

Quick Health Service

By

Nafees Ahmed Shihab

ID: 152-15-5824

Farzia Afrin

ID: 151-15-5201

Bijoy Ghosh

ID: 152-15-5556

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering.

Supervised by

Prof. Dr. Syed Akhter Hossain

Professor and Head

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

NOVEMBER 2019

APPROVAL

This Project titled “**The Quick Health Service**”, submitted by Nafees Ahmed Shihab ID No: 152-15-5824, Farzia Afrin ID No: 151-15-5201, Bijoy Ghosh ID No: 152-15-5556, to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 7 December 2019.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Md. Zahid Hasan

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Sadekur Rahman

Assistant Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Dewan Md. Farid

Associate Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Prof. Dr. Syed Akhter Hossain, Head, and Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree.

Supervised by:

Prof. Dr. Syed Akhter Hossain
Professor and Head
Department of CSE
Daffodil International University

Submitted by:

Nafees Ahmed Shihab
ID: 152-15-5824
Department of CSE
Daffodil International University

Farzia Afrin
ID: 151-15-5201
Department of CSE
Daffodil International University

Bijoy Ghosh
ID: 152-15-5556
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty Allah for His divine blessing makes us possible to complete the find year project successfully.

The real spirit of achieving a goal is through the way of excellence and austere discipline. We would have never succeeded in completing our task without the cooperation, encouragement, and help provided to us by various personalities.

We would like to express our deepest appreciation to our supervisor **Prof Dr. Syed Akhter Hossain** Head, Department of CSE. BSc in CSE continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project. We are especially grateful to our honorable teachers.

We would like to express our heartiest gratitude to **Prof Dr. Syed Akhter Hossain Head**, Department of CSE, for his kind help to finish our project and we would also like to admit with much appreciation the crucial role of the staff of Daffodil International University (DIU), who gave me to clear out our understandings. We have to appreciate the guidance given by the other supervisors and lecturers who have helped us to clear our understandings and created a concern and importance of completing the project report carefully with maintaining good knowledge and quality.

We would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

The Quick Health Service is an android application, which contains doctor information, medical service, blood bank service and so many feature. People can get sick anytime anywhere. Moreover, that sudden time people need emergency treatment. Sometimes we need doctor urgently. Therefore, in our app we build some feature that user can find doctors easily by searching their problems. In addition, user can find their nearest doctors information & the hospital location where they get the doctor. That is our most important feature and we called it "**Find Doctor**". This is very much needed and workable now a day. User here can register and login their profile here. User have headache, he/she search about headache and easily find nearest headache specialist doctor in a second. Some time we don't know that which doctor is good or bad. So we have public review option for every doctors. That's why user can choose best one of them. Before our generation we go too far from home for reach a doctor. But in this generation there are too many doctors everywhere. For this reason, many of them are not well professional and give us wrong treatment. So we make this application for the people that find best doctor & get better treatment easiest way. Why this application? Technology has upgraded. Everything is in our hands now. People want everything in easiest way as they can. So we think this will change a big step of medical term.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
Chapter 1: Introduction	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	3
Chapter 2: Background	4-6
2.1 Introduction	4
2.2 Related Works	4
2.3 Comparative Studies	5
2.4 Scope of the Problem	5
2.5 Challenges	6
Chapter 3 : Requirement Specification	7-24
3.1 Business Process Model	7
3.2 Requirement Collection and Analysis	9
3.3 Use Case Modeling and Description	10
3.4 Logical Data Model	19
3.5 Design Requirements	22
Chapter 4 : Design Specification	25-33
4.1 Front Design	25
4.2 Back-end Design	31

CONTENTS	PAGE
4.3 Interaction Design and UX	31
4.4 Implementation Requirements	31
Chapter 5 : Implementation And Testing	34-37
5.1 Implementation of Database	34
5.2 Implementation of Front-end Design	34
5.3 Implementation of Interactions	34
5.4 Testing Implementation	35
5.5 Test Results and Reports	35
Chapter 6 : Conclusion And Future Scope	38
6.1 Conclusion	38
6.2 Scope for Further Developments	38
Appendices	39
Reference	40

List of Figures

Figures	Page No
Figure 3.1.1: Waterfall Model	7
Figure 3.2.1: Business Process Model Diagram	8
Figure 3.3.1: Use Case Model for User	11
Figure 3.3.2: Use Case Model for Admin	12
Figure 3.3.3: Use Case Model for Full System	13
Figure 3.3.4: Sequence Diagram	18
Figure 3.4.1: Class Diagram	19
Figure 3.4.2.1: Client Server Architecture	21
Figure 3.5.1: Android Studio Logo	23
Figure 3.5.2: NetBeans Logo	23
Figure 3.5.3: Xampp Logo	24
Figure 4.1.1: Homepage	25
Figure 4.1.2: Hospital List	26
Figure 4.1.3: Doctor List	27
Figure 4.1.4: Near Me	28
Figure 4.1.5: Hospital Details	29
Figure 4.1.6: Ambulance Number	30

List of Tables

Tables	Page No
Table 2.3.1: Quick Health Service vs Other	5
Table 3.3.1: Use Case modeling table for user Registration	14
Table 3.3.2: Use Case modeling table for user Login	15
Table 3.3.3: Use case modeling table for Sending Notification	15
Table 3.3.4: Use case modeling table for Check History	16
Table 3.3.5: Use case modeling table for Viewing Promotional Offers in Hospital	16
Table 3.3.6: Use case modeling table for Checking Notifications	17
Table 5.4.1: Testing Implementation	35

CHAPTER 1

INTRODUCTION

1.1 Introduction

The “**Quick Health Service**” is an android application, which contains doctor information, Hospital Information’s and other Medical Services.

People can get sick anytime anywhere. Moreover, that sudden time people need emergency treatment. Sometimes we need doctor urgently. Therefore, in our app we build some feature that user can find doctors easily by searching their problems. In addition, user can find their nearest doctors information & the hospital location where they get the doctor.

Human Organ Donation, and Blood donation service will also include here in future. That human can get/donate organ, blood or manage tissue for transplant.

1.2 Motivation

Now a day, technology updated. We are living in modern generation. Therefore, people want technology easier. There are many hospitals, doctors in everywhere in the world. However, problem is we do not get our treatment perfectly. Sometimes we are late to treatment; sometime we got wrong treatment from wrong doctors.

Suppose, you have headache and you do not know the nearest doctor or specialist information.

- You do not know which hospital or doctor is best for your headache.
- You have to call someone relative to know that where is better doctor stayed.
- Sometimes you cannot find you desire doctor. Then you have to go random doctor for get your treatment. Which is not good, because there are so many fraud / illegal doctor around you.
- To find you the best doctor or hospital, we have plan this project.

- There is another major service is Donation of organ and tissue.
- Even you can find blood donor if you need blood.
- Blood search is easy, just search for your blood then you will get response from that blood donor.

1.3 Objectives

We are going to make “**Quick Health Service**” easier and safer for every people.

- You do not have to struggle to find doctor for your treatment.
- You will get basic treatment for some diseases in our application.
- You can find the best hospitals nearest you.
- Many people do not know the specialist doctor much well, so here is the solution that give you the better specialist doctor.
- You can rate doctor you visited, so it is easy to recognize doctor’s specialty.
- Sometimes you seeking for blood, but you cannot get it for miscommunication, or more people do not know about your concern.
- Therefore, there is a solution for blood seeker to find their wanted blood donor as soon as possible.
- Every day, 22 people die while waiting for an organ to become available. So we build a feature to reach for donor.
- You can request for blood or you can donate your blood easily.

At this stage of modern technology these must needed.

1.4 Expected Outcomes

1. A user-friendly secure medical application.
2. A complete medical application with all medical terms.
3. Saves time for patient who need treatment early.
4. Help user to lead their life more easily and secure.
5. Give user the valid information about hospital and doctors.

6. User can easily reach hospitals / doctors by this application.
7. People get organ from donor which is very hard to find in our world.

• **How “The Quick Health Service” will generate revenue?**

As a startup we will run this for free for some months. Then we will take reviews from the users. Then,

1. Medical promotion, there offers, address all we can give to our users for free. Where medical will give money to us.
2. In future, users can make their appointment with doctors. Contact there. Users have to paid for this. But doctor information and other medical information is free for all.
3. If we can be successful in bringing all hospitals & doctors under our service, then we will charge a light amount per transaction of medical service.

1.5 Report Layout

In **Chapter 1** we have discussed about introduction, motivation and expected outcomes of our project. Later followed by the report design.

In **Chapter 2** we discussed about the background conditions of our project.

In **Chapter 3** we showed business process model, pre-requisite gathering and investigation, the utilization case model, design and other.

In **Chapter 4** we showed all design of the project. Front-end design and back-end design, interaction design and UX and the implementation requirement.

In **Chapter 5** we have showed the implementation of database, front-end design, interactions, test implementations and the test results of the project.

In **Chapter 6** we discussed about the topics like- conclusion and future scope and future developments. Which will pretty much define our project.

CHAPTER 2

BACKGROUND

2.1 Introduction

Android is the most adoptable working system. Wide range of people are getting involved in android working structure. Our app is fully android based and easy to use. The user end and the merchant end app are fully connected. In a way it is an account based system where admin plays vital role as a mediator. Here, we meant customer (buyer) by user and shopkeeper (seller) by merchant. This application will be used to complete transaction using any existing payment provider and to search offers nearby for shopping. By our wallet, we will reduce the harassment user faces during cashless transaction and will give the user a fresh payment environment.

2.2 Related Works

Our app is totally different. Not copied from anywhere. As per our knowledge, our app will be a combination of all features of medical term. Find doctor as per as your disease, connect with them, find special category based hospitals near users, information of all hospitals & doctors, donate or get blood, organ, tissue which is rare to arrange for anyone in the world. It is totally unique app for medical term. It's make easier to people to get their treatment easy & fast than any other system.

2.3 Comparative Studies

In our study, we are trying to make a system which will be fast & reliable for medical term.

- How are we different and unique from others?

Table 2.3.1: Quick Health Service vs Other

Other	Quick Health Service
1. There are many doctor info apps but not accurate. Few information's available.	1. We have accurate doctor info with review option that user can review doctor and give proper quality of them.
2. Other medical apps can't make appointment with doctors.	2. We will give that chance to users and we are working for these.
3. Other have no promotional service for hospital or doctors.	3. We can make promotion of hospitals offers and doctor's service.
4. There are few apps which have only blood donation system.	4. We have blood donation, organ donation/registration, tissue donation.

2.4 Scope of the Problem

We are trying very hard to develop this application without any problems. However, due to many limitations and lack of experience our applications has a few problems. We will solve this problem in near future. First of all, we have to ensure the customers that is totally easy and secure for use. We need medical information deeply which will not be easy to collect. We will be going to collect these and make sure that we will use it for our users to help them. We have to make our server strong that user can use it 24/7.

2.5 Challenges

To establish something new we have to face many difficulties and challenges.

1. Advertising about this App
2. Application Stability
3. 24/7 Service
4. Cloud Security
5. Convincing the hospitals, doctor and other medical authorities
6. Need more user for review and ratings
7. Configuring a smooth cashless user friendly application
8. Fast and Smooth service.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

We are following waterfall model for our project.

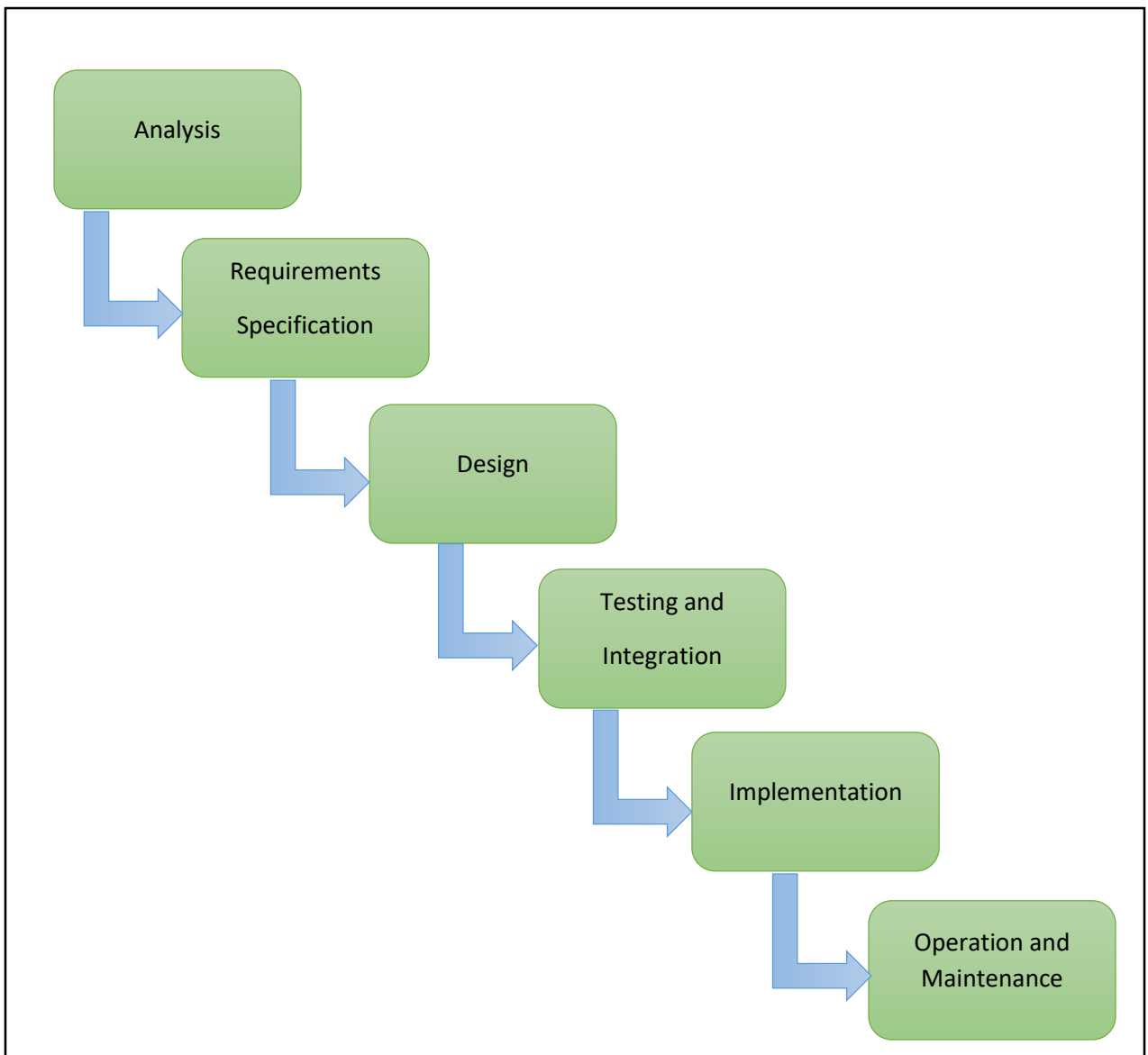


Figure 3.1.1: Waterfall Model

Waterfall model is a linear (sequential) development life cycle model that describes development as a chain of successive steps. No phase can be started before or simultaneously with the previous or current one. Let's consider the Waterfall model's main phases as they go. in which advancement is viewed as streaming consistently downwards (like a waterfall) through the periods of Analysis, Requirement Specification, Design, Implementation, Testing and Integration, Operation and Maintenance. On the off chance that task disappointment is distinguished, at that point it will require less exertion for discovering mistake. In the cascade model past stages are should have been right before continuing to the following. The method for working in this model guarantees that there are explicit stages left. In our project all requirements are clear and it is a big project. All the activities in our project are carried out by following waterfall model.

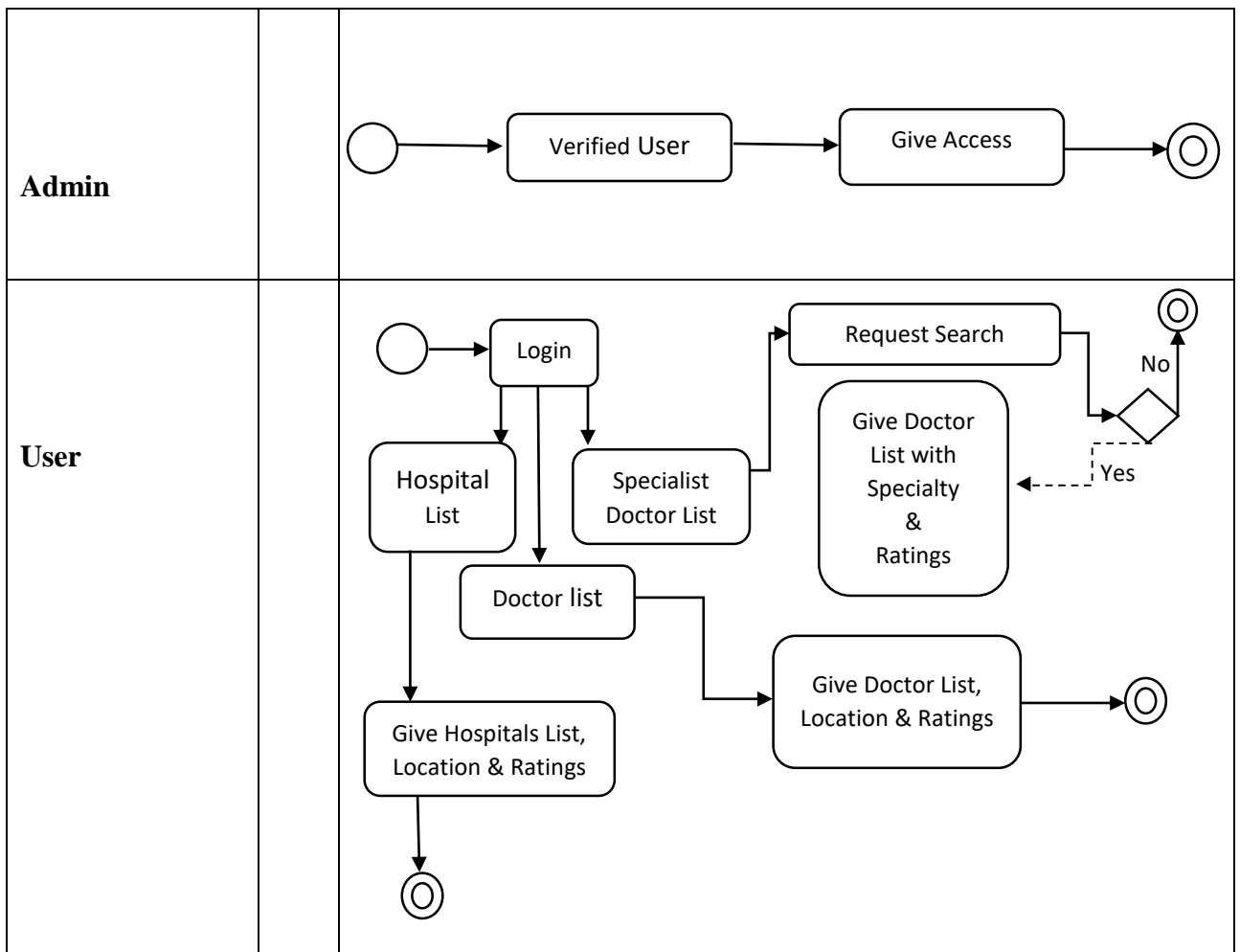


Figure 3.2.1: Business Process Model Diagram

3.2 Requirement Collection and Analysis

Our application will have a lot of information about medical information, disease, doctors, blood, organ, tissue donation related. We have to examine the information to provide user accurate and useful experience. There are some categories of collecting data. Doctor information, Hospital information.

3.2.1 Functional Requirements

- It should run on every smartphone.
- It should provide a stable service.
- It should show correct data.
- It should show notifications.
- It should show history.
- It should complete appointment with patient and doctors.
- It should show confirmation message.
- Overall, it should be user-friendly.

3.2.2 Non-Functional requirements

Safety Requirements:

If there any network problem or database crash, then there should be an alternate server to manage whole system otherwise it restarts itself.

Security Requirements:

We will ensure online support and security for this software.

Software Quality Attributes

- **Availability:** Our project hosting by our server. It will available 24/7.
- **Correctness:** This system will give user accurate response with their needs.
- **Maintenance:** This will maintain always correct information with correct response.

- **Usability:** This system is for maximum user to satisfy.

3.2.3 Hardware Requirements

- Android Smart Phone.
- Internet Connection.
- PC (desktop/ laptop)

3.2.4 Software Requirements

- Android Studio
- MySQL (back-end)
- PHP
- HTML
- MS Word
- Web Browser
- Operating System.

3.3 Use Case Modeling and Description

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. The usage case ought to contain all structure rehearses that have significance to the users.

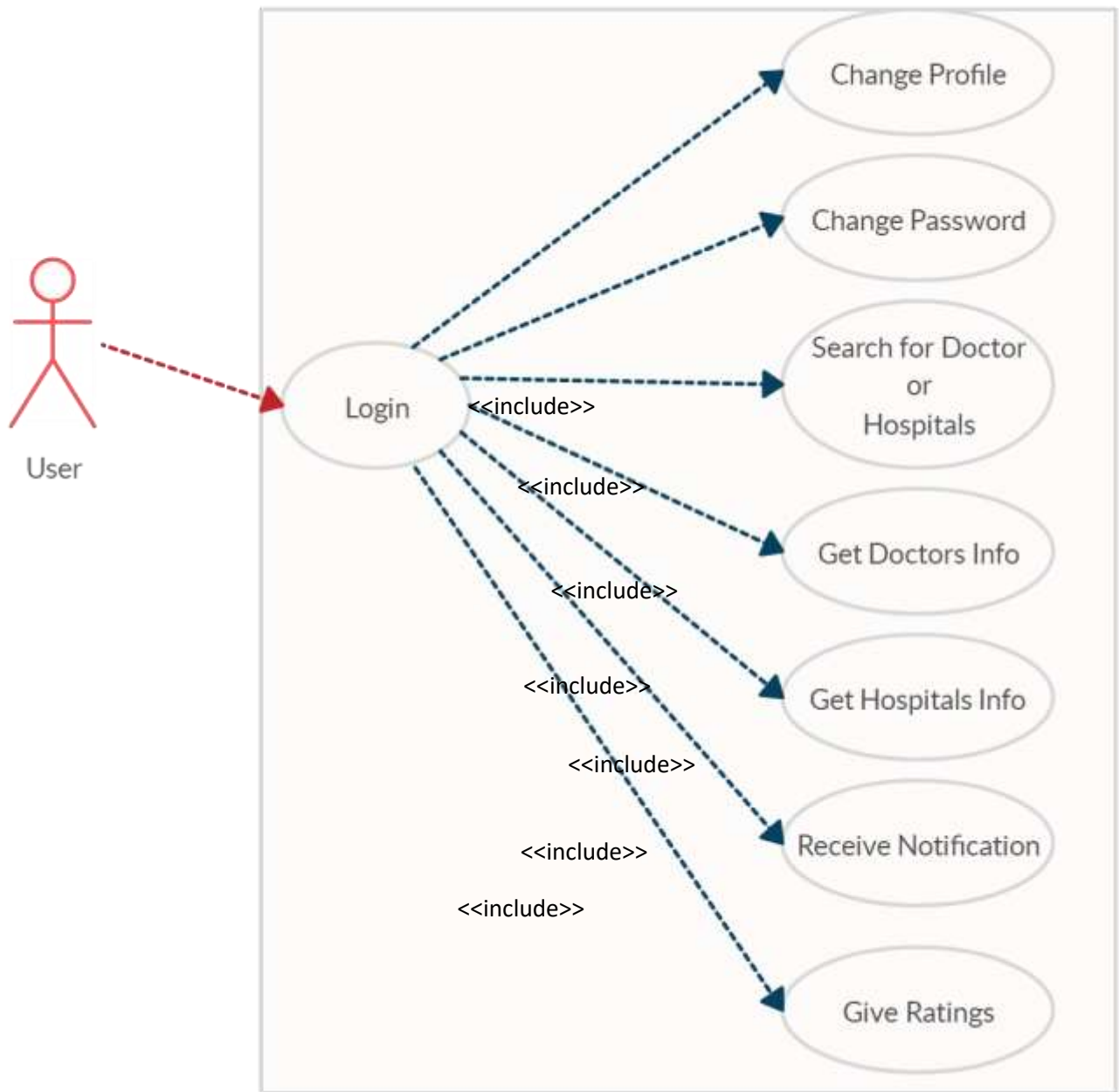


Figure 3.3.1: Use Case Model for User

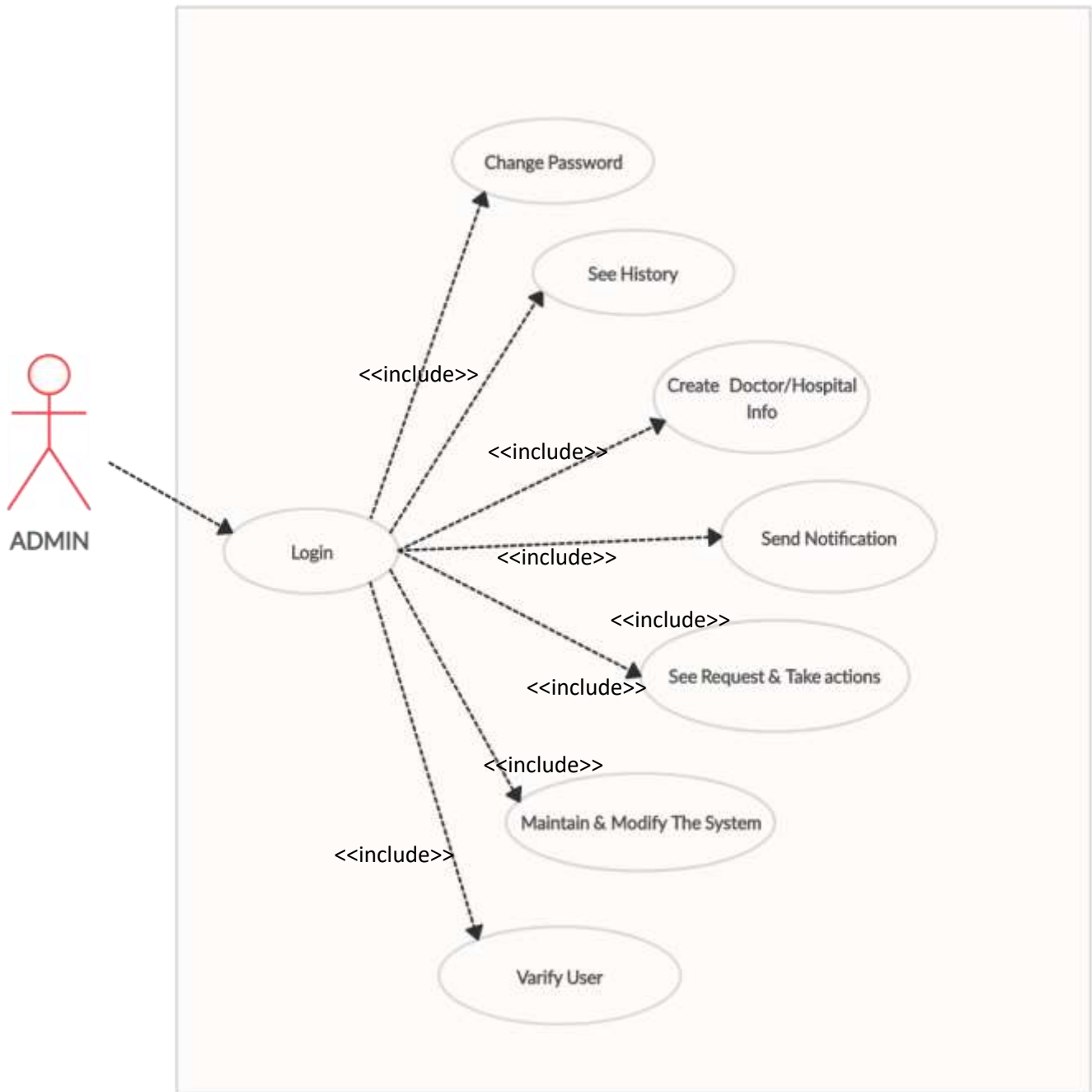


Figure 3.3.2: Use Case Model for Admin



Figure 3.3.3: Use Case Model for Full System

Table 3.3.1: Use Case modeling table for user Registration

Use Case Name	Use Case model for user Register
Actor	User
Pre-Condition	Not Needed
Internal Path	<ol style="list-style-type: none"> 1. Enter Name 2. Enter Mobile Number 3. Verify Mobile Number 4. Enter Email 5. Enter Password
Exception Path	<ol style="list-style-type: none"> 1.1 Please Enter Mobile Number 2.1 Please Verify First 3.1 Please Enter Email 4.1 Please Enter Password
Notes	User need to register to use the Apps major features.

This use case modeling table shows how the user registration process will work and what will be the exception messages.

Table 3.3.2: Use Case modeling table for user Login

Use Case Name	Use Case model for user Login
Actor	User
Pre-Condition	Registration
Internal Path	<ol style="list-style-type: none"> 1. Enter Mobile Number 2. Enter Password
Exception Path	<ol style="list-style-type: none"> 1.1 Please Enter Mobile Number 2.1 Please Enter Password
Notes	User need to Login to use the App major features.

This use case modeling table shows how the user login process will work and what will be the exception messages.

Table 3.3.3: Use case modeling table for Sending Notification

Use Case Name	Use case model for Sending Notification
Actor	Admin
Pre-Condition	Given access by authority
Internal Path	<ol style="list-style-type: none"> 1. Enter Hospital Name 2. Enter new offer / discount 3. Enter Any Event (Optional) 4. Tap Send
Exception Path	<ol style="list-style-type: none"> 1.1 Please enter Hospital Name 2.1 Please enter new offer / discount 3.1 Please enter Any Event (Optional) 4.1 Tap Send to send Notification
Notes	Admin need access by authority

This use case modeling table (Table 3.3.3) shows how Admin sending notification feature will work and what will be the exception messages of that.

Table 3.3.4: Use case modeling table for Check History

Use Case Name	Use case model for Check History
Actor	Admin
Pre-Condition	Given access by authority
Internal Path	<ol style="list-style-type: none"> 1. Check History 2. Expend History
Exception Path	<ol style="list-style-type: none"> 1.1 Scroll to view 2.1 Please Click to Expand
Notes	Admin need access by authority

This use case modeling table shows how the Admin history will be showed and what will be the exception messages if there is an error.

Table 3.3.5: Use case modeling table for Viewing Promotional Offers in Hospital

Use Case Name	Use case model Viewing Promotional Offers
Actor	User
Pre-Condition	Login
Internal Path	<ol style="list-style-type: none"> 1. Tap on the Offers 2. Tap on Hospital Location
Exception Path	<ol style="list-style-type: none"> 1.1 Tap on the Offers 2.1 Tap on Hospital Location
Notes	User need to login first to use this feature

This use case modeling table shows how the user will view selected area or city Hospital's promotional offers in the app and how to login first to use the feature.

Table 3.3.6: Use case modeling table for Checking Notifications

Use Case Name	Use case model for Checking Notifications
Actor	User
Pre-Condition	Login
Internal Path	1. Tap on Notification button
Exception Path	1.1 Please Tap on Notification button to check notification
Notes	User need to login first to use this feature

This use case modeling table shows how the user will check notifications in the app.

3.3.1 Sequence Diagram

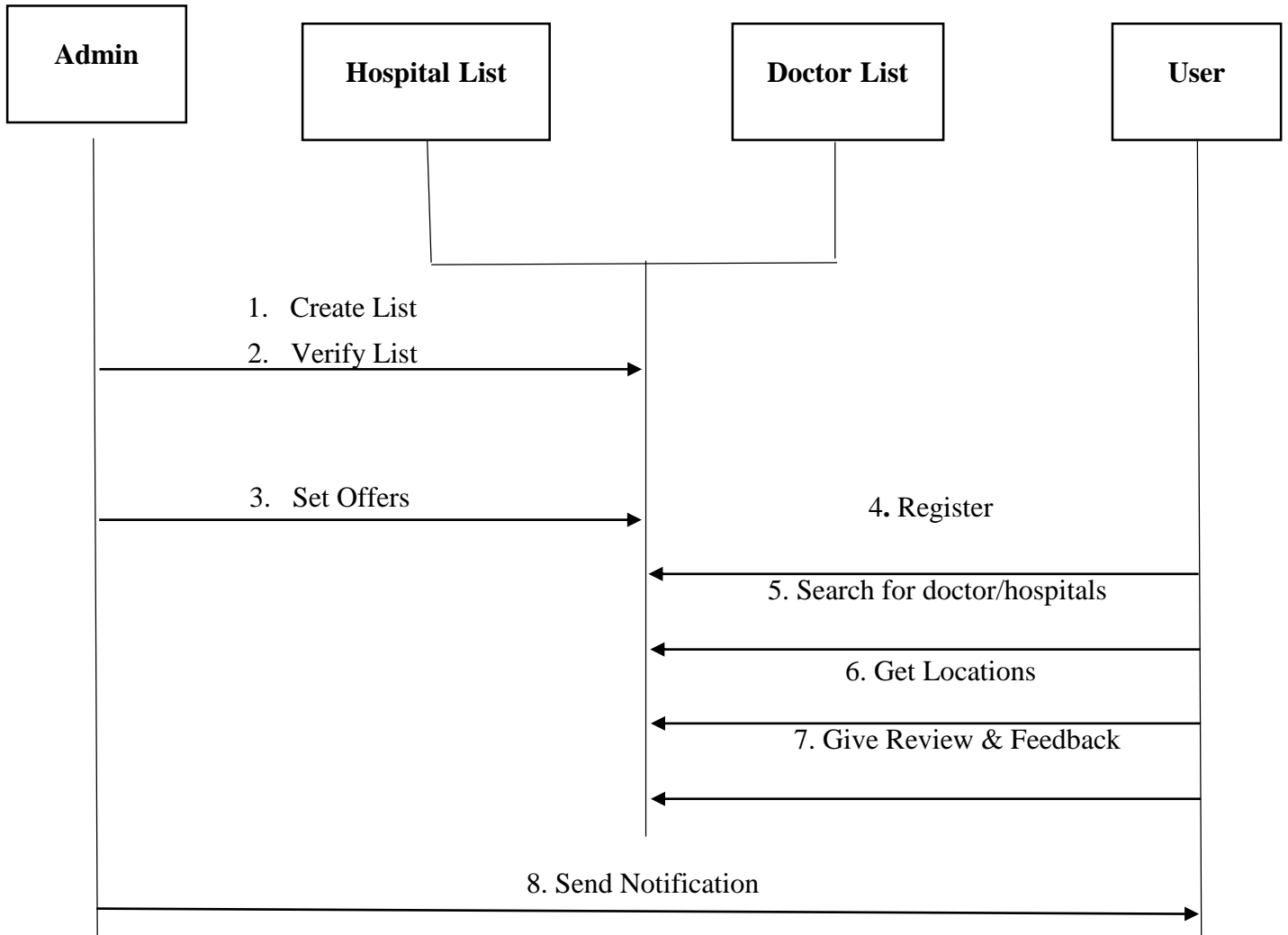


Figure 3.3.4: Sequence Diagram

This is the sequence diagram of the whole process of the both apps. How each action will work and which actor can do what is described here.

3.4 Logical Data Model

A logical data model generally consists of data entities, keys and the attributes and relationships between the entities.

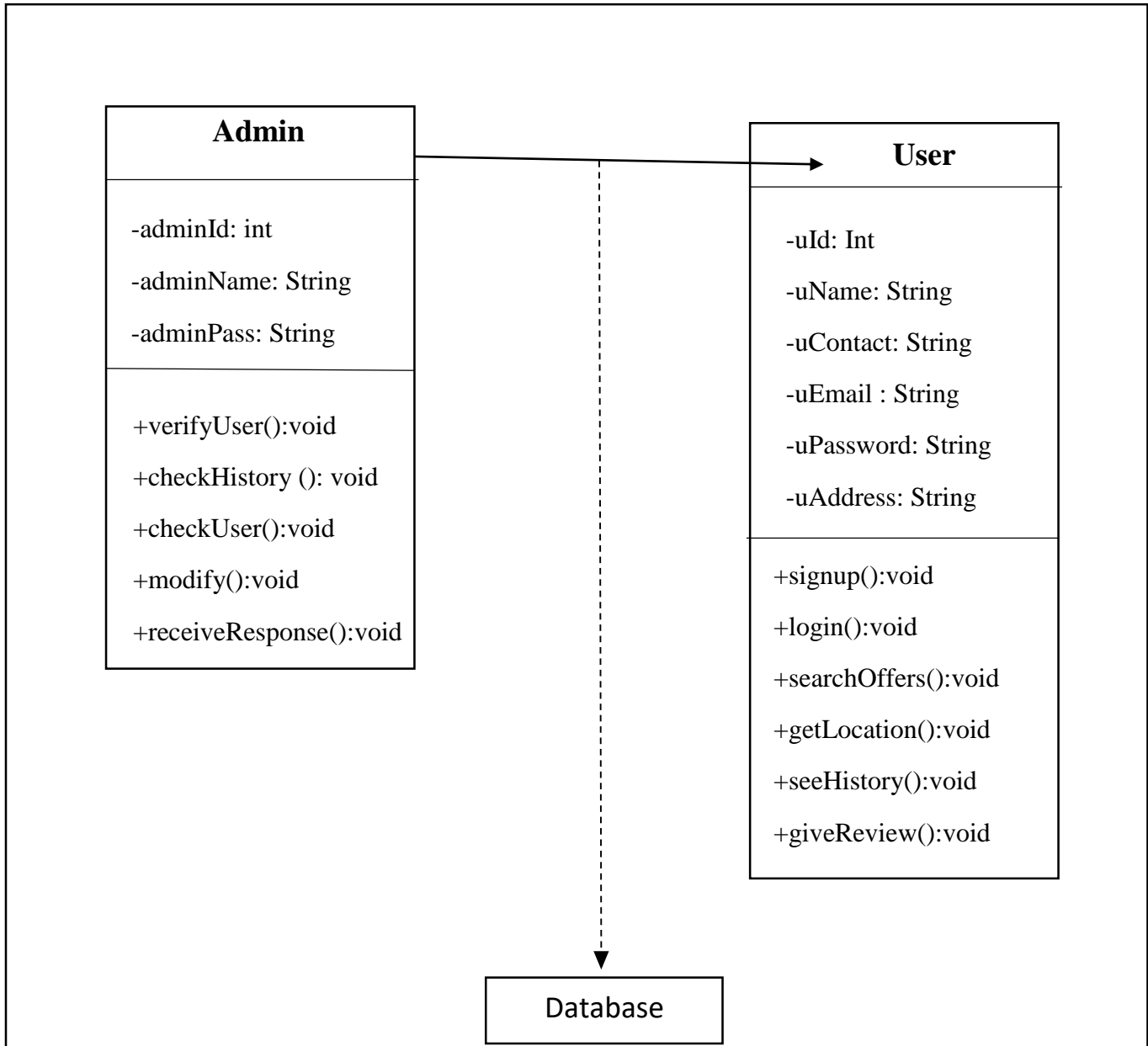


Fig 3.4.1: Class Diagram

Modules

1. Role of Admin

- Verify users
- Assign access rights
- Authenticate users
- Check problems.

2. Server Complaint

- 24/7 up time
- Fast Responses

3. Report Generation

- The system generate report automatically

4. Software and Hardware maintenance

- This system checks the hardware is responding properly or not.
- Also detects unusual activity.

4 Role of user

- Create account
- See offers
- Get hospital, doctor locations
- Refer to friend
- See history
- Give Review

3.4.2 Client Server Architecture

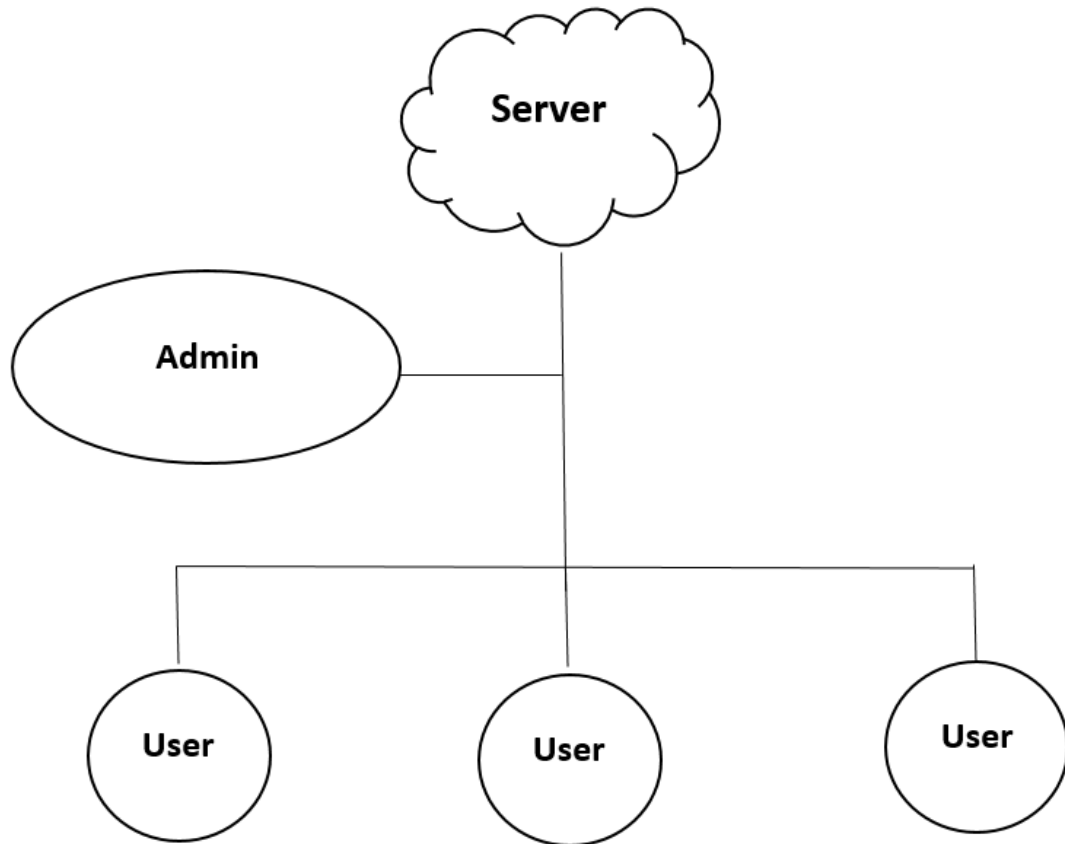


Fig 3.4.2.1: Client Server Architecture

This figure is about the relationship between the leading actors in our mobile payment solution. Here server is balancing connection between merchant and user. The association among user, server and owner are intended to improve adaptability, convenience, versatility and interoperability. Programming adaptability infers the capacity for a program to change effectively as per various clients and diverse framework necessities.

3.5 Design Requirements

To develop a useful application, it requires programming languages. We need to decide which programming language and platform will be best for the application. To develop our project, we choose JAVA for back-end, XML to design the front-end, PHP for server-side work and MySQL for database.

JAVA

Java is an object oriented programming language. It is used to develop the back-end of an application. Now a day, java programming language is the popular language to develop android application. We also used java to develop our project.

There are many advantages of java.

1. It is easy to implement and understand.
2. It an object oriented programming language.
3. It is easy to implement in any platform.
4. It a secure programming language.
5. It is capable to perform several tasks simultaneously within a program etc.

XML

XML stands for extensible Markup Language. It is used to design the front-end of an app which is known as GUI or Graphical User Interface. XML is design to store and transport data. It is designed to be self-descriptive.

MySQL Database

MySQL is a full-featured relational database management system (RDBMS). It is a freely available. MySQL database is used to work in the background of desktop applications, web applications, mobile applications and more. We used MySQL for storing and managing our data.

PHP

PHP now stands for Hypertext Preprocessor. Previously it stands for Personal Home Page. It is a server-side scripting language and it is used to develop dynamic websites or web applications. Though it is a server-side scripting language we used it to connect our android app to the server.

Android Studio

Android studio is the most powerful and popular IDE to develop android application. We also used android studio to develop our app.



Fig 3.5.1: Android Studio Logo

NetBeans

Apache NetBeans provides editors, wizards, and templates to help you create applications in Java, PHP and many other languages. We used NetBeans to maintain the server side works.



Fig 3.5.2: NetBeans Logo

XAMPP

XAMPP is a free and open-source cross-stage web server arrangement stack bundle created by Apache Friends, comprising principally of the Apache HTTP Server, Maria DB database, and translators for contents written in the PHP and Perl programming dialects. We utilized XAMPP to make the database of our task.



Fig 3.5.3: XAMPP Logo

CHAPTER 4

DESIGN SPECIFICATION

Our application mainly works with two main actors. One of them is merchant and another one is user. User need to login to access the system. Admin will get access by authority only. They get data from the database through the application.

4.1 Front Design

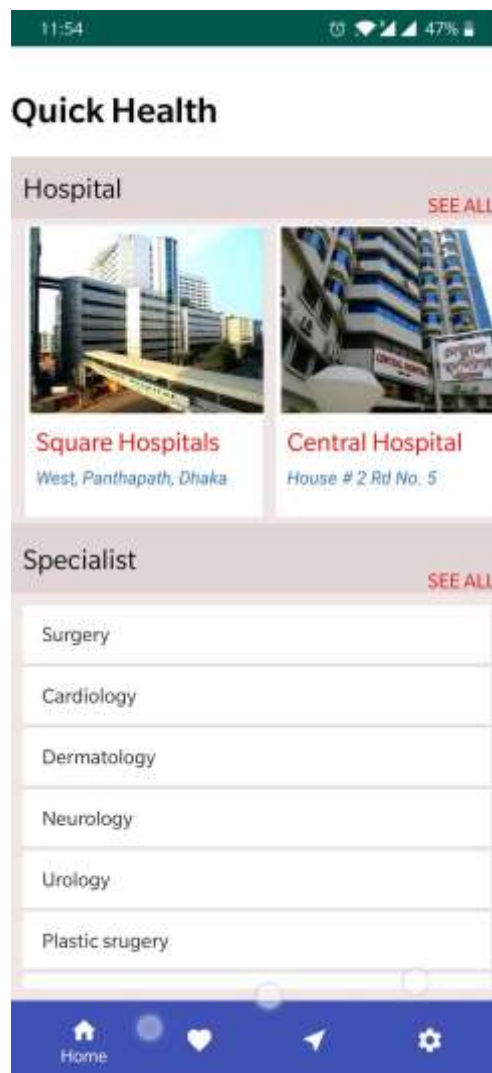


Fig 4.1.1: Homepage

The previous figure shows our main homepage of our app. User open & first page will be this page. On upper side there are hospital lists. On lowest side there is list of type of doctor's user want.

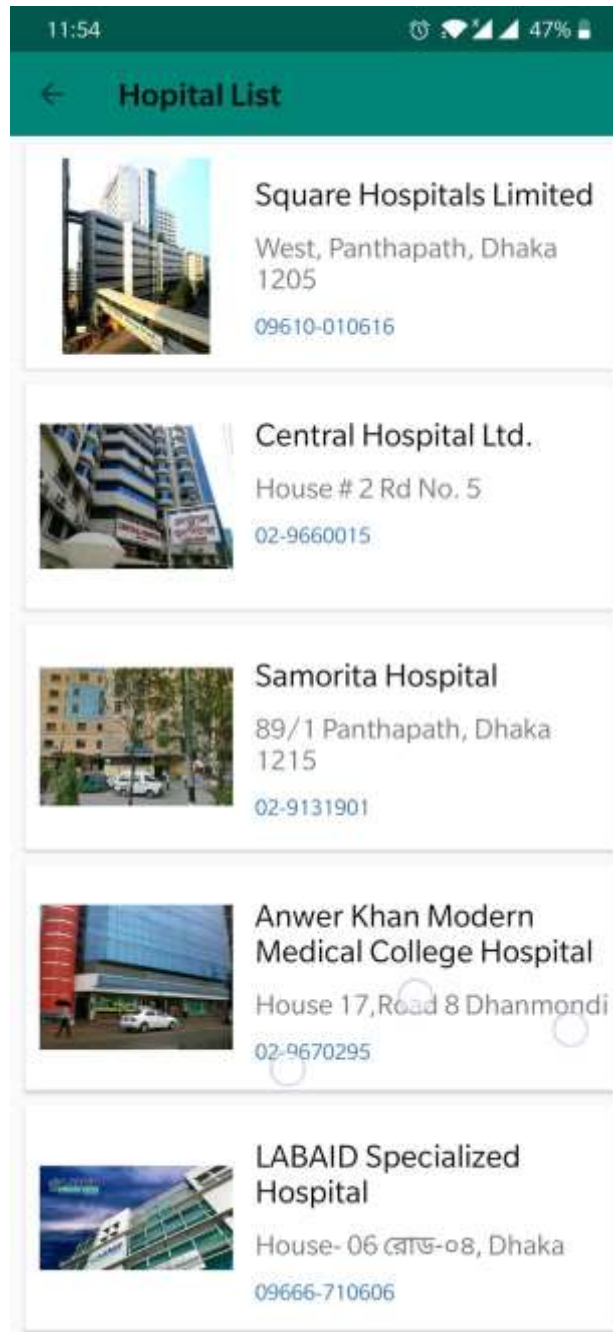


Fig 4.1.2: Hospital List

The previous figure is about lists of hospitals. User can see hospital list, location, contact number here.

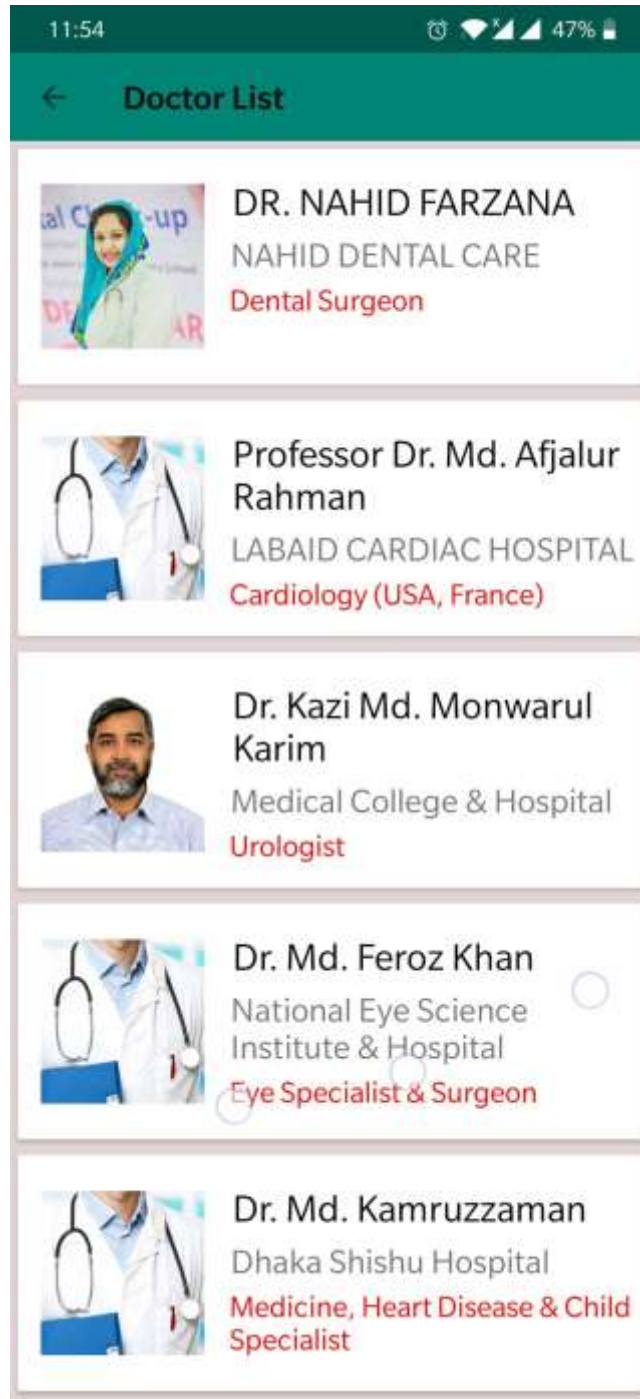


Fig 4.1.3: Doctor Lists

The previous figure shows the doctor list. User can find here what type of doctors they want. Here user can see the doctor's information, location, contact number, ratings, type of specialist etc.

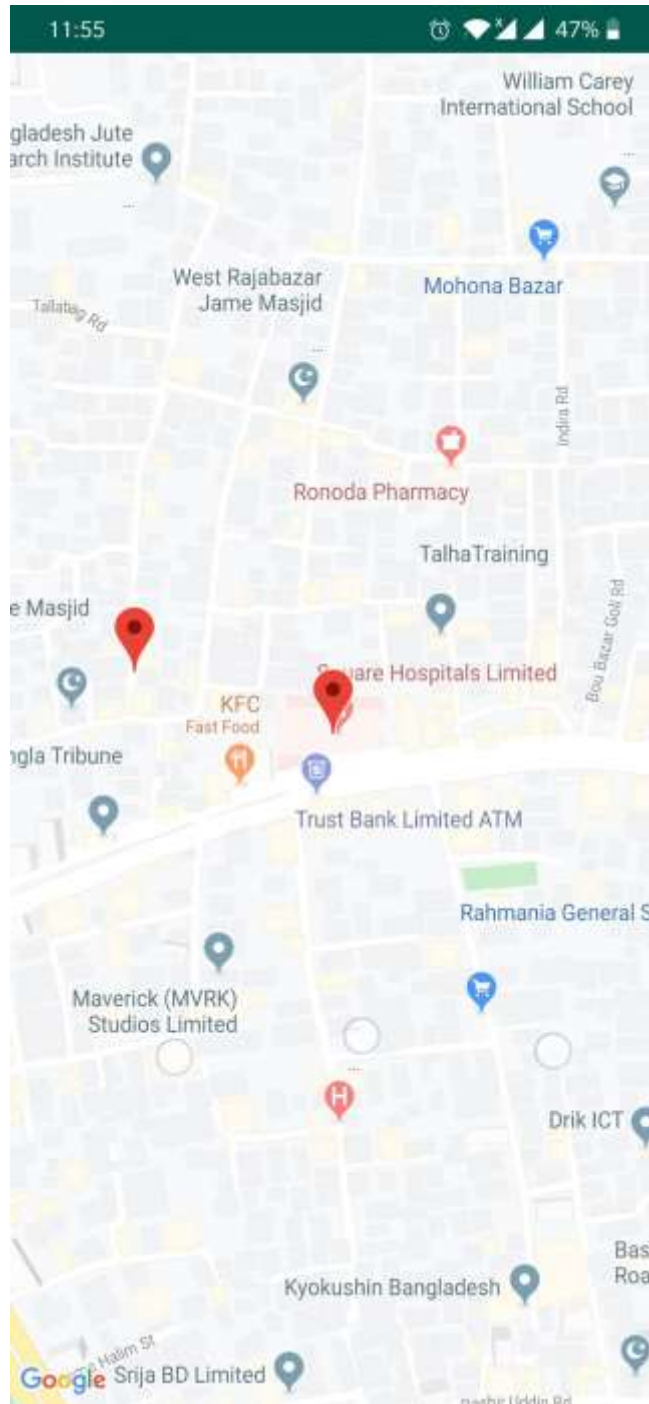


Fig 4.1.4: Near me

The previous figure shows the nearby hospitals from user. User can find here nearest hospital on map. And it's easy to find to located hospitals.

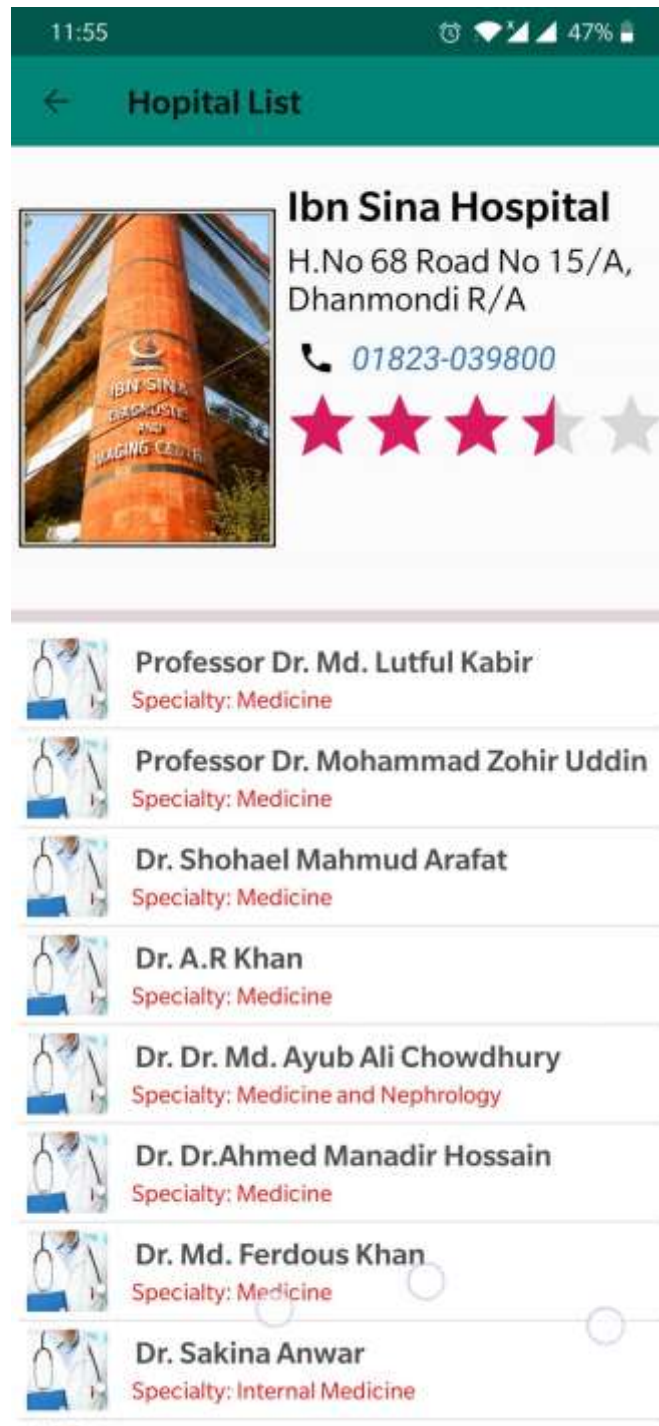


Fig 4.1.5: Hospital Details

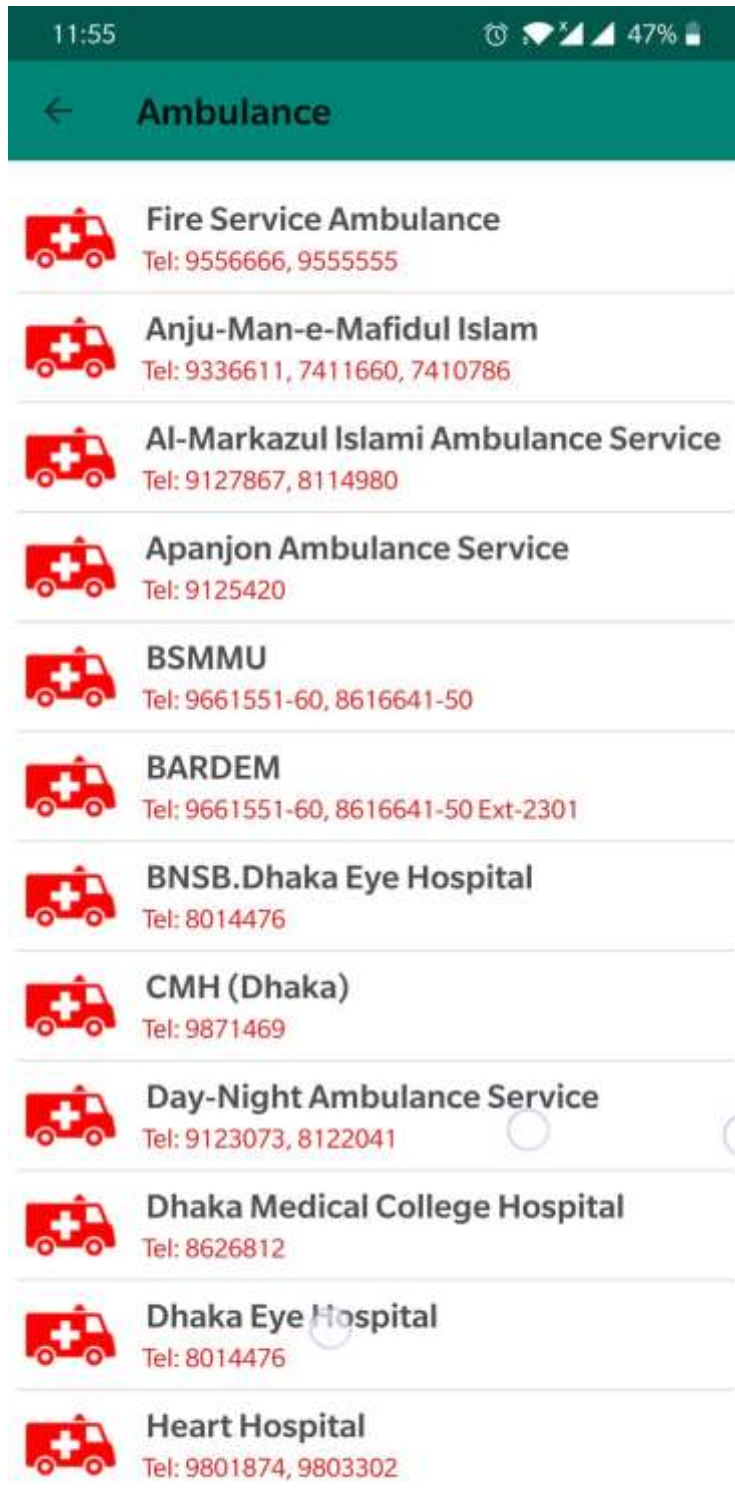


Fig 4.1.6: Ambulance Number

4.2 Back-end Design

The back-end design is the main part of the application. The whole application is dependent on the back-end. We used java to develop the backend and used MySQL to store all the data. The back-end structure is hidden from the merchant and the user. Only admin has access to it.

4.3 Interaction Design and UX

There is an interaction between the client and the application. The correlation performs some tasks according to users' instructions. In our application we have endeavored to give our user some nice feature and highlights. For better user experience, we attempted to keep our application straightforward and simpler to utilize.

4.4 Implementation Requirements

To complete a project, we need to utilize which kind of tools, components are need to build it effectively. So in our application we also utilized few tools and components. In the implementation requirement segment, we examined about all the tools and components that we used to complete our project.

4.4.1 Android Studio

Our project is a mobile application we used android studio as a virtual apparatus. Android Studio is the fastest to develop an application for android device. The main highlights are brilliant code changing, debugging, execution tooling, shaping structure and more. Android Studio IDE is main apparatus that we used. So firstly we must know some fundamental information about Android and Android Studio.

4.4.2 The Emulator

Android application development requires physical or virtual device to run test runs. The emulator helped use to test our app. After that we used physical devices.

4.4.3 Android SDK

In an android application, SDK is one of the very essential tool. SDK or Software Development

Kit contains the libraries for improved environment of an application.

4.4.4 MySQL

We stored our data in MySQL database. We hosted our server on an international server. We generated JSON from the MySQL database to use data for our application. MySQL is easy to use and easy to maintain.

4.4.5 Support Libraries

Android Support Library Core Utilities - 28.0.0

This support library is used to in Android apps to use or improve API services.

Android Multi Dex Library - 1.0.0

This support library is used to in Android apps to use API services that are either not available for the utility version.

Android Support CardView V7 - 28.0.0

This support library is used to work with FrameLayout with rounded corners and shadow based on its elevation.

Android Support RecyclerView V7 - 25.2.0

This view is a flexible and more proficient version of ListView. It used to render large data and scroll smoothly.

Android Volley 1.1.0

Volley is an HTTP Library that makes network related activity easier and faster for the Android app. With this library we can create multiple network connecting and do many more things.

Android Square up Picasso 2.5.2

It is an image library. It is used in android app to create or maintain images in square shape or load it or process it. It helps to process display image from the storage.

GitHub Bumptech Glide

It is used to crop or add filter to the bitmaps and also used to transform animated gifs.

Google GSON 2.3.1

We used to google GSON library to generate values from the server API response. It helped to transfer the data from used to server and server to user.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

This part is about how we designed our database. As we used MySQL for our database.

5.1.2 Normalization

- What we understand by normalization is to eradicate redundancy, which means information has to be stored only once.
- Normalization is the process of breaking a table into some small tables so that each table can work with different purpose.
- If the database is not properly designed, then it will create problem while working in the project. CRUD operation cannot be done properly.
- The database should be normalized up to 3NF. We have normalized our database up to 3NF.

5.2 Implementation of Front-end Design

The number of users are using the system is totally depending on the simple design and ease of the system. Android device is equipped with substantial equipment, a few has larger screen and few has little too. Another thing is android provides update to their OS. For that we need to design it indecently. So that it will support the upcoming OS updates. We tried our best to design the UI, so that user can use every feature by just looking at the symbol.

5.3 Implementation of Interactions

The system is totally relying upon the interaction with the user. Interaction makes a system usable and appealing to the user. It also plays a vital role to make the system intelligent.

We've built some exclusive features to interact our system with the user. We make our system with responsive UI for better user experience. Our system is very easy to use and effectively actualized, so the user can interact easily without any hesitation.

5.4 Testing Implementation

Test implementation is the procedure of testing the implementation of a system, where tester or developer will see different cases and details. That is, it implementable or it has restrictions. We used emulators and real device to get some results. We tested and debugged every feature.

Table 5.4.1: Testing Implementation

Installing Application	We tested our app in many versions of android devices. It installed and ran successfully.
Homepage	Each time we login into our app, our app shows the homepage successfully.
Near Me	Each time we login into our app, near me always shows near hospitals around me.
Hospital List	Each time we login into our app, It shows hospital list provided in our app.
Doctor List	Each time we login into our app, It shows doctor list provided in our app .

5.5 Test Results and Reports

The reason behind testing the app was to find errors. To avoid system failure, we must test our software again and again in different situation and condition. So that the user gets an error free system. To give the user a smooth performance test cases were revised with a purpose in mind. Multiple tests have been applied to find the error. Actually, during testing

phase if the implementations work accurately and efficiently before live operation commences that is a great success as app developer.

By observing the report, we can say that the app is ready to use or not. These are a few facts of testing:

- Installation / Upgrade
- Usability
- System interoperability
- Functionality
- Security
- Performance.

So, the outcomes of these testing result is important. The outcomes are:

- Better application quality.
- User friendly
- App is positively accepted by the users.
- Easy and understandable UI for interactions etc.

5.5.1 Unit Testing

Unit testing was done after the coding stage. This test is connected to discover mistakes in the module. A few changes were finished during the testing stage. At last, every one of the modules were tried start to finish at this stage.

5.5.2 Black Box Testing:

We connected it to check its reaction to a non-specialized individual. The individual did not know anything about the structure and codes. This test helped us to discover basic blunder. It additionally, helped us for sure to improve client ease of use.

5.5.3 White Box Testing

We utilized it to test our application for bugs. The task was a direct inverse of discovery testing. The analyzer thought about everything what is inside. We utilized this strategy for testing to check known blunders.

5.5.4 Integration Testing

When unit testing was finished. Every one of the modules were coordinated for reconciliation testing. Outside and inward interfaces were executed and were worked per structure, the exhibition of the framework isn't debased.

5.5.5 Validation Testing

At the point when unit testing was done. All of the modules were facilitated for compromise testing. Outside and internal interfaces were executed and were worked per structure, the display of the system isn't spoiled.

5.5.6 Acceptance Testing

This was the last phase of testing before framework was at long last acknowledged for operational use. Any necessity issue uncovered in acknowledgment testing were considered here and made blunder free.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This is a challenging project for us. And we are working so hard. There is no conclusion for this project because the end will be another step of beginning. We are planning plenty of feature that will be very useful for everyone. We will add appointment system, medical payment system, Donation, Tissue transplant registration or buying software from verified sites will be included.

6.2 Scope for Further Developments

We are currently focusing on building a user-friendly app that everyone using it easily. And all the information of hospital & doctors will be provided on Dhaka City. We are also planning to establish this service countrywide.

APPENDICES

Online System: The Entire system including medical services.

Android: It is totally android based app.

Doctor Lists: It provide doctors list around our city

Hospital Lists: It has hospital list all over Dhaka City.

Specialist Doctors: User can find their specialist doctor by selected category of doctors.

History: User will have their login history.

Admin: Admin fully maintained the system.

User: User can rate doctors. Also can rate hospitals.

APPENDICES II: Some Providers

- Emedicalpoint blog <http://blog.emedicalpoint.com/find-your-doctors-in-bangladesh/>
- Bangladesh Open Data blog <http://data.gov.bd/dataset/doctor-directory>
- International Medical College & Hospital blog <http://www.imchbd.com/hospital-faculty-member>
- Doc2p International blog <http://www.imchbd.com/hospital-faculty-member>

REFERENCES

- [1] Learn about hospitals of Bangladesh << https://en.wikipedia.org/wiki/List_of_hospitals_in_Bangladesh >>, last accessed on 27 Oct 2019 at 9.00 pm
- [2] Learn about types of doctors << <https://www.webmd.com/> >>, last accessed on 28 Oct 2019 at 10.00 am
- [3] Learn about Tissue Transplant << <https://www.donoralliance.org/> >> , last accessed on 3 Nov 2019 at 11.30 pm
- [4] Learn About Blood Donation << <http://www.bdrcc.org/donate-blood> >>, last accessed on 4 Nov 2019 at 11.00 pm
- [5] Learn about Organ Donation << https://journals.lww.com/transplantjournal/Abstract/2017/08002/Status_of_Organ_Donation_and_Transplantation_in.53.aspx >> , last accessed on 3 Nov 2019 at 10.00 pm
- [6] Learn about specialist doctors in Dhaka << <https://findoutadoctor.blogspot.com/2018/01/find-out-specialist-doctor-dhaka-city.html> >> , last accessed on 2 Nov 2019 at 11.00 pm

quick health service

ORIGINALITY REPORT

22%

SIMILARITY INDEX

9%

INTERNET SOURCES

0%

PUBLICATIONS

21%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	9%
2	Submitted to Christ University Student Paper	2%
3	krazytech.com Internet Source	1%
4	Submitted to University of Bahrain Student Paper	1%
5	Submitted to Asia Pacific International College Student Paper	1%
6	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
7	xbsoftware.com Internet Source	1%
8	Submitted to Universiti Tenaga Nasional Student Paper	1%
9	Submitted to Manchester Metropolitan University	<1%

10	www.techopedia.com Internet Source	<1%
11	netbeans.apache.org Internet Source	<1%
12	www.ittrainingnepal.com Internet Source	<1%
13	Submitted to Midlands State University Student Paper	<1%
14	Submitted to Thames Valley University Student Paper	<1%
15	www.novell.com Internet Source	<1%
16	Submitted to University of Greenwich Student Paper	<1%
17	Submitted to Westminster International University in Tashkent Student Paper	<1%
18	Submitted to The British College Student Paper	<1%
19	Submitted to University of Southern Queensland Student Paper	<1%
20	Submitted to Aberdeen College Student Paper	<1%
