



03-134142-075 Ovais Butt

03-134142-099 Fahad Ahmad

Medicure-Your Health Assistant

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Zia Ul Rehman

Department of Computer Sciences
Bahria University, Lahore Campus

June 2018

Certificate



We accept the work contained in the report titled
“MEDICURE-YOUR HEALTH ASSISTANT”,

written by

Ovais Butt

Fahad Ahmad

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor: Mr. Zia Ul Rehman

(Signature)

June 4th, 2018

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134142-075	Ovais Butt	
03-134142-099	Fahad Ahmad	

Date : _____

Specially dedicated to
my beloved grandparents, mother and father & siblings
(Ovais Butt)
my beloved grandparents, mother and father & siblings
(Fahad Ahmad)

ACKNOWLEDGEMENTS

Starting with the name of Allah (S.W.T), first we would like to thank Allah (S.W.T) very much for giving us strength to complete this project. We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude from the depth of our heart to my research supervisor, Mr. Zia Ul Rehman for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parents who stood beside us in the every hour of need, our friends who had helped us in completing this project and given me encouragement.

Our teachers who taught us all from the start and helping us with all our queries and ambiguities so that we could bring this idea into real shape and last but not the least all the doctors who helped us in understanding the diseases and other medical terms.

Ovais Butt
Fahad Ahmad

MEDICURE-YOUR HEALTH ASSISTANT

ABSTRACT

MediCure is a medical expert system that works as a general physician and diagnoses the diseases a patient might have after interacting with the patient and asking him what he feels and what symptoms he thinks he has. Furthermore, this App performs many more activities like saving patient's medical profile, allergies (if any), medical background record, setting up notifications and alerts for Diet, Medicine & Appointments, getting info regarding diseases, appointment with the doctor related to the disease, the user can keep track of the images of medical records & can also make a directory of his doctors.

This project will aware the patients about their diseases, can remove the gap between patients and doctors, and can facilitate the patients with the best nearest available doctor related to their disease without any hassle.

For making a medical chat bot we will use an appropriate algorithm and build a dataset of symptoms and diseases on which the chat bot will be trained, a database about the doctors will be generated which will tell all their contact information etc. and in what field they are practicing. We are working with doctors to make this happen.

TABLE OF CONTENTS

DECLARATION	iii
ACKNOWLEDGEMENTS	vii
ABSTRACT	ix
TABLE OF CONTENTS	x
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF SYMBOLS / ABBREVIATIONS	xvii

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Aims and Objectives	2
	1.4 Scope of Project	3
2	LITERATURE REVIEW (and/or SRS)	5
	2.1 Product Perspectives	5
	2.2 User Classes and Characteristics	5
	2.3 Operating environment	6
	2.4 Design and Implementation Constraints	6
	2.5 Assumptions and Dependencies	6
	2.6 User Interfaces	6
	2.7 Hardware Interfaces	7
	2.8 Software Interfaces	7
	2.9 Communication Interfaces	8
	2.10 System Use Cases	8
	2.10.1 Use case name and identifier	8

2.11	Other Non-functional Requirements	9
2.11.1	Performance Requirements	9
2.11.2	Safety Requirements	10
2.11.3	Security Requirements	10
2.11.4	Software Quality Attributes	10
3	DESIGN AND METHODOLOGY	11
3.1	Use Case Diagrams	11
3.1.1	User Authentication	11
3.1.2	Use Case Description	12
3.1.3	User Interaction with application	13
3.1.4	Use Case Description	14
3.1.5	Fix Appointment with the doctor	15
3.1.6	Use Case Description	15
3.2	Domain Model	17
3.3	Sequence Diagram	18
3.3.1	Sequence Diagram Description	19
4	DATA AND EXPERIMENTS	25
4.1	Input Event	25
4.2	Tools & Technologies	25
4.2.1	Develop for Android	25
4.3	Android Activity Lifecycle	26
4.4	Android Architecture	26
4.4	External APIs/ Libraries used	27
4.4.1	Permissions	27
4.4.2	Database & Hosting	27
4.5	System Testing	27
4.5.1	Software Testing	27
4.5.2	Unit Testing	28
4.5.3	Integration Testing	28
4.5.4	Acceptance Testing	28
4.6	Test Cases	29

4.6.1	Test Case 1: User Login	29
4.6.2	Test Case 2: Profile Update	30
4.6.3	Test Case 3: Patient Assessment	30
4.6.4	Test Case 4: Notification Alerts	31
4.7	Test Case Results	31
4.7.1	User login result	31
4.7.2	Profile Update Result	32
4.7.3	Patient Assessment Result	32
4.7.4	Notification Alerts Result	33
5	RESULTS AND DISCUSSIONS	35
5.1	General Information	35
5.2	Overview	35
5.2.1	System Summary	35
5.2.2	System Configuration	36
5.2.3	User Access level	36
5.2.4	Understanding the main menu	37
5.2.5	Starting Application	37
5.2.6	Start Assessment	38
5.2.7	Profile	39
5.2.8	Basic Information	39
5.2.9	Allergies	40
5.2.10	Health Background	40
5.2.11	Prescribe Treatment	41
5.2.12	Set notification alerts	41
5.2.13	Medicine Alerts	42
5.2.14	Diet Alert	42
5.2.15	Appointment Alert Activity	43
5.2.16	Medical records	43
5.2.17	Search Doctor	44
5.2.18	Appoint Doctor	44
5.2.19	My Doctor	45
5.2.20	About Us Activity	45

		xiii
	5.2.21 Logout	46
6	CONCLUSION AND RECOMMENDATIONS	47
	6.1 Conclusion	47
	6.2 Recommendations	48
	6.3 Learning Outcomes	48
	REFERENCES	49

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 1: Test Case User Login	29
	Table 2: Test Case Profile Update	30
	Table 3: Test Case Patient Assessment	30
	Table 4: Test Case Notification Alerts	31

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1:	System Use Case	8
Figure 2:	User Authentication	11
Figure 3:	Use case diagram of application	13
Figure 4:	Use case diagram of Fix Appointment	15
Figure 5:	Domain Model Diagram	17
Figure 6:	Sequence Diagram	18
Figure 7:	Collaboration Diagram	20
Figure 8:	Class Diagram	21
Figure 9:	Entity Relationship Diagram	22
Figure 10:	Android Activity life cycle	26
Figure 11:	Android Architecture	26
Figure 12:	User Login Test Result	31
Figure 13:	Profile Update Test Result	32
Figure 14:	Patient Assessment Test Result	32
Figure 15:	Notification Alerts Test Result	33
Figure 16:	User Access level	36
Figure 17:	Main Menu Activity	37
Figure 18:	Starting Application	38
Figure 19:	Start Assessment Activity	38

Figure 20: Profile Activity	39
Figure 21: Basic Information Activity	39
Figure 22: Allergies Activity	40
Figure 23: Health Background Activity	40
Figure 24: Prescribe Treatment Activity	41
Figure 25: Notification Alerts Activity	41
Figure 26: Medicine Alert Activity	42
Figure 27: Diet Alert Activity	42
Figure 28: Appointment Alert Activity	43
Figure 29: Medical Records Activity	43
Figure 30: Search Doctor Activity	44
Figure 31: Appoint Doctor Activity	44
Figure 32: Add My Doctor Activity	45
Figure 33: About Us Activity	45
Figure 34: Logout Activity	46

LIST OF SYMBOLS / ABBREVIATIONS

<i>MediCure</i>	Medical Cure
<i>Chat Bot</i>	Chatter bots
<i>AI</i>	Artificial Intelligence
<i>Ghz</i>	Gigahertz

CHAPTER 1

INTRODUCTION

1.1 Background

Medical expert System is an application based on artificial intelligence. It is a set of programs to manipulate knowledge to solve problems within a specialized domain that necessitates human expertise. An expert system is also called knowledge based system. MediCure is a quick way to get aware of the common diseases which are effecting many people now a days, the diseases are diagnosed according to the symptoms from which the patient is suffering.

It helps to give the initial precautionary measures regarding the particular disease. The most important thing in this application is that it will also suggest medicines on demand for general diseases like flu, fever, cough etc. Regular updates and notifications will be generated for taking medicines, eating the meals on proper time and getting reminders about the appointments and the details of the doctor with which the appointment is.

Moreover, it will also have an option of going through the records of doctors so that a patient can select a doctor of his requirements to consult and fix the appointment with the doctor for regular check-up. App will also keep proper record of the patient from disease to the treatment of disease. It can also keep the previous record of the patient if any so that even the new doctor appointed may know about the previous medical history of the particular patient.

The most important features of expert system are user interface, data representation, inference, explanations and advantages of expert system are fast response, increased reliability, reduced cost, reducing errors, multiple expertise, intelligent database, reduced danger.

1.2 Problem Statements

There are multiple problem statements which lead us of an idea for this project some of them are that people don't feel it feasible to visit doctor so often they just ignore common symptoms which may lead them to serious diseases.

Too much time is wasted on waiting in queues for the doctor to check them up or one may find it difficult to find a reliable doctor for his disease people have to consult others to refer them a doctor, patients often forget to take medicines, diets on time and go for their appointments on time.

Patients also face difficulties in keeping their previous or day to day medical records in one place or remembering their doctors.

1.3 Aims and Objectives

The objectives of the thesis are shown as following:

- Design and develop a mobile application based medical expert system which can diagnose common diseases such as fever, flu, headache etc.
- Diagnose diseases, can remove the gap between patients and doctors.
- Facilitate the patients with the best nearest available doctor related to their disease.
- Set day to day reminders for the patients.

- Save all their medical data at one place so that they might not face any difficulty when they need it.

1.4 Scope of Project

MediCure is an android based application which is a medical expert system and cure patients like a general physician where an Intelligent Bot named MEDI will ask the patients about the symptoms. Normal diseases like gastro, headache, vomiting, diarrhoea, malaria will be diagnosed, the diseases which require proper doctor's assistance and need proper testation to cure the disease will not be treated plus this app is not for emergency purposes. Here are the steps how the patient will land on the app and how will the app respond to its activities.

- User will login and will provide initial data like age, gender, weight etc.
- An AI based expert system Bot will interact and ask patient what he feels.
- Based on the symptoms app will generate keywords.
- Medi will then ask questions related to that particular symptom told by the user previously.
- User will have to reply with yes or no.
- At the end the bot will come up with one disease.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Product Perspectives

MediCure is being developed to facilitate the users to get knowledge about the symptoms they are facing, that what kind of disease it can be and what are the precautionary steps the patient has to take in the start so that he can cure it in start more over it also facilitates the users with the medicines to take and makes reminder alarms for them so that they take their medicines on time.

The app also facilitates with fixing appointments with the doctors. The patients can also save their medical history reports just in case to bring ease while consulting with any different doctor.

2.2 User Classes and Characteristics

Patients: Patients are the users who want to check what disease they might have, to keep the records of their medical history, to set medicine and diet alarms and notifications, to fix an appointment with the related doctor. They might be anyone using an android phone.

Doctors: Doctors are the users who want to work as freelancers. They can upload their profiles on the App where the patients can contact them and can fix appointments with them with ease.

2.3 Operating environment

This application will run on Android phones, android versions of KitKat and above will support this application.

2.4 Design and Implementation Constraints

MediCure is developed in Java (for back-end), PHP, MYSQL & SQLITE (for server-side scripting) & XML (for front-end) using android studio AIML is used for bot making. All of the modules are interlinked with each other whereas a single module can be used separately.

2.5 Assumptions and Dependencies

As the App is developed in java for only the android devices so a proper android (java) environment is required to run the Application which means a proper android mobile phone is needed to install and run the application.

2.6 User Interfaces

Splash Screen: The user will first see a splash screen which is just a welcome screen with App logo on it. The logo animates for few milliseconds and then it navigates the app to the login screen.

Registration Page: The new user will have to register in order to use the App, he will provide the app with his name, email address, password and mobile number. Later on he will have to login for the app just for the first time.

Login Screen: The login screen asks for credentials like username and password and takes you to the home screen where the user can see the chat page.

Home Screen: The home screen consists of a chat page where the chat bot named Medi interacts with the patient and asks for the symptoms then by analysing those symptoms it generates the best related disease. The home screen also contains a navigation bar which slides from left to right and contains following things in it.

Setting up notifications & alerts: The patient can set the medicine, diet and appointment alerts. So that it gets notified on time.

Profile Activity: Here the user will be able to save some of his medical profile, his allergies and medical background of the user.

Appointment: The patient can also fix an appointment with the doctor if he still wants to consult the doctor.

Previous Medical Records: The user can also save his previous medical record just to remove any problem while consulting with a different doctor.

2.7 Hardware Interfaces

The only hardware used in this project is a mobile phone which must have an android OS installed in it. Mobile must have minimum 1GB of ram, 8GB of storage and 1.2 Ghz. of Quad Core processor.

2.8 Software Interfaces

MediCure requires an android OS (minimum KitKat) to be installed on the target device which means the java is the main software interface required to run this Application as android runs purely on java.

2.9 Communication Interfaces

MediCure requires an active internet connection to communicate with the databases to store and fetch data from there.

2.10 System Use Cases

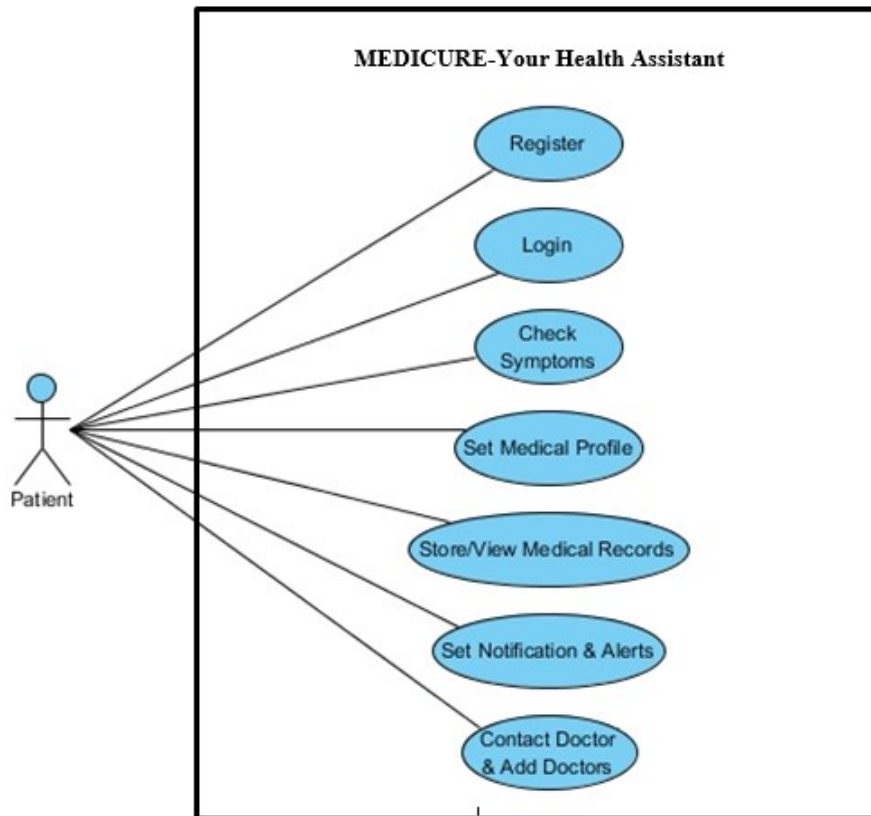


Figure 1: System Use Case

2.10.1 Use case name and identifier

Functional Requirements

- Register
- Login
- Check Symptoms

- Setup Medical Profile
- Set Notifications & Alerts
- Fix an appointment with the Doctor
- Store/View Medical Reports (history)

Actors

Patients: Users who want to check what disease they might have, to keep the records of their medical history, to generate medicine and diet plan, to fix an appointment with the related doctor. They might be anyone using an android phone.

Doctors: Doctors who want to work as freelancers. They can upload their profiles on the App where the patients can approach them.

Preconditions

- A user must have an android phone.
- User must install the application first.
- The user must have a running internet connection.
- A user must first register.
- If a user has logged in once then he would not login again.
-

Post conditions

- There are no such post conditions of the App.

2.11 Other Non-functional Requirements

2.11.1 Performance Requirements

MediCure requires a mobile phone which must have an android OS installed in it. Mobile must have minimum 1GB of ram, 8GB of storage and 1.2 Ghz. of Quad Core processor Graphics card is not a must in it because the App contains efficient graphics level. Screen size may vary and the app runs in all screen sizes from 4.7” to 6.0”.

2.11.2 Safety Requirements

In order to provide the users with the best app experience we bring time to time updates in our apps to prevent any bugs and fixes and to provide our patients with the best user experience.

2.11.3 Security Requirements

The Patient/user must provide their email, phone number and password to login. This will later be used as the user credentials and later on can be used in password reset and etc.

2.11.4 Software Quality Attributes

MediCure provides their users with both simple and advanced features. Due to its well designed and easy to use interface and frequent module transitions it can be used by both experts and typical users. However, users must already have a basic knowledge of using an android phone before using it.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Use Case Diagrams

Use Case diagrams are designed to understand which actor will interact with which use case in which way. Following are the high priority use case diagrams in my project:

3.1.1 User Authentication

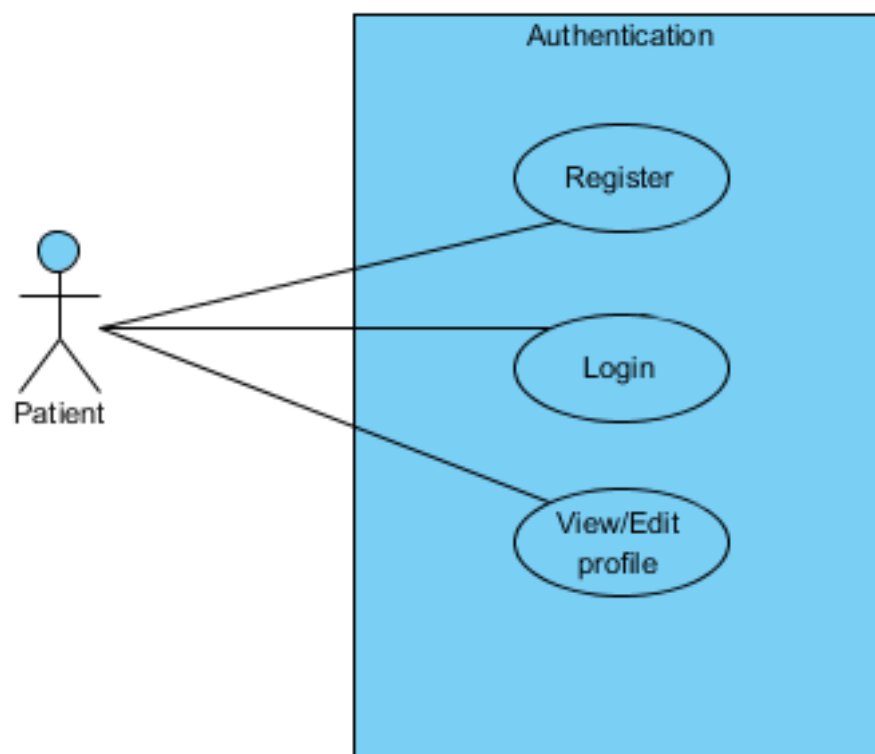


Figure 2: User Authentication

The Use Case in figure 2 is about the user authentication here the user can either register if he do not have the account in the app or he can just login to the app for registration the user must provide the app with the user name, email, password and the phone number once the user provides all the information the user can the register or can make an account in the app then the app will ask the user to login for one time only in order to verify the account credentials for login the app will require the email of the user and the password. The user can also view or make changes in the profile like changing the password in the profile option

3.1.2 Use Case Description

Figure Number

- Figure 3.1.1 Use case diagram of authentication

Brief Description

- User will register itself in the app and login at same time for first time.

Priority

- High

Source

- Patient, Doctor

Participating Actors

- Patient, Doctor

Flow of Events

- User starts the application.
- User enters all the fields required for registration.
- Account is created for the user.
- Users logins to the App by providing the same credentials

Alternative Flows

- Application didn't start.
- Android version is older than the required one.

Pre Conditions

- User must have an android running smart phone.

- User must have a valid email address.
- User must have the application installed in the phone.

Post Conditions

- The user will be successfully logged into the application.

3.1.3 User Interaction with application

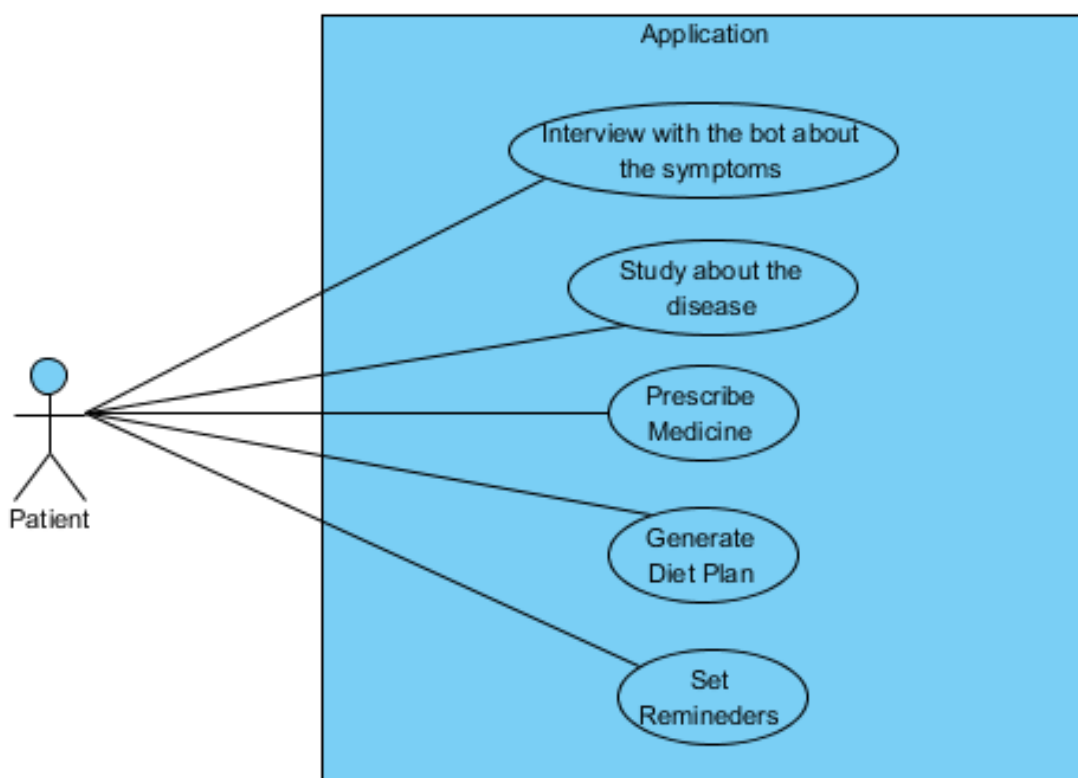


Figure 3: Use case diagram of application

The Use Case in figure 3.1.2 is about the user interaction with the application that how the user can walk through the app and how he can perform multiple tasks in the application. The user may perform multiple operations like he can talk with the bot any by telling the symptoms. He can diagnose what disease he is having this process goes like the user will just give the keyword what the user feels like if he has a head ace he will write that down and the bot will ask some questions regarding head ace and the user will reply with yes no and maybe and upon these answers the bot will deduce

the result which will be disease, he can also study about a specific disease and get to know about the do's and don'ts in the disease etc. then the user can also set reminders for the medicine, diet and the appointments with the doctors.

3.1.4 Use Case Description

Figure Number

- Figure 3.1.2 Use case diagram of application

Brief Description

- Patient will interact with the bot to know what disease he might have.

Priority

- High

Source

- Patient

Participating Actors

- Patient, Doctor

Flow of Events

- Patient tells the symptoms.
- Chat bot asks questions.
- Patient responds to those questions.
- Chat bot tells about the disease according to the symptoms.

Alternative Flows

- Patient's symptoms do not match from the database.
- Disease is not diagnose.

Pre Conditions

- Patient provides his basic info like gender, age, height and weight.
- Patient must know what he feels in sickness.
- Patient must know basic medical terms.
- Patient must be familiar with the allergies he have.

Post Conditions

- The patient will be successfully know the disease.

3.1.5 Fix Appointment with the doctor

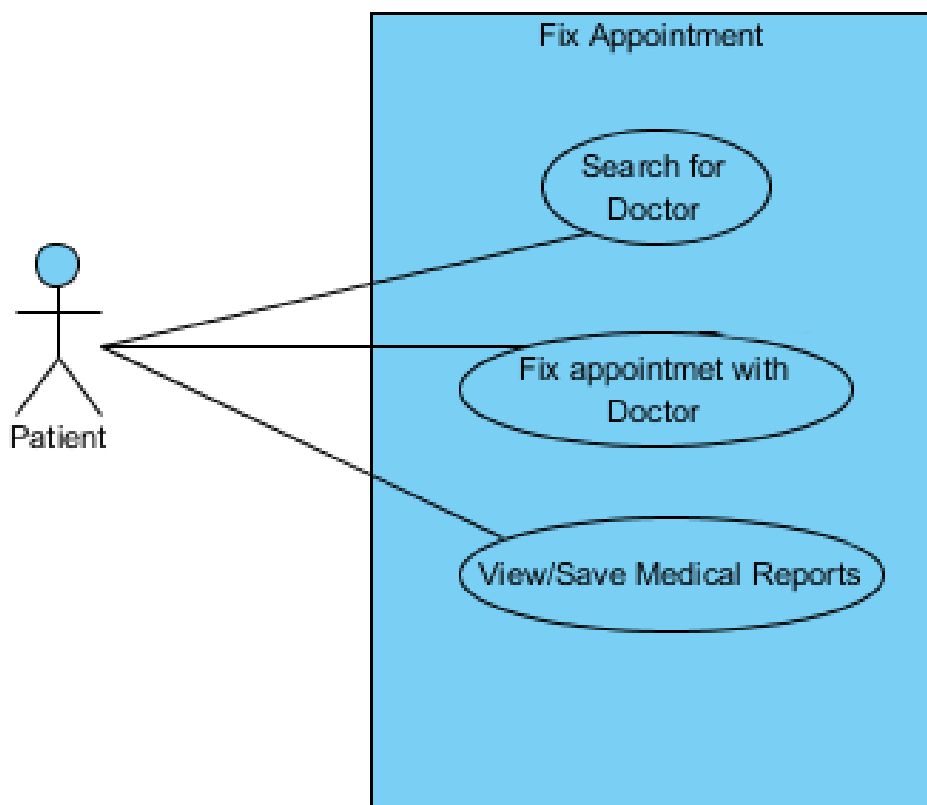


Figure 4: Use case diagram of Fix Appointment

The Use Case in figure 3.1.3 is about the user fixing appointment with the doctor, in this use case the user can perform 3 operations such as searching for a doctor fixing appointment with the desired doctor and saving the previous medical record in the image form and they can be viewed after wards. The user can search for the doctor by the disease he want to consult for and he can view the details of the doctors like the doctor timings, address, fees and contact info the user can leave a message or can call the doctor to fix appointment.

3.1.6 Use Case Description

Figure Number

- Figure 3.1.3 Use case diagram of Appointments

Brief Description

- Patient can fix appointment with the doctor.

Priority

- High

Source

- Patient

Participating Actors

- Patient, Doctor

Flow of Events

- Patient will search for Doctors.
- List of doctors will be generated.
- Patient can check doctors and contact the doctor he likes.

Alternative Flows

- Patient cannot find the doctor he needs.

Pre Conditions

- Patient must know what disease he has in order to select a doctor.

Post Conditions

- The patient will be successfully fix appointment with the doctor.

3.2 Domain Model

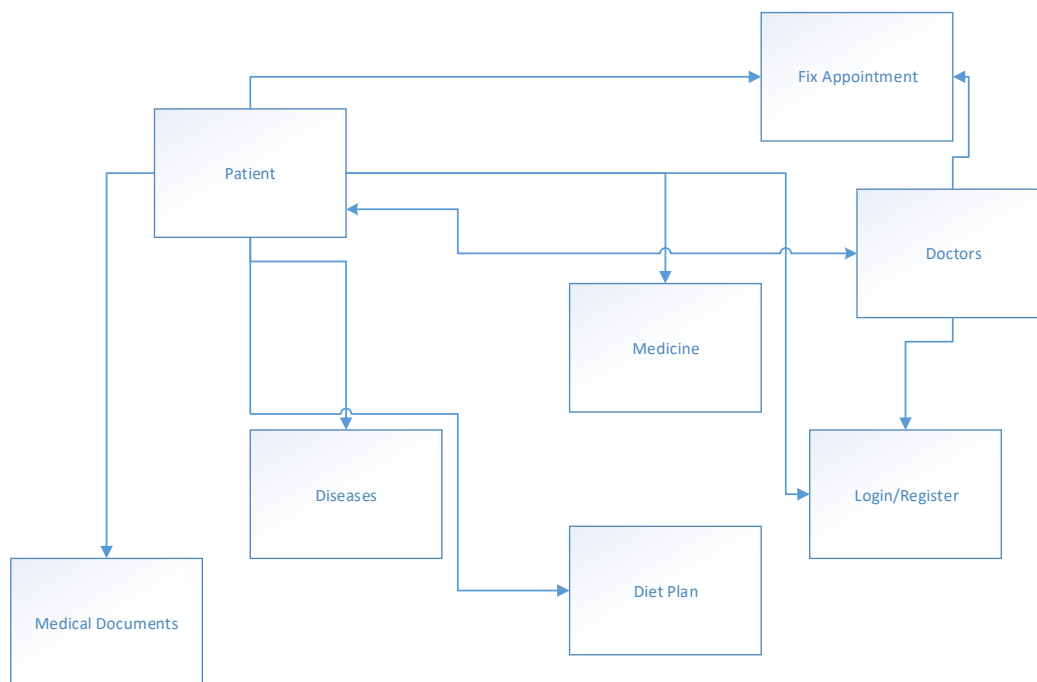


Figure 5: Domain Model Diagram

Domain model fig 3.2 of this project describes the domains and the modules which are included in the project and the interactions between all the domains included in it. In this model patient is the main domain which is interacting with all other entities regarding whether it is for the assessment of the disease or generating the diet plan or medicine plan of the patient. Each entity is connected with each other as shown in the model and have respective relationship according to the functionality of the module.

3.3 Sequence Diagram

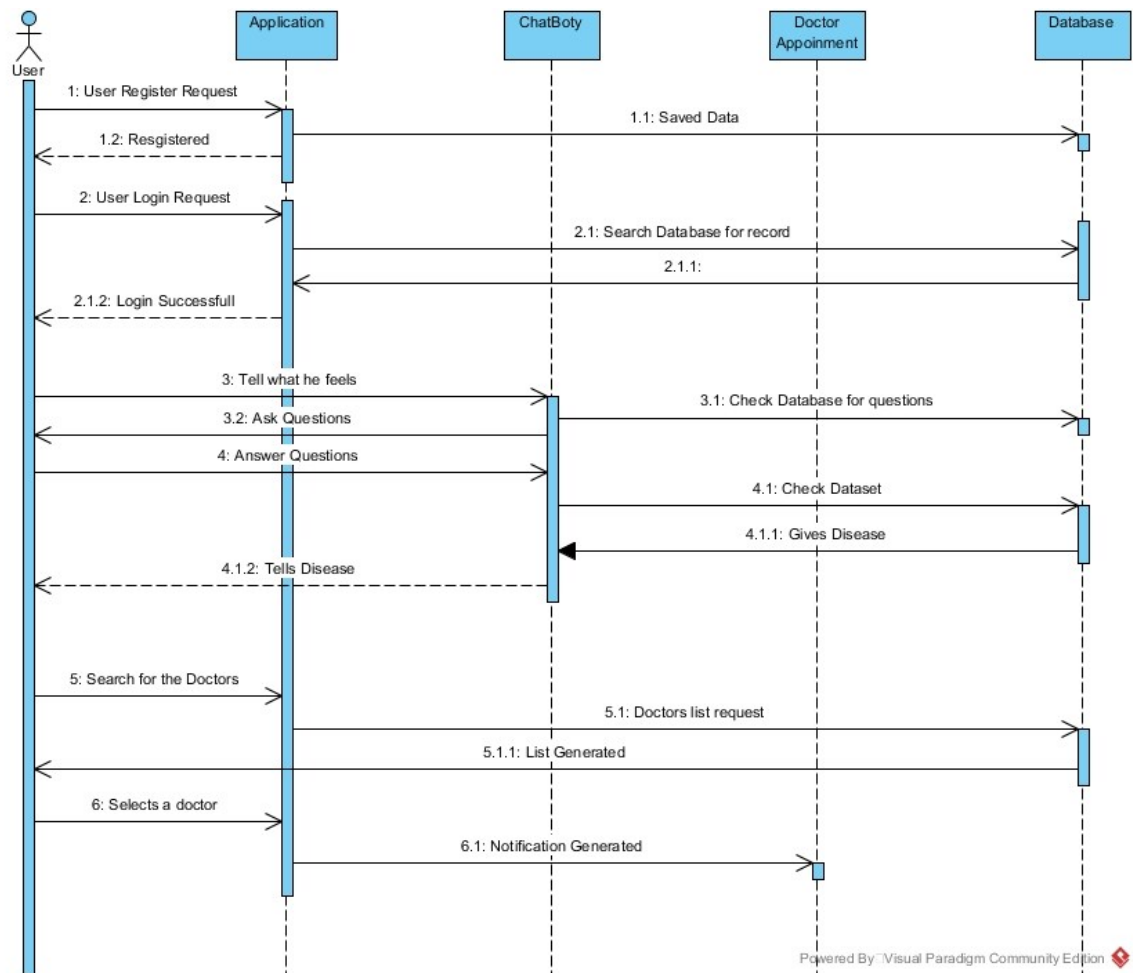


Figure 6: Sequence Diagram

Figure 3.3 is the sequence diagram about the high priority use case the most high priority task in this project is to get diagnosed by the chat bot and fix appointment with the doctor this sequence diagram starts with the login or registration of the user either the user should register for the app which is done in the way that the user first inputs the data and the data is then saved in the database the user then logs in with the same credentials he provides in the registration phase in the login operation the app verifies the data provided in the text fields from the database if the application finds the data from the database it logs the user in the app the user then tells the chat bot what he feels and on that keyword the chat bot asks relevant questions from the database the user has to reply those questions with yes, no and maybe based on the answers the chat bot predicts the disease. The user can also search for the doctor and choose the doctor based on his requirements and can fix appointment with the desired doctor.

3.3.1 Sequence Diagram Description

Actors

- User (Patient)

Class Roles or Objects

- Application
- Chat Bot
- Doctor Appointment
- Database

Objectives

- Diagnose the disease
- Fix appointment with the doctor

Messages

- Register & Login
- Tell symptoms to the chat bot
- Select a doctor for appointment

3.4 Collaborative Diagram

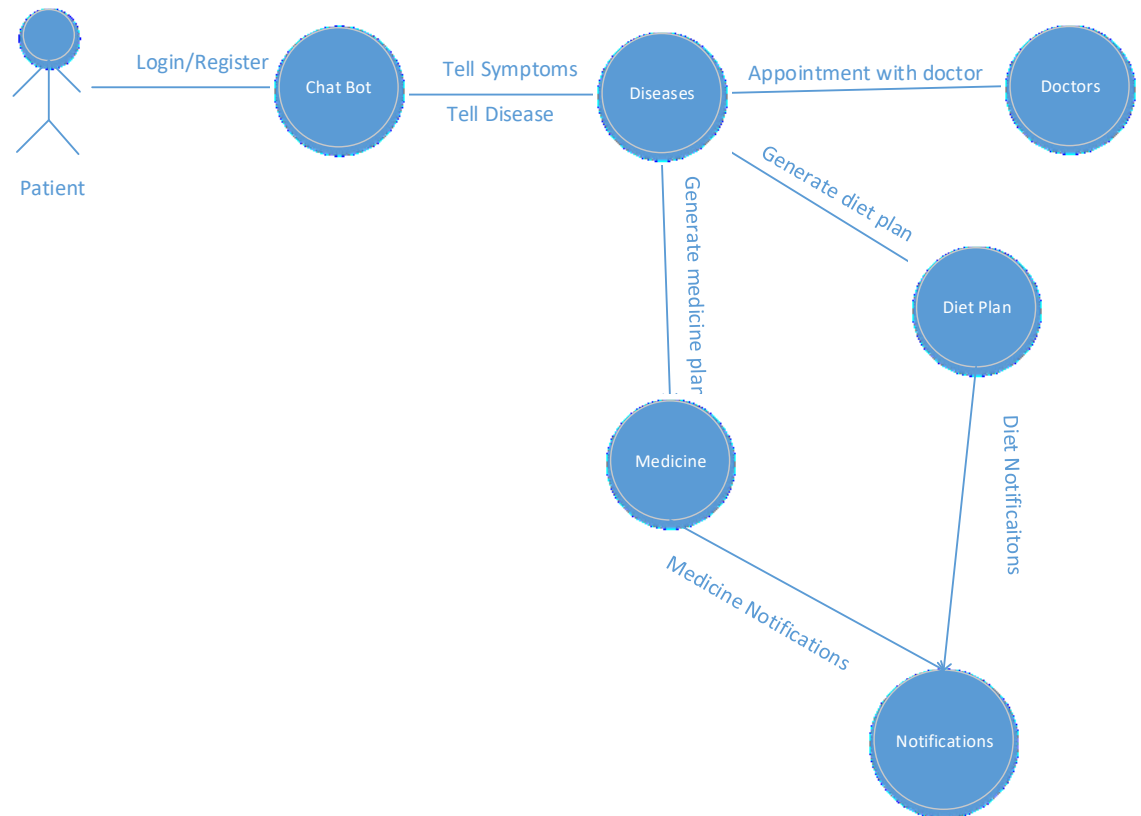


Figure 7: Collaboration Diagram

Collaboration diagram figure 3.4 gives the description of the collaboration of the entities interacting within the application. It shows that user will first have to login and register itself to the app then afterwards a chat bot will interact with the user and ask about the symptoms of the diseases and on the basis of those symptoms further actions will be performed regarding to the appointment with doctor or the medicine which will be suitable for the respective disease.

3.5 Class Diagram

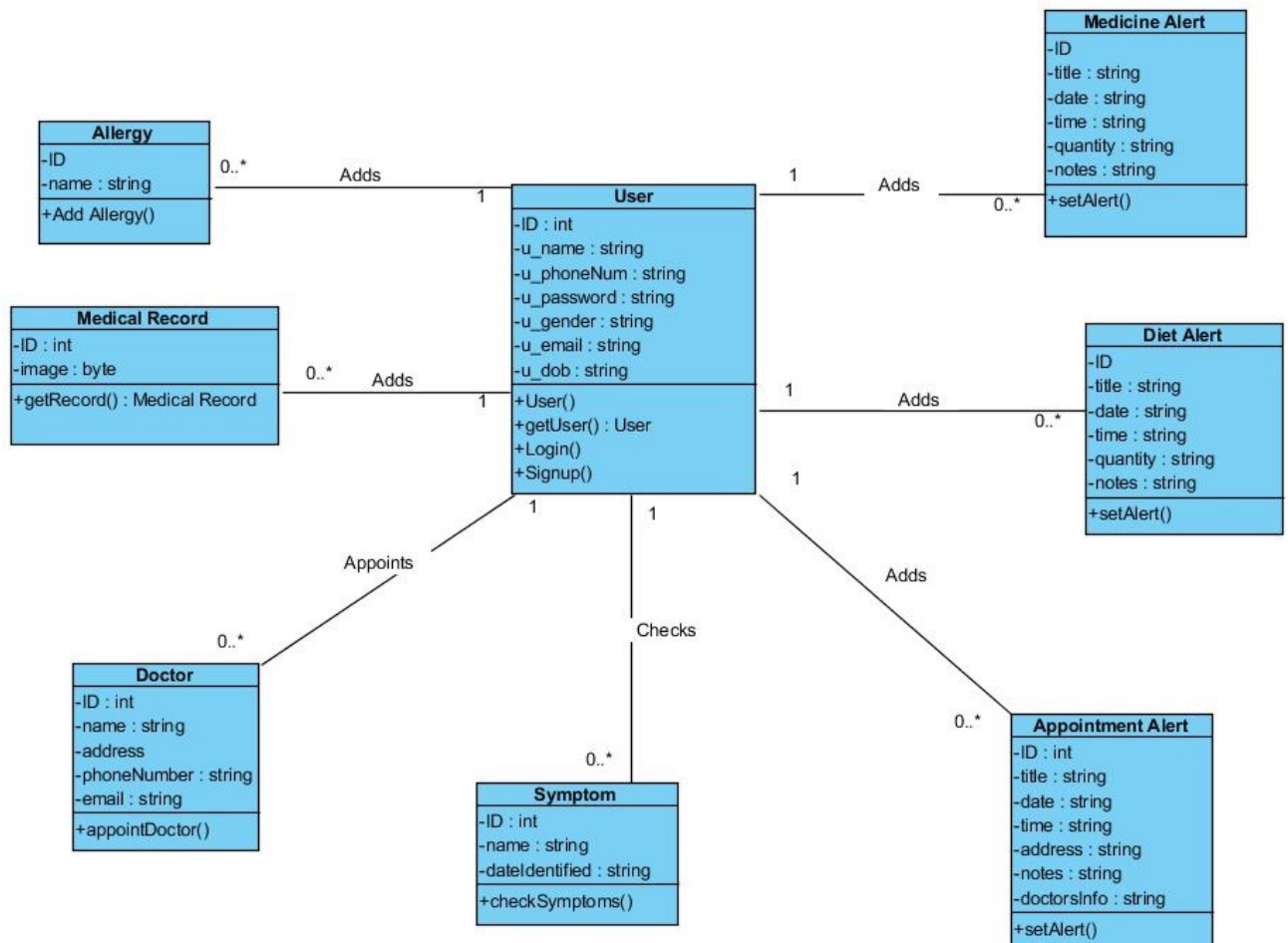


Figure 8: Class Diagram

Class diagram in figure 3.5 gives the description of the classes interacting within the application. Classes are also showing the functions used in them including the variables used in the functions.

3.6. Entity Relationship Diagram (ERD)

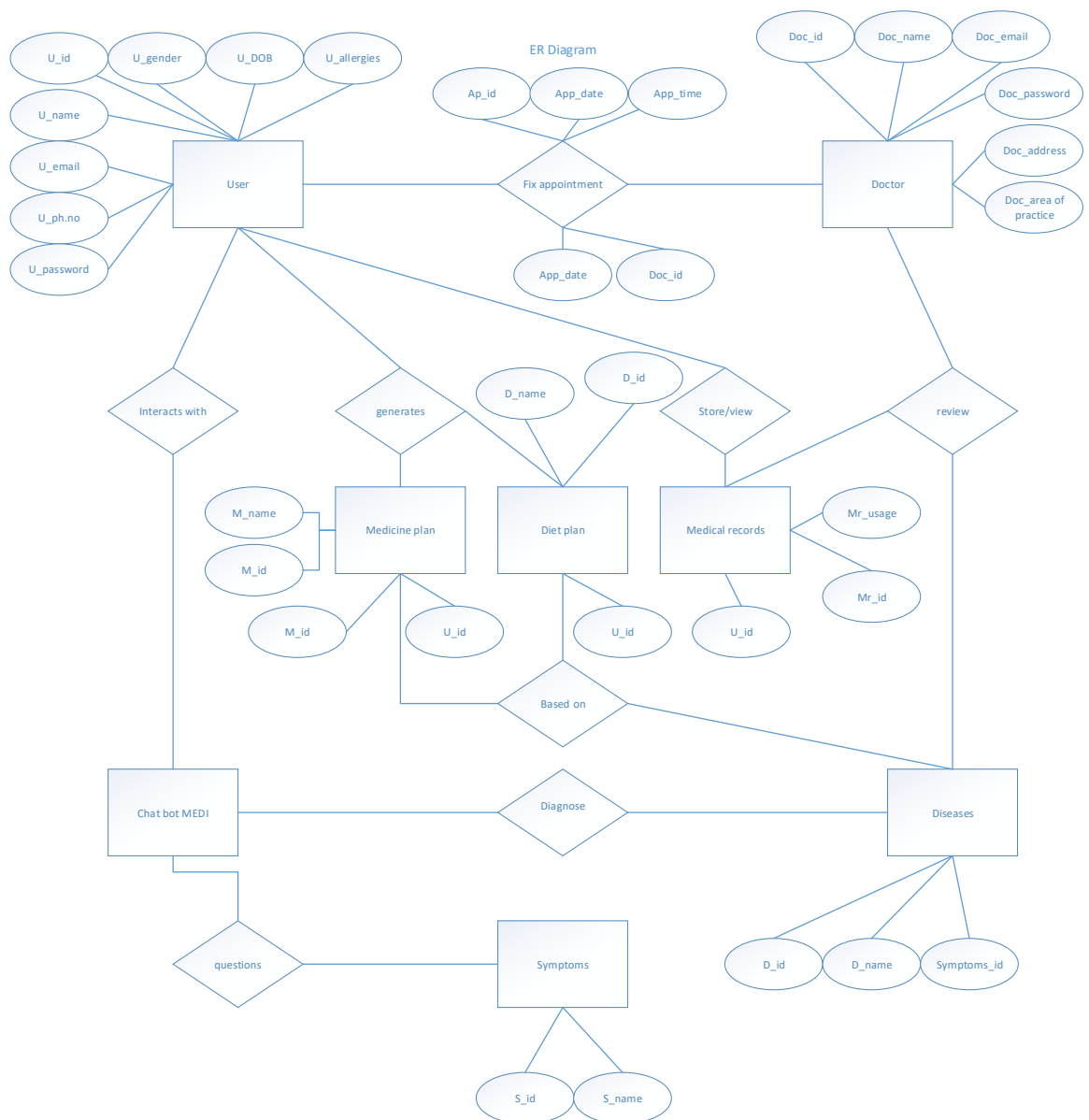


Figure 9: Entity Relationship Diagram

The ERD Entity Relationship Diagram in figure 7: gives the description of the entities and their relationships with each other following are the entities in this projects:

- User
- Doctor
- Medicine Plan
- Diet Plan
- Medical Records

- Chat Bot Medi
- Diseases
- Symptoms

Along with the entities their attributes and relationships are also listed which makes a complete ERD.

3.7 Methodology

The methodologies in the project includes:

- Requirement gathering
- Interface designing
- Collection and classification of data
- Making of dataset
- Making of BOT
- Integration of BOT and database with App
- Making of other modules of the App
- Testing
- Implementation of rest of modules
- Execution and Maintenance

Moreover we will be using some machine learning techniques and algorithms for classification of data which are as follows:

- Decision Tree

Deep learning is a subfield of machine learning that is a set of algorithms that is inspired by the structure and function of the brain.

Decision tree algorithm falls under the category of the supervised learning. They can be used to solve both regression and classification problems. Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any Boolean function on discrete.

CHAPTER 4

DATA AND EXPERIMENTS

4.1 Input Event

Input Design is the process of the converting user oriented inputs to a computer based format. Inaccurate Inputted data have the most common sense of error in data processing. Any ambiguity conceiving at input leads to a total fault in output. The goal of designing the input data is to make the data entry easy, logical and error free as possible, Inputs are made using forms and data validation is done at the time of inputting data.

4.2 Tools & Technologies

4.2.1 Develop for Android

Android, this word means a lot in present High-Tech World. Today Smartphone are known for its operating system which is Android. Earlier there is no option for operating systems like Android in mobile, as usual there are Symbian, java featured operating systems but today things had changed a lot, everyone wants a Smartphone which is functioned on Android only. In a very short span of time android created a reputed place in the market. The main motive of this move was to facilitate openness in mobile software and thus give customers a diversified software base which is not only less expensive but also convenient and easy to use. The Android platform has emerged from such a move. Android is an Operating System that is designed to be used with a variety of handsets that will be launched in the second half of 2008

4.3 Android Activity Lifecycle

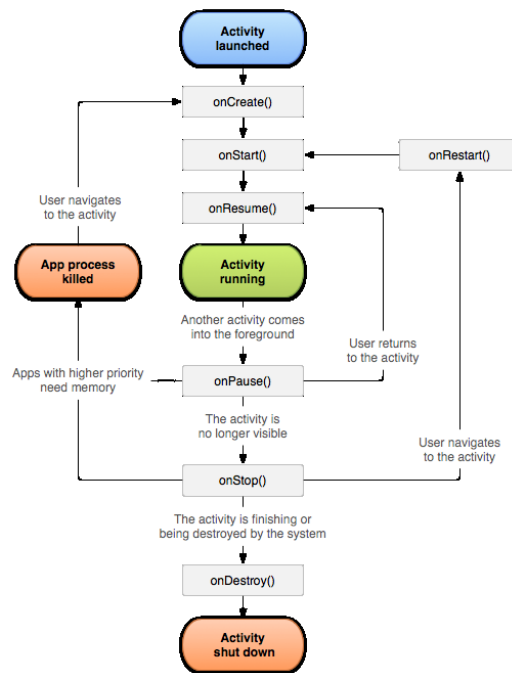


Figure 10: Android Activity life cycle

4.4 Android Architecture

The following diagram shows the major components of the Android operating system. Each Section is described in more detail below.

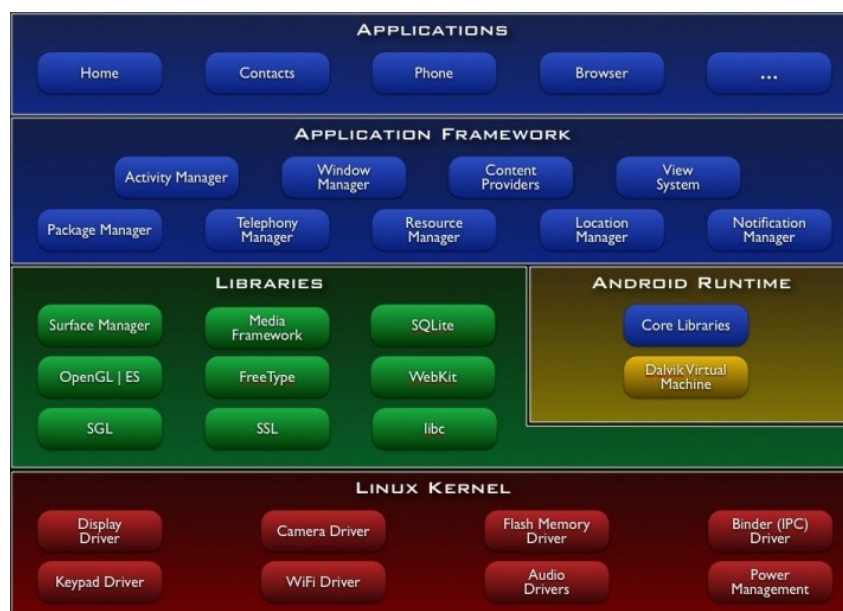


Figure 11: Android Architecture

4.4 External APIs/ Libraries used

- AIDL
- Volley
- RecyclerView
- Maven

4.4.1 Permissions

- Internet permissions
- Access network state permission
- Write external storage
- Read external storage
- Camera permission

4.4.2 Database & Hosting

Hosting space: <https://www.000webhost.com/>

Database Used: MYSQL & SQLITE

Coding Language: PHP.

4.5 System Testing

4.5.1 Software Testing

Software testing technique is used to test the system that either the software is performing the functionality according to the requirements or not. It can be stated as

the process of checking whether the software is performing the stated task (the task given by the user) or not. In software testing, we tested our application which is MediCure. It can be tested by running the program after each module is developed.

4.5.2 Unit Testing

Unit testing is used to test each unit of the system so that no error can be found in the system. We performed unit testing on every unit of the game such as assessment of disease is working properly or not, prescribing treatment of disease, sending notification alerts, appointment with doctor is working correctly or not etc.

4.5.3 Integration Testing

Integration testing is used to check how things have integrated in the system. In our project, we checked how our chat bot will interact with the user and user's response to the chat bot moreover how the data is accurately retrieved from the database.

4.5.4 Acceptance Testing

In acceptance testing, we will check whether the system is meeting the requirements or not. We test each requirement individually so that our system should be developed according to our requirements. In our project, it is our requirement that user should login through email id and password so we tested that either player is logging in through email id and providing the correct password or not.

4.6 Test Cases

Test cases are designed to test inputs and executable functionality of system.

4.6.1 Test Case 1: User Login

In this test case, user log in through email id and password to the application. It is Mandatory for every user to have an email id to login within the app and perform the desired operations.

Test Case ID	01
Description	To check user logged in using email id and password
Initial State	Mobile phone should be connected to internet
Function to be tested	User login through email id successfully or not
Test Execution	<ul style="list-style-type: none"> • Select login through email • Enter correct email id and password
Expected Results	User should be successfully logged in
Actual Results	User has successfully logged in
Status	Pass

Table 1: Test Case User Login

4.6.2 Test Case 2: Profile Update

In this test case user must have to enter correct date of birth, gender, body weight, and height to calculate the BMI and BMI category

Test Case ID	02
Description	To calculate the BMI and BMI category
Initial State	Enter the correct requirements needed to calculate
Function to be tested	BMI and BMI category
Test Execution	<ul style="list-style-type: none"> • Enter Date of birth • Gender • Body weight • Body Height
Expected Results	BMI and BMI category should be calculated accurately
Actual Results	BMI and BMI category calculated accurately
Status	Pass

Table 2: Test Case Profile Update

4.6.3 Test Case 3: Patient Assessment

In this test case patient/user interaction with the chat bot for the assessment of the disease.

Test Case ID	03
Description	User interaction with chat bot for assessment of disease
Initial State	Start conversation with chat bot
Function to be tested	Chat bot responding correctly on the given symptoms by the user
Test Execution	Asking questions about the related symptoms
Expected Results	Chat bot should be able to get the desired correct results about the disease
Actual Results	Chat bot diagnosed the correct disease based on symptoms
Status	In-process

Table 3: Test Case Patient Assessment

4.6.4 Test Case 4: Notification Alerts

In this test case, the notification alerts for medicines, diet alerts and appointment with the doctor.

Test Case ID	04
Description	Notification alerts for medicine, diet and appointment with doctor
Initial State	Set the time for getting alerts at required time set
Function to be tested	Notification alerts given to patient for medicine, diet and appointment to doctor
Test Execution	Giving alerts at correct time given for getting alerts
Expected Results	Notification alerts should be provided accurately
Actual Results	Notification alerts provided accurately
Status	Pass

Table 4: Test Case Notification Alerts

4.7 Test Case Results

4.7.1 User login result

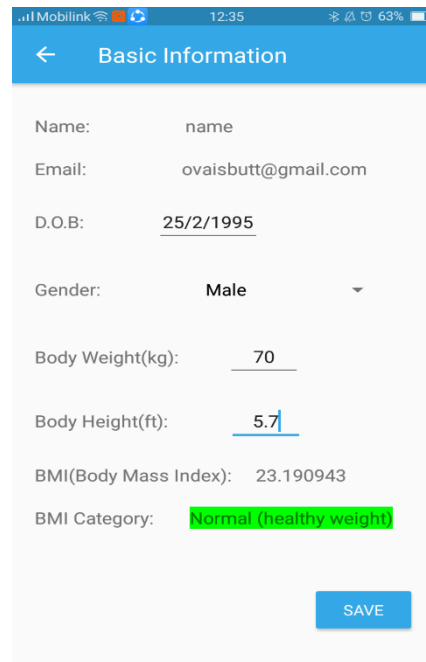
In this user login test case, it is checked whether the user is logged in through email address and correct password.

ID	user_nicename	user_email	user_pass	user_mbl	dob	gender	bodyweight	bodyheight	bmi	bmic
1	Ovais Butt	ovaisbutt786@gmail.com	123456	3218970058	15/12/1994	Female	110	5.8	35.1971	Save obes
2	Taimur	taimur@gmail.com	taimur	3154172828			0	0	0	
7	Ovi	ovi@gmail.com	1234567	3218970058	0000-00-00		0	0	0	
11	Ovais Butt	ovaisbutt@gmail.com	123456	3218970058	25/4/2018	Male	70	5.8	22.3981	Norm (heat weight)
16	Ovais	ovbutt@gmail.com	hello123	3218970058			0	0	0	

Figure 12: User Login Test Result

4.7.2 Profile Update Result

In this test case profile of the patient updated and calculated the correct BMI and BMI category



The screenshot shows a mobile application interface for updating a patient's profile. The title is "Basic Information". The form contains the following fields and values:

Name:	name
Email:	ovaisbutt@gmail.com
D.O.B:	25/2/1995
Gender:	Male
Body Weight(kg):	70
Body Height(ft):	5.7
BMI(Body Mass Index):	23.190943
BMI Category:	Normal (healthy weight)

A blue "SAVE" button is located at the bottom right of the form.

Figure 13: Profile Update Test Result

4.7.3 Patient Assessment Result

In this test case patient assessment of the disease is checked by interaction with chat bot providing the symptoms.

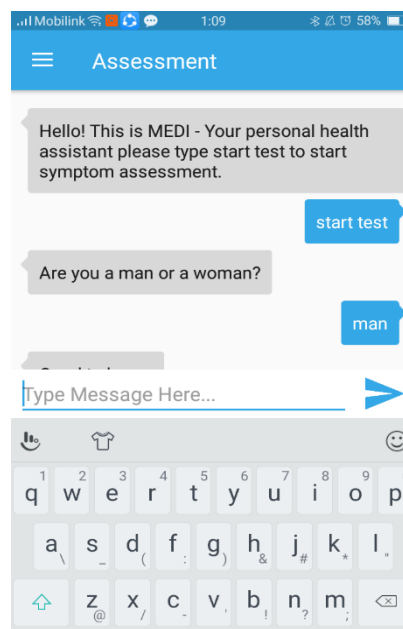


Figure 14: Patient Assessment Test Result

4.7.4 Notification Alerts Result

In this test case, the notification alerts for patient's medicine, diet alerts and appointment with doctor generated.

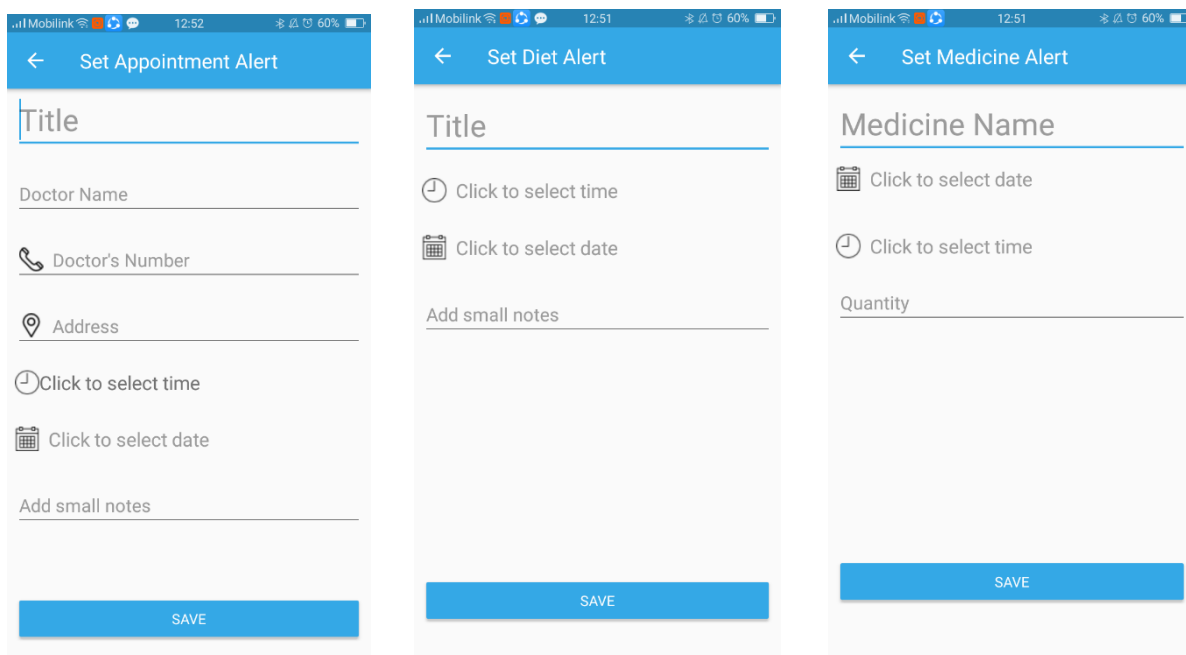


Figure 15: Notification Alerts Test Result

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 General Information

MEDICURE is an android application that provides a disease diagnosing medical expert system that can be downloaded from Google play store.

5.2 Overview

MEDICURE is an android based medical expert system which diagnoses diseases based on the symptoms provided by the patient and prescribes the treatment for that particular disease.

5.2.1 System Summary

The system designed for diagnosing diseases is android based operating system which registers users and provide user names to them. A chat bot interacts with the system and ask questions about the condition of the patient regarding the disease and gives directions accordingly, specially prescribes the treatment and relative physician to consult.

5.2.2 System Configuration

This application can be configured by downloading it from Google play store and install it in the android phone having minimum android version KitKat.

5.2.3 User Access level

User can access the app after installing it in the phone by signing up if the app is being used for the first time it will have to register to the app and then login to it otherwise if it is already registered to the app it can be directly logged in to the application.

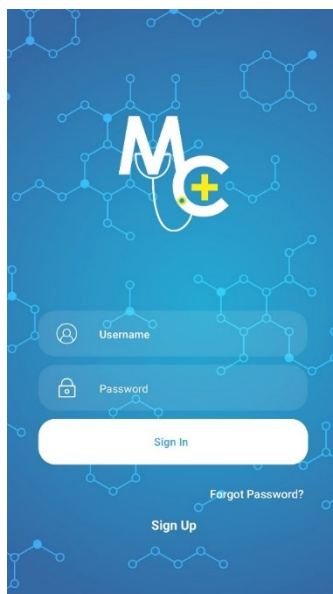


Figure 16: User Access level

5.2.4 Understanding the main menu

Main menu of the application will have the two options to choose first is signup and second on will be the login option. In sign up option user will have to provide the credentials required for registering to the application for the further processing.

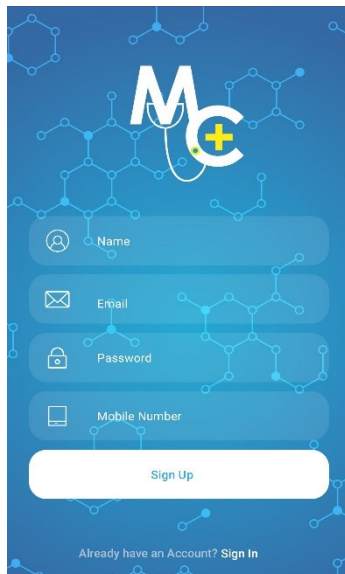


Figure 17: Main Menu Activity

5.2.5 Starting Application

After being logged in to the application a navigation drawer activity will open which will have following options to choose:

- Start Assessment
- Profile
- Prescribe Treatment
- Set notification alerts
- Medical records
- Search doctor
- About us
- Log Out

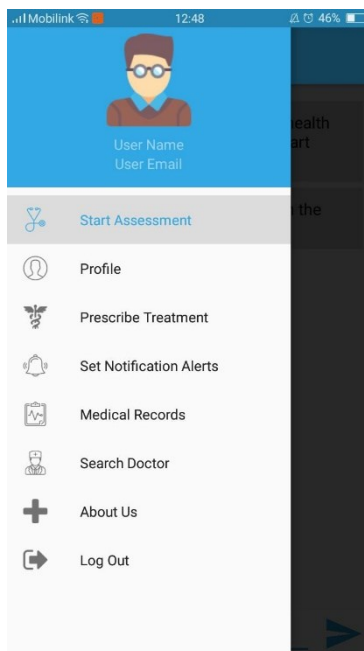


Figure 18: Starting Application

5.2.6 Start Assessment

In this activity user will interact with an AI based chat bot which will ask questions from the user for diagnosing the disease of the patient. Moreover user will also ask questions for the chat bot and also provide the allergies if any of them are present.

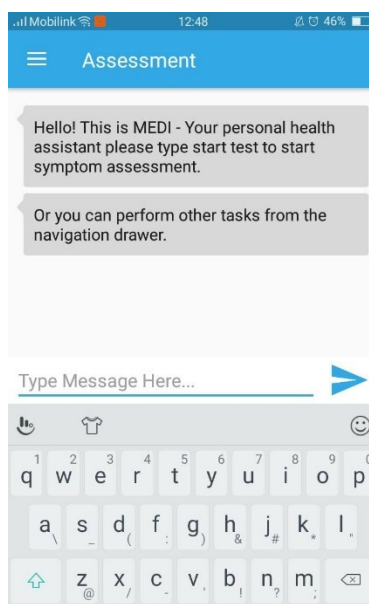


Figure 19: Start Assessment Activity

5.2.7 Profile

Here will be the all personal information of the patient who have registered to the application. It will contain the user name, email, date of birth, gender, body weight, body height, BMI (Body Mass Index), BMI category.

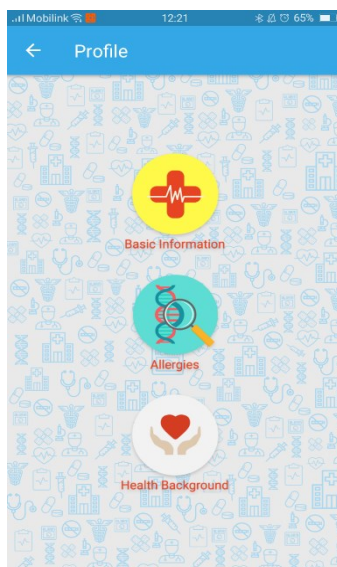


Figure 20: Profile Activity

5.2.8 Basic Information

In this activity the user can update its profile in the data base and can even calculate its BMI here more over the information will be shown at the runtime in this activity.

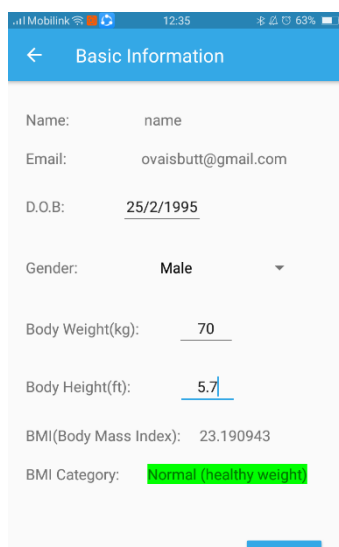


Figure 21: Basic Information Activity

5.2.9 Allergies

In this activity the user can list down his or her allergies just to keep the record this can be done by the plus button showing in left figure and the dialogue box will appear just like in the centre diagram and finally the allergy can be listed in the list view just like in the right figure.

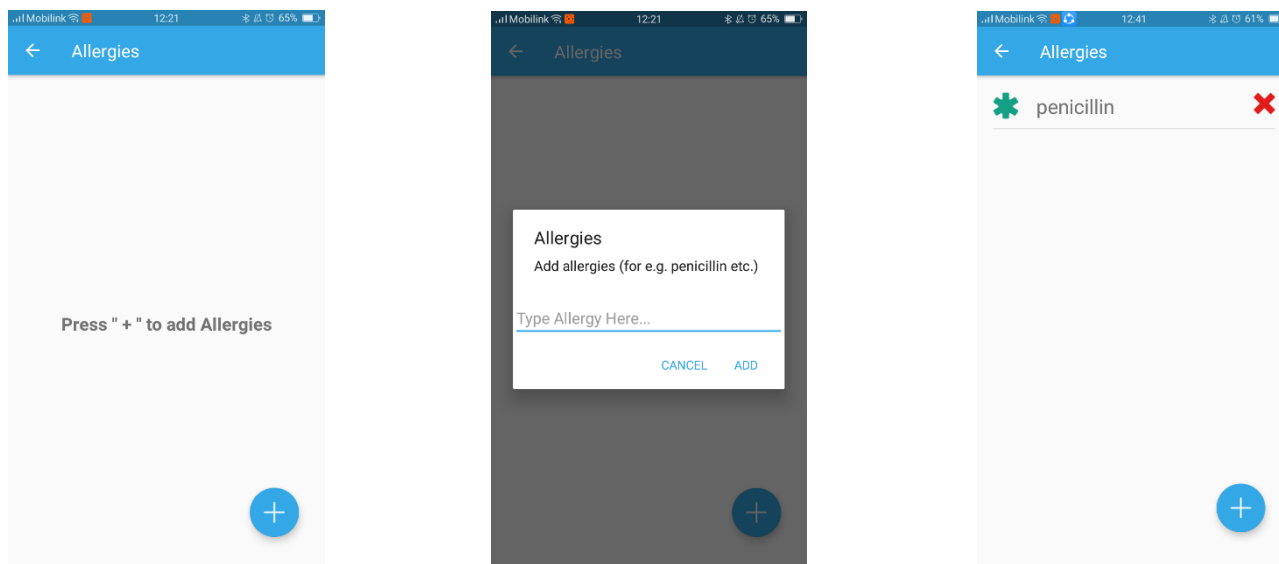


Figure 22: Allergies Activity

5.2.10 Health Background

In this activity the user can select among the 3 question which affect ones user's profile this consists of simple radio buttons and are saved in the shared preferences.

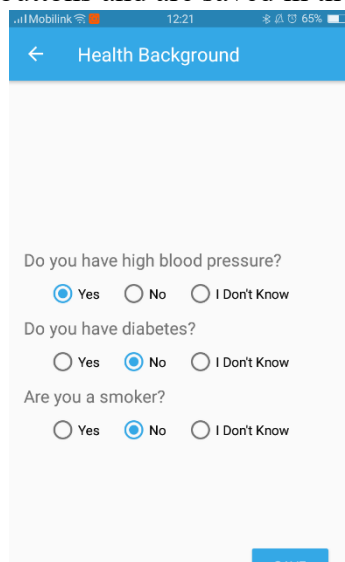


Figure 23: Health Background Activity

5.2.11 Prescribe Treatment

In this activity after analysing the symptoms provided by the patient respective treatment. It will also give the dos and don'ts and medicine that the patient can take.

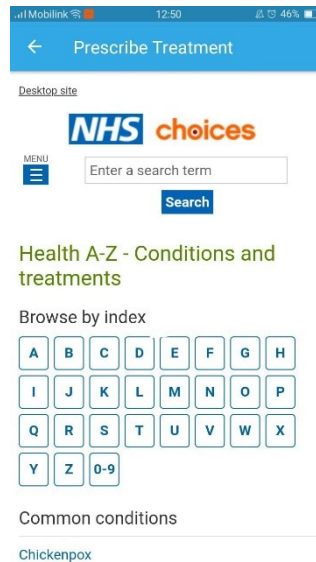


Figure 24: Prescribe Treatment Activity

5.2.12 Set notification alerts

This activity will give the alerts to the patient regarding the medicine and diet plan of the patient and moreover the appointment alerts of the patient with doctor.

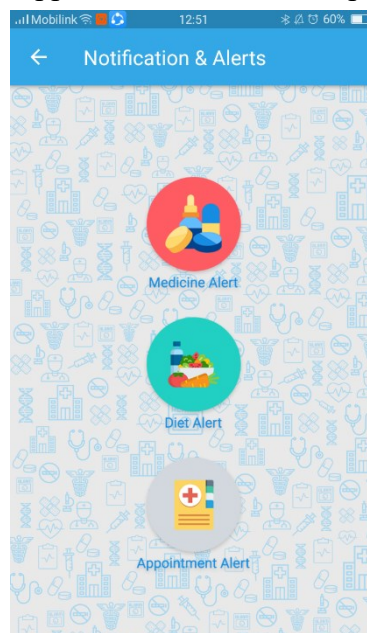


Figure 25: Notification Alerts Activity

5.2.13 Medicine Alerts

Here the user can add medicine alert to keep himself notified on time about the medicine he has to take, when to take and how must dosage he has to take.

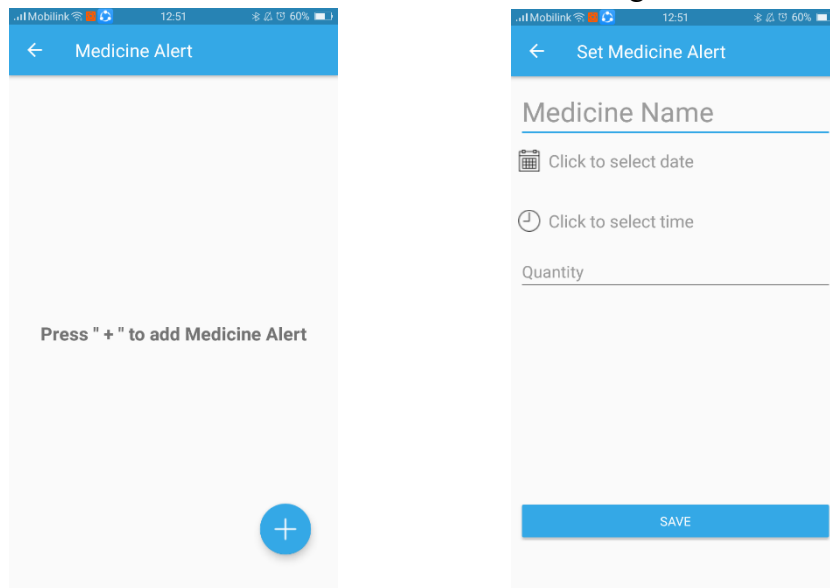


Figure 26: Medicine Alert Activity

5.2.14 Diet Alert

This Activity allows the user to add diet alert among with small notes where user can specify which diet to take and the ingredients of the diet etc.

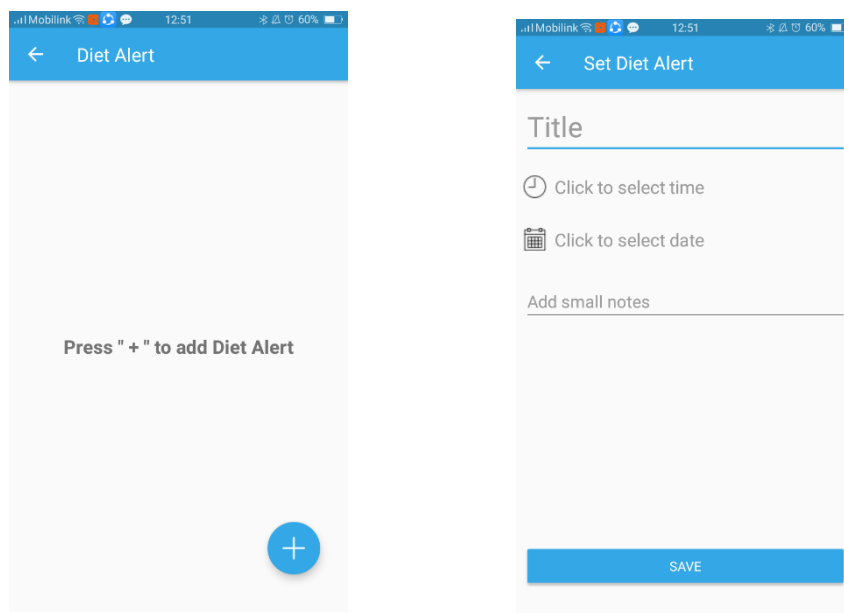


Figure 27: Diet Alert Activity

5.2.15 Appointment Alert Activity

This activity allows the user to save appointment alert with any doctor this allows you to remember the doctor's name, phone number address, time, date and some notes about the appointment.

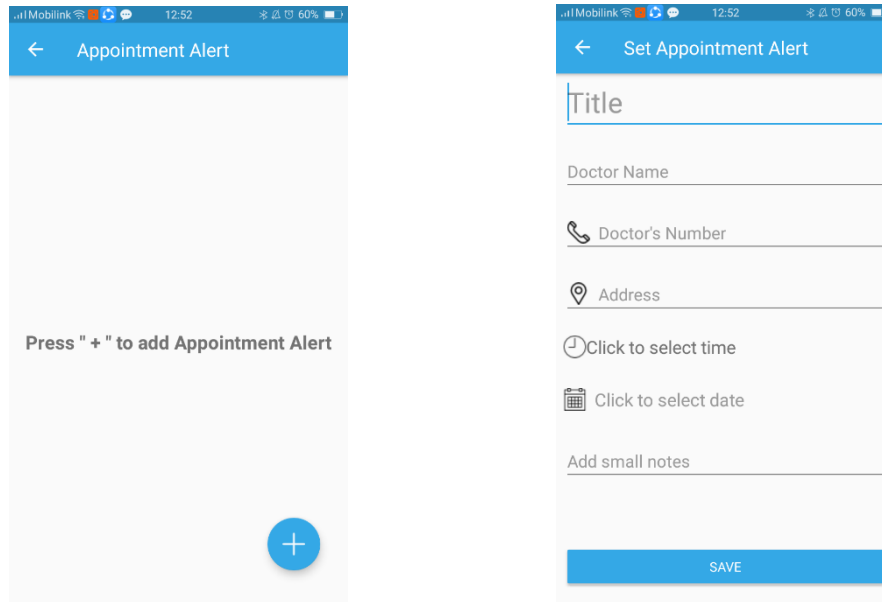


Figure 28: Appointment Alert Activity

5.2.16 Medical records

This activity will keep record of the past and present medical records of the patient here the user is asked to upload an image of 2MB max size which are then saved in the data base so that the user can review them afterwards.

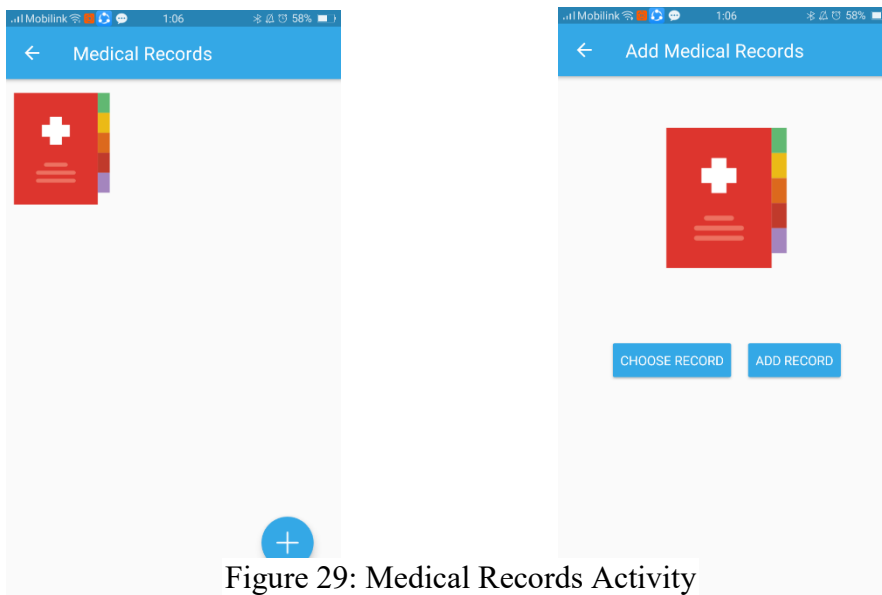


Figure 29: Medical Records Activity

5.2.17 Search Doctor

It will contain list of relevant doctors and physicians which can be consulted for further treatment

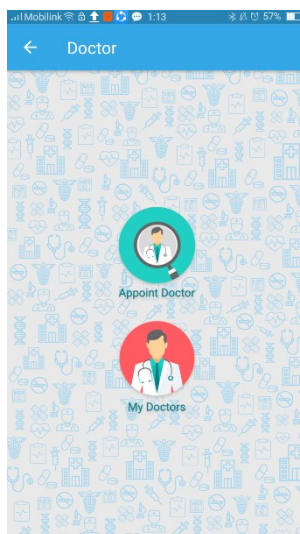


Figure 30: Search Doctor Activity

5.2.18 Appoint Doctor

This Activity allows you to access the doctors in your area and allow you to fix appointment with them by just a one click call button which allows you to call the doctor and fix appointment with them.

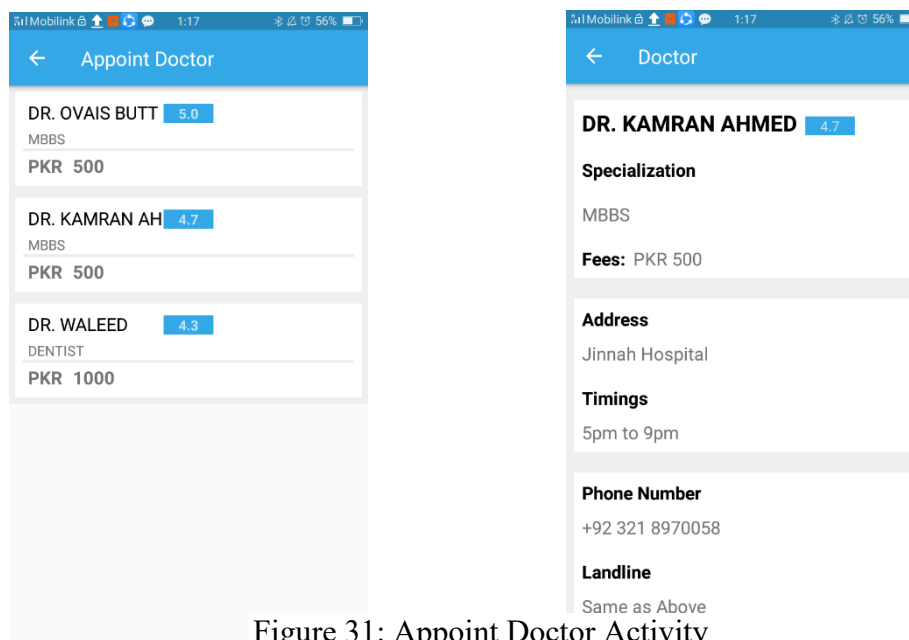


Figure 31: Appoint Doctor Activity

5.2.19 My Doctor

This Activity allows the user to add their own personal doctors among their complete profile like their name, phone number, landline number, email, address etc.

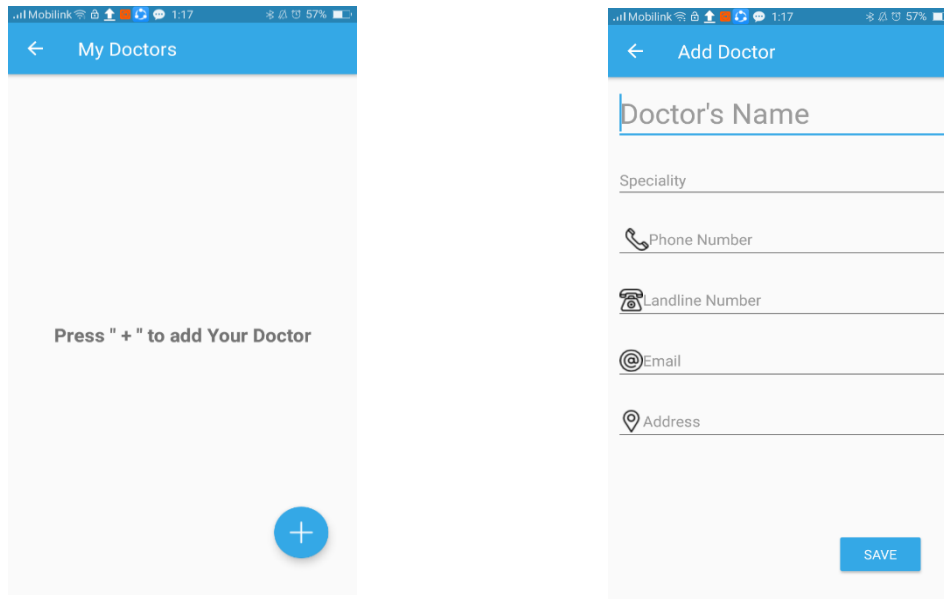


Figure 32: Add My Doctor Activity

5.2.20 About Us Activity

This Activity contains all the information about MediCure

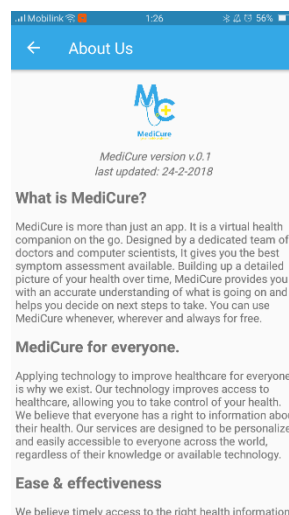


Figure 33: About Us Activity

5.2.21 Logout

This will logout the user from the application and return to the login page.

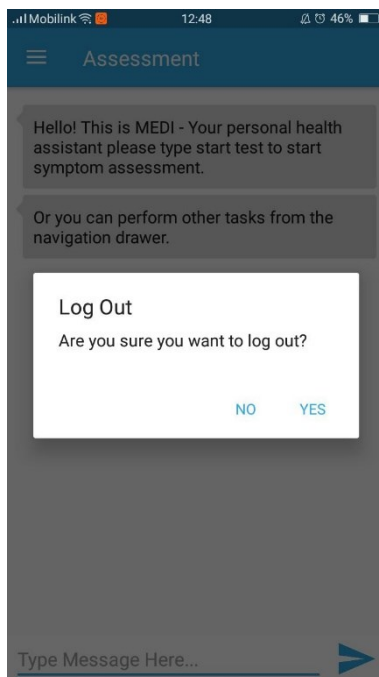


Figure 34: Logout Activity

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

We got inspired for the idea of this project by looking at the current situation of healthcare in our country where people don't bother to see a doctor due to their busy schedule and patients have to wait in long queues to see the doctors and even the patients cannot find appropriate doctor for disease, the patients also feel it difficult to arrange their info at one place.

The Medical Application we made is intended to facilitate the patients remotely. With this Medical App solution user can now easily get to know about the disease or can manage their records at one place and can even fix appointment with the appropriate doctor. This application will make the medical awareness among the people so fluent than before and will provide a freelance platform for the doctors. The GUI provided is so simple that any novice can also learn to use it.

This Medical App Solution will also make the facilitate doctors because they need not to open a clinic and make a big investment they can just call the patients at their offices or work places. All the Doctor has to do for his profile setup is contact us and show his degree or any valid proof of being a doctor and we will setup a profile for him where the patients will be able to see his profile and can contact him there.

6.2 Recommendations

Regarding this project I would like to recommend that a lot more work can be done on this project which may include live conversation between the patient and the doctor which might be a video call, the patients will appoint the doctor over this app and all the transactions will be done through this app, we can also maintain the database of medicines and diet plans which will be generated for the user on request and medicines will be prescribed keeping in view the health background of the patient and the disease he is dealing with.

A new paragraph should not begin on the last line of a page. A subsection title should not begin on the last line of a page. A new chapter must start on a new page.

6.3 Learning Outcomes

From this project I came to know how to make an android app under the supervision of our supervisor for the final year project and following the deadlines side by side and work with team structure. I came to know what to do and what not to do to make project unbeaten. It also gives me benefits to understand how real world Project carried out in IT Firm.

I worked with some of the Team Leaders and experienced persons who were having lots of experience in the same field. On the whole, the project has made me learn so many new as well as important things. The purpose of final semester training is fulfilled with this project. And above all, the things which I have learnt will be useful for all the upcoming projects and will give me an edge in the market.

REFERENCES

Journal Papers:

[1] Nohria, Rimpay. "Medical expert system-A comprehensive review." *International Journal of Computer Applications* 130.7 (2015): 44-50

Electronic Sources from Internet:

[2] Nhs.uk. (2018). Health A-Z - Conditions and treatments. [Online] Available at: <https://www.nhs.uk/Conditions/Pages/hub.aspx> [Accessed 10 March 2018].

[3] Medscape. 2018. Conditions Library. [ONLINE] Available at: <https://reference.medscape.com/>. [Accessed 10 March 2018].

[4] Google. 2018. Activity Life Cycle. [ONLINE] Available at: <https://developer.android.com/guide/components/activities/activity-lifecycle.html>. [Accessed 10 March 2018].