Global Scientific JOURNALS

KARATINA UNIVERSITY

SCHOOL OF PURE AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE AND

INFORMATICS

GPS BASED HOSPITAL LOCATION ANDROID APPLICATION.

OWUOR VERONICA AOKO

P100/1206G/16

PROJECT REPORT SUBMITTED TO THE SCHOOL OF PURE AND APPLIED

SCIENCES IN PARTIAL FULFILMENT FOR THE AWARD OF DEGREE IN

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY IN KARATINA

UNIVERSITY

OCTOBER, 2020

DECLARATION

I declare that this project is my original work except for parts quoted and cited as references and that it has not presented by any other person as a project in Karatina university.

Name…………………………………………………………………………

Signature………………………………………………………………………

Date………………………………………………………………………..

Id no…………………………………………………………………………..

SUPERVISOR

I hereby certify that that this is a true report for the project undertaken by the above named student under my supervision and that it has been submitted to Karatina University under my supervision.

Name…………………………………………………………………………

Signature………………………………………………………………………

Date………………………………………………………………………..

ACKNOWLEDGEMENT.

i

LIST OF ABBREVIATIONS.

AADS- Automated Ambulance dispatch system.

A- Administrator.

P- Patient.

Hos- Hospital.

ABSTRACT.

Health Scan application is a GPS based application aimed at helping people locate nearest hospitals to them. This follows the difficulty to access medical help in time when they are in unknown locations. As a result, this project will enable one to locate convenient hospitals using their mobile phones. As a student, my involvement in this project was to make sure that I come up with innovative ways of using technology to the betterment of people's lives, improving my confidence I problem solving and to utilize opportunities to relate to different categories of people likely to meet in real life situations. This report contains seven chapters. Chapter one is the introduction that commences the project, gives objectives, assumptions, budget and schedule of the project. Chapter two is the literature review and it discusses previous research done and published by other authors. Chapter three discusses the methodologies used n the development of this android application. Chapter four discusses the analysis of both functional and non-functional requirements while chapter five discusses the design methodologies for the android applications. Chapter six discusses the different programming languages used in implementation of the application, testing and changeover techniques. Chapter seven summarizes the project report and gives proper recommendations towards future improvement of the application.

Table of Contents

iii

iv

List of Tables

List of figures

CHAPTER ONE: INTRODUCTION

1.1 Introduction

When everyone hears the word 'ambulance' the first thing that comes to mind is the rescue process.in modern era people feel uncomfortable due to danger aspects of accidents, some known and unknown emergencies that require quick treatment because patients are unable to call for help therefore leading to loss in lives. The person in emergency is often unaware of the nearest hospital available to them resulting to delays in medical services.

Emergencies such as accidents require medical attentions where patients need to be transported from the place of incident to the hospital. The importance of taking a patient to the hospital can be judged by the fact if the arrival of the ambulance is delayed, it can worsen the patients' medical state and even cause death. The delays can be caused by the time taken in dialing the emergency number or the inability to locate the nearest hospital. The objective of this research was to minimize the time consumed for arrival of the ambulance by allowing patients to call for the ambulances themselves. If the emergency is not as serious such that it requires an ambulance, a patient was to take himself to the hospital.

1.2 Background of the study

A global positioning system works in such a way that it processes the distance between the receiver and four satellites to establish the receiver's geographic location. GPS technology can be used by civilians in figuring out their location while hiking or obtain driving directions. GPS is also used to locate and dispatch police cars, ambulances and other emergency vehicles. GPS was first created by the united states department of defense and was exclusively used by the united states military but now civilians are allowed to use the GPS and can be found in standard cars and cell phones. GPS is improving as the technology evolves. There also exists different types of GPS receivers as listed below:

1

1.   Car mounted GPS receivers

These devices are used among civilians to help them navigate directions while driving. While mounted GPS receivers can be purchased separately to be used with any car, almost all vehicles today offer an upgrade to include GPS built in the car.

2.   Hand held GPS receiver

These devices are commonly used amongst individuals to determine their geographical location while hiking.it can also be used by land surveyors, farmers or fishermen on their day to day jobs.

3.   Mobile phones integrated GPS receivers

These are the most common receivers mounted on every available smart phone. This allows easy access to GPS maps at any time without having to purchase an additional unit. They are commonly used in finding someone's location or finding a certain geographical feature be it natural or man-made using the maps superimposed on the maps from their phones.

The first ambulance dispatch system was a manual paper trail process that is, the 911 operator would take in the emergency call, write down the necessary information from the caller and manually locate a free ambulance. Without having an automated computer system there was no guarantee that the operator will locate the nearest ambulance in time. There is also a system that that allocated and mobilized suitable ambulances within minutes but still through the 911 calling. The system showed the location of the nearest ambulances from the hospitals.

2

1.3 Problem Statement(s)

The GPS based hospital location android application was developed as a result of the inability to efficiently and effectively locate hospitals by the patients in distress whenever they are in an unknown location. According to a study conducted in 2009 by the world health organization, response time to places of injuries takes long because patients do not have a way of calling for help which leads to loss of lives. In addition, the patient in distress was often unaware of the nearest hospital available to them especially when found in a new environment   leading to delays in medical services. The proposed android application provided patients in unknown locations or rural areas with capabilities to access medical services by allowing their phones' GPS to pick the nearest hospital and send a distress alert so that an ambulance can be sent to them.

1.4 Objectives

1.4.1 Main objective

The main aim of this research was to develop a GPS based hospital location android application that locates the nearest hospital from the place of injury.

1.4.2 Specific objectives

  i.   To investigate and identify the gap in the existing application in response to patient emergency response.

  ii.   To design and develop a platform where patients can send alerts to different hospitals.

  iii. To implement the google map module that generates nearest hospitals and their distances from places of distress

  iv.   To test the applications interactions with users.

3

1.5 Scope and Limitation of the study

The study aimed at understanding the working of a computer aided ambulance dispatch system in Nyeri county. This study was limited to only district and referral hospitals. It excluded smaller hospitals like clinics and health centers. The reason why the study was limited to district and referral hospitals is because smaller hospitals were not able to access as many ambulances that are needed to dedicate to the dispatching job alone.

1.6 Justification

During this century, its known that at least sixty percent of people own smartphones and are able to access the internet. The developed android application ensured that the nearest possible hospital was allocated and the distressed caller received medical help within the shortest time possible. Users had to properly sign up with their mobile numbers or email address for authentication so that irrelevant persons are not able to use this application for no specific reason. In case of an emergency in an unknown location, the user was to use the developed application to search for the nearest hospital and go for medical care or send an alert to the hospital's emergency centers where the request was to be processed and the ambulance sent immediately. The victim or the people close to the victim were to select the type of emergency if they have decided to call for an ambulance so that the hospital was able to equip the ambulance effectively.

1.7 Project Risk and Mitigation

If a project's success variables were not spelt out or should they be underestimated, the following project risks were encountered.

- There was a risk that the physical storage space for the required application may not be enough. This was avoided by allocating enough space to the application being developed and specifying minimum storage requirements before installation.

4

- There was a risk that data rates required to access the internet will be higher and exceed speculated capacity. This was minimized by allowing an offline access to the application

1.8 Assumptions.

During the course of this research, one needed to consider certain assumptions before understanding the emergency application. The android application was based on the following assumptions;

- The victim must have had an android based smartphone where the android application will be installed.

- The victim must have had internet connection with GPS activated which will help show the hospitals location from the user's phone.

- The victim was not be unconscious during the medical emergency and was in a position to press the send alert button or take himself to the GPS located hospital.

- In case the victim was unconscious, the victim's caretaker was to press the emergency button or take him to the nearest GPS located hospital.

1.9 Budget and Resources

EXPENDITURE	COST
Software	2000
Hardware	27000
Telecommunication	1800
Travel	6500
Accommodation	10000
Project specification training	3000
Support services	1500
Consultancy	6000
Contingency	4000
TOTALS	**61800**

Table 1.0: Project budget and resources

1.10 Project Schedule

This project took an estimated period of eleven weeks from January to April as represented by the Gantt chart. Every activity in the development process was documented as represented in the following figure.

6

Weekly activity	1	2	3	4	5	6	7	8	9	10	11
Project proposal	▦										
Feasibility study		▦									
Requirement analysis			▦								
System design				▦							
User Interface design					▦						
Coding						▦	▦	▦	▦	▦	
Integration and testing											▦
Documentation	▦	▦	▦	▦	▦	▦	▦	▦	▦	▦	▦

Table 1.1: project schedule.

7

8

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter presents a review of related literature on the subject under study, that is it presents literature on emergency medical services in general. GPS can be defined as a system that works in such a way that it processes the distance between the receiver and four satellites to establish someone's or something's exact geographical location.

2.2 Conceptualization of GPS ambulance systems and applications.

2.2.1 Smart Ambulance system.

(Ahmadi, 2017) studied on the literature of health care facility located and based on location management, they constructed a framework for different emergencies and non-emergency health facilities. (Vahid, 2016) said that the response time is the key factor in the reduction of irreversible injuries in accidents and use of technology can be considerably effective. This study agrees with the one done in Ota in 2000, in which the use of GPS and display maps in ambulances can effectively reduce both distance travelled and time taken. The GPS system employed in Tehran pre hospital emergency was only used in guiding systems for showing the most appropriate and nearest pathway for reaching the event location.

2.2.2 Ambulance surveillance system.

According to (Lam, 2015), the response time of ambulances and considered the risk factors that can be caused by delays. According to the real data about the incidents in Singapore, they proposed their model and defined the period from the moment that an ambulance is dispatched from the hospital or it's station to the location of the patient until departing the hospital after delivering the patient to the hospital. (Lam, 2015) also examined that the underlying risk factors that can possibly affect emergency response time for the purpose of delivering intentional measures that can shorten response time .They revealed that

9

the most significant factor affecting emergency response time was traffic followed by location of then scene and then weather.

(Atack et al, 2010) conducted a study to gain in-depth viewpoints of highly experienced emergency medical service practitioners, educators, administrators and physicians on major issues pertaining to the patient's safety. It indicated that emergency medical services are not well aligned with the increasing demand of health sector therefore change is needed in emergency medical services education in order to develop stronger clinical decision-making skills.

2.2.3 Emergency call for android.

A research by (Peter, 2008) proposed a system in which they used GPS to track the location of the victim to send a message in case of an accident so that the victim can be treated as soon as possible. According to the paper Enhanced Functionality Emergency Call For Android, the idea was to create an application for android which will use GPS embedded in smartphones to send their preregistered contacts by pressing a single button in case of an emergency.in the paper GSM and GPS based system by (Bennet, 2005), the system was for automatic accident notification and severity estimation. Their main objective to set up sensors inside the car to detect any dangerous situations ant then report it to the nearest control unit.

A study by (Saini, 2007) surveyed thirty-nine paramedic students and practicing emergency medical services personnel and found that electronic decision support could provide good accessibility and ease the use of emergency medical service resources. According to (Forny, 2011), dispatchers' decision was inappropriate for thirty percent of emergency service users and there was a substantial delay in time to reperfusion therapy, therefore accuracy of telephone triage was to be improved to mobilize emergency medical service response.

10

According to another study done by (Andersson et al, 2007), the ambulance relocation and dispatching system aimed at determining the location of the available as well as having the best dispatching policy where a set of pre-assigning list was used for each demand zone and were ordered according to the priority of the dispatching. If an emergency arises in a zone, the first idle ambulance on the list is dispatched to the call. If no ambulance is idle, the call was queued or redirected to another emergency agency.

2.2.4 Smartphone based hospital tracker.

In the paper Smartphone Based Enhancement in Health Service using GPS by (Alan et al, 1991), they proposed a system in which they used GPS to track the location of the victim so that the victim could be taken to the hospital as soon as possible. According to (Hirschberg et al, 2003) utilization of computer simulation technologies in their research to establish simulation models and they found that variables like number of injuries and rescue manpower planning should be considered.

Another designed system tracked the current ambulance with the help of GPS module. It sensed the parameters of the patient.it also sent the patients information to the hospital's servers. It also sent the position of the ambulance to traffic signals to control the trafficking and based on a research by (Shival, 2001) said that the traffic signals could automatically control the traffic to clear the paths for the ambulance ;traffic signals would turn red in all paths except the route where the ambulance was passing.

2.3 Conclusion.

The above systems and application were useful but lacked some capabilities. For instance, the ambulance surveillance system could only work in Singapore. Moreover, the Smartphone based tracker android version 6 and above only, therefore, only users with android versions above that could use the applications. Therefore, the developed android application was used to track nearest hospitals from place of emergency. The main aim of this project was to

11

ensure that patients reach the hospital within the shortest time possible for treatment. The GPS technology was used so that the victim or the one close to the victim can take immediate action which helps reduce severity. This developed application allowed victims in unknown locations to find hospitals and to specify their type of emergency. Moreover, it is not limited to geographical area or the version of the operating system.

CHAPTER THREE: RESEARCH METHODOLOGY

3.1 Introduction

Research methodology is a collection of methods, practices, procedures and rules that are used in carrying out research work in the field. This part of the research describes the procedures and methodologies that were employed in this study. It gives details about the site, the study design, methods and tools of data collection, as well as the manner in which the data was collected. It also gives details on the tools in the implementation and testing of the android application.

3.2 Approach

Iterative and incremental method

Since every model has a weakness and drawback even any new model launched today will have, I weighted the options and opted to use the iterative and incremental method which starts with the initial planning and ends with deployment with the cyclic interaction in between. Basically, the idea here was to develop a system through repeated cycles and in smaller a time. Doing it this way allowed me to develop a system based on the experience learned in earlier parts of the system. It consists of mini waterfall
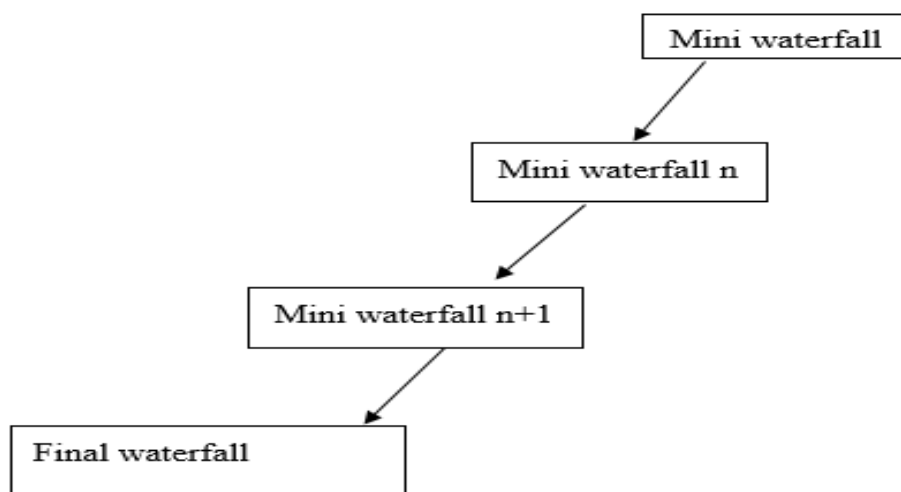
Figure 3.0: Iteration and incremental

13

Usage:

This method can be used in shrink-wrap application and large system which built in small phases. Also, can be used in systems that have separated components. It ensures:

    i.   Better use of scarce resources

   ii.   Can accommodate some change requests between increments

  iii.   More focused on customer value than linear approaches

  iv.   Problems can be detected at earlier stages

Drawbacks:

    i.   Requires heavy documentation

   ii.   Requires more customer involvement than linear approaches

  iii.   Partitioning functions and features might be problematic

  iv.   Follows a defined set of processes

3.3 designing tools

- HTML language
  - The hypertext markup language will be used for creating the basic structure of the web view part of the application.

- Java script.
  - Java script will be used to add or increase interactivity of the application.

- Java

  Java will be used to create the main android application and the standard features that should be contained in the application.

14

3.4 current application vs proposed application.

3.4.1 Current application

1.Automated ambulance dispatch system via one click smartphone application.

The AADS was developed by V. Khalique, S. Shsaikh, M. Dass, S.M Shah and M. Zaib (2017) from the Mehran University of Engineering and technology in Pakistan. During occurrence of an accident, the victim or the person lose to the victim presses the HELP button in the AADS smartphone application. This generates and sends the co-ordinates of the geographical location along with the victim's identity to the AADS server. This information is mandatory to identify and locate the requester. The AADS server stores the victim's information in the database table for maintaining the patients' records. The nearest vacant ambulance is located so that the AADS server forwards the victims location to the ambulance drivers. It can also enable the driver to accept or reject the emergency alert.

Disadvantages of AADS smartphone application.

  i.   The victim is not aware of the hospital that is closest to him. All the patient had to do was to call for help from any available hospital.

 ii.   The ambulance driver had to wait for forwarded information before accessing the patients' location which takes a longer period of time.

iii.   The user could not cancel the request in case he or she was aided before the arrival of the ambulance.

2. Kenya Red cross application.

The International Committee of the Red Cross (ICRC) is a private humanitarian institution founded in 1863 in Geneva, Switzerland, in particular by Henry Dunant and Gustave Moynier. It was 25-member committee that had a unique authority under international humanitarian law to protect the life and dignity of the victims of international

15

and internal armed conflicts. The ICRC was awarded the Nobel Peace Prize on three occasions that is, in 1917, 1944 and 1963.

The International Federation of Red Cross and Red Crescent Societies (IFRC) was founded in 1919 and today it coordinates activities between the 190 National Red Cross and Red Crescent Societies within the Movement. On an international level, the Federation leads and organizes, in close cooperation with the National Societies, relief assistance missions responding to large-scale emergencies. The International Federation Secretariat is based in Geneva, Switzerland.

The Kenya red cross organization is a non-governmental organization that provides helps in times of accidents, floods, drought and fires. Its android application is used to provide news about the emergencies. The disadvantage is that it does not provide the victim with the ability to see the nearest hospital or red cross Centre and that annual fees are expensive for both individual persons and families. The red cross application was launched in Kenya in 2016 and it allowed users to receive timely and lifesaving information during emergencies.

3.4.2 Proposed application.

Emergency hospital locator (Health Scan).

This application used GPS to locate the nearest hospital from the place of injury. Patients were able to either call for help or drive themselves to the hospital depending on the intensity of the emergency. The hospital locator was able to store the patients' previous medical history that can be viewed by the hospital servers. There was no need for ambulance drivers to wait for forwarded information because they were able to stay logged in using the same application so that whenever there is an emergency, they will get alerted via the same application. Cancelation of the request will be permitted in case the victim will be able to get help before the arrival of the ambulance.

3.5 Recommendations from the Study.

Issues of medical care is a critical issue when it comes to patients' location during the time of emergency. Thus, from the above findings, the study proposes the findings be recommended in utilizing the proposed applications in the hospital's emergency systems, adopting and practicing the use of technology in ambulance dispatch operations.

3.6 Conclusion.

The emergency ambulance GPS based application may lead to better route information about the hospital's location. The application will guide an emergency patient to nearest and appropriate hospital and also inform the patent of the services offered at each nearest hospital so that the patent will know which hospital to go to or request for an ambulance from.

CHAPTER FOUR: SYSTEM ANALYSIS AND REQUIREMENT MODELLING

4.1 Introduction.

This chapter describes the system study, analysis, strengths and weaknesses of the current application, context level diagrams, entity relation diagrams and the architectural design of the application.

4.2 Application design

Since the application is supposed to be used in all county and district hospitals in the country, my study was carried out using the red cross application which is currently the most used in the country. The main purpose of the study was to find out how the application worked and the mechanisms used to handle emergency services. The current application is considered ineffective in some areas because it only responds to major country wide crises

17

like droughts, fires or floods at the same time failing to consider individual emergencies. If individual emergencies are considered, then the patient already have a membership to the red cross community and the patient has to pay for the red cross services which are really expensive. Patients in unaware of the nearest hospital or do not have membership to the red cross organization may end up not receiving the services on time.

4.3 Application analysis.

During the study of the patients' emergencies, the requirements were categorized into user requirement, hospital requirements and the hardware requirements for the users.

4.4 Requirement specification.

Requirement analysis involves those tasks that determine the needs or the condition to be met for a new product, taking into account the possible conflicting requirements of various stakeholders. It also involves understanding the needs of the users and the behavior of the system to be developed using cases that describe all the interactions that users will have with the system. After the analysis of the collected data, the requirements were then formulated into user requirements, functional requirements, non-functional requirements and system requirements.
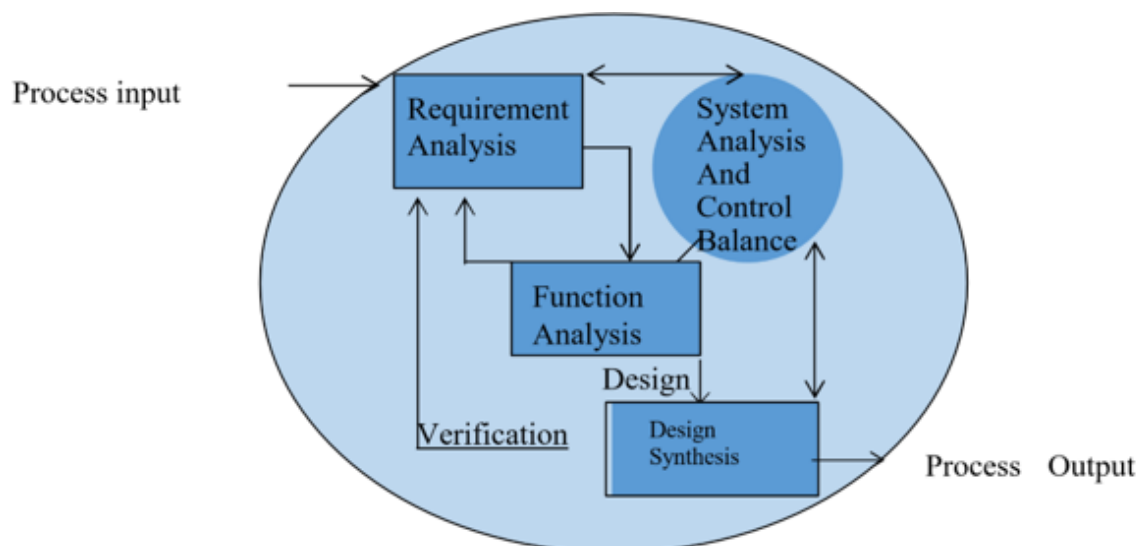
Figure 4.0: System specification diagram

4.4.1 User requirements.

For the developed application to be used effectively and efficiently, it is important that the users in this case the patients to be fully involved in the development process and are given the opportunity to participate inform of questions and comments to ensure that the human system interaction is at maximum. It also ensures that patients get accustomed to the new way of calling for medical help and finding hospitals in unknown locations in a faster way. During data collection, the investigation found out how the current application works and tried out which problems are faced and the best way to solve them. The user described some of their requirements as follows:

- Faster search for hospitals.

- Easier viewing of list of favorable hospitals, that is arranged from closest to furthest.

- Visibility of route from emergency places to hospitals.

4.4.2 Functional requirements.

Functional requirements are conditions that the system objects satisfy, fulfil or comply with. The desired application had the following functionality requirement:

19

- The application should provide shortest routes possible in case of existence of several routes to save time.

- The application should generate ten of the closest hospitals to provide patients with options.

- The application should authenticate the patients.

4.4.3 Non-functional requirements.

Non-functional requirements are requirements that specify the criteria that can be used to judge the operation of a system. The application should have the following non-functional requirements:

- The application should verify and validate all user inputs.

- All users must be notified in case error are detected in the course of using the application.

- The application should allow room for expansion.

- The application should have a higher performance and reliability level.

4.4.4 Application requirements.

This section describes the hardware components and the software requirements needed for effective development and efficient running of the application.

HARDWARE	MINIMUM SYSTEM REQUIREMENT
Processor	2.4 GHz processor speed and above.
Memory	4GB RAM and above.
Disk space	250GB hard disk space and above
Display	800X600 colors (1024X768 high color) 16-bit recommended

Table 4.0: Hardware requirements.

20

SOFTWARE	MINIMUM SYSTEM REQUIREMENTS.
Operating system	Windows 7 or later. Android version 4.1 or later
Database	Xampp
Runtime environment	Android studio IDE.

Table 4.1: Software requirements

4.5 Applications flow of activities.

The diagram figure 3 below shows the basic flow of activities in the developed application. During the interaction with this application, the user views the google map interface, clicks on the detail button to view the list of hospitals available to them. The patient can then send an alert by attaching a screenshot of their location or call the number available on the details interface. Ambulance attendants are able on standby for emergency alerts and to see the distance of emergency, through the database of the hospital.

21
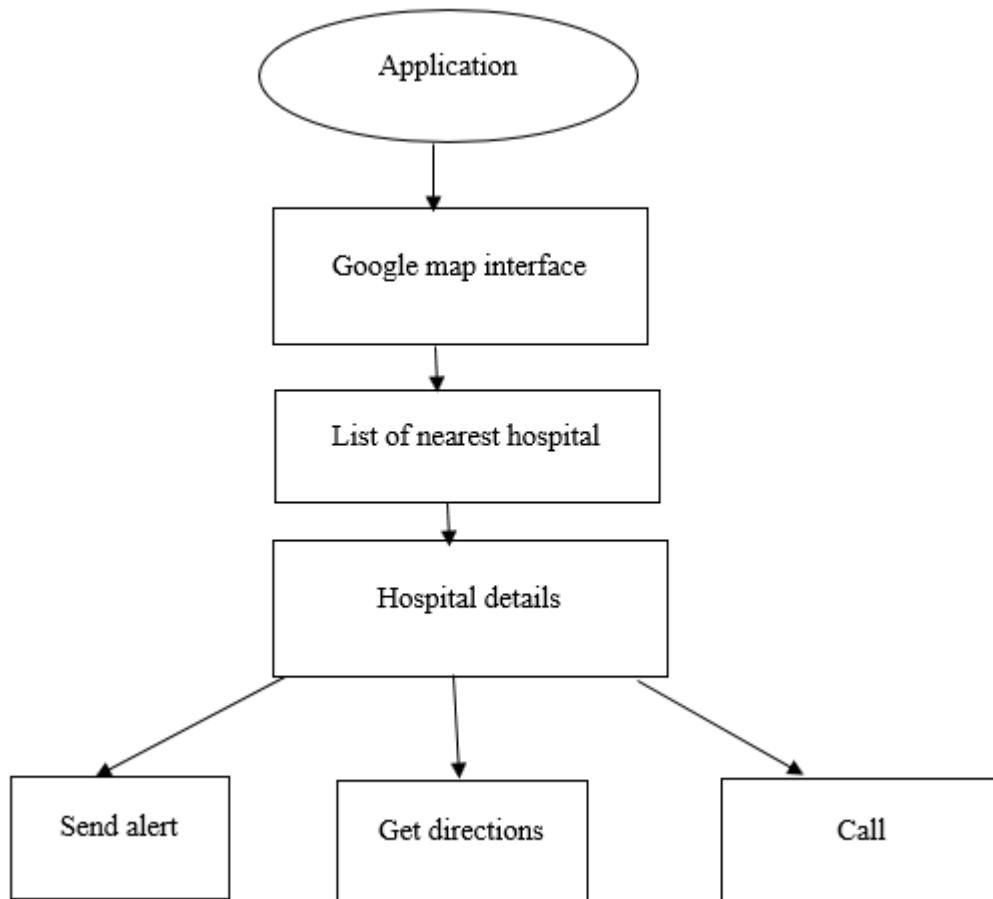
Figure 4.1: Application flow of activities.

4.5.1 Application architecture.

Application architecture gives a high leveled view of the new application with the main

component of the app, the services they provide and how the different interfaces

communicate. the application uses a three-tier architecture that comprises of the user

interface, process management and the emergency systems.

22

Front end

Process

Back end

Graphical user interface

User authentication

User registration

View information.

Edit profiles

Security strengths

Authenticity.

Confidentiality.

Data integrity.

Real time database.

Data.

Data integrity.
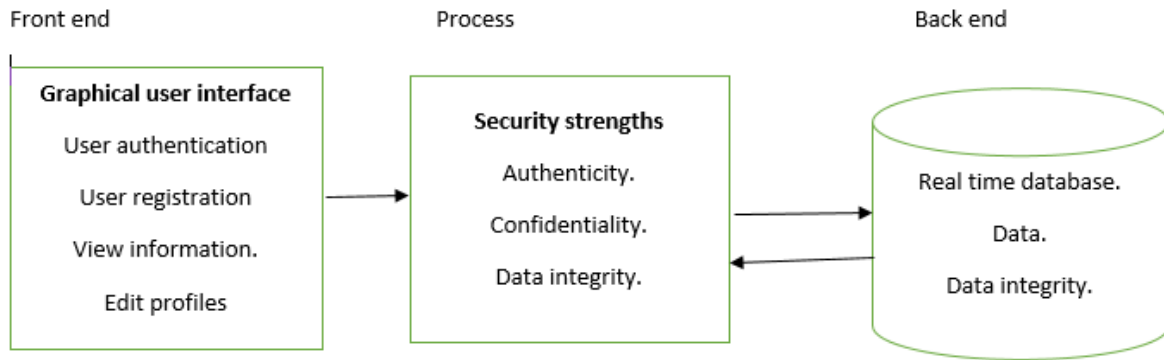
Figure 1.2: Application architecture.

Figure 4.3: Application architecture uml diagram

4.6 Module of the application.

The application will comprise of the following modules:

23

- Google map module.

- Log in and sign up modules.

- Send alert module.

4.6.1 Google map module.

This is a platform used by the patients to locate the nearest hospital to them. Through the

google map patient can perform the following tasks:

- View nearest hospital.

- Get route to these hospitals.

- Get a list of those nearest hospitals together with their emergency numbers.

24
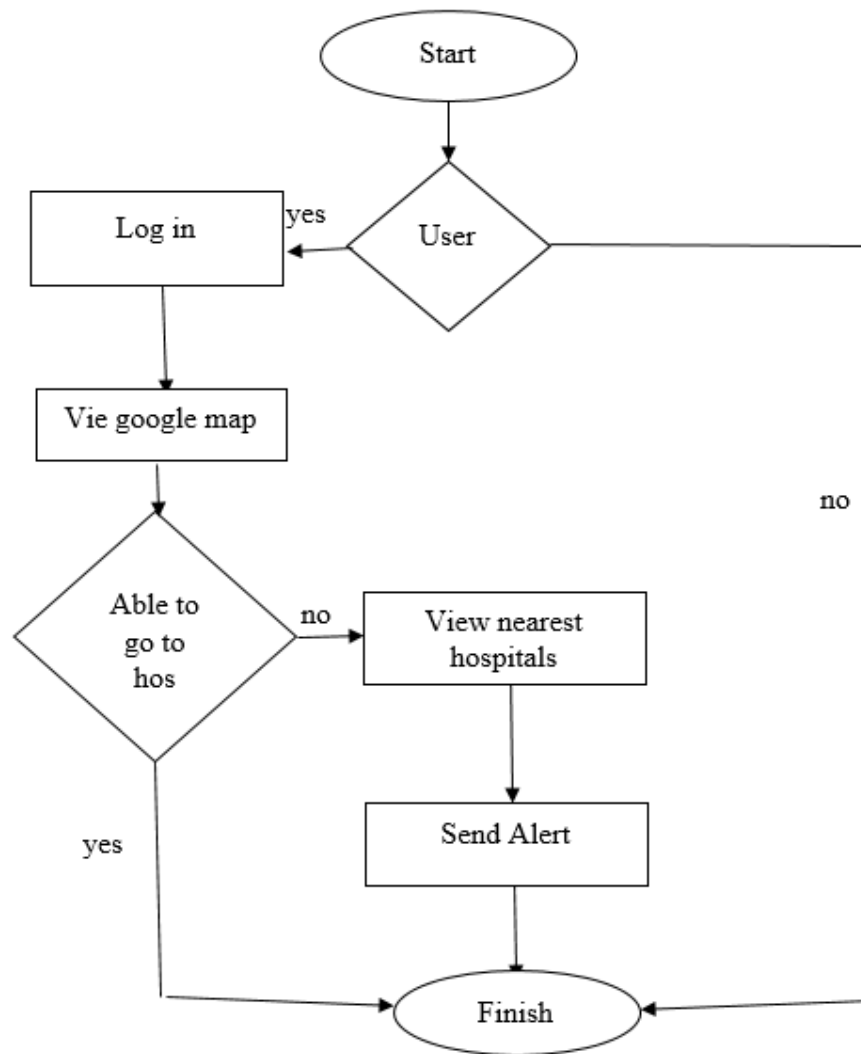
Figure 4.4: Google map module
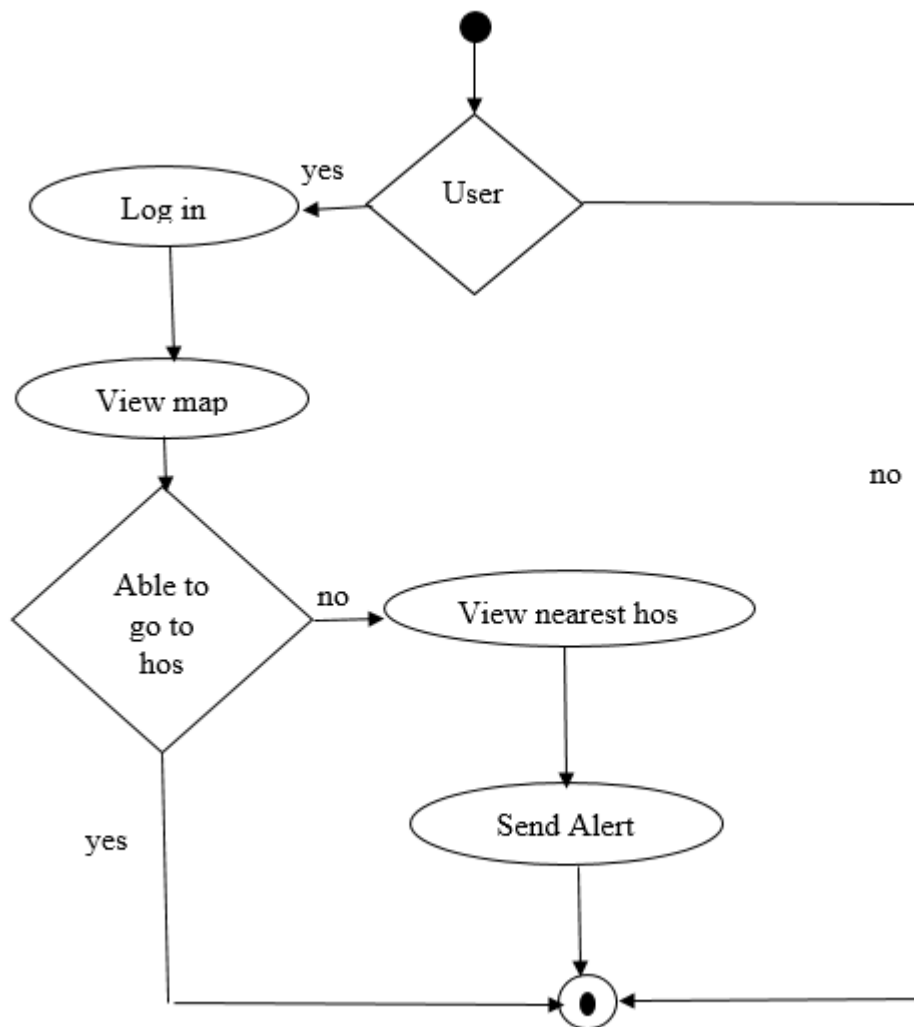
Figure 4.5: Google map activity diagram

4.6.2 Send alert module.

This module assists the patients to send their alerts to the specific chosen hospital

database so that they can receive help from the hospital.
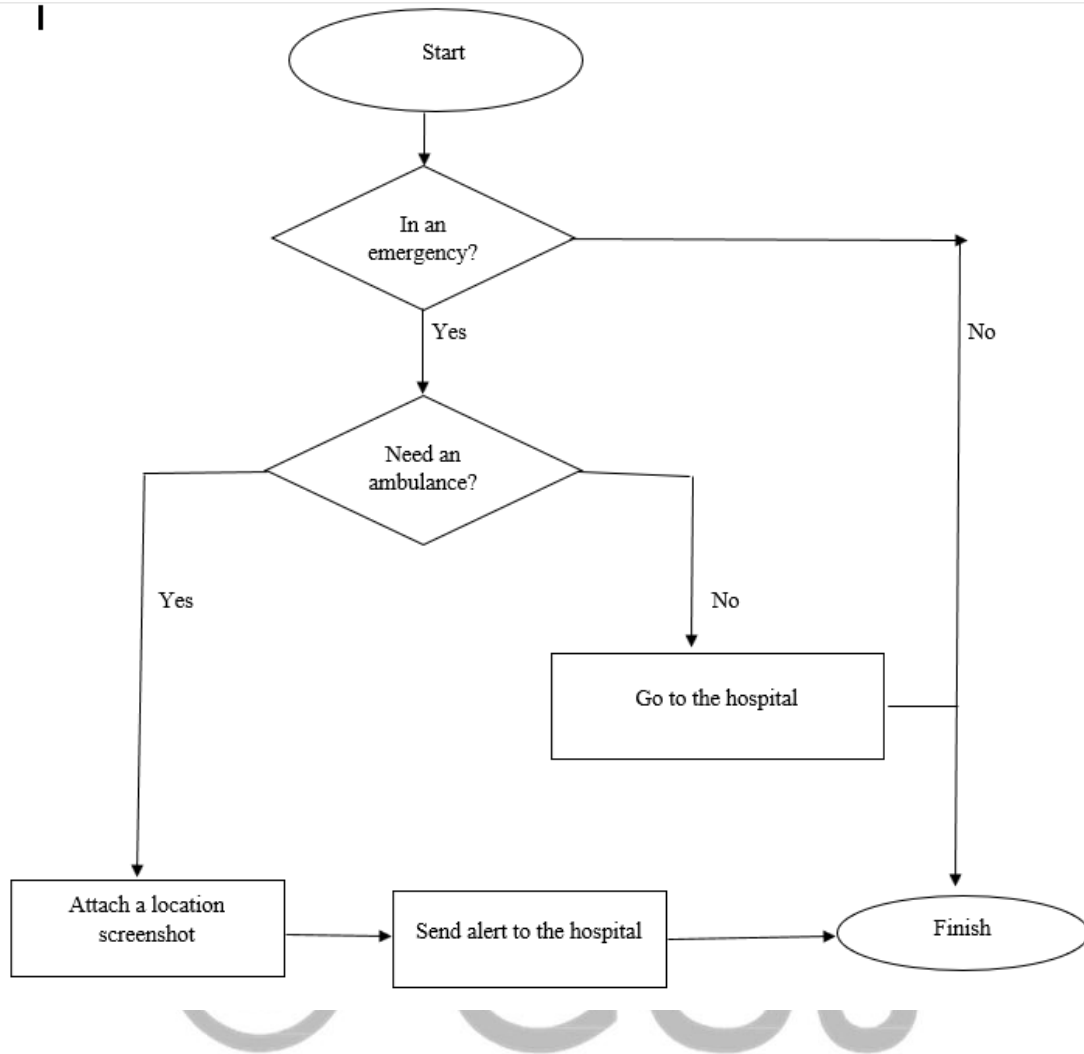
Figure 4.6: Sending alert Module.

4.6.4 Log in and sign up module.

Log in module allows the patient to log in into the application using their user name and a four-digit password. The sign-up module allows the patient to register for the application using their user name, phone number and a password. The UML class diagrams for the two modules are as follows.

27

Patient
-name: String
-email: String
-password: Int
+getName();
+getEmail();
#getPassword();

Table 4.2: Sign up module

Patient
-email: String
-password: Int
+getEmail();
#getPassword();

Table 4.3; Log in module

© GSJ

28

CHAPTER FIVE: SYSTEM DESIGN.

5.1 Introduction.

This chapter describes the design of the application and the architectures of each interface. It also discusses the strengths of the developed application.

5.2 System design.

A system design is a model of a real-world system that has many participating entities and relationships and acts as a basis for detailed implementation. It also offers communication between the designers of the application as well as providing information to the application maintainers about original intentions of the application designers.

5.3 System interface.

All interfaces in this chapter were created in java using the android studio IDE. Alerts were created using java script embedded in the codes.

5.3.1 Log in and sign up interfaces.

The log in interface only allows authorized user name and password to access the services offered by the application. When the wrong username and password is used, the application rejects access to services. When a user forgets his password, he can request for a reset code to be sent to his phone number.

29
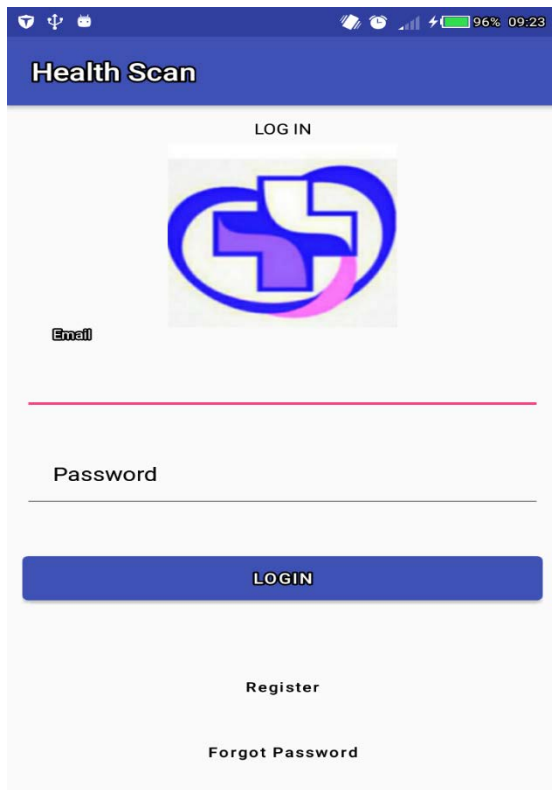
Figure 5.1 Login screenshot 1

The sign-up interface allows a new user to register his or her details like username, phone number and password to the application.

Figure 5.2 Sign up 1

30
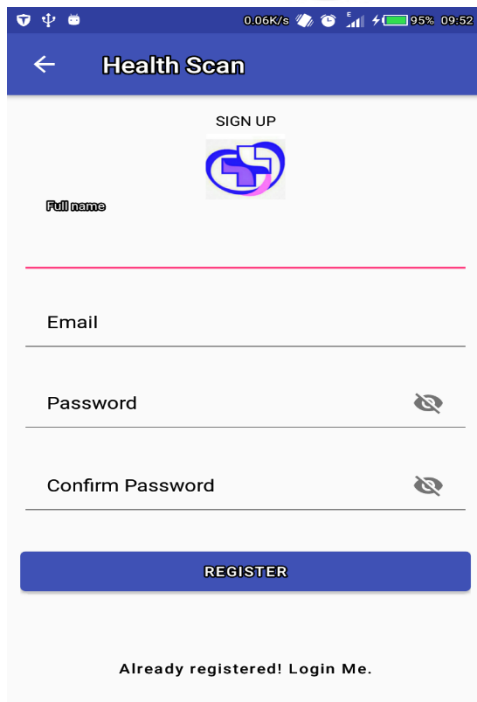
5.3.2 General user interface.

In this interface, all modules contained in the application are displayed. These modules include the google map module, the settings module, the module containing hospital lists and the module for switching accounts. It also includes the name of the logged in user.
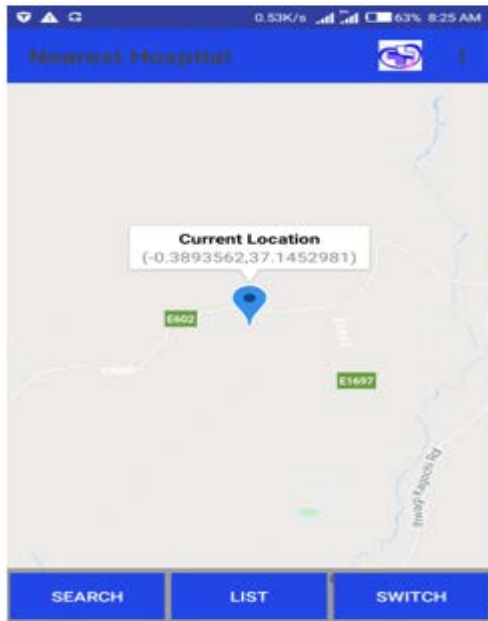
Figure 5.3 General user interface

5.3.3 Google map interface.

This interface contains a pinned location of the user and that of the nearest hospital to them. It also shows the route to the hospital and a circled location of the hospitals.

31
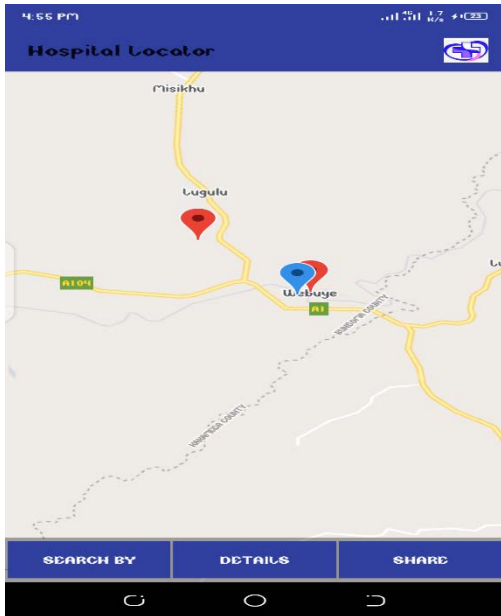
Figure 5.4 Google map interface

5.3.4 Communication interface.

This interface contains a list of generated emergency numbers where patients in distress

can send alerts to call for help. It also shows the position of the ambulance while waiting for

help.
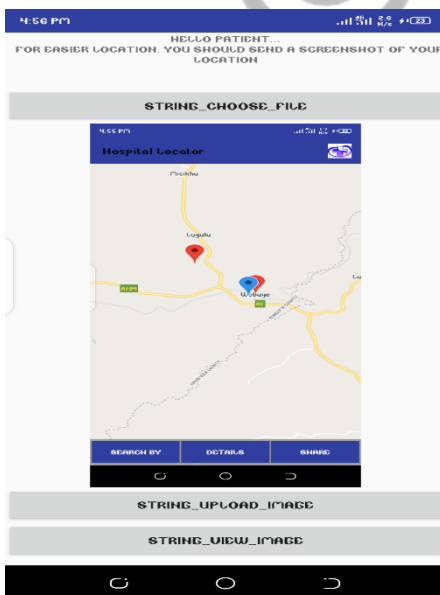
Figure 5.5 sending alert

32
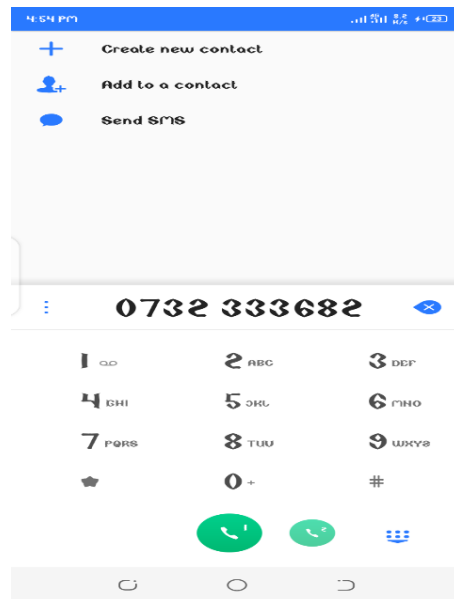
Figure 5.6 calling interface

5.3.5 Settings interface.

This interface gives the user the opportunity to view their dashboard information, share the application and change their themes.
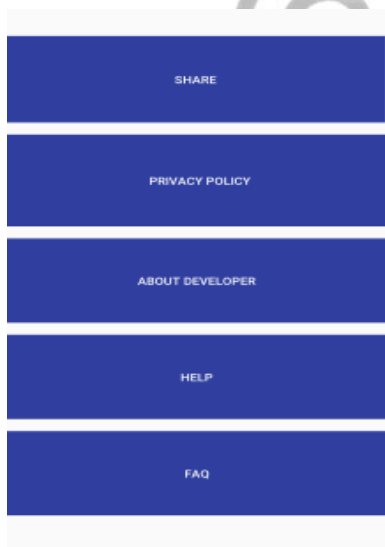
Figure 5.7 Settings interface

5.4 Database design.

A relational database design was used to design the database. A relational database management system (RDBMS) is an excellent tool for organizing large amount of data and defining the relationship between the datasets in a consistent and understandable way. A

33

RDBMS provides a structure which is flexible enough to accommodate almost any kind of data. Relationships between the tables were defined by creating special columns (keys), which contain the same set of values in each table. The tables can be joined in different combinations to extract the needed data. Database of the application was designed using Xampp with the database name Hospitalinfo.db containing the patients or users' image link of their current location.
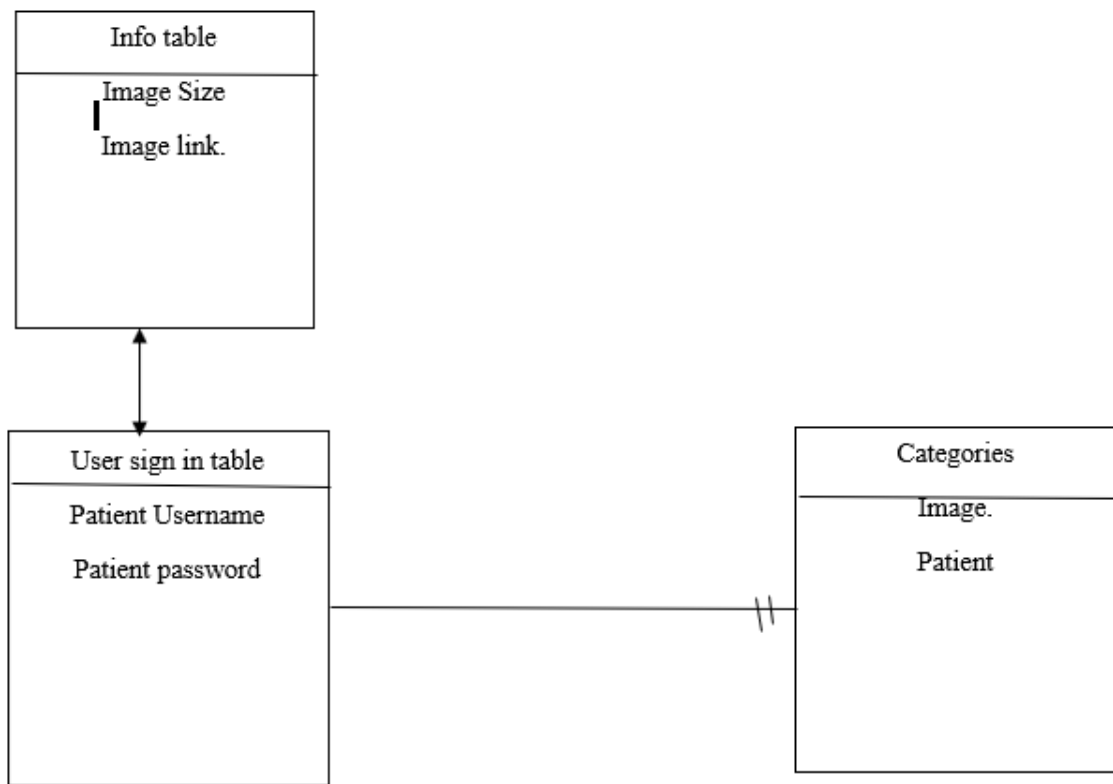
Figure 5.8 ERD for database

34
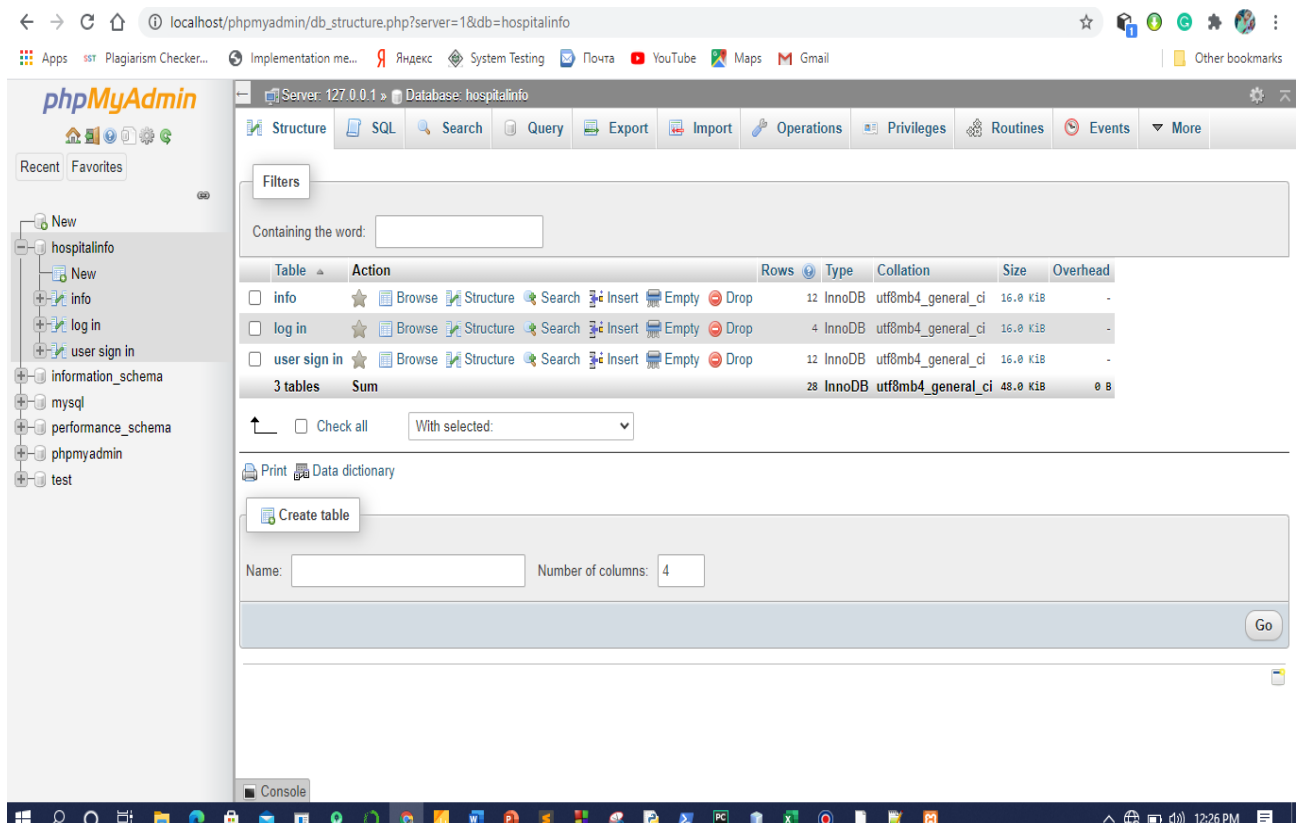
Figure 5.9 Database interface

35

CHAPTER SIX: SYSTEM IMPLEMENTATION.

6.1 Introduction.

This chapter describes how the application was implemented and tested. Implementation is the process of moving an idea from a concept to reality. It ensures that the correct application is delivered to the end users. Testing is the process in which a product or a piece of equipment is examined under extreme conditions. It ensures durability and confirms that the application meets the user requirements.

6.2 Software requirements.

These include the software and the development languages that were used to ensure the completion of the application.

- **Java.**

Java is an object-oriented programming language used for android application development. The goal of the language is to allow developers to develop classes and objects and facilitate communications within the different interfaces. Java applications compile to bytecode (class file) that can then run on a Java Virtual Machine (JVM).

- **MySQL.**

MySQL is an open source relational database management system (RDBMS) that uses structured query language to add, access and process data in databases. MySQL is mainly known for its speed, reliability and flexibility as well as robustness and scalability.

- **Xampp local database.**

36

SQLite studio is a software that responds to clients' requests by providing resources such as XHTML documents as well as powerful features included in a large set of modules example Authentication modules.

- **Android studio IDE.**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

6.3 System implementation.

This describes the tools used to implement the graphical user interface and the database. MySQL was used to create and connect relational tables to the database. Java coded in Android studio was used to develop the user interface. Java script was used to integrate interfaces and return alerts to users.

6.4 System testing.

Testing is usually done after the application is put in place. This was done in two stages: unit testing and integration testing. It requires the development of a test plan for testing to be done effectively.

6.4.1 Test plan.

The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan will identify items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the

37

resources and schedule required to complete testing. The purpose of the software test plan is such as:

- To achieve the correct code and ensure all Functional and Design requirements are implemented as specified in the documentation.

- To provide a procedure for Unit and System Testing.

- To identify the test methods for Unit and System Testing.

6.4.2 Process of test plan.

- Identify the requirements to be tested. All test cases shall be derived using the current design specification.

- Identify particular test to use to test each module.

- Identify the expected results for each test.

- Perform the test.

- Document the test data, test cases used during the testing process.

6.4.3 Unit testing.

Unit testing was carried out on individual modules of the system to ensure that they are fully functional units. This was done by examining each unit, for example the log in and sign up pages were tested to ensure that it validated the users before allowing them to use the application and to ensure that registration information was being sent successfully to the database. The success of each individual unit gave the go ahead to carryout integration testing. All identified errors were dealt with.

6.4.4 integration testing.

Integration testing was carried out after different modules had been put together to make a complete system. Integration was aimed at ensuring that modules are compatible and they can be integrated to form a complete working system. For example, the application was tested to

38

ensure that the patients were able to move from one interface to another with no or limited error possible.

6.4.5 System validation.

As one of the specific objectives of this study, validation of the application was very important. Validation of the application was done by comparing it to the by set objectives of the system. Most of their answers matched with what the system can do. JavaScript was used to validate user input and the respective input. For example, the application's log in and sign up pages do not accept blank field.

6.5 Presentation of Results.

The presentation of results of the application were analyzed in terms of interfaces of the application and outputs from the back end of the system. The following were the results after the implementation of the application.

6.5.1 Application interfaces.

All system interfaces in this chapter were created in android Studio IDE while creating java and java script interfaces.

Database of Health Scan created in Xampp

When the system is successfully created and imported to the SQLLite, the tables can be manipulated by the system administrator into detailed in the MySQL for instance he or she can be selected, deleted, updated, and altered tables.

39
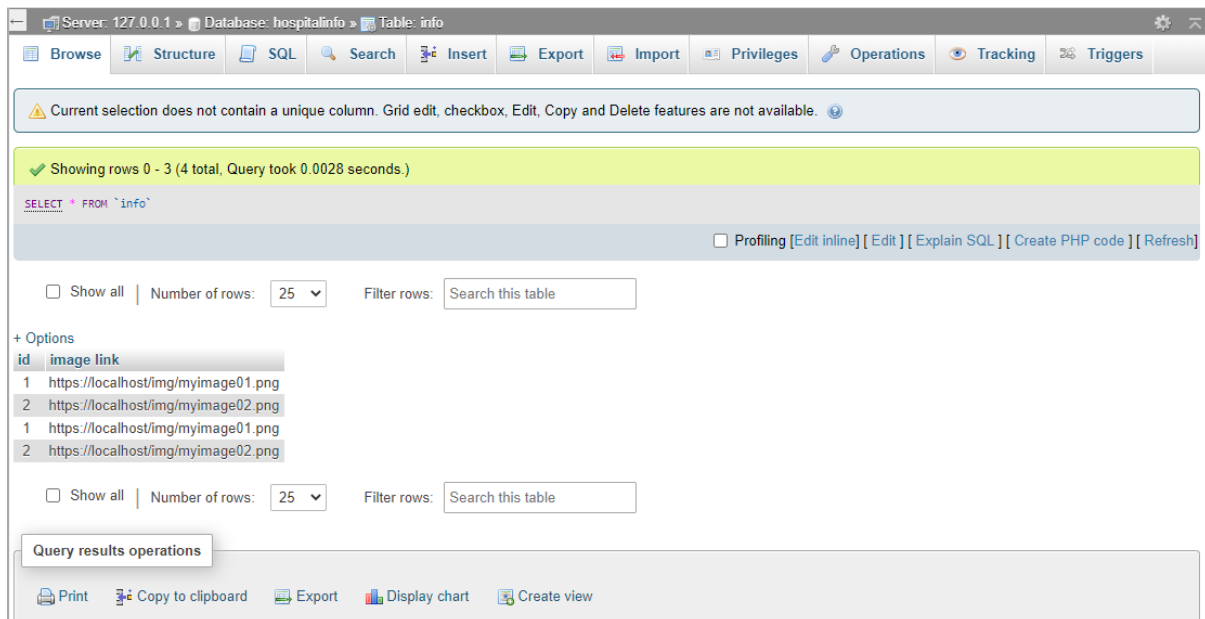
Table 6.0: image database.
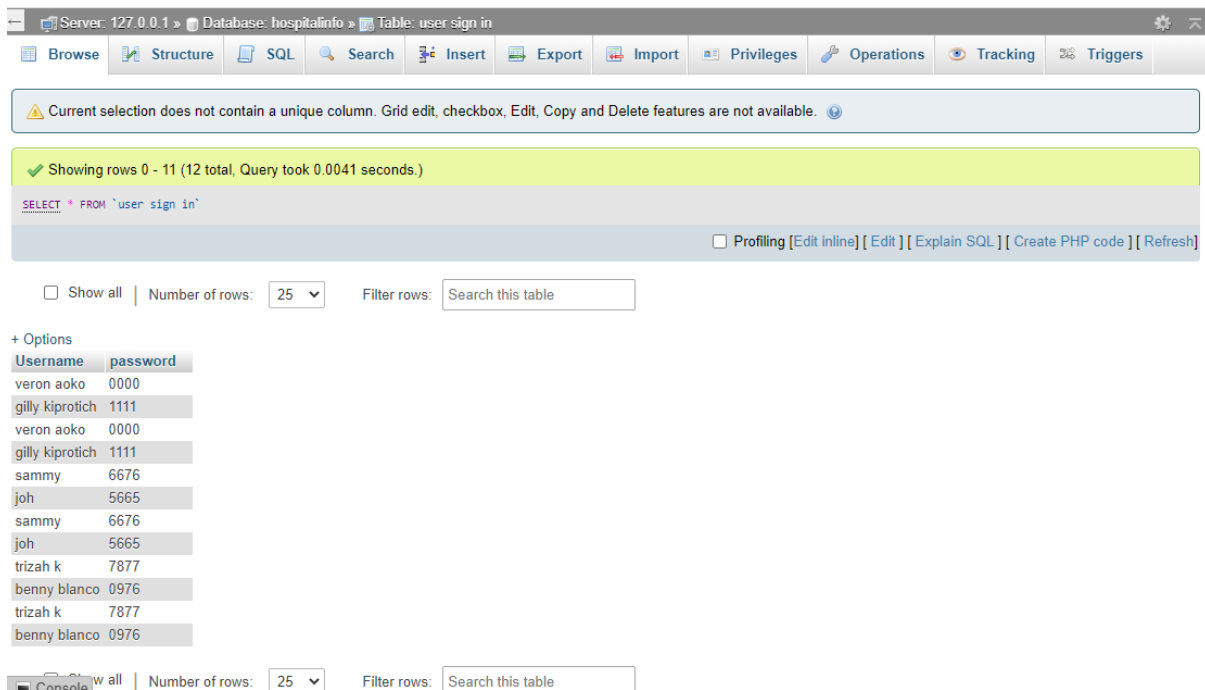
Table 6.1: Log in database.

6.6 Proposed change over techniques.

Pilot Changeover

40

With a pilot changeover, the new system is tried out at a test site before launching it company-wide. For example, a developer first tested the application at one of its collaborating hospitals. This branch is referred to as the pilot, or beta, site for the program and it allows companies to run the new system next to their old but on a much smaller scale. This makes the pilot changeover method much more cost-effective. After the kinks are worked out of the system at the test site, companies usually opt to use the direct changeover technique to launch the system company-wide.

CHAPTER SEVEN: SUMMARY, LIMITATIONS, CONCLUSIONS AND RECOMMENDATIONS.

7.1 Introduction.

This chapter describes the objectives of the application as stipulated in chapter one, limitations of the application, conclusions and recommendations of the system.

7.2 Summary.

As discussed in the previous chapters, the main problem facing patients is failure to get faster and efficient medical attention whenever in new and unfamiliar environments. This problem led me to develop the hospital locator android application to be used in Nyeri County and the country in general to enable patients in distress to locate the nearest hospital and go for treatment or call an ambulance. The project has implemented most of the objectives set in the previous chapters. This application offers a number of benefits to the ministry of health and to the user by capturing, adding, editing and deleting records posted on the database. Problems encountered during data collection was that there are few books written on android development and about hospitals. The main problems encountered during design were limited internet in the university hence difficulty in downloading codes and for research as well as inadequate financial support to facilitate the project.

7.3 Limitations.

This section describes the imperfections or shortcomings of the developed application. They included the following:

➢ User authentication and security of data was to be addressed adequately before the application is deployed into the public health care system.

➢ Further performance evaluation of the application was necessary in order for large scale implementation.

42

7.4 Conclusion.

I approached the problem of delayed arriving of patients to the hospital by developing the health scan application. This application allowed patients to access hospitals near them and be able to contact the hospital in case the emergency was fatal. Proper planning and research were needed to identify barriers and come up with faster adoption of the updates in the application. However, the success of this application depended on ever user being able to access an android smartphone without limitation to geographical boundaries or network problems. Further research on the application was to be done to enable upgrades and largescale implementation of the application.

7.5 Recommendations.

Currently, this application shows the static list of hospitals available in the particular area. This application can be enhanced to provide real-time information about the hospital present in particular time in the specific area. This application is designed and tested within the Nyeri County. However, this application can be upgraded to cover more cities and all major hospitals of Kenya and East Africa. Presently, a radius of fifty km is fixed in the coding and in later version user can be select their specific radius by their own. This application is designed for android devices, later it will be designed for iOS, windows OS and other popular mobile operating systems. A questionnaire survey will be helpful in determining the benefits and usability of this application in general users.

Opportunity and Lesson Learned

During the course of this project, I was able to understand better the concept of GPS and google map console as well as the local databases. This was effectively done through reading of literature and research. The whole process of developing the application was

43

an opportunistic challenge. Seeing the application proposal into a tangible working

application was a rewarding exercise.

44

REFERENCES.

Atack, L., & Maher, J. (2010). Emergency medical and health providers' perceptions of key issues in prehospital patient safety. *Prehospital Emergency Care*, *14*(1), 95-102.

Bennett, A. (2006). *Geneva Convention: The Hidden Origins of the Red Cross*. The History Press.

Brent, A. C., Rogers, D. E., Ramabitsa-Siimane, T. S., & Rohwer, M. B. (2007). Application of the analytical hierarchy process to establish health care waste management systems that minimise infection risks in developing countries. *European Journal of Operational Research*, *181*(1), 403-424.

Christie, P. M. J., & Levary, R. R. (1998). The use of simulation in planning the transportation of patients to hospitals following a disaster. *Journal of medical systems*, *22*(5), 289-300.

Favez, J. C., & Fletcher, J. (1999). *The red cross and the holocaust*. Cambridge University Press.

Fourny, M., Lucas, A. S., Belle, L., Debaty, G., Casez, P., Bouvaist, H., ... & Labarère, J. (2011). Inappropriate dispatcher decision for emergency medical service users with acute myocardial infarction. *The American journal of emergency medicine*, *29*(1), 37-42.

Hirshberg, A., Stein, M., & Walden, R. (1999). Surgical resource utilization in urban terrorist bombing: a computer simulation. *Journal of Trauma and Acute Care Surgery*, *47*(3), 545-550.

Neely, K. W., Bennison, A., Acker, J., Long, D., Norton, R. L., & Schriver, J. A. (1991). Computerized hospital on-line resources allocation link (CHORAL): a mechanism to

45

monitor and establish policy for hospital ambulance diversions. *Prehospital and disaster medicine*, *6*(4), 459-462.

Rau, H. H., Kumar, A., Hsu, C. Y., & Hung, N. C. (2011, August). Identification of variables to decide optimal hospital for emergency patients. In *2011 2nd IEEE International Conference on Emergency Management and Management Sciences* (pp. 254-257). IEEE.

Rau, H. H., Kumar, A., Hsu, C. Y., & Hung, N. C. (2011, August). Identification of variables to decide optimal hospital for emergency patients. In *2011 2nd IEEE International Conference on Emergency Management and Management Sciences* (pp. 254-257). IEEE.

Saini, D., Mirza, M., Gori, M. M., Godwin, C. J., Brown, T. B., Battles, M. H., & Orthner, H. F. (2007, October). Information needs of pre-hospital care providers-a requirements analysis. In *AMIA... Annual Symposium proceedings. AMIA Symposium* (pp. 1098-1098).

46

APPENDICES.
APPENDIX 1: ORGANISATIONS STRUCTURE.

Based on the uses of this application to the patients and the hospital in general, the organizational structure of the hospital responsible for the emergency services is as follows:
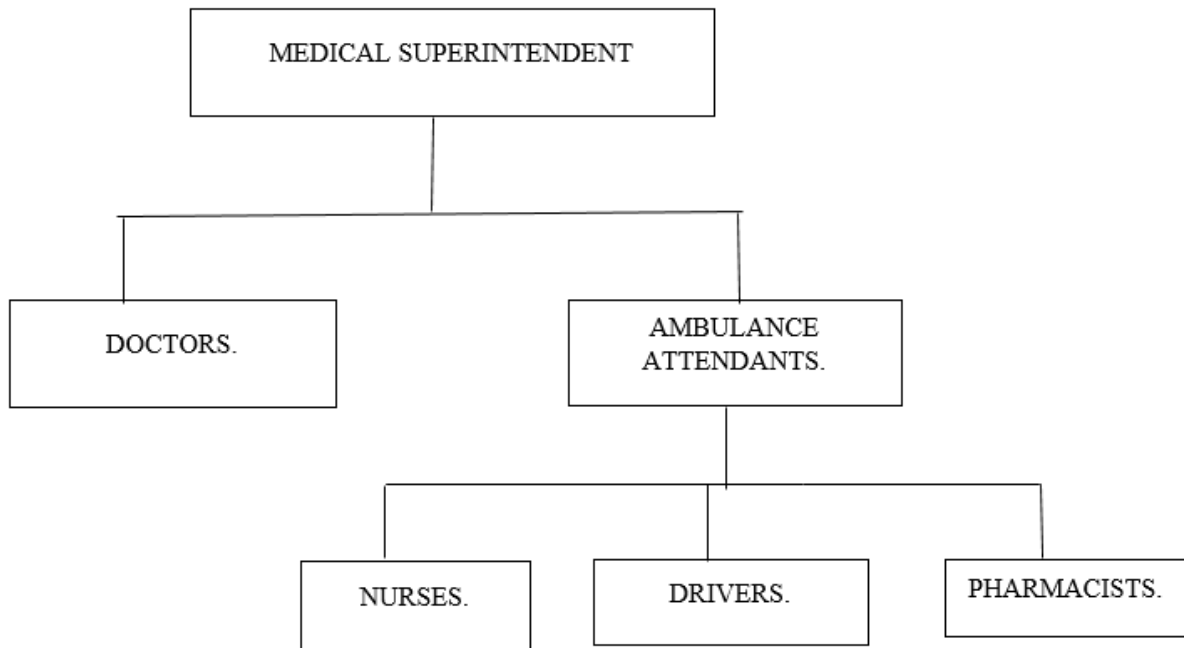

Figure 8.0: Organizational structure.

47

APPENDIX 2: DOCUMENT REVIEW.

In the current emergency services-based application, a patient is not clearly shown where to go for healthcare. The red cross ambulance will only come to aid if the emergency has occurred in large masses so individuals in need are unable to access their services on time

The developed application on the other hand deals with this issue by enabling any patient irrespective of their location to be able to access medical care within the appropriate time

APPENDIX 3: INTRESTING CODES.

The google map code.

```java
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    Marker curmarker;
    LatLng defLocation = new LatLng(28.5, 77); //Delhi
    LatLng curLocation = defLocation;

    int locationType = GooglePlacesApi.TYPE_HOSPITAL;
    int locationRankby = GooglePlacesApi.RANKBY_PROMINENCE;

    LocationManager locMan;
    LocationListener locLis;
    Context ctx = this;
    public static final String TAG = "HL";

    boolean mapReady = false;
    float mapAccuracy = 10000;

    FrameLayout fader;
    AVLoadingIndicatorView avi;

    GooglePlacesApi googlePlacesApi;
    HospitalListClient hospitalListClient;
    PlaceList placeList;
    DistanceResult distanceResult;

    FrameLayout mainFrame;
    Button btnFilter, btnDetails;
    AdView mAdView;

    Spinner spinnerType, spinnerRank;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar2);
        toolbar.setTitle("Health Scan");
        setSupportActionBar(toolbar);

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
```

48

```java
        fader = (FrameLayout) findViewById(R.id.fader);
        avi = (AVLoadingIndicatorView) findViewById(R.id.avi);

        btnFilter = (Button) findViewById(R.id.btnFilter);
        btnDetails = (Button) findViewById(R.id.btnDetails);


        mAdView = (AdView) findViewById(R.id.adView);
        mainFrame = (FrameLayout) findViewById(R.id.mainFrame);

        setLoadingAnimation();

        AdUtil.initAds(ctx);
        AdUtil.loadAds(mAdView);

//      MobileAds.initialize(ctx,"ca-app-pub-3940256099942544~3347511713");
//      AdRequest adRequest = new AdRequest.Builder().build();
//      mAdView.loadAd(adRequest);

        googlePlacesApi = new GooglePlacesApi();
        hospitalListClient = googlePlacesApi.getHospitalListClient();

        btnDetails.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDetailList();
            }
        });

        btnFilter.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showOptionDialog();
            }
        });



    if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION)
!= PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
        ActivityCompat.requestPermissions(this,
            new String[]{
                android.Manifest.permission.ACCESS_FINE_LOCATION,
                android.Manifest.permission.ACCESS_COARSE_LOCATION
            },
            123);
    } else {
        locMan = (LocationManager) getSystemService(LOCATION_SERVICE);
        boolean gpsEnabled, networkEnabled;

        gpsEnabled = locMan.isProviderEnabled(LocationManager.GPS_PROVIDER);
        networkEnabled = locMan.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!gpsEnabled && !networkEnabled) {
            AlertDialog.Builder dialog = new AlertDialog.Builder(ctx);
            dialog.setMessage(getResources().getString(R.string.gps_network_not_enabled));

            dialog.setPositiveButton(getResources().getString(R.string.open_location_settings),
```

49

```java
                        new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {
                                Intent i = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                                startActivity(i);
                                Toast.makeText(ctx, "Restart app after enabling GPS", Toast.LENGTH_SHORT).show();
                            }
                        });

            dialog.setNegativeButton(getResources().getString(R.string.cancel),
                        new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {
                                //Nothing to do here
                                Toast.makeText(ctx, "Enable GPS to allow app to function",
Toast.LENGTH_SHORT).show();
                            }
                        });

            dialog.show();
        } else {
            //Get current location coord
//          Location temp = locMan.getLastKnownLocation(LocationManager.GPS_PROVIDER);

            locLis = new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    curLocation = new LatLng(location.getLatitude(), location.getLongitude());
                    Log.d(TAG, "onLocationChanged: lat: "+curLocation.latitude);
                    Log.d(TAG, "onLocationChanged: long: "+curLocation.longitude);
                    Log.d(TAG, "onLocationChanged: accuracy: "+location.getAccuracy());
                    mapAccuracy = location.getAccuracy();
                    initMapPointer(curLocation);
                }

                @Override
                public void onStatusChanged(String provider, int status, Bundle extras) {

                }

                @Override
                public void onProviderEnabled(String provider) {

                }

                @Override
                public void onProviderDisabled(String provider) {

                }
            };

            Log.d(TAG, "onCreate: Trying for location");
            locMan.requestLocationUpdates(
                    LocationManager.GPS_PROVIDER,
                    100,
                    1,
                    locLis
            );

            locMan.requestLocationUpdates(
```

50

```
        LocationManager.NETWORK_PROVIDER,
        100,
        1,
        locLis
    );
    }
  }

}
```

The Xampp database code.

```php
        $sql ="SELECT<?php


if($_SERVER['REQUEST_METHOD']=='POST'){


    $image = $_POST['image'];


    require_once('dbConnect.php');
     id FROM photos ORDER BY id ASC";


    $res = mysqli_query($con,$sql);


    $id = 0;


    while($row = mysqli_fetch_array($res)){
            $id = $row['id'];
    }


    $path = "uploads/$id.png";


    $actualpath = "http://localhost/PhotoUpload/$path";


    $sql = "INSERT INTO photos (image) VALUES ('$actualpath')";


    if(mysqli_query($con,$sql)){
```

51

```php
file_put_contents($path,base64_decode($image));

echo "Successfully Uploaded";

}



mysqli_close($con);

}else{

echo "Error";

}
```

For database connecting.

```php
<?php

define('HOST','localhost');

define('USER','u502452270_andro');

define('PASS',          );

define('DB','HospitalInfo');

$con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
```

The hospitals' list code.

```java
public class ListActivity extends AppCompatActivity {

  RecyclerView recyclerHospital;
  ArrayList<SinglePlace> itemList;
  AdView mAdView;
  Toolbar toolbar;

  public static final String TAG = "list";

  @Override
  protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    Log.d(TAG, "onNewIntent: new intent ");
    setIntent(intent);
  }

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_list);
    toolbar = findViewById(R.id.toolbar);
    toolbar.setTitle("Details");
    setSupportActionBar(toolbar);

    Intent intent = getIntent();
    if(Intent.ACTION_SEARCH.equals(intent.getAction()))
      Log.d(TAG, "onCreate: search started");
```

52

```java
        itemList = Parcels.unwrap(getIntent().getParcelableExtra("itemList"));

        recyclerHospital = findViewById(R.id.up);
        recyclerHospital.setLayoutManager(new LinearLayoutManager(this));

        HospitalListRecycler hospitalListRecycler = new HospitalListRecycler(itemList,this);
        recyclerHospital.setAdapter(hospitalListRecycler);

        RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(this);
        recyclerHospital.setLayoutManager(layoutManager);

        mAdView = findViewById(R.id.adView);

        AdRequest adRequest = new AdRequest.Builder().build();
        mAdView.loadAd(adRequest);

        AdUtil.loadAds(mAdView);
    }


}
```

Sending alert code.

```java
public class SendalertActivity extends DetailActivity implements View.OnClickListener{


    public static final String UPLOAD_URL = "http://simplifiedcoding.16mb.com/PhotoUpload/upload.php";
    public static final String UPLOAD_KEY = "image";



    private int PICK_IMAGE_REQUEST = 1;

    private Button buttonSelect;
    private Button buttonUpload;
    private Button buttonView;

    private ImageView imageView;

    private Bitmap bitmap;

    private Uri filePath;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sendalert);

        buttonSelect = (Button) findViewById(R.id.buttonSelect);
        buttonUpload = (Button) findViewById(R.id.buttonUpload);
        buttonView = (Button) findViewById(R.id.buttonViewImage);

        imageView = (ImageView) findViewById(R.id.imageView);

        buttonSelect.setOnClickListener(this);
        buttonUpload.setOnClickListener(this);
        buttonView.setOnClickListener(this);
    }

    private void showFileChooser() {
```

53

```java
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null &&
data.getData() != null) {

            filePath = data.getData();
            try {
                bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
                imageView.setImageBitmap(bitmap);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public String getStringImage(Bitmap bmp){
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] imageBytes = baos.toByteArray();
        String encodedImage = Base64.encodeToString(imageBytes, Base64.DEFAULT);
        return encodedImage;
    }

    private void uploadImage(){
        class UploadImage extends AsyncTask<Bitmap,Void,String>{

            ProgressDialog loading;

            RequestHandler rh = new RequestHandler();

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                loading = ProgressDialog.show(SendalertActivity.this, "Uploading...", null,true,true);
            }

            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                loading.dismiss();
                Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
            }

            @Override
            protected String doInBackground(Bitmap... params) {
                Bitmap bitmap = params[0];
                String uploadImage = getStringImage(bitmap);

                HashMap<String,String> data = new HashMap<>();

                data.put(UPLOAD_KEY, uploadImage);
                String result = rh.sendPostRequest(UPLOAD_URL,data);
```

54

```java
        return result;
    }
}

    UploadImage ui = new UploadImage();
    ui.execute(bitmap);
}

@Override
public void onClick(View v) {
    if (v == buttonSelect) {
        showFileChooser();
    }

    if(v == buttonUpload){
        uploadImage();
    }

    if(v == buttonView){
        viewImage();
    }
}

private void viewImage() {
    startActivity(new Intent(this, ImageListView.class));
}
}
```
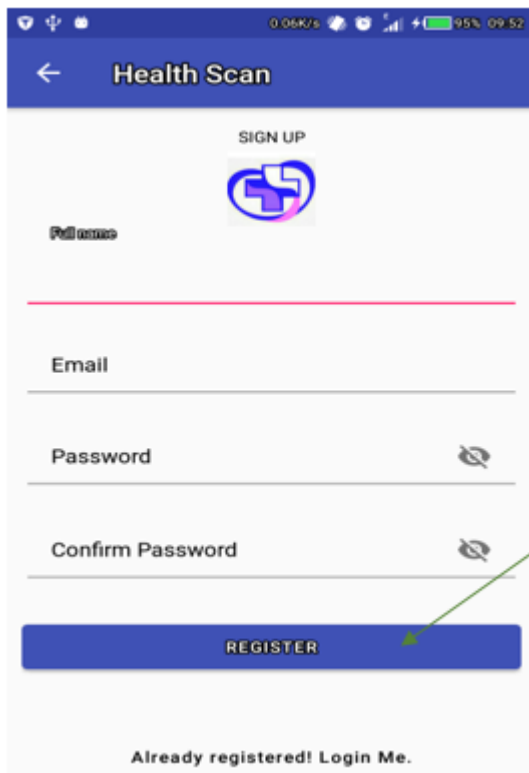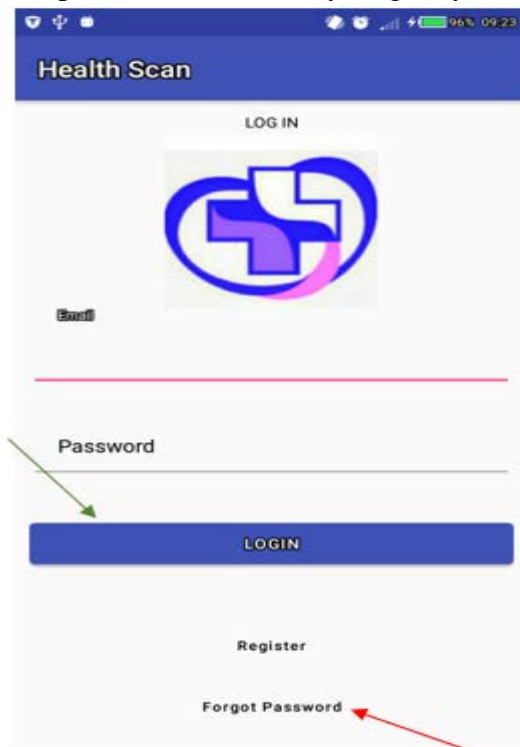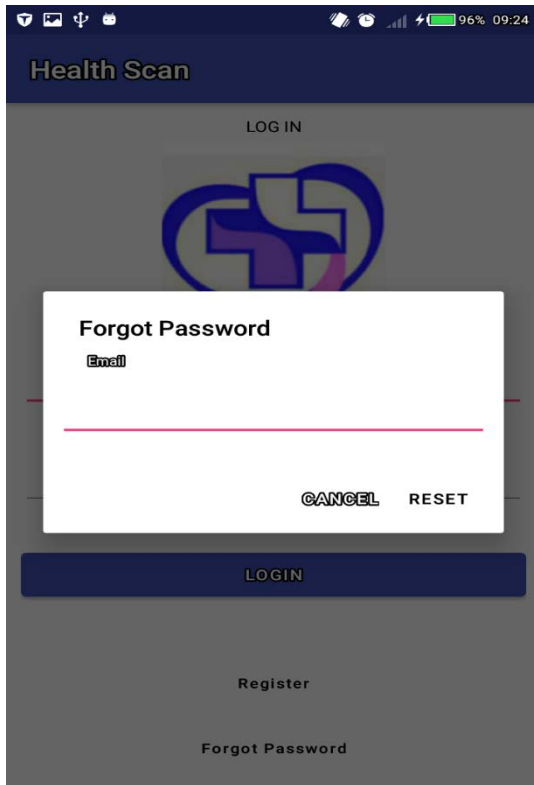
55

55

APPENDIX 4: USERS' MANUAL.

I.   Users of the application are required to register to the services on the application using their email, username and password.
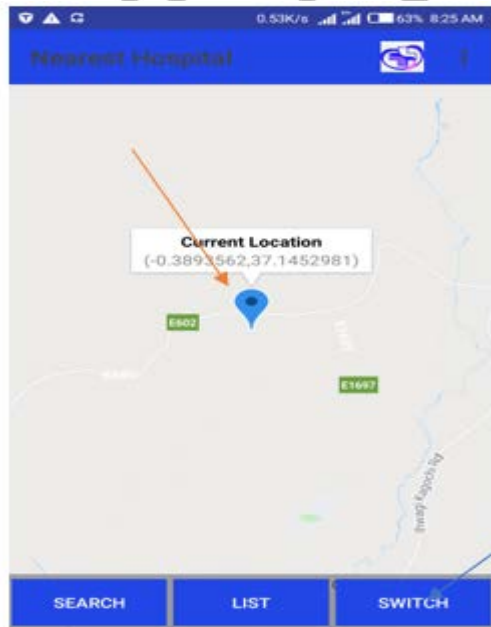

II.   After registration, users can log in using the email and password. A user can reset their passwords incase they forget by submitting their email address.
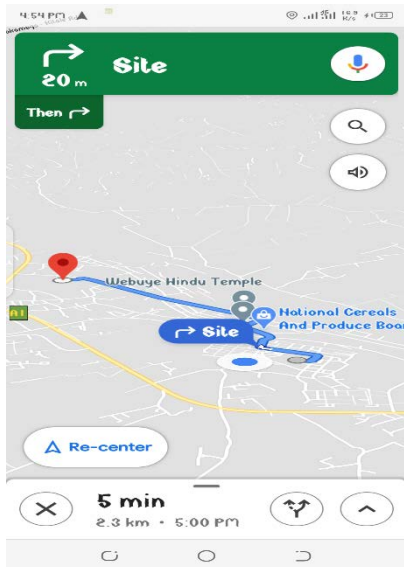

56

III. The application will then direct you the google map module automatically.

IV. Users can check nearest routes in the map.
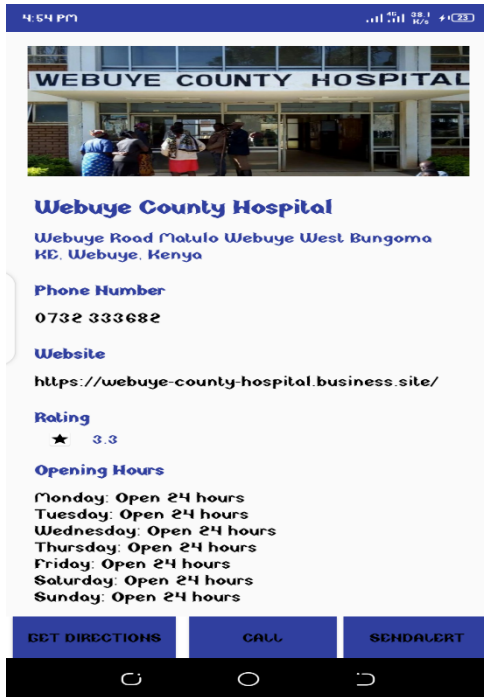
57

V.   Users can also view a list of nearest hospitals.

VI.   You can also view the details of a specific healthcare institution.

58

VII. Medical superintendents can view sent requests.

59