



ASHESI UNIVERSITY COLLEGE

CHILD WELFARE MODULE FOR EXISTING MHEALTH APPLICATION, YARESA

UNDERGRADUATE APPLIED PROJECT

B.Sc. Computer Science

David Tanoe-Kpanyi Tandoh

2016

ASHESI UNIVERSITY COLLEGE

**Child Welfare Module for Existing mHealth
application, Yaresa**

UNDERGRADUATE APPLIED PROJECT

Applied Project Report submitted to the Department of Computer Science,
Ashesi University College in partial fulfillment of the requirements for the
award of Bachelor of Science degree in Computer Science.

David Tanoe-Kpanyi Tandoh

May 2016

Declaration

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

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

Acknowledgement

I would like to thank God for giving me the strength and granting me the grace to go through the duration of this Applied project. I would also like extend my gratitude to my supervisor, Mr. Aelaf Dafla, for the guidance and support he gave during the entire period of this project.

Finally, I would also like to thank all the participants and interviewees of this project, especially the Community Health Workers of the Berekuso Health Center who played a huge role in helping with the gathering of the requirements of this project. It would have been impossible to accomplish this project without them.

Abstract

With Ghana striving to reach its Millennium Development Goals, the government has employed various tactics to improve healthcare delivery in Ghana. One of these strategies involves deploying Community Health Workers to rural areas to provide residents of such areas with good healthcare. As a result, various Community Health Centers have been set up in rural areas, and Community Health Workers have to attend to the healthcare needs of these communities. In recent past, mobile technology has been used to improve the efficiency of Community Health Workers in these areas. As such, a group of Ashesi students and faculty developed an mHealth application to also help Community Health Workers in various ways. This project presents a Child Welfare Module to be integrated with the existing mHealth application, Yaresa.

Table of Contents

DECLARATION	III
ACKNOWLEDGEMENT.....	IV
ABSTRACT.....	V
DEFINITIONS AND ACRONYMS.....	IX
CHAPTER 1: INTRODUCTION	1
1.1 INTRODUCTION AND BACKGROUND.....	1
1.2 RELATED WORKS	2
CHAPTER 2: REQUIREMENTS AND DESIGN	4
2.1 FEASIBILITY	4
2.2 PRODUCT SCOPE, PERSPECTIVE AND FUNCTIONS	5
2.3 USER CLASSES.....	5
2.4 OPERATING ENVIRONMENT.....	6
2.5 ASSUMPTIONS AND DEPENDENCIES	6
2.6 PROCEDURE FOR REQUIREMENTS GATHERING	6
2.7 SCENARIOS	7
2.7.1 <i>Scenario A</i>	7
2.7.2 <i>Scenario B</i>	8
2.8 USE CASES	9
2.9 FUNCTIONAL REQUIREMENTS	10
2.10 OTHER REQUIREMENTS	11
2.11 NONFUNCTIONAL REQUIREMENTS	11
2.12 ACTIVITY DIAGRAM	12
2.13 ARCHITECTURE PATTERN	13
2.14 MODULES.....	14
2.15 DATA MODEL	15

2.16	MINIMAL CHILD WELFARE CLASS DIAGRAM	18
CHAPTER 3: IMPLEMENTATION		19
3.1	TECHNOLOGICAL PLATFORM	19
3.2	IMPLEMENTATION OF DATABASE.....	19
3.3	INTERFACES AND FUNCTIONALITIES IN DETAIL.....	20
3.4	DESIGN AND IMPLEMENTATION ISSUES	30
CHAPTER 4: TESTING AND RESULTS.....		31
4.1	DEVELOPMENT TESTING	31
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS.....		35
5.1	LIMITATIONS AND RECOMMENDATIONS	35
5.2	FUTURE WORK	36
REFERENCES.....		37
APPENDIX.....		38
DETAILED FUNCTIONAL REQUIREMENTS.....		38
MONTHLY NUTRITION FORM		43

Table of Figures

Figure 2.1: Use case diagram.....	10
Figure 2.2: Activity diagram.....	12
Figure 2.3: Model view architecture.....	14
Figure 2.4: Layered view of modules	15
Figure 2.5: Database architecture	17
Figure 2.6: Minimal child welfare class diagram	18
Figure 3.1. upgradeToVersion15() method	20
Figure 3.2. Adding a child	22
Figure 3.3: code snippet of use of record states.....	23
Figure 3.4: Visiting an existing child's profile	24
Figure 3.5: Nutrition page.....	25
Figure 3.6: Vitamin A supplementation page.....	26
Figure 3.7: Health issues page	27
Figure 3.8: Snippet of code for adding and discharging from CMAM	28
Figure 3.9: Snippet of code for adding and discharging from CMAM ctd.....	29
Figure 3.10: Child Welfare Reports.....	30
Figure 4.1: Unit test for adding a child.....	31

Definitions and acronyms

CHO – Community Health Worker

CHPS – Community-based Health and Planning Services

GHS – Ghana Health Service

CMAM - The Community-Based Management of Acute Malnutrition

MOTECH - Mobile Technology for Community Health

Chapter 1: Introduction

1.1 Introduction and Background

The Ghanaian society is faced with a lot of healthcare issues such as malaria, HIV/AIDS, child mortality and other endemic diseases, and fighting them is a big challenge. In addition, there are a lot of things hindering progress in solving these issues including low literacy rates, poverty and poor organization in the health system. As a developing country, Ghana needs to pay a lot of attention to its healthcare systems, as it has been proven that there is a correlation between good health care and economic development. Studies have shown that improving the health of a nation can directly affect economic growth positively because then people can be more effective in their workplaces (Collins, 2015).

One strategy the Ghanaian government has employed is to set up the Community-based Health and Planning Services system (Nyonator et al, 2005). Fifty percent of the Ghanaian population live in the rural areas and do not have access to good healthcare (Ghana Statistical Service, 2012). Accordingly, the Community-based Health and Planning Services system was set up to increase access to quality healthcare across the nation as the one major objective of the Ghana Health Service is to ensure access to good quality health services across the country. This way, trained personnel are put in rural areas to ensure that the people in such communities stay healthy. One major reason why the CHPS was established was to reduce the infant mortality rate in rural areas. In the early stages of the CHPS system, it was realized that children that were exposed to the program had 12% lower mortality, and infant mortality generally dropped (Russel, 2008).

To make the work of Community health workers easier, CS faculty and students developed a mobile application called Yaresa, which is now being used by 24 CHPS zones in two districts. Currently, the Community Health Officers use the app to manage digital

health records, track out Patient Department Cases, immunization and family planning. However, there are more modules that have been proposed for the app from various surveys that have been conducted. My main objective is to work on one of the proposed modules, which involves developing a module that focuses on child welfare, nutrition and surveillance. With the completion of this project, Community health Officers would be able to track the health of young children on the tablets that have been provided for them. This in the long run is aimed towards reducing infant mortality rate with active supervision. The 2013 Reproductive and Child Health Annual Report by GHS states that woman and child mortality account for a major proportion of all ill-health and deaths in Ghana. It then states most of these illnesses and deaths are preventable and I believe this can be achieved by increased and improved monitoring of child health in the various CHPS zones.

1.2 Related Works

Mobile health in Ghana has seen a rise since the introduction of the Sene PDA Project in 2004 in the Sene District. Some of these include the Millennium Villages m-health projects, the SMS for Life project and MOTECH. All of these projects seek to improve the efficiency of healthcare workforce and systems (Finkel, 2012). As a matter of fact, of these projects, the MOTECH project is related to the module to be implemented in that, it was developed to help community health nurses combat maternal illness and death. Interestingly, the module to be implemented complements this, and is supposed to help reduce child mortality by increasing surveillance of child welfare and nutrition. They may not be performing the same functions but are definitely performing related tasks since reproductive and child health are interconnected. The final product is supposed to put the various child healthcare records in one place, and also help CHOs with the filling of the

monthly nutrition report form. This form can be seen in the appendix. Details of the application's functions will be discussed in the next chapter.

Chapter 2: Requirements and Design

The main objective of this project is to come up with an m-health application module to be added to an existing app, Yaresa. What this module seeks to do is to help provide a platform to aid CHOs monitor child welfare and nutrition at their various health.

The purpose of this chapter is to present a detailed description of the child welfare and nutrition application module that will be added unto the existing Yaresa application. It will explain the various features of the system, the main functions of the system, the conditions under which it will operate and how it will operate.

This chapter also defines the functional and non functional requirements of the module. The rest of the chapter comprises of the overall description and the Requirement specification sections. The latter part of the chapter provides a high level overview and explains the whole architecture of the Child Welfare Module. This part of the chapter seeks to describe the elements fundamental to guiding the implementation of the Child welfare module, and understanding this project as a whole.

2.1 Feasibility

The Yaresa app is already in use and adding a new module would not cost so much considering the available technologies. It is critical to have the necessary skill and human resource for a software project to be successful. To make this project successful, the necessary skills to develop this application on Android platform was developed through research and practice.

2.2 Product scope, perspective and functions

At the end of this project, the application is supposed to affect CHOs in designated zones by aiding them in improving child healthcare in their designated zones, in the long run helping the country achieve its millennium goal of reducing child mortality rate.

To be specific, this module will allow CHOs to keep records of children they attend to and keep better track of their growth and nutrition. It will also help them in preparing their monthly reports by generating various reports such as the amount of Vitamin A administered to children within a specific time frame amongst others.

This module is expected to provide CHOs with a more convenient way of keeping records and details of the children they attend to. It will also provide them with the necessary information they need to make sure that a child's growth and development is on the right path. Most importantly, it will be used to help CHO's prepare their monthly reports quicker and faster, thereby having more time to spend on taking care of their patients. As stated earlier, the result of this project should be an add-on to the Yaresa app, and should help the CHO's keep better records on child welfare and generate reports faster and easier.

2.3 User Classes

Direct users of this application will be Community health workers in the selected Health Zones that are already using the Yaresa app. These community health workers are already using the Yaresa app, and therefore will not need so much training with the use of tablets. In regard to level of education, they have all gone through some tertiary level of education.

2.4 Operating Environment

The Software will be run on an Android Tablet in any of the designated CHPS zones. There would be the need for a local database on the tablet, and a server hosted database.

2.5 Assumptions and Dependencies

This application will be used to take the records of children that will be registered with the various CHPS using the Yaresa application. It is not intended for outreach purposes. As part of their responsibilities, CHOs may attend to children that are not registered with the health center. The module to be implemented is not intended for such children.

2.6 Procedure for Requirements Gathering

The requirements gathering process comprised of three different methods. These were:

- An interview with a CHO on the main activities and records involved in child welfare – By interviewing the CHO I was able to identify the main concerns of each CHO in managing the records of the children. The CHO's main issue was having to fill many and different records with varying information in different records books. From that perspective, putting all these record books in one place will definitely lift some weight off the shoulders of the CHOs.
- Studying the Ghana Health Service Monthly Nutrition Form. - Studying the Nutrition Form gave the developer an idea of what information the CHO needed to capture when filling monthly reports, thus informing major decisions on the reporting aspect of the Child welfare module to be implemented.
- Studying the Child Health Records Booklet – Studying the Child Health Records Booklet gave the developer insights into how the CHO records the weight, Vitamin A supplementation records and the illnesses of a child. It also gave the developer an idea of the necessary details to account for when it came to these records.

2.7 Scenarios

To get a clearer understanding of the general workflow of a CHO, and his or her interaction with the module to be implemented, scenarios were used. The scenarios developed were based on interviews conducted. Additionally, the scenarios developed give a clearer understanding of how the child welfare module will be used, and what exactly it will be used for.

2.7.1 Scenario A

Mr. Emmanuel is a Community Health Officer at the Health Center in a community. He has been using the Yaresa application already and knows how to use an android tablet. One of his responsibilities is to attend to children under the age of 5 years when they visit the health center. He is supposed to perform thorough checks on them and make sure they are growing well. When a baby visits the health center for the first time, he is supposed to register the baby in the register book. Any other time the child is brought to the health center, Emmanuel will have to record the visit, with information such as the weight of the child and the health condition of the child ie. whether they are malnourished or not. When the child is declared malnourished after two consecutive visits to the health center, he or she is enrolled into the CMAM program. When this happens, Emmanuel will have to note it down in the CMAM Register. If the child is later declared as healthy, the child will be discharged from the CMAM program, and this will be noted by Mr. Emmanuel. Also, in the case where he also administers some Vitamin A to a child, he is supposed to log that in the Vitamin A Tally book.

One of the children he attends to is Kofi Osei. Kofi Osei's mother, Ama Nimo brings Kofi to the health center every two months for weighing. On one occasion, Kofi has a rash on his back. After entering Kofi's details in the attendance book, Emmanuel notes this rash

down along with treatment provided and other notes in Kofi's child record book. This can give Emmanuel some information about Kofi's rash the next time Kofi is brought to the health center.

Analysis of Scenario

From the first scenario, it is clear that the CHO who in this case is Mr. Emmanuel needs some form of data collection tool where data concerning children that visit the CHPS can be stored and retrieved regularly. Considering the registration of children, a concern is raised as to how to facilitate children already added as community members. This is later discussed in the Implementation chapter. Notwithstanding that fact, the decisions made concerning this issue are made at the database level.

2.7.2 Scenario B

Mr. Eben is a Community Health Officer in a community. At the end of the month, he needs to compile all record books and fill the monthly nutrition form which is to be submitted to the District Office. Some of the record books he needs to retrieve information from include the Vitamin A Tally Book, the CMAM record book, the registrants record book and the attendants record book. From all these books, the main information he needs to capture are the amount of vitamin A administered to various age groups of children, the number of children under various age groups that were registered in a month, the number of children enrolled and discharged from the CMAM program, the number of malnourished and healthy children that were attended to in a month amongst others. Having to go through all these record books is a lot of work, and is also time consuming. The time spent on compiling all these records can be spent on doing some more effective work, thus reducing productivity of health workers at the Health center.

Analysis of Scenario

Looking at Scenario B, it can be deduced that Mr. Eben will need a tool to aid him in compiling reports to fill the monthly nutrition form. Putting all records in one place will make compilation easier. Hence, having a system to take care of this will make work a lot easier for Mr. Eben. Particularly, a system where Mr. Eben can specify exactly what reports to be generated.

2.8 Use Cases

The use case diagram in Figure 2.1 defines the main requirements of the system and the interactions of the CHO with the system. These use cases have been developed based on the analysis of the two scenarios in the previous section. As a result, the use cases generally involve the CHO saving and retrieving some data on the tablet, and also, the system being able to generate some reports to aid the CHO in filling monthly reports. See Figure 2.1 for the detailed use case diagram.

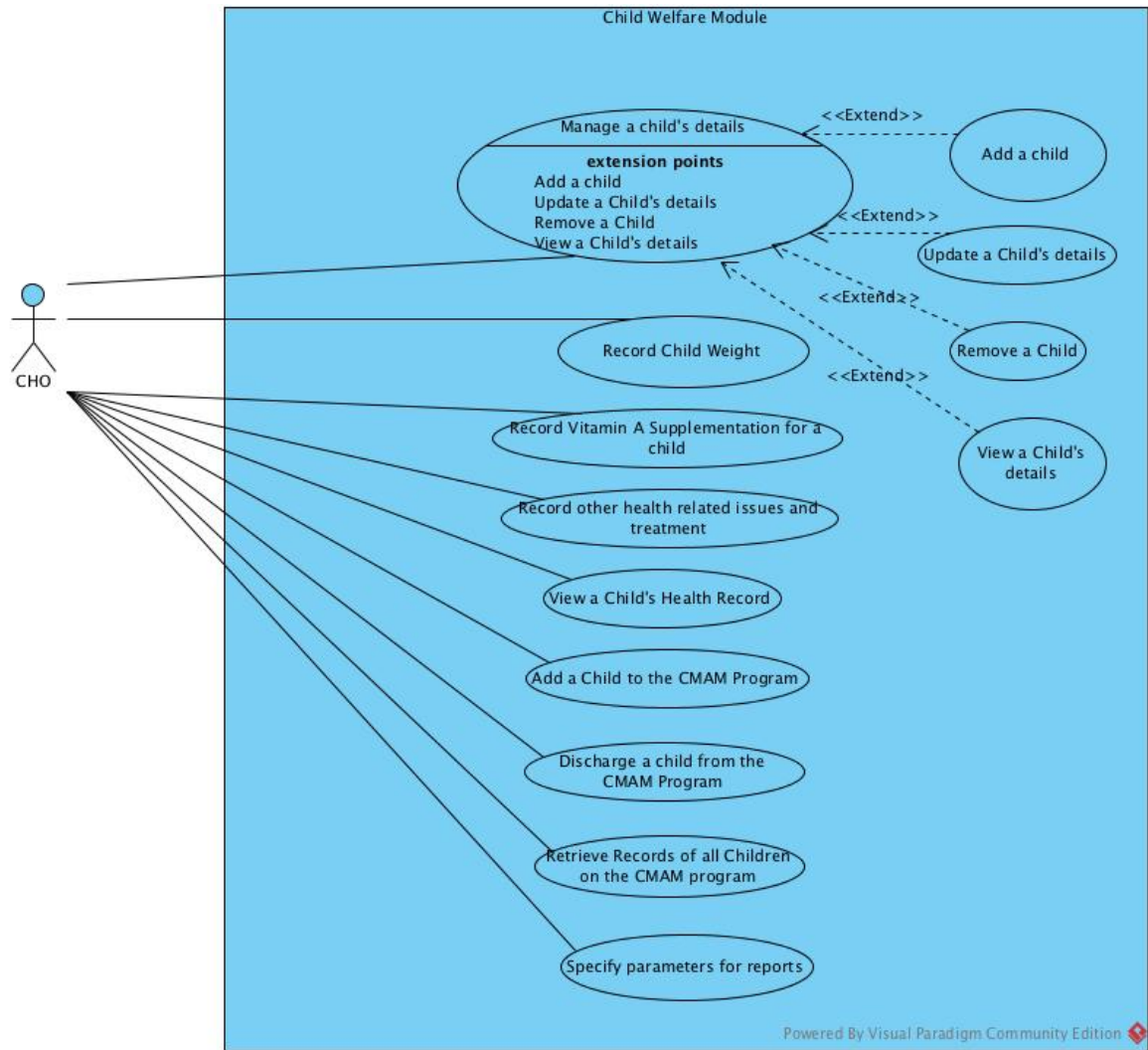


Figure 2.1: Use case diagram

2.9 Functional Requirements

From the information gathered during the requirements gathering process, the analysis of scenarios and the use cases developed, the functional requirements are as follows:

- The user can register a child and manage the child's details, which will include updating and deletion of details.
- The user can record a child's weight and state the health condition of the child. ie. whether the child is healthy or malnourished.
- The user can take records when Vitamin A is provided to a child.

- The user can record any health related issues of the child. ie. enter records of illnesses and treatments.
- The user can view a child's past records which encompass the weight, vitamin A supplementation records and the health related issues records.
- The user can add children to, and discharge them from the CMAM program.
- The user can determine/specify the necessary parameters for generating reports.

*For detailed documentation on these functional requirements and image of monthly nutrition form, see appendix.

2.10 Other Requirements

Some other necessary requirements may include the reporting section of the module, which will deal with the generation of reports to help the CHO fill the monthly nutrition report. Some of the information that will be captured include the number of children registered in a month, the amount of vitamin A supplied amongst others. Therefore, the CHO should be able to specify the exact information or reports he wants to be generated by the application.

2.11 Nonfunctional Requirements

- The application should always be accessible to the user. Thus, use of the application should not be disrupted by Network problems.
- The application should be able to generate reports.

2.12 Activity Diagram

This section presents an activity diagram which represents the process a CHO will go through when a child visits the health center. It also shows the CHO's interaction with the Child Welfare Module.

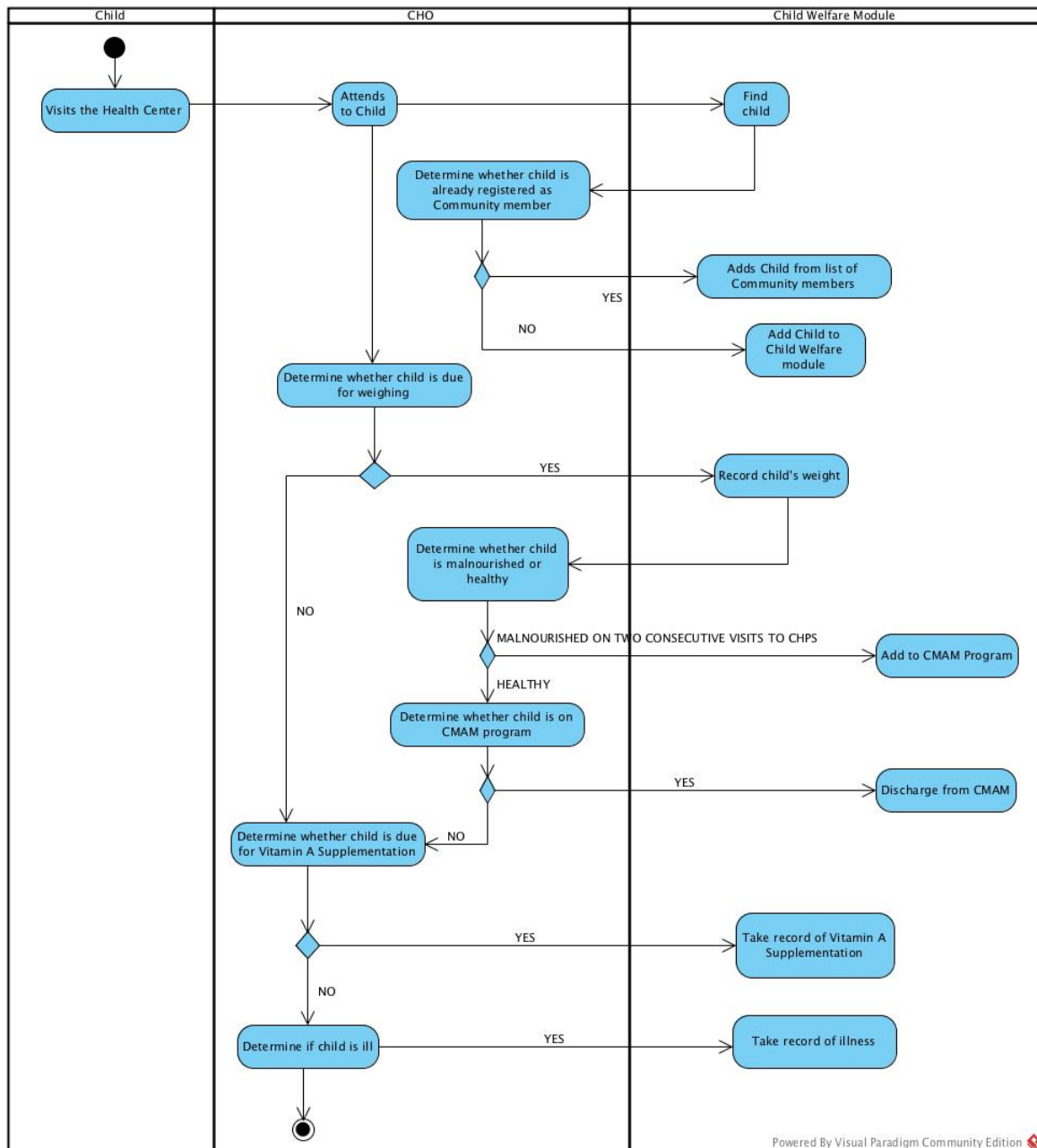


Figure 2.2: Activity diagram

2.13 Architecture pattern

The main Architecture pattern being used for the implementation of this module is the MVC pattern. First and foremost, Android Development generally makes use of the MVC pattern, and this works by separating the views and data, and their interactions from each other. This will allow me to focus on each aspect of the application separately in the sense that I can focus on designing all views before I work on the backend or vice versa. Another advantage will be the fact that it makes the testing of the model, which represents the data, easier.

Model – For this particular implementation, the model will include the SQLite database and the ‘DataClass’ which extends the ‘SQLiteOpenHelper’ class, and is responsible for the creation of the database. There also exist the classes which extend the ‘DataClass’ and are responsible for the creation of SQL statements for the various entities. These are the Children class, the ‘CMAMRecords’ class, the ‘ChildHealthIssuesRecords’ class, the ‘VitaminASupplementationRecords’ class and the ‘ChildNutritionRecords’ class.

Views – The Views and layouts of this module comprise of the many xml files located in the res folder and a few java classes responsible for initializing layout files to render the view.

Controller – The Controller comprises of the various classes that make use of methods that read or append some information to the database through the view. The user which in this case is the CHO enters some information in the view and submits it. The controller will then handle this event, and respond by performing a database transaction accordingly. The main controller classes for this project include the ‘Children’, ‘ChildHealthIssuesRecords’, ‘VitaminASupplementationRecords’ and ‘ChildNutritionRecords’ classes.

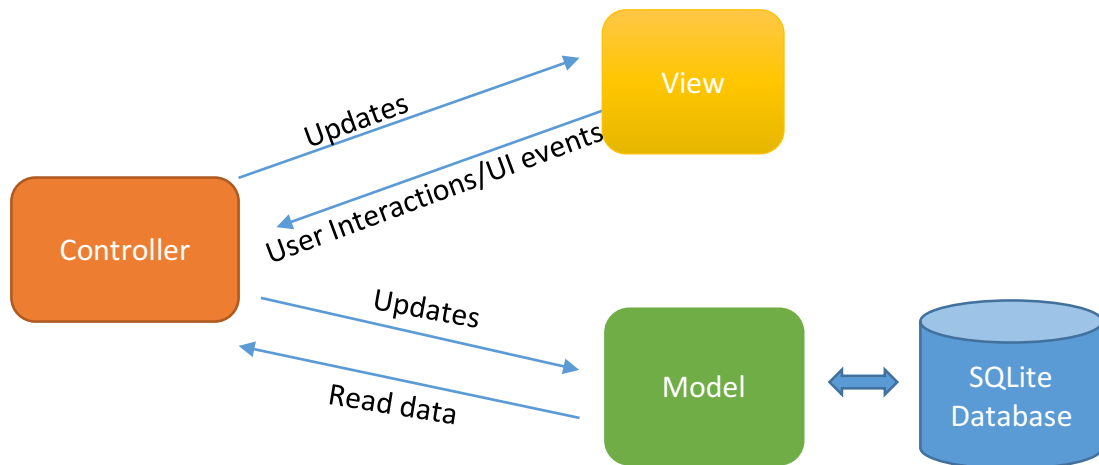


Figure 2.3: Model view architecture

2.14 Modules

The main modules of Child Welfare comprise of the Child Module, Health Issues and Treatment module, Vitamin A supplementation module, the nutrition module, the CMAM module, and the Reports Module. Each module consists of three different java classes and one main xml file to represent the view. However, a module may contain other xml files to represent other parts of the view. For instance, the Health Issues and Treatment module consist of a 'ChildHealthIssues' class, a 'ChildHealthIssuesRecords' class, a 'HealthIssuesFragment' class and a 'fragment_child_record_health_issues.xml' file.

- The 'ChildHealthIssues' class represents a Health Issue.
- The 'ChildHealthIssuesRecords' class acts as a controller and interacts with the database and performs functions such as adding, editing and deleting a health issue and treatment record.
- The 'HealthIssuesFragment' class acts initializes the view and provides model data to the view, and also interprets user actions such as button clicks.
- The 'fragment_child_record_health_issues.xml' file represents the view and displays model data.

Figure 2.4 represents a layered view of the main modules of the system to be implemented.

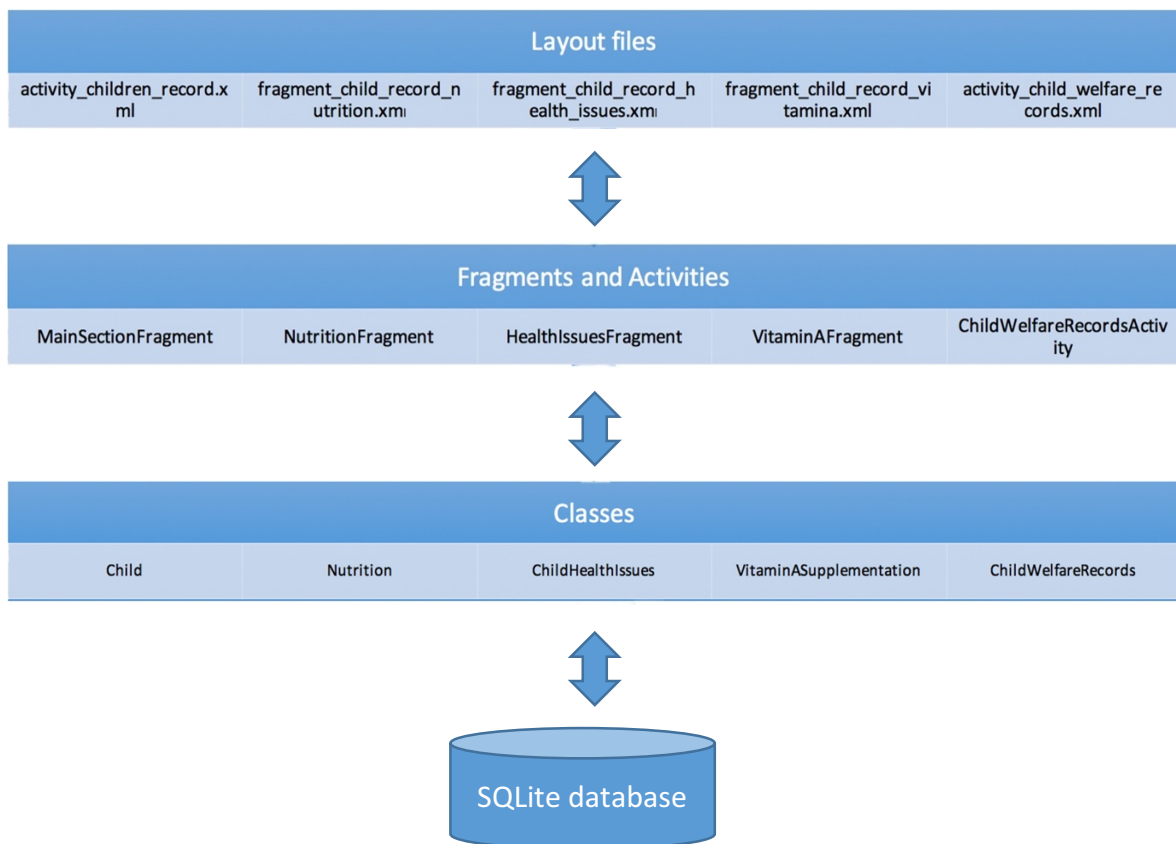


Figure 2.4: Layered view of modules

2.15 Data Model

Data persistence will be addressed using a relational database. To make sure that the data is always available to the CHO in the case of poor network connectivity, a local copy of the data must be kept to facilitate the CHO's work. This is attained by using an SQLite database for local storage on the tablet, and synchronizing with a MySQL Database on a specified server when the tablet is connected to the internet. The model of the database can be seen in Figure 2.5. In the rest of this section, the most important database decisions would be discussed.

For the child welfare module database, there are five main tables and these are:

- Child Table
- Nutrition Table
- CMAM Table
- Health Issues Table
- Vitamin A Supplementation Table

First of all, the Community member table is introduced in the database architecture because the table is altered to make way for the 'is_child' column. This column determines whether a community member is registered under the child welfare module or not. The child table also has a 'community_member_id' attribute which references that of the 'community_members' table. When a child is added to the child table, it is also added to the 'community_members' table but with fewer attributes. The same way, a child can be added from the community member table to the child table. The 'community_member_id' attribute links the two tables.

The CMAM, 'Vitamin_A', 'Nutrition' and 'Health_Issues' tables all have a common attribute: 'reg_no', which references the 'Child' table. This attribute represents the unique registration number of every child. CHO's use this number anytime they have to take any records of children in the various child welfare record books, and as such this number is supposed to connect different records in the four tables to a child.

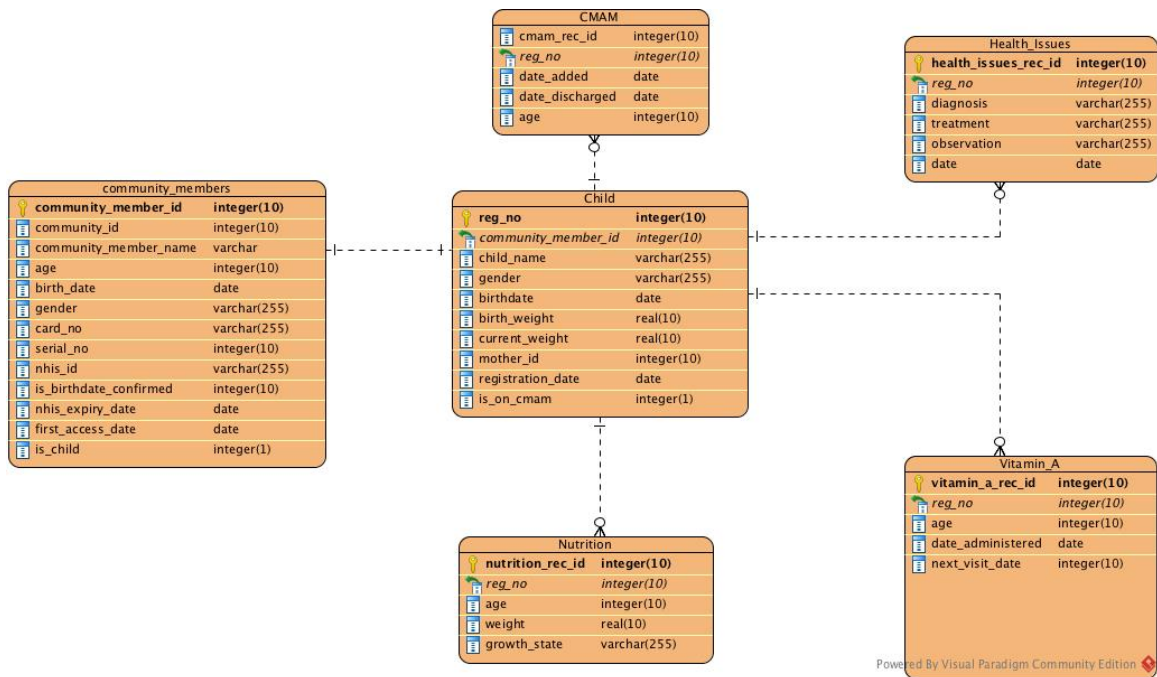


Figure 2.5: Database architecture

2.16 Minimal Child Welfare Class Diagram

Figure 2.6 presents a minimal representation of the main classes for the implementation of the proposed module.

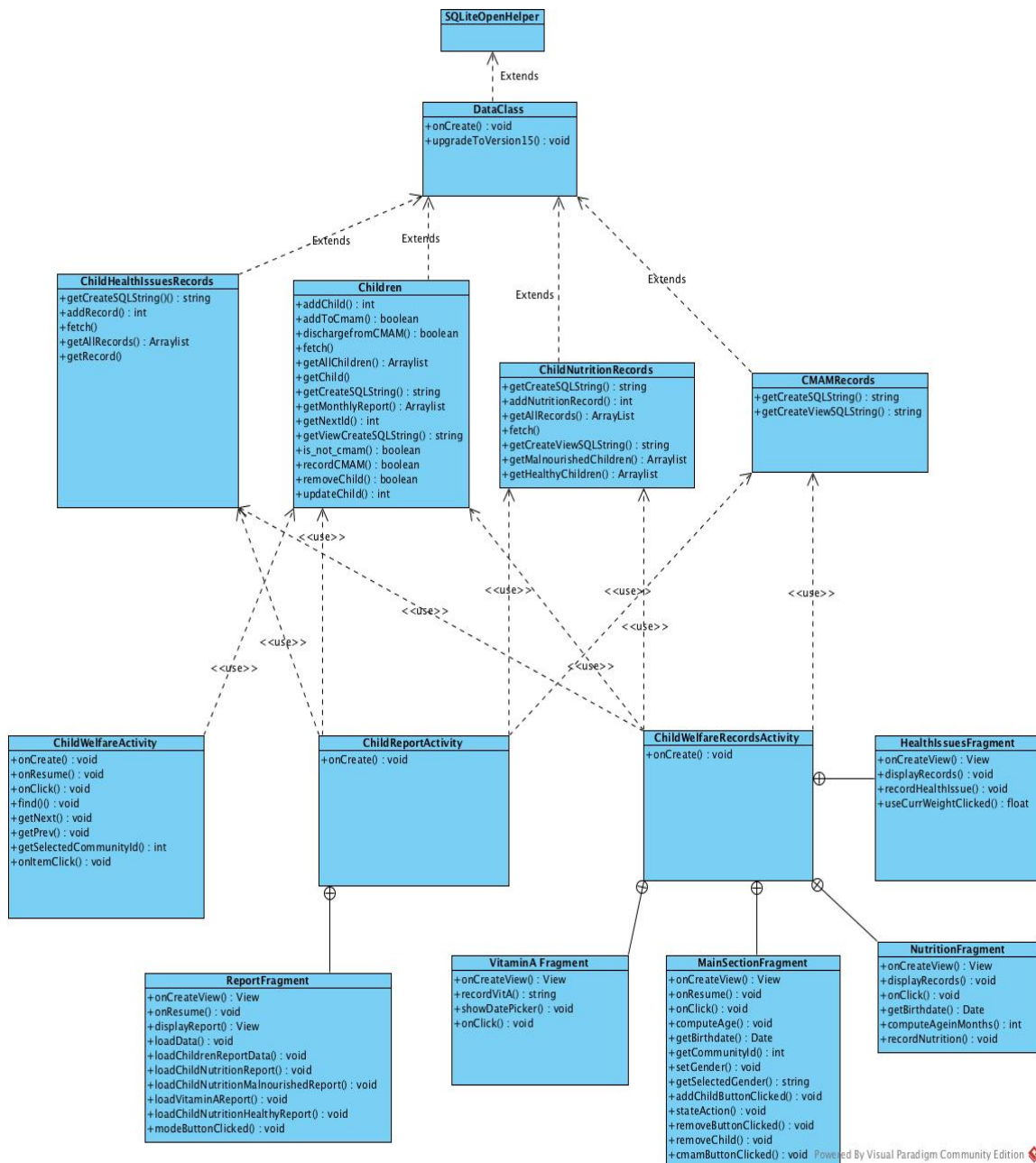


Figure 2.6: Minimal child welfare class diagram

Chapter 3: Implementation

This chapter presents the implementation of the Child Welfare Module. It provides a high level description of implementation including the various libraries, frameworks and components used in developing the prototype. It also explains and gives a better description of how a CHO will go through the process of adding and managing information concerning the children he/she attends to.

3.1 Technological Platform

For this module, an android development environment will be needed to merge the child welfare module with the existing android application. As such, Android Studio with all other related tools had to be installed by the developer. This included an Android 5.1.1 SDK Platform package with an API level of 22. A tablet was also necessary to aid the developer in the implementation and testing stages of developing the application. Additionally, for version management, Github was used to effectively integrate the child welfare module into the existing Yaresa application.

3.2 Implementation of Database

The Yaresa application makes use of an SQLite database. For the creation of tables, there is a DataClass which extends the Android SQLiteOpenHelper Class. This DataClass acts as an intermediary between the various controller classes and the sqlite database. For the developer to be able to make any updates or changes to the database, the DataClass had to be edited. There are various methods in the DataClass which represent updates made on the database. The latest of which was `upgradeVersion14()`. For that matter, the developer had to implement an `upgradeVersion15()` method to be able to add on changes. The main

reason for doing this is because there may be data already existing, and creating a new database will cause the existing data to be lost.

Furthermore, the SQL statements responsible for the various database transactions are found in their respective controller classes. For instance, the SQL statement for creating a child table is found in the Children controller class. Figure 3.1 shows the upgradeVersion14() method implemented and the various tables added, including changes made to existing tables.

```
private void upgradeToVersion15(SQLiteDatabase db){
    //creating children table for child welfare module- version 15

    db.execSQL(Children.getCreateSQLString());
    db.execSQL(ChildHealthIssuesRecords.getCreateSQLString());
    db.execSQL(ChildNutritionRecords.getCreateSQLString());
    db.execSQL(CMAMRecords.getCreateSQLString());
    db.execSQL(Children.getViewCreateSQLString());
    db.execSQL(ChildNutritionRecords.getCreateViewSQLString());
    db.execSQL(CMAMRecords.getCreateViewSQLString());
    db.execSQL(VitaminASupplementationRecords.getCreatViewSQLString());
    db.execSQL("alter table"+ CommunityMembers.TABLE_NAME_COMMUNITY_MEMBERS +" " +
        "add column "+ CommunityMembers.IS_CHILD+ " integer default 0");
    //resetDataBase();
    setDataVersion(db, DATABASE_NAME, 15);
}
```

Figure 3.1: upgradeToVersion15() method

3.3 Interfaces and Functionalities in Detail

This section will present the various interfaces of the Child welfare module, and will take readers through the process of using the application. This will be done by going through all functional requirements, and describing how the application implements or accomplishes them.

To add a child, the CHO will have to navigate to the home page of the Child Welfare Module from the Yaresa Main Page. He does this by clicking on the Child welfare button. At the home page, the CHO has the option of adding new child or adding a child from a list of children already added as community members on the application. When the CHO clicks the add child button, it takes him to the Child Profile page where he can fill the necessary

fields. Clicking on the the “Add Child” button adds a child to the database. Furthermore, the CHO can add a child from an existing list of children that have been added as community members on the application. By clicking on the “Add child from list of community members” button, the CHO is redirected to a page where he can add/register a child on the Child welfare from a list of children already existing on the application as community members. For this to be possible, an `is_child` attribute is added to the community members table and this determines whether a community member is under the child welfare and surveillance module. Once a child is added from community members, this attribute is set to true. Figure 3.3 gives a clearer view of the process of adding a child.

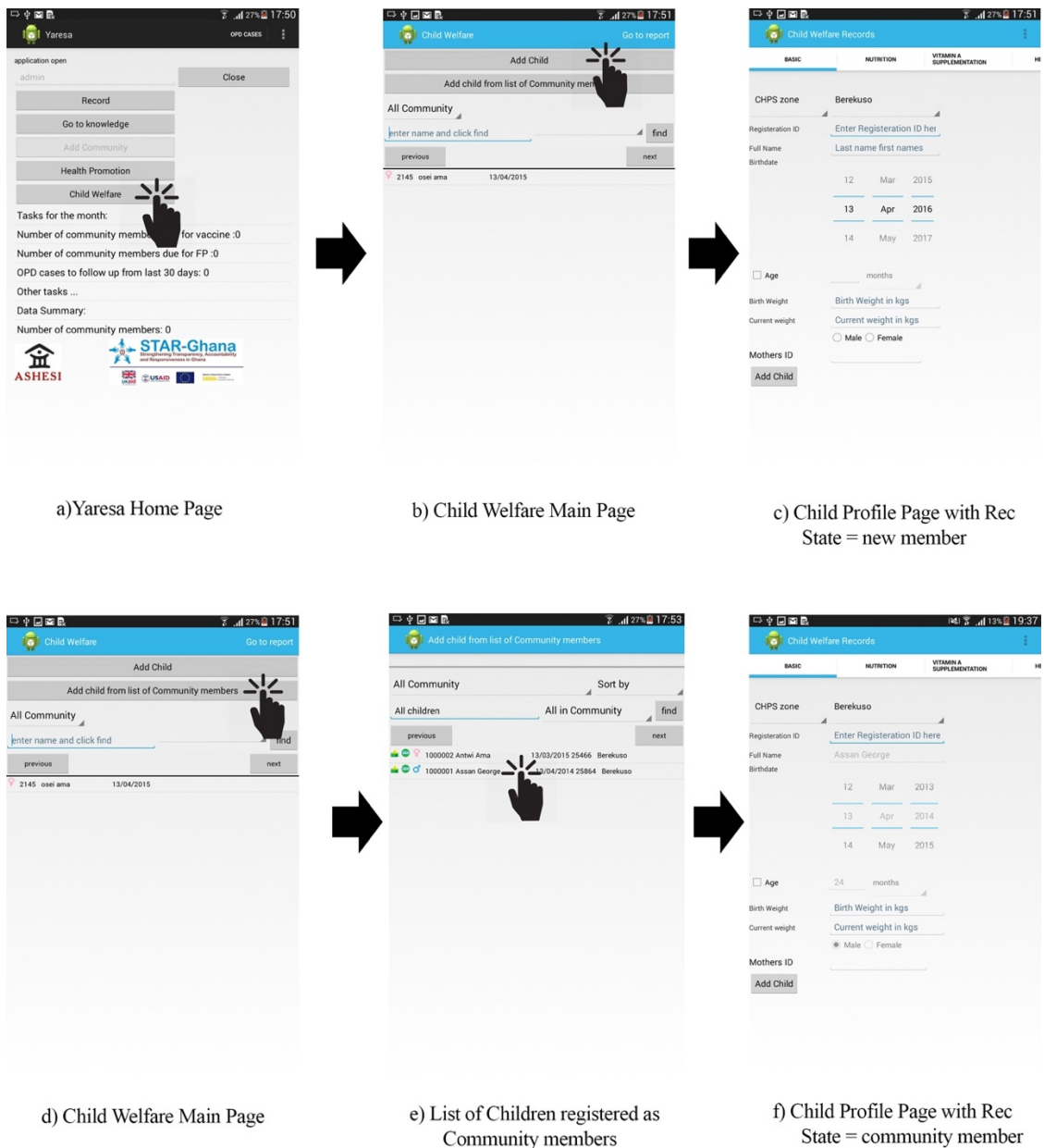


Figure 3.2: Adding a child

Additionally, the Child profile page makes use of record states. When a child is to be added, the record state is set to 'new member' otherwise the record state is set to 'edit member', and the page is updated accordingly. This implies that if a CHO is to click on a child in the list on the Child Welfare Home Page, he will be redirected to the Child Welfare page with a record state of 'edit member' which corresponds with a particular number, and with the child's details loaded in the appropriate fields. The snippet of code in Figure 3.3

shows how this is achieved. Figure 3.4 will also give a clearer view of the implementation of the Child profile page that holds a child's details.

```

if(state==STATE_NEW_MEMBER){
    fillCommunitiesSpinner(communityId);
}
else if(state==STATE_COMMUNITY_CHILD){
    CommunityMembers members=new CommunityMembers(getActivity().getApplicationContext());
    CommunityMember cm=members.getCommunityMember(communityMemberId);
    editChildName.setText(cm.getFullname());
    setGender(cm.getGender());
    setBirthdate(cm.getBirthdateDate());
}
else{
    //load child information
    Children children=new Children(getActivity().getApplicationContext());
    Child child=children.getChild(reg_no);

    String childregno = String.valueOf(child.getRegno());
    editRegNo.setText(childregno);

    setBirthdate(child.getBirthDate());

    editChildName.setText(child.getChildname());
    //if the birthdate is not confirmed, the user can confirm it;

    computeAge();

    String motherid = String.valueOf(child.getMotherid());
    editMotherId.setText(motherid);

    Log.e("Children.getAllChildren(int)","Exception "+child.getMotherid());

    String birthweight = String.valueOf(child.getBirthweight());
    editBirthWeight.setText(birthweight);
    String currentweight = String.valueOf(child.getCurrentweight());
    editCurrentWeight.setText(currentweight);
    //setNHISExpiryDate(cm.getNHISExpiryDateDate());
    setGender(child.getGender());
    //fillCommunitiesSpinner(child.getCommunityId());

    if(child.is_CMAM()== true){
        cmamBtn.setText("Discharge from CMAM");
    }
    else if(child.is_CMAM()== false){
        //cmamBtn.setText(R.string.addToCMAM);
        cmamBtn.setText("Add to CMAM");
    }
}

fillCommunitiesSpinner(communityId);
fillSubDistrictSpinner(

```

Figure 3.3: code snippet of use of record states

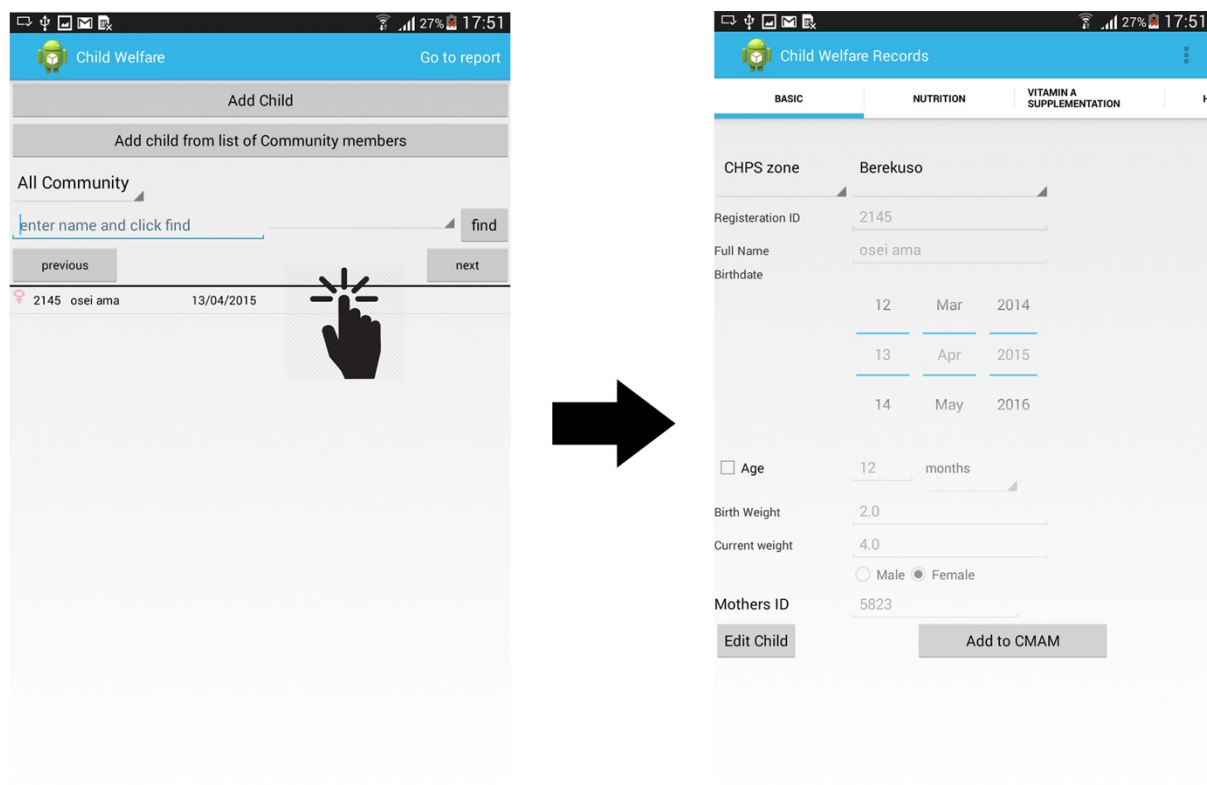


Figure 3.4: Visiting an existing child's profile

With respect to recording a child's weight, the CHO will again have to navigate to the Nutrition page of a Child's profile. There, the CHO can enter the child's weight, select the growth condition of the child, and then submit them. The record is then added to the list of existing records right below the form. Figure 3.5 shows the Nutrition page and the fields to be filled with the list of past records beneath it.

The screenshot displays the 'Child Welfare Records' application interface. At the top, there's a blue header with the app name and a status bar showing the time as 15:51. Below the header, there are four tabs: 'BASIC', 'NUTRITION' (which is selected), 'VITAMIN A SUPPLEMENTATION', and 'H'. The 'NUTRITION' tab contains a 'Weight' input field with a blue line graph, a 'Growth Condition' dropdown menu set to 'Normal', and a 'Record' button. Below the form, there is a list of records: '2016-03-27 Normal' and '2016-03-27 Malnourished'.

Figure 3.5: Nutrition page

Still on a child's profile, the CHO can navigate to the Vitamin-A supplementation page and take records any time he gives Vitamin A to a child. On clicking the 'Record' button, a date picker pops up and the CHO will have to choose the appropriate date. The date of next visit section is then automatically updated, or set to a date exactly 6 months later. Refer to Figure 3.6 for the Vitamin A Supplementation Page.

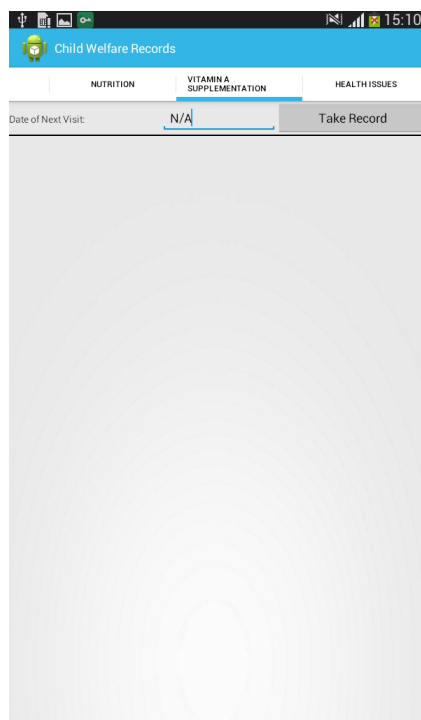


Figure 3.6: Vitamin A supplementation page

Again, on a child's profile, the CHO can navigate to Child Health Issues page where he can take note of a child's illness as and when the child visits the Health Center. He will have to fill all necessary fields and click on the 'Record button'. The record is then added to the list of health cases below the form. Clicking on a past record takes a CHO to a page where he can view detailed information on a past record. Figure 3.7 shows the health issues page with the list of past records beneath the form.

The screenshot shows a mobile application interface titled 'Child Welfare Records'. It has three tabs: 'NUTRITION', 'VITAMIN A SUPPLEMENTATION', and 'HEALTH ISSUES'. The 'HEALTH ISSUES' tab is selected. The form contains the following fields:

- Temperature: A horizontal slider bar.
- Weight: A checkbox labeled 'Use Child current weight' followed by a text input field labeled '/ Enter weight here'.
- Diagnosis: A text input field.
- Treatment: A text input field.
- Observation: A text input field.

At the bottom of the form is a grey button labeled 'Record'.

Figure 3.7: Health issues page

Furthermore, the CHO can add or discharge a child from the CMAM program by clicking on the “Add to CMAM” or “Discharge from CMAM” button. This will add a record in the database when the add button is clicked and will update an existing record when the discharge button is clicked. This is achieved by using record states and an ‘is_on_cmam’ attribute in the child table. When a record is added to the ‘cmam’ table, it is added with a record state of new. When a child is discharged from the CMAM table, the date discharged is updated and the record state is automatically changed to “deleted”. In the code, there are various checks to make sure that multiple records are not added to the table when there is an existing new record in the table. Thus, a new record can only be added when the record states of existing records have been changed to “deleted.” The snippet of code in Figures 3.8 and 3.9 show how this is achieved.

```

public boolean recordCMAM(int regno, int age ){
    try {
        db = getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(Children.REGISTRATION_NO, regno);
        cv.put(CMAMRecords.AGE, age);
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.UK);
        String date_added = "1900-01-01";
        Calendar c = Calendar.getInstance();
        date_added = dateFormat.format(c.getTime());
        cv.put(CMAMRecords.DATE_ADDED, date_added);
        cv.put(DataClass.REC_STATE, DataClass.REC_STATE_NEW);
        /**
         * if (is_cmam == true) {
         *     cv.put(CMAM, IS_NOT_CMAM);
         * } else {
         *     cv.put(CMAM, CMAM);
         * }
         */

        if (db.insert(CMAMRecords.TABLE_NAME_CMAM, null, cv) <= 0) {
            return false;
        }
        return true;
    } catch (Exception ex) {
        Log.e("Children.recordCMAM", "Exception ex" + ex.getMessage());
        return false;
    }
}

```

```

public boolean dischargefromCMAM(int regno) {
    try {
        String selector = REGISTRATION_NO + "=" + regno + " ";
        selector += " AND " + DataClass.REC_STATE + "=" + DataClass.REC_STATE_NEW;

        db = getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(Children.REGISTRATION_NO, regno);
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.UK);
        String date_discharged = "1900-01-01";
        Calendar c = Calendar.getInstance();
        date_discharged = dateFormat.format(c.getTime());
        cv.put(CMAMRecords.DATE_DISCHARGED, date_discharged);
        cv.put(DataClass.REC_STATE, DataClass.REC_STATE_DELETED);
        if (db.update(CMAMRecords.TABLE_NAME_CMAM, cv, selector, null) <= 0) {
            close();
            return false;
        }
        close();

        return true;
    } catch (Exception ex) {
        Log.e("CMAMRecord.dischargefromCMAM", "Exception " + ex.getMessage());
        return false;
    }
}

```

Figure 3.8: Snippet of code for adding and discharging from CMAM

```

public void cmamButtonClicked(){
    Children children=new Children(getActivity().getApplicationContext());
    int age = computeAgeinMonths();
    Child child=children.getChild(reg_no);

    if(!child.is_CMAM()) { //if the child is not on the CMAM program
        if (children.addToCmam(reg_no) && children.recordCMAM(reg_no, age)) {
            Toast.makeText(getActivity(), "Child successfully added to CMAM", Toast.LENGTH_SHORT).show();
            cmamBtn.setText("Discharge from CMAM");
        } else {
            showError("Error adding child to Cmam");
            return;
        }
    }
    else if(child.is_CMAM()){ //if the child is on the CMAM program
        if(children.dischargefromCMAM(reg_no) && children.is_not_cmam(reg_no)){
            Toast.makeText(getActivity(), "Child successfully discharged from CMAM", Toast.LENGTH_SHORT).show();
            cmamBtn.setText("Add to CMAM");
        }
        else{
            showError("Error discharging from Cmam");
            return;
        }
    }
}

```

Figure 3.9: Snippet of code for adding and discharging from CMAM etc..

Finally, the CHO can specify the parameters for generation of various reports. The CHO does this by navigating to the Reports section of the module. Here, the CHO has the option of generating reports on the number of Children registered in a month, the amount of vitamin A supplied in a month, the number of children admitted and discharged from the CMAM program, and records on the number of healthy and malnourished children. The Child Welfare report view can be seen in Fig 3.10.

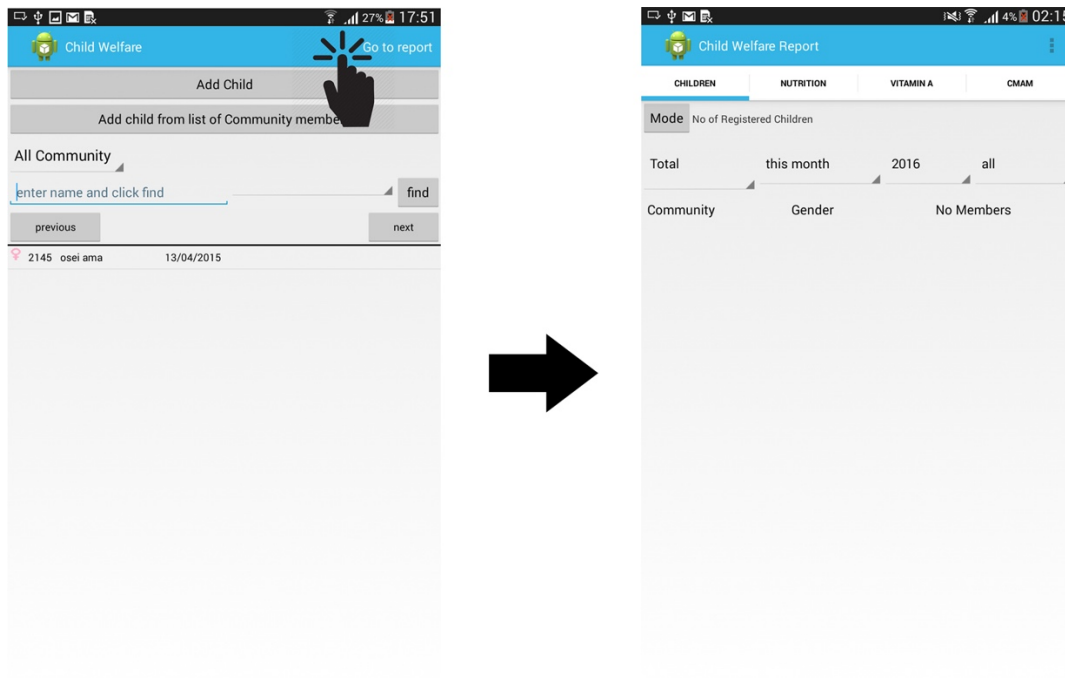


Figure 3.10: Child Welfare Reports

3.4 Design and Implementation Issues

In developing the child welfare module, the developer faced quite a number of challenges. One of the main challenges faced included migrating the existing android project to gradle to be able to work on it in Android Studio. This took about two to three days to figure out. Another pressing issue or difficulty was familiarizing myself with the existing project. As a matter of fact, it took a lot of time to fully understand the existing database architecture, and code to be able to start work on the module.

Chapter 4: Testing and Results

This chapter provides a description of the testing methods the various components of the system and the system as a whole went through. It will provide detailed description of testing of each functional requirement and present the results as required. Furthermore, it will give an analysis of all test results and explain why certain components or features may not be working as desired.

4.1 Development Testing

During development of the module, testing was carried out at three levels of granularity (Sommerville, 2011). The first of them being the **unit testing** where the functionality of various objects or methods were tested. An example can be seen in the snippet of code in Figure 4.1 where the developer tests the `getInsert()` method which was responsible for adding a child to the SQLite database, provided the right parameters. In going through the code, readers will come across a lot more of these unit tests which were meant to make sure that certain methods functioned properly.

```
public static String getInsert(int regno, String childname, String gender, String birthdate, float birthweight, float current_weight, int motherid) {
    Date date = new Date();
    DateFormat dt = new SimpleDateFormat("yyyy-MM-dd", Locale.UK);
    return "insert into "
        + TABLE_NAME_CHILDREN + " ("
        + REGISTRATION_NO + ", "
        + CHILD_NAME + ", "
        + GENDER + ", "
        + BIRTHDATE + ", "
        + BIRTH_WEIGHT + ", "
        + CURRENT_WEIGHT + ", "
        + MOTHER_ID + ", "
        + REGISTRATION_DATE + ", "
        + DataClass.REC_STATE
        + ") values("
        + " " + regno + ", "
        + childname + ", "
        + gender
        + " " + dt.format(date) + ", " +
        " " + birthweight + ", "
        + current_weight + ", "
        + motherid + ", " + DataClass.REC_STATE_NEW + " ) ";
}

db.execSQL(Children.getInsert(1234, "Ama Owusu", "female", "2016-02-25", 2.5, 8234));
```

Figure 4.1: Unit test for adding a child

The other forms of testing used were component and system testing, where the various modules of the system and the system developed as a whole were tested. For

instance, after developing the child module, it was tested. This involved adding a child, viewing a child's details, editing a child's details and removing a child.

The main goal of all the tests carried out were to make sure all functional requirements stated earlier were implemented and were functioning properly. Accordingly, test cases were developed to evaluate each functional requirement. For instance, for adding a child the test cases were as follows.

- Adding successfully.
- Adding- Unsuccessful when user leaves out some fields.
- Adding – Unsuccessful when user enters a registration number that already exists.

The test carried on most of the other functionalities that had to do with the user appending some data to the database followed the same format. Test cases were developed and tests were conducted. Consequently, the outcomes of the tests were evaluated against the expected results, and conclusions and recommendations developed. All tests performed came out with the expected results. Table 4.1 represents the various test cases considered, with their preconditions and outcomes.

Table 1: Test Cases

Test Case	Precondition/Steps	Expected result	Post condition	Result
Adding a child successfully to the child database through a form.	All fields are filled and registration number does not already exist.	Child is added to the database.	A toast message to show if the transaction was successful	Child was added to database and toast message was shown.

Error message when adding or updating a child's details.	Some fields in the form for adding a child are not filled.	Child is not added to the database and error message is displayed notifying user of field that has not been filled.	Error message is displayed notifying user of field.	Error message was displayed notifying user of empty fields, and child was not added to database.
Deleting a child's details from the database.	Click the remove child button	Child is deleted from the database.	Toast message to show if deletion was successful.	Child was deleted from database and toast message.
Successfully adding a child nutrition record to the database and to a list through a form.	All necessary fields are filled and record button is clicked.	Record is added to database and list below the form.	Toast message to show if the transaction was successful or not.	Record was added to database and list.
Adding a health issue record to the database and to a list through a form.	All necessary fields are filled and record button is clicked.	Record is added to database and list below the form.	Toast message to show if the transaction was successful or not.	Record was added to database and list.
Successfully returning an arraylist containing a report on the number of children registered depending on the age, gender and date of registration.	Specifying the age, gender and the date registered.	Arraylist is returned with desired result.	Grid adapter is updated with arraylist returned.	Arraylist was returned with desired result.
Successfully adding a child to the CMAM program.	Click the "Add to CMAM" button.	Record is added to the CMAM database with a	A toast message to show if the	Record was added to CMAM

		record state of new.	transaction was successful or not.	database with a record state of new and toast message was shown.
Successfully discharging a Child from the CMAM program.	Click the “Discharge from CMAM button.	Previous record added has its record state changed to deleted.	A toast message to show if the transaction was successful or not.	Previous record’s state is changed and toast message is shown.

Chapter 5: Conclusion and Recommendations

Presently, the Yaresa application aids CHOs in managing digital health records, tracking out Patient Department Cases, immunization and family planning. Nevertheless, due to surveys conducted, some necessary modules had to be considered and the Child welfare module was one of them. The Child welfare module was developed primarily to aid CHOs in keeping good tabs on the healthcare of children in the communities they work in, and also to make the work of compiling monthly reports easier for them. Currently, the module that has been implemented helps the CHO by putting various records in one place as opposed to having them in different record books. Now, a CHO can record the weight, illnesses, vitamin A supplementation records of children in one place. More importantly, they can generate reports from the data stored to aid them in filling their monthly reports. However, some limitations of this implementation have been identified. Some of these will be discussed in this section, along with some recommendations.

5.1 Limitations and Recommendations

- For the implemented module, a main limitation is the fact that it does not cater for the outreach work of CHOs. From time to time, CHOs visit schools, and also go on outreach, thus they come into contact with a number of children that are not accounted for in this module. With the outreach being an integral part of the day-to-day operations of the CHO, it has to be factored in the implementation of the module if any further work has to be done.
- Another limitation of the module implemented was detected when taking Vitamin A supplementation records. The current module takes in records by filling a form

and submitting it. Ideally, there should be a scheduled list just as is seen in the Child record book to improve usability.

5.2 Future Work

- Push notifications – On completion of the Child Welfare Module, it was suggested that the push notifications be integrated in the module to improve usability, and to alert CHOs when necessary. With the Vitamin A module for instance, push notifications will be a handy tool as this can be used to alert a CHO when a child is due for Vitamin A supplementation or weighing. Again, this will be a very effective way of improving child surveillance.

References

- Collins, F. S. (2015). *Growing importance of health in the economy-Outlook on the Global Agenda 2015*. Retrieved October 2015, from World Economic Forum Web site: <http://widgets.weforum.org/outlook15/10.html>
- Russel, S. (2008). Community-based Health and Planning Services: Decentralizing Ghana's Health System. *Georgetown University Journal of Health and Sciences* , 5 (1).
- Finkel, R. (2012, August 13). *Mobile Health Around the Globe: Ghana - Changing the Very Essence of Healthcare*. Retrieved April 2, 2016, from HealthWorks Collective: <http://www.healthworkscollective.com/rhona-finkel/42661/mobile-health-ghana-changing-very-essence-healthcare>
- Nyonator, F. K., Awoonor-Williams, J. K., Phillips, J. F., Jones, T. C., & Miller, R. A. (2005). The Ghana community-based health planning and services initiative for scaling up service delivery innovation. *Health policy and planning*, 20(1), 25-34.
- Sommerville, I. (2011). *Software engineering* (9th ed.). Boston: Pearson.
- Ghana Statistical Service. (2012). *2010 Population and housing census. Summary of final results*. Accra: Ghana Statistical Service.

Appendix

Detailed Functional Requirements

1. The user can Add/Register a child

Brief description

The CHO can add or register a child on his or her first visit to the CHPS. During the gathering of functional requirements however, it was noticed that a child could already exist in the community member table. This meant that provision will have to be made for such a case. The solution however, is to provide the CHO the option of adding a child from community members. Chapter 4 gives an in-depth explanation of how this is achieved. Below are the various inputs and outputs for adding or registering a child.

Inputs

- Registration number
- Child's full name
- Child's date of birth
- Child's sex
- ID of the child's mother
- Birth weight
- Current weight

Outputs

- A toast message to show whether the transaction was successful.
- An error message to fill some fields if they were incorrectly filled or not filled at all.
- Child is added to list of children.

Constraints

- A new record cannot be added if the ID of the mother, the child's name and date of birth correspond with an existing record.

2. The user can record a child's weight to measure whether the child is malnourished or not.

Brief description

A CHO is supposed to measure and record the weight of a child that is brought to the health center for weighing. When the weight of the child is taken, the CHO has to record it and determine whether the child is malnourished or not. This he does by entering the child's weight on a graph against the child's age. If the record on the graph falls below a particular threshold, the child is underweight and the CHO will give the mother the necessary recommendations. In the case of the Yaresa application, the CHO will just have the ability to record the weight and the health condition. Ideally the application should inform the CHO whether the child is malnourished or not but due to time constraints this will not be the case. He will have to graph in the Child record book and take record on the application.

Inputs

- Weight
- Health condition

Output

- A toast message to show if the record was posted successfully or not.
- Added to the list of the child's nutrition records.

3. The user can record vitamin A supplementation for a child.

Brief description

Every six months, a mother is supposed to bring her child for vitamin A supplementation. When the CHO attends to the child, he or she is supposed to search for the child's records and take record of the supplementation.

Input

- Date administered
- Date of next visit

Output

- A toast message to show if the record was posted successfully or not.
- Added to the list of Vitamin A records.
- Date of next visit is updated.

4. The user can record any other health related issues of the child. Enter records of illnesses and treatments. For instance, a record can be taken when the child has malaria.

Brief description

A child may be brought to the health center when the child is ill. Under normal circumstances, the CHO will have to record the child's illness in the the Child Record Book for future referencing.

Inputs

- Diagnosis
- Treatment
- Observation
- Child's Weight
- Child's age

Output

- The submitted details will be added to a list of all illnesses and treatments attributed to that particular child.

5. The user can view a child's health record or profile. This encompasses viewing previous illnesses of a child and the treatments provided, viewing the child Vitamin A schedule report and viewing the child's growth and nutrition record

Brief description

Once the CHO needs to make reference to some past records of a child, he would have to look them up in the many registers or from the Child record book. In the case of the application, he can access all records attributed to a child in one place.

6. The user can specify the necessary parameters for reports to be generated.

Brief description

At the end of the month, the CHO has to fill the monthly nutrition form which is supposed to be submitted to the District office. See Appendix for the Monthly Nutrition form.

Inputs

- Specific time frame

Output

- A report will be generated based on the selected parameters.

7. The CHO can mark a child as being part of the CMAM program and discharge a child from the CMAM program.

Brief description

After weighing a child, a CHO is supposed to determine whether the child is healthy or malnourished. In the case where the child is declared malnourished after two consecutive visits, the CHO will recommend that the child join the CMAM program to promote better growth and nutrition. If after four months, the child is declared healthy, the CHO can discharge the child from the CMAM program.

Input

- Date Enrolled
- Date discharged

Output

- A toast message to indicate if the transaction was successful or not.

Monthly Nutrition Form

GHANA HEALTH SERVICE MONTHLY NUTRITION REPORTING FORM							
NAME OF FACILITY:		District: Akwapim South		Region: Eastern		Month: Year:	
GROWTH MONITORING AND PROMOTION							
Age (Mths)	Registrants	Attendance	Moderate(Underweight) (-3 to -2)	Normal (-2 to +2)			
0-11							
12-23							
24-59							
Total							
EXCLUSIVE BREASTFEEDING				BIRTH WEIGHTS			
No. of Lactating Mothers Seen				Total births		Birthweight <2.5kg	
No. of Lac. Mothers Initiating EBF within 1hr of birth							
No. of Lac. Mothers Practising EBF							
Current No. of MTMSGs formed							
No. of MTMSG Meetings Held with Health Staff							
No. of BF Mothers Visited/Supported by MTMSGs							
VITAMIN A SUPPLEMENTATION							
Capsules starting stock		Number of 100,000iu Capsules Received		Capsules Used		Balance	
100,000iu	200,000iu			100,000iu	200,000iu	100,000iu	200,000iu
		Number of 200,000iu Capsules Received		Doses Administered			
				6-11mths	12-59mths	6-59mths	Post partum
COMMUNITY-BASED GROWTH PROMOTION							
Indicator				Male		Female	
Children <2 years registered this period							
Children <2 years currently in registers							
Children <2 years weighed this month							
Inadequate weight this month							
Inadequate weight two consecutive months							
ANAEMIA AMONG PRE-SCHOOL CHILDREN							
Age(mths)	Total Cases		Admissions Due to Anaemia		Deaths Due to Anaemia		
	Male	Female	Male	Female	Male	Female	
0-11							
12-23							
24-59							
Total							
SCHOOL FEEDING PROGRAMME				SUPPLEMENTARY FEEDING			
Children Enrolled				(2-5) Enrolled			
Children Fed				(2-5) Weighed			
Underweight children				Gained Weight			
Children with normal BMI				Below -2 SD			
Overweight children							
Obese children							
NUTRITIONAL REHABILITATION - CMAM							
Indicators				IN-PATIENT		OUT-PATIENT	
Cases at the beginning of the month							
Total SAM admissions							
Total discharges							
Cured							
Dead							
Defaulted							
Non-recovered							
NUTRITION REHABILITATION - NACS							
Indicators				SAM		MAM	
Cases at the beginning of the month							
Total admissions							
Total Discharged							
Recovery (SAM-MAM)							
Recovery (MAM-NORM)							
Cured							
Dead							
Defaulted							
Non-recovered							
ANAEMIA DURING PREGNANCY							
Total No.				No. with Anaemia			
New Pregnancy Registrants checked							
Pregnancies 36 weeks old checked							
Other Pregnancies							
Total							
Officer In-Charge							
Note: To be submitted no later than the 5th day of the following month to the District Director of Health Services.							