

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

**Tuberculosis Registry Analysis, Consolidation,
Knowledge management, Evaluation and Reporting Tool
(TRACKER)**

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

Submitted by:
Siegfrid P. Mendoza
April 2010

ACCEPTANCE SHEET

The Special Problem entitled “*Tuberculosis Registry Analysis, Consolidation, Knowledge management, Evaluation and Reporting Tool (TRACKER)*” is prepared and submitted by Siegfried P. Mendoza in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Vincent Peter C. Magboo, M.D., M.Sc.
Adviser

EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. (candidate)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Richard Bryann L. Chua, M.Sc.	_____	_____
4. Aldrich Colin K. Co, M.Sc. (candidate)	_____	_____
5. Ma. Sheila A. Magboo, M.Sc.	_____	_____
6. Geoffrey A. Solano, M.Sc.	_____	_____
7. Bernie B. Terrado, M.Sc. (candidate)	_____	_____

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

Geoffrey A. Solano, 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

Reynaldo H. Imperial, Ph.D.
Dean
College of Arts and Sciences

ABSTRACT

Tuberculosis has been a major cause of illness and death in the Philippines yet TB control efforts have historically, been fragmented and uncoordinated. The National TB Control Program of the Department of Health (DOH) has made significant advances in improving the quality and extent of its control efforts but the private sector and even other departments of government have not been integrated into the overall TB control activities. But the World Health Organization (WHO) said the “Philippines has a long way to go” in eradicating tuberculosis (TB), even as the Health department confidently announced that the country will be able to attain the millennium development goal of reducing TB deaths by 50% within five years or 2015.

Recognizing the need for a more unified and concerted effort the DOH, assisted by the Philippine Coalition Against Tuberculosis (PhilCAT) organized various stakeholders into a working group to develop a comprehensive and integrated policy for TB control in the Philippines.

The development of the Tuberculosis Registry Analysis, Consolidation, Knowledge management, Evaluation and Reporting Tool (TRACKER) helps the DOH and PhilCAT TB coordinators to better assess the situation of TB in the country. The reports generated by the system can be used by the coordinators to determine which province or locality needs more attention and support when it comes to the treatment and prevention of TB.

Keywords: tuberculosis, disease registry, patient registry, clinical information system, registry

TABLE OF CONTENTS

ACCEPTANCE SHEET	i
ABSTRACT.....	ii
TABLE OF CONTENTS	iii
I. INTRODUCTION.....	1
A. Background of the study	1
B. Statement of the problem	5
C. Objectives	6
D. Significance.....	8
E. Scope and limitations.....	9
II. REVIEW OF RELATED LITERATURE	10
III. THEORETICAL FRAMEWORK	14
IV. DESIGN AND IMPLEMENTATION	19
A. Entity Relationship Diagram.....	19
B. Database Dictionary.....	21
C. Context Diagram.....	24
D. Data Flow Diagram.....	25
E. Technical Architecture.....	29
F. Technical Requirements.....	30
G. Definition of Terms.....	31
V. RESULTS	32
VI. DISCUSSION	44
VII. CONCLUSION	47
VIII. RECOMMENDATION	48
IX. BIBLIOGRAPHY	49
X. APPENDIX.....	51
XI. ACKNOWLEDGEMENT.....	118

I. Introduction

A. Background of the Study

Tuberculosis, or TB, is an infectious bacterial disease caused by *Mycobacterium tuberculosis*, which most commonly affects the lungs. It is transmitted from person to person via droplets from the throat and lungs of people with the active respiratory disease ^[3]. Persons with TB can be cured through regular and complete intake of the prescribed anti TB medications. Because patients frequently stop taking their medications before completing treatment, the Directly Observed Treatment, Short-course (DOTS) strategy is recommended. Tuberculosis is treatable with a six-month course of antibiotics. ^{[1][2]}

According to the World Health Organization, the Philippines ranks ninth in the world for prevalence of cases of tuberculosis and is still included in the WHO watch-list of 22 high-burdened countries. ^[3] Almost two thirds of Filipinos have tuberculosis, and up to five million people are infected yearly in our country. In 1996, WHO introduced the Directly Observed Treatment Short Course (DOTS) to ensure completion of treatment. The DOTS strategy depends on five elements for its success: Microscope, Medicines for TB, Monitoring, Directly Observed Treatment, and Political Commitment. If any of these elements are missing, our ability to consistently cure TB patients slips through our fingers. ^{[1][4]}

In 2006, the Philippines signed the Global Plan on TB, which seeks to reduce the prevalence and mortality of tuberculosis by 50% between 2006 and 2015. The Global Plan outlines a benchmark figure of 70% case detection rate (CDR), 85% treatment success rate (TSR) and 85% cure rate. The Department of Health (DOH) implemented the National

Tuberculosis Program (NTP). The main goals of the program are the same with that of WHO: to cure at least 85% of TB patients discovered and detect 70% of existing smear positive cases of TB. The NTP uses the DOTS strategy. [5]

Monitoring is the part of the DOTS strategy, which documents the progress of patients until the patient is totally cured. Recording and Documentation is an integral part of TB Management because it allows the physician to check if patients have completed their treatment religiously [6]. NTP uses standardized TB registry forms filled out by a nurse of a health facility such as hospitals, clinics, *et cetera*, both private and public, Rural Health Units (RHU) or Barangay Health Centers (BHC). The nurse or health worker will organize the data from these forms to NTP reports.

Figure 1 illustrates the flow of data between health facilities and offices. For public health facilities in the National Capital Region (NCR), the reports will be submitted directly to the DOH, while the other health facilities in the other regions and provinces will submit their reports the Provincial Health Office (PHO), then to their Regional Health Office (RHO), who in turn, will submit the compiled report to the DOH.

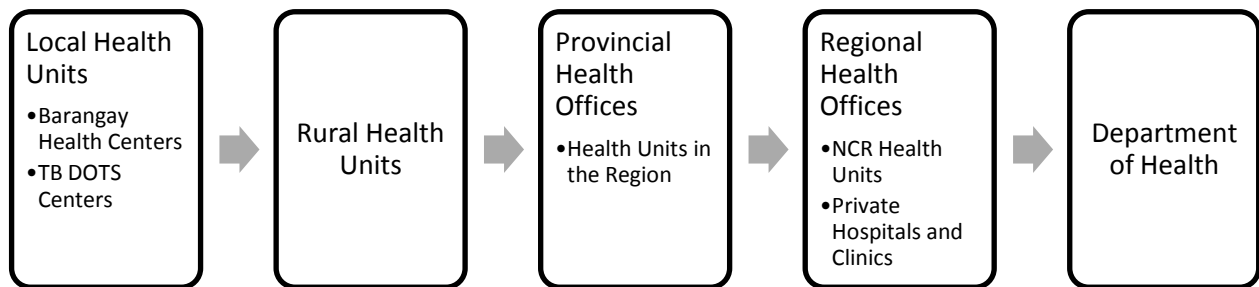


Figure 1. The transmission of collected TB data from different levels of health units and offices to the DOH main office, TRACKER

The data to be used by the DOH will generate yearly TB reports to be submitted to the WHO. The WHO in turn will use this data to aid them in monitoring the status of TB in different parts of the world.

The Philippine Coalition against Tuberculosis (PhilCAT) works in partnership with the DOH to spearhead collective efforts in controlling tuberculosis, together with leading local specialty societies of chest physicians, infectious disease specialists, and pharmaceutical industry representatives. The coalition serves as the “unifying force” to more than 50 highly-motivated organizations supporting the Philippines’ National TB Program ^[7].

One of the problems of the Health Sector in the Philippines is the lack of health information and services especially on the provincial or rural level. The National Telehealth Center (NThC) of the University of the Philippines Manila was established in June 1998 through a resolution of the University of the Philippines Board of Regents. The NThC was given the mandate to enhance the access to health care through information and communications technology. ^[8]

The NThC exists to promote the use of telehealth and related technologies in forming strategic partnerships with stakeholders in the delivery of services where such concepts are applicable. This shall be accomplished through responsible and innovative use of telecommunications and computer technologies. The Center aims to bridge the gaps between health services, educational needs, and community resources. ^[8]

Through the development of sustainable telehealth applications and through collaborations with organizations, health professionals, health workers, clients and the community at large, the NThC will lead in the promotion of the responsible use of technology in health, ensure quality in its endeavors, and empower its clients through information and knowledge exchange ^[8].

At present, there is no existing automated system that would support the transfer of reports from one level of the data flow to the next (e.g. rural health units to municipal health office, municipal to provincial health office, regional office to DOH head office). There is a solution introduced by the Philippine Tuberculosis Patient Registry system, a web-based information system that automates the collection TB data from various TB center in the Philippines and stores them in a database in a web server, and generates the TB reports required by the DOH.^[9] Also the Community Health Information Tracking System or CHITS presently has a National TB Control Program (NTP) module included which encodes data on the health center level and is part of the workflow of health workers in the health centers. Another problem is that not all of them have CHITS installed (only 18 sites in the Philippines and there are over 2,000 health centers in total) ^[10].

This paper proposes to develop a registry system that can be deployed in a local network environment, one that would not require Internet connection to function and that will allow users to encode, analyze, consolidate reports from its area of responsibilities, and manage previous reports, and other documents pertinent to TB center operations. These functions will allow TB program managers to evaluate the performance needs of each TB center throughout the country.

B. Statement of the Problem

One of the five key elements of the DOTS strategy is the standardized NTP recording and reporting. Accurate recording and reporting are important in the success of a TB control program. Reports are essential in the decision making process of key implementers as they reflect the volume of patients in need of drugs and other NTP supplies necessary for the operation of DOTS facilities.

Health facilities such as hospitals, clinics, both private and public, rural health units and barangay health centers and other health offices such as the Provincial and Regional Health office have to submit TB reports regularly. There are instances when submission of these reports are delayed or not received at all because of the difficulties of travelling from their region or province to the DOH Head Office in NCR, thus the action of the department will be delayed as well. The consolidation of data is also slower and the report submitted is more error prone because of the manual method of creating them. ^[10]

As of present, CHITS is installed in only 18 health centers throughout the country. It provides electronic data capture and management of patient health information. However, there is now a need to encode, view and analyze data at the provincial level. ^[10]

PhilCAT Monitoring and Evaluation (M & E) unit receives paper-based reports (TB registry reports, see Appendix) and conducts regular field monitoring visits to 254 Public-Private Mix DOTS (PPMD) centers around the country. As part of the monitoring procedures, NTP registers and reports of the units are reviewed. Discrepancies were noted in several records including the NTP Quarterly Report. Such minor discrepancies of data at the unit can cause

major effect once data is consolidated at the national level. These reports are encoded in PhilCAT's Epidemiology and Data Center. This process takes time and costly. ^[10]

C. Objectives

Main Objectives

Tuberculosis Registry Analysis, Consolidation, Knowledge-management, Evaluation and Reporting Tool (TRACKER), from its name, allows users to encode, analyze, consolidate reports from its area of responsibilities, and manage previous reports, and other documents pertinent to TB center operations such as certificate of accreditation, etc (knowledge-management). These functions allow TB program managers to evaluate the performance and needs of each TB center.

Specific Objectives

The system has the following functions:

1. Nurses or encoders of the Provincial Health Office have the following functionalities:
 - a. Encode TB registry cases
 - b. View TB register
 - c. Search for patient records
 - d. Update Health Facility information
 - e. View the TB statistics (no. of new/cured/treated/relapsed cases, no. of cases)
 - i. Cure Rate
 - ii. Completion Rate
 - iii. Death Rate
 - iv. Treatment Failure Rate
 - v. Defaulter Rate
 - vi. Transfer Out rate

2. The DOTS coordinator of the Provincial Health Office has the following functionalities:
 - a. View the encoded TB register specific to his facility or office.
 - b. View the NTP Quarterly Report on All Cases.
 - c. View the NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases specific to his facility or office.
 - d. View the Counting sheets for Quarterly report on the Treatment Outcome of Pulmonary TB Cases, Case Finding by Type
 - e. Update his account profile.
 - f. Update the profile of the facility or office.
 - g. Create user accounts for nurses and encoders for his particular province.
 - h. View the TB statistics (no. of new/cured/treated/relapsed cases, no. of cases)
 - i. Cure Rate
 - ii. Completion Rate
 - iii. Death Rate
 - iv. Treatment Failure Rate
 - v. Defaulter Rate
 - vi. Transfer Out rate

3. The PhilCAT analyst has the following functionalities:
 - a. View the NTP Quarterly Report on All Cases.
 - b. View the NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases specific to his facility or office.
 - c. View the Counting sheets for Quarterly report on the Treatment Outcome of Pulmonary TB Cases, Case Finding by Type
 - d. Update his account profile.
 - e. Update the profile of the facility or office.
 - f. View the TB statistics (no. of new/cured/treated/relapsed cases, no. of cases)
 - i. Cure Rate
 - ii. Completion Rate
 - iii. Death Rate

- iv. Treatment Failure Rate
 - v. Defaulter Rate
 - vi. Transfer Out rate
4. The system administrator has the following functionalities:
- a. Create new user accounts for the Provincial NTP and PhilCAT Coordinators and the nurse or encoder assigned to encode the reports
 - b. Manage the Provincial Health Offices registered in the system.
 - c. Manage existing user accounts
 - d. Update his profile
5. Guests and unregistered users, which may include the patients themselves, researchers, etc, can view the TB statistics (no. of new/cured/treated/relapsed cases, no. of cases per region) about a specific region
- a. Cure Rate
 - b. Completion Rate
 - c. Death Rate
 - d. Treatment Failure Rate
 - e. Defaulter Rate
 - f. Transfer Out rate

D. Significance of the Study

TRACKER aims to improve the capture and management of TB patient data by optimizing the report generation, specifically about TB. The existing CHITS-NTP module included is not optimized to suit the needs of the PhilCAT. TRACKER attempts to reduce the cost of maintaining a full-sized data center by providing a tool for data encoding, analysis, and visualization at the municipal, provincial and regional levels. Data processing will be transferred from PhilCAT to these levels in the health organization.

The development of this information system helps the different administrators, policy makers and scientists alike in determining the best and effective way in helping treat tuberculosis in the country. The compilation of the data being automated makes it easy for the users to make decisions, analyses and recommendations. It also helps the users to easily conduct surveys or statistical studies on the situation of the tuberculosis and its treatment in the country.

This registry system is intended to help the TB program managers to evaluate the performance and needs of each health center.

E. Scope and Limitations

1. The system will be available as an online web application and will only be fully functional with an internet connection.
2. The accuracy and correctness of the data inputted is the user's responsibility.
3. The system will only generate reports and other forms of results based on the TB registers provided by the health facilities.
4. Information inputted in the system will only come from the TB registers submitted by the RHUs of each provincial health office.
5. The nurse and the provincial NTP coordinator of a specific health office cannot access the TB Registry Information of the other facilities.

II. Review of Related Literature

Community-based primary care information systems are one of the building blocks for national health information systems. In the Philippines, after the devolution of health care to local governments, Tolentino, et al., observed “health information system islands” connected to national vertical programs being implemented in devolved health units. These structures lead to a huge amount of “information work” in the transformation of health information at the community level. Tolentino, et al., describes work done to develop and implement the open-source Community Based Health Information Tracking System (CHITS) Project, which was implemented to address this information management problem and its outcomes. They highlighted building community level information systems that link to national level health information systems. ^[11]

A disease registry is an organized system used for collection, storage, analysis and interpretation of data of individuals with a particular disease of concern. ^[12] There are many disease registries existing today, one of them is the Cancer Registry of the United States. By definition, a cancer registry is an information system designed for the collection, storage, management, and analysis of data on persons with cancer, usually covering a hospital or group of hospitals. ^[13]

The Cancer Registry is a particular type of disease registry and its major purposes are: 1) to establish and maintain a cancer incidence reporting system; 2) to be an informational resource for the investigation of cancer and its causes; and 3) to provide information to assist public health officials and agencies in the planning and evaluation of cancer prevention and cancer

control programs. Cancer registries are a primary source for unbiased population-based case control studies, the end points for cohort studies and clinical trials -- and perhaps most importantly, the beginning point for survival analysis ^[14].

Data from population-based registries can be used for monitoring the distribution of late-diagnosed cases of cancer of the types for which early diagnosis is the strategy for control, esp., communities, ethnicities, age and other demographic groups. Registries are less complex and simpler to setup than electronic medical records that according to a recent survey are only used by 9% of small offices where almost half of the US doctors work ^[15].

In a study presented by Hummel, Conceptual Framework for Chronic Illness Management, which mentioned the use of computerized registries in large clinical trials in order to keep track of all the patients entering the trial who enrol at different time over many months. He suggested that the registry should be able to distinguish between patients who have left a practice and those have simply not come in for a check-up. Those who have not come in for check-up should be contacted to remind them about their check-up. The registry should also include both dates and values for crucial intervention to identify patients who are either overdue for a monitoring intervention such as blood pressure. He further suggested that an individual with clerical or entry-level clinical experience should be the one that would add and update patient's data. It should be most preferable that the individual should be somebody that has knowledge about the disease. ^[16]

In addition to this, the paper stated that although working with data in the registry will be quite enjoyable to the physician, it is seldom cost-effective to have physicians perform registry

maintenance. The article concluded that a disease registry is an important element of the disease management program for any delivery system that seeks to manage their patients using the program integrated within a primary care delivery system.^[16]

Clinicomp CIS, a program installed in Brooke Army Medical Center at Texas, enables care providers throughout the Brooke Enterprise to gain access to complete in-patient records, which allows improvement in decision-making. Furthermore, Clinicomp CIS also integrates information from all areas including: the department systems, ambulatory facilities, and patient monitoring equipment, into a central repository. This allows quick and easy access to information and analysis.^[17]

In New York City's Tuberculosis Control Program, a computerized registry is used to document the suspected and confirmed cases of TB. The surveillance crew makes sure that the data gathered are entered into the registry. These data are routinely analyzed to detect outbreaks, trends, and instances of possible false positive specimen results, and to research issues of clinical and operational importance.^[18]

Pfeifer mentioned that the National Hansen's Disease Program is a health program whose mission is to conduct leprosy research, educate patients and health providers about the disease. This program maintains a computerized registry, and clinical rehabilitative and laboratory research. Because the disease is a noticeable disease, the data gathered is statistically analyzed and reported to the Centers for Disease Control and Prevention. Summary reports are also generated for different health agencies.^[19]

The Patient Clinical Information System (PatCIS) is a project at Columbia-Presbyterian Medical Center to provide patients with access to health information, including their own medical records (permitting them to contribute selected aspects to the record), educational materials and automated decision support. The architecture of the system allows for multiple, independent components which make use of central services for managing security and usage logging functions. ^[20]

Clinical information systems (CIS) and other information management tools hold the promise of improving the quality and safety of patient care and increasing the efficiency of health care personnel. However, the health care industry lags behind other sectors in information technology investment and, with few exceptions, has not fully benefited from the information revolution. ^[21]

A clinical information system is a collection of various information technology applications that provides a centralized repository of information related to patient care across distributed locations. This repository represents the patient's history of illnesses and interactions with providers by encoding knowledge capable of helping clinicians decide about the patient's condition, treatment options, and wellness activities. The repository also encodes the status of decisions, actions underway for those decisions, and relevant information that can help in performing those actions. ^[22]

III. Theoretical Framework

A. Disease Registry

Disease registries are databases that collect clinical data on patients with a specific disease (diabetes, asthma, TB, hypertension, etc) or keep track of specific medical tests (Pap smear, mammogram, etc).^[23] Most frequently registries vary in sophistication from simple spreadsheets that only can be accessed by a small group of physicians to very complex databases that are accessed online across multiple institutions.^[24]

They can provide health providers (or even patients) with reminders to check certain tests in order to reach certain quality goals.

B. Tuberculosis Treatment

Tuberculosis, or TB, is an infectious bacterial disease caused by *Mycobacterium tuberculosis*, which most commonly affects the lungs. It is transmitted from person to person via droplets from the throat and lungs of people with the active respiratory disease.^[1]

People with latent TB infection have TB germs in their bodies, but they are not sick because the germs are not active. These people do not have symptoms of TB disease, and they cannot spread the germs to others. However, they may develop TB disease in the future. They are often prescribed treatment to prevent them from developing TB disease. Because there are less bacteria in a person with latent TB infection, treatment is much easier.^[1]

Directly Observed Treatment Short-course (DOTS) Strategy is the recommended TB control strategy of WHO in the Western Pacific Region. With DOTS, TB can be cured, transmission can be prevented, and increasing trend of multi-drug resistance can be reversed. ^[4]

The DOTS strategy is recognized as one of the most cost-effective of all interventions. In addition to this, it is also the only intervention, which has consistently demonstrated high cure rates. It is important in this strategy that once anti-TB drugs are started the patient should be under directly observed treatment. This means that the drug intake should be supervised and done regularly.

A NTP Treatment Card is issued to a patient. A patient is given a unique TB Case Number to identify them from other TB cases. The patient is asked for its basic information such as its name and address. After this, the patient's TB classification is also determined, as well as its type of patient. The patient could be any of the following:

1. New - A patient who has never had treatment for TB or who has a history of taking anti-TB drugs previously for less than a month.
2. Relapse - A patient previously treated for tuberculosis that has been declared cured or treatment completed, and is diagnosed with bacteriology positive smear or culture.
3. Failure - A patient who, while on treatment, remains or become smear positive again at five (5) months or later during the course of treatment.
4. Return after Default (RAD) – A patient who returns to treatment with positive smear, following interruption of treatment for two (2) months or more.
5. Transfer-in – A patient who has been transferred from another facility with proper referral slip to continue treatment.
6. Other – all cases who do not fit into any of the abovementioned categories.

The patient is then subjected to a series of scheduled sputum examinations and daily drug intake. A patient who undergoes the treatment may achieve any of the following treatment outcomes:

1. Cure – A sputum smear positive patient with completed treatment and is sputum smear negative in the last month of the treatment or and on at least one previous occasion in the continuation phase
2. Completed Treatment – a patient who has completed the treatment but has not met the criteria for cure or failure
3. Died – a patient who died for any reason during the course of the treatment
4. Failed – a patient who is sputum smear-positive at five months or later during the treatment or an initially sputum smear-negative patient before starting the treatment who becomes smear-positive during the treatment
5. Defaulted – a patient who interrupted treatment for two consecutive months or more
6. Transfer-out – a patient who transferred out to another DOTS facility with proper referral slip for continuation of treatment and whose treatment outcome is not known.

C. NTP Recording and Reporting

Recording and reporting are important in the implementation of a successful TB control program. Availability of Records ensures provision of appropriate and effective care for patients. Through efficient recording, health workers can monitor that each TB symptomatic found is examined and cured. Records therefore should contain accurate, complete, and up-to-date information on patient's diagnosis, treatment, follow-up examinations, and treatment outcome. ^[5]

A nurse or health worker will summarize the NTP treatment cards into reports required by the NTP. The following reports are the submitted by the NTP Coordinator to the DOH:

1. NTP Quarterly Report on New TB Cases and Relapses

2. NTP Quarterly Report on the Treatment Outcome of Pulmonary Cases

Other reports that are also generated are as follows:

1. Counting Sheet for Case Finding by Type
2. Counting Sheet for Quarterly Report on the Treatment Outcome of Pulmonary TB Cases
3. TB Register – summary of all the TB patients of a certain health facility or office

These reports are not submitted to the DOH. These only aid the health facility or offices to generate the reports submitted to DOH.

D. Information System

A system, whether automated or manual, that comprises people, machines, and/or methods organized to collect, process, transmit, and disseminate data that represent user information. Information systems are the software and hardware systems that support data-intensive applications. It is the entire infrastructure, organization, personnel, and components for the collection, processing, storage, transmission, display, dissemination, and disposition of information. ^{[25][26]}

E. Clinical Information System

A clinical information system is a collection of various information technology applications that provides a centralized repository of information related to patient care across distributed locations. This repository represents the patient's history of illnesses and interactions with providers by encoding knowledge capable of helping clinicians decide about the patient's condition, treatment options, and wellness activities. The repository also encodes the status of

decisions, actions underway for those decisions, and relevant information that can help in performing those actions. The database could also hold other information about the patient, including genetic, environmental, and social contexts ^[23].

F. Patient Registry

A patient registry is a database that includes information gathered over time from a large number of patients. ^[27] Information contained in the registry may include patient's personal information (name, address, age, etc) and health related information (medical history, drug intake). ^[28] It may also contain clinical information of a patient population with a specific disease. This type of registry is particularly helpful in diseases that are relatively rare. ^[29] The purpose of this registry is to determine the number of people infected or are suffering from a specific illness and to provide the medical and scientific community with access to the patient population for treatment. ^[30]

IV. Design and Implementation

A. Entity Relationship Diagram

An Entity Relationship Diagram as shown in Figure 2 is a graphical representation of the entities and the relationships between them.

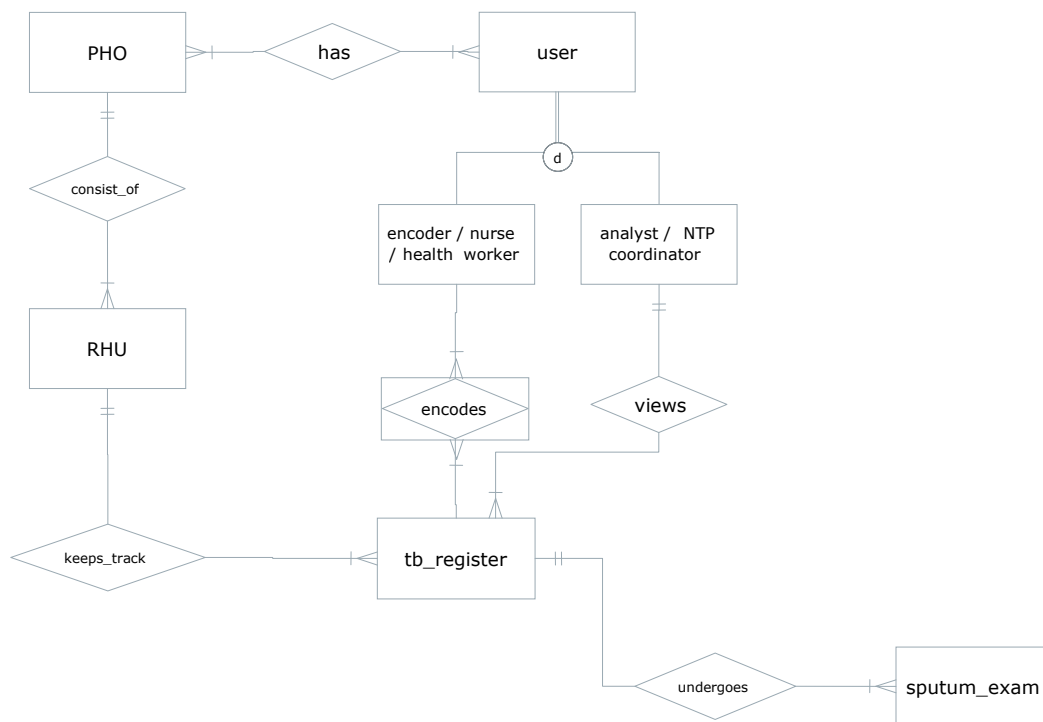


Figure 2. Entity Relationship Diagram, TRACKER

Based on Figure 2, the entities of the TRACKER are the following:

1. PHO – stores the information about the Provincial Health Office
2. RHU – stores the information about the Rural Health Units
3. tb_register – stores the information of each TB register which contains information about each TB case recorded in that particular RHU
4. sputum_exam – stores the information about the patient's TB case's sputum examination
5. user – stores information about the users of the system. It can be the system administrator, nurse or rural health worker or NTP coordinator of a health facility, and the PhilCAT Monitoring and Evaluation Unit analyst.

The following are the relationships between entities:

1. A PHO has many RHUs within its area of responsibility
2. RHUs keeps track one or many TB cases.
3. Each TB case has only one treatment information.
4. A TB case can undergo at least one sputum examination
5. An encoder encodes many TB registers.
6. The PhilCAT analyst or the NTP provincial coordinator can view the encoded TB registers.
7. A user is a member of one PHO.
8. A health office can have many users.

B. Data Dictionary

1. PHO – table of the provincial health offices

data field	data type	description
<u>PHO_ID</u>	tinyint(20)	unique identifier of each provincial health office
PHO_NAME	varchar(100)	name of the provincial health office
PHO_TYPE	varchar(100)	type of facility
PHO_REGION	varchar(10)	region where the health office is located
PHO_PROVINCE	varchar(30)	province where the health office is located
PHO_STREET	varchar(30)	street where the health office is located
PHO_CITY	varchar(30)	city where the health office is located
PHO_CONTACT	varchar(30)	contact number of the health office

RHU – table of rural health units / hospitals/ PPMD centers

data field	data type	description
<u>RHU_ID</u>	tinyint(20)	unique identifier of each rural health unit
<u>PHO_ID</u>	tinyint(20)	identifies the PHO where the RHU belongs
RHU_NAME	varchar(100)	name of the RHU
RHU_TYPE	varchar(100)	type of RHU facility
RHU_STREET	varchar(30)	street where the health office is located
RHU_CITY	varchar(30)	city
RHU_CONTACT	varchar(30)	contact number of the health office

tb_register – table of patients of the TB register

data field	data type	description
<u>TB_CASENO</u>	varchar(8)	unique identifier of each TB case
<u>PHO_ID</u>	tinyint(20)	identifies the provincial health office where the patient belongs
<u>RHU_ID</u>	tinyint(20)	identifies which RHU the TB registry came from
FNAME	varchar(100)	first name of the patient

MI	varchar(5)	middle initial of the patient
LNAME	varchar(30)	last name of the patient
SEX	varchar(1)	gender of the patient
AGE	int(2)	age of the patient
DATE_OF_REG	Date	date of registration
P_CITY	varchar(30)	city name of the patient's address
HEALTH_FAC	varchar(30)	name of the health facility where the patient is receiving treatment
P_SOURCE	varchar(20)	source of the patient, public or private sector
DOC_NAME	varchar(30)	name of the referring physician
TB_CLASS	varchar(2)	TB classification, P for Pulmonary, EP for Extra-Pulmonary
P_TYPE	varchar(30)	patient type (New, Relapse, Transfer-In, etc)
P_CATEGORY	int(2)	prescribed treatment regimen by category (I, II or III)
START_DATE	Date	start date of the treatment
OUTCOME	varchar(50)	outcome of the treatment
PARTNER	varchar(100)	name of the treatment partner
REMARKS	Text	remarks made about the treatment
REVIEW	varchar(5)	review decision (Yes, No)

sputum_exam – stores the information about the patient's TB case's sputum examination

data field	data type	description
<u>TB_CASENO</u>	varchar(8)	unique identifier of each TB case
<u>MONTH</u>	varchar(20)	month of treatment
DATE_EXAM	Date	date of the sputum examination
RESULT	varchar(20)	result of the examination
WEIGHT	int(5)	weight of the sputum examination

users – table of users of the system

data field	data type	description
ID	smallint(6)	unique id of each user
<u>PHO_ID</u>	tinyint(10)	identifies the PHO where the user belongs
<u>USERNAME</u>	varchar(20)	unique identifier of each user
PASSWORD	varchar(20)	password
NAME	varchar(100)	name of the user
MIDDLENAME	varchar(20)	middle name of the user
LAST_NAME	varchar(100)	last name of the user
DESIGNATION	varchar(20)	designation of the user
USERTYPE	varchar(20)	type of user, determines the permissions granted to a user
STATUS	varchar(5)	account activation status
GENDER	varchar(5)	gender of the user

C. Context Diagram

There are five (5) users of the TRACKER as shown in the Context Diagram in Figure 3. These are the system administrator, NTP coordinator of the provincial health office, PhilCAT analyst, the guests (e.g. researchers, TB coordinators of other provinces, etc), and the nurses or encoders at the provincial level. The PhilCAT analyst can view information and other statistics about TB on a certain province. The NTP Coordinator can view the reports submitted by the health facilities. The nurse or health worker of the health office can create, update, and view TB records of patients in their province.

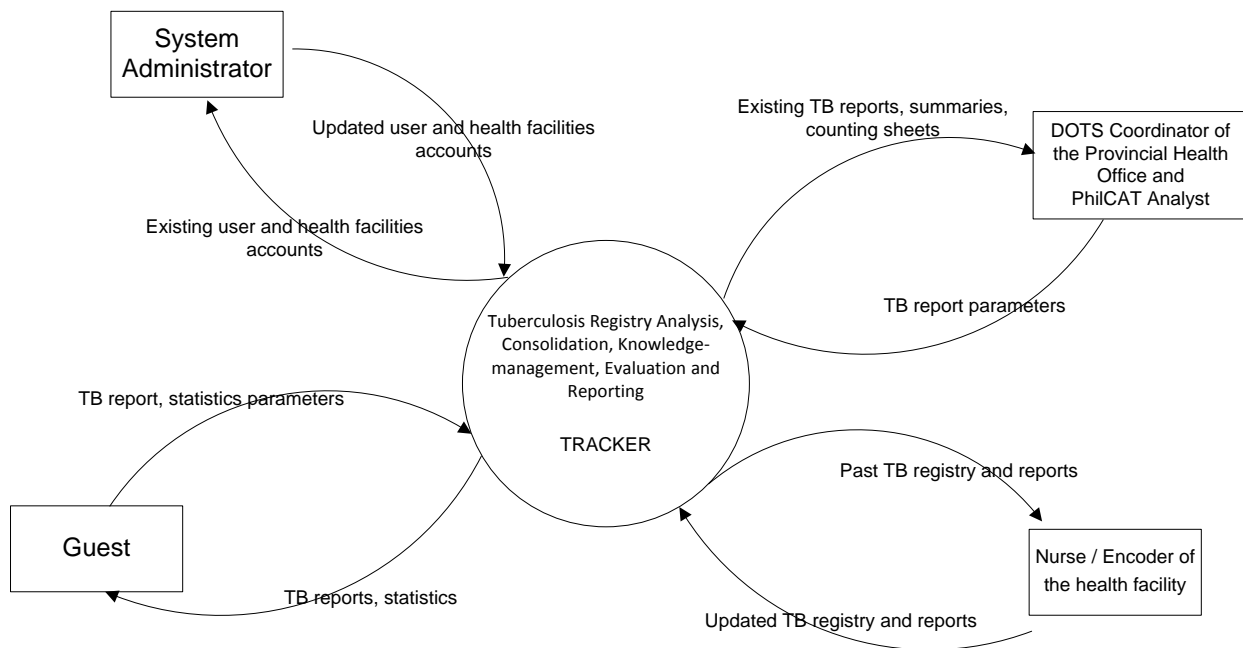


Figure 3. Context Diagram, TRACKER

D. Data Flow Diagram

The top-level data flow diagram of the TRACKER is seen in the next figure. The diagram shows 5 main processes of the system. Registered users need to login first, for authentication and verification. The four users namely, provincial NTP coordinator, PhilCAT analyst, nurse or health worker and the system administrator can then manage their account information.

The nurses and the provincial NTP coordinator can update and view the TB registry of their facility. For the PhilCAT analyst and NTP coordinator, they can generate and view TB related reports and statistics of their area of responsibility. Finally the last users are the guests, which can only view TB summary reports and TB information.

A PhilCAT analyst and NTP coordinator can generate reports, summaries, etc. by providing report parameters (these may include, but not limited to: year, month, province, region, TB category, etc). Because not all reports can be viewed by all users, the type of user should be verified first.

After generating reports and verifying that all required information are complete (all RHUs have been accounted), the NTP coordinator may then print these reports for further administrative actions or analyses then forward it to the to the Regional Health Office.

In updating the TRACKER accounts, users can, add, edit, and view their account. Users have to login first before they can do so.

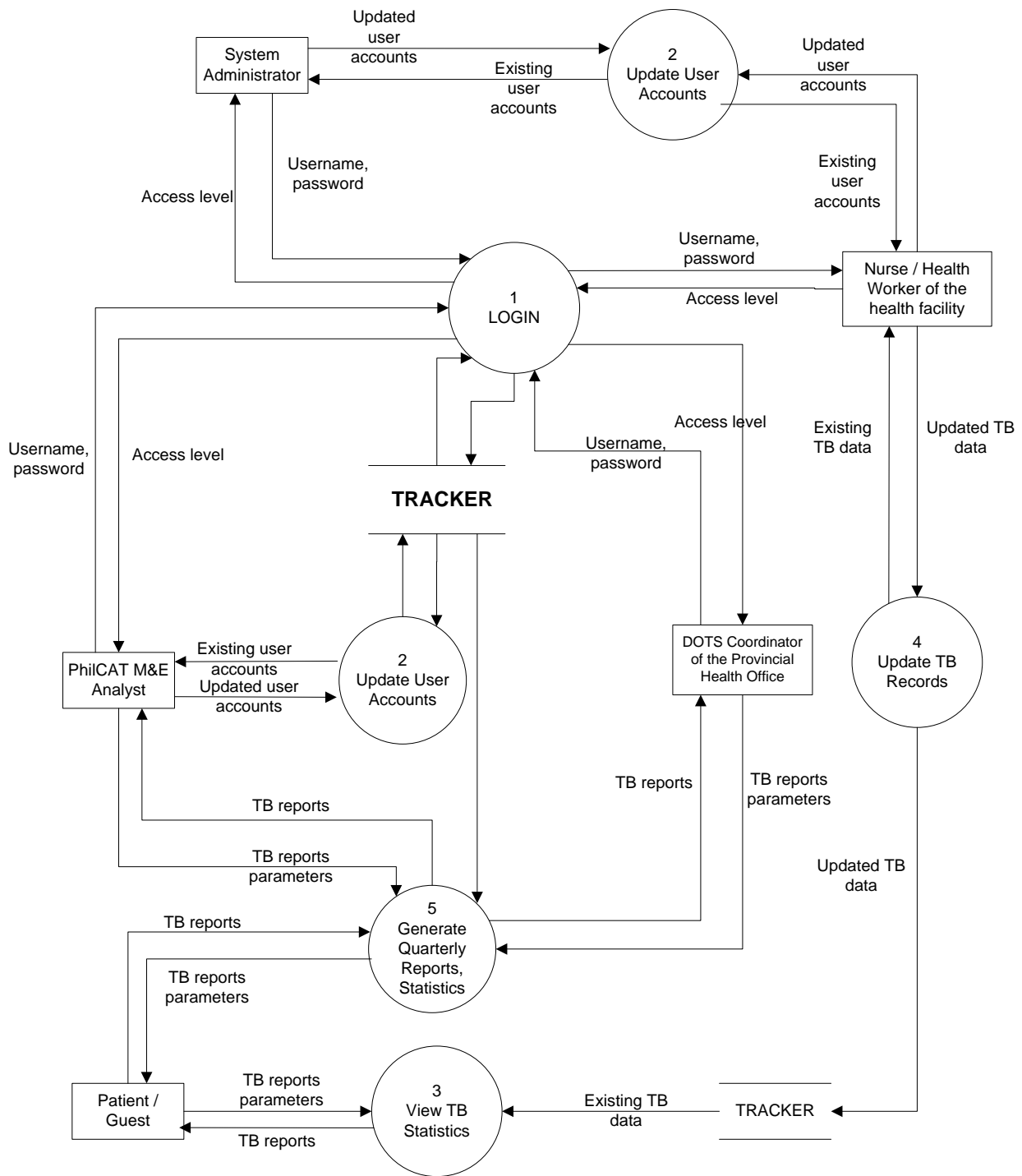


Figure 4. Top-Level Data Flow Diagram, TRACKER

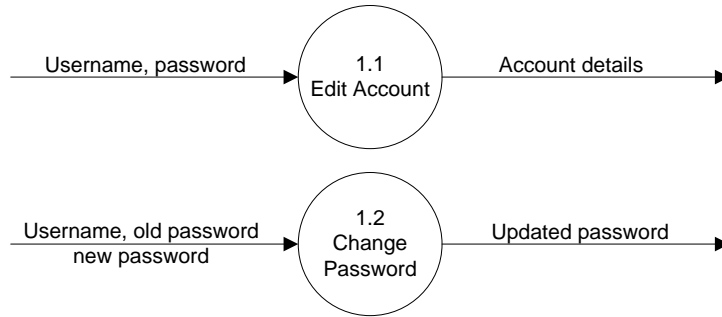


Figure 5. Sub-explosion of Process 1, Login, TRACKER

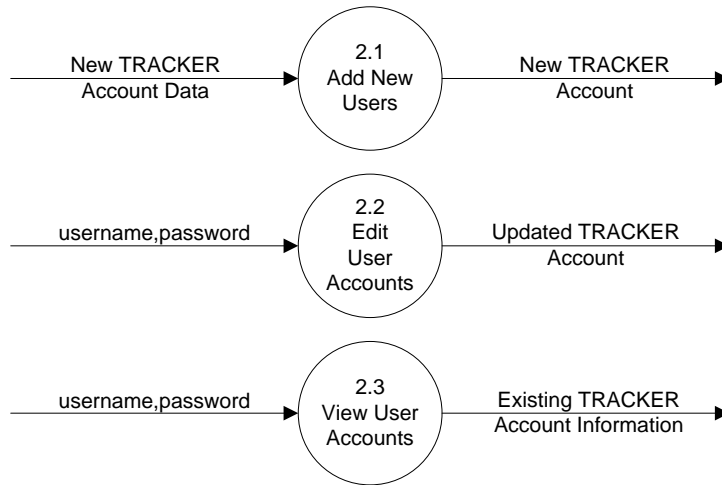


Figure 6. Sub-explosion of Process 2, Update User Accounts, TRACKER

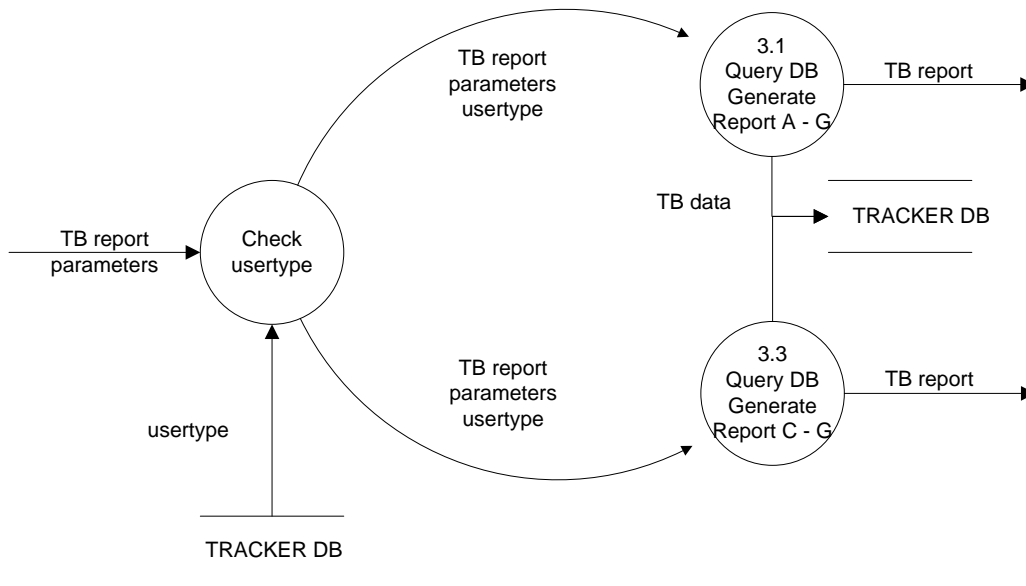


Figure 7. Sub-explosion of Process 3, Generate Summaries, Reports, Tables, TRACKER

The following are the required reports to be generated by the TRACKER system:

1. Report A – NTP Quarterly Report on New Cases and Relapses of Tuberculosis and on Drug Requirement
2. Report B – NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases Registered 13-15 months earlier
3. Report C – TB Register specific to his facility or office
4. Report D – Counting Sheet for Quarterly Report on the Treatment Outcome of Pulmonary TB Cases
5. Report E – Counting Sheet for Case Finding by Type
6. Report F – Summary Reports on Pulmonary Smear-positive cases
7. Report G – Summary Reports on the treatment outcomes for each New Smear-Positive Cases, Relapse Cases and Failure Cases

E. Technical Architecture

The TRACKER system is a web-based information system and uses the client-server architecture. This type of network architecture is composed of computers which can either be clients, these are the workstations on which the users run the application, and servers, which are powerful computers dedicated to managing disk drives, printers or network traffic. In the case of the system, this type of architecture will allow the clients to send and retrieve data to the database. Upon receiving the client's data or request, the server will then process it and return the results to the client.

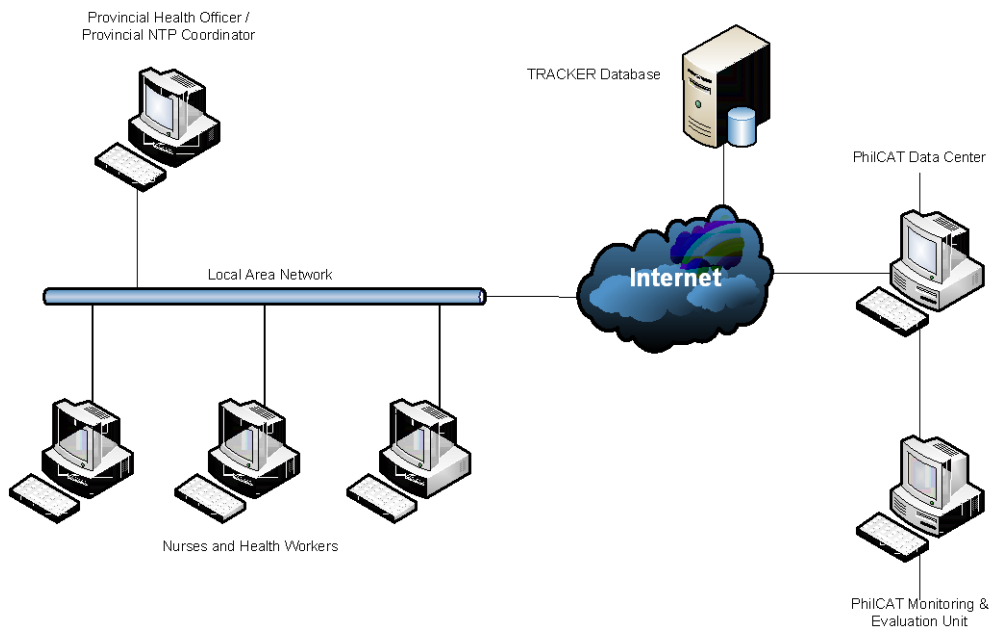


Figure 8. Technical Architecture, TRACKER

F. Technical Requirements:

The server machine:

Hardware Specifications

Processor of at least 500 MHz

RAM should be at least 256MB

Software Specifications

Operating System should be at least Microsoft 98

The following applications should be installed for the system to run:

PHP 5.3 or later

Apache HTTP Server 1.3 or later

XAMPP 1.7 or later

The Client machine:

Hardware Specifications

Processor of at least 500 MHz

RAM should be at least 128MB

Software Specifications

Operating System should be at least Microsoft 98

Internet browser should have JavaScript enabled

G. Definition of Terms

1. TB case – a patient in whom the diagnosis has been confirmed bacteriologically or a patient in whom a presumptive diagnosis of active TB is made on the basis of radiological evidence and whom a physician decides to treat with a full-course of anti-TB therapy ^[5].
2. Health Facility – either a hospital, rural health unit, or barangay health station
3. Rural Health Unit (RHU) – Municipal Health centers
4. Barangay Health Station (BHS) – Barangay Health centers / community health centers
5. Health Office – can either be a Provincial health office or Regional health office
6. Sputum examination – preferred method for the diagnosis of TB. Three (3) sputum specimens are acquired. The specimens are then examined for existence of bacilli. A result is positive if at least two (2) specimens are positive, negative if all three tested negative, and doubtful if only one is positive ^[5]
7. Smear positive – at least one bacilli exists ^[5]
8. Smear negative – no bacilli exists ^[5]
9. Public-Private Mix DOTS (PPMD) – a strategy implemented by the DOH to improve TB case detection with the participation of private physicians ^[31]

V. Results

The TRACKER homepage is shown in Figure 8. There are links provided so that guest users can view information about TB (DOH, WHO, PhilCAT). On the same page, members can also login and access their accounts and gain access to the TRACKER system. Also there is a link where guest can generate statistics about TB. Figure 9 shows some of the information that any user can view upon entering the General Information Page.

TRACKER
TB Registry Analysis Consolidation Knowledge-management Evaluation and Reporting Tool

You are logged in as **nurse12**. You are an Encoder
Palawan PHO

Home
Your Account
TB Registry
Health Facility
TB Statistical Reports

Welcome to the TRACKER SYSTEM!

Tuberculosis, or TB, is an infectious bacterial disease caused by *Mycobacterium tuberculosis*, which most commonly affects the lungs. It is transmitted from person to person via droplets from the throat and lungs of people with the active respiratory disease.

Persons with TB can be cured through regular and complete intake of the prescribed anti TB medications. Because patients frequently stop taking their medications before completing treatment, the Directly Observed Treatment, Short-course (DOTS) strategy is recommended. Tuberculosis is treatable with a six-month course of antibiotics.

The Philippine Coalition Against Tuberculosis (PhilCAT) was founded in 1994 amidst the realization that collaboration and partnership with the private sector was vitally needed to succeed in the country's fight against tuberculosis, an infectious disease whose victims continue to be viewed with stigma. The major founding members were the Department of Health (DOH), leading specialty societies of chest physicians, infectious disease specialists, and pharmaceutical industry representatives.

 **University of the
Philippines Manila**

 **PhilCAT**
Fighting TB Through Unified Action

 **Department of Health**
Republic of the Philippines

 **World Health
Organization**

Figure 8. Home Page, TRACKER

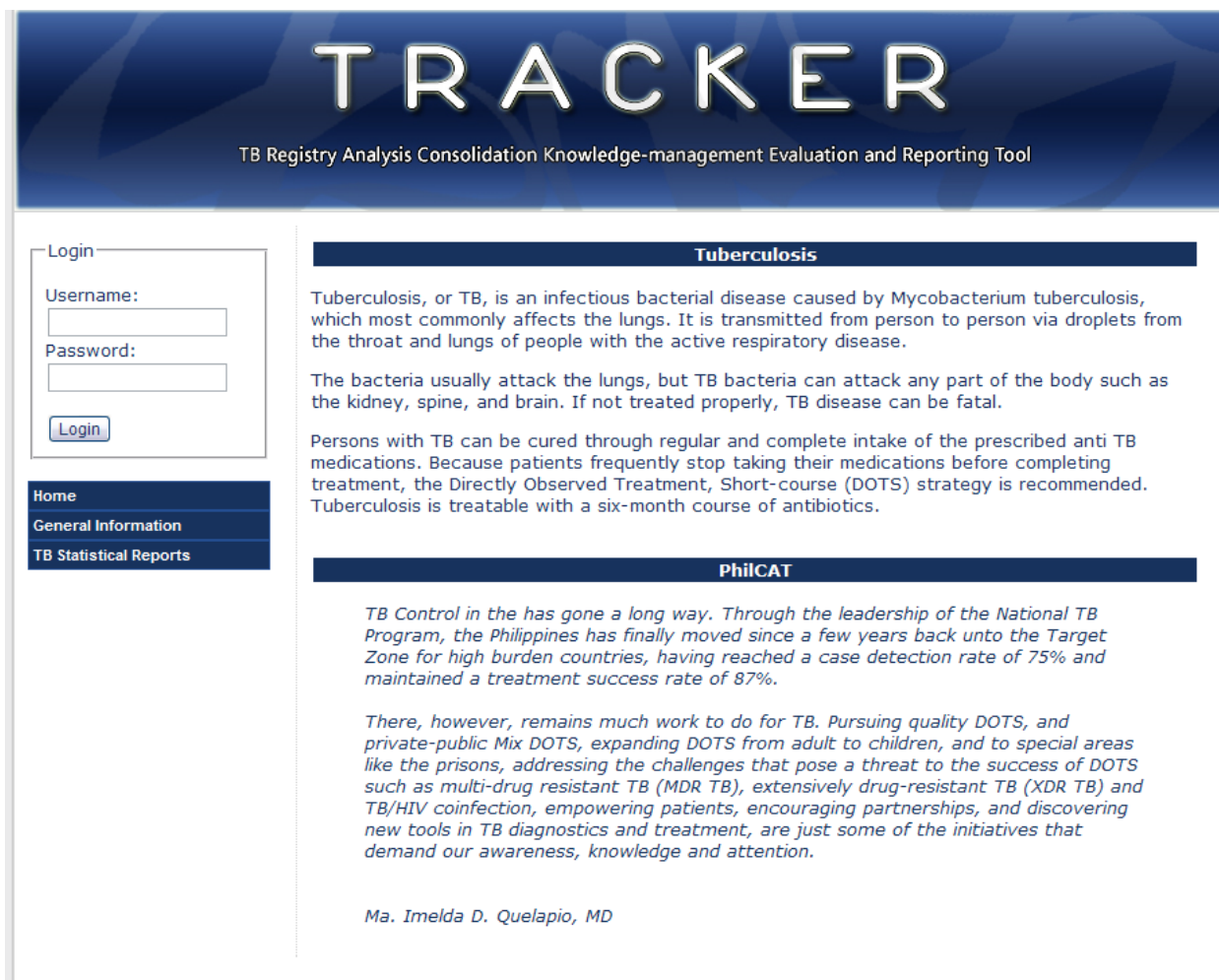


Figure 9. General Information Page, TRACKER

Users can also view TB statistical reports. An example is the Proportion of pulmonary smear positive cases out of all pulmonary cases. In the following figure, the user can select the year range on which the data is selected. A breakdown of data per sex can also be viewed on the same page.

MALE			
Age	1	2	3
0 - 14	0	0	0 %
15 - 24	0	0	0 %
25 - 34	0	0	0 %
35 - 44	0	0	0 %
45 - 54	0	0	0 %
55 - 64	0	0	0 %
65 above	0	0	0 %

FEMALE			
Age	1	2	3
0 - 14	0	0	0 %
15 - 24	0	0	0 %
25 - 34	0	0	0 %
35 - 44	0	0	0 %
45 - 54	0	0	0 %
55 - 64	0	0	0 %
65 above	0	0	0 %

1* Total number of Pulmonary smear positive cases (New and Relapse) registered
2* Total number of pulmonary (New smear-positive, New smear-negative and Relapse) registered
3* Proportion of Pulmonary smear cases out of all pulmonary cases

Figure 10. TB Statistics Generation Page, TRACKER

Figure 12. TB Statistics Results Page, TRACKER

Proportion of pulmonary smear-positive cases out of all pulmonary cases			
2000 to 2010			
Region	1	2	3
NCR	0	0	0 %
REGION 1	0	0	0 %
REGION 2	1	1	100 %
REGION 3	0	0	0 %
REGION 4A	0	0	0 %
REGION 4B	2	3	67 %
REGION 5	0	0	0 %
REGION 6	0	0	0 %
REGION 7	0	0	0 %
REGION 8	0	0	0 %
REGION 9	0	0	0 %
REGION 10	0	0	0 %
REGION 11	0	0	0 %
REGION 12	0	0	0 %
CAR	0	0	0 %
ARMM	0	0	0 %

Figure 11. TB Statistics Results Page, TRACKER

There is also a graphical representation of the data available for easier viewing and analysis of the TB data collated from a given region, province or locality.

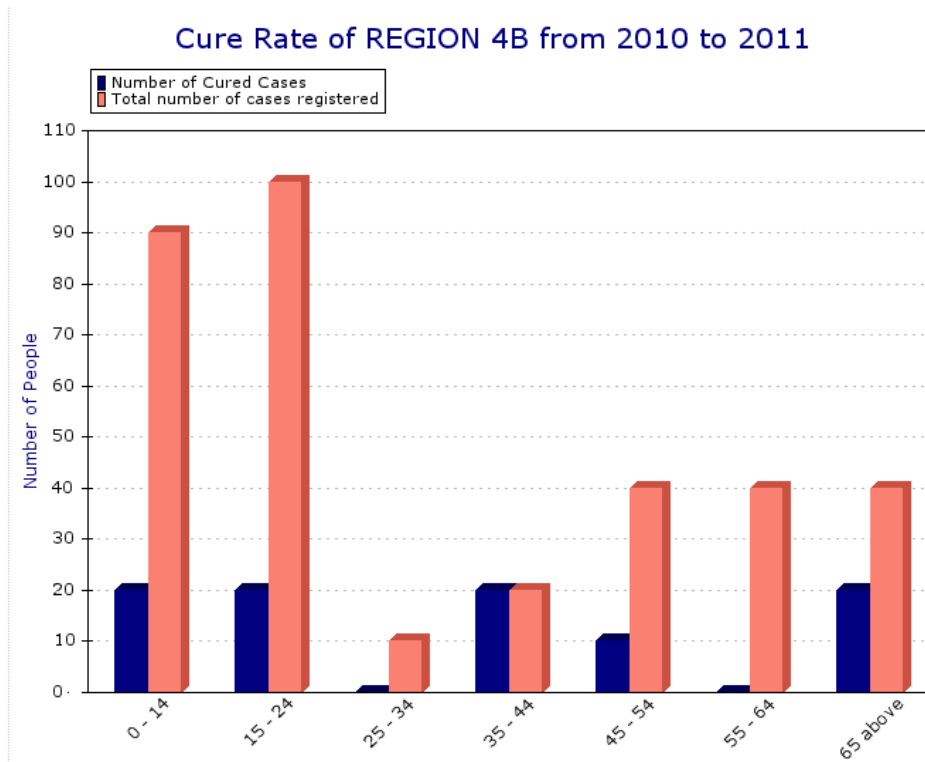


Figure 13. TB Statistics Results Page (Charts), TRACKER

Cure Rate from 2010 to 2011

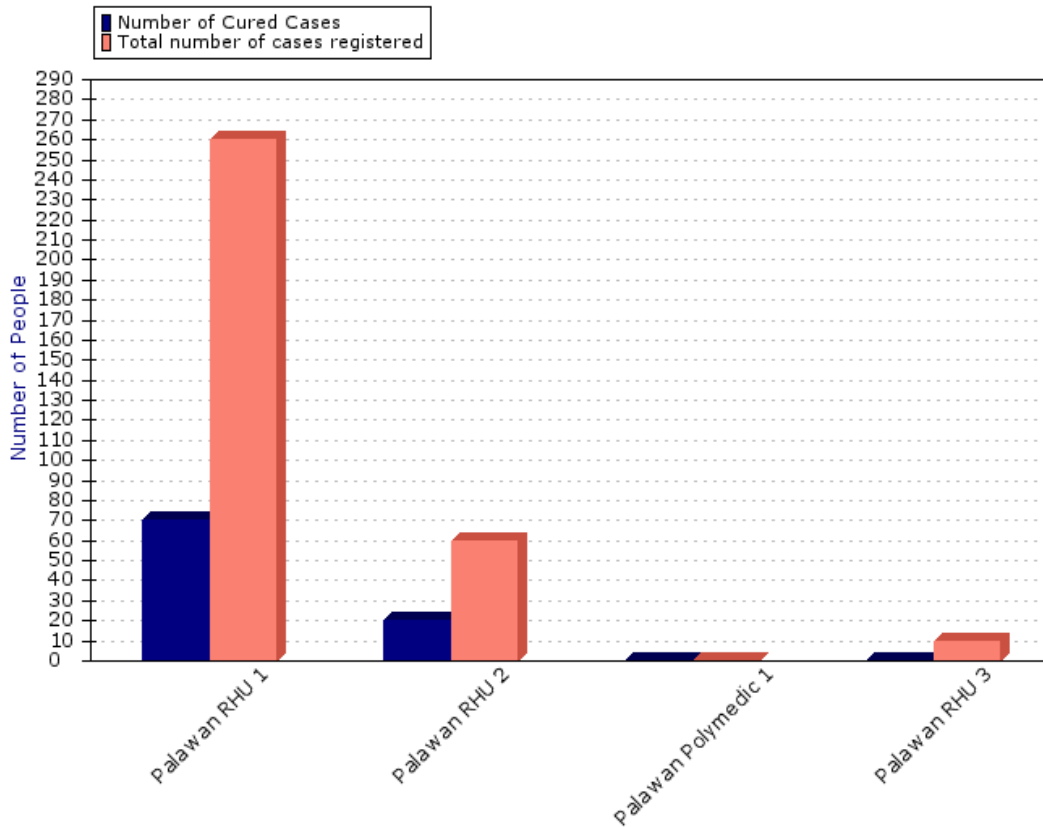


Figure 14. TB Statistics Results Page (Charts), TRACKER

The Provincial NTP Coordinator can also generate TB reports. They can generate reports specific to their health office and by the rural health units in within their locality. Figure 15 shows one of the two reports that a provincial health officer can generate.

TRACKER
TB Registry Analysis Consolidation Knowledge-management Evaluation and Reporting Tool

You are logged in as **bossman**. You are a Provincial NTP Coordinator
Palawan PHO

- Home
- Your Account
- User Accounts
- TB Registry
 - View Register
 - Create NTP Quarterly Report on Treatment Outcome
 - Create NTP Quarterly Report on all Cases
 - Create Counting Sheets
- Provincial Health Office
- Health Facility
- TB Statistical Reports

View Provincial NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases

Patients registered during the Quarter of

Reported this :

View Individual Health Unit NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases

TB Patients registered during the Quarter of

Patients Registered in :

Reported this :

Figure 15. Form for the NTP Quarterly Report on the treatment outcome of Pulmonary TB cases, TRACKER

The generated report can be downloaded as a PDF document or it can be viewed in a printer friendly format. Provincial coordinators can also edit the details of their profile and their health office, add new and add health units specific to their health office. (see Figures 16-17)

NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases								
Name of Region : REGION 4B		Patients registered		Date Reported : 2010-03-03				
Name of PHO/CHO : Palawan PHO		during the:		Prepared by : Boss Man				
Name of DOTS Facility : Palawan PHO		1st Quarter of Year 2010		Designation : Provincial NTP Coordinator				
Total Number of Pulmonary TB Cases	Type	Cured	Comp Tx	Died	Failed	Default	Trans Out	Total No. Evaluated
	1. New Cases							
5	1.1 Smear (+)	4	0	0	1	0	0	5
1	1.2 Smear (-)	0	1	0	0	0	0	1
	2. Re-treatment							
2	2.1 Relapse	0	1	0	0	0	1	2
11	2.2 Treatment Failure	1	1	8	1	0	0	11
0	2.3 Return After Default	0	0	0	0	0	0	0
Grand Total :								19
<p>* Of these, 2 were excluded from evaluation of chemotherapy for the following reasons:</p> <p style="padding-left: 40px;">Trans-in</p> <p style="padding-left: 40px;"><input checked="" type="checkbox"/> Extra-pulmonary</p> <p style="padding-left: 40px;">Other</p>								

[download report](#)

Figure 16. The Generated NTP Quarterly Report, TRACKER

NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases

Name of Region : REGION 4B		Patients registered		Date Reported : 2010-03-03				
Name of PHO/CHO : Palawan PHO		during the:		Prepared by : Boss Man				
Name of DOTS Facility : Palawan PHO		1st Quarter of Year 2010		Designation : Provincial NTP Coordinator				

Total Number of Pulmonary TB Cases	Type	Cured	Comp Tx	Died	Failed	Default	Trans Out	Total No. Evaluated
	1. New Cases							
5	1.1 Smear (+)	4	0	0	1	0	0	5
1	1.2 Smear (-)	0	1	0	0	0	0	1
	2. Re-treatment							
2	2.1 Relapse	0	1	0	0	0	1	2
11	2.2 Treatment Failure	1	1	8	1	0	0	11
0	2.3 Return After Default	0	0	0	0	0	0	0
Grand Total :								19

* Of these, **2** were excluded from evaluation of chemotherapy for the following reasons:

Trans-in

Extra-pulmonary

Other

Figure 17. The Generated NTP Quarterly Report in PDF, TRACKER

TRACKER
TB Registry Analysis Consolidation Knowledge-management Evaluation and Reporting Tool

You are logged in as **bossman**. You are a Provincial NTP Coordinator
Palawan PHO

- Home
- Your Account
- User Accounts
- TB Registry
- Provincial Health Office
- Health Facility
- TB Statistical Reports

Update Provincial Health Office Details

Provincial Health Office Details

Name :

Province :

Street / Village :

City / Municipality:

Contact Number:

Figure 18. Update Provincial Health Office Details, TRACKER

You are logged in as **bossman**. You are a Provincial NTP Coordinator
Palawan PHO

- Home
- Your Account
- User Accounts
- TB Registry
- Provincial Health Office
- Health Facility
- TB Statistical Reports

Create an Account

User Details

Username :
(min 6 characters)

Password:
(min 6 characters)

Re-type Password:

Designation: ▼

First name:

Middle name:

Last name:

Gender: Male Female

Figure 19. Create new users for the Health Office, TRACKER

Add Health Facility

Health Facility Details

ID :

Name :

Type of Facility: ▼

City / Municipality:

Street / Village :

Contact Number:

Figure 20. Add Health Facility for the Health Office, TRACKER

Once added to the system, nurses and encoders can now add TB patient record to the registry. The patient record is divided into two sections. The first section contains the personal information and initial diagnosis of the patient. The second section contains the treatment information and the sputum examination details. (see Figures 21-22)

Add TB Case

Patient TB Information

TB Case Number :

Date of Registration : (yyyy-mm-dd)

First Name :

MI :

Last Name :

City :

Age :

Sex : Male Female

DOTS Information

Name of DOTS Facility : ▼

Source of Patient : Public Private

Name of Physician :

Disease Classification : Pulmonary Extra Pulmonary

I. 6-SCC (2HRZE/4HR)

II. 8-SCC (2HRZE/1HRZE/5HRE)

III. 8-SCC (2HRZ/4HR)

Type of Patient : ▼

Figure 21. Add TB case, section one, Personal and DOTS, TRACKER

TB Case Number : **TBCASE13** Patient Name : **Ana Liz Van Der Wee**

Treatment Result

Date Treatment Started : (yyyy-mm-dd)
Treatment Outcome :
Date Treatment Ended : (yyyy-mm-dd)

Treatment Partner

Partner :
TBDC Review (Y/N) :
Remarks :

Sputum Examination Results / Weight

Treatment			
Month	Date Examined (yyyy-mm-dd)	Result	Weight
0	<input type="text" value="2010-02-01"/>	<input type="text" value="Doubtful"/>	<input type="text" value="12"/>
2nd	<input type="text" value="2010-03-03"/>	<input type="text" value="Negative"/>	<input type="text" value="32"/>
3rd	<input type="text"/>	<input type="text" value="-Select one-"/>	<input type="text"/>
4th	<input type="text"/>	<input type="text" value="-Select one-"/>	<input type="text"/>
5th	<input type="text"/>	<input type="text" value="-Select one-"/>	<input type="text"/>
6th	<input type="text"/>	<input type="text" value="-Select one-"/>	<input type="text"/>
> 7th	<input type="text"/>	<input type="text" value="-Select one-"/>	<input type="text"/>

<< previous section

add case

Figure 22. Add TB case, section two, Treatment and Sputum examination, TRACKER

Finally, Provincial Coordinators as well as the System Administrator can manage accounts of the system. However, the Provincial Coordinator can only manage users belonging to his health office. (see Figures 23-24)

You are logged in as **siegfrid**. You are a System Administrator

Your Account

Create New Account

View Users

Activate User Accounts

Deactivate User Accounts

Delete User Accounts

Manage Active User Accounts

Username	Usertype	Designation	Region	Status
user99	Encoder	Encoder	PHO 5	<input type="checkbox"/>
health1	Health Worker	Health Worker	PHO 5	<input type="checkbox"/>
nurse1	Nurse	Nurse	PHO 5	<input type="checkbox"/>
nurse2	Nurse	Nurse	PHO 2	<input type="checkbox"/>
philcat	PhilCAT Coordinator	PhilCAT Coordinator	PHO 10	<input type="checkbox"/>
sheila	Provincial NTP Coordinator	Provincial NTP Coordinator	PHO 5	<input type="checkbox"/>
user01	Provincial NTP Coordinator	Provincial NTP Coordinator	PHO 2	<input type="checkbox"/>
bossman	Provincial NTP Coordinator	Provincial NTP Coordinator	PHO 5	<input type="checkbox"/>

check all **uncheck all**

Figure 23. Manage Active Users, TRACKER

You are logged in as **siegfrid**. You are a System Administrator

Your Account

Create New Account

View Users

Activate User Accounts

Deactivate User Accounts

Delete User Accounts

Manage Inactive User Accounts

Username	Usertype	Designation	Region	Status
coord123	Provincial NTP Coordinator	Provincial NTP Coordinator	PHO 1	<input type="checkbox"/>

check all **uncheck all**

Figure 24. Manage Inactive Users, TRACKER

The system administrator can also manage user accounts of all users as well as delete them. Also, the system administrator is the only one who can create provincial coordinator accounts for each provincial health office. Like all user types, the system administrator can also update his own account. (see Figures 25-26)

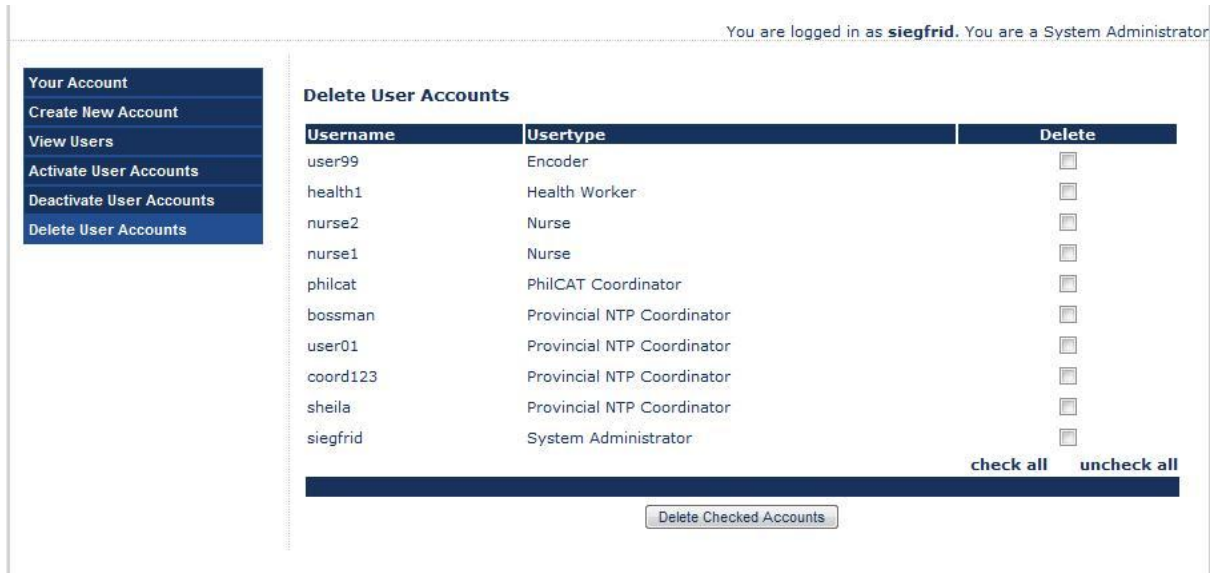


Figure 25. Delete Users, TRACKER



Figure 26. Update account profile, TRACKER

VI. Discussion

The Tuberculosis Registry Analysis, Consolidation, Knowledge-management, Evaluation and Reporting Tool (TRACKER) is a web based TB registry which allows storage of data based on the TB Registers. The TRACKER system has a number of users (guests or unregistered users, DOTS coordinator, PhilCAT coordinator, encoders and system administrator). Guests and other unregistered users can view statistics and other related data about TB. They can select the TB Statistical Reports option to view data about the whole country or a selected region.

The DOTS coordinator of a provincial health office and the PhilCAT coordinator have all the functionalities that the guest can access. In addition, he/she can view and create NTP quarterly reports as required by the DOH. These are the NTP Quarterly report on all cases and NTP quarterly report on the Treatment Outcome of TB Cases. Provincial coordinators can also update the information of their specific provincial health office.

Another type of user is the Encoders and Nurses. The encoders also have the functionalities as of that of a guest. Encoders are the ones tasked to encode the data from the TB registers from the local health units to the TRACKER system. They are the ones responsible for the integrity of data in the system. On the other hand, nurses/health workers have the same functions as that of an encoder and have also the Search Patient and Update Patient Record option. This feature enables them to correct/update any patient record that has been encoded incorrectly.

The system administrator has total control over managing active and inactive users of the system. However, unlike the other registered users, the system administrator cannot view nor generate reports and data about TB. They also cannot encode/input data to the TRACKER system.

Since the data is stored in a database, there is minimum risk of the information being lost due to handling errors. In addition to this, retrieval and analysis of data and the generation of reports is faster compared to the manual system because the need for manual counting, computation and report generation will be eliminated.

The system also automates the generation of NTP Quarterly reports such as (1) Quarterly report on All TB Cases and the (2) Quarterly Report on the Treatment Outcome of Pulmonary TB Cases. The system also previews the Counting Sheets for the said Quarterly Reports. The Provincial NTP Coordinator only needs to login and access the system to be able to generate reports specific to his health office. This is in contrast to the manual system, where nurses, have to go through each TB register and tally the data by hand to be able to generate the needed reports. Since the generation of the reports is faster, the coordinators can evaluate and make administrative decisions about TB control much faster.

The system also allows a partner NGO, PhilCAT to access its resources. PhilCAT is a major partner of the DOH in helping control TB in the Philippines. With PhilCAT having access to the system used by the DOH, it can actively participate in the formulation of plans, policies and guidelines in controlling TB in the Philippines.

The system also provides links to external web sites such as the Department of Health, World Health Organization and PhilCAT website.

However, the system only accepts and analyzes data based on the accomplished TB Registers on the Rural Health Unit level. Other data related to the patient are no longer included. Nonetheless, all information needed to generate reports for analysis is included in the system.

The TRACKER system target users, apart from the PhilCAT M & E are the ones at the provincial health office level. Unlike other patient records system which target users are mainly those on the ground or the field, TRACKER is easily deployable and implementable since only a small group of users will be using the system.

The TRACKER system eliminated the need for it to be deployed to a huge number of users since the people on the ground that are would-be users of the system are also busy doing the actual DOTS monitoring which may include a door-to-door visit of the patients. Also training, education and information dissemination to the staff that will be using the system will be minimized.

VII. Conclusion

The TRACKER system is a web-based disease registry system, specific to Tuberculosis, which allows nurses and encoders of specific health offices to input data on to the system and view TB related information, such as treatment outcomes, cure rate, death rate, among others. The system also allows the Provincial Coordinator as well as the PhilCAT analyst to generate, view, print and download reports.

For the Provincial Coordinators, they can only view and analyze TB data specific to their health facility, as well as manage the users designated to their health office. The system also provides the general users information about TB and also TB statistical reports.

The TRACKER system is only suited for the needs of the PhilCAT Monitoring and Evaluation (M & E) team. Features and functionalities of the system is defined and tailored in cooperation with members of the PhilCAT M & E. Users from other NGOs which are not affiliated with PhilCAT may not be able to use the system to its full potential.

Data stored in the TRACKER database only comes from the TB registers (see Appendix B). Other patient data related to TB collected on the barangay or municipal levels such as drug intake are not considered.

TRACKER is not a CHITS module and therefore cannot be integrated with the CHITS application.

VIII. Recommendations

This system is only suited for the needs of the PhilCAT Monitoring and Evaluation team and the Provincial Health Offices of the DOH. For future development, it is recommended that it can also suit the needs of the Regional Offices and also those offices beneath the Provincial Health Offices. Also, it is recommended that other partner NGOs of the DOH will be able to use this system as well.

The system currently only accepts information based on the TB Registers, other TB data gathered on the Rural and/or Municipal level should also be included for further analysis. Moreover, other medical records of the patients should also be included in the system because this can contribute to the analysis of the patient's treatment outcome.

Since this system is not part of the CHITS system nor it is a CHITS module, porting or further developing the system for it to become a CHITS module will be greatly beneficial and a welcome development to support the cause of improving and promoting telehealth in the country.

IX. Bibliography

1. World Health Organization. *Tuberculosis*. <http://www.who.int/topics/tuberculosis/en/>.
2. Centers for Disease Control and Prevention. <http://www.cdc.gov/tb/>. October 20, 2009.
3. Dateline Philippines. *75 Filipinos still die of tuberculosis every day*.
http://124.107.59.58/index.php?option=com_content&view=article&id=8&Itemid=14.
March 24, 2010
4. Billo, Nils E. and Lee, J.W “DOTS Expansion.” The Newsletter of the Global Partnership Movement to Stop TB. December 2001. pp 10
5. Department of Health. *Comprehensive and Unified Policy for TB Control in the Philippines*.
March 2003.
6. Department of Health. *Manual of Procedures for the National Tuberculosis Control Program*.
4th Ed. 2005.
7. Philippine Coalition Against Tuberculosis. <http://www.philcat.org/default/getpage/id/94/>.
8. National Telehealth Center. <http://plone.telehealth.ph:8081/NTHC>.
9. Sumabat, Kit. Email. October 16, 2009.
10. Sumabat, Kit. Personal communication. October 26, 2009.
11. Tolentino, Herman, et al. “Linking Primary Care Information Systems and Public Health Information Networks: Lessons from the Philippines.” Studies in Health Technology and Informatics. Vol. 116. pp 955-960. (2005).
12. Disease Registry. <http://training.seer.cancer.gov/glossary.html#DiseaseRegistry>.
13. Cancer Registration. *Types of Registries. Population-Based Registry*.
<http://training.seer.cancer.gov/registration/types/population.html>.
14. Cancer Registration. *Cancer Registry*. <http://training.seer.cancer.gov/registration/registry/>.
15. DesRoches, Catherine M., Campbell, Eric G. Ph, et al. “Electronic Health Records in Ambulatory Care — A National Survey of Physicians.” The New England Journal of Medicine. Vol. 359. pp 50-60. (July 2008).
16. Hummel, Jeffrey. “Building a Computerized Disease Registry for Chronic Illness Management of Diabetes.” Clinical Diabetes. 18.3 (2000).
17. Versweyveld, Leslie. “Clinicomp to install enterprise-wide clinical information system at Brooke Army medical Center.” Virtual Medical Worlds. (September 1999).

18. Bureau of Tuberculosis Control, New York City Department of Health and Mental Hygiene. "Controlling TB through Community Partnerships."
<http://www.ci.nyc.ny.us/html/doh/html/tb/tb.html>.
19. Pfeifer, Larry A. "A Summary of Hansen's Disease in the United States 2001."
http://bphc.hrsa.gov/nhdp/images/pdfs/hd_registry_report_2001_new.pdf.
20. Cimico, James J., et al. "Architecture for a Web-Based Clinical Information System that Keeps the Design Open and the Access Closed." Journal of the American Medical Informatics Association <https://www2.amia.org/pubs/symposia/d004694.pdf>.
21. Raymond, Brian and Dold, Cynthia. "Clinical Information Systems: Achieving The Vision" The Informatics Review. Vol. 6. No. 18. (September 2003).
22. Disease Registry. http://en.wikipedia.org/wiki/Disease_registry
23. Sittig, Dean F., Hazlehurst, Brian L., et al. "A Clinical Information System Research Landscape" The Permanente Journal Vol. 6. No. 2. (2002).
24. Darves, Bonnie "Patient registries: a key step to quality improvement." ACP Observer. September 2005. <http://www.acpinternist.org/archives/2005/09/patient.htm#help>
25. Information System. <http://www.atis.org/glossary/definition.aspx?id=4529>. October 16, 2009.
26. Information System.
<http://www.elsevier.com/wps/find/journaldescription.cwshome/236/description#description>
October 24, 2009.
27. Patient Registry. <http://www.umdf.org/patientregistry>
28. Patient Registry. The Myositis Association. http://www.myositis.org/patient_registry
29. Goss, Christopher H. and Parker H. W. "CF Patient Registry Serves as a Vital Resource to Improved Clinical Care." Homeline. <http://www.cff.org/images/customcontent/hlfeb02-ga.pdf>.
30. Acid Maltase Deficiency Association. "Patient Registry Information."
<http://www.members.aol.com/amdapage/registry.htm>.
31. Department of Health. *Operational Guidelines for Public-Private Mix DOTS in the Philippines*. 2004.

X. Appendix

Appendix A

NTP Quarterly Report on All TB Cases

Quarterly Report on All TB Cases

(Source of Data – TB Register)

Name of CHD:	Patients Registered during the Quarter of year
Name of PHO/CHO:	Date Submitted:
Name of DOTS facility:	Prepared by: _____ Name and Designation

A. All TB cases registered during the quarter:

Type of Patient/Sex/ DOTS facility	New Smear Positive (1)		Relapse (2)		Trans-in (3)		RAD (3)		Treatment Failure (5)		Other (6)			New Smear Negative (7)			Extra-pulmonary (8)	
	Positive		Negative		Positive		Negative		Positive		Positive			Negative			Positive	
	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F
Public																		
Private																		
Subtotal																		
Total																		

B. Breakdown of New Pulmonary smear-positive cases by age and sex:

0-9		10-14		15-24		25-34		35-44		45-54		55-64		65 and above		Total	
																M	F
M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F

* Source be the same as the total in Column 1 of A

C. Treatment Regimen given:

Type of Treatment Regimen	Category I	Category II	Category III
Cases initiated treatment			

Appendix C

NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases

NTP Quarterly Report on the Treatment Outcome of Pulmonary TB Cases (Source of Data – TB Register)

Name of Region: _____	Patients registered during the _____ Quarter of _____	Date Reported : _____
Name of PHO/CHO: _____		Prepared by : _____
Name of DOTS facility : _____		Designation: _____

Total Number of Pulmonary TB Cases (Copy the total number reported in the Case Finding Report during the same quarter)	Type	(1)	(2)	(3)	(4)	(5)	(6)	(7)	
		Cured	Completed Treatment	Died	Failed	Defaulted	Transferred out	Total No. Evaluated	
	1. New Cases								
	1.1 Smear (+)								
	1.2 Smear (-)								
	2. Re-treatment								
	2.1 Relapse								
	2.2 Treatment Failure								
	2.3 Return After Default								
Grand Total									

* Of these, _____(number) were excluded from evaluation of chemotherapy for the following reasons:

- _____ Trans-in
- _____ Extra-pulmonary
- _____ Other

Appendix D

Summary of the different TB patients and category of treatments

(12)

New Smear Positive		Relapse		Trans-in		Return After Default		Treatment Failure		Other Positive		Other Negative		Smear Negative		EP	
M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F	M	F

(13)

Total Number of Patients under the following Categories		
Category I	Category II	Category III

Appendix E

Counting sheet for the Quarterly report on the treatment outcome per type of patient

Type of Patient	Cured	Tx completed	Died	Failed	Defaulted	Trans-out
New Smear-Positive						
New Smear-Negative						
Relapse						
Treatment Failure						
Return After Default						

Appendix F

Source Codes

Controllers

Account Controller

```
class Account extends Controller {
var $SYSADUSER = 'System Administrator';
VAR $PROVUSER = 'Provincial NTP Coordinator';

function fetch_pho_name($data) {
    $rec = $this->db->query("SELECT * FROM pho
WHERE PHO_ID = '$data'");
    $rec = $rec->result();
    $name = $rec[0]->PHO_NAME;
    return $name;
}

function update_account() {
    if ($this->session->userdata('logged_in') == TRUE) {
        $this->update_validation(TRUE);
        if ($this->validation->run() == FALSE) {

            $this->load->model('account_model');
            $data['det'] = $this->account_model-
>fetch_profile($this->session-
>userdata('username'));
            $data['valid'] = FALSE;
            $this->layout-
>buildPage('account/update_account', $data);
        } else {
            $this->load->model('account_model');
            $_POST["flag"] = $this->account_model-
>edit_account($_POST);
            if ($_POST["flag"]) {
                $_POST["page"] = "Sorry, the system
encountered an error while updating your account.
Please try again.";
            } else {
                $data = array(
                    'username' => $_POST["username"],
                    'logged_in' => TRUE,
                    'name' => $_POST["first_name"],
                    'lname' => $_POST["last_name"],
                    'desig' => $this->session-
>userdata('desig'),
```

```
                    'usertype' => $this->session-
>userdata('usertype'),
                    'pho' => $this->session->userdata('pho'),
                    'pho_name' => $this->session-
>userdata('pho_name'),
                    'region' => $this->session-
>userdata('pho'),
                    'id' => $this->session->userdata('id')
                );
                $this->session->set_userdata($data);
                $_POST["page"] = "Account Successfully Updated.";
            }

            $this->layout->buildPage('welcome/status',
$_POST);
        }
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function view_account() {
    $data = $this->session->userdata('username');
    $this->load->model('account_model');
    $data = $this->account_model-
>fetch_profile($data);
    $this->layout->buildPage("account/profile",
$data);
}

function view_users() {
    $this->load->library('pagination');
    $this->load->helper('url');
    $base_url = site_url('account/view_users');

    $config['per_page'] = 10;
    $config['full_tag_open'] = '<p>';
    $config['full_tag_close'] = '</p>';
    $config['base_url'] = $base_url;
    $config['uri_segment'] = 3;

    if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
```

```

    $this->load->model('account_model');
    $data = $this->account_model-
>fetch_users($config['per_page'], $this->uri-
>segment(3));

    $config['total_rows'] = $this->account_model-
>fetch_users_count();

    $this->pagination->initialize($config);
    $this->layout->buildPage("account/view_users",
$data);
    } elseif ($this->session->userdata('usertype') ==
$this->PROVUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_province_users($config['per_page'], $this-
>uri->segment(3));

        $config['total_rows'] = $this->account_model-
>fetch_province_users_count();

        $this->pagination->initialize($config);
        $this->layout-
>buildPage("account/view_users_province", $data);
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function add_account() {
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER || $this->session->userdata('usertype')
== $this->PROVUSER) {

        $this->account_validation(TRUE);
        if ($this->validation->run() == FALSE) {
            $this->arr = $this->copy_arr();
            $this->load->model('account_model');
            $_POST["phos"] = $this->account_model-
>fetch_phos();
            $_POST["page"] = "Create an Account";

```

```

        $this->layout-
>buildPage('account/create_account', $_POST);
    } else {
        $this->load->model('account_model');
        if ($this->session->userdata('usertype') ==
$this->PROVUSER) {
            $count = $this->account_model-
>add_province_account($_POST);
        } else {
            $count = $this->account_model-
>add_account($_POST);
        }

        if ($count == 0) {
            $_POST["page"] = "Account Successfully
Created.";
            $this->layout->buildPage('welcome/status',
$_POST);

            $this->session->set_flashdata('message',
'<div id="success">Account Successfully
Created.</div>');
            redirect('welcome/status');
        } else {
            $this->session->set_flashdata('message',
'<div id="message">Username already exists. Please
try again.</div>');
            redirect('account/add_account');
        }
    }
} else {
    $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
    redirect('welcome/status');
}
}

function account_validation($flag) {
    if ($flag) {
        $rules['username'] =
"required|min_length[6]|max_length[20]";
        $rules['password'] =
"required|matches[password2]";
        $rules['password2'] =
"required|matches[password]";

```

```

    $rules['user_type'] = "required";
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
        $rules['region'] = "required";
    }
    $rules['first_name'] =
"required|max_length[30]";
    $rules['last_name'] =
"required|max_length[30]";
    $rules['gender'] = "required";
}
$this->validation->set_rules($rules);
if ($flag) {
    $fields['username'] = "Username";
    $fields['password'] = "Password";
    $fields['password2'] = "Password2";
    $fields['user_type'] = "Designation";
    $fields['first_name'] = "First Name";
    $fields['last_name'] = "Last Name";

```

```

    if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
        $fields['region'] = "Region";
    }
    $fields['gender'] = "Sex";
}
$this->validation->set_fields($fields);
}

```

```

function update_validation($flag) {
    if ($flag) {
        $rules['username'] =
"required|min_length[6]|max_length[20]";
        $rules['first_name'] =
"required|max_length[30]";
        $rules['last_name'] =
"required|max_length[30]";
        $rules['gender'] = "required";
    }
    $this->validation->set_rules($rules);
    if ($flag) {
        $fields['username'] = "Username";
        $fields['first_name'] = "First Name";
        $fields['last_name'] = "Last Name";
        $fields['gender'] = "Sex";
    }
}

```

```

    $this->validation->set_fields($fields);
}

```

```

function pass_validation($flag) {
    if ($flag == "a") {
        $rules['password'] = "required|min_length[6]";
        $rules['password1'] =
"required|matches[password2]";
        $rules['password2'] =
"required|matches[password1]";
    }
    $this->validation->set_rules($rules);
    if ($flag == "a") {
        $fields['password'] = "Current";
        $fields['password1'] = "New";
        $fields['password2'] = "Re-Type";
    }
    $this->validation->set_fields($fields);
}

```

```

function change_pass() {
    $this->pass_validation('a');
    if ($this->validation->run() == FALSE) {
        $this->session->set_flashdata('message', '<div
id="message">Please try again.</div>');
        $this->layout-
>buildPage('account/change_passwd');
    } else {
        $this->load->model('account_model');
        $data = $this->account_model-
>update_pass($_POST);

        if ($data["message"] == "Password Updated.") {
            $_POST["page"] = "Password Successfully
Updated";
            $this->layout->buildPage('welcome/status',
$_POST);

            $this->session->set_flashdata('message', '<div
id="success">Password Successfully
Updated.</div>');
            redirect('welcome/status');
        } else {
            $this->session->set_flashdata('message', '<div
id="message">Incorrect Password.</div>');

```

```

        $this->layout-
>buildPage('account/change_passwd');
    }
}

function manage_users() {
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_active_users();
        $this->layout-
>buildPage("account/active_account", $data);
    } elseif ($this->session->userdata('usertype') ==
$this->PROVUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_active_province_users();
        $this->layout-
>buildPage("account/active_account", $data);
    } else {
        $_POST["page"] = "You do not have enough
permissions to view this page.";
        $this->layout->buildPage('welcome/status',
$_POST);
    }
}

function manage_inactive_users() {
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_inactive_users();
        $this->layout-
>buildPage("account/inactive_account", $data);
    } elseif ($this->session->userdata('usertype') ==
$this->PROVUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_inactive_province_users();
        $this->layout-
>buildPage("account/inactive_account", $data);
    } else {

```

```

        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function activity() {
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER || $this->session->userdata('usertype')
== $this->PROVUSER) {
        $data = $_POST;
        $this->load->model('account_model');
        $this->account_model->deactivate($_POST);
        $_POST["page"] = "Accounts Successfully
Updated.";
        $this->layout->buildPage('welcome/status',
$_POST);
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function activity_off() {
    if ($this->session->userdata('usertype') == $this-
>SYSADUSER || $this->session->userdata('usertype')
== $this->PROVUSER) {
        $data = $_POST;
        $this->load->model('account_model');
        $this->account_model->activate($_POST);
        $_POST["page"] = "Accounts Successfully
Updated.";
        $this->layout->buildPage('welcome/status',
$_POST);
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function delete_users() {

```

```

$this->load->library('pagination');
$this->load->helper('url');
$base_url = site_url('account/delete_users');

$config['per_page'] = 10;
$config['full_tag_open'] = '<p>';
$config['full_tag_close'] = '</p>';
$config['base_url'] = $base_url;
$config['uri_segment'] = 3;

if ($this->session->userdata('usertype') == $this-
>SYSADUSER) {
    $this->load->model('account_model');
    $data = $this->account_model-
>fetch_users($config['per_page'], $this->uri-
>segment(3));

    $config['total_rows'] = $this->account_model-
>fetch_users_count();
    $this->pagination->initialize($config);

    $this->layout-
>buildPage("account/delete_account", $data);
    } elseif ($this->session->userdata('usertype') ==
$this->PROVUSER) {
        $this->load->model('account_model');
        $data = $this->account_model-
>fetch_province_users($config['per_page'], $this-
>uri->segment(3));

        $config['total_rows'] = $this->account_model-
>fetch_province_users_count();

        $this->pagination->initialize($config);
        $this->layout-
>buildPage("account/delete_account", $data);
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function delete_acc() {

```

```

if ($this->session->userdata('usertype') == $this-
>SYSADUSER || $this->session->userdata('usertype')
== $this->PROVUSER) {
    $data = $_POST;
    $this->load->model('account_model');
    $this->account_model-
>delete_account($_POST);
    $_POST["page"] = "Accounts Successfully
Deleted.";
    $this->layout->buildPage('welcome/status',
$_POST);
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

```

PHO Controller

```

class Pho extends Controller {
function view_phos() {
    $this->load->model('pho_model');
    $data = $this->pho_model->fetch_phos();
    $this->layout->buildPage("pho/view_pho", $data);
}

function add_pho() {
    $this->pho_validation(TRUE);
    if ($this->validation->run() == FALSE) {
        $_POST["page"] = "Add Provincial Health
Office";
        $this->layout->buildPage('pho/create_pho',
$_POST);
    } else {
        $this->load->model('pho_model');
        $count = $this->pho_model->add_pho($_POST);

        if ($count == 0) {
            $_POST["page"] = "Provincial Health Office
successfully created.";
            $this->layout->buildPage('welcome/status',
$_POST);

            $this->session->set_flashdata('message', '<div
id="success">Provincial Health Office ' .
$_POST["pho_name"] . ' successfully created.</div>');
        }
    }
}
}

```

```

        redirect('pho/add_pho');
    } else {
        $this->session->set_flashdata('message', '<div
id="message">Health Office with the same name
exists. Please try another name.</div>');
        redirect('pho/add_pho');
    }
}

function pho_validation($flag) {
    if ($flag) {
        $rules['pho_name'] = "required";
        $rules['pho_type'] = "required";
        $rules['pho_region'] = "required";
        $rules['pho_province'] = "required";
        $rules['pho_street'] = "required";
        $rules['pho_city'] = "required";
    }

    $this->validation->set_rules($rules);

    if ($flag) {
        $fields['pho_name'] = "Name";
        $fields['pho_type'] = "Type";
        $fields['pho_region'] = "Region";
        $fields['pho_province'] = "Province";
        $fields['pho_street'] = "Street / Village";
        $fields['pho_city'] = "City / Municipality";
    }
    $this->validation->set_fields($fields);
}

function update_validation($flag) {
    if ($flag) {
        $rules['pho_name'] = "required";
        $rules['pho_region'] = "required";
        $rules['pho_province'] = "required";
        $rules['pho_street'] = "required";
        $rules['pho_city'] = "required";
    }

    $this->validation->set_rules($rules);

    if ($flag) {
        $fields['pho_name'] = "Name";
        $fields['pho_region'] = "Region";
        $fields['pho_province'] = "Province";
        $fields['pho_street'] = "Street / Village";
        $fields['pho_city'] = "City / Municipality";
    }
    $this->validation->set_fields($fields);
}

function update_pho() {
    if ($this->session->userdata('logged_in') == TRUE )
    {
        $this->update_validation(TRUE);
        if ($this->validation->run() == FALSE) {
            $this->load->model('pho_model');
            $data['det'] = $this->pho_model-
>fetch_pho_profile($this->session->userdata('pho'));
            $data['valid'] = FALSE;
            $this->layout->buildPage('pho/update_pho',
            $data);
        } else {
            $this->load->model('pho_model');
            $success = $this->pho_model-
>edit_pho($_POST);
            if (!$success) {
                $this->session->set_flashdata('message',
                '<div id="message">Sorry, the system encountered
an error while updating your account. Please try
again.</div>');
            } else {
                $this->session-
>unset_userdata('pho_name');
                $this->session->set_userdata('pho_name',
                $_POST['pho_name']);
                $this->session->set_flashdata('message',
                '<div id="success">Provincial Health Office details
successfully updated.</div>');
            }
            redirect('welcome/status');
        }
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

```



```

function update_select_pho() {
    if ($this->session->userdata('logged_in') == TRUE )
    {
        $this->load->helper('url');
        $pho_id = $this->uri->segment(3);

        $this->update_validation(TRUE);
        if ($this->validation->run() == FALSE) {
            $this->load->model('pho_model');
            $data['det'] = $this->pho_model-
>fetch_pho_profile($pho_id);

            $this->layout->buildPage('pho/update_pho',
$data);
        } else {
            $this->load->model('pho_model');
            $success = $this->pho_model-
>edit_pho($_POST);
            if (!$success) {
                $this->session->set_flashdata('message',
'<div id="message">Sorry, the system encountered
an error while updating your account. Please try
again.</div>');
            } else {
                $this->session->set_flashdata('message',
'<div id="success">Provincial Health Office profile
successfully updated.</div>');
            }
            redirect('welcome/status');
        }
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}
?>

```

Rate Controller

```

class Rate extends Controller {

function display_smear1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');
    $_POST['smear'] = $this->rate_model-
>fetch_smear1($_POST);
    $_POST['smear2'] = $this->rate_model-
>fetch_smear2($_POST);
    $_POST['age1'] = $this->rate2_model-
>fetch_smear_age1($_POST);
    $_POST['age2'] = $this->rate2_model-
>fetch_smear_age2($_POST);

    $titles['title'] = 'Smear Rate';
    $titles['legend_1'] = 'Number of Smear Positive
Cases';
    $titles['legend_2'] = 'Total number of Pulmonary
cases';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_chart($_POST['smear'],
'smear1', $_POST['smear2'], 'smear2', $titles,
$range);

    $this->layout->buildPage('rate/smear_rate',
$data);
}

function display_cure1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');

    $_POST['stat'] = $this->rate_model-
>fetch_cure1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_cure_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);
}
}

```

```

    $titles['title'] = 'Cure Rate';
    $titles['legend_1'] = 'Number of Cured Cases';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_chart($_POST['stat'],
'cure1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout->buildPage('rate/cure_rate', $data);
}

```

```

function display_death1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');
    $_POST['stat'] = $this->rate_model-
>fetch_death1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_death_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Death Rate';
    $titles['legend_1'] = 'Number of cases who died
during the treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_chart($_POST['stat'],
'death1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout->buildPage('rate/death_rate',
$data);
}

```

```

function display_default1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');

```

```

    $_POST['stat'] = $this->rate_model-
>fetch_default1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_default_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Defaulter Rate';
    $titles['legend_1'] = 'Number of cases who are
defaulted';
    $titles['legend_2'] = 'Total number of cases
registered';

```

```

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_chart($_POST['stat'],
'default1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout->buildPage('rate/default_rate',
$data);
}

```

```

function display_fail1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');
    $_POST['stat'] = $this->rate_model-
>fetch_fail1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_fail_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Treatment Failure Rate';
    $titles['legend_1'] = 'Cases who are still positive at
five or more treatments';
    $titles['legend_2'] = 'Total number of cases
registered';

```

```

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

```

```

    $data = $this->display_chart($_POST['stat'], 'fail1',
    $_POST['stat2'], 'total', $titles, $range);

```

```

    $this->layout->buildPage('rate/fail_rate', $data);
}

```

```

function display_transfer1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');
    $_POST['stat'] = $this->rate_model-
>fetch_transfer1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_transfer_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);

```

```

    $titles['title'] = 'Transfer-out Rate';
    $titles['legend_1'] = 'Number of cases who
transferred to another facility';
    $titles['legend_2'] = 'Total number of cases
registered';

```

```

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

```

```

    $data = $this->display_chart($_POST['stat'],
'transfer1', $_POST['stat2'], 'total', $titles, $range);

```

```

    $this->layout-
>buildPage('rate/transfer_rate',$data);
}

```

```

function display_complete1() {
    $this->load->model('rate_model');
    $this->load->model('rate2_model');
    $_POST['stat'] = $this->rate_model-
>fetch_complete1($_POST);
    $_POST['stat2'] = $this->rate_model-
>fetch_total_case1($_POST);
    $_POST['stat3'] = $this->rate2_model-
>fetch_complete_age1($_POST);
    $_POST['stat4'] = $this->rate2_model-
>fetch_total_age1($_POST);

```

```

    $titles['title'] = 'Treatment Completion Rate';
    $titles['legend_1'] = 'Number of cases who
completed treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

```

```

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

```

```

    $data = $this->display_chart($_POST['stat'],
'complete1', $_POST['stat2'], 'total', $titles, $range);

```

```

    $this->layout-
>buildPage('rate/complete_rate',$data);
}

```

```

function display_smear2() {
    $this->load->model('rate3_model');
    $_POST['age1'] = $this->rate3_model-
>fetch_smear_age1($_POST);
    $_POST['age2'] = $this->rate3_model-
>fetch_smear_age2($_POST);

```

```

    $titles['title'] = 'Smear Rate';
    $titles['legend_1'] = 'Pulmonary smear positive
cases registered';
    $titles['legend_2'] = 'Total number of pulmonary
cases registered';

```

```

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

```

```

    $data = $this-
>display_region_chart($_POST['age1'],
$_POST['age2'], $titles, $range);

```

```

    $this->layout-
>buildPage('rate2/region_smear_rate',$data);
}

```

```

function display_cure2() {
    $this->load->model('rate3_model');
    $_POST['stat3'] = $this->rate3_model-
>fetch_cure_age1($_POST);

```

```

    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Cure Rate';
    $titles['legend_1'] = 'Number of Cured Cases';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_region_chart($_POST['stat3'],
$_POST['stat4'], $titles, $range);

    $this->layout-
>buildPage('rate2/region_cure_rate',$data);
}
function display_death2() {
    $this->load->model('rate3_model');
    $_POST['stat3'] = $this->rate3_model-
>fetch_death_age1($_POST);
    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Death Rate';
    $titles['legend_1'] = 'Number of cases who died
during the treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_region_chart($_POST['stat3'],
$_POST['stat4'], $titles, $range);

    $this->layout-
>buildPage('rate2/region_death_rate', $data);
}
function display_default2() {
    $this->load->model('rate3_model');

```

```

    $_POST['stat3'] = $this->rate3_model-
>fetch_default_age1($_POST);
    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Defaulter Rate';
    $titles['legend_1'] = 'Number of cases who are
defaulted';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_region_chart($_POST['stat3'],
$_POST['stat4'], $titles, $range);

    $this->layout-
>buildPage('rate2/region_default_rate', $data);
}
function display_fail2() {
    $this->load->model('rate3_model');
    $_POST['stat3'] = $this->rate3_model-
>fetch_fail_age1($_POST);
    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Treatment Failure Rate';
    $titles['legend_1'] = 'Cases who are still positive at
five or more treatments';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_region_chart($_POST['stat3'],
$_POST['stat4'], $titles, $range);

    $this->layout->buildPage('rate2/region_fail_rate',
$data);
}

```

```

function display_transfer2() {
    $this->load->model('rate3_model');
    $_POST['stat3'] = $this->rate3_model-
>fetch_transfer_age1($_POST);
    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Transfer-out Rate';
    $titles['legend_1'] = 'Number of cases who
transferred to another facility';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_region_chart
($_POST['stat3'], $_POST['stat4'], $titles, $range);

    $this->layout->buildPage
('rate2/region_transfer_rate', $data);
}
function display_complete2() {
    $this->load->model('rate3_model');
    $_POST['stat3'] = $this->rate3_model-
>fetch_complete_age1($_POST);
    $_POST['stat4'] = $this->rate3_model-
>fetch_total_age1($_POST);

    $titles['title'] = 'Treatment Completion Rate';
    $titles['legend_1'] = 'Number of cases who
completed treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['region'] = $_POST['region'];
    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_region_chart($_POST['stat3'],
$_POST['stat4'], $titles, $range);

    $this->layout-
>buildPage('rate2/region_complete_rate',$data);

```

```

}
function display_smear3() {
    $this->load->model('rate4_model');
    $_POST['smear'] = $this->rate4_model-
>fetch_smear1($_POST);
    $_POST['smear2'] = $this->rate4_model-
>fetch_smear2($_POST);

    $titles['title'] = 'Smear Rate';
    $titles['legend_1'] = 'Pulmonary smear positive
cases registered';
    $titles['legend_2'] = 'Total number of pulmonary
cases registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_province_chart
($_POST['smear'], 'smear1', $_POST['smear2'],
'smear2', $titles, $range);

    $this->layout-
>buildPage('rate3/province_smear_rate', $data);
}
function display_cure3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_cure1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Cure Rate';
    $titles['legend_1'] = 'Number of Cured Cases';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'cure1',
$_POST['stat2'], 'total', $titles, $range);

```

```

$this->layout-
>buildPage('rate3/province_cure_rate', $data);
}
function display_death3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_death1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Death Rate';
    $titles['legend_1'] = 'Number of cases who died
during the treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'death1',
$_POST['stat2'], 'total', $titles, $range);

    $this->layout-
>buildPage('rate3/province_death_rate',$data);
}
function display_default3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_default1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Defaulter Rate';
    $titles['legend_1'] = 'Number of cases who are
defaulted';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'default1',
$_POST['stat2'], 'total', $titles, $range);

```

```

$this->layout-
>buildPage('rate3/province_default_rate',$data);
}
function display_fail3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_fail1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Treatment Failure Rate';
    $titles['legend_1'] = 'Cases who are still positive at
five or more treatments';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'fail1',
$_POST['stat2'], 'total', $titles, $range);

    $this->layout->buildPage
('rate3/province_fail_rate',$data);
}
function display_transfer3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_transfer1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Transfer-out Rate';
    $titles['legend_1'] = 'Number of cases who
transferred to another facility';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'transfer1',
$_POST['stat2'], 'total', $titles, $range);

```

```

    $this->layout-
>buildPage('rate3/province_transfer_rate',$data);
}
function display_complete3() {
    $this->load->model('rate4_model');
    $_POST['stat'] = $this->rate4_model-
>fetch_complete1($_POST);
    $_POST['stat2'] = $this->rate4_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Treatment Completion Rate';
    $titles['legend_1'] = 'Number of cases who
completed treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_province_chart($_POST['stat'], 'complete1',
$_POST['stat2'], 'total', $titles, $range);

    $this->layout-
>buildPage('rate3/province_complete_rate',$data);
}

function display_smear4() {
    $this->load->model('rate5_model');
    $_POST['smear'] = $this->rate5_model-
>fetch_smear1($_POST);
    $_POST['smear2'] = $this->rate5_model-
>fetch_smear2($_POST);

    $titles['title'] = 'Smear Rate';
    $titles['legend_1'] = 'Pulmonary smear positive
cases registered';
    $titles['legend_2'] = 'Total number of pulmonary
cases registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this-
>display_pho_chart($_POST['smear'], 'smear1',
$_POST['smear2'], 'smear2', $titles, $range);

```

```

$this->layout->buildPage('rate4/pho_smear_rate',
$data);
}

function display_cure4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_cure1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Cure Rate';
    $titles['legend_1'] = 'Number of Cured Cases';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'cure1', $_POST['stat2'], 'total', $titles, $range);

$this->layout->buildPage('rate4/pho_cure_rate',
$data);
}

function display_death4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_death1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Death Rate';
    $titles['legend_1'] = 'Number of cases who died
during the treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'death1', $_POST['stat2'], 'total', $titles, $range);

```

```

    $this->layout->buildPage('rate4/pho_death_rate',
$data);
}
function display_default4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_default1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Defaulter Rate';
    $titles['legend_1'] = 'Number of cases who are
defaulted';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'default1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout-
>buildPage('rate4/pho_default_rate', $data);
}
function display_fail4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_fail1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Treatment Failure Rate';
    $titles['legend_1'] = 'Cases who are still positive at
five or more treatments';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'fail1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout->buildPage('rate4/pho_fail_rate',
$data);

```

```

}
function display_transfer4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_transfer1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Transfer-out Rate';
    $titles['legend_1'] = 'Number of cases who
transferred to another facility';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'transfer1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout-
>buildPage('rate4/pho_transfer_rate', $data);
}
function display_complete4() {
    $this->load->model('rate5_model');
    $_POST['stat'] = $this->rate5_model-
>fetch_complete1($_POST);
    $_POST['stat2'] = $this->rate5_model-
>fetch_total_case1($_POST);

    $titles['title'] = 'Treatment Completion Rate';
    $titles['legend_1'] = 'Number of cases who
completed treatment';
    $titles['legend_2'] = 'Total number of cases
registered';

    $range['to'] = $_POST['to'];
    $range['from'] = $_POST['from'];

    $data = $this->display_pho_chart($_POST['stat'],
'complete1', $_POST['stat2'], 'total', $titles, $range);

    $this->layout-
>buildPage('rate4/pho_complete_rate', $data);
}

```



```

function display_chart($cases, $casesName, $total,
$totalName, $titles, $range) {
    $data['to'] = $range['to'];
    $data['from'] = $range['from'];

    $graphData = array();

    foreach ($this->REGIONS as $region) {
        $data1val = array();

        $val1 = $cases[$casesName][$region];
        $val2 = $total[$totalName][$region];
        $rate = 0;

        if (empty($val1)) $val1 = 0;
        if (empty($val2)) $val2 = 0;

        if (is_numeric(stripos($region, 'region'))) {
            $region = substr($region, 7);
        }

        array_push($data1val, $region, $val1, $val2);
        array_push($graphData, $data1val);
    }

    $params = array('SetLegend' =>
array($titles['legend_1'], $titles['legend_2']),
        'SetPlotType' => 'bars'
    );

    $title = $titles['title'].' from '.$range['from'].'" to "
.$range['to'];
    $graphDetails = $this-
>get_graph_details($graphData, $params, $title,
$range);

    $data['graph_img2'] =
$graphDetails['graph_img2'];

    return $data;
}

function display_region_chart($cases, $total, $titles,
$range) {

```

```

    $data['to'] = $range['to'];
    $data['from'] = $range['from'];

    $graphData = array();

    for($i = 0; $i < 7; $i++)
    {
        $data1val = array();
        $val1 = $cases[$i]['M'] + $cases[$i]['F'];
        $val2 = $total[$i]['M'] + $total[$i]['F'];

        array_push($data1val, $this->AGES[$i], $val1,
$val2);
        array_push($graphData, $data1val);
    }

    $params = array('SetLegend' =>
array($titles['legend_1'], $titles['legend_2']),
        'SetPlotType' => 'bars'
    );

    $title = $titles['title'] . ' of '.$range['region'].' from
'.$range['from'].'" to " . $range['to'];
    $graphDetails = $this-
>get_graph_details($graphData, $params, $title,
$range);

    $data['graph_img2'] =
$graphDetails['graph_img2'];

    return $data;
}

function display_province_chart($cases, $caseName,
$total, $totalName, $titles, $range) {
    $data['to'] = $range['to'];
    $data['from'] = $range['from'];

    $graphData = array();

    $rhus = $cases['rhus'];
    $num = count($rhus);

    for ($i = 0; $i < $num; $i++) {
        $data1val = array();
        $val1 = $cases[$caseName][$i];

```

```

    $val2 = $totals[$totalName][$i];

    array_push($data1val, $rhus[$i]->RHU_NAME,
$val1, $val2);
    array_push($graphData, $data1val);
}

$params = array('SetLegend' =>
array($titles['legend_1'], $titles['legend_2']),
'SetPlotType' => 'bars'
);

$title = $titles['title'] . ' from ' . $range['from'] . " to " .
$range['to'];
$graphDetails = $this-
>get_graph_details($graphData, $params, $title,
$range);

$data['graph_img2'] =
$graphDetails['graph_img2'];

return $data;
}

function display_pho_chart($cases, $caseName,
$total, $totalName, $titles, $range) {
    $this->load->library('graph');

    $data['to'] = $range['to'];
    $data['from'] = $range['from'];

    $graphData = array();

    $rhus = $cases['phos'];
    $num = count($rhus);

    for ($i = 0; $i < $num; $i++) {
        $phoName = $rhus[$i]->PHO_NAME;
        $data1val = array();
        $val1 = $cases[$caseName][$phoName];
        $val2 = $totals[$totalName][$phoName];

        array_push($data1val, $phoName, $val1, $val2);
        array_push($graphData, $data1val);
    }
}

```

```

$params = array('SetLegend' =>
array($titles['legend_1'], $titles['legend_2']),
'SetPlotType' => 'bars'
);

$title = $titles['title'] . ' from ' . $range['from'] . " to " .
$range['to'];
$graphDetails = $this-
>get_graph_details($graphData, $params, $title,
$range);

$data['graph_img2'] =
$graphDetails['graph_img2'];

return $data;
}

function get_graph_details($data, $params, $title,
$range) {
    $this->load->library('graph');
    $this->graph->init(array ('width' => 650, 'height' =>
550));

    $this->graph->setdata($data, $params);

    $this->graph->obj->SetBackgroundColor('white');
    $this->graph->obj->SetTitleColor('navy');
    $this->graph->obj->SetTitle($title);
    $this->graph->obj->SetDataColors(array('navy',
'salmon'));
    $this->graph->obj-
>SetDrawPlotAreaBackground(TRUE);
    $this->graph->obj->SetLegendPixels(42, 35);
    $this->graph->obj->SetLegendStyle('left', 'left');
    $this->graph->obj->SetPlotAreaPixels(40, 80, 635,
NULL);
    $this->graph->obj->SetPlotBgColor('#ffffff');
    $this->graph->obj->SetYTitle('Number of People');
    $this->graph->obj->SetYTickIncrement(1);
    $this->graph->obj->SetXLabelAngle(45);
    // Graph generation
    $this->graph->draw();

    $graph_details['graph_img2'] = $this->graph-
>getimg();
}

```

```

return $graph_details;
}

function validate_fields($to, $from) {
    if ($to == $this->EMPTY_YR || $from == $this->EMPTY_YR) {
        if ($this->session->userdata('logged_in')) {
            $this->session->set_flashdata('message', '<div id="message">Don\'t leave any of the Year Range fields blank. Please try again.</div>');
            redirect('welcome/status');
        } else {
            $_POST["page"] = "Please don't leave any of the Year Range fields blank. Please try again.";
            $this->layout->buildPage('welcome/status', $_POST);
            return FALSE;
        }
    }
    return TRUE;
}

```

Report Controller

```
class Report extends Controller {

function create_report0() {
    $this->load->model('rhu_model');
    $this->load->model('pho_model');
    if($this->session->userdata('usertype') == 'System
Administrator' || $this->session-
>userdata('usertype') == 'PhilCAT Coordinator'
|| $this->session->userdata('usertype') ==
"DOH Coordinator") {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name();
        $_POST["phos"] = $this->pho_model-
>fetch_pho_name();
        $this->layout-
>buildPage('report/create_report0_philcat',
$_POST);
    } else {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name2();
        $this->layout-
>buildPage('report/create_report0', $_POST);
    }
}

function create_report1() {
    $this->load->model('rhu_model');
    $this->load->model('pho_model');
    if($this->session->userdata('usertype') == 'System
Administrator' || $this->session-
>userdata('usertype') == 'PhilCAT Coordinator'
|| $this->session->userdata('usertype') ==
"DOH Coordinator") {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name();
        $_POST["phos"] = $this->pho_model-
>fetch_pho_name();
        $this->layout-
>buildPage('report/create_report1_philcat',
$_POST);
    } else {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name2();
        $this->layout-
>buildPage('report/create_report1', $_POST);
    }
}
}
```

```
function create_report2() {
    $this->load->model('rhu_model');
    if($this->session->userdata('usertype') == 'System
Administrator' || $this->session-
>userdata('usertype') == 'PhilCAT Coordinator'
|| $this->session->userdata('usertype') ==
"DOH Coordinator") {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name();
    } else {
        $_POST["rhus"] = $this->rhu_model-
>fetch_rhu_name2();
    }
    $this->layout->buildPage('report/create_report2');
}
```

```
function display_quarter_report0() {
    $this->load->model('report_model');
    $_POST['report_out'] = $this->report_model-
>fetch_patient_type0($_POST);
    $_POST['ex'] = $this->report_model-
>fetch_EP($_POST);
    $_POST['cat1'] = $this->report_model-
>fetch_new($_POST);
    $_POST['cat2'] = $this->report_model-
>fetch_retreatment($_POST);
    $this->layout-
>buildPage('report/quarter_report0');
}
```

```
function display_quarter_report1() {
    $this->load->model('report_model');
    $_POST['all'] = $this->report_model-
>fetch_patient_type3($_POST);
    $_POST['cat'] = $this->report_model-
>fetch_category($_POST);
    $_POST['cat_age'] = $this->report_model-
>fetch_patient_type4($_POST);
    $this->layout-
>buildPage('report/quarter_report1');
}
```

```
function display_report2() {
    $this->load->model('report_model');
    $_POST['cat'] = $this->report_model-
>fetch_category($_POST);
    $_POST['crazy'] = $this->report_model-
>fetch_patient_type($_POST);
    $_POST['sex'] = $this->report_model-
>fetch_patient_type2($_POST);
    $this->layout-
>buildPage('report/category_report', $_POST);
}
```

```

function display_all_report2() {
    $this->load->model('report_model');
    $_POST['cat'] = $this->report_model-
>fetch_category($_POST);
    $_POST['crazy'] = $this->report_model-
>fetch_patient_type($_POST);
    $_POST['sex'] = $this->report_model-
>fetch_patient_type2($_POST);
    $this->layout-
>buildPage('report/category_report', $_POST);
}

function display_all_quarter_report0() {
    $this->load->model('report_model');
    $_POST['report_out'] = $this->report_model-
>fetch_all_patient_type0($_POST);
    $_POST['ex'] = $this->report_model-
>fetch_all_EP($_POST);
    $_POST['cat1'] = $this->report_model-
>fetch_all_new($_POST);
    $_POST['cat2'] = $this->report_model-
>fetch_all_retreatment($_POST);
    $this->arr = $this->copy_arr();
    $this->layout-
>buildPage('report/quarter_report0_province');
}

function display_all_quarter_report1() {
    $this->load->model('report_model');
    $_POST['all'] = $this->report_model-
>fetch_all_patient_type3($_POST);
    $_POST['cat'] = $this->report_model-
>fetch_all_category($_POST);
    $_POST['cat_age'] = $this->report_model-
>fetch_all_patient_type4($_POST);

    $this->layout-
>buildPage('report/quarter_report1_province');
}

function display_print_quarter_report0() {

require_once(dirname(__FILE__).'/3rdparty/html2pdf/html2pdf.class.php');
    $this->load->model('report_model');

    $_POST['report_out'] = $this->report_model-
>fetch_patient_type0($_POST);
    $_POST['ex'] = $this->report_model-
>fetch_EP($_POST);
    $_POST['cat1'] = $this->report_model-
>fetch_new($_POST);

```

```

    $_POST['cat2'] = $this->report_model-
>fetch_retreatment($_POST);

    $html = NULL;
    if(isset($_POST['mysubmit'])) {
        $html = $this->load-
>view('default/content/report/quarter_print_report
0', $_POST, TRUE);
    } else if(isset($_POST['mysubmit2'])) {
        $html = $this->load-
>view('default/content/report/print_report0',
$_POST, TRUE);
    }

    $html2pdf = new HTML2PDF('P','A4','en');
    $html2pdf->WriteHTML($html);
    $html2pdf->Output("Report_Outcome.pdf", "D");
}

function display_print_all_quarter_report0() {

require_once(dirname(__FILE__).'/3rdparty/html2pdf/html2pdf.class.php');
    $this->load->model('report_model');

    $_POST['report_out'] = $this->report_model-
>fetch_all_patient_type0($_POST);
    $_POST['ex'] = $this->report_model-
>fetch_all_EP($_POST);
    $_POST['cat1'] = $this->report_model-
>fetch_all_new($_POST);
    $_POST['cat2'] = $this->report_model-
>fetch_all_retreatment($_POST);
    $this->arr = $this->copy_arr();

    $html = NULL;
    if(isset($_POST['mysubmit'])) {
        $html = $this->load-
>view('default/content/report/quarter_print_report
0_province', $_POST, TRUE);
    } else if(isset($_POST['mysubmit2'])) {
        $html = $this->load-
>view('default/content/report/print_report0_provin
ce', $_POST, TRUE);
    }

    $html2pdf = new HTML2PDF('P','A4','en');
    $html2pdf->WriteHTML($html);
    $html2pdf->Output("Report_Outcome.pdf", "D");
}

function display_print_quarter_report1() {

require_once(dirname(__FILE__).'/3rdparty/html2pdf/html2pdf.class.php');

```

```

$this->load->model('report_model');

$_POST['all'] = $this->report_model-
>fetch_patient_type3($_POST);
$_POST['cat'] = $this->report_model-
>fetch_category($_POST);
$_POST['cat_age'] = $this->report_model-
>fetch_patient_type4($_POST);

$html = NULL;
if(isset($_POST['mysubmit'])) {
    $html = $this->load-
>view('default/content/report/quarter_print_report
1', $_POST, TRUE);
} else if(isset($_POST['mysubmit2'])) {
    $html = $this->load-
>view('default/content/report/print_report1',
$_POST, TRUE);
}

$html2pdf = new HTML2PDF('P','A4','en');
$html2pdf->WriteHTML($html);
$html2pdf->Output("Report_All_TB.pdf", "D");
}
function display_print_all_quarter_report1() {
    $this->load->model('report_model');

require_once(dirname(__FILE__).'/3rdparty/html2pdf
f/html2pdf.class.php');

$_POST['all'] = $this->report_model-
>fetch_all_patient_type3($_POST);
$_POST['cat'] = $this->report_model-
>fetch_all_category($_POST);
$_POST['cat_age'] = $this->report_model-
>fetch_all_patient_type4($_POST);
$this->arr = $this->copy_arr();

$html = NULL;
if(isset($_POST['mysubmit'])) {
    $html = $this->load-
>view('default/content/report/quarter_print_report
1_province', $_POST, TRUE);
} else if(isset($_POST['mysubmit2'])) {
    $html = $this->load-
>view('default/content/report/print_report1_provin
ce', $_POST, TRUE);
}

$html2pdf = new HTML2PDF('P','A4','en');
$html2pdf->WriteHTML($html);
$html2pdf->Output("Report_All_TB.pdf", "D");

```

RHU Controller

```

class Rhu extends Controller {

function view_rhus() {
    $this->load->model('rhu_model');
    if ($this->session->userdata('usertype') == "System
Administrator") {
        $data = $this->rhu_model->fetch_rhus();
    } else {
        $data = $this->rhu_model->fetch_rhus2();
    }
    $this->layout->buildPage("rhu/view_rhu", $data);
}

function add_rhu() {
    if ($this->session->userdata('logged_in') == TRUE) {
        $this->rhu_validation(TRUE);
        if ($this->validation->run() == FALSE) {

            $this->layout->buildPage('rhu/create_rhu', $_POST);
        } else {
            $_POST["pho_id"] = $this->session-
>userdata('pho');
            $this->load->model('rhu_model');
            $count = $this->rhu_model->add_rhu($_POST);

            if ($count == 0) {
                $this->session->set_flashdata('message',
'<div id="success">Health Facility ' .
$_POST["rhu_name"] . ' successfully
created.</div>');
                redirect('rhu/add_rhu');
            } else {
                $this->session->set_flashdata('message',
'<div id="message">Health Facility with the same
name exists. Please try another name.</div>');
                redirect('rhu/add_rhu');
            }
        }
    } else {
        $this->session->set_flashdata('message', '<div
id="message">You do not have enough permissions
to view this page.</div>');
        redirect('welcome/status');
    }
}

function rhu_validation($flag) {

    if ($flag) {
        $rules['rhu_name'] = "required";
        $rules['rhu_type'] = "required";
    }
}

```

```

    $rules['rhu_city'] = "required";
}

$this->validation->set_rules($rules);

if ($flag) {
    $fields['rhu_name'] = "Name";
    $fields['rhu_type'] = "Type";
    $fields['rhu_city'] = "City / Municipality";
}
$this->validation->set_fields($fields);
}

function update_rhu() {
    if ($this->session->userdata("logged_in") == TRUE) {
        $this->rhu_validation(TRUE);
        $rhu_id = $this->uri->segment(3);

        if ($this->validation->run() == FALSE) {
            $this->load->model('rhu_model');
            $data = $this->rhu_model-
>fetch_rhu_profile($rhu_id);
            $this->layout->buildPage('rhu/update_rhu',
            $data);
        } else {
            $_POST["pho_id"] = $this->session-
>userdata('pho');
            $this->load->model('rhu_model');
            $success = $this->rhu_model-
>update_rhu($_POST);

            if ($success) {
                $this->session->set_flashdata('message',
                '<div id="success">Health Facility ' .
                $_POST["rhu_name"] . ' successfully
                updated.</div>');
            } else {
                $this->session->set_flashdata('message',
                '<div id="message">Sorry, the system encountered
                an error while updating the recprd. Please try
                again.</div>');
            }

            redirect('welcome/status');
        }
    } else {
        $this->session->set_flashdata('message', '<div
        id="message">You do not have enough permissions
        to view this page.</div>');
        redirect('welcome/status');
    }
}

```

Models

Account Model

```

class Account_model extends Model {

    function fetch_users_count() {
        $user = $this->db->query("SELECT ID FROM
        `users` ORDER BY USERTYPE ASC");
        return count($user->result());
    }

    function fetch_province_users_count() {
        $pho = $this->session->userdata('pho');
        $user = $this->db->query("SELECT ID FROM
        `users` WHERE PHO_ID = $pho ORDER BY USERTYPE
        ASC");
        return count($user->result());
    }

    function fetch_users($num, $offset) {
        $query_str = (($offset == NULL) ? " LIMIT $num"
        : " LIMIT $num, $offset");
        $user = $this->db->query("SELECT USERNAME,
        USERTYPE, DESIGNATION, PHO_NAME FROM `users`
        LEFT JOIN `pho` ON users.PHO_ID = pho.PHO_ID
        ORDER BY USERTYPE ASC $query_str");
        return $user->result();
    }

    function fetch_active_users() {
        $user = $this->db->query("SELECT
        users.PHO_ID, `USERNAME`, `NAME`, `LAST_NAME`,
        `DESIGNATION`, `USERTYPE`, PHO_REGION FROM
        users LEFT JOIN pho ON users.PHO_ID = pho.PHO_ID
        WHERE STATUS = 'active' AND USERTYPE != 'System
        Administrator' ORDER BY USERNAME ASC");
        return $user->result();
    }

    function fetch_inactive_users() {
        $user = $this->db->query("SELECT
        users.PHO_ID, `USERNAME`, `NAME`, `LAST_NAME`,
        `DESIGNATION`, `USERTYPE`, PHO_REGION FROM
        users LEFT JOIN pho ON users.PHO_ID = pho.PHO_ID
        WHERE STATUS = 'inactive' AND USERTYPE !=
        'System Administrator' ORDER BY USERNAME ASC");
        return $user->result();
    }

    function fetch_province_users() {
        $pho = $this->session->userdata('pho');
        $user = $this->db->query("SELECT * FROM
        `users` LEFT JOIN `pho` ON users.PHO_ID =
        pho.PHO_ID WHERE pho.PHO_ID = $pho ORDER BY
        USERTYPE ASC");
    }
}

```

```

    return $user->result();
}
function fetch_active_province_users() {
    $pho = $this->session->userdata('pho');
    $user = $this->db->query("SELECT * FROM users
WHERE PHO_ID = $pho AND STATUS = 'active' AND
USERTYPE != 'System Administrator' ORDER BY
USERTYPE ASC");
    return $user->result();
}
function fetch_inactive_province_users() {
    $pho = $this->session->userdata('pho');
    $user = $this->db->query("SELECT * FROM users
WHERE PHO_ID = $pho AND STATUS = 'inactive' AND
USERTYPE != 'System Administrator' ORDER BY
USERTYPE ASC");
    return $user->result();
}

function fetch_profile($data) {
    $user = $this->db->query("SELECT * FROM users
WHERE USERNAME = '$data'");
    return $user->result();
}

function fetch_phos() {
    $user = $this->db->query("SELECT * FROM pho
ORDER BY PHO_REGION ASC");
    return $user->result();
}

function fetch_pho_id($data) {
    $user = $this->db->query("SELECT * FROM pho
WHERE PHO_ID = '$data'");
    return $user->result();
}

function isRecordsValid($data) {
    $this->db->where('USERNAME',
$data["username"]);
    $query = $this->db->get('users');
    $row = $query->result_array();
    return count($row);
}

function add_account($data) {
    $flag = $this->isRecordsValid($data);
    if ($flag == 0) {
        $sarr["USERNAME"] =
mysql_real_escape_string(stripslashes($data["usern
ame"]));

```

```

        $sarr["PASSWORD"] =
mysql_real_escape_string(stripslashes(md5($data["p
assword"])));
        $sarr["USERTYPE"] =
mysql_real_escape_string(stripslashes($data["user_t
ype"]));
        $sarr["DESIGNATION"] =
mysql_real_escape_string(stripslashes($data["user_t
ype"]));
        $sarr["STATUS"] = "active";

        if ($data["region"] == "DOH") {
            $sarr["PHO_ID"] = 0;
        } else {
            $sarr["PHO_ID"] =
mysql_real_escape_string(stripslashes($data["region
"]));
        }

        $sarr["NAME"] =
mysql_real_escape_string(stripslashes($data["first_n
ame"]));
        $sarr["MIDDLE_NAME"] =
mysql_real_escape_string(stripslashes($data["middl
e_name"]));
        $sarr["LAST_NAME"] =
mysql_real_escape_string(stripslashes($data["last_n
ame"]));
        $sarr["GENDER"] =
mysql_real_escape_string(stripslashes($data["gende
r"]));
        $result = $this->db->insert('users', $sarr);
    }
    return $flag;
}

function add_province_account($data) {
    $flag = $this->isRecordsValid($data);
    if ($flag == 0) {
        $sarr["USERNAME"] =
mysql_real_escape_string(stripslashes($data["usern
ame"]));
        $sarr["PASSWORD"] =
mysql_real_escape_string(stripslashes(md5($data["p
assword"])));
        $sarr["USERTYPE"] =
mysql_real_escape_string(stripslashes($data["user_t
ype"]));
        $sarr["DESIGNATION"] =
mysql_real_escape_string(stripslashes($data["user_t
ype"]));
        $sarr["STATUS"] = "active";

```



```

    $arr["PHO_ID"] = $this->session-
>userdata('pho');
    $arr["NAME"] =
mysql_real_escape_string(stripslashes($data["first_n
ame"]));
    $arr["MIDDLE_NAME"] =
mysql_real_escape_string(stripslashes($data["middl
e_name"]));
    $arr["LAST_NAME"] =
mysql_real_escape_string(stripslashes($data["last_n
ame"]));
    $arr["GENDER"] =
mysql_real_escape_string(stripslashes($data["gende
r"]));
    $result = $this->db->insert('users', $arr);
}
return $flag;
}
function edit_account($data) {
    $id = $this->session->userdata('id');

    $arr["USERNAME"] =
mysql_real_escape_string(stripslashes($data["usern
ame"]));

    $arr["NAME"] =
mysql_real_escape_string(stripslashes($data["first_n
ame"]));
    $arr["MIDDLE_NAME"] =
mysql_real_escape_string(stripslashes($data["middl
e_name"]));
    $arr["LAST_NAME"] =
mysql_real_escape_string(stripslashes($data["last_n
ame"]));
    $arr["GENDER"] =
mysql_real_escape_string(stripslashes($data["gende
r"]));

    $result = $this->db->update('users', $arr, "ID =
'$id'");

    return !$result;
}

function update_pass($data) {
    $username = $this->session-
>userdata('username');
    $password =
md5(stripslashes($data['password']));

    $rec = $this->db->query("SELECT * FROM users
WHERE USERNAME = '$username' AND STATUS
='active'");

```

```

    $rec = $rec->result();

    if ($rec[0]->PASSWORD == $password) {
        $arr["PASSWORD"] =
md5($data["password1"]);
        $result = $this->db->update('users', $arr,
"USERNAME = '$username'");
        $arr["message"] = "Password Updated.";
    } else {
        $arr["message"] = "Incorrect Password.";
    }
    return $arr;
}

function deactivate($data) {
    $usernum = $data['user_num'];
    for($i = 0; $i < $usernum; $i++) {
        $user = $data['change_stat'.$i];
        if (isset($data['checkbox'.$i])) {
            $arr["STATUS"] = "inactive";
            $this->db->update('users', $arr,
"USERNAME = '$user'");
        }
    }
    return $usernum;
}

function activate($data) {
    $usernum = $data['user_num'];
    for($i = 0; $i < $usernum; $i++) {
        $user = $data['change_stat'.$i];
        if (isset($data['checkbox'.$i])) {
            $arr["STATUS"] = "active";
            $this->db->update('users', $arr,
"USERNAME = '$user'");
        }
    }
}

function delete_account($data) {
    $usernum = $data['user_num'];
    for($i = 0; $i < $usernum; $i++) {
        $user = $data['del_stat'.$i];
        if (isset($data['checkbox'.$i])) {
            $this->db->where('USERNAME', $user);
            $this->db->delete('users');
        }
    }
}

```

PHO Model

```
class Pho_model extends Model {

function fetch_pho_profile($data) {
    $user = $this->db->query("SELECT * FROM pho
WHERE PHO_ID = '$data'");
    return $user->result();
}

function fetch_pho_name() {
    $user = $this->db->query("SELECT * FROM pho
ORDER BY PHO_REGION ASC");
    return $user->result();
}

function fetch_phos() {
    $user = $this->db->query("SELECT * FROM pho
ORDER BY PHO_REGION ASC");
    return $user->result();
}

function isRecordsValid($data) {
    $this->db->where('PHO_NAME',
$data["pho_name"]);
    $query = $this->db->get('pho');
    $row = $query->result_array();
    return count($row);
}

function add_pho($data) {
    $flag = $this->isRecordsValid($data);
    if ($flag == 0) {
        $arr["PHO_NAME"] =
mysql_real_escape_string(stripslashes($data["pho_n
ame"]));
        $arr["PHO_TYPE"] =
mysql_real_escape_string(stripslashes($data["pho_t
ype"]));
        $arr["PHO_REGION"] =
mysql_real_escape_string(stripslashes($data["pho_r
egion"]));
        $arr["PHO_PROVINCE"] =
mysql_real_escape_string(stripslashes($data["pho_p
rovince"]));
        $arr["PHO_STREET"] =
mysql_real_escape_string(stripslashes($data["pho_s
treet"]));
        $arr["PHO_CITY"] =
mysql_real_escape_string(stripslashes($data["pho_c
ity"]));
```

```
        $arr["PHO_CONTACT"] =
mysql_real_escape_string(stripslashes($data["pho_c
ontact"]));
        $arr["ACTIVE"] = "Y";
        $result = $this->db->insert('pho', $arr);
    }
    return $flag;
}

function edit_pho($data) {
    $id = $this->session->userdata('pho');

    $arr["PHO_NAME"] =
mysql_real_escape_string(stripslashes($data["pho_n
ame"]));
    $arr["PHO_PROVINCE"] =
mysql_real_escape_string(stripslashes($data["pho_p
rovince"]));
    $arr["PHO_STREET"] =
mysql_real_escape_string(stripslashes($data["pho_s
treet"]));
    $arr["PHO_CITY"] =
mysql_real_escape_string(stripslashes($data["pho_c
ity"]));
    $arr["PHO_CONTACT"] =
mysql_real_escape_string(stripslashes($data["pho_c
ontact"]));

    $result = $this->db->update('pho', $arr,
"PHO_ID = '$id'");

    return $result;
}
}
```

RHU Model

```
class Rhu_model extends Model {

function fetch_rhu_name() {
    $user = $this->db->query("SELECT RHU_NAME,
RHU_ID FROM rhu");
    return $user->result();
}

function fetch_rhu_name2() {
    $pho = $this->session->userdata('pho');
    $user = $this->db->query("SELECT RHU_NAME,
RHU_ID FROM rhu WHERE PHO_ID=$pho");
    return $user->result();
}

function fetch_rhus() {
    $user = $this->db->query("SELECT * FROM rhu");
```

```

return $user->result();
}

function fetch_rhus2() {
    $pho = $this->session->userdata('pho');
    $user = $this->db->query("SELECT * FROM rhu
WHERE PHO_ID=$pho");
    return $user->result();
}

function fetch_rhu_profile($data) {
    $user = $this->db->query("SELECT * FROM rhu
WHERE RHU_ID = '$data'");
    return $user->result();
}

function isRecordsValid($data) {
    $this->db->where('RHU_NAME',
$data['rhu_name']);
    $query = $this->db->get('rhu');
    $row = $query->result_array();
    return count($row);
}

function add_rhu($data) {
    $flag = $this->isRecordsValid($data);
    if ($flag == 0) {
        $sarr["RHU_NAME"] =
mysql_real_escape_string(stripslashes($data["rhu_n
ame"]));
        $sarr["RHU_TYPE"] =
mysql_real_escape_string(stripslashes($data["rhu_t
ype"]));
        $sarr["PHO_ID"] =
mysql_real_escape_string(stripslashes($data["pho_i
d"]));
        $sarr["RHU_STREET"] =
mysql_real_escape_string(stripslashes($data["rhu_st
reet"]));
        $sarr["RHU_CITY"] =
mysql_real_escape_string(stripslashes($data["rhu_ci
ty"]));
        $sarr["RHU_CONTACT"] =
mysql_real_escape_string(stripslashes($data["rhu_c
ontact"]));
        $sarr["ACTIVE"] = "Y";

        $result = $this->db->insert('rhu', $sarr);
    }
    return $flag;
}

function update_rhu($data) {

```

```

        $sarr["RHU_NAME"] =
mysql_real_escape_string(stripslashes($data["rhu_n
ame"]));
        $sarr["RHU_TYPE"] =
mysql_real_escape_string(stripslashes($data["rhu_t
ype"]));
        $sarr["PHO_ID"] =
mysql_real_escape_string(stripslashes($data["pho_i
d"]));
        $sarr["RHU_STREET"] =
mysql_real_escape_string(stripslashes($data["rhu_st
reet"]));
        $sarr["RHU_CITY"] =
mysql_real_escape_string(stripslashes($data["rhu_ci
ty"]));
        $sarr["RHU_CONTACT"] =
mysql_real_escape_string(stripslashes($data["rhu_c
ontact"]));
        $sarr["ACTIVE"] = "Y";
        $rhu_id =
mysql_real_escape_string(stripslashes($data["rhu_id
"]));

        $result = $this->db->update('rhu', $sarr, "RHU_ID =
'$rhu_id'");

        return $result;

```

Register Model

```

class Register_model extends Model {

function fetch_search_case($data, $num, $offset) {
    $pho = $this->session->userdata('pho');
    $query_str = "SELECT * FROM tb_register WHERE
PHO_ID = $pho";

    if ($data['tb_search']) {
        $query_str = $query_str . " AND TB_CASENO
LIKE '%" . $data['tb_search'] . "%'";
    }

    if ($data['search_name']) {
        $query_str = $query_str . " AND FNAME LIKE '%"
. $data['search_name'] . "%'";
    }

    if ($data['search_lname']) {
        $query_str = $query_str . " AND LNAME LIKE '%"
. $data['search_lname'] . "%'";
    }
}

```

```

    if ($data['gender'] != "X") {
        $query_str = $query_str . " AND SEX = '" .
        $data['gender'] . "'";
    }

    if ($offset == NULL) {
        $query_str = $query_str . " LIMIT $num";
    } else {
        $query_str = $query_str . " LIMIT $num,
        $offset";
    }

    $user = $this->db->query($query_str);
    return $user->result();
}

function fetch_search_case_count($data) {
    $pho = $this->session->userdata('pho');
    $query_str = "SELECT TB_CASENO FROM
    tb_register WHERE PHO_ID = $pho";

    if ($data['tb_search'] != NULL) {
        $query_str = $query_str . " AND TB_CASENO = '" .
        $data['tb_search'] . "'";
    }

    if ($data['search_name'] != NULL) {
        $query_str = $query_str . " AND FNAME LIKE '%" .
        $data['search_name'] . "%'";
    }

    if ($data['search_lname'] != NULL) {
        $query_str = $query_str . " AND LNAME LIKE '%" .
        $data['search_lname'] . "%'";
    }

    if ($data['gender'] != "X") {
        $query_str = $query_str . " AND SEX = '" .
        $data['gender'] . "'";
    }

    $query = $this->db->query($query_str);
    $row = $query->result_array();
    return count($row);
}

function fetch_case($data) {
    $pho = $this->session->userdata('pho');
    $query_str = "SELECT * FROM tb_register WHERE
    PHO_ID = $pho AND TB_CASENO = '$data'";
    $user = $this->db->query($query_str);
    return $user->result();
}

```

```

function fetch_phos() {
    $user = $this->db->query("SELECT * FROM pho");
    return $user->result();
}

function fetch_province_cases_count() {
    $pho = $this->session->userdata('pho');
    $query = $this->db->query("SELECT TB_CASENO
    FROM tb_register WHERE PHO_ID = $pho");
    $row = $query->result_array();
    return count($row);
}

function fetch_province_cases($num, $offset) {
    if ($offset == NULL) {
        $pho = $this->session->userdata('pho');
        $user = $this->db->query("SELECT * FROM
        tb_register WHERE PHO_ID = $pho LIMIT $num");
        return $user->result();
    }

    $pho = $this->session->userdata('pho');
    $user = $this->db->query("SELECT * FROM
    tb_register WHERE PHO_ID = $pho LIMIT $num,
    $offset");
    return $user->result();
}

function fetch_sputum_exam($data) {
    $user = $this->db->query("SELECT * FROM
    sputum_exam WHERE TB_CASENO = '$data' ORDER
    BY MONTH ASC");
    return $user->result();
}

function fetch_rhu_id($data) {
    $user = $this->db->query("SELECT RHU_ID FROM
    rhu WHERE RHU_NAME = '$data'");
    return $user->result();
}

function isRecordsValid($data) {
    $this->db->where('TB_CASENO', $data['case_no']);
    $query = $this->db->get('tb_register');
    $row = $query->result_array();
    return count($row);
}

function add_register($data) {
    $flag = $this->isRecordsValid($data);
    if ($flag == 0) {

```

```

    $rhu_id = $this-
>fetch_rhu_id($data['health_fac']);
    $arr['PHO_ID'] =
mysql_real_escape_string(stripslashes($this-
>session->userdata('pho')));
    $arr['TB_CASENO'] =
mysql_real_escape_string(stripslashes($data["case_
no"]));
    $arr['RHU_ID'] =
mysql_real_escape_string(stripslashes($rhu_id[0]-
>RHU_ID));
    $arr['FNAME'] =
mysql_real_escape_string(stripslashes($data['patien
t_name']));
    $arr['MI'] =
mysql_real_escape_string(stripslashes($data['patien
t_mi']));
    $arr['LNAME'] =
mysql_real_escape_string(stripslashes($data['patien
t_lname']));
    $arr['P_CITY'] =
mysql_real_escape_string(stripslashes($data['patien
t_add']));
    $arr['AGE'] =
mysql_real_escape_string(stripslashes($data['age']))
;
    $arr['SEX'] =
mysql_real_escape_string(stripslashes($data['patien
t_sex']));
    $arr['HEALTH_FAC'] =
mysql_real_escape_string(stripslashes($data['health
_fac']));
    $arr['DATE_OF_REG'] =
mysql_real_escape_string(stripslashes($data['reg_da
te']));

    $arr['P_SOURCE'] =
mysql_real_escape_string(stripslashes($data['patien
t_source']));
    $arr['DOC_NAME'] =
mysql_real_escape_string(stripslashes($data['doc_n
ame']));
    $arr['TB_CLASS'] =
mysql_real_escape_string(stripslashes($data['tb_cla
ss']));
    $arr['P_TYPE'] =
mysql_real_escape_string(stripslashes($data['patien
t_type']));
    $arr['P_CATEGORY'] =
mysql_real_escape_string(stripslashes($data['catego
ry']));

```

```

    $arr['START_DATE'] =
mysql_real_escape_string(stripslashes($data['start_
date']));
    $arr['OUTCOME'] =
mysql_real_escape_string(stripslashes($data['treat_
out']));
    $arr['END_DATE'] =
mysql_real_escape_string(stripslashes($data['end_d
ate']));
    $arr['PARTNER'] =
mysql_real_escape_string(stripslashes($data['partne
r']));
    $arr['REVIEW'] =
mysql_real_escape_string(stripslashes($data['review
']));
    $arr['REMARKS'] =
mysql_real_escape_string(stripslashes($data['remar
ks']));

    $result = $this->db->insert('tb_register', $arr);

    $arr = "";
    for ($i = 1; $i <= 7; $i++) {
        if ($data['exam_date' . $i] != null &&
        $data['result' . $i] != null) {
            $arr['TB_CASENO'] =
mysql_real_escape_string(stripslashes($data['case_n
o']));
            $arr['MONTH'] =
mysql_real_escape_string(stripslashes($data['month
' . $i]));
            $arr['DATE_EXAM'] =
mysql_real_escape_string(stripslashes($data['exam_
date' . $i]));
            $arr['WEIGHT'] =
mysql_real_escape_string(stripslashes($data['weight
' . $i]));
            $arr['RESULT'] =
mysql_real_escape_string(stripslashes($data['result
' . $i]));

            $result = $this->db->insert('sputum_exam',
            $arr);
        }
    }
    return $flag;
}

function update_register($data) {
    $rhu_id = $this->fetch_rhu_id($data['health_fac']);
    $old_caseno = $data['old_case_no'];

```

```

    $arr['PHO_ID'] =
mysql_real_escape_string(stripslashes($this-
>session->userdata('pho')));
    $arr['TB_CASENO'] =
mysql_real_escape_string(stripslashes($data['case_n
o']));
    $arr['RHU_ID'] =
mysql_real_escape_string(stripslashes($rhu_id[0]-
>RHU_ID));
    $arr['FNAME'] =
mysql_real_escape_string(stripslashes($data['patien
t_name']));
    $arr['MI'] =
mysql_real_escape_string(stripslashes($data['patien
t_mi']));
    $arr['LNAME'] =
mysql_real_escape_string(stripslashes($data['patien
t_lname']));
    $arr['P_CITY'] =
mysql_real_escape_string(stripslashes($data['patien
t_add']));
    $arr['AGE'] =
mysql_real_escape_string(stripslashes($data['age']))
;
    $arr['SEX'] =
mysql_real_escape_string(stripslashes($data['patien
t_sex']));
    $arr['HEALTH_FAC'] =
mysql_real_escape_string(stripslashes($data['health
_fac']));
    $arr['DATE_OF_REG'] =
mysql_real_escape_string(stripslashes($data['reg_da
te']));

    $arr['P_SOURCE'] =
mysql_real_escape_string(stripslashes($data['patien
t_source']));
    $arr['DOC_NAME'] =
mysql_real_escape_string(stripslashes($data['doc_n
ame']));
    $arr['TB_CLASS'] =
mysql_real_escape_string(stripslashes($data['tb_cla
ss']));
    $arr['P_TYPE'] =
mysql_real_escape_string(stripslashes($data['patien
t_type']));
    $arr['P_CATEGORY'] =
mysql_real_escape_string(stripslashes($data['catego
ry']));

    $arr['START_DATE'] =
mysql_real_escape_string(stripslashes($data['start_
date']));

```

```

    $arr['OUTCOME'] =
mysql_real_escape_string(stripslashes($data['treat_
out']));
    $arr['END_DATE'] =
mysql_real_escape_string(stripslashes($data['end_d
ate']));
    $arr['PARTNER'] =
mysql_real_escape_string(stripslashes($data['partne
r']));
    $arr['REVIEW'] =
mysql_real_escape_string(stripslashes($data['review
']));
    $arr['REMARKS'] =
mysql_real_escape_string(stripslashes($data['rema
rks']));

    $result = $this->db->update('tb_register', $arr,
"TB_CASENO = '$old_caseno'");

    $arr = "";
    for ($i = 1; $i <= 7; $i++) {
        if ($data['exam_date' . $i] != null &&
$data['result' . $i] != null) {
            $arr['TB_CASENO'] =
mysql_real_escape_string(stripslashes($data['case_n
o']));
            $arr['MONTH'] =
mysql_real_escape_string(stripslashes($data['month
' . $i]));
            $arr['DATE_EXAM'] =
mysql_real_escape_string(stripslashes($data['exam_
date' . $i]));
            $arr['WEIGHT'] =
mysql_real_escape_string(stripslashes($data['weight
' . $i]));
            $arr['RESULT'] =
mysql_real_escape_string(stripslashes($data['result
' . $i]));

            $month = $arr['MONTH'];

            $result = $this->db->update('sputum_exam',
$arr, "TB_CASENO = '$old_caseno' AND MONTH =
'$month'");
        }
    }
    return $result;

```

Rate Model

```
class Rate_model extends Model {

function fetch_smear1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE (P_TYPE = 'New Pos' OR P_TYPE
= 'Relapse') AND pho.PHO_REGION = '$region'
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31') AND
TB_CLASS = 'P'");
            $cat['smear1'][$region] = count($query-
>result());
        }
        return $cat;
    }
}

function fetch_smear2($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE (P_TYPE = 'New Pos' OR P_TYPE
= 'Relapse' OR P_TYPE = 'New Neg') AND
pho.PHO_REGION = '$region'
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31') AND
TB_CLASS = 'P'");
            $cat['smear2'][$region] = count($query-
>result());
        }
        return $cat;
    }
}

function fetch_cure1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
```

```
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Cured' AND
pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31') ");
            $cat['cure1'][$region] = count($query-
>result());
        }
        return $cat;
    }
}

function fetch_total_case1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE pho.PHO_REGION = '$region'
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31')");
            $cat['total'][$region] = count($query-
>result());
        }
        return $cat;
    }
}

function fetch_death1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Died' AND
pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
```

```

        AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31');
        $cat['death1'][$region] = count($query-
>result());
    }
    return $cat;
}
}

```

```

function fetch_default1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Default' AND
pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31' ");
            $cat['default1'][$region] = count($query-
>result());
        }
    }
    return $cat;
}
}

```

```

function fetch_fail1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Failed' AND
pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31');
            $cat['fail1'][$region] = count($query-
>result());
        }
    }
}

```

```

        return $cat;
    }
}

```

```

function fetch_complete1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Treatment
Completed' AND pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31' ");
            $cat['complete1'][$region] = count($query-
>result());
        }
    }
    return $cat;
}
}

```

```

function fetch_transfer1($data) {
    $year1 = $data['from'];
    $year2 = $data['to'];

    if ($year1 != null && $year2 != null) {
        foreach ($this->REGIONS as $region) {
            $query = $this->db->query("SELECT
TB_CASENO FROM tb_register
LEFT JOIN pho ON tb_register.PHO_ID =
pho.PHO_ID
WHERE OUTCOME = 'Transferred Out'
AND pho.PHO_REGION = '$region'
AND (P_TYPE = 'New Pos' OR P_TYPE =
'Relapse' OR P_TYPE = 'New Neg' OR P_TYPE =
'Treatment Failure')
AND (DATE_OF_REG >= '$year1-01-01'
AND DATE_OF_REG <= '$year2-12-31');
            $cat['transfer1'][$region] = count($query-
>result());
        }
    }
    return $cat;
}
}

```


Layout Model

```
class Layout_model extends Model {
    var $common;
    var $theme;

    /**
     * Build the menu
     */
    function menu() {
        $retval['menu'] = array();
        if ($this->session->userdata('usertype') ==
"System Administrator") {
            $retval['menu'] = array(

                anchor('dashboard/index', 'Your
Account')=> array(
                    anchor('account/update_account',
'Update Account'),
                    anchor('account/change_pass',
'Change Password'),
                    anchor('login/logout', 'Log out')
                ),
                anchor('account/add_account', 'Create
New Account'),
                anchor('account/view_users', 'View
Users'),
                anchor('account/manage_inactive_users',
'Activate User Accounts'),
                anchor('account/manage_users',
'Deactivate User Accounts'),
                anchor('account/delete_users', 'Delete
User Accounts'),
                anchor("", 'Provincial Health Units') =>
array(
                    anchor('pho/add_pho', 'Add
Provincial Unit'),
                    anchor('pho/view_phos', 'View
Provincial Units'),
                )
            );
        } elseif ($this->session->userdata('usertype') ==
"Nurse" || $this->session->userdata('usertype') ==
"Encoder"
            || $this->session->userdata('usertype') ==
"Health Worker") {
            $retval['menu'] = array(
                anchor('#', 'Home'),
                anchor('dashboard/index', 'Your
Account')=> array(
                    anchor('account/update_account',
'Update Account'),
```

```
                    anchor('account/change_pass',
'Change Password'),
                    anchor('login/logout', 'Log out')
                ),
                anchor("", 'TB Registry') => array(
                    anchor('register/add_case', 'Encode
TB Case'),
                    anchor('register/view_register',
'View Register'),
                    anchor('register/view_patient', 'Search
Patient')
                ),
                anchor("", 'Health Facility') => array(
                    anchor('rhu/add_rhu', 'Add Health
Facility'),
                    anchor('rhu/view_rhus', 'View Health
Facility List'),
                ),
                anchor("", 'TB Statistical Reports') => array(
                    anchor('stats/create_smear2',
'Proportion of Pulmonary Smear-Positive Cases'),
                    anchor('stats/create_cure2', 'Cure
Rate'),
                    anchor('stats/create_complete2',
'Completion Rate'),
                    anchor('stats/create_death2', 'Death
Rate'),
                    anchor('stats/create_fail2',
'Treatment Failure Rate'),
                    anchor('stats/create_default2',
'Defaulter Rate'),
                    anchor('stats/create_transfer2',
'Transfer-out Rate'),
                )
            );
        } elseif ($this->session->userdata('usertype') ==
"Provincial NTP Coordinator") {
            $retval['menu'] = array(
                anchor('#', 'Home'),
                anchor('dashboard/index', 'Your
Account')=> array(
                    anchor('account/update_account',
'Update Account'),
                    anchor('account/change_pass',
'Change Password'),
                    anchor('login/logout', 'Log out')
                ),
                anchor('dashboard/index', ' User
Accounts') => array(
                    anchor('account/add_account',
'Create New Account'),
                    anchor('account/view_users', 'View
Users'),
```

```

anchor('account/manage_inactive_users', 'Activate
User Accounts'),
    anchor('account/manage_users',
'Deactivate User Accounts')),
    anchor("", 'TB Registry') => array(
    anchor('register/view_register', 'View
Register'),
    anchor('report/create_report0', 'Create
NTP Quarterly Report on Treatment Outcome'),
    anchor('report/create_report1', 'Create
NTP Quarterly Report on all Cases'),
    anchor('report/create_report2', 'Create
Counting Sheets')
    ),
    anchor("", 'Provincial Health Office') =>
array(
    anchor('pho/update_pho', 'Update
Provincial Health Office Profile'),
    ),
    anchor("", 'Health Facility') => array(
    anchor('rhu/add_rhu', 'Add Health
Facility'),
    anchor('rhu/view_rhus', 'View Health
Facility List'),
    ),
    anchor("", 'TB Statistical Reports') => array(
    anchor('stats/create_smear2', 'Proportion
of Pulmonary Smear-Positive Cases'),
    anchor('stats/create_cure2', 'Cure Rate'),
    anchor('stats/create_complete2',
'Completion Rate'),
    anchor('stats/create_death2', 'Death Rate'),
    anchor('stats/create_fail2', 'Treatment
Failure Rate'),
    anchor('stats/create_default2',
'Defaulter Rate'),
    anchor('stats/create_transfer2',
'Transfer-out Rate'),
    )
    );
} elseif ($this->session->userdata('usertype') ==
"PhilCAT Coordinator") {
    $retval['menu'] = array(
    anchor('dashboard/index', 'Your
Account')=> array(
    anchor('account/update_account', 'Update
Account'),
    anchor('account/change_pass', 'Change
Password'),
    anchor('login/logout', 'Log out')),
    anchor("", 'TB Reports') => array(
        anchor('report/create_report0',
'Create NTP Quarterly Report on Treatment
Outcome'),
        anchor('report/create_report1',),
        anchor("", 'Provincial Health Units') =>
array(
            anchor('pho/view_phos', 'View
Provincial Units'),
            ),
        anchor("", 'TB Statistical Reports') => array(
            anchor('stats/create_smear3', 'Proportion
of Pulmonary Smear-Positive Cases'),
            anchor('stats/create_cure3', 'Cure Rate'),
            anchor('stats/create_complete3',
'Completion Rate'),
            anchor('stats/create_death3', 'Death Rate'),
            anchor('stats/create_fail3', 'Treatment
Failure Rate'),
            anchor('report/create_report0',
'Create NTP Quarterly Report on Treatment
Outcome'),
            anchor('report/create_report1',
'Create NTP Quarterly Report on all Cases'),
            ),
        anchor("", 'Provincial Health Units') =>
array(
            anchor('pho/view_phos', 'View
Provincial Units'),
            ),
        anchor("", 'TB Statistical Reports') => array(
            anchor('stats/create_smear3', 'Proportion
of Pulmonary Smear-Positive Cases'),
            anchor('stats/create_cure3', 'Cure Rate'),
            anchor('stats/create_complete3',
'Completion Rate'),
            anchor('stats/create_death3', 'Death Rate'),
            anchor('stats/create_fail3', 'Treatment
Failure Rate'),
            ));
} elseif ($this->session->userdata('usertype') ==
"DOH Coordinator") {
    $retval['menu'] = array(
    anchor('dashboard/index', 'Your Account')=>
array(
    anchor('account/update_account', 'Update
Account'),
    anchor('account/change_pass', 'Change
Password'),
    anchor('login/logout', 'Log out')
    anchor("", 'TB Reports') => array(
        anchor('report/create_report0',
'Create NTP Quarterly Report on Treatment
Outcome'),
        anchor('report/create_report1',),
        anchor("", 'Provincial Health Units') => array(
            anchor('pho/view_phos', 'View Provincial
Units'),
            ),
        anchor("", 'TB Statistical Reports') => array(
            anchor('stats/create_smear3', 'Proportion
of Pulmonary Smear-Positive Cases'),
            anchor('stats/create_cure3', 'Cure Rate'),
            anchor('stats/create_complete3',
'Completion Rate'),
            anchor('stats/create_death3', 'Death Rate'),
            anchor('stats/create_fail3', 'Treatment
Failure Rate'),
            ));
}

```

```

        anchor('stats/create_default3', 'Defaulter Rate'),
        anchor('stats/create_transfer3', 'Transfer-out
Rate'),
    ),
    anchor("", 'Provincial Health Units') => array(
        anchor('pho/add_pho', 'Add Provincial Unit'),
        anchor('pho/view_phos', 'View Provincial Units'),
    )
    );
} elseif ($this->session->userdata('usertype') ==
null) {
    $retval['menu'] = array(
        anchor('#', 'Home'),
        anchor('info/general_info', 'General
Information'),
        anchor("", 'TB Statistical Reports') => array(
            anchor('stats/create_smear', 'Proportion of
Pulmonary Smear-Positive Cases'),
            anchor('stats/create_cure', 'Cure Rate'),
            anchor('stats/create_complete',
'Completion Rate'),
            anchor('stats/create_death', 'Death Rate'),
            anchor('stats/create_fail', 'Treatment
Failure Rate'),
            anchor('stats/create_default', 'Defaulter
Rate'),
            anchor('stats/create_transfer', 'Transfer-out
Rate'),
        )
    );
}
return $this->load->view($this->common .
"menu", $retval, true);
}

function copyright() {
    $retval['copyright'] = "Copyright &copy; 2010
TRACKER System";
    return $this->load->view($this->common .
"copyright", $retval, true);
}
}
?>

```

Report Model

```

class Report_model extends Model {

private $PCUSER = 'PhilCAT Coordinator';
private $DOHUSER = 'DOH Coordinator';

function get_quarter($data) {
    $quarter['start'] = "";

```

```

    $quarter['end'] = "";
    if($data == '1') {
        $quarter['start'] = "-01-01";
        $quarter['end'] = "-03-31";
    } else if($data == '2') {
        $quarter['start'] = "-04-01";
        $quarter['end'] = "-06-30";
    } else if($data == '3') {
        $quarter['start'] = "-07-01";
        $quarter['end'] = "-09-30";
    } else if($data == '4') {
        $quarter['start'] = "-10-01";
        $quarter['end'] = "-12-31";
    }
    return $quarter;
}

function fetch_category($data) {
    if(isset($_POST['mysubmit'])){
        $health = $_POST['health_fac'];
    }

    $pho = $this->session->userdata('pho');
    $year = $data['year2'];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];
    $end = $quarter['end'];

    if($quarter != null && $year != null) {
        if($this->session->userdata('usertype') == $this-
>PCUSER || $this->session->userdata('usertype') ==
$this->DOHUSER) {
            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
HEALTH_FAC = '$health' AND P_CATEGORY = 'I' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
            $cat['cat1'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
HEALTH_FAC = '$health' AND P_CATEGORY = 'II'
AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
            $cat['cat2'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
HEALTH_FAC = '$health' AND P_CATEGORY = 'III'
AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
            $cat['cat3'] = count($user->result());
        } else {

```

```

        if(isset($_POST['mysubmit2'])) {
            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'I' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat1'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'II' AND (DATE_OF_REG
>= '$year$start' AND DATE_OF_REG <=
'$year$end')");
            $cat['cat2'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'III' AND (DATE_OF_REG
>= '$year$start' AND DATE_OF_REG <=
'$year$end')");
            $cat['cat3'] = count($user->result());
        } else {
            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND HEALTH_FAC = '$health' AND
P_CATEGORY = 'I' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat1'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND HEALTH_FAC = '$health' AND
P_CATEGORY = 'II' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat2'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND HEALTH_FAC = '$health' AND
P_CATEGORY = 'III' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat3'] = count($user->result());
        }
    }
    return $cat;
}

function fetch_all_category($data) {
    $pho = $this->session->userdata('pho');
    $year = $data['year2'];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];

```

```

    $end = $quarter['end'];

    if($quarter != null && $year != null) {
        if($this->session->userdata('usertype') == $this-
>PCUSER || $this->session->userdata('usertype') ==
$this->DOHUSER) {
            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
P_CATEGORY = 'I' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat1'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
P_CATEGORY = 'II' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat2'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE
P_CATEGORY = 'III' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat3'] = count($user->result());
        } else {
            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'I' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat1'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'II' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat['cat2'] = count($user->result());

            $user = $this->db->query("SELECT
P_CATEGORY FROM tb_register WHERE PHO_ID =
$pho AND P_CATEGORY = 'III' AND (DATE_OF_REG
>= '$year$start' AND DATE_OF_REG <=
'$year$end')");
            $cat['cat3'] = count($user->result());
        }
    }
    return $cat;
}

function fetch_patient_type($data) {
    $health="";
    if(isset($_POST['mysubmit'])) {
        $health = $_POST['health_fac'];
    }
}

```

```

}

$pho = $this->session->userdata('pho');
$year = $data['year2'];
$quarter = $this->get_quarter($data['quarter2']);
$start = $quarter['start'];
$end = $quarter['end'];
$types = array("New Pos", "New Neg", "Relapse",
"Trans-In", "Treatment Failure", "Default");
$out = array("Cured", "Treatment Completed",
"Died", "Failed", "Defaulted", "Transferred Out");
if($quarter != null && $year != null) {
    foreach ($types as $type) {
        foreach ($out as $out) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT *
FROM tb_register WHERE HEALTH_FAC = '$health'
AND OUTCOME = '$out' AND P_TYPE = '$type' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
            } else {
                if(isset($_POST['mysubmit2'])) {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
OUTCOME = '$out' AND P_TYPE = '$type' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
                } else {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND OUTCOME = '$out'
AND P_TYPE = '$type' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
                }
            }

            $cat[$type][$out] = count($user->result());
        }
    }
}

return $cat;
}

function fetch_patient_type2($data) {
    $health="";
    if(isset($_POST['mysubmit'])) {
        $health = $_POST['health_fac'];
    }
    $pho = $this->session->userdata('pho');
    $year = $data['year2'];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];

```

```

$end = $quarter['end'];
$types2 = array("New Pos", "Relapse", "Trans-
In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg");
$sexes = array("M", "F");

if($quarter != null && $year != null) {
    foreach ($types2 as $type) {
        foreach ($sexes as $sex) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT *
FROM tb_register WHERE HEALTH_FAC = '$health'
AND SEX = '$sex' AND P_TYPE = '$type' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
            } else {
                if(isset($_POST['mysubmit2'])) {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND SEX =
'$sex' AND P_TYPE = '$type' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
                } else {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = '$sex' AND
P_TYPE = '$type' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
                }
            }

            $cat[$type][$sex] = count($user->result());
        }
    }
}

if($this->session->userdata('usertype') == $this-
>PCUSER || $this->session->userdata('usertype') ==
$this->DOHUSER) {
    $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND SEX
= 'M' AND TB_CLASS = 'EP' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
    $cat['EP']['M'] = count($user->result());
    $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND SEX
= 'F' AND TB_CLASS = 'EP' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
    $cat['EP']['F'] = count($user->result());
} else {
    $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = 'M' AND

```

```

TB_CLASS = 'EP' AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end');
    $cat['EP']['M'] = count($user->result());
    $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = 'F' AND
TB_CLASS = 'EP' AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end')");
    $cat['EP']['F'] = count($user->result());
    }

    return $cat;
}
}

function fetch_patient_type3($data) {
    $health="";
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data["year2"];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $sources = array ("Public", "Private");
    $types2 = array("New Pos", "Relapse", "Trans-
In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg");
    $sexs = array("M", "F");

    if($quarter != null && $year != null) {
        foreach ($sources as $source) {
            foreach ($types2 as $type) {
                foreach ($sexs as $sex) {
                    if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                        $user = $this->db->query("SELECT *
FROM tb_register WHERE HEALTH_FAC = '$health'
AND SEX = '$sex' AND P_TYPE = '$type' AND
P_SOURCE = '$source' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
                    } else {
                        $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = '$sex' AND
P_TYPE = '$type' AND P_SOURCE = '$source' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
                    }
                    $cat[$source][$type][$sex] =
count($user->result());
                }
            }
        }
    }
}

```

```

    }
}
foreach ($sources as $source) {
    if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
        $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND SEX
= 'M' AND TB_CLASS = 'EP' AND P_SOURCE =
'$source' AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
        $cat[$source]['EP']['M'] = count($user-
>result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND SEX
= 'F' AND TB_CLASS = 'EP' AND P_SOURCE = '$source'
AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
        $cat[$source]['EP']['F'] = count($user-
>result());
    } else {
        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = 'M' AND
TB_CLASS = 'EP' AND P_SOURCE = '$source' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
        $cat[$source]['EP']['M'] = count($user-
>result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND SEX = 'F' AND
TB_CLASS = 'EP' AND P_SOURCE = '$source' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
        $cat[$source]['EP']['F'] = count($user-
>result());
    }
}
return $cat;
}
}

function fetch_all_patient_type3($data) {
    $pho = $this->session->userdata('pho');
    $year = $data["year2"];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $sources = array ("Public", "Private");
    $types2 = array("New Pos", "Relapse", "Trans-
In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg");
}

```

```

$sexs = array("M","F");

if($quarter != null && $year != null) {
    foreach ($sources as $source) {
        foreach ($types2 as $type) {
            foreach ($sexs as $sex) {
                if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE SEX = '$sex' AND P_TYPE =
'$type' AND P_SOURCE = '$source' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
                } else {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND SEX =
'$sex' AND P_TYPE = '$type' AND P_SOURCE =
'$source' AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
                }
                $cat[$source][$type][$sex] =
count($user->result());
            }
        }
    }
    foreach ($sources as $source) {
        if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
            $user = $this->db->query("SELECT * FROM
tb_register WHERE SEX = 'M' AND TB_CLASS = 'EP'
AND P_SOURCE = '$source' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat[$source]['EP']['M'] = count($user-
>result());
            $user = $this->db->query("SELECT * FROM
tb_register WHERE SEX = 'F' AND TB_CLASS = 'EP'
AND P_SOURCE = '$source' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end')");
            $cat[$source]['EP']['F'] = count($user-
>result());
        } else {
            $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND SEX = 'M'
AND TB_CLASS = 'EP' AND P_SOURCE = '$source'
AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
            $cat[$source]['EP']['M'] = count($user-
>result());
            $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND SEX = 'F'
AND TB_CLASS = 'EP' AND P_SOURCE = '$source'

```

```

AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end')");
            $cat[$source]['EP']['F'] = count($user-
>result());
        }
    }
    return $cat;
}
}

```

```

function fetch_patient_type4($data) {
    $health="";
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data['year2'];
    $quarter = $this->get_quarter($data['quarter2']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $supper = array("9","14","24","34","44","54","64",
"999");
    $lower = array("0","10","15","25","35","45","55",
"65");
    $sexs = array("M","F");

```

```

    if($quarter != null && $year != null) {
        for ($i = 0; $i < 8;$i++) {
            foreach ($sexs as $sex) {
                if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE P_TYPE = 'New Pos' AND
HEALTH_FAC = '$health' AND SEX = '$sex' AND (AGE
>= '$lower[$i]' AND AGE <= '$supper[$i]') AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
                } else {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE P_TYPE = 'New Pos' AND
PHO_ID = $pho AND HEALTH_FAC = '$health' AND
SEX = '$sex' AND (AGE >= '$lower[$i]' AND AGE <=
'$supper[$i]') AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end')");
                }
                $cat[$i][$sex] = count($user->result());
            }
        }
    }
    return $cat;
}
}

```

```

function fetch_all_patient_type4($data) {
    $pho = $this->session->userdata('pho');

```

```

$year = $data['year2'];
$quarter = $this->get_quarter($data['quarter2']);
$start = $quarter['start'];
$end = $quarter['end'];
$supper = array("9", "14", "24", "34", "44", "54", "64",
"999");
$lower = array("0", "10", "15", "25", "35", "45", "55",
"65");
$sexes = array("M", "F");

if($quarter != null && $year != null) {
    for ($i = 0; $i < 8; $i++) {
        foreach ($sexes as $sex) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT *
FROM tb_register WHERE P_TYPE = 'New Pos' AND
SEX = '$sex' AND (AGE >= '$lower[$i]' AND AGE <=
'$supper[$i]') AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end')");
            } else {
                $user = $this->db->query("SELECT *
FROM tb_register WHERE P_TYPE = 'New Pos' AND
PHO_ID = $pho AND SEX = '$sex' AND (AGE >=
'$lower[$i]' AND AGE <= '$supper[$i]') AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
            }

            $cat[$i][$sex] = count($user->result());
        }
    }
    return $cat;
}
}

```

```

function fetch_patient_type0($data) {
    $health="";
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $types = array("New Pos", "New Neg", "Relapse",
"Treatment Failure", "Default");
    $souts = array("Cured", "Treatment Completed",
"Died", "Failed", "Defaulted", "Transferred Out");

    if($quarter != null && $year != null) {
        foreach ($types as $type) {

```

```

        foreach ($souts as $out) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                if(isset($_POST['mysubmit_pc'])) {
                    $user = $this->db->query("SELECT *
FROM tb_register LEFT JOIN pho ON
tb_register.PHO_ID = pho.PHO_ID WHERE
PHO_NAME = '$health' AND OUTCOME = '$out' AND
P_TYPE = '$type' AND TB_CLASS = 'P' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
                } else {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE HEALTH_FAC = '$health'
AND OUTCOME = '$out' AND P_TYPE = '$type' AND
TB_CLASS = 'P' AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end')");
                }
            } else {
                $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND OUTCOME = '$out'
AND P_TYPE = '$type' AND TB_CLASS = 'P' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end')");
            }
            $cat[$type][$out] = count($user->result());
        }
    }
    return $cat;
}
}

```

```

function fetch_all_patient_type0($data) {
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $types = array("New Pos", "New Neg", "Relapse",
"Treatment Failure", "Default");
    $souts = array("Cured", "Treatment Completed",
"Died", "Failed", "Defaulted", "Transferred Out");

    if($quarter != null && $year != null) {
        foreach ($types as $type) {
            foreach ($souts as $out) {
                if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                    $user = $this->db->query("SELECT *
FROM tb_register WHERE OUTCOME = '$out' AND

```



```

P_TYPE = '$type' AND TB_CLASS = 'P' AND
(Date_of_Reg >= '$year$start' AND Date_of_Reg
<= '$year$end');
    } else {
        $user = $this->db->query("SELECT *
FROM tb_register WHERE PHO_ID = $pho AND
OUTCOME = '$out' AND P_TYPE = '$type' AND
TB_CLASS = 'P' AND (Date_of_Reg >= '$year$start'
AND Date_of_Reg <= '$year$end')");
    }
    $cat[$type][$out] = count($user->result());
}
}
return $cat;
}
}
function fetch_new($data) {
    $year = "";
    $quarter = "";
    $health = "";
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $smears = array("New Pos", "New Neg");

    if($quarter != null && $year != null) {
        foreach ($smears as $smear) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND
TB_CLASS = 'P' AND P_TYPE = '$smear' AND
(Date_of_Reg >= '$year$start' AND Date_of_Reg
<= '$year$end')");
            } else {
                $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND TB_CLASS = 'P' AND
P_TYPE = '$smear' AND (Date_of_Reg >=
'$year$start' AND Date_of_Reg <= '$year$end')");
            }

            $cat['new_count'][$smear] = count($user-
>result());
        }
    }
    return $cat;
}
}

```

```

function fetch_all_new($data) {
    $year = "";
    $quarter = "";
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $smears = array("New Pos", "New Neg");
    if($quarter != null && $year != null) {
        foreach ($smears as $smear) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT * FROM
tb_register WHERE TB_CLASS = 'P' AND P_TYPE =
'$smear' AND (Date_of_Reg >= '$year$start' AND
Date_of_Reg <= '$year$end')");
            } else {
                $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND TB_CLASS =
'P' AND P_TYPE = '$smear' AND (Date_of_Reg >=
'$year$start' AND Date_of_Reg <= '$year$end')");
            }
            $cat['new_count'][$smear] = count($user-
>result());
        }
    }
    return $cat;
}
}

function fetch_retreatment($data) {
    $year = "";
    $quarter = "";
    $health = "";
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $smears = array("Relapse", "Treatment Failure",
"Default");
    if($quarter != null && $year != null) {
        foreach ($smears as $smear) {
            if($this->session->userdata('usertype') ==
$this->PCUSER || $this->session-
>userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT * FROM
tb_register WHERE HEALTH_FAC = '$health' AND
TB_CLASS = 'P' AND P_TYPE = '$smear' AND

```

```

    (DATE_OF_REG >= '$year$start' AND DATE_OF_REG
    <= '$year$end');
    } else {
        $user = $this->db->query("SELECT
    TB_CASENO FROM tb_register WHERE PHO_ID =
    $pho AND HEALTH_FAC = '$health' AND TB_CLASS =
    'P' AND P_TYPE = '$smear' AND (DATE_OF_REG >=
    '$year$start' AND DATE_OF_REG <= '$year$end')");
    }

    $cat['ret_count'][$smear] = count($user-
    >result());
    }
    return $cat;
    }
}

function fetch_all_retreatment($data) {
    $year = "";
    $quarter = "";
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];
    $smears = array("Relapse", "Treatment Failure",
    "Default");
    if($quarter != null && $year != null) {
        foreach ($smears as $smear) {
            if($this->session->userdata('usertype') ==
            $this->PCUSER || $this->session-
            >userdata('usertype') == $this->DOHUSER) {
                $user = $this->db->query("SELECT * FROM
            tb_register WHERE TB_CLASS = 'P' AND P_TYPE =
            '$smear' AND (DATE_OF_REG >= '$year$start' AND
            DATE_OF_REG <= '$year$end')");
            } else {
                $user = $this->db->query("SELECT * FROM
            tb_register WHERE PHO_ID = $pho AND TB_CLASS =
            'P' AND P_TYPE = '$smear' AND (DATE_OF_REG >=
            '$year$start' AND DATE_OF_REG <= '$year$end')");
            }

            $cat['ret_count'][$smear] = count($user-
            >result());
        }
        return $cat;
    }
}

function fetch_EP($data) {
    $health = $_POST['health_fac'];
    $pho = $this->session->userdata('pho');
    $year = $data['year'];

```

```

    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];

    if($this->session->userdata('usertype') == $this-
    >PCUSER || $this->session->userdata('usertype') ==
    $this->DOHUSER) {
        $user = $this->db->query("SELECT * FROM
    tb_register WHERE HEALTH_FAC = '$health' AND
    (P_TYPE = 'Trans-In' OR P_TYPE = 'Other Pos' OR
    P_TYPE = 'Other Neg' OR TB_CLASS = 'EP') AND
    (DATE_OF_REG >= '$year$start' AND DATE_OF_REG
    <= '$year$end')");
        $cat['excluded'] = count($user->result());

        $user = $this->db->query("SELECT * FROM
    tb_register WHERE HEALTH_FAC = '$health' AND
    P_TYPE = 'Trans-In' AND (DATE_OF_REG >=
    '$year$start' AND DATE_OF_REG <= '$year$end')");
        $cat['ex_trans'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
    tb_register WHERE HEALTH_FAC = '$health' AND
    TB_CLASS = 'EP' AND (DATE_OF_REG >= '$year$start'
    AND DATE_OF_REG <= '$year$end')");
        $cat['ex_ep'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
    tb_register WHERE HEALTH_FAC = '$health' AND
    P_TYPE = 'Other Pos' OR P_TYPE = 'Other Neg' AND
    (DATE_OF_REG >= '$year$start' AND DATE_OF_REG
    <= '$year$end')");
        $cat['ex_other'] = count($user->result());
    } else {
        $user = $this->db->query("SELECT * FROM
    tb_register WHERE PHO_ID = $pho AND
    HEALTH_FAC = '$health' AND (P_TYPE = 'Trans-In' OR
    P_TYPE = 'Other Pos' OR P_TYPE = 'Other Neg' OR
    TB_CLASS = 'EP') AND (DATE_OF_REG >= '$year$start'
    AND DATE_OF_REG <= '$year$end')");
        $cat['excluded'] = count($user->result());

        $user = $this->db->query("SELECT * FROM
    tb_register WHERE PHO_ID = $pho AND
    HEALTH_FAC = '$health' AND P_TYPE = 'Trans-In'
    AND (DATE_OF_REG >= '$year$start' AND
    DATE_OF_REG <= '$year$end')");
        $cat['ex_trans'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
    tb_register WHERE PHO_ID = $pho AND
    HEALTH_FAC = '$health' AND TB_CLASS = 'EP' AND
    (DATE_OF_REG >= '$year$start' AND DATE_OF_REG
    <= '$year$end')");
        $cat['ex_ep'] = count($user->result());
    }
}

```

```

    $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND
HEALTH_FAC = '$health' AND P_TYPE = 'Other Pos'
OR P_TYPE = 'Other Neg' AND (DATE_OF_REG >=
'$year$start' AND DATE_OF_REG <= '$year$end'");
    $cat['ex_other'] = count($user->result());
}
return $cat;
}
function fetch_all_EP($data) {
    $pho = $this->session->userdata('pho');
    $year = $data['year'];
    $quarter = $this->get_quarter($data['quarter']);
    $start = $quarter['start'];
    $end = $quarter['end'];

    if($this->session->userdata('usertype') == $this-
>PCUSER || $this->session->userdata('usertype') ==
$this->DOHUSER) {
        $user = $this->db->query("SELECT * FROM
tb_register WHERE P_TYPE = 'Trans-In' OR P_TYPE =
'Other Pos' OR P_TYPE = 'Other Neg' OR TB_CLASS
='EP' AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end'");
        $cat['excluded'] = count($user->result());

        $user = $this->db->query("SELECT * FROM
tb_register WHERE P_TYPE = 'Trans-In' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end'");
        $cat['ex_trans'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE TB_CLASS = 'EP' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end'");
        $cat['ex_ep'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE P_TYPE = 'Other Pos' OR P_TYPE
= 'Other Neg' AND (DATE_OF_REG >= '$year$start'
AND DATE_OF_REG <= '$year$end'");
        $cat['ex_other'] = count($user->result());
    } else {
        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND P_TYPE =
'Trans-In' OR P_TYPE = 'Other Pos' OR P_TYPE =
'Other Neg' OR TB_CLASS = 'EP' AND (DATE_OF_REG
>= '$year$start' AND DATE_OF_REG <=
'$year$end'");
        $cat['excluded'] = count($user->result());

        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND P_TYPE =

```

```

'Trans-In' AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end'");
        $cat['ex_trans'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND TB_CLASS
='EP' AND (DATE_OF_REG >= '$year$start' AND
DATE_OF_REG <= '$year$end'");
        $cat['ex_ep'] = count($user->result());
        $user = $this->db->query("SELECT * FROM
tb_register WHERE PHO_ID = $pho AND P_TYPE =
'Other Pos' OR P_TYPE = 'Other Neg' AND
(DATE_OF_REG >= '$year$start' AND DATE_OF_REG
<= '$year$end'");
        $cat['ex_other'] = count($user->result());
    }
    return $cat;
}

```

Views

active_account.php

```

<h3>Manage Active User Accounts</h3>
<?php echo form_open('account/activity',
"name=active_users", $_POST); ?>
<table width="100%">
    <tr class="header2">
        <td align="left"
width="100px"><b>Username</b></td>
        <td align="left"
width="200px"><b>Usertype</b></td>
        <td align="left"
width="250px"><b>Designation</b></td>
        <?php if($this->session->userdata('usertype') ==
"System Administrator") { ?>
            <td align="left"
width="150px"><b>Region</b></td>
            <?php } ?>
        <td align="center"
width="100px"><b>Status</b></td>
    </tr>
<?php
$num = count($data)-3;
if($num != 0) {
    for ($i = 0; $i < $num; $i++) {
        echo ("<tr>
        <td align=left>" . $data[$i]->USERNAME .
"</td>
        <td align=left>" . $data[$i]->USERTYPE .
"</td>
        <td align=left>" . $data[$i]->DESIGNATION .
"</td>");
    }
}

```



```

        <td width="35%">
            <input name="username" type="text"
value="<?php echo $this->validation->username;?>"
/><br><i>(min 6 characters)</i> <br/>
            </td>
            <td class="error" width="35%"><?php echo
$this->validation->username_error; ?></td>
        </tr>
        <tr>
            <td align="right">Password:</td>
            <td>
                <input name="password" id="password"
type="password" value="<?php echo $this-
>validation->password;?>"><br><i>(min 6
characters)</i>
            </td>
            <td class="error"> <?php echo $this->validation-
>password_error; ?></td>
        </tr>
        <tr>
            <td align="right">Re-type Password:</td>
            <td>
                <input name="password2" id="password2"
type="password" value="<?php echo $this-
>validation->password2;?>">
            </td>
        </tr>
        <tr>
            <td align="right">Designation:</td>
            <td>
                <select name="user_type" id="user_type">
                    <option value="">-Select one-</option>
                    <option value="Nurse" <?php if($this-
>validation->user_type == "Nurse") echo
"selected";?>>Nurse</option>
                    <option value="Encoder" <?php if($this-
>validation->user_type == "Encoder") echo
"selected";?>>Encoder</option>
                    <?php if($this->session-
>userdata('usertype') == "System Administrator") {
                        echo "<option value='PhilCAT
Coordinator'";
                        if($this->validation->user_type ==
'PhilCAT Coordinator') echo "'selected'";
                        echo ">PhilCAT Monitoring</option>";
                        echo "<option value='Provincial NTP
Coordinator'";
                        if($this->validation->user_type ==
'Provincial NTP Coordinator') echo "'selected'";
                        echo ">Provincial NTP
Coordinator</option>";
                        echo "<option value='Regional
Coordinator'";

```

```

            if($this->validation->user_type ==
'Regional Coordinator') echo "'selected'";
            echo ">Regional Coordinator</option>";
            echo "<option value='DOH Coordinator'";
            if($this->validation->user_type == 'DOH
Coordinator') echo "'selected'";
            echo ">DOH Coordinator</option>";
            echo "<option value='System
Administrator'";
            if($this->validation->user_type ==
'System Administrator') echo "'selected'";
            echo ">System Administrator</option>";
        } ?>
    </select>
</td>
    <td class="error"><?php echo $this->validation-
>user_type_error; ?></td>
</tr>
<?php if($this->session->userdata('usertype') ==
"System Administrator") {
    echo "<tr><td align='right'> Health Facility
:</td><td>
        <select name='region' id='region'>
            <option value="">-Select one-</option>
            <option value='DOH'>DOH Head
Office</option>";
            $phos = $_POST["phos"];
            $num = count($phos);
            for ($i = 0; $i < $num; $i++) {
                echo ("<option value=''."$phos[$i]-
>PHO_ID.'">."$phos[$i]->PHO_NAME."</option>
                ");
            }
            echo "</select></td><td class='error'>";
            echo $this->validation->region_error;
            echo "</td></tr>";
        } ?>
    <tr>
        <td align="right">First name: </td>
        <td>
            <input name="first_name" type="text"
value="<?php echo $this->validation-
>first_name;?>">
            </td>
        <td class="error"><?php echo $this->validation-
>first_name_error; ?></td>
    </tr>
    <tr>
        <td align="right">Middle name:</td>
        <td><input name="middle_name" type="text"
/></td>
    </tr>

```

```

<tr>
  <td align="right">Last name:</td>
  <td>
    <input name="last_name" type="text"
value="<?php echo $this->validation-
>last_name;?>"/>
  </td>
  <td class="error"><?php echo $this->validation-
>last_name_error; ?></td>
</tr>

```

```

<tr>
  <td align="right">Gender:</td>
  <td>
    <input type="radio" name="gender"
value="male" />Male
    <input type="radio" name="gender"
value="female" />Female
  </td>
</tr>
<tr>
  <td colspan="3" class="header2">&nbsp;</td>
</tr>
</table>
<center>
  <?php echo form_submit('mysubmit', 'Create
Account'); ?>
</center>
<?php echo form_close(); ?>

```

view_users_province.php

```

<h3>User Accounts List</h3>
<table width="100%">
  <tr> <td colspan="4" class="header2">User
List</td></tr>
  <tr class="header2">
    <td width="15%"><b>Username</b></td>
    <td width="30%"><b>Usertype /
Designation</b></td>
    <td width="25%"><b>Name</b></td>
  </tr>
  <?php
$num = count($data)-3;

for ($i = 0; $i < $num; $i++) {
  echo ("<tr>
    <td align="left" width="15%">" . $data[$i]-
>USERNAME . "</td>
    <td align="left" width="30%">" . $data[$i]-
>USERTYPE . "</td>

```

```

    <td align="left" width="25%">" . $data[$i]->NAME
. " " . $data[$i]->LAST_NAME . "</td>
    </tr>");
  }
  ?>
  <tr> <td colspan="4"
class="header2">&nbsp;</td></tr>
</table>
<br />
<div><?php echo $this->pagination->create_links();
?></div>

```

complete_rate.php

```
<?php
$ages = array("0 - 14", "15 - 24", "25 - 34", "35 -
44", "45 - 54", "55 - 64", "65 above");
?>
<center>
<?php include_once 'view_graph.php'; ?>
<table width="600" border="0">
  <tr class="header2" height="30px">
    <td colspan="4">Treatment Completion
Rate</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td colspan="3"><b><?php echo
$_POST['from'] . " to " . $_POST['to']; ?></b></td>
  </tr>
  <tr class="header2">
    <td width="40%">Region</td>
    <td width="20%">Number of cases who
completed treatment</td>
    <td width="20%">Total number of cases
registered</td>
    <td width="20%">Treatment Completion
Rate</td>
  </tr>
  <?php
  foreach ($regions as $region) {
    echo "<tr align='center'>
      <td class='header2'>".$region."</td>
      <td>".$_POST['stat1']['complete1'][$region]."</td>
      <td>".$_POST['stat2']['total'][$region]."</td>";
      if($_POST['stat2']['total'][$region] == 0) {
        echo "<td>0 %</td>";
      } else {
        echo
"<td>".round(($_POST['stat1']['complete1'][$region]/
$_POST['stat2']['total'][$region])*100) . "%</td>";
      }
      echo "</tr>";
    }
  ?>
  <tr>
    <td colspan="4"
class="header2">&nbsp;</td>
  </tr>
```

```
</table>
<table width="600" border="0" >
  <tr>
    <td>&nbsp;</td>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td align="center">MALE</td>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr class="header2">
    <td width="40%">Age</td>
    <td width="20%">Number of cases who
completed treatment</td>
    <td width="20%">Total number of cases
registered</td>
    <td width="20%">Treatment Completion
Rate</td>
  </tr>
  <?php
  for($i = 0; $i < 7; $i++) {
    echo "<tr align='center'>
      <td class='header2'>".$ages[$i]."</td>
      <td>".$_POST['stat3'][$i]['M']."</td>
      <td>".$_POST['stat4'][$i]['M']."</td>";
      if($_POST['stat4'][$i]['M'] == 0) {
        echo "<td>0 %</td>";
      } else {
        echo
"<td>".round(($_POST['stat3'][$i]['M']/$_POST['stat4
'][$i]['M'])*100) . "%</td>";
      }
      echo "</tr>";
    }
  ?>
</table>
<table width="600" border="0" >
  <tr>
    <td>&nbsp;</td>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr>
    <td align="center">FEMALE</td>
    <td colspan="3">&nbsp;</td>
  </tr>
  <tr class="header2">
    <td width="40%">Age</td>
    <td width="20%">Number of cases who
completed treatment</td>
    <td width="20%">Total number of cases
registered</td>
```



```

        echo
"<td>".round(($_POST['stat3'][$i]['M']/$_POST['stat4
'][$i]['M'])*100) ." %</td>";
    }
    echo "</tr>";
}
?>

</table>
<table width="600" border="0" >
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td align="center">FEMALE</td>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr class="header2">
<td width="40%">Age</td>
<td width="20%">Number of cases who
completed treatment</td>
<td width="20%">Total number of cases
registered</td>
<td width="20%">Treatment Completion
Rate</td>
</tr>
<?php
for($i = 0; $i < 7; $i++) {
    echo "<tr align='center'>
        <td class='header2'>".$ages[$i]."</td>
        <td>".$_POST['stat3'][$i]['F']."</td>
        <td>".$_POST['stat4'][$i]['F']."</td>";
        if($_POST['stat4'][$i]['F'] == 0) {
            echo "<td>0 %</td>";
        } else {
            echo
"<td>".round(($_POST['stat3'][$i]['F']/$_POST['stat4
'][$i]['F'])*100) ." %</td>";
        }
        echo "</tr>";
    }
?>
</table>

```

</center>

transfer_rate.php

```

<?php
$ages = array("0 - 14","15 - 24","25 - 34","35 -
44","45 - 54","55 - 64","65 above");
?>

```

```

<center>
<?php include_once 'view_graph.php'; ?>
<table width="600" border="0">
<tr class="header2" height="30px">
<td colspan="4">Transfer-out Rate</td>
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td colspan="3"><b><?php echo
$_POST['from'] . " to " . $_POST['to']; ?></b></td>
</tr>
<tr class="header2">
<td width="40%">Region</td>
<td width="20%">Number of cases who
transferred to another facility</td>
<td width="20%">Total number of cases
registered</td>
<td width="20%">Transfer Out Rate</td>
</tr>
<?php
foreach ($regions as $region) {
    echo "<tr align='center'>
        <td class='header2'>".$region."</td>
<td>".$_POST['stat']['transfer1'][$region]."</td>
<td>".$_POST['stat2']['total'][$region]."</td>";
        if($_POST['stat2']['total'][$region] == 0) {
            echo "<td>0 %</td>";
        } else {
            echo
"<td>".round(($_POST['stat']['transfer1'][$region]/$_
POST['stat2']['total'][$region])*100) ." %</td>";
        }
        echo "</tr>";
    }
?>
<tr>
<td colspan="4"
class="header2">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
<table width="600" border="0" >
<tr>
<td>&nbsp;&nbsp;&nbsp;</td>
<td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td align="center">MALE</td>

```

```

        <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr class="header2">
    <td width="40%">Age</td>
    <td width="20%">Number of cases who
transferred to another facility</td>
    <td width="20%">Total number of cases
registered</td>
    <td width="20%">Transfer Out Rate</td>
</tr>
<?php
for($i = 0; $i < 7; $i++) {
    echo "<tr align='center'>
        <td class='header2'>". $ages[$i]. "</td>
        <td>". $_POST['stat3'][$i]['M']. "</td>
        <td>". $_POST['stat4'][$i]['M']. "</td>";
        if($_POST['stat4'][$i]['M'] == 0) {
            echo "<td>0 % </td>";
        } else {
            echo
"<td>". round(($_POST['stat3'][$i]['M']/$_POST['stat4
'][$i]['M'])*100) . " % </td>";
        }
        echo "</tr>";
    }
}
?>

```

```

</table>
<table width="600" border="0" >
<tr>
    <td>&nbsp;&nbsp;&nbsp;</td>
    <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
    <td align="center">FEMALE</td>
    <td colspan="3">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr class="header2">
    <td width="40%">Age</td>
    <td width="20%">Number of cases who
transferred to another facility</td>
    <td width="20%">Total number of cases
registered</td>
    <td width="20%">Transfer Out Rate</td>
</tr>
<?php
for($i = 0; $i < 7; $i++) {
    echo "<tr align='center'>
        <td class='header2'>". $ages[$i]. "</td>
        <td>". $_POST['stat3'][$i]['F']. "</td>
        <td>". $_POST['stat4'][$i]['F']. "</td>";
        if($_POST['stat4'][$i]['F'] == 0) {
            echo "<td>0 % </td>";

```

```

    } else {
        echo
"<td>". round(($_POST['stat3'][$i]['F']/$_POST['stat4
'][$i]['F'])*100) . " % </td>";
    }
    echo "</tr>";
}
?>
</table>
</center>

```

tb_register.php

```

<?php
//header("Content-Type: application/vnd.ms-
word");
//header("Expires: 0");
//header("Cache-Control: must-revalidate, post-
check=0, pre-check=0");
//header("Content-disposition: attachment;
filename=\"mydocument_name.doc\"");
?>

<h3> TB Register for PHO : <?php echo $this-
>session->userdata('pho_name'); ?></h3>

<table width="1200" border="1" cellspacing="0">
    <tr class="header2">
        <td width="75" rowspan="2">Date of
Registration </td>
        <td width="60" rowspan="2">TB Case No. </td>
        <td width="61" rowspan="2">Name</td>
        <td width="25" rowspan="2">Age</td>
        <td width="23" rowspan="2">Sex</td>
        <td width="92" rowspan="2">Address</td>
        <td width="103" rowspan="2">Health
Facility</td>
        <td width="150" colspan="2"
height="20px">Source of Patient </td>
        <td width="104" rowspan="2">Name of
Referring Physician </td>
        <td width="53" rowspan="2">Class of TB (P/EP)
</td>
        <td colspan="6">Type of Patient </td>
        <td width="61" rowspan="2">Category</td>
        <td width="75" rowspan="2">DATE STARTED TX
</td>
        <td colspan="7">SPUTUM EXAMINATION
RESULTS </td>
        <td colspan="6">TREATMENT OUTCOME </td>

```



```

        if($data[$i]->OUTCOME=='Failed' ) echo "x
</td>";
        echo "<td align='center' rowspan='2'>";
        if($data[$i]->OUTCOME=='Defaulted') echo "x
</td>";
        echo "<td align='center' rowspan='2'>";
        if($data[$i]->OUTCOME=='Transferred Out')
echo "x </td>";
        echo "<td align='center' rowspan='2'> .
$data[$i]->PARTNER . "</td>";
        echo "<td align='center' rowspan='2'> .
$data[$i]->REVIEW . "</td>";
        echo "</tr><tr>";
        $k = 0;
        for ($j = 0; $j < 7; $j++) {
            if(isset($sputum[$k]->MONTH) &&
            $sputum[$k]->MONTH == $j) {
                echo "<td class=''.Seven.'">".
            $sputum[$k]->RESULT. "</td>";
                $k++;
            } else echo "<td
            class=''.Seven.'">&nbsp;&nbsp;&nbsp;</td>";
        }

        echo "</tr>";
    }
    ?>

```

```
</table>
```

```

<div><?php echo $this->pagination->create_links();
?></div>
<?php if ($num == 0) echo " <h3> No TB Cases in the
registry for this PHO yet.</h3>"; ?>

```

ntp_id.php

```

<h3 class="title">Add TB Case</h3>
<?php echo $this->session->flashdata('message'); ?>
<?php echo form_open('register/add_sputum',
"name='tb_form'", $this->arr) . "\n"; ?>
<table width="100%" cellspacing="5px">
    <tr>
        <td colspan="3" class="header2">Patient TB
Information </td>
    </tr>
    <tr>
        <td width="25%" align="right">TB Case Number
: </td>
        <td width="40%"><input name="case_no"
id="case_no" type="text" maxlength="8" size="40"
value="<?php echo $this->validation->case_no;
?>" /></td>

```

```

        <td width="35%" class="error"><?php echo
$this->validation->case_no_error; ?></td>
    </tr>
    <tr>
        <td align="right"> Date of Registration : </td>
        <td><input type="text" name="reg_date"
id="_reg_date" size="10" maxlength="10" readonly
value="<?php echo $this->validation->reg_date;
?>" />
        <i>(yyyy-mm-dd)</i></td>
        <td class="error"><?php echo $this->validation-
>reg_date_error; ?></td>
    </tr>
    <tr>
        <td align="right">First Name : </td>
        <td><input name="patient_name"
id="patient_name" type="text" size="40"
value="<?php echo $this->validation-
>patient_name; ?>" /></td>
        <td class="error"><?php echo $this->validation-
>patient_name_error; ?></td>
    </tr>
    <tr>
        <td align="right">MI : </td>
        <td><input name="patient_mi" id="patient_mi"
type="text" maxlength="3" size="10"
size="4" /></td>
        <td class="error"><?php echo $this->validation-
>patient_mi_error; ?></td>
    </tr>
    <tr>
        <td align="right">Last Name : </td>
        <td><input name="patient_lname"
id="patient_lname" type="text" size="40"
value="<?php echo $this->validation-
>patient_lname; ?>" /></td>
        <td class="error"><?php echo $this->validation-
>patient_lname_error; ?></td>
    </tr>
    <tr>
        <td align="right">City : </td>
        <td><textarea name="patient_add" cols="42"
rows="2"><?php echo $this->validation-
>patient_add; ?></textarea></td>
        <td class="error"><?php echo $this->validation-
>patient_add_error; ?></td>
    </tr>
    <tr>
        <td align="right">Age : </td>
        <td><input type="text" name="age" id="age"
onchange="numChecker('age')" maxlength="2"
size="10" value="<?php echo $this->validation->age;
?>" /></td>

```

```

        <td class="error" id="age_stats"><?php echo
$this->validation->age_error; ?></td>
    </tr>
    <tr>
        <td align="right">Sex :</td>
        <td>
            <input type="radio" name="patient_sex"
value="M" />Male
            <input type="radio" name="patient_sex"
value="F" />Female
        </td>
        <td class="error"><?php echo $this->validation-
>patient_sex_error; ?></td>
    </tr>
</table>
<table width="100%" cellpadding="5px">
    <tr>
        <td colspan="3" class="header2">DOTS
Information </td>
    </tr>
    <tr>
        <td width="25%" align="right">DOTS Facility :
</td>
        <td width="40%">
            <select name="health_fac">
                <option value="">-Select one-</option>
                <?php
                    $rhus = $_POST["rhus"];
                    $num = count($rhus);
                    for ($i = 0; $i < $num; $i++) {
                        echo ("<option value="" . $rhus[$i]-
>RHU_NAME . ""> . $rhus[$i]->RHU_NAME .
" </option>");
                    }
                <?>
            </select></td>
        <td class="error" width="35%"><?php echo
$this->validation->health_fac_error; ?></td>
    </tr>
    <tr>
        <td align="right">Source of Patient : </td>
        <td>
            <input type="radio" name="patient_source"
value="Public" />Public
            <input type="radio" name="patient_source"
value="Private" />Private
        </td>
        <td class="error"><?php echo $this->validation-
>patient_source_error; ?></td>
    </tr>
    <tr>
        <td align="right">Name of Physician : </td>

```

```

        <td><input name="doc_name" id="doc_name"
type="text" size="40" value="<?php echo $this-
>validation->doc_name; ?>" /></td>
        <td class="error"><?php echo $this->validation-
>doc_name_error; ?></td>
    </tr>
    <tr>
        <td align="right">Disease Classification : </td>
        <td>
            <input type="radio" name="tb_class"
value="P" />Pulmonary
            <input type="radio" name="tb_class"
value="EP" />Extra Pulmonary
        </td>
        <td class="error"><?php echo $this->validation-
>tb_class_error; ?></td>
    </tr>
    <tr>
        <td align="right">TB Category : </td>
        <td>
            <input type="radio" name="category"
value="I" />I. 6-SCC (2HRZE/4HR)<br>
            <input type="radio" name="category"
value="II" />II. 8-SCC (2HRZE/1HRZE/5HRE)<br>
            <input type="radio" name="category"
value="III" />III. 8-SCC (2HRZ/4HR)<br>
        </td>
        <td class="error"><?php echo $this->validation-
>category_error; ?></td>
    </tr>
    <tr>
        <td align="right">Type of Patient : </td>
        <td>
            <select name="patient_type">
                <option value="">-Select one-</option>
                <option value="New Pos" <?php if
((isset($_POST["patient_type"])) &&
$_POST["patient_type"] == "New Pos") || ($this-
>validation->patient_type == "New Pos"))
                    echo "selected"; ?>>New Smear (+)
            </option>
                <option value="New Neg" <?php if
((isset($_POST["patient_type"])) &&
$_POST["patient_type"] == "New Neg") || ($this-
>validation->patient_type == "New Neg"))
                    echo "selected"; ?>>New Smear (-)
            </option>
                <option value="Relapse" <?php if
((isset($_POST["patient_type"])) &&
$_POST["patient_type"] == "Relapse") || ($this-
>validation->patient_type == "Relapse"))
                    echo "selected"; ?>>Relapse
            </option>

```

```

        <option value="Trans-In" <?php if
((isset($_POST["patient_type"]) &&
$_POST["patient_type"] == "Trans-In") || ($this-
>validation->patient_type == "Trans-In"))
        echo "selected"; ?>>Trans-In
        </option>
        <option value="Treatment Failure" <?php if
((isset($_POST["patient_type"]) &&
$_POST["patient_type"] == "Treatment Failure") ||
($this->validation->patient_type == "Treatment
Failure"))
        echo "selected"; ?>>Treatment
Failure
        </option>
        <option value="Default" <?php if
((isset($_POST["patient_type"]) &&
$_POST["patient_type"] == "Default") || ($this-
>validation->patient_type == "Default"))
        echo "selected"; ?>>Return after
Default
        </option>
        <option value="Other Pos" <?php if
((isset($_POST["patient_type"]) &&
$_POST["patient_type"] == "Other Pos") || ($this-
>validation->patient_type == "Other Pos"))
        echo "selected"; ?>>Others (+)
        </option>
        <option value="Other Neg" <?php if
((isset($_POST["patient_type"]) &&
$_POST["patient_type"] == "Other Neg") || ($this-
>validation->patient_type == "Other Neg"))
        echo "selected"; ?>>Others (-)
        </option>
    </select>
</td>
    <td class="error"><?php echo $this->validation-
>patient_type_error; ?></td>
</tr>
<tr>
    <td colspan="3" class="header2">&nbsp;</td>
</tr>
<tr>
    <td></td>
    <td colspan="2"><?php echo
form_submit('mysubmit', ' Next Section >> ');
?></td>
</tr>
</table>
<?php echo form_close(); ?>

```

search_patient.php

```

<h3 class="title">Search TB Case</h3>
<?php echo $this->session->flashdata('message'); ?>
<?php echo form_open('register/search',
"name='tb_form') . "\n"; ?>
<table width="100%" cellpadding="5px">
    <tr>
        <td colspan="3" class="header2">Search TB
Case Record</td>
    </tr>
    <tr>
        <td width="25%" align="right">TB Case Number
: </td>
        <td width="35%"><input name="tb_search"
type="text" id="tb_search" maxlength="8"/></td>
    </tr>
    <tr>
        <td width="25%" align="right">First Name :
</td>
        <td width="35%"><input name="search_name"
type="text" id="search_name"
maxlength="8"/></td>
    </tr>
    <tr>
        <td width="25%" align="right">Last Name :
</td>
        <td width="35%"><input name="search_lname"
type="text" id="search_lname"
maxlength="8"/></td>
    </tr>
    <tr>
        <td width="25%" align="right">Gender : </td>
        <td width="35%">
            <input type="radio" name="gender"
value="M" id="gender"/>Male
            <input type="radio" name="gender"
value="F" id="gender"/>Female
            <input type="radio" name="gender"
value="X" id="gender" checked/>Unknown
        </td>
    </tr>
</table>
<center>
    <?php echo form_submit('mysubmit', 'Search'); ?>
</center>
<?php echo form_close(); ?>

```



```

?></td>
</tr>

<tr>
<td>Remarks : <?php echo $data[0]->REMARKS;
?></td>
<td>&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td colspan="2" >&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
<table width="100%" >
<tr>
<td colspan="5" class="header2">Sputum
Examination Results / Weight</td>
</tr>
<tr align="center">
<td width="10%">Month</td>
<td width="27%">Date Examined</td>
<td width="21%">Result</td>
<td width="15%">Weight</td>
</tr>
<?php
$num = count($data) - 3;

for ($i = 0; $i < $num; $i++) {
    $this->load->model('register_model');
    $sputum = $this->register_model-
>fetch_sputum_exam($data[$i]->TB_CASENO);
    $k = 0;
    for ($j = 0; $j < 7; $j++) {

        $monthName = $j;
        switch ($j) {
            case 1: $monthName = "2nd";
                break;
            case 2: $monthName = "3rd";
                break;
            case 3: $monthName = "4th";
                break;
            case 4: $monthName = "5th";
                break;
            case 5: $monthName = "6th";
                break;
            case 6: $monthName = "> 7th";
                break;
            default: $monthName = $j;
                break;
        }

        if (isset($sputum[$k]->MONTH) &&
        $sputum[$k]->MONTH == $j) {

```

```

        echo "<tr align='center'><td height='21'>" .
        $monthName . "</td>";
        echo "<td>" . $sputum[$k]->DATE_EXAM .
        "</td>";
        echo "<td>" . $sputum[$k]->RESULT .
        "</td>";
        echo "<td>" . $sputum[$k]->WEIGHT .
        "</td></tr>";
        $k++;
    } else {
        echo "<tr align='center'><td height='21'>" .
        $monthName . "</td>";
        echo "<td>&nbsp;&nbsp;&nbsp;</td>";
        echo "<td>&nbsp;&nbsp;&nbsp;</td>";
        echo "<td>&nbsp;&nbsp;&nbsp;</td></tr>";
    }
}
?>
</table>

<?php echo form_open('register/edit_patient',
"name='patient_form') . "\n"; ?>
<input type="hidden" value="<?php echo $data[0]-
>TB_CASENO ?>" id="patient_caseno"
name="patient_caseno"/>
<center>
<?php echo form_submit('submit_edit', 'Update
Patient Details'); ?>
</center>

<?php echo form_close(); ?>

```

sputum.php

```

<h3>TB Treatment Information</h3>
<?php echo form_open('register/preview_register',
"name='tb_form'", $this->arr) . "\n"; ?>
<table width="100%">
<tr>
<td colspan="2" class="header2">Patient</td>
</tr>
<tr>
<td colspan="2">TB Case Number : <b><?php echo
echo $_POST['case_no']; ?></b></td>
</tr>
<tr>
<td colspan="2">Name : <b><?php echo
$_POST['patient_name'] . " " .
$_POST['patient_lname']; ?></b></td>
</tr>
<tr>
<td colspan="2">&nbsp;&nbsp;&nbsp;</td>

```



```

</tr>
<tr>
  <td colspan="2" class="header2">Treatment
  Result</td>
</tr>
<tr>
  <td>Date Treatment Started : </td>
  <td>
    <input type="text" name="start_date"
    size="23" id="_start_date" maxlength="10"
    readonly/>
    <i>(yyyy-mm-dd)</i>
  </td>
</tr>
<tr>
  <td>Treatment Outcome : </td>
  <td><select name="treat_out">
    <option value="">-Select one-</option>
    <option value="Cured" >Cured</option>
    <option value="Treatment Completed"
  >Treatment Completed</option>
    <option value="Died" >Died</option>
    <option value="Failed" >Failed</option>
    <option value="Defaulted"
  >Defaulted</option>
    <option value="Transferred Out"
  >Transferred Out</option>
  </select></td>
</tr>
<tr>
  <td>Date Treatment Ended : </td>
  <td> <input type="text" name="end_date"
  size="23" id="_end_date" maxlength="10"
  readonly/>
  <i>(yyyy-mm-dd)</i></td>
</tr>
<tr>
  <td colspan="2">&nbsp;</td>
</tr>
<tr>
  <td colspan="2" class="header2">Treatment
  Partner </td>
</tr>
<tr>
  <td>Partner : </td>
  <td><select name="partner">
    <option value="">-Select one-</option>
    <option value="PHM" >Provincial Health
  Nurse</option>
    <option value="RHM" >Rural Health
  Midwife</option>
    <option value="BHW" >Barangay Health
  Worker</option>
  </select>
</td>
</tr>

```

```

    <option value="FM" >Family
  Member</option>
    <option value="Other" >Other</option>
  </select></td>
</tr>
<tr>
  <td>TBDC Review (Y/N) : </td>
  <td><select name="review">
    <option value="">-Select one-</option>
    <option value="Y" >Yes</option>
    <option value="N" >No</option>
  </select></td>
</tr>
<tr>
  <td>Remarks : </td>
  <td><textarea name="remarks" rows="3"
  cols="40"></textarea></td>
</tr>
<tr>
  <td colspan="2">&nbsp;</td>
</tr>
<tr>
  <td colspan="2">&nbsp;</td>
</tr>
</table>

<h3>Sputum Examination Results / Weight </h3>
<table width="100%" >
  <tr>
    <td colspan="5" class="header2">Treatment
  </td>
</tr>
<tr align="center">
  <td width="10%">Month</td>
  <td width="27%">Date Examined <i>(yyyy-mm-
  dd)</i></td>
  <td width="21%">Result</td>
  <td width="15%">Weight</td>
  <td width="27%">&nbsp;</td>
</tr>
<tr align="center">
  <td height="39" align="center">0</td>
  <td>
    <input type="text" name="exam_date1"
    id="_exam_date" size="15"/>
  </td>
  <td>
    <select name="result1">
    <option value="">-Select one-</option>
    <option value="Positive">Positive</option>
    <option
  value="Negative">Negative</option>
  </select>
  </td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

```

```

        <option
value="Doubtful">Doubtful</option>
        </select>
    </td>
    <td><input type="text" maxlength="3" size="4"
name="weight1"/></td>
</tr>
<tr align="center">
    <td height="39" align="center">2nd</td>
    <td><input name="exam_date2" type="text"
id="_exam_date2" size="15"/>
    </td>
    <td><select name="result2">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>
        <option
value="Doubtful">Doubtful</option>
    </select></td>
    <td><input name="weight2" type="text"
id="weight2" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td height="39" align="center">3rd</td>

    <td><input name="exam_date3" type="text"
id="_exam_date3" size="15"/>
    </td>
    <td><select name="result3">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>
        <option
value="Doubtful">Doubtful</option>
    </select></td>
    <td><input name="weight3" type="text"
id="weight3" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td height="39" align="center">4th</td>

    <td><input name="exam_date4" type="text"
id="_exam_date4" size="15"/>
    </td>
    <td><select name="result4">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>
        <option
value="Doubtful">Doubtful</option>

```

```

        </select></td>
    <td><input name="weight4" type="text"
id="weight4" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td height="39" align="center">5th</td>

    <td><input name="exam_date5" type="text"
id="_exam_date5" size="15"/>
    </td>
    <td><select name="result5">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>
        <option
value="Doubtful">Doubtful</option>
    </select></td>
    <td><input name="weight5" type="text"
id="weight5" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td height="39" align="center">6th</td>

    <td><input name="exam_date6" type="text"
id="_exam_date6" size="15"/>
    </td>
    <td><select name="result6">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>
        <option
value="Doubtful">Doubtful</option>
    </select></td>
    <td><input name="weight6" type="text"
id="weight6" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td height="39" align="center"> > 7th</td>

    <td><input name="exam_date7" type="text"
id="_exam_date7" size="15"/>
    </td>
    <td><select name="result7">
        <option value="">-Select one-</option>
        <option value="Positive">Positive</option>
        <option
value="Negative">Negative</option>

```

```

        <option
value="Doubtful">Doubtful</option>
    </select></td>
    <td><input name="weight7" type="text"
id="weight7" size="4" maxlength="3"/></td>
</tr>
<tr align="center">
    <td colspan="5" class="header2">&nbsp;</td>
</tr>
<tr>
    <td colspan="5"></td>
</tr>
<tr>
    <td></td>
    <td align="right"><input type="button" value="
<< Previous Section " onclick="javascript:goBack()
"/></td>
    <td align="right"><?php echo
form_submit('mysubmit', ' Preview '); ?></td>
</tr>
</table>
<input type="hidden" name="month1" value="0"/>
<input type="hidden" name="month2" value="1"/>
<input type="hidden" name="month3" value="2"/>
<input type="hidden" name="month4" value="3"/>
<input type="hidden" name="month5" value="4"/>
<input type="hidden" name="month6" value="5"/>
<input type="hidden" name="month7" value="6"/>
<?php echo form_close(); ?>

```

print_report_province.php

```

<?php
if ($_POST['quarter2'] == 1) $quarters = "1st";
elseif ($_POST['quarter2'] == 2) $quarters = "2nd";
elseif ($_POST['quarter2'] == 3) $quarters = "3rd";
elseif ($_POST['quarter2'] == 4) $quarters = "4th";
else $quarters = "----";

?>

<table height="520" style="border: solid;">
    <tr>
        <td width="100%" align="center"><p>NTP
Quarterly Report on All TB Cases </p>
        <table width="100%" style="border: solid;">
            <tr>
                <td width="250">Name of Region : <b>
<?php echo $this->session->userdata('region');
?></b> </td>
                <td width="170">Patients registered
during the :</td>

```

```

        <td width="242">Date Reported : <?php
echo $_POST['report_date']; ?></td>
    </tr>
    <tr>
        <td>Name of PHO/CHO : <b>
        <?php if($this->session-
>userdata('usertype') == 'PhilCAT Coordinator') {
            echo "PhilCAT";
        } else {
            echo $this->session-
>userdata('pho_name');
        } ?></b> </td>
        <td rowspan="2" align="center"><?php
echo $quarters; ?> Quarter of Year <?php echo
$_POST['year2']; ?> </td>
        <td>Prepared by : <b> <?php echo $this-
>session->userdata('name')." ". $this->session-
>userdata('lname'); ?></b></td>
    </tr>
    <tr>
        <td>Name of DOTS Facility : <b> <?php
echo $this->session->userdata('pho_name'); ?></b>
</td>
        <td>Designation : <b> <?php echo $this-
>session->userdata('desig'); ?></b></td>
    </tr>
</table> </td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
<tr>
    <td>A. All TB Cases Registered during the
<b><?php echo $quarters; ?></b> quarter : </td>
</tr>
<tr>
    <td><table width="100%" height="186"
border="1" cellspacing="0">
        <tr class="header2">
            <td width="65"
rowspan="3">&nbsp;</td>
            <td colspan="2" rowspan="2">New
Smear Positive </td>
            <td colspan="2" rowspan="2">Relapse</td>
            <td colspan="2" rowspan="2">Trans-
in</td>
            <td colspan="2" rowspan="2">RAD</td>
            <td colspan="2" rowspan="2">Treatment
Failure </td>
            <td colspan="4">Other</td>

```

```

        <td colspan="2" rowspan="2">Smear
Negative </td>
        <td colspan="2" rowspan="2">EP</td>
</tr>
<tr class="header2">
        <td colspan="2">Positive</td>
        <td colspan="2">Negative</td>
</tr>
<tr class="header2">
        <?php for($i = 0; $i < 9; $i++) {
            echo "<td width='23px'>M</td>";
            echo "<td width='23px'>F</td>";
        } ?>
</tr>
<tr align="center">
        <td class="header2">Public</td>
        <?php
            $types = array("New Pos", "Relapse",
"Trans-In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg", "EP");
            $sexs = array ("M", "F");
            foreach ($types as $type) {
                foreach ($sexs as $sex) {
                    echo "<td>"
.$_POST['all']['Public'][$type][$sex] . "</td>";
                }
            }
        ?>
</tr>
<tr align="center">
        <td class="header2">Private</td>
        <?php
            $types = array("New Pos", "Relapse",
"Trans-In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg", "EP");
            $sexs = array ("M", "F");
            foreach ($types as $type) {
                foreach ($sexs as $sex) {
                    echo "<td>"
.$_POST['all']['Private'][$type][$sex] . "</td>";
                }
            }
        ?>
</tr>
<tr align="center">
        <td class="header2">Subtotal</td>
        <?php
            $types = array("New Pos", "Relapse",
"Trans-In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg", "EP");
            $sexs = array ("M", "F");

```

```

        foreach ($types as $type) {
            foreach ($sexs as $sex) {
                $sub =
$_POST['all']['Public'][$type][$sex] +
$_POST['all']['Private'][$type][$sex];
                echo "<td>" . $sub . "</td>";
            }
        }
    ?>
</tr>
<tr align="center">
        <td class="header2">Total</td>
        <?php
            $types = array("New Pos", "Relapse",
"Trans-In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg", "EP");
            $sexs = array ("M", "F");
            foreach ($types as $type) {
                $subm =
$_POST['all']['Public'][$type]['M'] +
$_POST['all']['Private'][$type]['M'];
                $subf =
$_POST['all']['Public'][$type]['F'] +
$_POST['all']['Private'][$type]['F'];
                $totals = $subm + $subf;
                echo "<td colspan='2'>" . $totals .
"</td>";
            }
        ?>
</tr>
</table></td>
</tr>
<tr>
        <td>&nbsp;</td>
</tr>
<tr>
        <td>B. Breakdown of New Pulmonary Smear-
Positive cases by age and sex : </td>
</tr>
<tr>
        <td><table width="100%" height="121"
border="1" cellspacing="0">
            <tr class="header2">
                <td colspan="2">0 - 9 </td>
                <td colspan="2">10 - 14</td>
                <td colspan="2">15 - 24</td>
                <td colspan="2">25 - 34 </td>
                <td colspan="2">35 - 44 </td>
                <td colspan="2">45 - 54</td>
                <td colspan="2">55 - 64 </td>
                <td colspan="2">65 and above</td>

```

```

        <td colspan="3">Total</td>
</tr>

<tr class="header2">
    <?php for($i = 0; $i < 9; $i++) {
        echo "<td width='30px'>M</td>";
        echo "<td width='30px'>F</td>";
    } ?>
    <td>Total</td>
</tr>
<tr align="center">
    <?php
    $sexs = array ("M", "F");
    $summ = 0;
    $sumf = 0;
    for ($i = 0; $i < 8; $i++) {
        foreach ($sexs as $sex) {
            echo "<td>"
.$_POST['cat_age'][$i][$sex] . "</td>";
            if ($sex == 'M') {
                $summ = $summ +
$_POST['cat_age'][$i][$sex];
            } else {
                $sumf = $sumf +
$_POST['cat_age'][$i][$sex];
            }
        }
    }
    $totalA = $summ + $sumf;
    echo "<td>" . $summ . "</td>";
    echo "<td>" . $sumf . "</td>";
    echo "<td>" . $totalA . "</td>";
?>
</tr>
</table></td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
<td>C. Treatment Regimen given : </td>
</tr>
<tr>
    <td><table width="100%" border="1"
cellspacing="0">

        <tr class="header2">
            <td width="174">Type of Treatment
given: </td>
            <td width="131">Category I </td>
            <td width="141">Category II </td>
            <td width="139">Category III</td>
        </tr>

```

```

        <tr align="center">
            <td height="30" class="header2">Cases
initiated Treatment </td>
            <td><?php echo $_POST['cat']['cat1'];
?></td>
            <td><?php echo $_POST['cat']['cat2'];
?></td>
            <td><?php echo $_POST['cat']['cat3'];
?></td>
        </tr>
</table></td>
</tr>
<tr>
    <td>&nbsp;</td>
</tr>
</table>

```

category_report.php

```

<?php
if ($_POST['quarter2'] == 1) $quarters = "1st";
elseif ($_POST['quarter2'] == 2) $quarters = "2nd";
elseif ($_POST['quarter2'] == 3) $quarters = "3rd";
elseif ($_POST['quarter2'] == 4) $quarters = "4th";
else $quarters = "---";
?>
<h3><?php echo $quarters; ?> Quarter of Year
<?php echo $_POST['year2']; ?> of
<?php if(isset($_POST['mysubmit'])) echo
$_POST['health_fac'];
else echo $this->session->userdata('pho_name');
?></h3>
<table width="100%">
    <tr><td class="header2" width="100%">Counting
Sheets</td></tr>
    <tr><td >&nbsp;</td></tr>
    <tr>
        <td align="center">
            <table width="600" border="1"
cellspacing="0" height="70px">
                <tr>
                    <td colspan="3" class="header2">Total
Number of Patients under the following Categories
</td>
                    </tr>
                    <tr align="center">
                        <td width="33%">Category I </td>
                        <td width="33%">Category II </td>
                        <td width="33%">Category III</td>
                    </tr>
                    <tr align="center">
                        <td><?php echo $_POST['cat']['cat1'];
?></td>

```

```

        <td><?php echo $_POST['cat']['cat2'];
?></td>
        <td><?php echo $_POST['cat']['cat3'];
?></td>
    </tr>
</table>
<br>
<br>
<table width="600" border="1"
cellspacing="0" height="200px">
    <tr class="header2">
        <td width="133">Type of Patient </td>
        <td width="71">Cured</td>
        <td width="75">Tx Completed </td>
        <td width="65">Died</td>
        <td width="65">Failed</td>
        <td width="68">Defaulted</td>
        <td width="78">Trans-out</td>
    </tr>
    <?php
    $outs = array("Cured", "Treatment
Completed", "Died", "Failed", "Defaulted",
"Transferred Out");
    $outs2 = array("Treatment Completed",
"Died", "Failed", "Defaulted", "Transferred Out");
    ?>
    <tr>
        <td>New Smear-Positive</td>
        <?php
        foreach ($outs as $out) {
            echo "<td align=center>"
.$_POST['crazy']['New Pos'][$out] . "</td>";
        }
        ?>
    </tr>
    <tr>
        <td>New Smear-Negative </td>
        <td bgcolor="#cdcdcd">&nbsp;&nbsp;&nbsp;</td>
        <?php
        foreach ($outs2 as $out) {
            echo "<td align=center>"
.$_POST['crazy']['New Neg'][$out] . "</td>";
        }
        ?>
    </tr>
    <tr>
        <td>Transferred-In </td>
        <?php
        foreach ($outs as $out) {
            echo "<td align=center>"
.$_POST['crazy']['Trans-In'][$out] . "</td>";
        }
    </tr>

```

```

        ?>
    </tr>
    <tr>
        <td>Relapse</td>
        <?php
        foreach ($outs as $out) {
            echo "<td align=center>"
.$_POST['crazy']['Relapse'][$out] . "</td>";
        }
        ?>
    </tr>
    <tr>
        <td>Treatment Failure </td>
        <?php
        foreach ($outs as $out) {
            echo "<td align=center>"
.$_POST['crazy']['Treatment Failure'][$out] . "</td>";
        }
        ?>
    </tr>
    <tr>
        <td>Return After Default </td>
        <?php
        foreach ($outs as $out) {
            echo "<td align=center>"
.$_POST['crazy']['Default'][$out] . "</td>";
        }
        ?>
    </tr>
</table>
<br>
<br>
<table width="600" border="1"
cellspacing="0" >
    <tr class="header2" >
        <td colspan="2" rowspan="2">New
Smear Positive </td>
        <td colspan="2"
rowspan="2">Relapse</td>
        <td colspan="2" rowspan="2">Trans-
in</td>
        <td colspan="2" rowspan="2">Return
After Default </td>
        <td colspan="2" rowspan="2">Treatment
Failure </td>
        <td colspan="4">Other</td>
        <td colspan="2" rowspan="2">Smear
Negative </td>
        <td colspan="2" rowspan="2">EP</td>
    </tr>
    <tr class="header2">
        <td colspan="2">Positive</td>
        <td colspan="2">Negative</td>
    </tr>

```

```

</tr>
<tr align="center" height="20px">
  <?php for($i = 0; $i < 9; $i++) {
    echo "<td width='25px'>M</td>";
    echo "<td width='25px'>F</td>";
  } ?>
</tr>
<tr cellpadding="0" height="20px">
  <?php
    $types = array("New Pos", "Relapse",
"Trans-In", "Default", "Treatment Failure", "Other
Pos", "Other Neg", "New Neg", "EP");
    $sexs = array ("M", "F");
    foreach ($types as $type) {
      foreach ($sexs as $sex) {
        echo "<td align=center>"
.$_POST['sex'][$type][$sex] . "</td>";
      }
    }
  ?>
</tr>
</table>
</td>
</tr>
</table>

```

quarter_report.php

```

<?php
if ($_POST['quarter'] == 1) $quarters = "1st";
elseif ($_POST['quarter'] == 2) $quarters = "2nd";
elseif ($_POST['quarter'] == 3) $quarters = "3rd";
elseif ($_POST['quarter'] == 4) $quarters = "4th";
else $quarters = "----";

$types = array("New Pos", "New Neg", "Relapse",
"Treatment Failure");
$outcomes = array("Cured", "Treatment Completed",
"Died", "Failed", "Defaulted", "Transferred Out");
?>

<table style="border: solid;">
  <tr>
    <td width="100%" height="30"
class="header2">NTP Quarterly Report on the
Treatment Outcome of Pulmonary TB Cases </td>
  </tr>
  <tr>
    <td height="100"><table width="100%"
style="border: solid;">
      <tr>

```

```

        <td width="250">Name of Region : <b>
<?php echo $this->session->userdata('region');
?></b> </td>
        <td width="180" align="center">Patients
registered</td>
        <td width="242">Date Reported : <b>
<?php echo $_POST['report_date']; ?></b></td>
      </tr>
    <tr>
      <td>Name of PHO/CHO :<b>
        <?php if($this->session-
>userdata('usertype') == 'PhilCAT Coordinator') {
          echo "PhilCAT";
        } else {
          echo $this->session-
>userdata('pho_name');
        } ?></b> </td>
      <td align="center">during the:</td>
      <td>Prepared by : <b> <?php echo $this-
>session->userdata('name')." ". $this->session-
>userdata('lname'); ?></b></td>
    </tr>
    <tr>
      <td>Name of DOTS Facility :<b align =
"center"> <?php echo $_POST['health_fac']; ?></b>
</td>
      <td align="center"><b><?php echo
$quarters; ?> Quarter of</br> Year <?php echo
$_POST['year']; ?></b> </td>
      <td>Designation : <b align = "center">
<?php echo $this->session->userdata('desig');
?></b></td>
    </tr>
  </table></td>
</tr>
<tr>
  <td><table width="100%" height="300"
border="1" cellpadding="0">
    <tr class="header2">
      <td width="110">Total Number of
Pulmonary TB Cases </td>
      <td width="170">Type</td>
      <td width="39">Cured</td>
      <td width="58">Comp Tx </td>
      <td width="38">Died</td>
      <td width="41">Failed</td>
      <td width="43">Default</td>
      <td width="50">Trans Out </td>
      <td width="52">Total No. Evaluated
</td>
    </tr>
  </table>
  <td>&nbsp;</td>

```



```

        </tr>
    </table></td>
</tr>
<tr class="header2"><td>&nbsp;</td></tr>
<tr><td>&nbsp;</td></tr>
<tr>
    <td><table width="100%" border="0">
        <tr>
            <td colspan="2">* Of these, <b><?php
echo $_POST['ex']['excluded']; ?></b> were excluded
from evaluation of chemotherapy for the following
reasons: </td>
        </tr>
        <tr>
            <td width="119" align="right"><u><?php
if($_POST['ex']['ex_trans'] != 0) echo "X";?></u>
&nbsp;</td>
            <td width="523">Trans-in</td>
        </tr>
        <tr>
            <td align="right"><u><?php
if($_POST['ex']['ex_ep'] != 0) echo "X";?></u> &nbsp;</td>
            <td>Extra-pulmonary</td>
        </tr>
        <tr>
            <td align="right"><u><?php
if($_POST['ex']['ex_other'] != 0) echo "X";?></u>
&nbsp;</td>
            <td>Other</td>
        </tr>
    </table></td>
</tr>
</table>
<?php echo
form_open('report/display_print_quarter_report0',
"name='create_rate'") . "\n"; ?>

<input type="hidden" value="<?php echo
$_POST['year']?>" name="year"/>
<input type="hidden" value="<?php echo
$_POST['quarter']?>" name="quarter"/>
<input type="hidden" value="<?php echo
$_POST['health_fac']?>" name="health_fac"/>
<input type="hidden" value="<?php echo
$_POST['report_date']?>" name="report_date"/>

<?php echo form_submit('mysubmit', 'download
report'); ?>
<?//php echo form_submit('mysubmit2', 'printer
friendly version'); ?>
<?php echo form_close(); ?>

```

XI. ACKNOWLEDGEMENTS

To my mommy and daddy, thanks for providing me everything I needed, esp. the laptop, to get by and finish this course. Thank you for your non-stop reminders for me to do and finish my SP. Thank you for all the breakfast meals you've prepared when I was doing this SP. Also, thanks for supporting my decision of choosing computer science and for being so lenient when it comes to my studies.

To my professors, thank you for inspiring me in ways you never thought you did. Ma'am Sheila and my SP adviser Doc Magboo, thank you for your mga constructive criticisms and inputs/ suggestions on how I can possibly improve the system. SP was hard but nothing beats the feeling of seeing my own SP's hardbound copy. Thanks you also to Sir Chua, Sir Co, Ma'am Carpio, Sir Solano, Sir JM, Sir Bernie and of course Ate Eden, for all the consideration and patience for four long years.

To my alma mater, UP Manila, thank you for all those valuable experiences and life lessons you've taught me. To Dean Imperial, thank you for this great chance you've given me.

To my blockmates, college friends, acquaintances thank you for the memories. May we see each other again.

To those who helped me during my SP proposal and defense days. I very much thank you, for without you guys I wouldn't have made it through those days. ☺

To the only person who has actively participated and actually did something in helping me complete this paper. Thank you.

A special mention to the Yumol family for letting me stay at their place the night before my SP proposal. Tita Doris, I thank you for all the help and support you've given us whenever you can.