**CHAPTER THREE**

**PROJECT METHODOLOGY**

**3.1** **Chapter Introduction**

The implementation phase of this drug reminder mobile application involved decomposing the whole process into smaller bits and defining the relationship among the constituent bits. Top down coding approach was employed in the application software implementation using Java programming language. This involved dividing the implementation process into subunits or modules and each subunit being further divided into even smaller subs. This process of division is repeated until each unit is sufficiently small enough to be conveniently coded (implemented) from scratch as an independent entity that performs a clearly defined operation.

The analysis and comparison of existing mobile applications was done and this resulted from the review of related implemented methodologies. The critical analysis led to the adoption of the scheme of requirement specifications that highlighted the nature of the drug reminder mobile app implemented in this project work. The process followed in the implementation of the mobile is succinctly written in the following sections of this chapter.

**3.2 Architectural Design**

**3.2.1 Design Features of the System**

The technological approach for implementing the mobile drug reminder is based on an architecture which provides the necessary framework for the services, components and interfaces. At the back-end, the services include the use of android alarm, calendar and other vital services. The hardware and software components that were used provide logical relationships between applications, end-user services, and underlining back-end services. The database, the mobile android phone, and the developed application form the technological infrastructure for accessing, storing and managing patients’ drug information. From the diagram in Figure 3.1, when the user saves the drug and dosage information on the system, the information will be called from the servers whenever the alarm notification is to be sounded. After the choosing of the necessary dosage plan the information will be forwarded to the server. After this, the user would be alerted through the phone’s alarm system about the next drug to take may be the next appointment with the doctor is due.

Basically, the patient mostly interacts with the Google alarm and calendar apps that is already preinstalled on every Android device. The front end of the system is a mobile application interface which allows users create an account using their email address, then, using the mobile app, they can conveniently add their medication reminders, view their current medication list, add doctor’s appointments and view doctor’s appointments. This application then distributes the information supplied, to all the patient’s android devices synced with the email address and creates a reminder for each activity. Input into the system is the information entered by the patient which include login details, drug description details (such as name, dosage and time of use), and doctor’s appointment details (such as the date and time for the future appointments). The output of the medication reminder system are the medication reminder alert and the doctor’s appointment reminder.

Figure 3.1: Architectural design of the reminder mobile application.

**3.2.2 The System Modules**

This final year project work proposes to develop a healthcare application that provides reminder services to users so that they can adhere to their medication regiment. The user creates a new medication schedule by setting a schedule name and specifying the date and time for the first medication notification. Once a schedule is created, additional medication details, such as medicine name, duration of use and dosage can be added. Once the schedule and the medication details have been created, the user will be directed to the view schedule list. While viewing the schedule list, the user can choose to activate the schedule reminder immediately or at a later time.

When it is time for the medication alert, Google Alarm on the user’s devices will ring a notification for the medication or appointment. The Google Alarm notification will also display the medication details which include medicine name, dosage and the duration of use in both the notification bar and main screen of the mobile phone. The following are the modules of the mobile android based reminder application system.

**3.2.2.1 Login Module**

Figure 3.2 highlights the patient login module. In the beginning when the patient opens the app, he/she has to log in using their email address and the system automatically authenticates the user with their password. After login, the patient will be able to view homepage which contains the list of all the medications, appointments, profile and a log out option. Each option on the homepage can be accessed by the user, and modified as needed.

Figure 3.2 Patient login module

**3.2.2.2 Profile Module**

The profile module contains demographic and health information about the patient. Demographic information such as name, age, gender, phone number, address, email address, social media handles etc. are to be considered. Healthcare data like height, blood group, blood genotype etc. are to be captured.

**3.2.2.3 Medication Module**

This module allows the patient view his current medication list. On this screen, he/she can see at a glance all the drugs they are currently on, and allows for modifications and updates. Addition and deletion of medications can be made to the list. A new drug can be added to the list by inputting details about the drug such as the name of the drug, the dosage and how long it would be used.

**3.2.2.4 Appointment Module**

The appointment module contains information about all the patient’s doctor’s appointments. The dates and times of future appointments are displayed on the appointment screen. It also allows for modification and updates. The patient can enter new appointments and delete existing ones.

**3.3 System Requirements**

**3.3.1 Functional Requirements (Use Cases)**

A functional requirement defines a function of a software system or its component. The functional requirement specifies specific functionality that defines what a system is supposed to achieve. Each functional requirement has a unique reference ID which is defined as: Functional Requirement-Number.

* 1. FR-1 -The system shall enable users to login to their accounts.
  2. FR-2-The system shall be display patient’s profile.
  3. FR-3-The system shall enable the user to view medication list.
  4. FR-4-The system shall enable the user to view doctor appointments.
  5. FR-5- The system shall use the alarm module of the android services to notify the user of the upcoming dose of drug and doctor appointment.
  6. FR-5-The system shall enable the user to log out of the system.

The following show the description for the above use case scenarios. Each use case has unique identifier UC (use case) number. The description details how the each actor interacts with the system in order to use the services implemented. The five main scenarios which are identified according to the above functional requirements are described in details as below.

**Scenario 1**: Log In(Actor: Patient and System)

1. Patient opens the system.

2. Patient can access the system using their email address.

3. The patient waits for the home page.

**Scenario 2**: Display Patient Profile(Actor: System)

1. The system displays patient’s information with home page.

2. Allows the patient edit their profile.

**Scenario 3**: View Medication List(Actor: Patient and System)

1. Patient can view their current medication list on the system.

2. Patient can also add a new medication to the list.

**Scenario 4**: View Doctor’s Appointments(Actor: Patient and System)

1. Patients can view their future doctor’s appointment date.

2. Patients can also add a new appointment to the list.

**Scenario 5:** Sign Out(Actor: Patient)

1. Patients closes the application by logging out.

**3.3.2 Non- functional Requirements**

Non-Functional requirements describe user visible aspects of the system that are not designated to the functional behavior of the system. Each non-functional requirement has a unique reference ID which is defined as: Non-functional Requirement-Number. The requirements include from user interface to security issues. Nonfunctional requirements of the system are described as follow:

**3.3.2.1 NFR- 1 Performance**

The Server must respond to user requests maximum in one minute unless the connection is interrupted. The system should also respond to user click maximum in five seconds. Since the system is going to be accessed by different users with different needs, it should be capable of handling and processing their queries quickly. Since the system is an online mobile app system, it is difficult to tell exactly how many users will be using the system at a time. However, the system should handle its users concurrently.

**3.3.2.2 NFR-2 User interface**

Since the system runs on a mobile phone for better management of the small screen, the right type and amount of interactive user interfaces shall be used.

**3.3.2.3 NFR-3 Security**

The system should be developed in a way that the reminder system is exposed to its users in a secure way. The system should be developed in a way by protecting the integrity of data that is exchanged between the system and its users. The system shall also accept only a valid data and not transmit to a third party for any purpose.

**3.3.2.4 NFR-4 Portability**

As the system primarily aimed to be accessed from mobile phones, which has accesses to internet connection, it should work on mobile devices be it tablets etc., so far they run on android OS.

**3.4 System Analysis Models**

**3.4.1 Use Case Diagrams**

The use case diagram represents the functionality (functional requirements) of the system from a user’s point of view. Use cases define the boundaries of the system. The following use cases detail the mobile drug reminder system for the Nigerian healthcare delivery industry.

Use Case tools are used for capturing system’s functional requirements. A use case is an approach used in system analysis to identify, simplify, and organize system requirements. Use case diagrams show how users will interact with the system. A use case diagram contains four components.

* The boundary, which defines the system of interest in relation to the world around it.
* The actors, usually individuals involved with the system defined according to their roles.
* The use cases, are the specific roles played by the actors within and around the system.
* The relationships between and among the actors and the use cases.

For this proposed mobile reminder application, actors involved as described in the previous sub section are patients and system as shown in figure 3.3a , 3.3b. The other uses case elements are shown in figure 3.4. Figure 3.5 shows the use case diagram for the reminder system.



Figure 3.3a Actor 1: The Patient Figure 3.3b Actor 2: The System

Figure 3.4 The other use case elements



Figure 3.5: Use case diagram of the reminder mobile app

Table 3.1 Database items and their data types

|  |  |  |
| --- | --- | --- |
| **Field name** | **Field type** | **Width** |
| Nature of the drug,  Start time for a drug dosage  end time for a drug dosage  User category i\_d  Adults (age 12+)  Children (2-11)  Infants (under 2)  Title, | Character  Integer  Integer  Integer  Character  Character  Character  Character | 11  4  4  3  3  10  10  4 |
| First name,  Last name,  Mobile phone,  Contact no,  Flight no,  Seat preference  Email address  Confirm email address | Character  Character  Character  Character  Character  Character  Character  Character | 20  11  35  10  3  25  25  25 |

**3.4.3 Class Diagram**

A class diagram describes the structure of a system in terms of classes, their attributes, operations (also called methods) and the relationships among the classes. In this section, the identified classes of in the project are presented in Figure 3.7.

**3.4.4 Activity Diagram**

Activity diagram is a graphical representation of work flows of stepwise activities and actions with support for choice, iteration and concurrency. In this section, activity diagrams shown in figure 3.6 is used to describe the work flow of use cases identified in Section 3.4.2. Figure 3.8 shows the activities of the user of the system navigating through the drug reminder and doctor’s appointment reminder functionalities if the system. Figure 3.9 shows the activities and inter relationships between the components of the reminder system.

**3.5 Database Design**

The data is simply stored on the server. The system stores records of nearly all of the information it is given, and all conclusions that it derives on a permanent database on the server. Although it does not store model parameters, as the model with which it will work varies and so will the number and meaning of the parameters. The rest is stored in order to preserve as much data as possible for further research; in a database system dedicated to data collection, preservation of all data that can possibly be useful is an ideal feature. The model needs to have access to the parameters for it, as determined bythe rest of the system, at the time database is run. Rather than placing the values in a file with Java, the language in which this program was written and then having database open and read that file, the values are simply inserted into the database file itself. This saves a lot of work, and is very little additional work for the Java module, as good processing abilities are a strength of Java. The method of database design chosen for this particular mobile app is the top down file design. This is so that each module could be removed or added without necessarily affecting the entire problem.

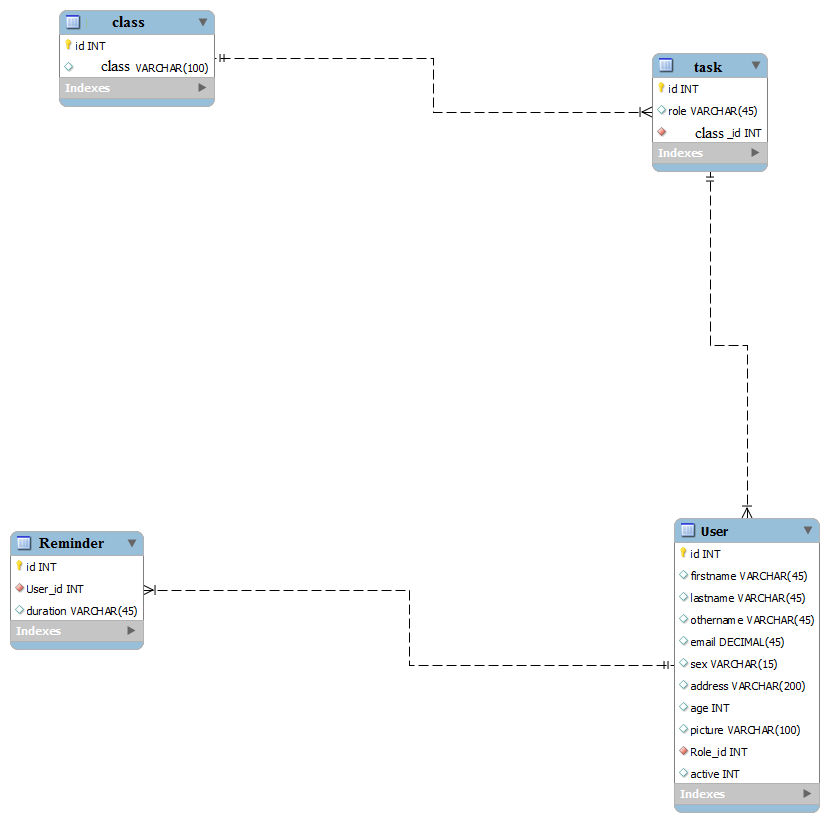


Figure 3.6 Entity relational diagram of the system

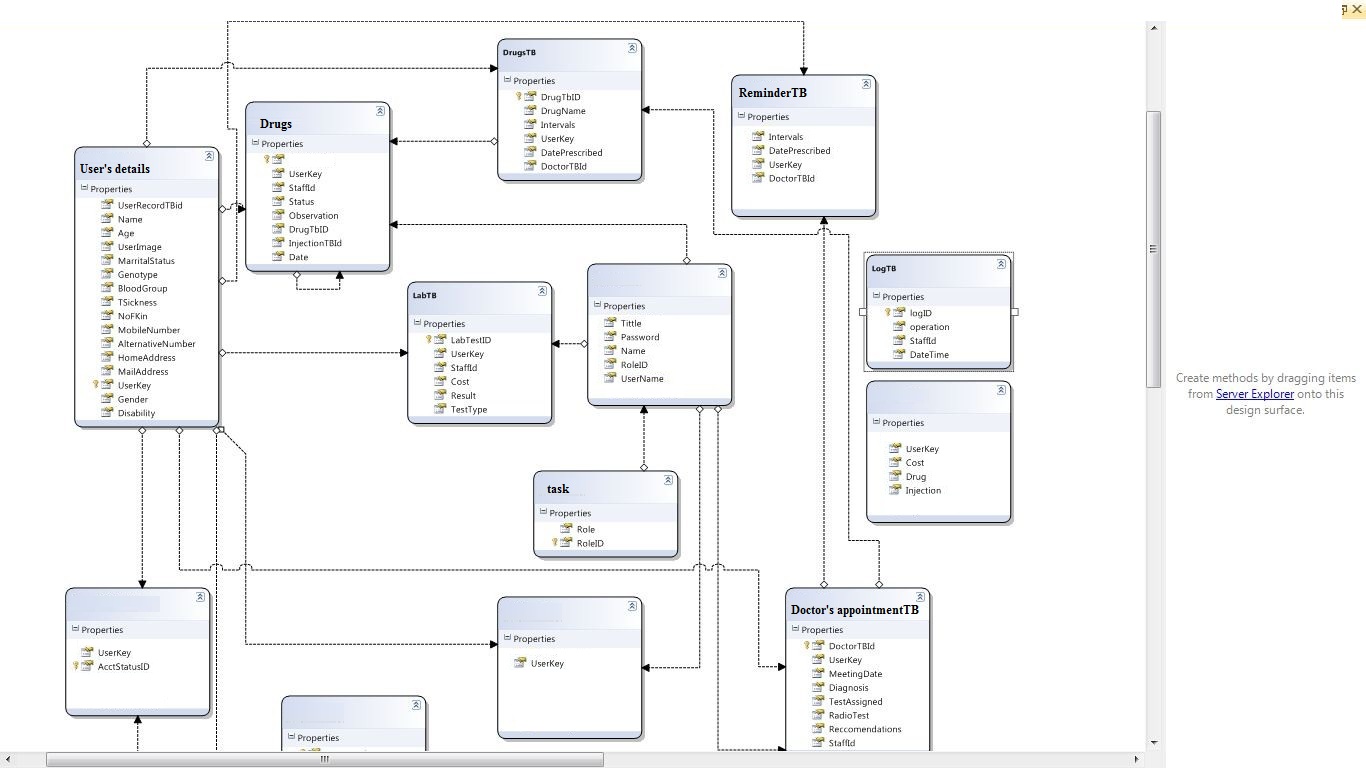


Figure 3.7 Diagram showing the database schema for the reminder system

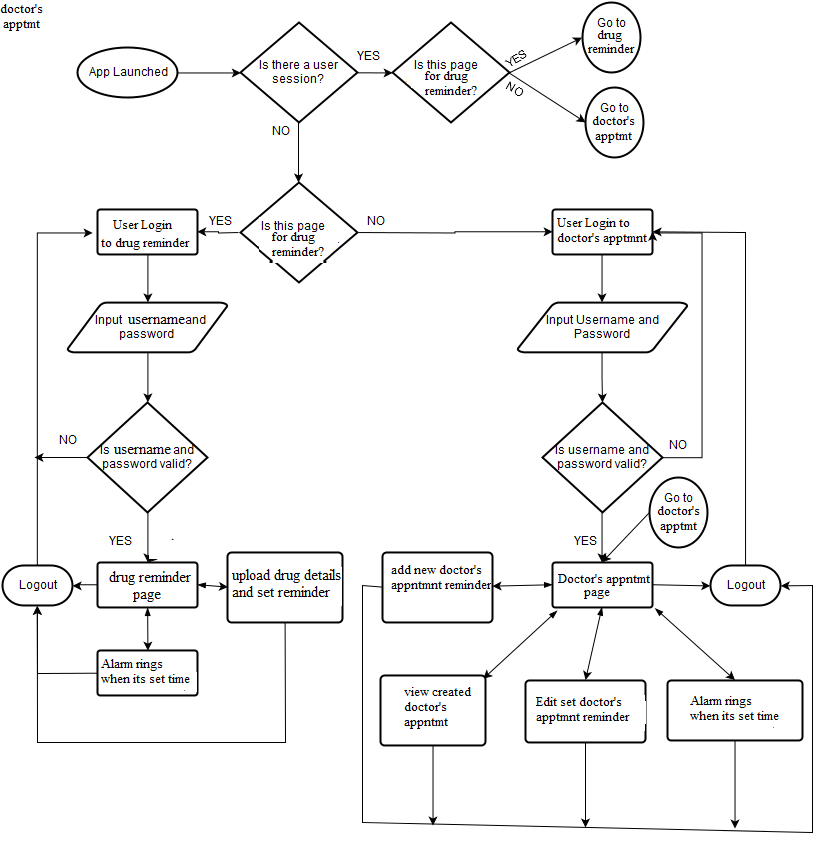


Figure 3.8 Users Activity diagram

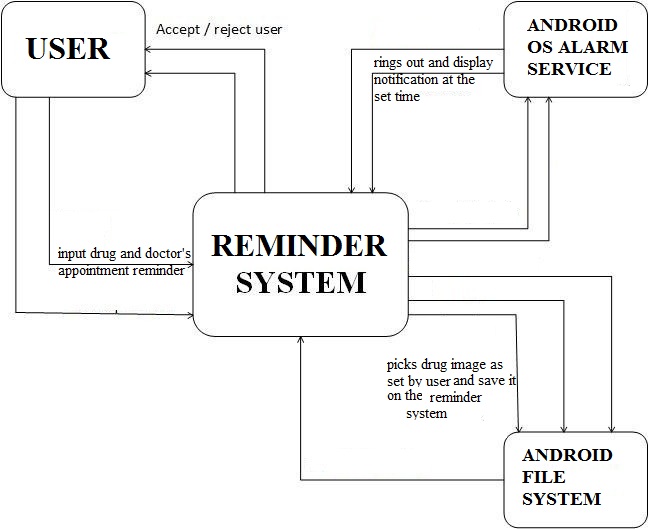


Figure 3.9 Flow of activity within the components of the reminder system

A database with the description and properties shown in table 3.1 was used. The design of the new system was carefully drawn on paper considering the old existing systems’ limitations. Flowcharts were used as an effective graphical representation of the program, as well as a design tool and it is aided in the evaluation of a logical correct flow of data and process. Figures 3.10 to 3.13 shows the design flowcharts for the mobile application; consisting of procedure flow, files layout flow, data entry input validity flowchart, and database management flowchart.

The database is a very important part in developing a mobile app. The database used in this system was developed using SQLite database, this stores the ailments’ symptoms, patients’ diagnosis, prescription, patients’ details etc. SQLite is an open source database management system. It stores data in its own format based on access jet database engine. Database design includes the creation of tables used in the system database. These tables are used to store the information about each entities shown in figure 3.7

In the database design, the persistent management is presented as it is an abstract and conceptual representation of the data. Figure 3.7 represents the conceptual representation (schema) of the database the project uses. In the diagram, the notation represents entities as boxes, and relationships as lines between the boxes. Different shapes at the ends of these lines represent the cardinality of the relationship (i.e. one or many).

START

ACCEPT INPUT

DATA

CHECT FOR CORRECTION

SELECT MENU OPTION

PROCESSING

GENERATION OF

ERROR REPORT

STOP

Figure 3.10 Process Flowchart

START

INPUT MAIN

MENU OPTION

IS

1

=

1

?

IS

=

1

2

?

IS

1

=

3

?

IS

4

=

1

?

WRONG CHOICE TRY AGAIN

DO DATA

ENTRY

INPUT VALIDATION CHECK

DO FILE

MAINTENANCE

STOP

Figure 3.11 Input (for Text and Drug image) validation flowchart

START

SELECT THE ORDER TO

DISPLAY THE DRUGS

A

IF

A = 1

IF

A = 2

IF

A = 3

IF

A = 4

DISPLAY

DRUGS

DISPLAY DRUGS BY

DOSAGE

STOP

ELSE

DISPLAY DRUGS BY

TIMELINE

DISPLAY DOCTOR APPOINTEMENT

DISPLAY ALL REMINDERS

AVAILABLE

THEN

Figure 3.12 Display Layout of system files Flowchart

START

PRINT “SELECT

CREATE DATA

FILE

IF

A = 1

IF

A = 2

APPEND RECORDS

IF

A = 3

IF

A = 4

ELSE

STOP

UPDATE RECORDS

DELETE RECORDS

Figure 3.13 Database Management Flowchart

**3.6 Procedure Phases and Time Frame**

The implementation of the reminder system project was divided into the following phases:Requirement phase, Analysis Phase, Design phase, Implementation and testing phase.

* Requirement phase: here, a complete description of the behavior of the system was done. The interaction users would have with the software was clearly mapped out for clear development (coding).
* Analysis Phase: Formal enquiry was carried out in order to identify a better course of action to develop the system, thoughts were shared, and online resources were perused for clear information.
* Design phase: The Graphical User Interface for the mobile application was developed. This was done using the Java programming language.
* Implementation Phase: This is the part of the process where the software engineering of the expert system was actually done. The coding from scratch using Java on android studio and SQLite. Scripts for different modules were written and fully tested okay. This phase is in progress.

The project was completed in a six month timeline (first and second semesters). It is divided into two segments completed in each semester. The first segment is the gathering of relevant information and other resources for the successful completion of the project, this entailed the preparation of the chapter one (introduction) and two (review of literatures on past related works) of the project report. The second segment involved the actual implementation and discussion of the results attained.