ASHESI UNIVERSITY COLLEGE

HEALTH RECORDS MANAGEMENT SYSTEM FOR COMMUNITY HEALTH CENTERS

By

FLORENCE LOMOTIORKOR JONES

Applied Project submitted to the Department Computer Science,

Ashesi University College

In partial fulfillment of Science degree in Computer Science

APRIL 2013

DECLARATION

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: ……………………………………………………………………

Candidate's Name: ……………………………………………………

Date: ……………………………..

I hereby declare that the preparation and presentation of the dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor's Signature: …………………………………………………………………..

Supervisor's Name: …………………………………………………….

Date: ………………………………………..

ACKNOWLEDGEMENTS

My sincere gratitude goes to my supervisor, Mr. Aelaf Dafla for guiding me throughout this project and providing ideas leading this project to fruition. I also want to thank my parents for being understanding and very supportive throughout this period. Finally, I would like to thank my colleagues Abigail Asamoah, Fabiola Amedo and Efua Gambrah-Sampaney for all the support and ideas given me during this period.

ABSTRACT

Individuals resident in rural areas have the problem of long distances between their places of abode and the nearest health center. The goal of organizations and research bodies has been to come up with solutions to bridge this gap. Some solutions that have evolved include setting up community health centers in various rural areas and adopting the use of mobile technology to bring health care to the door step of residents in these areas. Some key aspects of such solutions include ensuring mobility and providing tools to record data. Due to the ever increasing number of people who own mobile phones, mobile technology has become one solution with various areas of interests in order to develop mobile applications that contribute to solving this problem. A mobile application that provides Community Health officers with a tool for recording data to be made available to district officers for decision making will help in various ways to reduce epidemics and assuage existent epidemics in rural areas.

Contents

ACRONYMNS

HTML- Hypertext Markup Language

PHP- Hypertext Preprocessor

CSS- Cascading Style Sheets

MySQL- Structured Query Language

GPS- Global Positioning System

CHO- Community Health Officer

JSON- JavaScript Object Notation

PDF- Portable Document Format

SMS-Short Message Service

IDE- Integrated Development Environment

API- Application Programming Interface

HRMS-Health Records Management System

MDGs- Millenium Development Goals

UI- User Interface

GHS- Ghana Health Service

SDK-Software Development Kit

DOM- Document Object Model

CHAPTER 1: INTRODUCTION

1.0 INTRODUCTION OVERVIEW

Health care is a very vital component in every day existence. There are a myriad of resources needed to provide quality health care for every individual. One of such resources includes data or information needed to make decisions efficiently. In Ghana, the organization or body meant to provide communities with basic health care services is the does this through the Community Health Officers (CHO). These officers provide basic health service and valuable health surveillance service to the Ghana Health Service (GHS) by reporting back their observations to the district health office for decision making purposes. These officers in addition to the health services they provide keep health records on paper which is inefficient given the current global shift towards technological solutions and digital data storage. The Health Records Management System was designed to provide the management of a community health center with a tool that will assist in effective data recording and decision making in order to ensure quality health care delivery.

The motivation to carry out this project stems from a variety of sources. These sources include the current MoTECH project implemented by the Grameen Foundation and the Millennium Development Goals (MDGs). One other motivation to undertake this project is a personal fascination about the Android Platform by Google and its powerful functionalities and how these functionalities can be used for many purposes that yield a greater good. The

Grameen Foundation started The Mobile Technology for Community Health (MoTECH) project which aimed at providing a mobile phone intervention to improve maternal and child health [1]. This project stemmed from the need to support the achievement of the Millennium Development Goals of improving maternal health and decreasing infant and maternal mortality [2]. Mobile technology is an even more effective way of carrying out daily activities. Its portability and mobility provides a convenient and effective way of performing tasks of various forms. This makes the mobile technology the perfect tool for implementing a platform that allows for efficient data collection. This is because in a bid to solve the problem of distance, mobility and bringing health care to the door step of residents in a particular community is a feasible solution. Android's implementation of the Global Positioning System (GPS) with its in built local database makes it an attractive resource for the implementation of the project.

1.1 BACKGROUND

Over the years, the government of Ghana as well as the Ghana health Service under the caption "Health for All" sought to improve health care in the country [3]. In this quest, one area of concern has been the various rural areas in the country. The Ghana Community-based Health Planning and Services Initiative was introduced in a bid "to improve accessibility, efficiency and quality of health and family planning care" [3]. This initiative resulted from an experiment in Navrongo under the Navrongo Health Research Centre (NHRC). Findings however indicated that "70% of Ghanaians still lived over 8km from the nearest health care provider" and the rural infant mortality rates were still sky-rocketing as much as double that of the urban rates [3].

Some solutions provided for this problem include the Village Health Workers (VHWs) which was to provide affordable health services. Scaling up this initiative was not successful which led to the introduction of Community Health Nurses who were to provide more professional services to members of a community. The Community-based Health Planning and Services (CHPS) was then finally introduced in 1990 as a national health policy that aimed at reducing the barriers of geographical access to health care. This program involved providing a trained Community Health Officer (CHO) who would be resident in the community as well as a Community Health Compound to facilitate the activities of a CHO. The role of Community Health Officers was to provide mobile doorstep services to community residents by travelling on a motorcycle from compound to compound. Some services they provide on such trips include "immunizations, family planning, supervising delivery, antenatal/postnatal care, treatment of minor ailments as well as health education" [3]. CHOs are supported by volunteers who assist in various ways which include maintenance of community records or registers.

As earlier described, one role of community health workers is to provide data to the district officers for effective decision making. The efficiency of a CHO is heavily dependent on how successful door to door or compound to compound visits are. On such visits, a CHO is responsible for collecting data which will be used to update community registers. Collecting data on paper however, poses a number of difficulties. These include:

- ***Mobility***: It is more difficult to carry paper around with the idea of collecting data. This is because important papers could be forgotten at the source. Carrying a mobile device however ensures that all vital elements are present since they all exist on that one mobile device.
- ***Possible Loss of Data***: Apart from the difficulty of carrying papers around, some of these papers can be lost en route. This risk is however reduced when data is recorded on a mobile device since backups of data are saved on a remote database make it next to impossible to lose this data if the device is lost or damaged.
- ***Difficulty in Data Analysis***: Having data on paper makes it difficult to analyze information. A typical example is collecting numbers on paper and having to enter all these numbers on a spreadsheet in order to easily analyze them. This makes it hampering and time consuming and could probably reduce the motivation of workers.

The Health Records Management System (HRMS), which is developed as part of this project, is a tool that runs on the Android platform. This tool also provides a web application that will assist in retrieving data at the district level to ensure that data gathered can be easily accessible and useful. The Health Records Management System will benefit the society in various ways. These include:

- ***The ability to facilitate decision making***: The HRMS provides an easy way to retrieve data on a universal platform which is the web in order to ensure universal access for key decision makers in the health care industry. The tool also provides some functionality for analyzing data retrieved in order to further enhance decision making.
- ***Benefit to members of community***: This tool provides community health officers with an android application that allows them to enter information on patients. With this information, the officers can send SMS reminders for instance to expectant mothers on the day of an antenatal appointment to ensure they do not miss this appointment. These SMS reminders can also be sent to parents to remember to bring their children for weighing. This is very important as some patients do not keep track of such dates and end up missing these appointments.
- ***Support Millennium Development Goals***: This project implements components that support maternal and infant health. These are two elements embedded in the MDGs. By benefitting these members of the society who are key figures in the target of the MDGs, the HRMS will

serve as a tool for supporting the achievement of the Millennium Development Goals.

CHAPTER 2: DESIGN

2.1 USER REQUIREMENTS

The Health Records Management System is anticipated to fulfill the following user requirements:

Community Health Officer

- Community health officers should be able to view forms and enter information where necessary
- CHOs should be able to view information where necessary
- CHOs must also be able to have access to the application at all times.

District Officers

- District officers should be able to log in to the system to access data
- District officers should be able to view information needed
- They should be able to find and download data in Excel or PDF format
- They should also be able to view GPS coordinates on a map with details of specific household and health issue encountered at that household

2.2 SYSTEM REQUIREMENTS

2.2.1 FUNCTIONAL REQUIREMENTS

The health records management system is built for a Community Health Officer as well as District health management team. Community Health Officers will be able to enter information on patients under various categories and District officers will be able to view information entered by CHOs for decision purposes. The system has a web interface for retrieval of data by District officers and an android application for data entry by CHOs. The android application consists of components and these are as follows: the *birth registration system*, *the expectant mothers' registration system, door-to-door visits with GPS log system, the health incidence log system and the inventory of medication system*. The functional requirements of these components are explained as follows:

The Birth Registration System: This system takes the basic details of a child when he or she is born. The basic information that is recorded include:

- Name of Child
- Date of Birth
- Name of Mother
- Name of Father
- Contact information of Parents

This system will also keep track of the weighing dates of the child. With these dates recorded in the database, the system will send SMS reminders to the parents of the child based on contact information provided. An SMS confirmation will also be sent to the parents to confirm the registration of the child with the details of the data collected during registration.

The Expectant Mothers' Registration System: This system will record the basic information of an expectant mother. The information that will be recorded includes:

- Name of mother
- Due Date (this will record the due date based on speculated time of conception.)
- Contact information of mother
- Name of community

This system will send SMS reminders when it is time for an expectant mother to go for antenatal care.

Door-to-door visit with GPS Log: This component will keep track of the location of patients in the community. When community health officers go on door-to-door visits, they will be able to log the GPS coordinates of their location in the database. This system will also log the time in and time out of the CHO on each visit. The information that will be recorded includes:

- Name of Household (Name of head of a household)
- Location coordinates (Longitude & Latitude. These values will be provided by the system)
- Time in
- Time out

- Name of CHO
- Date of visit
- Health incidence or condition (This will record health condition observed on visit)
- Name of community

Health Incidence Log System: This system logs the health incidence of a patient at any time if need be. For instance if a patient visits the health center, the CHO will record the health condition of the patient. The details recorded in the database include:

- Name of Patient
- Health Condition or incidence information
- Date recorded
- Name of community

Inventory of Medication: There is the need to keep track of the medication at the community center. This will record basic information on the particular medication and when this dose is administered, the system calculates based on an assumption, the amount remaining and this entered into the database. The information that will be recorded includes:

- Name of Medication
- Amount Acquired
- Date Acquired
- Amount Remaining (Calculated when a button is clicked)
- Date Recorded

The system provides a web interface for district officers to view data entered in the database. This component of the application must implement the following functional requirements:

- Provide data needed for decision making purposes
- Generate reports in PDF and Excel format
- Plot GPS coordinates on a map

2.2.2 NON-FUNCTIONAL REQUIREMENTS

The Health Records Management System is expected to meet the following non-functional requirements:

Performance and Usability

- Records provided in the system must be up to date, readily available and easily accessible.
- For the web application, when users request PDF and Excel reports of data, they must be readily available for download.
- The application must reduce bandwidth usage by reducing the amount of data updates done via an internet connection. In order to ensure this, any data retrieved to update the user interface of the mobile application must be retrieved from the local database (SQLite).

Security

- The web application for the district officers must grant every user a unique identification in the form of a username and a password.

- The administrator will be the only individual to update users in the system. This way not everyone can register to view the data in the system.

Privacy

- All data entered into the system by the community health officers must be kept private and made available only to authorized users

Reliability and Availability

- The system must be up and running 24 hours a day to make data entry and retrieval reliable
- The system must provide a store and forward when there is not connectivity in order to render the system functional at all times.

Maintainability

- The system must provide mechanisms for effective error detection and correction

2.2.3 SYSTEM ARCHITECTURE

The Figure 1 below shows the system architecture of the Health Records Management System. The system architecture is modeled in a layered architectural pattern. In this diagram, the database layer constitutes the SQLite and MySQL. SQLite is the local database on the mobile device implemented to store data needed to update the user interface as well as validate log in details of CHOs and the system administrator. MySQL is the remote database that stores data from the mobile application. This data is needed to provide information on the web application for viewing by the District officers.

The data access layer is the link between the database layer and the application layer. This is where data is processed for saving or retrieval purposes.

The application layer consists of various components that make up the application. These components include birth registration, health incidence log, door to door GPS log, inventory of medication logs and registration of expectant mothers as well as the web application for district officers. This layer provides forms where the community health officers enter data which is saved in the remote database. These data are needed to update the user interface and also to reduce the amount of data entered by a community health officer. The layer also provides a web application that allows district officers to retrieve data needed for decision making purposes. The application user interface provides an interface for every user in the system. CHOs have access to the forms in the android application which they use to enter information. District officers also have an interface in the form of a web application where they can retrieve data saved in the database by CHOs. The system administrator being responsible for the entry of data such as information on nurses, district officers as well as communities that use the application has an android application which provides an interface to enter such information.

Parents and expectant mothers receive SMS confirmations of registration as well as SMS reminders for appointments.

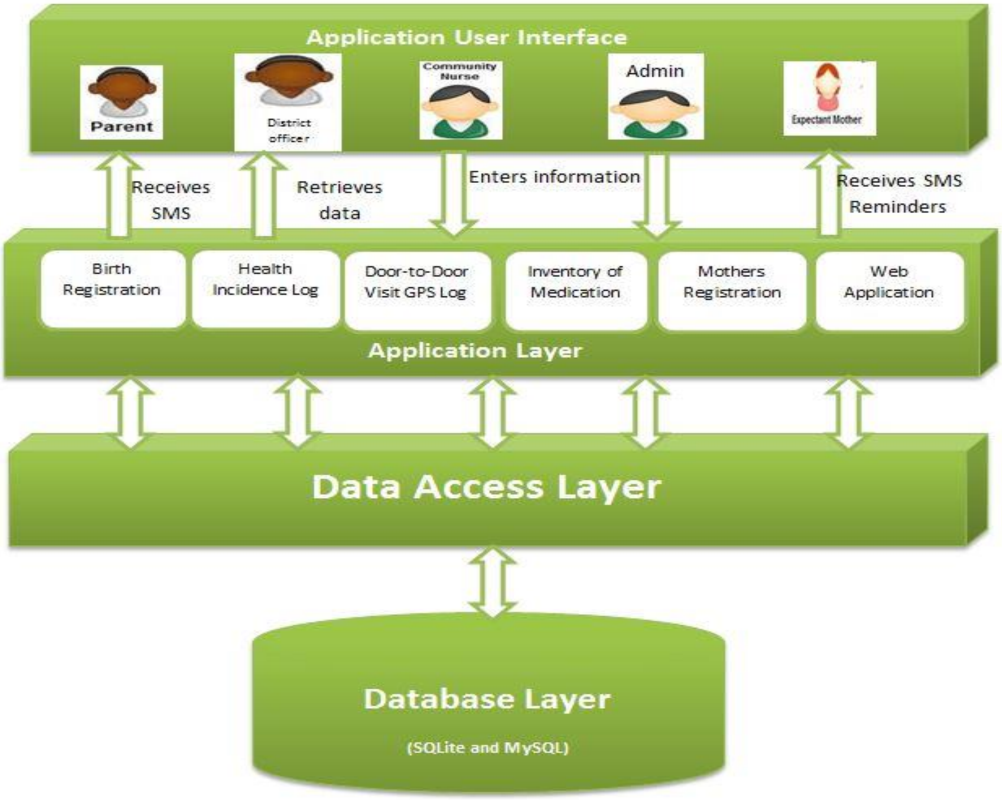


Figure 1 System Architecture

2.2.4 DATABASE ARCHITECTURE

The remote database (MySQL) is designed to store all the data from the forms in the mobile application and all data that is retrieved and display in the web application is retrieved from the remote database. The table implements some relations in the form of foreign key constraints as a way of setting a limit on what kind of data can be entered in the database in relation to parent tables. The figure below shows the database design:

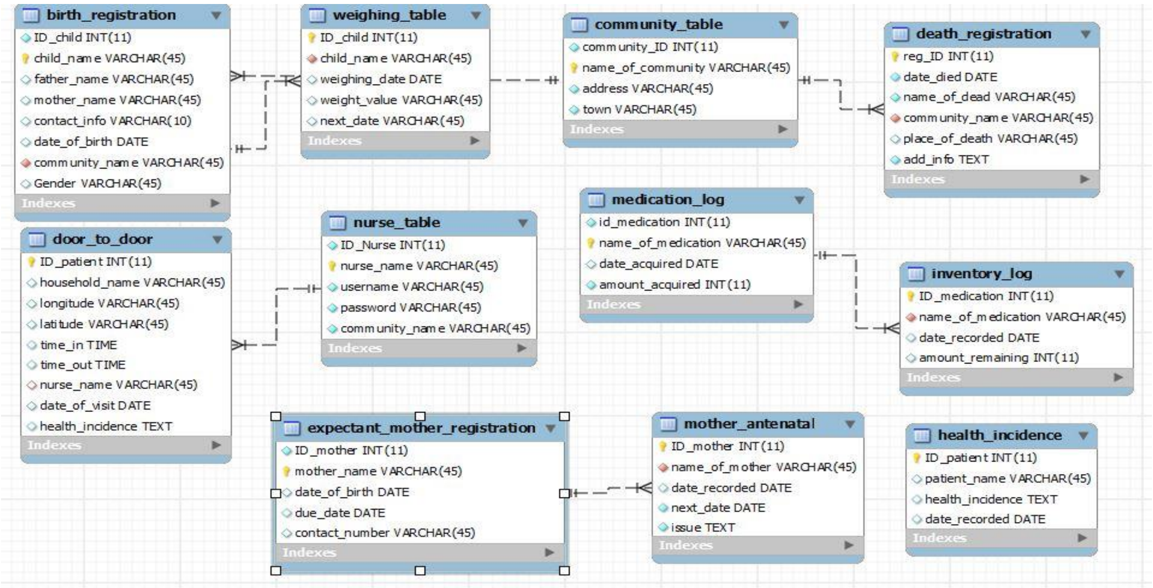


Figure 2 Database Design

The database is designed in order to restrict certain entries in the table. For instance, details for weighing cannot be entered for a child who has not been registered in the birth registration system. An expectant mother whose details are entered in the antenatal care table must be registered in the mother's registration table. The following table shows the primary keys in this table and the foreign keys that relate to them.

Primary Keys	Foreign Keys
Community_table (name_of_community)	Birth_registration (name_of_community), death_registration (name_of_community), nurse_table (name_of_commmunity), door_to_door (name_of_community)
Birth_registration (child_name)	Weighing_table (child_name)
Expectant_mother_registration (name_of_mother)	Mother_antenatal (mother_name)
Medication_log (name_of_medication)	Inventory_logs (name_of_medication)

Table 1 Primary keys and foreign keys

2.2.5 USE CASES

The diagram below shows a use case diagram indicating how the application will be used by the community officer as well as how a district health officer will interact with the web application.

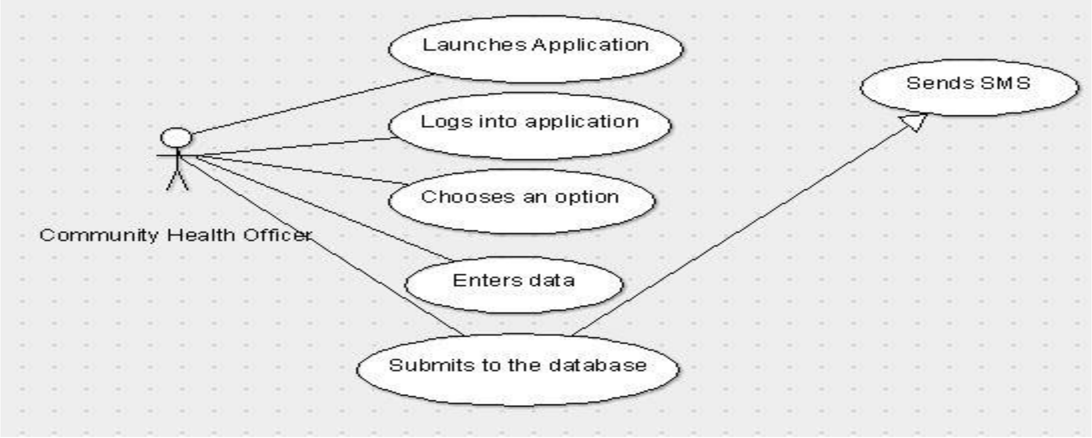


Figure 3 Use case diagram for CHO

The use case diagram above indicates how a community health officer will interact with the system. The role of a community health officer is to enter data based on the particular components implemented into the system. In a typical scenario of entering data into the birth registration component, the CHO logs in and launches the application and chooses the birth registration option. When he or she is done entering the data and clicks submit, the data is saved in the database. The officer then clicks on the send SMS button which sends a confirmation SMS to the parent of the child being registered with the details of the registration.

In another scenario, a district officer visits the Health Records Management System web application to view the data needed for decision making purposes. In order to access the data, he or she must log in with details provided by a system administrator. When the user is logged in, he or she has the option of viewing various datasets on the website and based on need, the user can develop PDF of excel reports of the data being viewed at a point in time. Figure 4 below shows a use case diagram of this scenario:

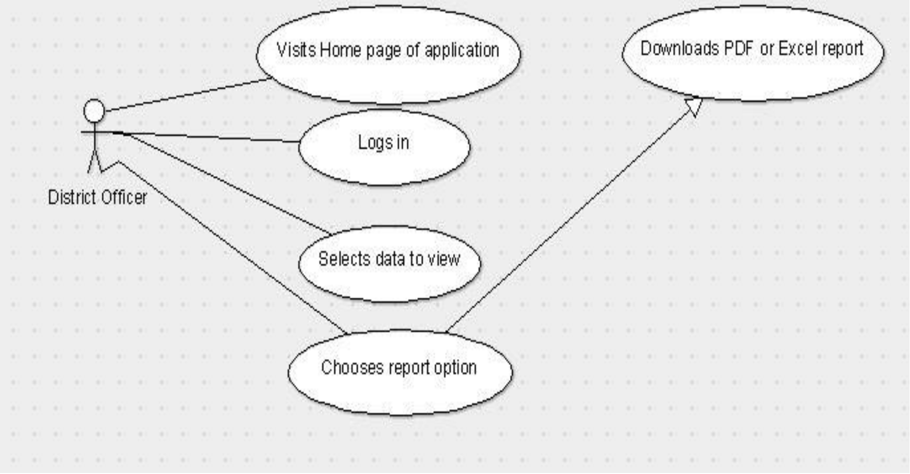


Figure 4 Use case diagram for district officer

Figure 5 indicates the use case diagram which shows how an administrator interacts with the system. In a typical scenario, the administrator logs into the admin account of the application. The system administrator is responsible for providing information on community and district officers as well as information on communities currently using the system.

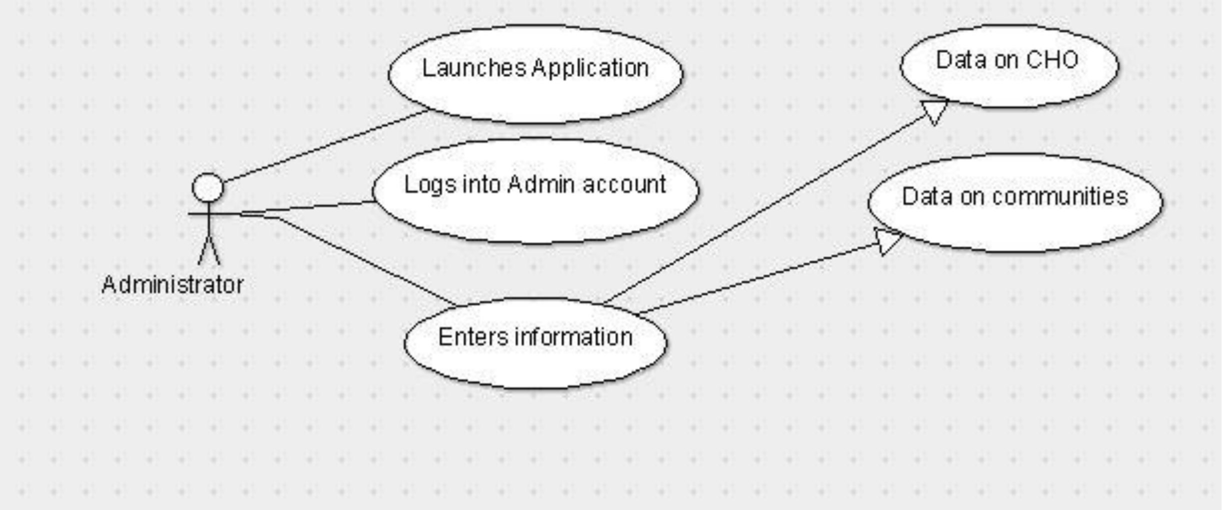
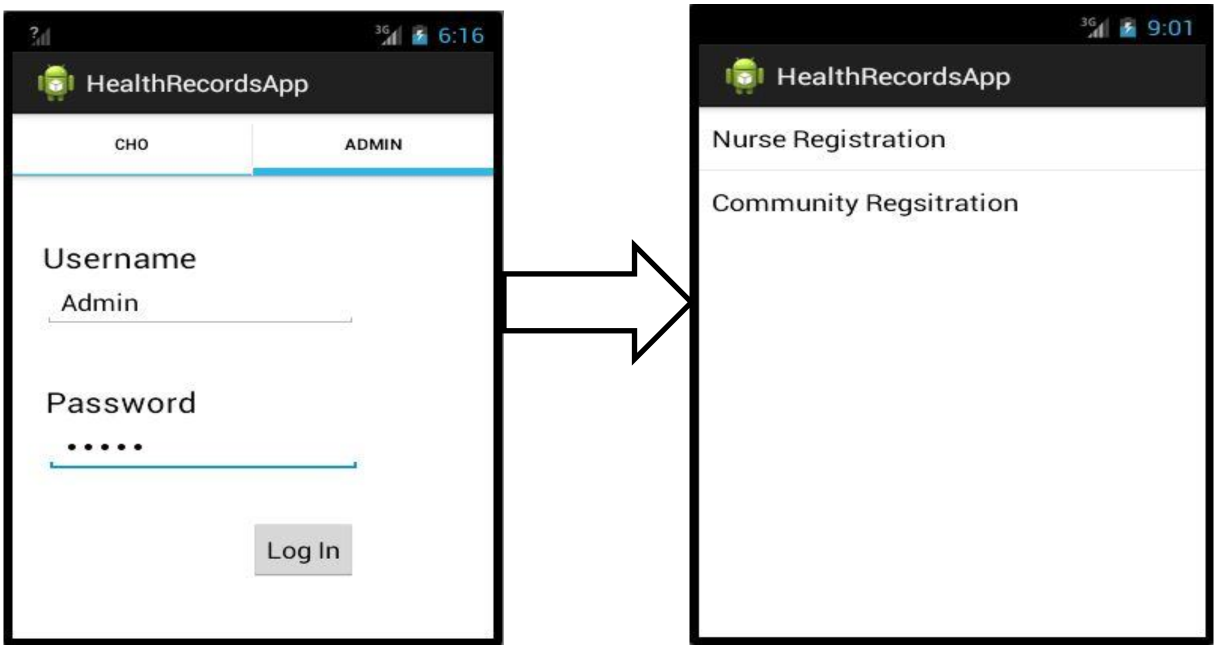


Figure 5 Use case diagram for system administrator

CHAPTER 3: IMPLEMENTATION

3.1 IMPLEMENTATION OVERVIEW

The main view of the application involves a log in screen where CHOs can log in to access the application or a system administrator can log in to the system. When the application is deployed initially, the details of a system administrator are logged into the system. This will ensure that when the application starts, an administrator can have access to the system and enter the details of the CHOs and community details to be saved in the system. Once this information is entered, the CHO can now log into the system and access various options available for a CHO. The log in activity for the system takes the details entered in the form and authenticates the user via the local database which grants access to the CHO. This will ensure that even without a network connection, the application is accessible to the user. When a CHO logs in, his or her log in details including the name and community he or she belongs to will be set to the various fields corresponding to such information in the forms in the application. This will reduce the amount of data entry a CHO has to make to ensure efficient use of the system. The diagrams below illustrate the initial activities in the system.



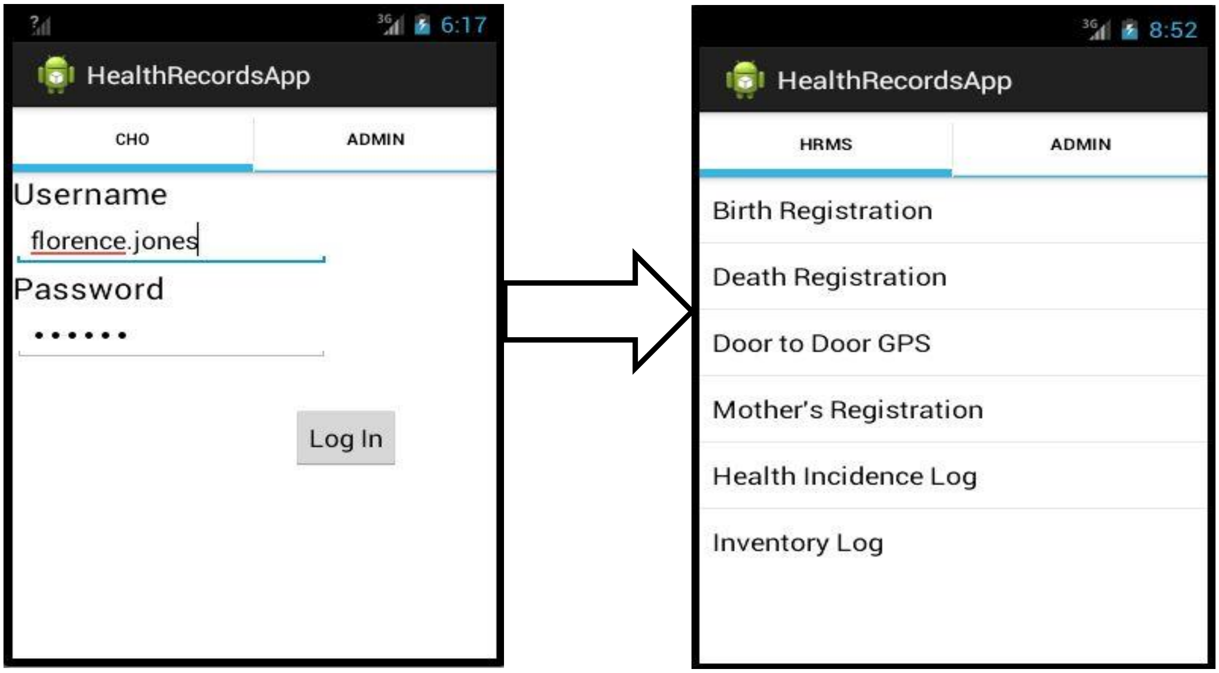
Username

Admin

Password

•••••

Figure 6 UI for administrator sign in and menu view



Username

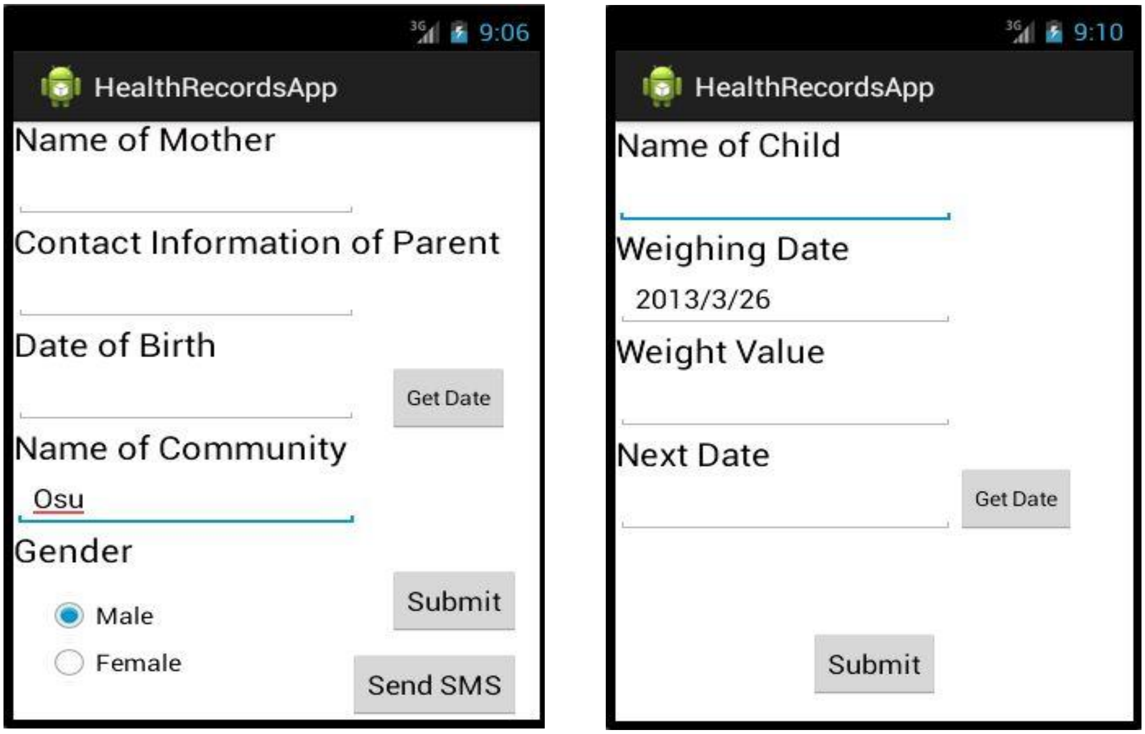
florence.jones

Password

••••••

Figure 7 UI for CHO sign in and menu view

When the CHO clicks on the birth registration menu item, he or she will have some options which include whether to enter information or enter data on weighing. Another possibility in this setting is to send SMS reminders. These reminders are sent via an SMS gateway to parents or guardians of children whose next weighing date recorded in the database is the same as the current date in the system. The message serves as a reminder to attend the weighing session on the said date. When a CHO has finished registering an infant, he or she has the option of sending an SMS confirmation to the contact number provided by parents or guardians of the child indicating the details of the registration. Figure 8 below shows the forms used for entering data on birth registration and weighing.



Name of Mother

Contact Information of Parent

Date of Birth

Name of Community
Osu

Gender
● Male
○ Female

Name of Child

Weighing Date
2013/3/26

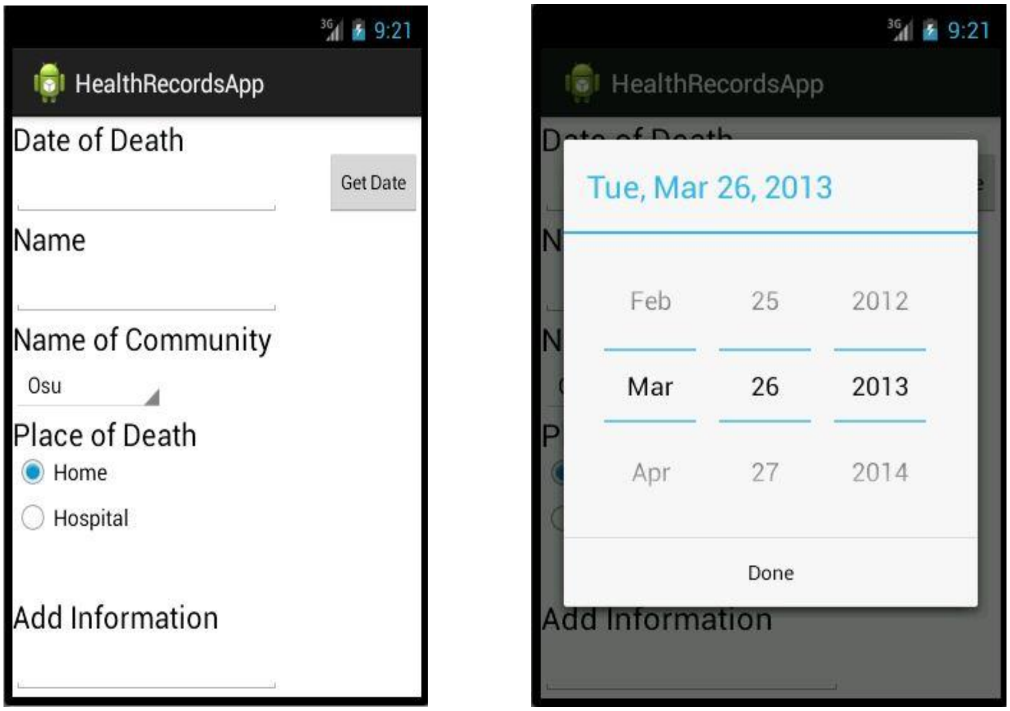
Weight Value

Next Date

Figure 8 UI for birth registration and weighing

The name of community set in the field on the birth registration form corresponds to the community the CHO currently logged in belongs to. The weighing date on the weighing form also corresponds to the current system date. This is to reduce the amount of data entry thereby facilitating the data entry process with reduced effort.

The next component in HRMS is the Death Registration component. This section collects data on dead persons in the community. Some data collected include the date of death as well as the place of death. The form provides a drop down menu of the community names registered in the system. Figure 9 below shows the death registration form.



Date of Death

Name

Name of Community

Osu

Place of Death

● Home

○ Hospital

Add Information

Tue, Mar 26, 2013

Feb 25 2012

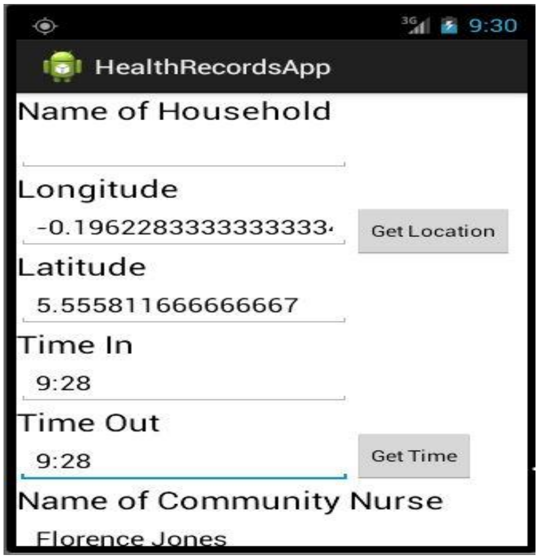
Mar 26 2013

Apr 27 2014

Figure 9 UI for death registration

When the button get date is clicked on the form, it provides a date picker where the CHO can enter the date on which a particular individual died.

The door to door GPS option provides a form where data is entered on various elements such as the name of the household being visited, time spent at the location which is done by recording the time in and time out of the location. The name of the CHO recorded corresponds to the name that is logged into the current session. As the name suggests, GPS coordinates are also logged at the location. A button is clicked which triggers a service in the background. This service then retrieves and returns the GPS coordinates of the location. The figure below indicates the flow during this action.



HealthRecordsApp

Name of Household

Longitude

-0.1962283333333333

Latitude

5.555811666666667

Time In

9:28

Time Out

9:28

Name of Community Nurse

Florence Jones

Figure 10 UI for door to door GPS Log

Figure 11 indicates the service used by application to retrieve GPS coordinates. This application uses the Location Manager which accesses the GPS system on the mobile device and through this. The GPS coordinates are logged and saved into the system.

```
public Location getLocation() {
              try {
                  locationManager = (LocationManager) mContext
                          .getSystemService(LOCATION_SERVICE);

                  // getting GPS status
                  isGPSEnabled = locationManager
                          .isProviderEnabled(LocationManager.GPS_PROVIDER);

                  // getting network status
                  isNetworkEnabled = locationManager

.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

                  if (!isGPSEnabled && !isNetworkEnabled) {
                      // no network provider is enabled
                  } else {
                      this.canGetLocation = true;
                      // First get location from Network Provider
                      if (isNetworkEnabled) {
                          locationManager.requestLocationUpdates(

                                  LocationManager.NETWORK_PROVIDER,
                                  MIN_TIME_BW_UPDATES,
                                  MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                          Log.d("Network", "Network");
                          if (locationManager != null) {
                              location = locationManager

.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                              if (location != null) {
                                  latitude = location.getLatitude();
                                  longitude = location.getLongitude();
                              }
                          }
                      }
                      // if GPS Enabled get lat/long using GPS Services
                      if (isGPSEnabled) {
                          if (location == null) {
                              locationManager.requestLocationUpdates(
                                      LocationManager.GPS_PROVIDER,
                                      MIN_TIME_BW_UPDATES,
                                      MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                              Log.d("GPS Enabled", "GPS Enabled");
                              if (locationManager != null) {
                                  location = locationManager

.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                                  if (location != null) {
                                      latitude = location.getLatitude();
                                      longitude = location.getLongitude();
                                  }
                              }
                          }
                      }
                  }
```

Figure 11 Snippet of code showing how GPS coordinates are logged into the system

The system also logs the system time as the time in at a particular location. The time that is logged is fixed at the time that particular option was launched. The following figure shows the snippet of code that displays the system time in the application:

```
final Calendar c = Calendar.getInstance();
            year = c.get(Calendar.YEAR);
            month = (c.get(Calendar.MONTH)+1);
            day = c.get(Calendar.DAY_OF_MONTH);
            hour=c.get(Calendar.HOUR_OF_DAY);
            minute=c.get(Calendar.MINUTE);
            second=c.get(Calendar.SECOND);
            timein.setText(hour+":"+minute+":"+second);
```

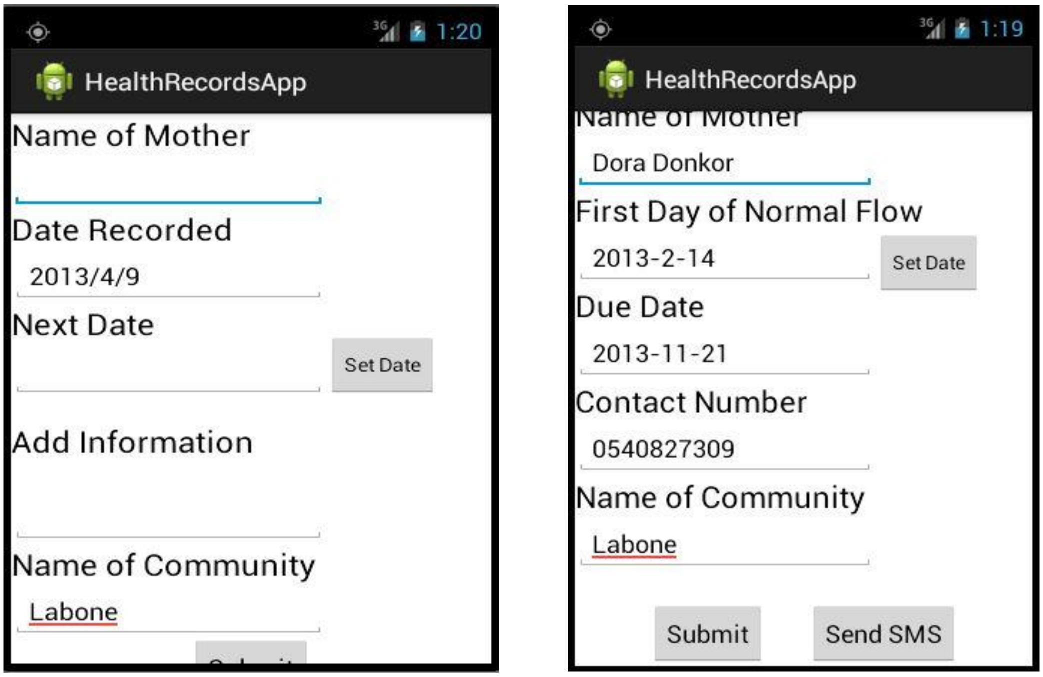
Figure 12 Snippet of code showing System time in log

When the CHO is about to leave a particular location he or she has to log a time out. When the application is launched, the time out field starts a timer which keeps timing until the submit button is clicked. The figure below shows the thread that does this timing:

```
public void doWork() {
          runOnUiThread(new Runnable() {
              public void run() {
                  try{
                          Date dt = new Date();
                          int hours = dt.~~getHours~~();
                          int minutes = dt.~~getMinutes~~();
                          int seconds = dt.~~getSeconds~~();
                          String curTime = hours + ":" + minutes + ":" +
seconds;
                          timeout.setText(curTime);
                  }catch (Exception e) {}
              }
          });
      }
```

Figure 13 Snippet of code showing time out log

Choosing the mother's registration option opens a menu where a CHO can enter information for registration purposes; enter information based on antenatal care checkups and also to send SMS reminders. When a CHO has to register an expectant mother, he or she will choose that option which opens a form. The due date in the form is calculated based on the first day of flow entered in the form and 280 days are added to that date to produce the due date. During the antenatal care data entry process, a CHO will enter the next date for an antenatal checkup and with this, SMS reminders will be sent to remind expectant mothers to come for their antenatal checkup. The figure below shows the data entry forms for mother's registration and antenatal care.


HealthRecordsApp

Name of Mother

Date Recorded
2013/4/9

Next Date

Add Information

Name of Community
Labone

HealthRecordsApp

Name of Mother
Dora Donkor

First Day of Normal Flow
2013-2-14

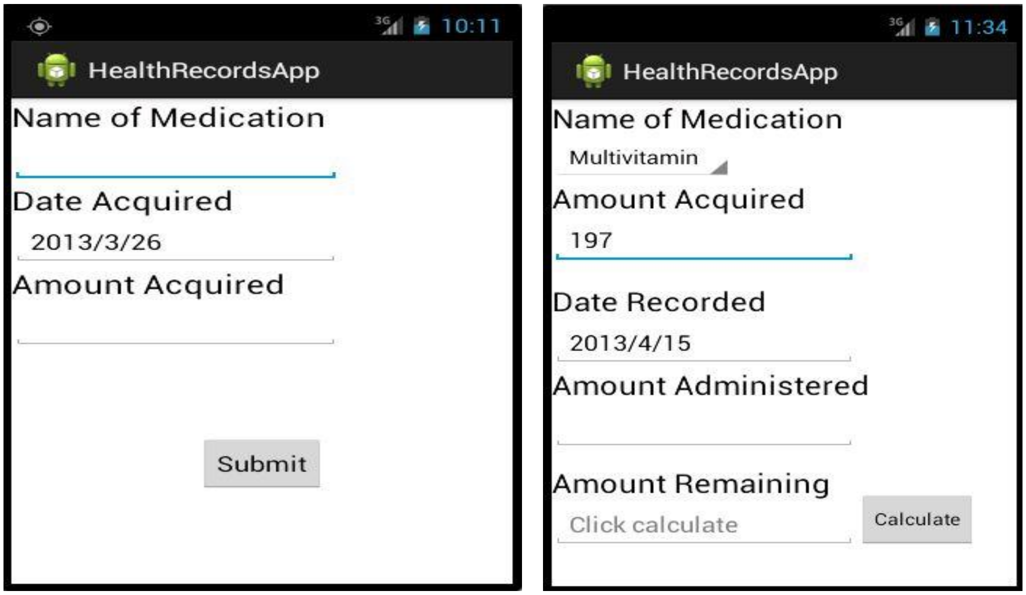
Due Date
2013-11-21

Contact Number
0540827309

Name of Community
Labone

Figure 14 UI for mother's registration and antenatal care log

The inventory log system provides options such as medication log, inventory log and the option to view log reports. Medication log provides a form for data to be entered on various medicines received at the community health center. The medication system adopts the insert or update feature; if the particular medication already exists in the system then the field in the database is updated with the sum of the incoming entry and the amount acquired value that exists in the database. If the entry however, does not exist, then the entry is inserted. This is to avoid duplicates in the system and to ensure consistency. The inventory log system then provides an option to view the various medicines registered in the system and the amount that is registered in the system. When an option is selected in the dropdown menu, the amount entered in the system corresponding to that selected item is set to the amount acquired field.



HealthRecordsApp

Name of Medication

Date Acquired
2013/3/26

Amount Acquired

HealthRecordsApp

Name of Medication
Multivitamin

Amount Acquired
197

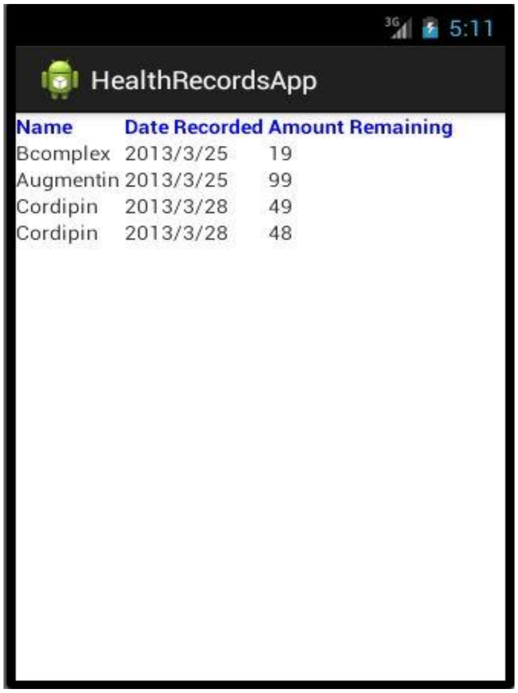
Date Recorded
2013/4/15

Amount Administered

Amount Remaining
Click calculate

Figure 15 UI for medication log form and inventory log form

A calculate button is provided to log inventory into the system. The calculate button takes the value from the amount administered field and subtracts the amount from the amount acquired. When this calculation is done, the CHO has to update the medication log table to ensure consistency by changing the amount remaining in the system. This will ensure that the next time the CHO accesses the inventory log system; the amount will be what is left. This component also provides a table report of all the inventory log activity in the system. This way, the CHO can keep track of the activity in the system.



Name	Date Recorded	Amount Remaining
Bcomplex	2013/3/25	19
Augmentin	2013/3/25	99
Cordipin	2013/3/28	49
Cordipin	2013/3/28	48

Figure 16 UI for inventory log report

SMSs are sent from this application for various purposes. They are sent as reminders to expectant mothers and parents or guardians of infants registered in the system. They are also sent as confirmations with registration details to respective individuals registered in the system. SMSs are sent via an SMS gateway provided by SMS GH. SMS GH requires authentication to connect to the API. This is done through a username and password which is set during the registration for an account with mytxtbox, mytxtbox pro and other packages offered by SMS GH. The current account being used to connect to the SMS GH API is mytxtbox. The figure below shows a snippet of code indicating how the server side script in PHP communicates with the SMS gateway.

```
$user = "username";
$password = "password";
$api_id = "123456";

switch($cmd){
    case 1:
                $name_of_child=$_GET['val1'];
                $name_of_father=$_GET['val2'];
                $name_of_mother=$_GET['val3'];
                $community=$_GET['val4'];
                $number=$_GET['val5'];
$text =urlencode("Your child has been registered with the following details:
"."\n"."Name: ".$name_of_child."\n"."Father: ".$name_of_father."\n"."Mother:
".$name_of_mother."\n"."Community: ".$community);
$to=$_GET['val5'];
$url="http://www.mytxtbox.com/smsghapi.ashx/sendmsg?api_id=123456&user={$user
}&password={$password}&to={$to}&text={$text}&from=florence.jones";
$ret = file($url);
$send = split(":",$ret[0]);
if ($send[0] == "ID"){

    echo "Message sent!";
}else{
echo "Message sending failed";
}
break;
```

Figure 17 Snippet of code showing how SMS through SMS GH gateway

The web application for district officers provides an interface for District Officers to log into the application. Users are registered into the system by a system administrator. When a district officer visits the web page, he or she will see the page below:

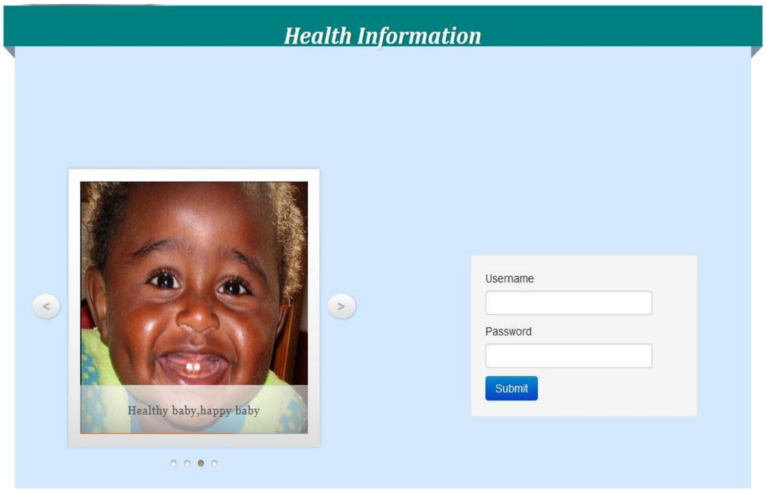
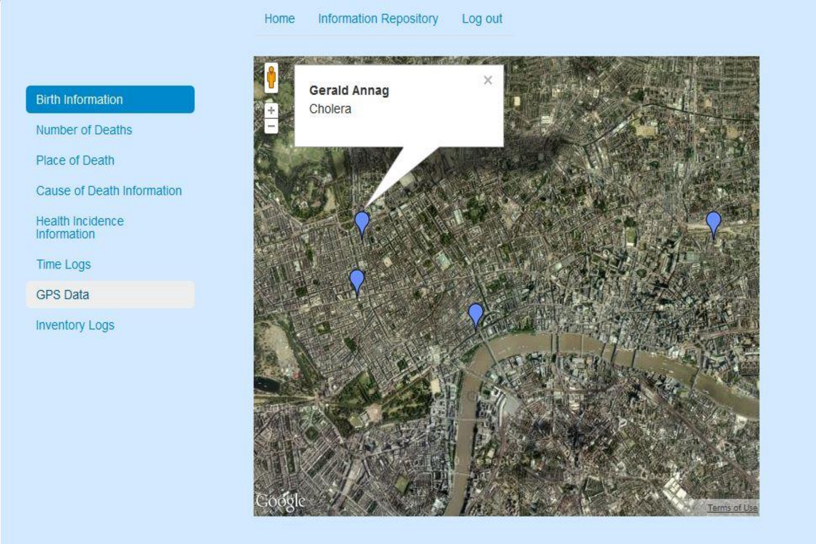


Figure 18 Home page for District Officers web application

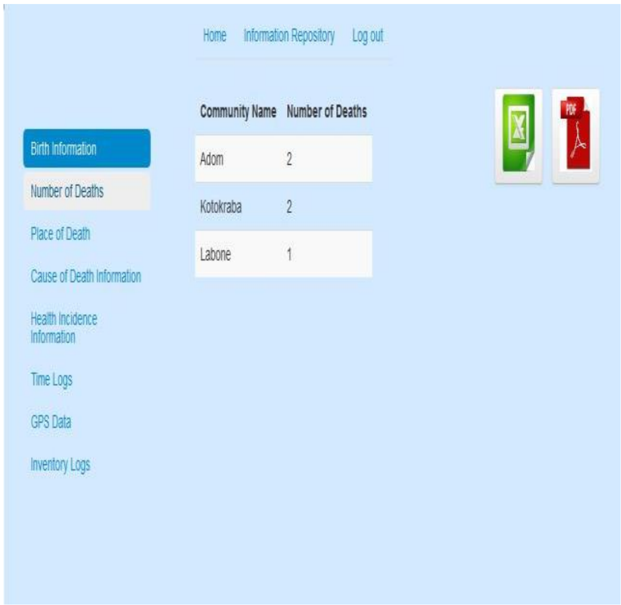
On this interface, a registered district officer will log into the application and have access to the main page where data needed can be accessed. Data that can be retrieved from the web application include GPS data. The GPS data comprises of GPS coordinates that have been plotted on the map with a description showing the name of the household associated to the GPS coordinate and the health issue that was recorded at that location. This information is retrieved from the database through the information that was entered by CHOs.



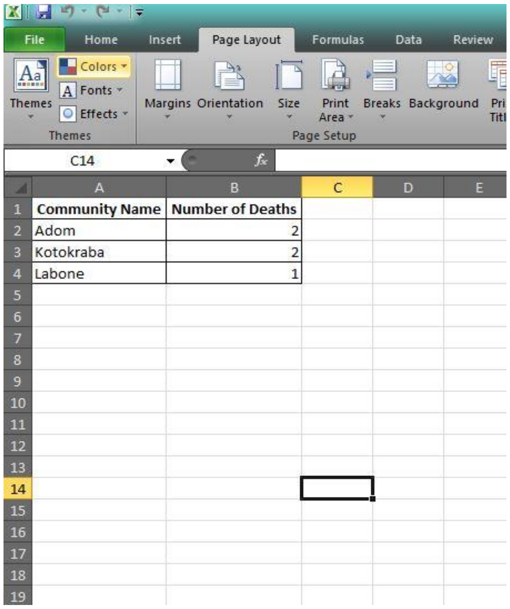
Gerald Annag
Cholera

Figure 19 Google maps with GPS coordinates plotted showing description

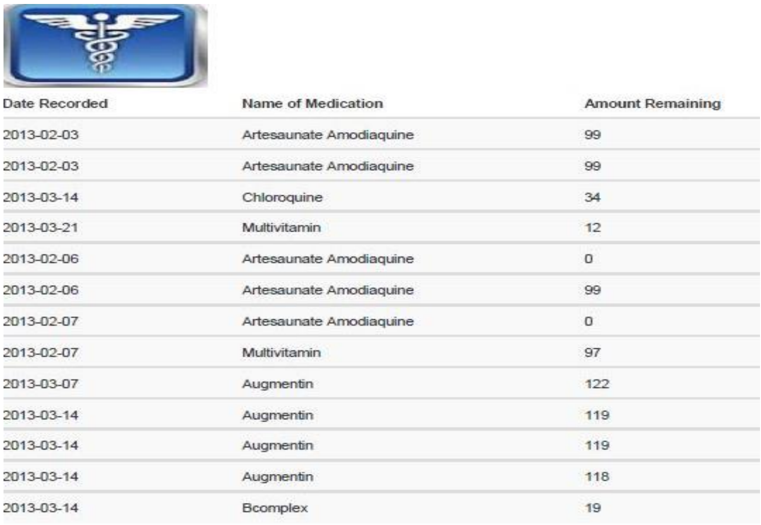
District officers have the option of downloading PDF or excel formats of reports generated on the web page. The figure below indicates a report on the number of deaths recorded in various communities, an excel report generated from this page and a PDF report generated on data provided under inventory logs on the web application.



Community Name	Number of Deaths
Adom	2
Kotokraba	2
Labone	1



Community Name	Number of Deaths
Adom	2
Kotokraba	2
Labone	1



Date Recorded	Name of Medication	Amount Remaining
2013-02-03	Artesaunate Amodiaquine	99
2013-02-03	Artesaunate Amodiaquine	99
2013-03-14	Chloroquine	34
2013-03-21	Multivitamin	12
2013-02-06	Artesaunate Amodiaquine	0
2013-02-06	Artesaunate Amodiaquine	99
2013-02-07	Artesaunate Amodiaquine	0
2013-02-07	Multivitamin	97
2013-03-07	Augmentin	122
2013-03-14	Augmentin	119
2013-03-14	Augmentin	119
2013-03-14	Augmentin	118
2013-03-14	Bcomplex	19

Figure 20 Report on web application and Excel and PDF reports generated

3.2 TOOLS, PLATFORMS AND FRAMEWORKS

In developing the Health records management system, a number of tools, platforms and frameworks were used. These are explained as follows:

3.2.1 ECLIPSE

In developing the mobile application for this system, the Eclipse Integrated Development Environment (IDE) was used. This tool was chosen because it is an open source IDE that provides plugins to enable development on the Android platform. The IDE facilitates programming by providing auto-complete and auto-insert coding options. The IDE also provides a plugin for creating an emulator which facilitates testing of the application. This IDE was also chosen because it provides an easy setup for android development. Once this setup is complete, development and deployment of the application is less stressful. The IDE also provides information on errors making it easier for errors to be detected and fixed to ensure smooth application development.

3.2.2 ANDROID

Android is a mobile development platform that has evolved over the years. This platform provides various features that are fascinating and essentially important to the development of this system. Its GPS as well as its useful and interesting widgets provides an application that is physically satisfying as well as internally useful and powerful. This platform comes with an inbuilt local database known as SQLite. This feature is useful to the application considering the conditions that this application will be used on. With this feature, the application can store some information such as log in details and content that has to be loaded dynamically in the application. The former will ensure that the application is usable at all times even if there is no connectivity and the latter will reduce the amount of bandwidth consumed when loading and using this application.

3.2.3 PHP, MySQL, HTML5, CSS3

In developing the web application for the HRMS, the interface for the web application was designed using HTML5 and CSS3. These tools were chosen due to their ability to produce a stunning user interface with useful features for the purposes of this application. The features provided by these tools ensure a finished product that is physically appealing and internally extraordinary. PHP provided an ability to embed server side scripting in an HTML page to produce a dynamic web application. This allowed for the production of a web application that interacts with MySQL database to produce results needed for the application to run effectively. PHP was also used to provide server side scripts to the mobile application to ensure that data submitted is saved in the remote database.

3.2.4 JSON AND JQUERY

The results retrieved from the MySQL database were presented to the data layer in the system architecture using the JavaScript Object Notation (JSON). This is a lightweight data-interchange format and is very easy for humans to read and write. It is also very easy for machines to generate this format and parse it [4]. JQuery which is a JavaScript library was used to parse the JSON objects in order to produce user readable data in tables.

CHAPTER 4: TESTING

4.1 TESTING OVERVIEW

In testing the HRMS, various features were taken into consideration. These features correspond to the functional and non-functional requirements of the system. Tests conducted include:

4.1.1 DATA STORAGE IN DATABASE

One of the major features of this application is its ability to store data in the databases both remotely and locally. This android application was therefore tested for the following:

- Ability to save into SQLite database
- Ability to save into MySQL database

RESULTS

The android application passed these tests as the values entered in the various forms were saved into the local and remote database simultaneously.

4.1.2 DATA RETRIEVAL

The web application component of this application is supposed to make it easy for District officers to be able to retrieve data that is submitted by CHOs. CHOs must also be able to view a tabular report on the inventory table to make it easy for to manage their inventory in their mobile application. The application was therefore tested for these features.

RESULTS

The application passed this testing phase. All the data that was available and needed to be retrieved were accessible. These data were physically available on the web application and district officers can view these data. The android application also provides a tabular report on inventory in the system.

4.1.3 REPORT GENERATION

The web application must be able to generate reports for download by the user. The features that were tested include:

- Ability to generate PDF report
- Ability to generate Excel report

RESULTS

The web application was able to generate PDF and excel reports on data where necessary. This fulfills the functional requirement of the system being able to produce or generate PDF and Excel reports on the data being generated in the application.

4.1.4 SMS FUNCTIONALITY

Another requirement of the application is its ability to send messages under various conditions specified. The features tested under this include:

- Ability to send SMS through SMS Gateway (SMS GH)
- Ability to send SMSs to multiple numbers through above mentioned gateway

- Ability to meet conditions under which to send SMS messages to respective individuals.

RESULT

The application passed this test because the text messages were sent through the gateway and were received during the testing period. Some conditions have to be met when sending an SMS which include the ability to compare a particular date in the database to the current date in the system. Based on this date, an SMS is sent to expectant mothers as well as parents or guardians of infants registered in the system to remember to make it for their antenatal appointment and weighing appointment respectively. Below is a sample of the SMS sent to expectant mothers:

"Dear Sarah Jones,

Remember to come for your antenatal appointment today.

Thank you"

The application must also send SMS confirmations with details of the registration to parents as well as expectant mothers. Below is a sample confirmation text message sent to a parent concerning his or her child's registration details:

"*Your child has been registered with the following details:*

Name: Sandra Bullock

Father: Terry Bullock

Mother: Torry Davids

Community: Osu"

4.1.5 COMPATIBILITY TESTING

This application was tested for its ability to run smoothly on two versions of the android platform. The versions that that the application was tested on include the FROYO (2.2) and the Ice cream sandwich (4.0) platform. The following features were tested:

- Ability to display widgets effectively

RESULTS

This phase of testing encountered a problem as some widgets implemented in the application such as the date and time pickers were not displayed in the application when it was tested on the FROYO platform. On the ice cream sandwich platform however, these widgets displayed very well. Widgets such as the dropdown menus and alert dialog boxes however were displayed in the application in both platforms that were tested.

4.1.6 GPS AND GOOGLE MAPS FUNCTIONALITY

The application was tested for GPS functionality in the mobile application and Google maps functionality in the web application. The following elements were tested:

- Ability to retrieve and display longitude and latitude coordinates
- Ability to plot GPS coordinates on Google maps
- Ability to show a description on each GPS coordinate

RESULTS

The application passed this stage of testing as GPS coordinates were retrieved and successfully stored in the database from the mobile application. GPS coordinates stored in the database were retrieved from the database and these coordinates were displayed successfully on the Google maps embedded in the web application with a description comprising of the household name and the specific health issue present at that location.

CHAPTER 5: CHALLENGES, FUTURE WORK, CONCLUSION

5.1 CHALLENGES

5.1.1 ANDROID VERSION CHALLENGES

Some challenges were faced while developing on the android platform. Some of these challenges include issues with the different versions of android. Android has a lot of versions that have developed on it platform. Many users are still on the lower versions of android. In developing application, the minimum SDK used was 8 which is the version 2.2 (FROYO) and the maximum SDK was 17 which is version 4.2 (JELLY BEAN). While testing the application most of the features were downgraded where necessary but some widgets such as the date picker and the table layout were not visible in the lower version FROYO. Anytime these features were accessed, the application would forcefully close.

5.1.2 ANDROID SQLite Challenges

The SQLite database on the android device is fairly easy to implement but not entirely intuitive. This made it time consuming to understand the processes and to eventually implement this feature in the application. The database can be implemented in two ways which involves using a content provider or using a database handler. The content provider provides the extra feature that supports the store and forward feature. This will allow the application to function in a reliable way so that in the case of no network connectivity, the application can store the data in the local database and this can be forwarded when there is an internet connection. Understanding the content provider and how this can be linked to the remote database was a challenge and as a result this feature was not implemented completely.

5.1.2 JQUERY CHALLENGES

Challenges were faced while developing the web application using JQuery as the JSON parser. Understanding the framework of JQuery and eventually using the tool was a major challenge. The syntax was not complicated but was not entirely intuitive. As a result some parts of the syntax were very difficult to understand in order to be tailored to suit the purpose for which it was being used. After parsing the JSON object successfully, another difficulty faced was knowing how to link the result from JavaScript to the HTML DOM elements in order to be able to successfully display the parsed object. In displaying the result in an HTML table, the results of the display were distorted with the table header displaying for every row of data that was fetched.

5.2 FUTURE WORK

In the event of limited or no connectivity, the application must be able to store this data which will later be forwarded to the remote database when there is connectivity. This feature was not implemented in the application and therefore there is the tendency of data being lost in transit. The data collected is however, stored in the remote database and this can be a point of access in order to create back up for the data. The remote database can then be synchronized with the remote database in order ensure that the data being provided by the application is consistent and reliable.

This application seeks to provide data for decision making purposes and for this reason more research can be done to find out what kind of information can be recorded in order to provide more material for decision making.

5.3 CONCLUSION

The main aim of the Health Records Management System was to provide the management of a community health center with a tool that will assist in effective data recording and decision making in order to ensure quality health care delivery. This system has ensured this by providing a mobile application that ensures effective data entry. The system also provides a web application that allows district officers to view the data entered and also provides some useful tools such as Excel and PDF reports to facilitate effective data analysis. This will go a long way in benefitting the recipients of health care services in the rural areas to ensure that effective actions are taken to give quality health care to individuals. Measures can be put in place to prevent epidemics based on past data and actions can be taken to assuage current epidemics.

Bibliography

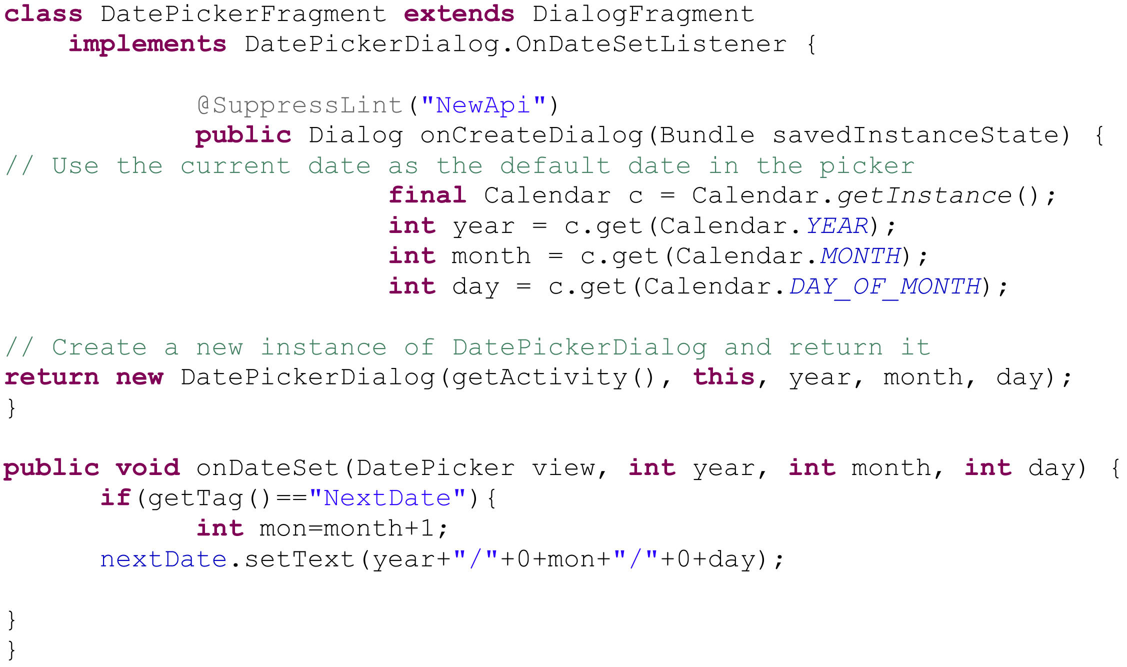
[1] The Grameen Foundation, "Mobile Technology For Community Health in Ghana," September 2012. [Online]. Available: http://www.grameenfoundation.org. [Accessed 11 March 2013].

[2] P. N. Mechael, "MoTECH: mHealth Ethnography Report," August 2009. [Online]. Available: http://www.grameenfoundation.org. [Accessed 11 March 2013].

[3] F. K. Nyonator, K. Awoonor-Williams, J. F. Phillips, T. C. Jones and R. A. Miller, "The Ghana Community-based Health Planning Services Initiative for scaling up service delievery innovation," *Health and Policy Planning,* pp. 25-34, 2005.

[4] JSON, "Introducing JSON," [Online]. Available: http://www.json.org/. [Accessed 30 March 2013].

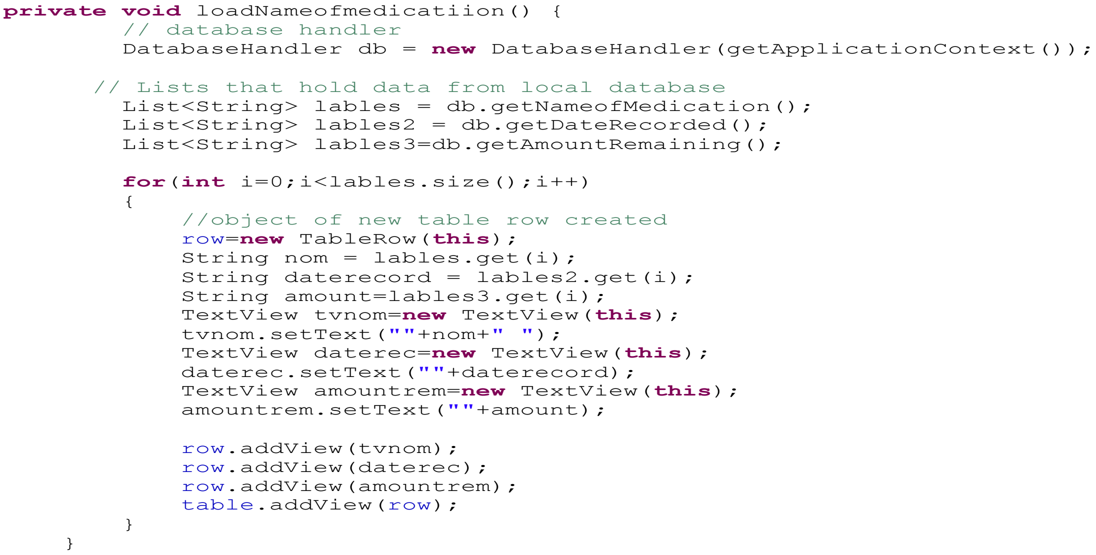
APPENDIX



```
class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {

            @SuppressLint("NewApi")
            public Dialog onCreateDialog(Bundle savedInstanceState) {
// Use the current date as the default date in the picker
                        final Calendar c = Calendar.getInstance();
                        int year = c.get(Calendar.YEAR);
                        int month = c.get(Calendar.MONTH);
                        int day = c.get(Calendar.DAY_OF_MONTH);

// Create a new instance of DatePickerDialog and return it
return new DatePickerDialog(getActivity(), this, year, month, day);
}

public void onDateSet(DatePicker view, int year, int month, int day) {
      if(getTag()=="NextDate"){
            int mon=month+1;
      nextDate.setText(year+"/"+0+mon+"/"+0+day);

}
}
```

Figure 21 Snippet of code showing Date Picker Dialog

Fig 1.1 Snippet of code showing Date Picker Dialog



```
private void loadNameofmedicatiion() {
        // database handler
        DatabaseHandler db = new DatabaseHandler(getApplicationContext());

      // Lists that hold data from local database
        List<String> lables = db.getNameofMedication();
        List<String> lables2 = db.getDateRecorded();
        List<String> lables3=db.getAmountRemaining();

        for(int i=0;i<lables.size();i++)
        {
            //object of new table row created
            row=new TableRow(this);
            String nom = lables.get(i);
            String daterecord = lables2.get(i);
            String amount=lables3.get(i);
            TextView tvnom=new TextView(this);
            tvnom.setText(""+nom+"  ");
            TextView daterec=new TextView(this);
            daterec.setText(""+daterecord);
            TextView amountrem=new TextView(this);
            amountrem.setText(""+amount);

            row.addView(tvnom);
            row.addView(daterec);
            row.addView(amountrem);
            table.addView(row);
        }
    }
```

Figure 22 Snippet of code showing how inventory table report is generated

```
$dompdf = new DOMPDF();
      $html = "
      <head>
    <link rel='stylesheet' type='text/css' href='healthweb.css'/>
    </head>
      <img src='img/healthicond.jpg' width='150' height='100'>
        <table class='table table-striped'>
        <thead>
        <tr>
        <th>Name of CHO</th>
        <th>Time In</th>
        <th>Time Out</th>
        <th>Time Difference</th>
        </tr>
        </thead>";
      while($row)
        {
        $timein=$row['time_in'];
        $timeout=$row['time_out'];
        $firstTime=strtotime($timein);
        $secondTime=strtotime($timeout);
        $timediff=$secondTime-$firstTime;
        $timecorrect=$timediff/60;
        $html .="<tr>
        <td>" . $row['nurse_name'] . "</td><td>" . $row['time_in'] .
"</td><td>". $row['time_out'] ."</td><td>".$timecorrect."</td></tr>";
          $row=$logObj->fetch();
        }
        $date=date("Y/m/d");
        $html .= "</table>";
        $dompdf->load_html($html);
        $dompdf->render();
        $dompdf->stream("Time Logs Report".$date.".pdf");
```

Figure 23 Snippet of code indicating how PDF reports are generated

```
$logObj= new healthRecordsWebPage();
     $content=$logObj->get_time();
     $head="<table border=1>
            <thead>
                <tr>
        <th>Name of CHO</th>
        <th>Time in</th>
        <th>Time out</th>
        <th>Time Difference</th></tr></thead></table>";
        echo $head;
      while( $row=$logObj->fetch())
        {
        $timein=$row['time_in'];
        $timeout=$row['time_out'];
        $firstTime=strtotime($timein);
        $secondTime=strtotime($timeout);
        $timediff=$secondTime-$firstTime;
        $timecorrect=$timediff/60;
        $file="TimeLogReport.xls";
$test="<table
border='1'><tr><td>{$row['nurse_name']}</td><td>{$row['time_in']}</td><td>{$r
ow['time_out']}</td><td>{$timecorrect}</td></tr></table>";
header("Content-type: application/vnd.ms-excel");
header("Content-Disposition: attachment; filename=$file");
echo $test;
```

Figure 24 Snippet of code indicating how Excel reports are generated

```
var icon = new
google.maps.MarkerImage("http://maps.google.com/mapfiles/ms/micons/blue.png",
 new google.maps.Size(32, 32), new google.maps.Point(0, 0),
 new google.maps.Point(16, 32));
 var center = null;
 var map = null;
 var currentPopup;
 var bounds = new google.maps.LatLngBounds();
  function addMarker(lat, lng, info) {
 var pt = new google.maps.LatLng(lat, lng);
 bounds.extend(pt);
 var marker = new google.maps.Marker({
 position: pt,
 icon: icon,
 map: map
 });
 var popup = new google.maps.InfoWindow({
 content: info,
 maxWidth: 300
 });
 google.maps.event.addListener(marker, "click", function() {
 if (currentPopup != null) {
 currentPopup.close();
 currentPopup = null;
 }
 popup.open(map, marker);
 currentPopup = popup;
 });
 google.maps.event.addListener(popup, "closeclick", function() {
 map.panTo(center);
 currentPopup = null;
 });
 }
 function initMap() {
 document.getElementById('map').style.visibility='visible';
    document.getElementById('time').style.visibility='hidden';
    document.getElementById('placeofdeath').style.visibility='hidden';
    document.getElementById('ajaxresultinventory').style.visibility='hidden';
    document.getElementById('birthinformation').style.visibility='hidden';
    document.getElementById('deathinformation').style.visibility='hidden';
    document.getElementById('causeofdeath').style.visibility='hidden';
 map = new google.maps.Map(document.getElementById("map"), {
 center: new google.maps.LatLng(0, 0),
 zoom: 14,
 mapTypeId: google.maps.MapTypeId.SATELLITE,
 mapTypeControl: false,
 mapTypeControlOptions: {
 style: google.maps.MapTypeControlStyle.HORIZONTAL_BAR
 },
 navigationControl: true,
 navigationControlOptions: {
 style: google.maps.NavigationControlStyle.SMALL
 }
```

Figure 25 Snippet of code to generate Google Maps

```
<script src="js/jquery-1.9.1.js"></script>
<script src="js/jquery.loadJSON.js"></script>
<script>
$(document).ready(function(){


    var $inventory;
    $inventory =
$.get("http://localhost/Applied/HealthWeb/healthwebActions.php?cmd=1",
function(data) {
    $inventory=JSON.parse(data);
    $("#ajaxresultinventory").loadJSON($inventory);
    });
</script>


<table class="table table-striped">
    <thead>
    <th>Name of CHO</th>
    <th>Time In</th>
    <th>Time Out</th>
    <th>Time Difference</th>
    <thead>
    <tr id="details">
    <td class="nurse_name"></td>
    <td class="time_in"></td>
    <td class="time_out"></td>
    <td class="time_difference"></td>
    </tr>

    </table>
```

Figure 26 JQuery and JSON