message errorClass="error" for="delivery_date"
/>
        </h:panelGroup>
        <h:outputLabel value="Place of Delivery:" />
        <h:panelGroup>
            <h:inputText id="delivery_place" label="Place of
Delivery" value="#{laborDelBean.laborDel.delivery_place}"
required="true" />
            <h:message errorClass="error" for="delivery_place"
/>
        </h:panelGroup>
        <h:outputLabel value="Birth Weight in Grams:" />
        <h:panelGroup>
            <h:inputText id="birth_weight" label="Birth
Weight" value="#{laborDelBean.laborDel.birth_weight}"
size="5" required="true" />
            <h:message errorClass="error" for="birth_weight"
/>
        </h:panelGroup>
        <h:outputLabel value="Post-Partum Hemorrhage
500 CC+:" />
        <h:selectOneRadio
value="#{laborDelBean.laborDel.pp_hemorrhage}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Baby Alive:" />
        <h:selectOneRadio
value="#{laborDelBean.laborDel.baby_alive}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Baby Healthy:" />
        <h:selectOneRadio
value="#{laborDelBean.laborDel.baby_healthy}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Birth Attendant:" />
        <h:panelGroup>
            <h:inputText id="attendant" label="Birth
Attendant" value="#{laborDelBean.laborDel.attendant}" />
            <h:message errorClass="error" for="attendant" />
        </h:panelGroup>
        </h:panelGrid>
        <h:commandButton
rendered="#{laborDelBean.laborDel.preg_record_id == -1}"
type="submit" value="Submit"
action="#{laborDelBean.addLaborDel}" />
        <h:panelGroup
rendered="#{laborDelBean.laborDel.preg_record_id != -1}">
            <h:commandButton value="Update"
action="#{laborDelBean.update}" />
            <h:commandButton type="reset" value="Reset" />
        </h:panelGroup>
    </h:form>

```

```

<script type="text/javascript">
$(function() {
    $("#delivery_date").datepicker({
        showOn: "button",
        buttonImage: "resources/images/calendar.gif",
        buttonImageOnly: true
    });
});
</script>
</ui:define>
</ui:composition>

```

/login.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
<h:panelGroup rendered="{!userBean.loggedIn}">
<div id="login_box">
<h:form prependId="false">
<h:panelGrid columns="2">
<h:outputLabel value="Username:"/>
<h:inputText id="username"
label="Username" value="{loginBean.username}"
required="true"/>
<h:outputLabel value="Password:"/>
<h:inputSecret id="password"
label="Password" value="{loginBean.password}"
required="true"/>
</h:panelGrid>
<h:messages errorClass="error"/>
<h:commandButton id="login_button"
type="submit" value="Login" action="{loginBean.login}"/>
</h:form>
</div>
</h:panelGroup>
</ui:define>
</ui:composition>

```

/new_patient.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
<h:form id="newPatientForm" prependId="false">
<h1>New Patient Form</h1>
<div id="patient_info">
<h2>Patient Information</h2>
<h:panelGrid columns="2">
<h:outputLabel value="Name*:" />
<h:panelGrid columns="3">
<h:inputText label="First Name" id="firstName"
value="{patientBean.patient.first_name}" required="true"/>
<h:inputText label="Middle Name"
id="middleName"
value="{patientBean.patient.middle_name}"
required="true"/>
<h:inputText label="Last Name" id="lastName"
value="{patientBean.patient.last_name}" required="true"/>
<h:outputLabel value="First Name" />
<h:outputLabel value="Middle Name" />

```

```

<h:outputLabel value="Last Name" />
<h:message errorClass="error" for="firstName" />
<h:message errorClass="error" for="middleName"
/>
<h:message errorClass="error" for="lastName" />
</h:panelGrid>
<h:outputLabel value="Blood Type:"/>
<h:inputText
value="{patientBean.patient.blood_type}" size="2"
maxlength="2" />
<h:outputLabel value="Family Serial No.:"/>
<h:inputText
value="{patientBean.patient.family_serial_no}"/>
<h:outputLabel value="Age*:"/>
<h:panelGroup>
<h:inputText id="age" label="Age"
value="{patientBean.patient.age}" size="3" maxlength="3"
required="true"/> yr.
<h:message errorClass="error" for="age" />
</h:panelGroup>
<h:outputLabel value="Height*:"/>
<h:panelGroup>
<h:inputText id="height" label="Height"
value="{patientBean.patient.height}" size="3"
maxlength="3" required="true"/> cm.
<h:message errorClass="error" for="height" />
</h:panelGroup>
<h:outputLabel styleClass="textarea"
value="Address:"/>
<h:inputTextarea
value="{patientBean.patient.address}" cols="25" rows="3"
/>
<h:outputLabel value="Contact No.:"/>
<h:inputText
value="{patientBean.patient.contact_no}" />
</h:panelGrid>
</div>
<div id="obstetrical_history">
<h2>Obstetrical History</h2>
<h:panelGrid columns="2">
<h:outputLabel value="Previous Pregnancies*:" />
<h:panelGroup>
<h:inputText label="Previous Pregnancies"
id="prevPregnancies"
value="{patientBean.obsHistory.prev_pregnancies}"
size="2" required="true"/>
<h:message errorClass="error"
for="prevPregnancies" />
</h:panelGroup>
<h:outputLabel value="Previous Caesarean Section:"
/>
<h:selectOneRadio
value="{patientBean.obsHistory.prev_caesarean}">
<f:selectItem itemLabel="No" itemValue="N"/>
<f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="3 Consecutive Miscarriages:"
/>
<h:selectOneRadio
value="{patientBean.obsHistory.cons_miscarriages}">
<f:selectItem itemLabel="No" itemValue="N"/>
<f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Stillbirth:" />
<h:selectOneRadio
value="{patientBean.obsHistory.stillbirth}">
<f:selectItem itemLabel="No" itemValue="N"/>

```

```

        <f:selectItem itemLabel="Yes" itemValue="Y"/>
    </h:selectOneRadio>
    <h:outputLabel value="Post-Partum Hemorrhage:" />
</div>
    <h:selectOneRadio
value="#{patientBean.obsHistory.pp_hemorrhage}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
    </h:selectOneRadio>
</h:panelGrid>
</div>
<div id="health_problems">
    <h2>Health Problems</h2>
    <h:panelGrid columns="2">
        <h:outputLabel value="Tuberculosis (14 Days+ of
Cough):" />
        <h:selectOneRadio
value="#{patientBean.healthProbs.tuberculosis}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Heart Disease:" />
        <h:selectOneRadio
value="#{patientBean.healthProbs.heart_disease}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Diabetes:" />
        <h:selectOneRadio
value="#{patientBean.healthProbs.diabetes}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Brochial Asthma:" />
        <h:selectOneRadio
value="#{patientBean.healthProbs.bronchial_asthma}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Goiter:" />
        <h:selectOneRadio
value="#{patientBean.healthProbs.goiter}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
    </h:panelGrid>
</div>
    <h:commandButton type="submit" value="Add Patient"
action="#{patientBean.addPatient}" />
</h:form>
</ui:define>
</ui:composition>

```

/obstetrical_history.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{obsHistoryBean.patientId ne 1}">
            <f:param name="pid"
value="#{laborDelBean.patientId}" />
        </h:link>

```

```

    <h2>Obstetrical History</h2>
    <h:panelGrid columns="2">
        <h:outputLabel value="Previous Pregnancies:" />
        <h:panelGroup>
            <h:inputText id="prevPregnancies"
label="Previous Pregnancies"
value="#{obsHistoryBean.obsHistory.prev_pregnancies}"
size="2" required="true"/>
            <h:message errorClass="error"
for="prevPregnancies" />
        </h:panelGroup>
        <h:outputLabel value="Previous Caesarean Section:" />
        <h:selectOneRadio
value="#{obsHistoryBean.obsHistory.prev_caesarean}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="3 Consecutive Miscarriages:" />
        <h:selectOneRadio
value="#{obsHistoryBean.obsHistory.cons_miscarriages}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Stillbirth:" />
        <h:selectOneRadio
value="#{obsHistoryBean.obsHistory.stillbirth}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Post-Partum Hemorrhage:" />
        <h:selectOneRadio
value="#{obsHistoryBean.obsHistory.pp_hemorrhage}">
            <f:selectItem itemLabel="No" itemValue="N"/>
            <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
    </h:panelGrid>
    <h:commandButton value="Update"
action="#{obsHistoryBean.update}" />
</h:form>
</ui:define>
</ui:composition>

```

/patient_list.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <f:metadata>
        <f:event type="preRenderView"
listener="#{patientListBean.initialize}" />
    </f:metadata>
    <h:form prependId="false">
        <h:panelGrid columns="2">
            <h:panelGroup>
                <h2>
                    Patient List
                <h:outputText value=" -
#{userBean.user.health_center_name}"
rendered="#{userBean.user.health_center_name ne null}" />
            </h2>
        </h:panelGroup>

```

```

    <h:link id="new_patient" outcome="new_patient"
value="New Patient" />
</h:panelGrid>
    <h:outputText value="There are no patients under this
barangay health center." rendered="{empty
patientListBean.patients}" />
    <h:dataTable id="patient_list"
value="{patientListBean.patients}"
    var="patient" border="1" rendered="{not empty
patientListBean.patients}">

        <h:column headerClass="name_col">
            <f:facet name="header">Name</f:facet>
            <h:link outcome="view_patient">
                #{patient.last_name}, #{patient.first_name}
#{patient.middle_name}
                <f:param name="pid" value="{patient.patient_id}"
            />
        </h:link>

        </h:column>

        <h:column>
            <f:facet name="header">Address</f:facet>
            #{patient.address}
        </h:column>

    </h:dataTable>
    <h:panelGroup
rendered="{patientListBean.paging.totalPages > 0}">
        <h:commandLink value="&lt;&lt; First"
            disabled="{patientListBean.paging.page
eq 1}">
            <f:param name="page" value="1" />
        </h:commandLink>
        <h:commandLink value=" &lt; Prev"
            disabled="{patientListBean.paging.page
eq 1}">
            <f:param name="page"
value="{pageIndex - 1}" />
        </h:commandLink>
        <ui:repeat var="pageIndex"
value="{patientListBean.paging.pageList}"
varStatus="count">
            <h:commandLink value=" #{pageIndex} "
                disabled="{patientListBean.paging.page
eq pageIndex}">
                <f:param name="page"
value="{pageIndex}" />
            </h:commandLink>
        </ui:repeat>
        <h:commandLink value="Next &gt; "
            disabled="{patientListBean.paging.page
eq patientListBean.paging.totalPages}">
            <f:param name="page"
value="{pageIndex+1}" />
        </h:commandLink>
        <h:commandLink value="End &gt;&gt;"
            disabled="{patientListBean.paging.page
eq patientListBean.paging.totalPages}">
            <f:param name="page"
value="{patientListBean.paging.totalPages}" />
        </h:commandLink>
    </h:panelGroup>
</h:form>
<script type="text/javascript">
    $(function() {

```

```

        $("#new_patient").button({ icons: { primary: "ui-icon-
person" } });
    });
</script>
</ui:define>
</ui:composition>

</postpartum_care.xhtml>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
    <ui:define name="content">
        <h:form prependId="false">
            <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="{ppCareBean.patientId ne -1}">
                <f:param name="pid"
value="{ppCareBean.patientId}" />
            </h:link>
            <h:panelGroup rendered="{ppCareBean.dangerList
ne null}">
                <div class="danger_box
danger#{ppCareBean.dangerLevel}_box">
                    <h:panelGroup
rendered="{ppCareBean.dangerLevel == 3}">
                        Patient should be referred to the hospital
due to the following:
                    </h:panelGroup>
                    <h:panelGroup
rendered="{ppCareBean.dangerLevel == 2}">
                        Patient should be referred to a
physician/RHU due to the following:
                    </h:panelGroup>
                    <h:panelGroup
rendered="{ppCareBean.dangerLevel == 1}">
                        Close observation or action is
recommended due to the following:
                    </h:panelGroup>
                    <ul>
                        <ui:repeat
value="{ppCareBean.dangerList}" var="danger">
                            <li>#{danger}</li>
                        </ui:repeat>
                    </ul>
                </div>
            </h:panelGroup>
            <h2>Postpartum Care</h2>
            <h:panelGrid columns="2">
                <h:outputLabel value="Date of Visit:" />
                <h:panelGroup>
                    <h:inputText id="visit_date" label="Visit Date"
value="{ppCareBean.ppCare.visit_date}" size="10"
required="true">
                        <f:convertDateTime
pattern="MM/dd/yyyy" />
                    </h:inputText>
                    <h:message errorClass="error" for="visit_date" />
                </h:panelGroup>
                <h:outputLabel value="Exclusive Breastfeeding:" />
                <h:selectOneRadio
value="{ppCareBean.ppCare.breastfeed_only}">
                    <f:selectItem itemLabel="No" itemValue="N" />
                    <f:selectItem itemLabel="Yes" itemValue="Y" />
                </h:selectOneRadio>
                <h:outputLabel value="Intends to use Family
Planning:" />

```



```

        <h:selectOneRadio
value="#{ppCareBean.ppCare.will_family_plan}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Fever 38&#176;C and
Above:" />
<h:selectOneRadio
value="#{ppCareBean.ppCare.fever}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Foul-smelling Vaginal
Discharge:" />
<h:selectOneRadio
value="#{ppCareBean.ppCare.vaginal_discharge}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Excessive Vaginal Bleeding:"
/>
<h:selectOneRadio
value="#{ppCareBean.ppCare.vaginal_bleeding}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Pallor:" />
<h:selectOneRadio
value="#{ppCareBean.ppCare.pallor}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Cord OK?:" />
<h:selectOneRadio
value="#{ppCareBean.ppCare.cord_ok}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Vitamin A 200,000 IU:" />
<h:selectOneRadio
value="#{ppCareBean.ppCare.vitamin_a}">
    <f:selectItem itemLabel="No" itemValue="N"/>
    <f:selectItem itemLabel="Yes" itemValue="Y"/>
</h:selectOneRadio>
<h:outputLabel value="Iron/Folate #/RX:" />
<h:panelGroup>
    <h:inputText id="iron_folate" label="Iron Folate"
value="#{ppCareBean.ppCare.iron_folate_rx}" size="4"
required="true"/>
    <h:outputText value="tablets"/>
    <h:message errorClass="error" for="iron_folate" />
</h:panelGroup>
</h:panelGrid>
<h:panelGroup
rendered="#{ppCareBean.ppCare.pp_care_id == -1}">
    <h:commandButton value="Submit"
action="#{ppCareBean.addPpCare}"/>
    <h:commandButton type="button" value="Back"
onclick="history.go(-1)"/>
</h:panelGroup>
<h:panelGroup
rendered="#{ppCareBean.ppCare.pp_care_id != -1}">
    <h:commandButton value="Update"
action="#{ppCareBean.update}"/>
    <h:commandButton value="Delete"
action="#{ppCareBean.delete}"/>
    <h:commandButton type="reset" value="Reset" />

```

```

</h:panelGroup>
</h:form>
<script type="text/javascript">
    $(function() {
        $("#visit_date").datepicker({
            showOn: "button",
            buttonImage: "resources/images/calendar.gif",
            buttonImageOnly: true
        });
    });
</script>
</ui:define>
</ui:composition>

/pregnancy_checkup.xhtml
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{pregCheckupBean.patientId ne -1}">
            <f:param name="pid"
value="#{pregCheckupBean.patientId}"/>
        </h:link>
        <h2>Pregnancy Checkup</h2>
        <h:panelGrid columns="2">
            <h:outputLabel value="Date of this Visit:" />
            <h:panelGroup>
                <h:inputText id="visit_date" label="Visit Date"
value="#{pregCheckupBean.pregCheckup.visit_date}"
required="true" size="10">
                    <f:convertDateTime
pattern="MM/dd/yyyy" />
                </h:inputText>
                <h:message errorClass="error" for="visit_date" />
            </h:panelGroup>
            <h:outputLabel value="Vaginal Bleeding:" />
            <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.vaginal_bleeding}"
>
                <f:selectItem itemLabel="No" itemValue="N"/>
                <f:selectItem itemLabel="Yes" itemValue="Y"/>
            </h:selectOneRadio>
            <h:outputLabel value="Urinary Track Infection:" />
            <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.uti}">
                <f:selectItem itemLabel="No" itemValue="N"/>
                <f:selectItem itemLabel="Yes" itemValue="Y"/>
            </h:selectOneRadio>
            <h:outputLabel value="Weight in kg.:" />
            <h:panelGroup>
                <h:inputText id="weight" label="Weight"
value="#{pregCheckupBean.pregCheckup.weight}"
maxlength="3" size="3" required="true"/>
                <h:message errorClass="error" for="weight" />
            </h:panelGroup>
            <h:outputLabel value="Blood Pressure:"
required="true"/>
            <h:panelGroup>
                <h:inputText id="bp1" label="BP-1"
value="#{pregCheckupBean.bp1}" maxlength="3" size="3"
required="true"/>

```

```

        <h:inputText id="bp2" label="BP-2"
value="#{pregCheckupBean.bp2}" maxlength="3" size="3"
required="true"/>
        <h:message errorClass="error" for="bp1" />
<h:message errorClass="error" for="bp2" />
        </h:panelGroup>
        <h:outputLabel value="Fever 38&#176;C and above:"
/>
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.fever}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Pallor:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.pallor}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Abnormal Fundal Height:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.abn_fundal_height
}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Abnormal Presentation:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.abn_presentation}
">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Missing Fetal Heartbeat:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.missing_heartbeat
}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Edema:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.edema}">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        <h:outputLabel value="Vaginal Infection:" />
        <h:selectOneRadio
value="#{pregCheckupBean.pregCheckup.vaginal_infection}
">
        <f:selectItem itemLabel="No" itemValue="N"/>
        <f:selectItem itemLabel="Yes" itemValue="Y"/>
        </h:selectOneRadio>
        </h:panelGrid>
        <h:commandButton value="Submit"
action="#{pregCheckupBean.addCheckup}"/>
</h:form>
<script type="text/javascript">
        $(function() {
                $("#visit_date").datepicker({
                        showOn: "button",
                        buttonImage: "resources/images/calendar.gif",
                        buttonImageOnly: true
                });
        });
</script>
</ui:define>

```

```

</ui:composition>

</pregnancy_record.xhtml>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
        <h:form prependId="false">
                <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{pregRecordBean.patientId ne -1}">
                        <f:param name="pid"
value="#{pregRecordBean.patientId}"/>
                </h:link>
                <h1>Pregnancy Record</h1>
                <h:panelGrid columns="3">
                        <h:outputLabel value="LMP:" />
                        <h:inputText id="lmp" label="LMP"
value="#{pregRecordBean.pregRecord.lmp}" required="true"
size="10">
                                <f:convertDateTime pattern="MM/dd/yyyy" />
                                </h:inputText>
                        <h:message errorClass="error" for="lmp" />
                        <h:outputLabel value="EDC:" />
                        <h:inputText id="edc" label="EDC"
value="#{pregRecordBean.pregRecord.edc}" required="true"
size="10">
                                <f:convertDateTime pattern="MM/dd/yyyy" />
                                </h:inputText>
                        <h:message errorClass="error" for="edc" />
                </h:panelGrid>
                <h:commandButton
rendered="#{pregRecordBean.pregRecord.preg_record_id ==
-1}" value="Add Record"
action="#{pregRecordBean.addRecord}"/>
                <h:panelGroup
rendered="#{pregRecordBean.pregRecord.preg_record_id != -
1}">
                        <h:commandButton value="Update"
action="#{pregRecordBean.update}"/>
                        <h:commandButton type="reset" value="Reset" />
                </h:panelGroup>
                </h:form>
<script type="text/javascript">
        $(function() {
                $("#lmp").datepicker({
                        showOn: "button",
                        buttonImage: "resources/images/calendar.gif",
                        buttonImageOnly: true
                });
                $("#edc").datepicker({
                        showOn: "button",
                        buttonImage: "resources/images/calendar.gif",
                        buttonImageOnly: true
                });
        });
</script>
</ui:define>
</ui:composition>

</referral.xhtml>
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"

```

```

    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
<ui:define name="content">
    <h:form prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="{refBean.patientId ne -1}">
            <f:param name="pid" value="{refBean.patientId}"/>
        </h:link>
        <h2>Referral</h2>
        <h:panelGrid columns="3">
            <h:outputLabel value="Referral Date:"/>
            <h:inputText id="referral_date" label="Referral
Date" value="{refBean.referral.referral_date}" size="10"
required="true">
                <f:convertDateTime pattern="MM/dd/yyyy" />
            </h:inputText>
            <h:message errorClass="error" for="referral_date" />
            <h:outputLabel value="Problem:"/>
            <h:inputText id="problem" label="Problem"
value="{refBean.referral.problem}" required="true" />
            <h:message errorClass="error" for="problem" />
            <h:outputLabel value="Action Taken:"/>
            <h:inputText id="initial_action" label="Action
Taken" value="{refBean.referral.initial_action}"/>
            <h:message errorClass="error" for="initial_action" />
            <h:outputLabel styleClass="textarea"
value="Remarks:"/>
            <h:inputTextarea id="remarks"
value="{refBean.referral.remarks}" rows="3" cols="25"/>
            <h:message errorClass="error" for="remarks" />
            <h:outputLabel value="Referred To:"/>
            <h:inputText id="referred_to" label="Referred To"
value="{refBean.referral.referred_to}" required="true" />
            <h:message errorClass="error" for="referred_to" />
        </h:panelGrid>
        <h:commandButton value="Add Referral"
action="{refBean.addReferral}"/>
    </h:form>
<script type="text/javascript">
    $(function() {
        $("#referral_date").datepicker({
            showOn: "button",
            buttonImage: "resources/images/calendar.gif",
            buttonImageOnly: true
        });
    });
</script>
</ui:define>
</ui:composition>

```

/register_barangay.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form id="registerHC" prependId="false">
        <h1>Register Barangay Health Center</h1>
        <p>Please enter your barangay health center
information and user details. Field names with asterisks(*)
are required.</p>
        <br />
        <h:panelGrid columns="3">
            <h:outputLabel value="Region*:" />

```

```

        <h:selectOneMenu id="regionMap"
value="{registerHCBean.regionId}" label="Region"
required="true">
            <f.selectItems value="{registerHCBean.regionMap}"
/>
            <f:ajax listener="{registerHCBean.changeRegion}"
render="provinceMap cityMap" />
        </h:selectOneMenu>
        <h:message errorClass="error" for="regionMap" />
        <h:outputLabel value="Province*:" />
        <h:selectOneMenu label="Province" id="provinceMap"
value="{registerHCBean.provinceId}" required="true">
            <f.selectItem itemLabel="-----" itemValue="" />
            <f.selectItems
value="{registerHCBean.provinceMap}" />
            <f:ajax listener="{registerHCBean.changeProvince}"
render="cityMap" />
        </h:selectOneMenu>
        <h:message errorClass="error" for="provinceMap" />
        <h:outputLabel value="City*:" />
        <h:selectOneMenu label="City" id="cityMap"
value="{registerHCBean.healthCenter.city_id}"
required="true">
            <f.selectItem itemLabel="-----" itemValue="" />
            <f.selectItems value="{registerHCBean.cityMap}" />
        </h:selectOneMenu>
        <h:message errorClass="error" for="cityMap" />
        <h:outputLabel value="Barangay*:" />
        <h:inputText label="Barangay" id="barangay"
value="{registerHCBean.healthCenter.barangay}"
required="true" />
        <h:message errorClass="error" for="barangay" />
        <h:outputLabel value="Street Address*:" />
        <h:inputText label="Street Address"
id="streetAddress"
value="{registerHCBean.healthCenter.street_address}"
required="true" />
        <h:message errorClass="error" for="streetAddress" />
        <h:outputLabel value="Health Center Name*:" />
        <h:inputText label="Health Center Name"
id="hcName"
value="{registerHCBean.healthCenter.health_center_name}"
required="true" />
        <h:message errorClass="error" for="hcName" />
        <h:outputLabel value="Health Center Contact
Number:" />
        <h:inputText label="Contact Number" id="hcContact"
value="{registerHCBean.healthCenter.contact_no}"/>
        <h:message errorClass="error" for="hcContact" />
    </h:panelGrid>
    <h2>Administrator Account</h2>
    <h:panelGrid columns="2">
        <h:outputLabel value="Name*:" />
    </h:panelGroup>
        <h:panelGrid columns="3">
            <h:inputText id="firstName" label="First name"
value="{registerHCBean.user.first_name}" required="true"
/>
            <h:inputText id="middleName" label="Middle name"
value="{registerHCBean.user.middle_name}"
required="true" />
            <h:inputText id="lastName" label="Last name"
value="{registerHCBean.user.last_name}" required="true"
/>
            <h:outputLabel value="First Name" />
            <h:outputLabel value="Middle Name" />
            <h:outputLabel value="Last Name" />

```

```

    <h:message errorClass="error" for="firstName" />
    <h:message errorClass="error" for="middleName" />
    <h:message errorClass="error" for="lastName" />
    </h:panelGrid>
</h:panelGroup>
<h:outputLabel value="Job Title*:" />
<h:selectOneMenu
value="#{registerHCBBean.user.job_title}" required="true">
    <f:selectItems value="#{registerHCBBean.title_list}" />
</h:selectOneMenu>
<h:outputLabel value="Affiliation:" />
<h:inputText
value="#{registerHCBBean.user.affiliation}" />
<h:outputLabel value="Username*:" />
<h:panelGroup>
    <h:inputText id="username" label="Username"
value="#{registerHCBBean.user.username}" required="true">
        <f:validateLength minimum="6" maximum="24"/>
    </h:inputText>
    <h:message errorClass="error" for="username" />
</h:panelGroup>
<h:outputLabel value="Password*:" />
<h:panelGroup>
    <h:inputSecret id="password" label="Password"
binding="#{passwordComponent}"
value="#{registerHCBBean.user.password}" required="true">
        <f:validateLength minimum="6" maximum="24"/>
    </h:inputSecret>
    <h:message errorClass="error" for="password" />
</h:panelGroup>
<h:outputLabel value="Repeat Password*:" />
<h:panelGroup>
    <h:inputSecret id="repeatPassword" label="Repeat
Password" required="#{not empty
passwordComponent.value}">
        <f:validator
validatorId="confirmPasswordValidator" />
        <f:attribute name="passwordComponent"
value="#{passwordComponent}" />
    </h:inputSecret>
    <h:message errorClass="error" for="repeatPassword"
/>
</h:panelGroup>
<h:outputLabel value="Email Address:" />
<h:panelGroup>
    <h:inputText id="email" label="Email address"
value="#{registerHCBBean.user.email_address}">
        <f:validateRegex
pattern="([\^.@]+\)(\.[\^.@]+)*@([\^.@]+\.[\^.@]+)" />
    </h:inputText>
    <h:message errorClass="error" for="email" />
</h:panelGroup>
<h:outputLabel value="Contact Number:" />
<h:inputText
value="#{registerHCBBean.user.contact_no}" />
</h:panelGrid>
<h:commandButton type="submit" value="Register"
action="#{registerHCBBean.register}" />
</h:form>
</ui:define>
</ui:composition>

```

/register_success.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"

```

```

    xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h2>Registration Successful</h2>
    <p>You have successfully registered your Barangay
Health Center. Please wait until the PregSS Administrator
approves your request.</p>
    <br />
    <h:link value="Back to Home" outcome="home" />
</ui:define>
</ui:composition>

```

/search_patient.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <f:metadata>
        <f:viewParam name="lastName"
value="#{searchPatientBean.lastName}" />
        <f:viewParam name="firstName"
value="#{searchPatientBean.firstName}" />
        <f:viewParam name="middleName"
value="#{searchPatientBean.middleName}" />
        <f:viewParam name="scope"
value="#{searchPatientBean.scope}" />
        <f:event type="preRenderView"
listener="#{searchPatientBean.initialize}" />
    </f:metadata>
    <h:form prependId="false">
        <h2>Search Results</h2>
        <h:outputText value="There are no results matching
your search." rendered="#{empty
searchPatientBean.patients}" />
        <h:dataTable id="user_list"
value="#{searchPatientBean.patients}"
var="patient" border="1" rendered="#{not empty
searchPatientBean.patients}">
            <h:column>
                <f:facet name="header">Name</f:facet>
                <h:link outcome="view_patient">
                    #{patient.last_name}, #{patient.first_name}
                    #{patient.middle_name}
                <f:param name="pid" value="#{patient.patient_id}"
/>
            </h:link>
            </h:column>
            <h:column>
                <f:facet name="header">Address</f:facet>
                #{patient.address}
            </h:column>
            <h:column rendered="#{searchPatientBean.scope eq
'all}">
                <f:facet name="header">Health Center</f:facet>
                #{patient.health_center_name}
            </h:column>
        </h:dataTable>
        <h:panelGroup
rendered="#{searchPatientBean.paging.totalPages > 0}">
            <h:commandLink value="&lt;&lt; First"
disabled="#{searchPatientBean.paging.page eq 1}">

```

```

        <f:param name="page" value="1"/>
    </h:commandLink>
    <h:commandLink value=" &lt; Prev"

    disabled="#{searchPatientBean.paging.page eq 1}">
        <f:param name="page"
value="#{pageIndex - 1}" />
    </h:commandLink>
    <ui:repeat var="pageIndex"
value="#{searchPatientBean.paging.pageList}"
varStatus="count">
        <h:commandLink value=" #{pageIndex} "

        disabled="#{searchPatientBean.paging.page eq
pageIndex}">
            <f:param name="page"
value="#{pageIndex}" />
        </h:commandLink>
        <ui:repeat>
            <h:commandLink value="Next &gt; "

            disabled="#{searchPatientBean.paging.page eq
searchPatientBean.paging.totalPages}">
                <f:param name="page"
value="#{pageIndex+1}" />
            </h:commandLink>
            <h:commandLink value="End &gt;&gt;"

            disabled="#{searchPatientBean.paging.page eq
searchPatientBean.paging.totalPages}">
                <f:param name="page"
value="#{searchPatientBean.paging.totalPages}" />
            </h:commandLink>
        </h:panelGroup>
    </h:form>
</ui:define>
</ui:composition>

/tetanus_toxoid.xhtml
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
    <ui:define name="content">
        <h:form prependId="false">
            <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{ttBean.patientId ne -1}">
                <f:param name="pid" value="#{ttBean.patientId}" />
            </h:link>
            <h2>Tetanus Toxoid</h2>
            <h:dataTable id="vaccines" value="#{ttBean.vaccines}"
var="vaccine"
binding="#{table}" columnClasses="center_text">
                <h:column>
                    <f:facet name="header">Tetanus Toxoid
No.</f:facet>
                    #{table.rowIndex + 1}
                </h:column>

                <h:column>
                    <f:facet name="header">Date Given</f:facet>
                    <h:outputText value="#{vaccine.date_given}"
required="true">
                        <f:convertDateTime
pattern="MM/dd/yyyy" />
                    </h:outputText>

```

```

    </h:column>

    <h:column>
        <h:commandLink value="Delete"
action="#{ttBean.deleteVaccine}" styleClass="icon
delete_icon">
            <f:param name="ttid" value="#{vaccine.tt_id}" />
        </h:commandLink>
    </h:column>
</h:dataTable>
<h:dataTable id="newVaccines"
value="#{ttBean.newVaccines}" var="newVaccine">

    <h:column>
        <h:inputText label="Date Given" id="date"
value="#{newVaccine.date_given}" size="10"
styleClass="datepicker" required="true">
            <f:convertDateTime pattern="MM/dd/yyyy" />
        </h:inputText>
    </h:column>

    <h:column>
        <h:message for="date" />
    </h:column>
</h:dataTable>
<h:panelGrid columns="2">
    <h:commandLink value="Add"
action="#{ttBean.addNewVaccine}" styleClass="icon
add_icon">
        <h:commandLink value="Remove"
action="#{ttBean.removeNewVaccine}" immediate="true"
styleClass="icon remove_icon">
    </h:panelGrid>
    <h:commandButton value="Submit"
action="#{ttBean.addVaccines}" rendered="#{not empty
ttBean.newVaccines}" />
</h:form>
<script type="text/javascript">
    $(function() {
        $(".datepicker").datepicker({
            showOn: "button",
            buttonImage: "resources/images/calendar.gif",
            buttonImageOnly: true
        });
    });
</script>
</ui:define>
</ui:composition>

/transfer_patient.xhtml
<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
    <ui:define name="content">
        <h:form prependId="false">
            <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{transferPatientBean.patient.patient_id
ne -1}">
                <f:param name="pid"
value="#{transferPatientBean.patient.patient_id}" />
            </h:link>
            <h2>Transfer Patient</h2>
            You are about to transfer patient '
            <h:outputText
value="#{transferPatientBean.patient.last_name}" />,&#160;

```

```

    <h:outputText
value="#{transferPatientBean.patient.first_name}"/>&#160;
    <h:outputText
value="#{transferPatientBean.patient.middle_name}"/>'
    to your health center, <i><h:outputText
value="#{userBean.user.health_center_name}"/></i>.<br />
    <h:commandButton value="Continue"
action="#{transferPatientBean.transfer}"/>
</h:form>
</ui:define>
</ui:composition>

```

/update_patient.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form id="updatePatient" prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{patientBean.patient.patient_id ne -1}"/>
        <f:param name="pid"
value="#{patientBean.patient.patient_id}"/>
        </h:link>
        <h2>Personal Information</h2>
        <h:inputHidden
value="#{patientBean.patient.patient_id}"/>
        <h:panelGrid columns="2">
            <h:outputLabel value="Name*:" />
            <h:panelGrid columns="3">
                <h:inputText label="First Name" id="firstName"
value="#{patientBean.patient.first_name}" required="true"/>
                <h:inputText label="Middle Name" id="middleName"
value="#{patientBean.patient.middle_name}"
required="true"/>
                <h:inputText label="Last Name" id="lastName"
value="#{patientBean.patient.last_name}" required="true"/>
            <h:outputLabel value="First Name" />
            <h:outputLabel value="Middle Name" />
            <h:outputLabel value="Last Name" />
            <h:message errorClass="error" for="firstName" />
            <h:message errorClass="error" for="middleName" />
            <h:message errorClass="error" for="lastName" />
        </h:panelGrid>
        <h:outputLabel value="" />
        <h:inputHidden value="#{patientBean.patient.sex}"/>
        <h:outputLabel value="Blood Type:"/>
        <h:inputText
value="#{patientBean.patient.blood_type}" size="2"
maxlength="2" />
        <h:outputLabel value="Family Serial No."/>
        <h:inputText
value="#{patientBean.patient.family_serial_no}"/>
        <h:outputLabel value="Age:"/>
        <h:panelGroup>
            <h:inputText id="age" label="Age"
value="#{patientBean.patient.age}" size="3" maxlength="3"
required="true"/> yr.
            <h:message errorClass="error" for="age" />
        </h:panelGroup>
        <h:outputLabel value="Height:"/>
        <h:panelGroup>
            <h:inputText id="height" label="Height"
value="#{patientBean.patient.height}" size="3"
maxlength="3" required="true"/> cm.
            <h:message errorClass="error" for="height" />

```

```

    </h:panelGroup>
    <h:outputLabel value="Address:"/>
    <h:inputTextarea
value="#{patientBean.patient.address}" cols="25" rows="3"
/>
    <h:outputLabel value="Contact No.:"/>
    <h:inputText
value="#{patientBean.patient.contact_no}"/>
    </h:panelGrid>
    <h:commandButton type="submit" value="Update
Patient" action="#{patientBean.updatePatient}"/>

    </h:form>
</ui:define>
</ui:composition>

```

/update_referral.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets">
<ui:define name="content">
    <h:form prependId="false">
        <h:link outcome="view_patient" value="&lt; Back to
Patient" rendered="#{refBean.patientId ne -1}"/>
        <f:param name="pid" value="#{refBean.patientId}"/>
        </h:link>
        <h2>Referral</h2>
        <h3>Problems identified and action taken</h3>
        <h:panelGrid columns="3">
            <h:outputLabel value="Referral Date:"/>
            <h:inputText id="referral_date" label="Referral Date"
value="#{refBean.referral.referral_date}" size="10"
required="true">
                <f:convertDateTime pattern="MM/dd/yyyy" />
            </h:inputText>
            <h:message errorClass="error" for="referral_date" />
            <h:outputLabel value="Problem:"/>
            <h:inputText id="problem" label="Problem"
value="#{refBean.referral.problem}" required="true" />
            <h:message errorClass="error" for="problem" />
            <h:outputLabel value="Action Taken:"/>
            <h:inputText id="initial_action" label="Action Taken"
value="#{refBean.referral.initial_action}"/>
            <h:message errorClass="error" for="initial_action" />
            <h:outputLabel styleClass="textarea"
value="Remarks:"/>
            <h:inputTextarea id="remarks"
value="#{refBean.referral.remarks}" rows="3" cols="25"/>
            <h:message errorClass="error" for="remarks" />
            <h:outputLabel value="Referred To:"/>
            <h:inputText id="referred_to" label="Referred To"
value="#{refBean.referral.referred_to}" required="true" />
            <h:message errorClass="error" for="referred_to" />
        </h:panelGrid>
        <h3>Results from Referral</h3>
        <h:panelGrid columns="2">
            <h:outputLabel value="Examination Date:"/>
            <h:panelGroup>
                <h:inputText id="examination_date"
label="Examination Date"
value="#{refBean.referral.examination_date}" size="10"
required="true">
                    <f:convertDateTime pattern="MM/dd/yyyy"/>
                </h:inputText>

```

```

        <h:message errorClass="error"
for="examination_date" />
    </h:panelGroup>
    <h:outputLabel value="Physician:"/>
    <h:inputText value="#{refBean.referral.physician}" />
    <h:outputLabel value="Referral Center:"/>
    <h:panelGroup>
    <h:inputText
value="#{refBean.referral.referral_center}" />
    </h:panelGroup>
    <h:outputLabel value="Findings:"/>
    <h:inputText value="#{refBean.referral.findings}" />
    <h:outputLabel value="Action Taken:"/>
    <h:inputText
value="#{refBean.referral.secondary_action}" />
    <h:outputLabel styleClass="textarea"
value="Instructions:"/>
    <h:inputTextarea
value="#{refBean.referral.instructions}" rows="3" cols="25"
/>
    </h:panelGrid>
    <h:commandButton value="Save"
action="#{refBean.update}" />
    <h:commandButton value="Delete"
action="#{refBean.delete}" immediate="true" />
    <h:commandButton type="reset" value="Reset" />
    </h:form>
<script type="text/javascript">
    $(function() {
        $("#referral_date").datepicker({
            showOn: "button",
            buttonImage: "resources/images/calendar.gif",
            buttonImageOnly: true
        });
        $("#examination_date").datepicker({
            showOn: "button",
            buttonImage: "resources/images/calendar.gif",
            buttonImageOnly: true
        });
    });
</script>
</ui:define>
</ui:composition>

```

/view_patient.xhtml

```

<ui:composition template="/templates/main_template.xhtml"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:fn="http://java.sun.com/jsp/jstl/functions">
<ui:define name="content">
    <div id="patient_info">
        <h:form prependId="false">
            <h2>Personal Information
            <h:link outcome="update_patient" value="[Edit]"
styleClass="icon edit_icon">
                <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
            </h:link>
            <h:commandLink id="delete_patient"
value="[Delete]" action="#{patientBean.deletePatient}"
styleClass="icon trash_icon"
onclick="return confirm('Are you sure you want to
delete this patient and all her records?)" />
            </h2>

```

```

    <h:panelGroup
rendered="#{userBean.user.health_center_id ne
patientBean.patient.health_center_id}"
styleClass="error">
        This patient is currently under a different health
center.
        Do you want to transfer the patient?
        <h:link value="Transfer"
outcome="transfer_patient">
            <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
        </h:link>.
    </h:panelGroup>
    <h:inputHidden
value="#{patientBean.patient.patient_id}" />
    <h:panelGrid columns="2" columnClasses="col1,
col2">
        <h:panelGrid columns="2">
            <h:outputLabel value="Name:" />
            <h:panelGroup>
                <h:outputText
value="#{patientBean.patient.last_name}" />,&#160;
                <h:outputText
value="#{patientBean.patient.first_name}" />&#160;
                <h:outputText
value="#{patientBean.patient.middle_name}" />
            </h:panelGroup>
            <h:outputLabel value="Sex:" />
            <h:outputText
value="#{patientBean.patient.sex}" />
            <h:outputLabel value="Blood Type:" />
            <h:outputText
value="#{patientBean.patient.blood_type}" />
            <h:outputLabel value="Family Serial
No." />
            <h:outputText
value="#{patientBean.patient.family_serial_no}" />
            <h:outputLabel value="Age:" />
            <h:outputText
value="#{patientBean.patient.age}" />
            <h:outputLabel value="Height:" />
            <h:outputText
value="#{patientBean.patient.height}" />
            <h:outputLabel value="Address:" />
            <h:outputText
value="#{patientBean.patient.address}" />
            <h:outputLabel value="Contact No.:" />
            <h:outputText
value="#{patientBean.patient.contact_no}" />
        </h:panelGrid>
        <h:panelGroup
rendered="#{patientBean.dangerList ne null}">
            <div class="danger_box
danger#{patientBean.dangerLevel}_box">
                <h:panelGroup
rendered="#{patientBean.dangerLevel == 3}">
                    Hospital delivery and
referral is recommended due to the following:
                </h:panelGroup>
                <h:panelGroup
rendered="#{patientBean.dangerLevel == 2}">
                    Patient should be
referred to a physician/RHU due to the following:
                </h:panelGroup>
                <h:panelGroup
rendered="#{patientBean.dangerLevel == 1}">

```

```

                Close observation or
action is recommended due to the following:
                </h:panelGroup>
                <ul>
                    <ui:repeat
value="#{patientBean.dangerList}" var="danger">
                <li>#{danger}</li>
                    </ui:repeat>
                </ul>
                </div>
            </h:panelGroup>
            <h:panelGroup>
                <h:link id="perform_checkup"
value="Perform Pregnancy Check-up"
outcome="pregnancy_checkup">
                    <f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>
                    <f:param name="prid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}"
"/>
                    </h:link>
                </h:panelGroup>
                <h:link id="new_record" value="Create New
Pregnancy Record" outcome="pregnancy_record"
onclick="return confirm('Creating a new
pregnancy record would override the current one.
Continue?')">
                    <f:param name="pid"
value="#{patientBean.patient.patient_id}"></f:param>
                </h:link>
            </h:panelGrid>
        </h:form>
    </div>
    <div id="record_tabs">
        <ul>
            <li><a href="#patient_history">History</a></li>
            <li><a href="#present_pregnancy">Present
Pregnancy</a></li>
            <h:panelGroup
rendered="#{presentPregnancyBean.pregRecord ne null}">
                <li><a href="#labor_delivery">Labor and
Delivery</a></li>
                <li><a href="#family_planning">Family
Planning</a></li>
                <li><a href="#postpartum_care">Postpartum
Care</a></li>
                <li><a href="#referrals">Referrals</a></li>
            </h:panelGroup>
        </ul>
        <div id="patient_history">
            <h:form prependId="false">
                <h2>Obstetrical History
                    <h:link outcome="obstetrical_history"
value="[Edit]" styleClass="icon edit_icon">
                        <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
                    </h:link>
                </h2>
                <h:panelGrid columns="2">
                    <h:outputLabel value="Previous
Pregnancies:" />
                    <h:outputText
value="#{patientBean.obsHistory.prev_pregnancies}" />
                    <h:outputLabel value="Previous
Caesarean Section:" />

```

```

                <h:outputText
value="#{patientBean.obsHistory.prev_caesarean}" />
                <h:outputLabel value="3 Consecutive
Miscarriages:" />
                <h:outputText
value="#{patientBean.obsHistory.cons_miscarriages}" />
                <h:outputLabel value="Stillbirth:" />
                <h:outputText
value="#{patientBean.obsHistory.stillbirth}" />
                <h:outputLabel value="Post-Partum
Hemorrhage:" />
                <h:outputText
value="#{patientBean.obsHistory.pp_hemorrhage}" />
            </h:panelGrid>
            <h2>Health Problems
                <h:link outcome="health_problems"
value="[Edit]" styleClass="icon edit_icon">
                    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
                </h:link>
            </h2>
            <h:panelGrid columns="2">
                <h:outputLabel value="Tuberculosis (14
Days+ of Cough):" />
                <h:outputText
value="#{patientBean.healthProbs.tuberculosis}" />
                <h:outputLabel value="Heart Disease:"
/>
                <h:outputText
value="#{patientBean.healthProbs.heart_disease}" />
                <h:outputLabel value="Diabetes:" />
                <h:outputText
value="#{patientBean.healthProbs.diabetes}" />
                <h:outputLabel value="Brochial
Asthma:" />
                <h:outputText
value="#{patientBean.healthProbs.bronchial_asthma}" />
                <h:outputLabel value="Goiter:" />
                <h:outputText
value="#{patientBean.healthProbs.goiter}" />
            </h:panelGrid>
            <h:panelGroup
rendered="#{presentPregnancyBean.pregRecord ne null}">
                <h2>Tetanus Toxoid
                    <h:link value="[Edit]"
outcome="tetanus_toxoid" styleClass="icon edit_icon">
                        <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
                    </h:link>
                    <f:param
name="prid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}"
"/>
                </h2>
            </h:link>
            </h2>
            <h:panelGroup rendered="#{empty
presentPregnancyBean.vaccines}">
                <h:outputText value="There is
no vaccine history information for this patient yet. ">
                <h:link value="Add Vaccines"
outcome="tetanus_toxoid">
                    <f:param name="pid"
value="#{patientBean.patient.patient_id}" />
                </h:link>
                <f:param
name="prid"
value="#{presentPregnancyBean.pregRecord.preg_record_id}"
"/>
            </h:link>

```



```

        </h:panelGroup>
        <h:dataTable id="vaccines"
value="#{presentPregnancyBean.vaccines}" var="vaccine"
rendered="#{not empty
presentPregnancyBean.vaccines}" binding="#{tt_table}"
columnClasses="center_text">
        <h:column>
                <f:facet
name="header">Tetanus Toxoid No.</f:facet>

                #{tt_table.rowIndex + 1}
        </h:column>

        <h:column>
                <f:facet
name="header">Date Given</f:facet>

                <h:outputText value="#{vaccine.date_given}"
required="true">

                <f:convertDateTime pattern="MM/dd/yyyy" />

        </h:outputText>
        </h:column>
</h:dataTable>
</h:panelGroup>
</h:form>
</div>

<ui:include src="/record/present_pregnancy.xhtml" />
<h:panelGroup
rendered="#{presentPregnancyBean.pregRecord ne null}">
<ui:include src="/record/labor_delivery.xhtml" />
<ui:include src="/record/family_planning.xhtml" />
<ui:include src="/record/postpartum_care.xhtml" />
<ui:include src="/record/referrals.xhtml" />
</h:panelGroup>
</div>
<h:outputScript library="js" name="record.js" />
</ui:define>
</ui:composition>

```

Mapping File

mapping.ttl

```

@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix d2r: <http://sites.wiwiss.fu-berlin.de/suhl/bizer/d2r-server/config.rdf#> .
@prefix meta: <http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/metadata#> .

```

```

@prefix PregSS: <http://www.pregss.com/owl/pregss.owl#> .
@prefix OpenGALEN:
<http://www.openGALEN.org/owl/openGALEN.owl#> .

```

```

<> a d2r:Server;
rdfs:label "PregSS D2R Server";
d2r:baseURI <http://agila.upm.edu.ph:8083/pregssd2rq/>;
d2r:port 2020;
d2r:vocabularyIncludeInstances true;

d2r:sparqlTimeout 300;
d2r:pageTimeout 5;

meta:datasetTitle "PregSS dataset" ;
meta:datasetSource "PregSS" ;

meta:operatorName "Darryl Ong" ;
.

map:database a d2rq:Database;
d2rq:jdbcDriver "com.mysql.jdbc.Driver";
d2rq:jdbcDSN "jdbc:mysql://localhost/PregnancyMonitor";
d2rq:username "PregnancyMonitor";
d2rq:password "r6dB837nuNXcRZDP";
jdbc:autoReconnect "true";
jdbc:zeroDateTimeBehavior "convertToNull";
.

# Table action
map:action a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern "action/@@@action.checkup_id@@@";
d2rq:class PregSS:Action;
d2rq:classDefinitionLabel "action";
.

map:action__label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property rdfs:label;
d2rq:pattern "action #@@@action.checkup_id@@@";
.

map:action_iron_folate_rx a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property OpenGALEN:IronDrugCompound;
d2rq:propertyDefinitionLabel "action iron_folate_rx";
d2rq:column "action.iron_folate_rx";
d2rq:datatype xsd:integer;
.

map:action_iodine_high_risk a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property OpenGALEN:IodineAsDrug;
d2rq:propertyDefinitionLabel "action iodine_high_risk";
d2rq:column "action.iodine_high_risk";
.

map:action_malaria_prophylaxis a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property OpenGALEN:CholoroquinePhosphate;
d2rq:propertyDefinitionLabel "action
malaria_prophylaxis";
d2rq:column "action.malaria_prophylaxis";
.

map:action_will_breastfeed a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:WillBreastfeed;
d2rq:propertyDefinitionLabel "action will_breastfeed";
d2rq:column "action.will_breastfeed";
.

```

```

map:action_danger_sign_advice a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:AdvisedDangerSigns;
d2rq:propertyDefinitionLabel "action danger_sign_advice";
d2rq:column "action.danger_sign_advice";
.
map:action_dental_checkup a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:HadDentalCheckup;
d2rq:propertyDefinitionLabel "action dental_checkup";
d2rq:column "action.dental_checkup";
.
map:action_emergency_plan a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:EmergencyPlans;
d2rq:propertyDefinitionLabel "action emergency_plan";
d2rq:column "action.emergency_plan";
.
map:action_risk a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:Risking;
d2rq:propertyDefinitionLabel "action risk";
d2rq:column "action.risk";
.
map:action_checkup_id_ref a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:action;
d2rq:property PregSS:checkup_id;
d2rq:refersToClassMap map:pregnancy_checkup;
d2rq:join "action.checkup_id =>
pregnancy_checkup.checkup_id";
.
# Table barangay_health_center
map:barangay_health_center a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern
"barangay_health_center/@@barangay_health_center.health_center_id@@";
d2rq:class PregSS:BarangayHealthCenter;
d2rq:classDefinitionLabel "barangay_health_center";
.
map:barangay_health_center_label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property rdfs:label;
d2rq:pattern "barangay_health_center
#@@barangay_health_center.health_center_id@@";
.
map:barangay_health_center_health_center_id a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS:health_center_id;
d2rq:propertyDefinitionLabel "barangay_health_center
health_center_id";
d2rq:column "barangay_health_center.health_center_id";
d2rq:datatype xsd:integer;
.
map:barangay_health_center_health_center_name a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS:Name;
d2rq:propertyDefinitionLabel "barangay_health_center
health_center_name";
d2rq:column
"barangay_health_center.health_center_name";
.
map:barangay_health_center_contact_no a
d2rq:PropertyBridge;

```

```

d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS>ContactNumber;
d2rq:propertyDefinitionLabel "barangay_health_center
contact_no";
d2rq:column "barangay_health_center.contact_no";
.
map:barangay_health_center_street_address a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS:StreetAddress;
d2rq:propertyDefinitionLabel "barangay_health_center
street_address";
d2rq:column "barangay_health_center.street_address";
.
map:barangay_health_center_barangay a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS:Barangay;
d2rq:propertyDefinitionLabel "barangay_health_center
barangay";
d2rq:column "barangay_health_center.barangay";
.
map:barangay_health_center_city_id a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:barangay_health_center;
d2rq:property PregSS:city_id;
d2rq:propertyDefinitionLabel "barangay_health_center
city_id";
d2rq:column "barangay_health_center.city_id";
d2rq:datatype xsd:integer;
.
# Table city
map:city a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern "city/@@city.city_id@@";
d2rq:class PregSS:City;
d2rq:classDefinitionLabel "city";
.
map:city_label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:city;
d2rq:property rdfs:label;
d2rq:pattern "city #@@city.city_id@@";
.
map:city_city_id a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:city;
d2rq:property PregSS:city_id;
d2rq:propertyDefinitionLabel "city city_id";
d2rq:column "city.city_id";
d2rq:datatype xsd:integer;
.
map:city_name a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:city;
d2rq:property PregSS:Name;
d2rq:propertyDefinitionLabel "city name";
d2rq:column "city.name";
.
map:city_province_id_ref a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:city;
d2rq:property PregSS:province_id;
d2rq:refersToClassMap map:province;
d2rq:join "city.province_id => province.province_id";
.
# Table family_planning
map:family_planning a d2rq:ClassMap;
d2rq:dataStorage map:database;

```

```

d2rq:uriPattern
"family_planning/@@family_planning.family_plan_id@@";
d2rq:class PregSS:FamilyPlanning;
d2rq:classDefinitionLabel "family_planning";
.
map:family_planning__label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property rdfs:label;
d2rq:pattern "family_planning
#@family_planning.family_plan_id@";
.
map:family_planning_family_plan_id a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:family_plan_id;
d2rq:propertyDefinitionLabel "family_planning
family_plan_id";
d2rq:column "family_planning.family_plan_id";
d2rq:datatype xsd:integer;
.
map:family_planning_visit_date a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:VisitDate;
d2rq:propertyDefinitionLabel "family_planning visit_date";
d2rq:column "family_planning.visit_date";
d2rq:datatype xsd:date;
.
map:family_planning_method a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:FamilyPlanningMethod;
d2rq:propertyDefinitionLabel "family_planning method";
d2rq:column "family_planning.method";
.
map:family_planning_quantity_given a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:QuantityGiven;
d2rq:propertyDefinitionLabel "family_planning
quantity_given";
d2rq:column "family_planning.quantity_given";
.
map:family_planning_remarks a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:Remarks;
d2rq:propertyDefinitionLabel "family_planning remarks";
d2rq:column "family_planning.remarks";
.
map:family_planning_preg_record_id_ref a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:family_planning;
d2rq:property PregSS:preg_record_id;
d2rq:refersToClassMap map:pregnancy_record;
d2rq:join "family_planning.preg_record_id =>
pregnancy_record.preg_record_id";
.
# Table health_problems
map:health_problems a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern
"health_problems/@@health_problems.patient_id@";
d2rq:class PregSS:HealthProblems;
d2rq:classDefinitionLabel "health_problems";
.
map:health_problems__label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property rdfs:label;
d2rq:pattern "health_problems
#@health_problems.patient_id@";

```

```

.
map:health_problems_tuberculosis a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property OpenGALEN:Tuberculosis;
d2rq:propertyDefinitionLabel "health_problems
tuberculosis";
d2rq:column "health_problems.tuberculosis";
.
map:health_problems_heart_disease a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property OpenGALEN:HeartDisease;
d2rq:propertyDefinitionLabel "health_problems
heart_disease";
d2rq:column "health_problems.heart_disease";
.
map:health_problems_diabetes a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property OpenGALEN:DiabetesMellitus;
d2rq:propertyDefinitionLabel "health_problems diabetes";
d2rq:column "health_problems.diabetes";
.
map:health_problems_bronchial_asthma a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property OpenGALEN:Asthma;
d2rq:propertyDefinitionLabel "health_problems
bronchial_asthma";
d2rq:column "health_problems.bronchial_asthma";
.
map:health_problems_goiter a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property OpenGALEN:Goiter;
d2rq:propertyDefinitionLabel "health_problems goiter";
d2rq:column "health_problems.goiter";
.
map:health_problems_patient_id_ref a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:health_problems;
d2rq:property PregSS:patient_id;
d2rq:refersToClassMap map:patient;
d2rq:join "health_problems.patient_id =>
patient.patient_id";
.
# Table lab_result
map:lab_result a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern
"lab_result/@@lab_result.lab_result_id@";
d2rq:class PregSS:LabResult;
d2rq:classDefinitionLabel "lab_result";
.
map:lab_result__label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property rdfs:label;
d2rq:pattern "lab_result #@lab_result.lab_result_id@";
.
map:lab_result_lab_result_id a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:lab_result_id;
d2rq:propertyDefinitionLabel "lab_result lab_result_id";
d2rq:column "lab_result.lab_result_id";
d2rq:datatype xsd:integer;
.
map:lab_result_test_type a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:TestType;

```

```

d2rq:propertyDefinitionLabel "lab_result test_type";
d2rq:column "lab_result.test_type";
.
map:lab_result_date a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:ExaminationDate;
d2rq:propertyDefinitionLabel "lab_result date";
d2rq:column "lab_result.date";
d2rq:datatype xsd:date;
.
map:lab_result_test_center a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:TestCenter;
d2rq:propertyDefinitionLabel "lab_result test_center";
d2rq:column "lab_result.test_center";
.
map:lab_result_results a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:LabResults;
d2rq:propertyDefinitionLabel "lab_result results";
d2rq:column "lab_result.results";
.
map:lab_result_checkup_id_ref a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:lab_result;
d2rq:property PregSS:checkup_id;
d2rq:refersToClassMap map:pregnancy_checkup;
d2rq:join "lab_result.checkup_id =>
pregnancy_checkup.checkup_id";
.
# Table labor_and_delivery
map:labor_and_delivery a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern
"labor_and_delivery/@@labor_and_delivery.preg_record_id@
@";
d2rq:class PregSS:LaborAndDelivery;
d2rq:classDefinitionLabel "labor_and_delivery";
.
map:labor_and_delivery_label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property rdfs:label;
d2rq:pattern "labor_and_delivery
#@@labor_and_delivery.preg_record_id@@";
.
map:labor_and_delivery_imm_breastfeed a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property
OpenGALEN:presencewhichisExistenceOfBreastFeeding;
d2rq:propertyDefinitionLabel "labor_and_delivery
imm_breastfeed";
d2rq:column "labor_and_delivery.imm_breastfeed";
.
map:labor_and_delivery_delivery_type a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property OpenGALEN:DeliveryProcess;
d2rq:propertyDefinitionLabel "labor_and_delivery
delivery_type";
d2rq:column "labor_and_delivery.delivery_type";
.
map:labor_and_delivery_delivery_date a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:DeliveryDate;

```

```

d2rq:propertyDefinitionLabel "labor_and_delivery
delivery_date";
d2rq:column "labor_and_delivery.delivery_date";
d2rq:datatype xsd:date;
.
map:labor_and_delivery_delivery_place a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:DeliveryPlace;
d2rq:propertyDefinitionLabel "labor_and_delivery
delivery_place";
d2rq:column "labor_and_delivery.delivery_place";
.
map:labor_and_delivery_birth_weight a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:BirthWeight;
d2rq:propertyDefinitionLabel "labor_and_delivery
birth_weight";
d2rq:column "labor_and_delivery.birth_weight";
d2rq:datatype xsd:integer;
.
map:labor_and_delivery_pp_hemorrhage a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property
OpenGALEN:presencewhichisExistenceOfPostpartumHaem
orrhage;
d2rq:propertyDefinitionLabel "labor_and_delivery
pp_hemorrhage";
d2rq:column "labor_and_delivery.pp_hemorrhage";
.
map:labor_and_delivery_baby_alive a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:BabyAlive;
d2rq:propertyDefinitionLabel "labor_and_delivery
baby_alive";
d2rq:column "labor_and_delivery.baby_alive";
.
map:labor_and_delivery_baby_healthy a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:BabyHealthy;
d2rq:propertyDefinitionLabel "labor_and_delivery
baby_healthy";
d2rq:column "labor_and_delivery.baby_healthy";
.
map:labor_and_delivery_preg_record_id_ref a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:labor_and_delivery;
d2rq:property PregSS:preg_record_id;
d2rq:refersToClassMap map:pregnancy_record;
d2rq:join "labor_and_delivery.preg_record_id =>
pregnancy_record.preg_record_id";
.
# Table obstetrical_history
map:obstetrical_history a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern
"obstetrical_history/@@obstetrical_history.patient_id@@";
d2rq:class PregSS:ObstetricalHistory;
d2rq:classDefinitionLabel "obstetrical_history";
.
map:obstetrical_history_label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:obstetrical_history;
d2rq:property rdfs:label;

```

```

    d2rq:pattern "obstetrical_history
#@@obstetrical_history.patient_id@@";
.
map:obstetrical_history_prev_pregnancies a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property PregSS:PreviousPregnancies;
    d2rq:propertyDefinitionLabel "obstetrical_history
prev_pregnancies";
    d2rq:column "obstetrical_history.prev_pregnancies";
    d2rq:datatype xsd:integer;
.
map:obstetrical_history_prev_caesarean a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property
OpenGALEN:PreviousPersonalHistorywhichisHistoryofCaes
areanSection;
    d2rq:propertyDefinitionLabel "obstetrical_history
prev_caesarean";
    d2rq:column "obstetrical_history.prev_caesarean";
.
map:obstetrical_history_cons_miscarriages a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property PregSS:ConsecutiveMiscarriages;
    d2rq:propertyDefinitionLabel "obstetrical_history
cons_miscarriages";
    d2rq:column "obstetrical_history.cons_miscarriages";
.
map:obstetrical_history_stillbirth a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property PregSS:Stillbirth;
    d2rq:propertyDefinitionLabel "obstetrical_history
stillbirth";
    d2rq:column "obstetrical_history.stillbirth";
.
map:obstetrical_history_pp_hemorrhage a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property PregSS:PostPartumHaemorrhage;
    d2rq:propertyDefinitionLabel "obstetrical_history
pp_hemorrhage";
    d2rq:column "obstetrical_history.pp_hemorrhage";
.
map:obstetrical_history_patient_id_ref a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:obstetrical_history;
    d2rq:property PregSS:patient_id;
    d2rq:refersToClassMap map:patient;
    d2rq:join "obstetrical_history.patient_id =>
patient.patient_id";
.
# Table patient
map:patient a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "patient/@@patient.patient_id@";
    d2rq:class OpenGALEN:Patient;
    d2rq:classDefinitionLabel "patient";
.
map:patient__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:patient;
    d2rq:property rdfs:label;
    d2rq:pattern "patient #@@patient.patient_id@";
.
map:patient_patient_id a d2rq:PropertyBridge;

```

```

    d2rq:belongsToClassMap map:patient;
    d2rq:property PregSS:patient_id;
    d2rq:propertyDefinitionLabel "patient patient_id";
    d2rq:column "patient.patient_id";
    d2rq:datatype xsd:integer;
.
map:patient_health_center_id_ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:patient;
    d2rq:property PregSS:health_center_id;
    d2rq:refersToClassMap map:barangay_health_center;
    d2rq:join "patient.health_center_id =>
barangay_health_center.health_center_id";
.
# Table postpartum_care
map:postpartum_care a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern
"postpartum_care/@@postpartum_care.pp_care_id@";
    d2rq:class PregSS:PostPartumCare;
    d2rq:classDefinitionLabel "postpartum_care";
.
map:postpartum_care__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property rdfs:label;
    d2rq:pattern "postpartum_care
#@@postpartum_care.pp_care_id@";
.
map:postpartum_care_pp_care_id a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property PregSS:pp_care_id;
    d2rq:propertyDefinitionLabel "postpartum_care
pp_care_id";
    d2rq:column "postpartum_care.pp_care_id";
    d2rq:datatype xsd:integer;
.
map:postpartum_care_visit_date a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property PregSS:VisitDate;
    d2rq:propertyDefinitionLabel "postpartum_care
visit_date";
    d2rq:column "postpartum_care.visit_date";
    d2rq:datatype xsd:date;
.
map:postpartum_care_breastfeed_only a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property PregSS:ExclusivelyBreastfeeding;
    d2rq:propertyDefinitionLabel "postpartum_care
breastfeed_only";
    d2rq:column "postpartum_care.breastfeed_only";
.
map:postpartum_care_will_family_plan a
d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property PregSS:IntendsToFamilyPlan;
    d2rq:propertyDefinitionLabel "postpartum_care
will_family_plan";
    d2rq:column "postpartum_care.will_family_plan";
.
map:postpartum_care_fever a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:postpartum_care;
    d2rq:property PregSS:Fever;
    d2rq:propertyDefinitionLabel "postpartum_care fever";
    d2rq:column "postpartum_care.fever";
.

```

```

map:postpartum_care_vaginal_discharge a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:postpartum_care;
  d2rq:property OpenGALEN:VaginalDischargeSymptom;
  d2rq:propertyDefinitionLabel "postpartum_care
vaginal_discharge";
  d2rq:column "postpartum_care.vaginal_discharge";
.
map:postpartum_care_vaginal_bleeding a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:postpartum_care;
  d2rq:property
OpenGALEN:VaginalHaemorrhageSymptom;
  d2rq:propertyDefinitionLabel "postpartum_care
vaginal_bleeding";
  d2rq:column "postpartum_care.vaginal_bleeding";
.
map:postpartum_care_pallor a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:postpartum_care;
d2rq:property OpenGALEN:PallorSymptom;
d2rq:propertyDefinitionLabel "postpartum_care pallor";
d2rq:column "postpartum_care.pallor";
.
map:postpartum_care_cord_ok a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:postpartum_care;
d2rq:property PregSS:CordOK;
d2rq:propertyDefinitionLabel "postpartum_care cord_ok";
d2rq:column "postpartum_care.cord_ok";
.
map:postpartum_care_vitamin_a a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:postpartum_care;
d2rq:property OpenGALEN:VitaminAAsDrug;
d2rq:propertyDefinitionLabel "postpartum_care
vitamin_a";
d2rq:column "postpartum_care.vitamin_a";
.
map:postpartum_care_iron_folate_rx a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:postpartum_care;
d2rq:property OpenGALEN:IronDrugCompound;
d2rq:propertyDefinitionLabel "postpartum_care
iron_folate_rx";
d2rq:column "postpartum_care.iron_folate_rx";
d2rq:datatype xsd:integer;
.
map:postpartum_care_preg_record_id_ref a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:postpartum_care;
  d2rq:property PregSS:preg_record_id;
  d2rq:refersToClassMap map:pregnancy_record;
  d2rq:join "postpartum_care.preg_record_id =>
pregnancy_record.preg_record_id";
.
# Table pregnancy_checkup
map:pregnancy_checkup a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern
"pregnancy_checkup/@@pregnancy_checkup.checkup_id@@";
  d2rq:class PregSS:PregnancyCheckup;
  d2rq:classDefinitionLabel "pregnancy_checkup";
.
map:pregnancy_checkup_label a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property rdfs:label;
d2rq:pattern "pregnancy_checkup
###pregnancy_checkup.checkup_id###";
.

```

```

map:pregnancy_checkup_checkup_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property PregSS:checkup_id;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
checkup_id";
  d2rq:column "pregnancy_checkup.checkup_id";
  d2rq:datatype xsd:integer;
.
map:pregnancy_checkup_visit_date a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property PregSS:VisitDate;
d2rq:propertyDefinitionLabel "pregnancy_checkup
visit_date";
d2rq:column "pregnancy_checkup.visit_date";
d2rq:datatype xsd:date;
.
map:pregnancy_checkup_vaginal_bleeding a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property
OpenGALEN:VaginalHaemorrhageSymptom;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
vaginal_bleeding";
  d2rq:column "pregnancy_checkup.vaginal_bleeding";
.
map:pregnancy_checkup_uti a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property OpenGALEN:UrinaryTractInfection;
d2rq:propertyDefinitionLabel "pregnancy_checkup uti";
d2rq:column "pregnancy_checkup.uti";
.
map:pregnancy_checkup_weight a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property OpenGALEN:BodyWeightSymptom;
d2rq:propertyDefinitionLabel "pregnancy_checkup
weight";
d2rq:column "pregnancy_checkup.weight";
d2rq:datatype xsd:integer;
.
map:pregnancy_checkup_blood_pressure a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property OpenGALEN:CurrentBloodPressureLevel;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
blood_pressure";
  d2rq:column "pregnancy_checkup.blood_pressure";
.
map:pregnancy_checkup_fever a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property OpenGALEN:Fever;
d2rq:propertyDefinitionLabel "pregnancy_checkup fever";
d2rq:column "pregnancy_checkup.fever";
.
map:pregnancy_checkup_pallor a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property OpenGALEN:PallorSymptom;
d2rq:propertyDefinitionLabel "pregnancy_checkup pallor";
d2rq:column "pregnancy_checkup.pallor";
.
map:pregnancy_checkup_abn_fundal_height a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property PregSS:AbnormalFundalHeight;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
abn_fundal_height";
  d2rq:column "pregnancy_checkup.abn_fundal_height";
.

```

```

map:pregnancy_checkup_abn_presentation a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property
OpenGALEN:presencewhichisExistenceOfFetalMalpresentation;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
abn_presentation";
  d2rq:column "pregnancy_checkup.abn_presentation";
.
map:pregnancy_checkup_missing_heartbeat a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property PregSS:MissingFetalHeartbeat;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
missing_heartbeat";
  d2rq:column "pregnancy_checkup.missing_heartbeat";
.
map:pregnancy_checkup_edema a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:pregnancy_checkup;
d2rq:property OpenGALEN:OedemaSymptom;
d2rq:propertyDefinitionLabel "pregnancy_checkup edema";
d2rq:column "pregnancy_checkup.edema";
.
map:pregnancy_checkup_vaginal_infection a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property PregSS:VaginalInfection;
  d2rq:propertyDefinitionLabel "pregnancy_checkup
vaginal_infection";
  d2rq:column "pregnancy_checkup.vaginal_infection";
.
map:pregnancy_checkup_preg_record_id__ref a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_checkup;
  d2rq:property PregSS:preg_record_id;
  d2rq:refersToClassMap map:pregnancy_record;
  d2rq:join "pregnancy_checkup.preg_record_id =>
pregnancy_record.preg_record_id";
.
# Table pregnancy_record
map:pregnancy_record a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern
"pregnancy_record/@@pregnancy_record.preg_record_id@@";
  d2rq:class PregSS:PregnancyRecord;
  d2rq:classDefinitionLabel "pregnancy_record";
.
map:pregnancy_record__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property rdfs:label;
  d2rq:pattern "pregnancy_record
#@@pregnancy_record.preg_record_id@";
.
map:pregnancy_record_preg_record_id a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS:preg_record_id;
  d2rq:propertyDefinitionLabel "pregnancy_record
preg_record_id";
  d2rq:column "pregnancy_record.preg_record_id";
  d2rq:datatype xsd:integer;
.
map:pregnancy_record_lmp a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS>LastMenstrualPeriod;
  d2rq:propertyDefinitionLabel "pregnancy_record lmp";
  d2rq:column "pregnancy_record.lmp";
  d2rq:datatype xsd:date;
.
map:pregnancy_record_edc a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS:EstimatedDateOfConfinement;
  d2rq:propertyDefinitionLabel "pregnancy_record edc";
  d2rq:column "pregnancy_record.edc";
  d2rq:datatype xsd:date;
.
map:pregnancy_record_date_added a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS:DateAdded;
  d2rq:propertyDefinitionLabel "pregnancy_record
date_added";
  d2rq:column "pregnancy_record.date_added";
  d2rq:datatype xsd:dateTime;
.
map:pregnancy_record_health_center_id__ref a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS:health_center_id;
  d2rq:refersToClassMap map:barangay_health_center;
  d2rq:join "pregnancy_record.health_center_id =>
barangay_health_center.health_center_id";
.
map:pregnancy_record_patient_id__ref a
d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:pregnancy_record;
  d2rq:property PregSS:pregnancy_record_patient_id;
  d2rq:refersToClassMap map:patient;
  d2rq:join "pregnancy_record.patient_id =>
patient.patient_id";
.
# Table province
map:province a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "province/@@province.province_id@@";
  d2rq:class PregSS:Province;
  d2rq:classDefinitionLabel "province";
.
map:province__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:province;
  d2rq:property rdfs:label;
  d2rq:pattern "province #@@@province.province_id@";
.
map:province_province_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:province;
  d2rq:property PregSS:province_id;
  d2rq:propertyDefinitionLabel "province province_id";
  d2rq:column "province.province_id";
  d2rq:datatype xsd:integer;
.
map:province_name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:province;
  d2rq:property PregSS:Name;
  d2rq:propertyDefinitionLabel "province name";
  d2rq:column "province.name";
.
map:province_region_id__ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:province;
  d2rq:property PregSS:region_id;
  d2rq:refersToClassMap map:region;
  d2rq:join "province.region_id => region.region_id";
.

```

```

# Table referral
map:referral a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "referral/@@referral.referral_id@";
  d2rq:class PregSS:Referral;
  d2rq:classDefinitionLabel "referral";
.
map:referral__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property rdfs:label;
  d2rq:pattern "referral #@@referral.referral_id@";
.
map:referral_referral_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:referral_id;
  d2rq:propertyDefinitionLabel "referral referral_id";
  d2rq:column "referral.referral_id";
  d2rq:datatype xsd:integer;
.
map:referral_referral_date a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:ReferralDate;
  d2rq:propertyDefinitionLabel "referral referral_date";
  d2rq:column "referral.referral_date";
  d2rq:datatype xsd:date;
.
map:referral_problem a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:Problem;
  d2rq:propertyDefinitionLabel "referral problem";
  d2rq:column "referral.problem";
.
map:referral_initial_action a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:InitialAction;
  d2rq:propertyDefinitionLabel "referral initial_action";
  d2rq:column "referral.initial_action";
.
map:referral_remarks a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:Remarks;
  d2rq:propertyDefinitionLabel "referral remarks";
  d2rq:column "referral.remarks";
.
map:referral_referred_to a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:ReferredTo;
  d2rq:propertyDefinitionLabel "referral referred_to";
  d2rq:column "referral.referred_to";
.
map:referral_examination_date a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:ExaminationDate;
  d2rq:propertyDefinitionLabel "referral examination_date";
  d2rq:column "referral.examination_date";
  d2rq:datatype xsd:date;
.
map:referral_physician a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:Physician;
  d2rq:propertyDefinitionLabel "referral physician";
  d2rq:column "referral.physician";
.
map:referral_referral_center a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:ReferralCenter;

```

```

  d2rq:propertyDefinitionLabel "referral referral_center";
  d2rq:column "referral.referral_center";
.
map:referral_findings a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:Findings;
  d2rq:propertyDefinitionLabel "referral findings";
  d2rq:column "referral.findings";
.
map:referral_secondary_action a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:SecondaryAction;
  d2rq:propertyDefinitionLabel "referral secondary_action";
  d2rq:column "referral.secondary_action";
.
map:referral_instructions a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:Instructions;
  d2rq:propertyDefinitionLabel "referral instructions";
  d2rq:column "referral.instructions";
.
map:referral_preg_record_id_ref a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:referral;
  d2rq:property PregSS:preg_record_id;
  d2rq:refersToClassMap map:pregnancy_record;
  d2rq:join "referral.preg_record_id =>
pregnancy_record.preg_record_id";
.
# Table region
map:region a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "region/@@region.region_id@";
  d2rq:class PregSS:Region;
  d2rq:classDefinitionLabel "region";
.
map:region__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:region;
  d2rq:property rdfs:label;
  d2rq:pattern "region #@@region.region_id@";
.
map:region_region_id a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:region;
  d2rq:property PregSS:region_id;
  d2rq:propertyDefinitionLabel "region region_id";
  d2rq:column "region.region_id";
  d2rq:datatype xsd:integer;
.
map:region_name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:region;
  d2rq:property PregSS:Name;
  d2rq:propertyDefinitionLabel "region name";
  d2rq:column "region.name";
.
# Table tetanus_toxoid
map:tetanus_toxoid a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern
"tetanus_toxoid/@@tetanus_toxoid.tt_id@";
  d2rq:class PregSS:TetanusToxoid;
  d2rq:classDefinitionLabel "tetanus_toxoid";
.
map:tetanus_toxoid__label a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:tetanus_toxoid;
  d2rq:property rdfs:label;
  d2rq:pattern "tetanus_toxoid #@@tetanus_toxoid.tt_id@";

```



```
.
map:tetanus_toxoid_tt_id a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:tetanus_toxoid;
d2rq:property PregSS:tt_id;
d2rq:propertyDefinitionLabel "tetanus_toxoid tt_id";
d2rq:column "tetanus_toxoid.tt_id";
d2rq:datatype xsd:integer;
.
map:tetanus_toxoid_date_given a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:tetanus_toxoid;
d2rq:property PregSS:DateGiven;
d2rq:propertyDefinitionLabel "tetanus_toxoid date_given";
d2rq:column "tetanus_toxoid.date_given";
```

```
d2rq:datatype xsd:date;
.
map:tetanus_toxoid_preg_record_id__ref a
d2rq:PropertyBridge;
d2rq:belongsToClassMap map:tetanus_toxoid;
d2rq:property PregSS:preg_record_id;
d2rq:refersToClassMap map:pregnancy_record;
d2rq:join "tetanus_toxoid.preg_record_id =>
pregnancy_record.preg_record_id";
.
```

XII. Acknowledgement

Well, I haven't got all the luxury of time to write this, but here's what I can deliver.

It's been real hard work doing this, with all the sleepless nights, caffeine and sugar overdose, work, focuslessness and distractions, etc. But, without all these people (and higher beings), I wouldn't have finished this. Here let me introduce them to you.

Hah, first would be the higher being! Thanks god, even though I am being a crappy human at times, you still gave me chances. You surprisingly gave me a supporting family and awesome friends and abilities so that I can enjoy life. And yeah, I think we worked hand in hand on this, but I don't know what percentage I did, but I think I did a huge chunk, so I still get the credit, right? Nah, I love you god, you the best!

Next people, who I give tremendous, gigantic thanks would be my family. Le dad who works hard to be able to make me and my sisters study in good schools. Le mom who is very caring and cooks great and tolerates me even if I screwed up. Le sisters who are willing to be my slaves even without compensation. Le cousin who is very reliable and always there when you need her. Le tita who is very supportive and thoughtful. And all others, I can't mention cause it would be long, so.

Delving back on my early years in college, well, I got to give a hand for my buddies Ryan, Chard, Maq, Jman, Quindoy, Alvin, Mau, my gamer friends and awesome buddies who I enjoy being distracted with, up to now. My gal pals Arvi, Eunice, Kee, Jam, I thank you for being girls, and for keeping me safe from harm. To Carmy too, who helped by reverse psychology-ing me and agreeing to have friendship pacts with me, not to mention

she's thoughtful. To Doodie, who always looks out for me whenever I need to do something I'm not aware of. To Iehlee for supplying me with really good-tasting pastries so I get to have energy through the day. To Monie, who is good company to begin with, and ready to help whenever. Thanks buddies!

And here's to my office buddies at Cambridge! Special thanks to my bro, Citoman, who is ever so wisdomful and not-obviously-thoughtful-but-he-is, while being a tyrant at the same time, and supplying me with almost-dying chocolates. To Mr. Karlson, for letting me adopt his baby of a laptop named kralnos, which increased my productivity by 2x while also reducing my focus by 3x. To Eunice and Ethel, for being my superfriends, all-around tea buddies, lunch buddies, who I'm also really happy to serve. To all my teammates too, and Boss Jer, who tolerates my absentness and gives me lessons in the field, thanks!

To Miki, who I get into arguments with everyday. To Euls, who is always ready to accompany on whatever adventure I go to, except when she needs to go home. To Auradee, who has unequaled reliability. To Joy and Joanna, my friends I shared these hard times with. To Russy, who always likes to distract me, and I always fall for it, anyways. Thanks!

To my village friends who I have missed out a lot on, I'm sorry, but thanks for being with me through all of this.

To me, you awesome you. See, we can do the impossible?