



BSCS-F18-003

03-134142-061 MOHAMMAD HAMZA

03-134142-049 MOAZAM HUSSAIN

### **GLUCO-CARE**

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

Supervisor: Mr. Tahir Iqbal

Department of Computer Sciences  
Bahria University, Lahore Campus

January 2019



# Certificate



We accept the work contained in the report titled

**“GLUCO-CARE”**

written by

**Mohammad Hamza**

**Moazam Hussain**

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. Tahir Iqbal**

\_\_\_\_\_  
(Signature)

January 11<sup>th</sup>, 2019

**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
03-134142-061	MOHAMMAD HAMZA	
03-134142-049	MOAZAM HUSSAIN	

Date : 11-01-2019

Specially dedicated to

My beloved grandmother, mother and father

(Mohammad Hamza)

My beloved grandmother, mother and father

(Moazam Hussain)

## ACKNOWLEDGEMENTS

We would like to acknowledge our deepest gratitude to our supervisor, Mr. Tahir Iqbal for his guidance, constant attention, valuable suggestion, enthusiastic support and personal concern during the project. Furthermore, our appreciation also goes to Mr. Aasim Ali from the Department of Computer Science & IT for his time, attention and guidance. Finally, we would like to thank all the teachers that assisted us with this project. In addition, we would also like to express my gratitude to our loving parents and friends who had helped and given us encouragement.

Mohammad Hamza

Moazam Hussain

## **GLUCO-CARE**

### **ABSTRACT**

Gluco-care is the first Arduino based Android application that helps the diabetic patients to check their glucose readings and suggesting a workout and diet plan according to it.

Staying physically fit and healthy significantly helps in managing diabetes. It's easy to start an exercise routine once you've decided it's the time for a change but keeping it up is a challenge for many people. And Gluco-Care gives you the perfect reason to keep up with your fitness regime.

Gluco-Care will give the facility to the user to find nearby doctors and user can request the available doctor for appointment to fight against this chronic illness.

Glucose reading will be collected through a strip jack attached to Arduino Nano. Arduino Nano will be attached to smart phone using OTG (on-the-go).

## TABLE OF CONTENTS

<b>DECLARATION</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF SYMBOLS / ABBREVIATIONS</b>	<b>xii</b>

### CHAPTERS

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Aims and Objectives	2
	1.4 Scope of Project	3
	<b>SOFTWARE REQUIREMENT SPECIFICATION</b>	<b>4</b>
	2.1 User Classes and Characteristics	4
	2.2 Operating Environment	5
	2.3 Design and Implementation Constraints	5
	Environmental Constraints	5
	2.4 Hardware Interfaces	6
	2.5 Software Interfaces	6
	2.6 Communication Interfaces	6
	2.7 Functional Requirements	6



2.8	Safety Requirements	9
2.9	Security Requirements	10
2.10	Software Quality Attributes	10
2.11	Other Requirements	10
2.12	System Requirements Chart	11
<b>3</b>	<b>DESIGN AND METHODOLOGY</b>	<b>12</b>
3.1	Design	12
3.2	Use Cases	12
3.2.1	Use case name and identifier	12
3.2.2	Use case Diagram (refined and updated)	25
3.3	Sequence Diagram Description	26
3.3.1	Sequence Diagram	27
3.4	Collaboration Diagram	28
3.4.1	Login	28
3.4.2	Sign Up	29
3.4.3	Test	30
3.4.4	View Test	31
3.4.5	Appointment	32
3.4.6	View Appointment	33
<b>4</b>	<b>IMPLEMENTATION</b>	<b>34</b>
4.1	Implementation	34
4.2	Android Application	34
4.2.1	Libraries Used for UI	34
4.3	Server-Side Development	35
4.3.1	Libraries used in NODE JS:	35
4.4	Backend Development	35
4.5	Hardware	36
4.6	Record	37
4.7	Main Server	37
<b>5</b>	<b>USER MANUAL</b>	<b>38</b>

5.1	User Android Application	38
5.1.1	Getting Started	38
5.1.2	Application Welcome Page	39
5.1.3	The User Type	40
5.1.4	Sign In	41
5.1.5	Sign Up	42
5.2	Application Overview	43
5.2.1	Patient Profile Screen	43
5.2.2	Home Screen	44
5.2.3	App Navigation Drawer	45
5.2.4	Health Chart	46
5.2.5	Diet Chart	47
5.2.6	Workout Chart	48
5.2.7	Patient Settings	49
5.3	Doctors End	50
5.3.1	Main	50
5.3.2	Doctor's Settings	51
5.4	Web Application (Admin Panel)	52
5.4.1	Login Page	52
5.4.2	Admin panel Home Screen	53
5.4.3	Admin panel Home Screen	54
5.4.4	Entity Relationship Diagram (ERD)	55
<b>6</b>	<b>CONCLUSION AND RECOMMENDATIONS</b>	<b>56</b>
6.1	Conclusion	56
6.2	Recommendation	57
6.2.1	PCB	57
6.2.2	Wireless Connection	57
6.2.3	Painless Testing	57

**LIST OF TABLES**

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
Table 1	System Requirement Chart	11
Table 2	U1	12
Table 3	U2	14
Table 4	U3	18
Table 5	U4	19
Table 6	U5	20
Table 7	U6	21
Table 8	U7	22
Table 9	U8	23

**LIST OF FIGURES**

<b>FIGURE</b>	<b>TITLE</b>	<b>PAGE</b>
Figure 1:	Use Case Diagram	25
Figure 2:	Sequence diagram	27
Figure 3:	Login Collaboration Diagram	28
Figure 4:	Sign Up Collaboration Diagram	29
Figure 5:	Test Collaboration Diagram	30
Figure 6:	View Test Collaboration Diagram	31
Figure 7:	Appointment Collaboration Diagram	32
Figure 8:	View Appointment Collaboration Diagram	33
Figure 9:	Application Welcome Page	39
Figure 10:	The User Type	40
Figure 11:	Sign In Page	41
Figure 12:	Create account page	42
Figure 13:	Patient Profile Screen	43
Figure 14:	Main page	44
Figure 15:	Application Navigation Drawer	45
Figure 16:	Heath Chart	46
Figure 17:	Diet Plan	47
Figure 18:	Workout Chart	48
Figure 19:	Patient Setting	49

Figure 20 Doctor Main Page	50
Figure 21 Doctor's Setting	51
Figure 22: Admin Login	52
Figure 23: Home Screen	53
Figure 24: Pending Doctor Verification	53
Figure 25: Home Screen	54
Figure 26: Pending Doctor Verification	54
Figure 27: ERD of Project	55

**LIST OF SYMBOLS / ABBREVIATIONS**

<i>OTG</i>	On-The-Go
<i>HwSw</i>	Hardware Software
<i>ER</i>	Entity Relation
<i>PCB</i>	Printed Circuit Board
<i>mg/dl</i>	Milligram per Deciliter

## CHAPTER 1

### INTRODUCTION

#### 1.1 Background

Diabetes illness is considered a significant public issue all over the world. Diabetes is a disease that occurs when your blood glucose, also called blood sugar is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy. Sometimes your body doesn't make enough—or any—insulin or doesn't use insulin well. Glucose then stays in your blood and doesn't reach your cells. Over time, having too much glucose in your blood can cause health problems. Although diabetes has no cure, you can take steps to manage your diabetes and stay healthy.

Sometimes people call diabetes “a touch of sugar” or “borderline diabetes.” These terms suggest that someone doesn't really have diabetes or has a less serious case, but every case of diabetes is serious.

With help of Gluco-care, let us pledge to dispel ignorance around this chronic illness and help people with diabetes and those at the risk of diabetes.

For now, it's time to act.

## 1.2 Problem Statements

Diabetes is a major cause of deaths happening around the world daily. Due to the increased population, there is a hike of demand of Accu-Meters or Accu-Check for day to day Sugar level testing. Our first of kind Arduino based accu-meter called Gluco-Meter is a first step towards this business.

Pakistan is the 7th country in terms of diabetic patients and the most noticeable reason for a person's death during diabetes is less monitoring the previous test records and the absence of awareness about proper diet plans and workout routines, which is highly recommended by doctors to any diabetic patient. Gluco-care helps the diabetic patients to check their glucose readings and generating a chart showing maximum and minimum glucose readings and suggesting a workout and diet plan based on minimum of 6 readings. Workouts will be specially arranged for a diabetic patient so that the patient can easily start the workout without spending any amount on a personal trainer. Moreover Gluco-Care will give the facility to the diabetic patients to find nearby doctors and to request the available doctor for appointment to fight against this long-lasting disease.

Because staying physically fit and healthy significantly helps in managing diabetes. It's easy to start an exercise routine once you've decided it's the time for a change but keeping it up is a challenge for many people. And Gluco-Care gives you the perfect reason to keep up with your fitness regime.

## 1.3 Aims and Objectives

Designing an Arduino based Gluco-meter and an android application which will:

- i) Analyse the sugar level of patients through strips.
- ii) Develop an android application to track glucose readings and generating a graph. Suggesting proper diet plans and workout based on these readings.



iii) Providing a platform to book an appointment with concerned doctor.

This paragraph define the scope of project.

#### **1.4 Scope of Project**

Create an Arduino based smartphone device to check blood glucose level through an application that when connected in the OTG (on -the-go) will ask the user to place the blood sample on the sugar strip. Collect the readings from the patient's a blood and save them as a record that can be viewed by both the patient and the doctor for better treatment and will show the control of the blood Glucose level in percentage to tell the user how much sugar they can use to balance their diet and suggest workout and diet plans with respect to their condition. Create a platform that will allow diabetic patients to book appointments with the best doctors available.

## CHAPTER 2

### SOFTWARE REQUIREMENT SPECIFICATION

#### 2.1 User Classes and Characteristics

Following are the user classes and characteristics of Gluco-Care.

- **Check Blood Sugar (Patient End)**

This is for patient use.

- New
- Previous
- Patient's Chart
- Manual Input

- **Doctor**

This is for doctor use.

- Make an Appointment
- Route to Address of Doctor
- Doctor's fee

- **Appointment (Doctor's End)**

This is for doctor use.

- List of appointed patients
- Cancel appointment
- Accept appointment

## 2.2 Operating Environment

Following is the operating environment for this project.

- **Operating System:** Android(kit-kat)
- **Database Management System:** PostgreSQL
- **REST API:** Node.js
- **Hardware:** Arduino NANO

## 2.3 Design and Implementation Constraints

Following are the implementation constraints of this project.

### Application Constraints

Following are the application constraints.

- Internet should be connected (for appointments).
- Device should be connected to the smartphone through OTG before reading.
- GPS availability for doctor appointment.

### Hardware Constraints

Following are the hardware constraints.

- The device should be connected through OTG with smart phone.

### Environmental Constraints

Following are the environment constraints.

- Strips should be clean.
- Lancet should be sterilized.

### Cultural cons

Following are the cultural constraints if any.

- There are no cultural constraints.

## 2.4 Hardware Interfaces

The hardware we are going to use for this project are Arduino Nano, Operational Amplifier, Sugar jack, Sugar Testing Strips, OTG(On-The-Go) and micro controller to connect the above setup with our application.

## 2.5 Software Interfaces

Gluco-care requires minimum Android level of 17 (i.e. Jelly Bean). Gluco-Care uses PostgreSQL as database to store the data of users, PostgreSQL is also used to implement the back-end logic.

## 2.6 Communication Interfaces

Gluco-Care requires an internet connection to connect with the Rest API. The application requires HTTP to communicate with the server. The application communicates with Arduino Gluco-Meter via OTG to get the output from the sugar jack.

## 2.7 Functional Requirements

Following are the fundamental requirements of this project.

### **ID: FR1**

**TITLE:** Download mobile application

**DESC:** A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download.

**ID: FR2**

**TITLE:** User registration - Mobile application

**DESC:** Given that a user has downloaded the mobile application, then the Admin should be able to register through the mobile application. The administrator must provide user-name, phone-number, password and email address. The user can choose to provide a regularly used phone number.

**RAT:** For a user to register on the mobile application.

**DEP:** FR1

**ID: FR3**

**TITLE:** User Log-in- Mobile application

**DESC: DESC:** Given that a User has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

**RAT:** For a User to register on the mobile application.

**ID: FR4**

**TITLE:** Doctor Registration - Mobile application

**DESC:** Given that a doctor has downloaded the mobile application, then the doctor should be able to register through the mobile application. The user must provide user-name, Phone No., password and e-mail address. The user can choose to provide a regularly used phone number.

**ID: FR5**

**TITLE:** Doctor Log-in - Mobile application

**DESC:** Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

**RAT:** For a user to register on the mobile application.

**DEP:** FR1

**ID: FR6**

**TITLE:** Change password

**DESC:** Given that a user has registered, then the user should be able to change his/her password by application.

**RAT:** For a user to change his/her password.

**DEP:** FR1, FR2, FR3, FR4, FR5.

**ID: FR7**

**TITLE:** Mobile application -Patient Options

**DESC:** Given that a user is logged in to the mobile application, then the first page that is shown should be the option page. The user should be able to select one of the options, according to several options. The search options are check blood sugar, make an appointment, health and workout chart.

**RAT:** For user to view options in menu

**DEP:** FR2, FR3, FR7

**ID: FR8**

**TITLE:** Mobile application – option check blood sugar

**DESC:** once a user selected this option, the user should be able to check his/her blood sugar using custom device, the user can view his/her previous readings, the user can also manually input readings and the user will also get an average that will tell the consistency of his/her blood sugar levels.

**RAT:** In order check blood sugar option is selected.

**DEP:** FR2, FR3, FR7

**ID: FR9**

**TITLE:** Mobile application – option make an appointment:

**DESC:** through this option the user will be able to make an appointment with any doctor we have in the app list.

**RAT:** In selection of option make an appointment

**DEP:** FR7

**ID: FR10**

**TITLE:** Mobile application – Doctor Option

**DESC:** Doctor when logged in will have options of Immediate Help (optional), Appointments, Check Blood Sugar levels and Order Medicine.

**RAT:** For a doctor to view options.

**DEP:** FR1, FR4, FR5, FR6

**ID: FR11**

**TITLE:** Doctor log-in -Option Appointments

**DESC:** In this option the doctor can mark appointments, check new appointment requests and check previous appointments.

**RAT:** For doctors to set appointments.

**DEP:** FR12

**ID: FR12**

**TITLE:** Mobile application – option check blood sugar

**DESC:** once a user selected this option, the user should be able to check his/her blood sugar using custom device, the user can view his/her previous readings, the user can also manually input readings and the user will also get an average that will tell the consistency of his/her blood sugar levels and will generate health and diet chart and will also recommend workout.

**RAT:** In order check blood sugar option is selected.

**DEP:** FR12

## **2.8 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## 2.9 Security Requirements

There is no security requirement in this project, but the patient must use device for its own blood sugar test in case of keeping record of glucose level record in database.

## 2.10 Software Quality Attributes

Following are the software quality attributes of this project.

- **AVAILABILITY:**

The appointment should be available for the patient.

- **CORRECTNESS:**

The glucose reading should be correct if entered manually.

- **MAINTAINABILITY:**

The doctor and the patient should have correct schedules to avoid any mishandling.

- **USABILITY:**

The appointment schedule should satisfy maximum number of patients.

## 2.11 Other Requirements

There are no other requirements for this project at this level.



## 2.12 System Requirements Chart

Following is a system requirement chart that shows priority, type and functional importance of a use case.

Table 1: System Requirement Chart.

<b>ID</b>	<b>Priority</b>	<b>Type</b>	<b>Source</b>	<b>Used in Use case</b>	<b>Description</b>
1	High	Functional	End User's	U1	Download an app from Play store.
2	High	Functional	End User's	U2	Creates new account to use the application.
3	High	Functional	End User's	U3	Log in required to use the application features.
4	Medium	Functional	End User's	U4	Forgot Password can be retrieved
5	High	Functional	End User's	U5	Main menu.
6	High	Functional	End User's	U6	Blood Sugar can be checked
7	High	Functional	End User's	U7	Appointments can be handled

## CHAPTER 3

### DESIGN AND METHODOLOGY

#### 3.1 Design

- Use case diagram
- Use case Diagram Description
- Design Class Diagram.
- Sequence Diagram
- ERD (Entity Relationship Diagram)

#### 3.2 Use Cases

The use case diagrams of Gluco-Care are as follows.

##### 3.2.1 Use case name and identifier

Table 2

<b>Unique Identifier:</b>	U1
<b>Name</b>	App download from Google play store
<b>Objective</b>	The customer will download an app to use it from Google play store.

<b>Actor(s)</b>	Android Users
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when a system user is not logged in to the system and goes to the login page.</p> <ol style="list-style-type: none"> <li>1. The user will open the Google play store.</li> <li>2. The user will search for an app in Play store.</li> <li>3. The user will click the download button.</li> <li>4. The user will give the app permissions required to download an app.</li> <li>5. The app will be started and downloaded completed.</li> </ol>	
<b>Alternate Flows</b>	
<b>TITLE</b>	<b>Description</b>
No Account	<p>If the User doesn't have Google pay store account:</p> <ol style="list-style-type: none"> <li>1. The app prompts the user to create the new google account.</li> <li>2. The user creates the account by providing necessary information.</li> <li>3. Google play store account successfully created</li> <li>4. The Basic Flow continues where the User will search an app in play store (see step 2 of the Basic Flow).</li> </ol>
<b>Pre-Conditions</b>	
<b>TITLE</b>	<b>Description</b>
Registered	The user must be logged in to play store account and internet is available.
<b>Post-Conditions</b>	
<b>TITLE</b>	<b>Description</b>
Success	The app will be successfully downloaded in the device.

Failure	The app unsuccessfully downloaded in the device due to some reasons.
<b>Notes/Issues</b>	
None	

Table 3

<b>Unique Identifier:</b>	U2
<b>Name</b>	Signup
<b>Brief Description</b>	A user of the System creates an account
<b>Actor(s)</b>	Android User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when a system user has no account and goes to the Signup page.</p> <ol style="list-style-type: none"> <li>1. The System prompts user for registration information, IP address, username, and password.</li> <li>2. The user enters in their information.</li> <li>3. System verifies information and creates account.</li> <li>4. The use case ends.</li> </ol>	
<b>Alternate Flows</b>	
Table 3	<b>Description</b>
<b>TITLE</b>	
Invalid Information Entered	<ol style="list-style-type: none"> <li>1. User clicks submit after entering information system asked for.</li> <li>2. System displays information with appropriate message to correct invalid information.</li> <li>3. User re-enters information.</li> <li>4. The Flow continues where the User enters new information.</li> </ol>

**Pre-Conditions**

<b>TITLE</b>	<b>Description</b>
Account	User must not be a registered user.
<b>Post-Conditions</b>	
<b>TITLE</b>	<b>Description</b>
Success	The user entered successful information and is returned to the home page as a Logged In User
Failure	User is unable to sign up for one or more reasons and is returned to the login/signup selection page.
<b>Extension Points</b>	
None	

Table 4

<b>Unique Identifier:</b>	U3
<b>Name</b>	Login
<b>Objective</b>	A user of the System logs in to the System.
<b>Actor(s)</b>	Registered Users
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when a system user is not logged in to the system and goes to the login page.</p> <ol style="list-style-type: none"> <li>1. The System prompts the user for a IP address and password or signup new account</li> <li>2. The user enters Username and password.</li> <li>3. The system validates the entered contact no. and password by calling an API via Internet connection, making sure that the entered contact no. is a valid contact no. in the System, and that the required password is entered for the entered contact no.</li> </ol>	

4. When the user taps on the login button the system sends data to the API for authentication.
5. The user is signed in and returned to the home page as a Logged in User.
6. The use case ends.

#### Alternate Flows

TITLE	Description
User Fails Authentication	<p>If the User entered an invalid contact no. and/or password, the following occurs:</p> <ol style="list-style-type: none"> <li>1. The system describes the reasons why the User failed authentication.</li> <li>2. The system presents the User with suggestions for changes necessary to allow the User to pass authentication.</li> <li>3. The system prompts the User to re-enter the valid information.</li> <li>4. The Flow continues where the User enters new information.</li> </ol>

#### Pre-Conditions

TITLE	Description
Registered	User must have an account.

#### Post-Conditions

TITLE	Description
Success	The User is authenticated and the system displays a home page based on the user type.
Failure	User is unable to log in for one or more reasons.

#### Notes/Issues

None

Table 5

<b>Unique Identifier:</b>	U4
<b>Name</b>	Forgot Password
<b>Brief Description</b>	User will recover the password
<b>Actor(s)</b>	Registered User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when the user taps on the forget password link.</p> <ol style="list-style-type: none"> <li>1. The system will validate the user by sending a unique code to email or text message.</li> <li>2. The system will validate the code sent and code entered.</li> <li>3. The system sends the notification to administrator for new password request and generate a new password.</li> <li>4. Use Case ends.</li> </ol>	
<b>Alternate Flows</b>	
<b>TITLE</b>	<b>Description</b>
Invalid Contact Number Entered	<ol style="list-style-type: none"> <li>1. System displays information with appropriate message to correct phone number and email address.</li> <li>2. User phone number and email address.</li> <li>3. The Flow continues where the User enters phone number and email address.</li> </ol>
<b>Pre-Conditions</b>	
<b>TITLE</b>	<b>Description</b>
Account	User must have an account.
<b>Post-Conditions</b>	
<b>TITLE</b>	<b>Description</b>
Success	Password will be reset and updated in database.



Notes/Issues
None

Table 6

<b>Unique Identifier:</b>	U5
<b>Name</b>	Main Menu
<b>Brief Description</b>	The user will able to view menu on the basis of its role(patient and doctor)
<b>Actor(s)</b>	Registered Android User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when the user successfully login</p> <ol style="list-style-type: none"> <li>1. The System displays the list of options according to the role (patient and doctor).</li> <li>2. Use Case ends.</li> </ol>	
<b>Alternate Flows</b>	
<b>Title</b>	<b>Description</b>
Menu list Error	<ol style="list-style-type: none"> <li>1. Prompt message of network not available, try again later.</li> <li>2. User will refresh the application.</li> <li>3. The Flow continues.</li> </ol>
<b>Pre-Conditions</b>	
<b>Title</b>	<b>Description</b>
Permissions	User must be logged in.
<b>Post-Conditions</b>	

<b>Title</b>	<b>Description</b>
Success	Successfully views the menu.
<b>Notes/Issues</b>	
None	

Table 7

<b>Unique Identifier:</b>	U6
<b>Name</b>	Check Blood Sugar
<b>Brief Description</b>	User can Check Blood sugar by using custom device or by manually adding the values and the user can also view previous readings.
<b>Actor(s)</b>	Logged in Android User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when the user selects the Check Blood Sugar option from menu.</p> <ol style="list-style-type: none"> <li>1. The application will ask the user to connect the device and place a blood sample or enter the reading (if manually is selected).</li> <li>2. The user will get his/her readings.</li> <li>3. The user can review his or her previous readings.</li> <li>4. The user will also be shown the doctor's chart.</li> <li>5. Use Case ends.</li> </ol>	
<b>Pre-Conditions</b>	
<b>Title</b>	<b>Description</b>
Check Blood Sugar	Check Blood Sugar option must be selected by the user.
Permission	User must be logged in and have selected Check Blood Sugar option from within the menu.

<b>Alternate Flows</b>	
<b>Title</b>	<b>Description</b>
Other options	<ol style="list-style-type: none"> <li>1. The user must go back to the menu screen.</li> <li>2. User must select Check Blood Sugar option.</li> <li>3. The Basic Flow continues</li> </ol>
<b>Post-Conditions</b>	
<b>Title</b>	<b>Description</b>
Success	Blood Sugar Checked or reading added successfully
Failure	Blood Sugar Reading not gathered
<b>Extension Points</b>	
None	

Table 8

<b>Unique Identifier:</b>	U7
<b>Name</b>	Health And Diet Chart and Workout
<b>Brief Description</b>	The app will generate a health chart on the basis of patient's blood sugar readings and will recommend appropriate workout on the basis of patient's current blood sugar condition.
<b>Actor(s)</b>	Logged in Android User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when the user selects the Check Blood Sugar option from menu.</p> <ol style="list-style-type: none"> <li>1. The application will generate a char on the basis of user's Blood sugar readings.</li> <li>2. The app will also recommend workout on the basis of the Users blood sugar condition.</li> </ol>	

3. Use Case ends.	
<b>Pre-Conditions</b>	
<b>Title</b>	<b>Description</b>
Diet and Workout	Diet and Workout option must be selected by the user.
Permission	User must be logged in and have selected Check Blood Sugar option from within the menu.
<b>Alternate Flows</b>	
<b>Title</b>	<b>Description</b>
Other options	<ol style="list-style-type: none"> <li>1. The user must go back to the menu screen.</li> <li>2. User must select Diet and Workout option.</li> <li>3. The Basic Flow continues</li> </ol>
<b>Post-Conditions</b>	
<b>Title</b>	<b>Description</b>
Success	Diet and Workout Checked successfully
Failure	No blood sugar readings
<b>Extension Points</b>	
None	

Table 9

<b>Unique Identifier:</b>	U8
<b>Name</b>	Appointments

<b>Brief Description</b>	The patient can place appointment and the Doctor can answer them.
<b>Actor(s)</b>	Logged in Android User
<b>Flow of Events</b>	
<b>Basic Flow</b>	
<p>This use case starts when the user selects the appointment option from menu.</p> <ol style="list-style-type: none"> <li>1. The Patient requests appointment from the doctor.</li> <li>2. The doctor will accept if time available.</li> <li>3. If the time is not available, the doctor will provide available time and ask patient confirmation.</li> <li>4. The confirmed time will be provided by the patient.</li> <li>5. Use Case ends.</li> </ol>	
<b>Alternate Flows</b>	
<b>Title</b>	<b>Description</b>
Simultaneous Appointments of same date and time	<ol style="list-style-type: none"> <li>1. The system keeps the appointment received first on top and next on bottom.</li> <li>2. The Basic Flow continues</li> </ol>
<b>Pre-Conditions</b>	
<b>Title</b>	<b>Description</b>
Appointment	Patient must place an appointment and should be logged in.
Permission	Appointments option must be selected from within the menu.
<b>Post-Conditions</b>	
<b>Title</b>	<b>Description</b>
Success	Appointment set
Failure	Appointment not approved
<b>Extension Points</b>	

None
------

3.2.2 Use case Diagram (refined and updated)

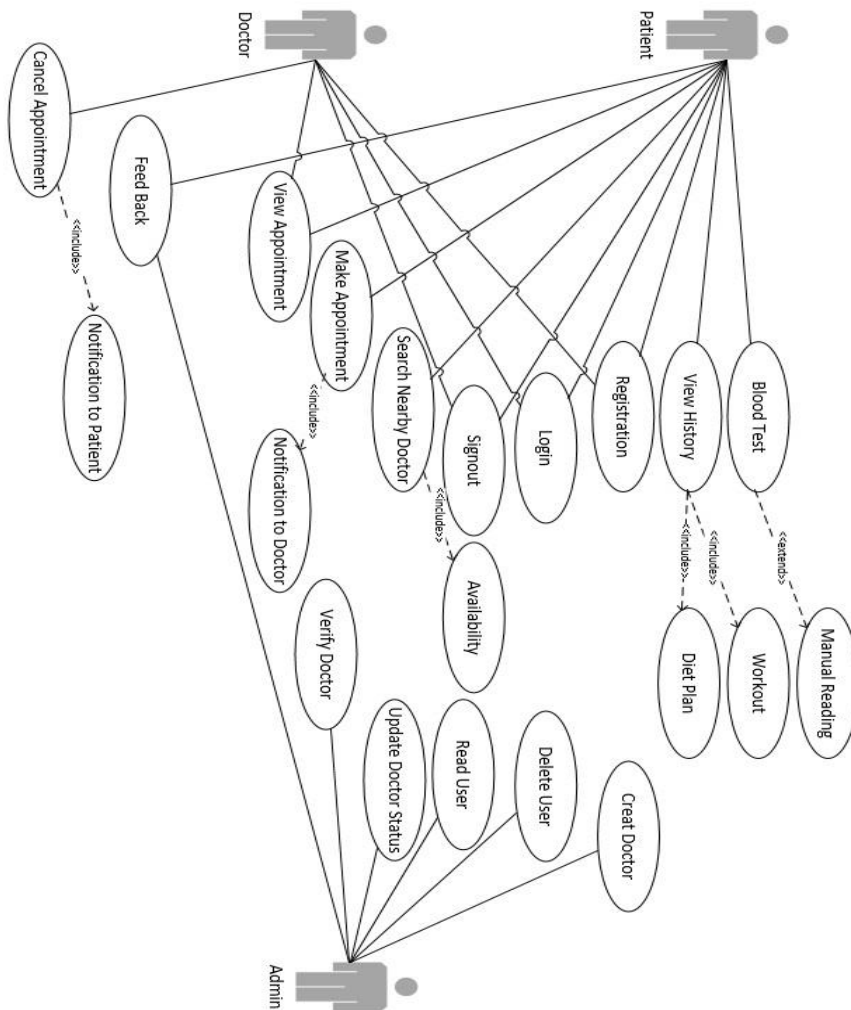


Figure 1: Use Case Diagram

### **3.3 Sequence Diagram Description**

A Sequence diagram depicts the sequence of actions that occur in a system. A Sequence diagram is two-dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects. There exists sequence diagram against every use case.



### 3.3.1 Sequence Diagram

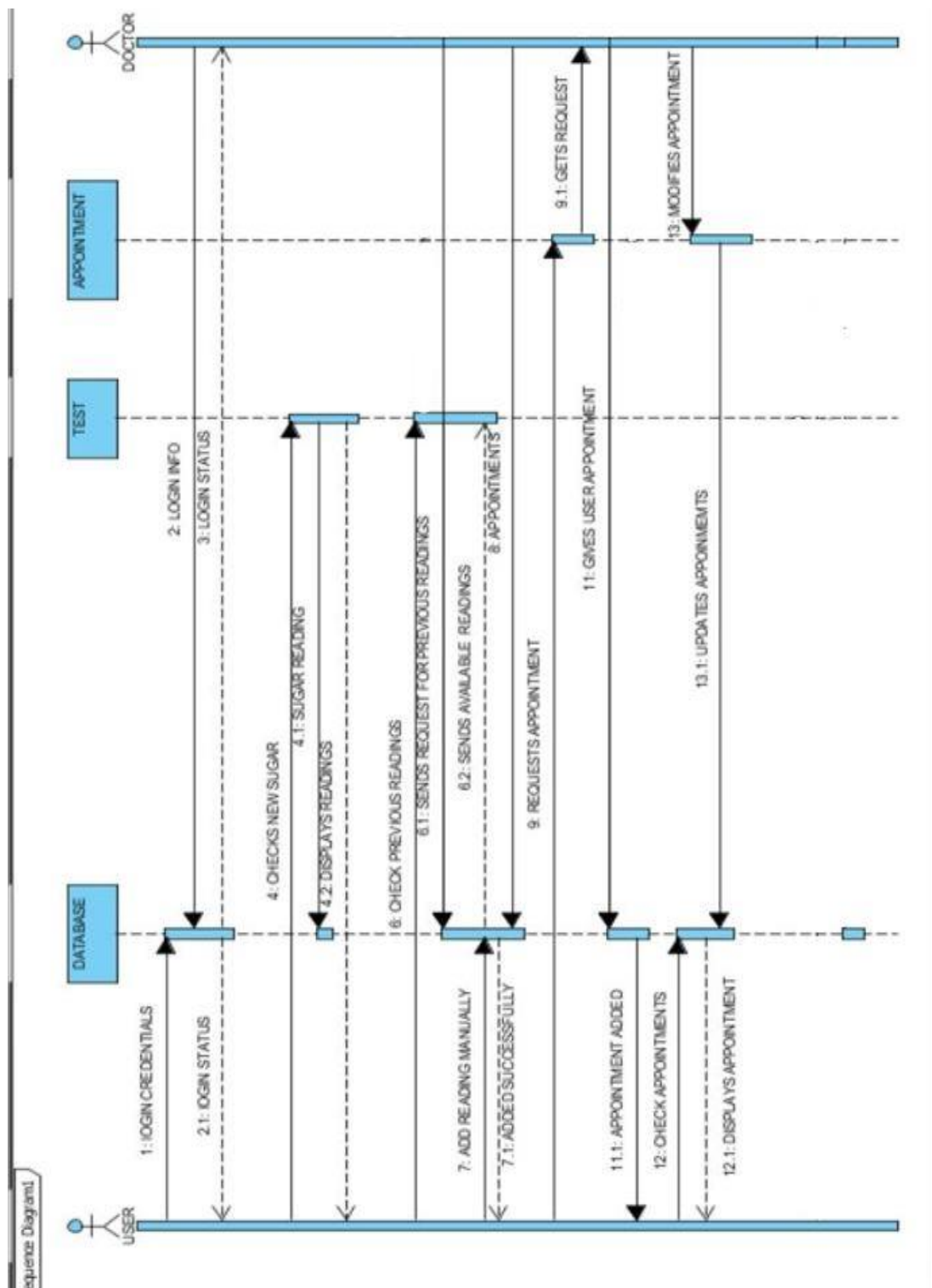


Figure 2: Sequence diagram

### 3.4 Collaboration Diagram

Following are the collaboration diagrams of Gluco care

#### 3.4.1 Login

Doctor and Patient will log in to their respective applications in this diagram.

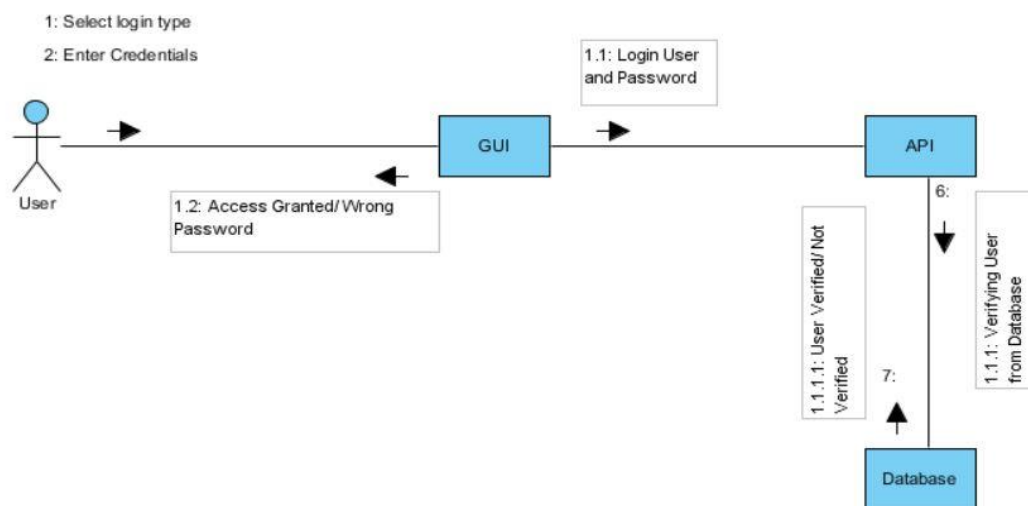


Figure 3: Login Collaboration Diagram

### 3.4.2 Sign Up

Patient and Doctor both can sign up in the following diagram.

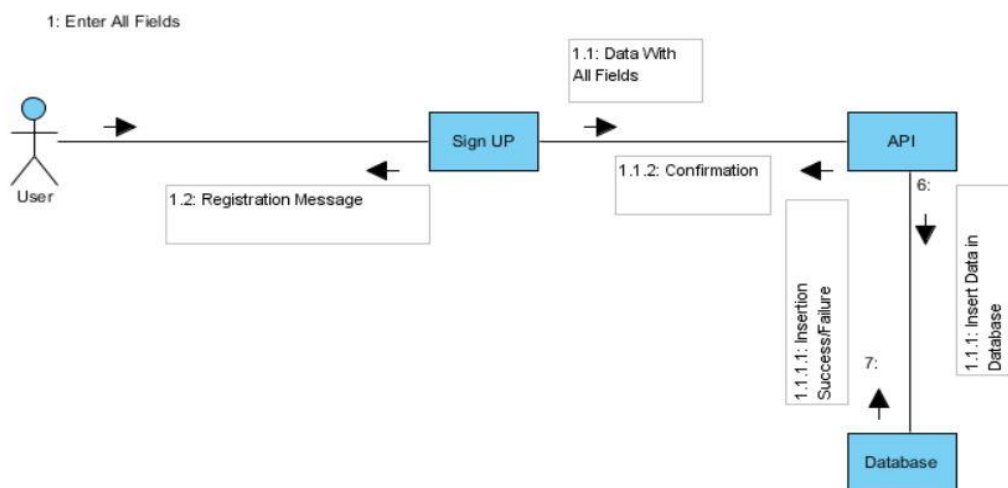


Figure 4: Sign Up Collaboration Diagram

### 3.4.3 Test

In this diagram the Patient and the Doctor can check their blood sugar levels by using the custom device or by manual input.

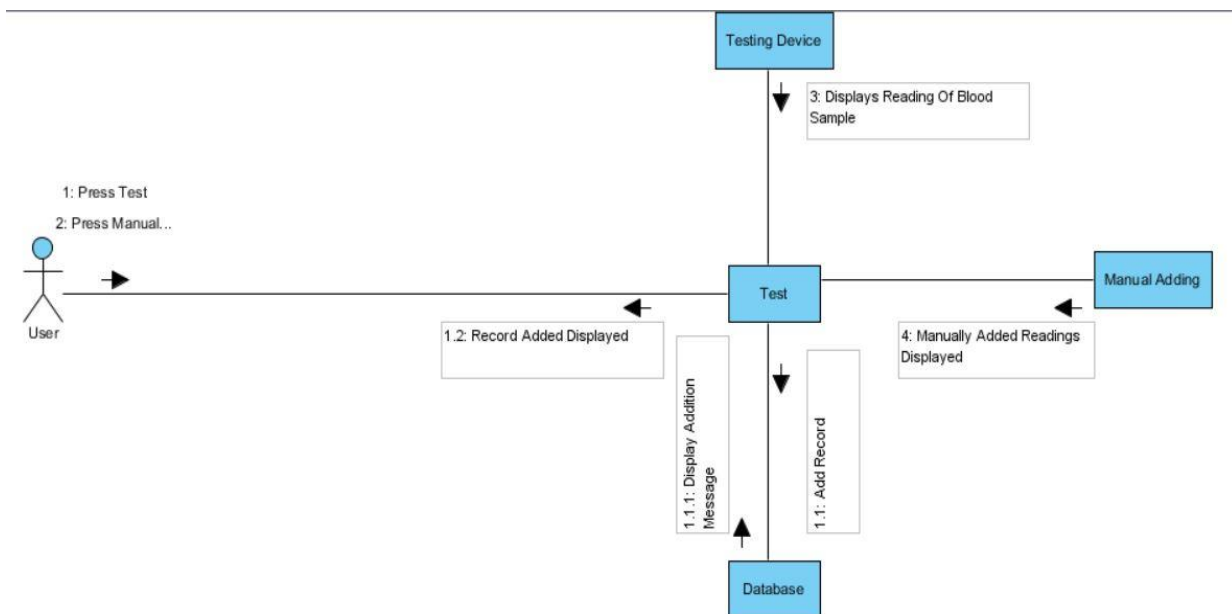


Figure 5: Test Collaboration Diagram

### 3.4.4 View Test

In this diagram the user will check their previous test results.

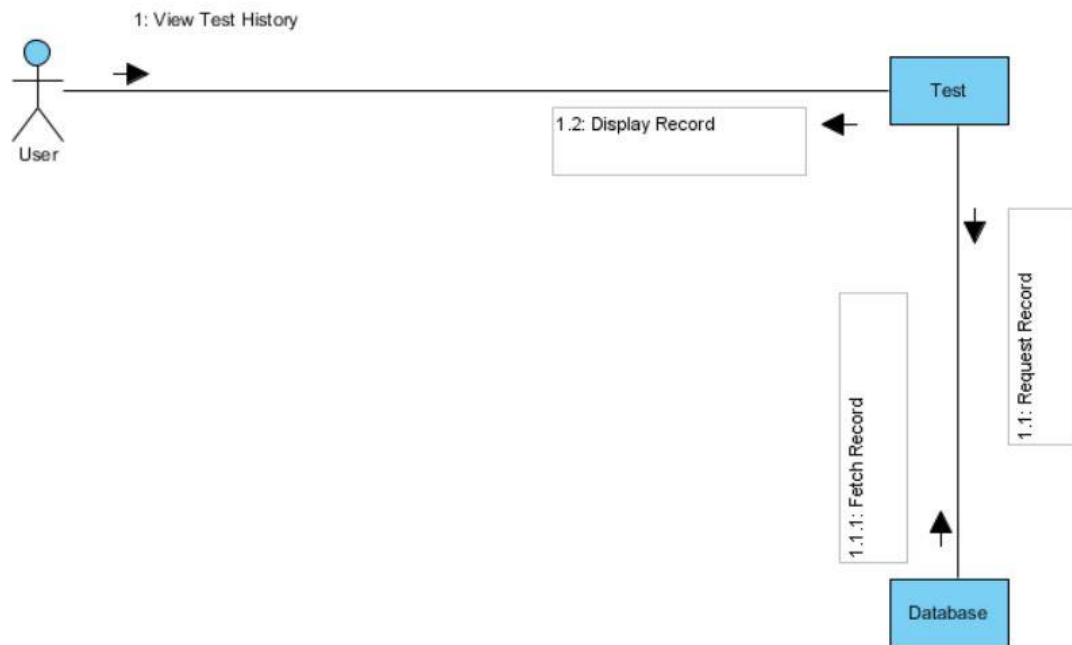


Figure 6: View Test Collaboration Diagram

### 3.4.5 Appointment

Patients can book appointment in the following diagram.

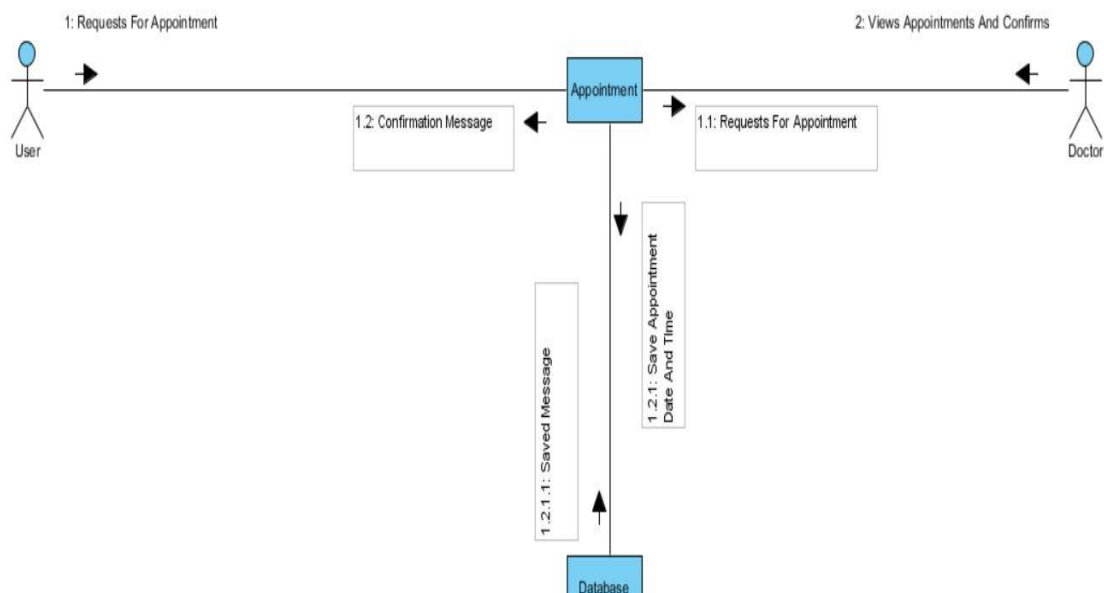


Figure 7:Appointment Collaboration Diagram

### 3.4.6 View Appointment

In the following diagram Doctor can view his appointments and can view the present or future appointments.

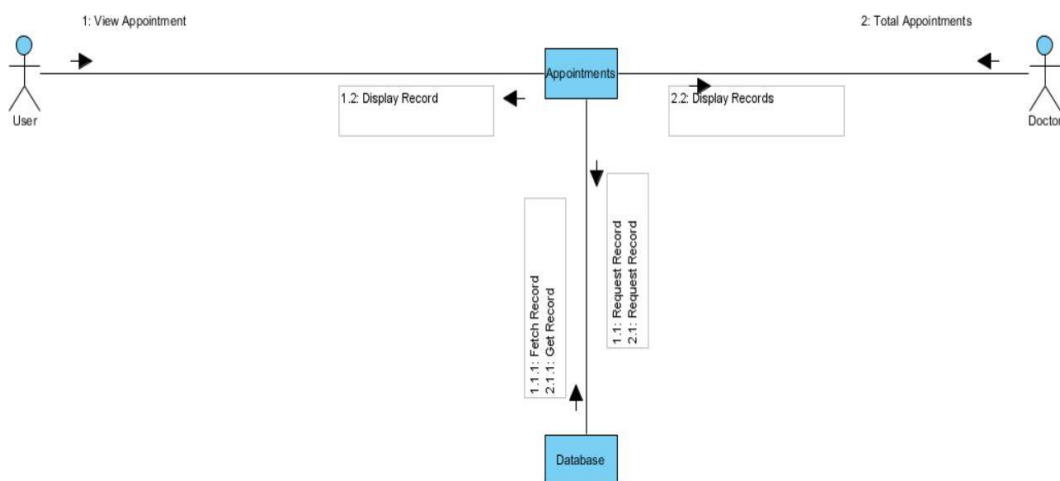


Figure 8: View Appointment Collaboration Diagram

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 Implementation

#### 4.2 Android Application

The IDE we are using for the android application development is Android Studio. Android Studio is an integrated development environment (IDE) from Google that provides developers with tools needed to build applications for the Android OS platform. Application will give us the access of checking blood sugar levels, will generate reports and diet charts according to the received readings and patients can also book appointments through this application.

##### 4.2.1 Libraries Used for UI

The API's we built for the interaction of application with the database is NODE JS. Node.js is known for Event-driven I/O server-side JavaScript environment.

- Node.js is fast: Node.js uses JavaScript in the backend, and that's enough to understand how fast the codes execute.
- The ever-growing NPM: The number of modules in the Node Package Manager (NPM) has increased at a considerable pace.
- It's perfect for handling lots of requests that are I/O driven (e.g. operations on database) and scales very nicely.
- Ideal for data-heavy websites and apps



- Capable of handling traffic spikes

#### **4.2.1.1 Notification Library:**

Android Built-in library us used for notification.

### **4.3 Server-Side Development**

The API's we built for the interaction of application with the database is NODE JS. Node.js is known for Event-driven I/O server-side JavaScript environment.

- Node.js is fast: Node.js uses JavaScript in the backend, and that's enough to understand how fast the codes execute.
- The ever-growing NPM: The number of modules in the Node Package Manager (NPM) has increased at a considerable pace.
- It's perfect for handling lots of requests that are I/O driven (e.g. operations on database) and scales very nicely.
- Ideal for data-heavy websites and apps
- Capable of handling traffic spikes

#### **4.3.1 Libraries used in NODE JS:**

- AJV: It is a module for Node.js applications to validate request JSON.
- Mustache: Mustache.js is an implementation of the mustache template system in javascript. mustache is a logic-less template syntax

### **4.4 Backend Development**

The database used for the backend development is PostgreSQL or simply called Postgres and it's a type of object-relational database server with open source user license. Postgres database server has evolved through more than 15 years of development and maintenance and is considered one of the most reliable databases

with extensive focus on data reliability and correctness. Postgres supports all the modern features of any database system and can be installed and run on variety of platforms including Linux, Windows, and all the versions of UNIX.

There are several reasons why we should use Postgres for database development purposes

### **Open Source**

The first major benefit of using Postgres is that it is open source and can be customized.

### **Large Developer Community**

Postgres has been in the market for more than 15 years and its developer community has immensely grown in this time. Large developer community means good support and help in solving database related problems.

### **Portable**

Good thing about Postgres is that it is portable with almost all the major platforms and programming languages. This database is ideal for applications targeting multiple platforms.

### **Developers Tool and GUI**

Postgres database server does not require extensive command line configurations. Several tools and GUI interfaces have been developed which can aid you in easy installation and management of the database server.

### **Reliability and Stability**

Postgres is worldwide recognized as being the most reliable and stable database. The chances of database crashing are minimal and even if database crashes there are ways and features which allow you to restore and recover the data.

## **4.5 Hardware**

The Arduino Nano plays a pivotal role. It gets the reading using standard sugar checking strip when the blood sample is placed on it. The Arduino Nano calculates the

readings and sends it to the OTG point from where the android application fetches the information and displays it.

#### **4.6 Record**

For database handling we are using PostgreSQL. Four types of databases will be created Doctor Info, Patient Info, Test Readings and Appointments. These databases will be handled by the admin and the admin will also have a database of admin info.

#### **4.7 Main Server**

To manage all users and database. Main server will be required to save appointment info regarding patient and doctors.

## **CHAPTER 5**

### **USER MANUAL**

#### **5.1 User Android Application**

The application is created for android platform and android user

##### **5.1.1 Getting Started**

The application begins when the user clicks on Gluco care icon

### 5.1.2 Application Welcome Page

When the application is launched for the first time, login/signup page will be displayed. If the user is not registered or the user wants to create an account, he must choose Sign Up button which will take him to the Sign-up screen.

If the user is already registered, then he must choose login button that will take him to login screen.

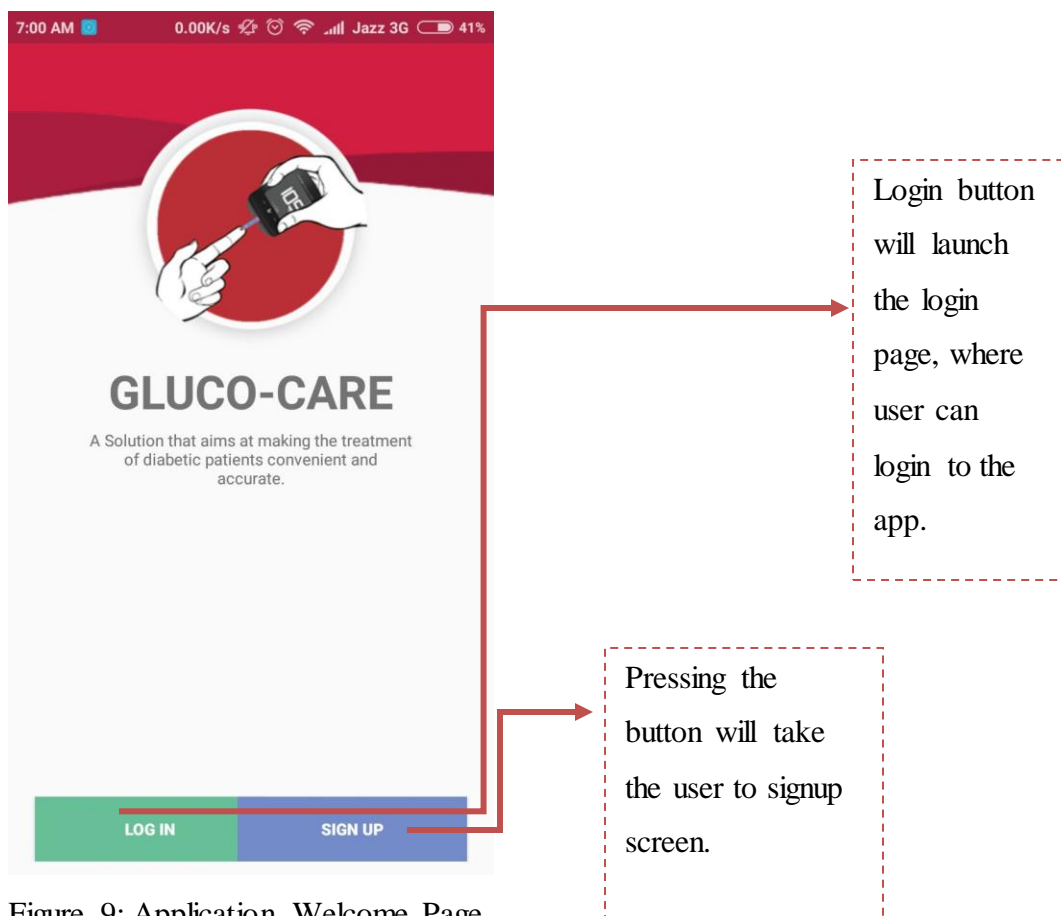


Figure 9: Application Welcome Page

### 5.1.3 The User Type

In this page the user can select which type of a user he/she is like user can select as they want to use the app as a doctor or as a patient. Same page is used when the user signs up on the page.

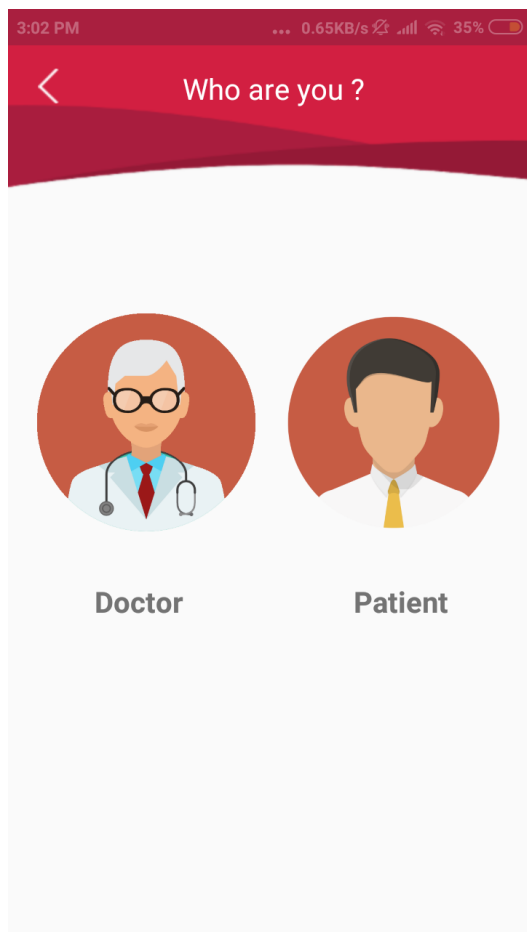


Figure 10: The User Type

### 5.1.4 Sign In

To use the application, you need to sign in using the email address and password of your Gluco-Care account.

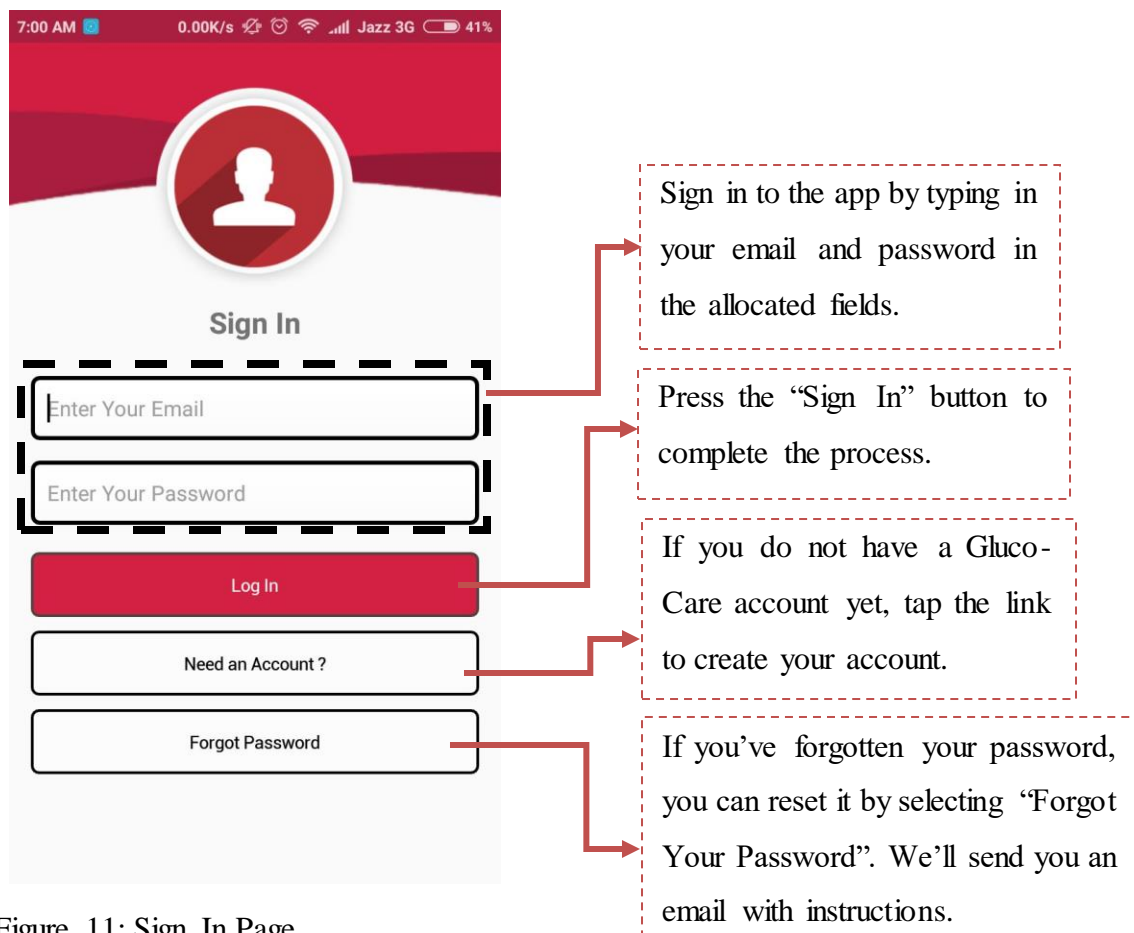


Figure 11: Sign In Page

### 5.1.5 Sign Up

To use the Gluco-Care, you must be a registered user.

7:00 AM 1.06K/s Jazzy 3G 41%

**Create an Account**

Name

Email

Password

Confirm Password

Contact Number

Create Account

Enter your Full Name in this field. Your Full Name should have space between. such as: FirstName LastName

Enter Your Email address. Email address format should be like example@provider.com

Enter Your password. password length must be greater than 6. Your password should not contain name.

Re-type your password

Type your contact number. Format: +92xxxxxxxx

After inputting all the required filed. Tap the Button to create your account.

Figure 12: Create account page



## 5.2 Application Overview

After successfully log in or sign up, application main/home screen will be launch. It will contain Drawer Menu button and a meter to show glucose reading as shown in the figure and other functionality as described below in detail.

### 5.2.1 Patient Profile Screen

This page shows the information of the patient like his name, mobile number, blood group and his blood sugar type.

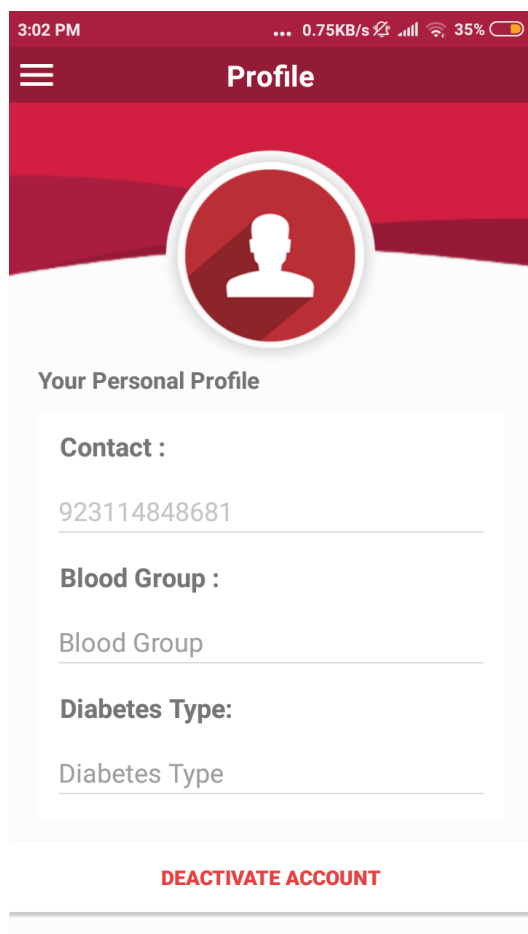


Figure 13: Patient Profile Screen

## 5.2.2 Home Screen

Home screen is the main page of the application that shows the glucose reading and shows whether the device is connected to the smart phone or not.

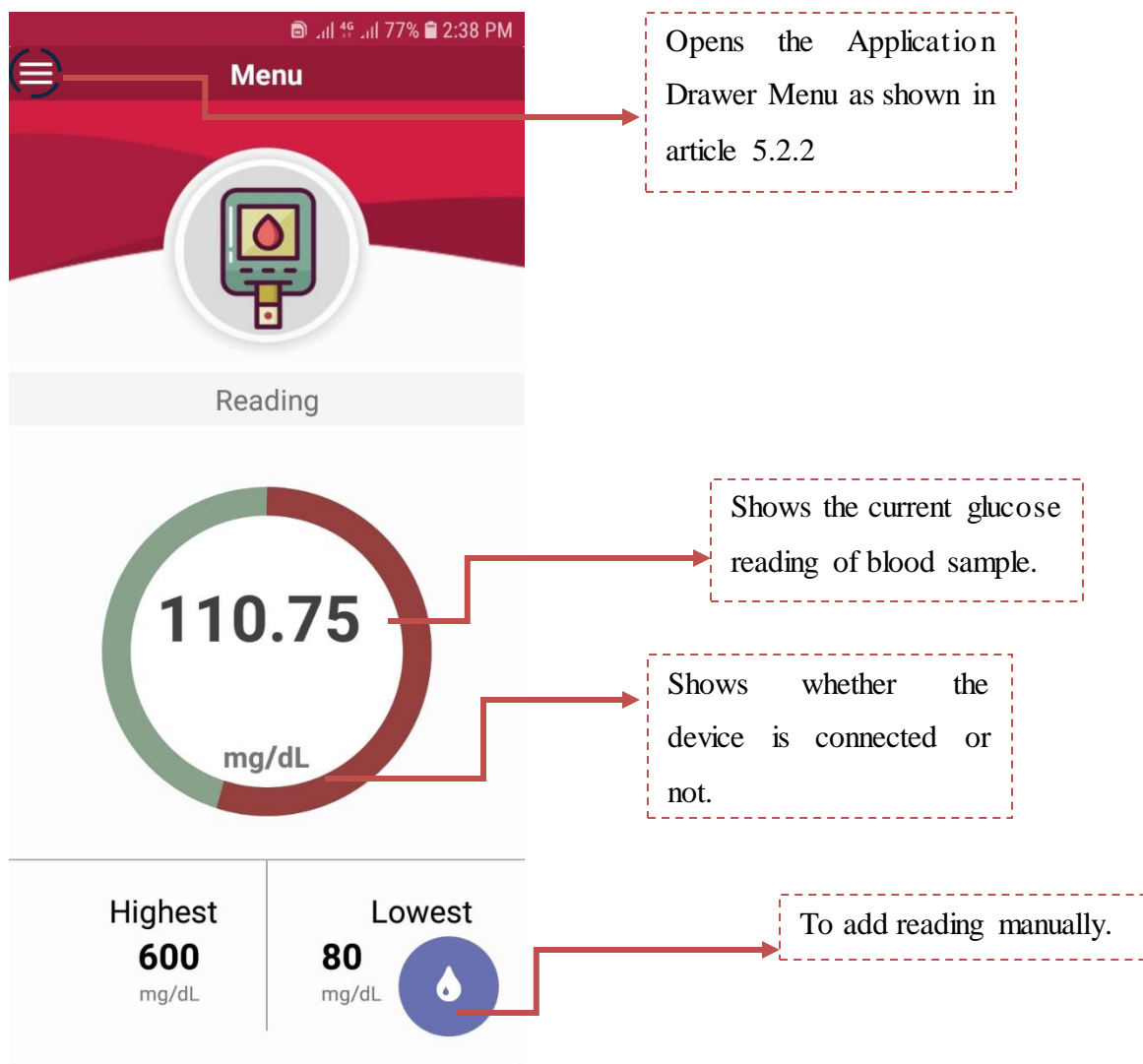


Figure 14: Main page

### 5.2.3 App Navigation Drawer

The application's navigation drawer allows the user to access all the available tabs in the application.

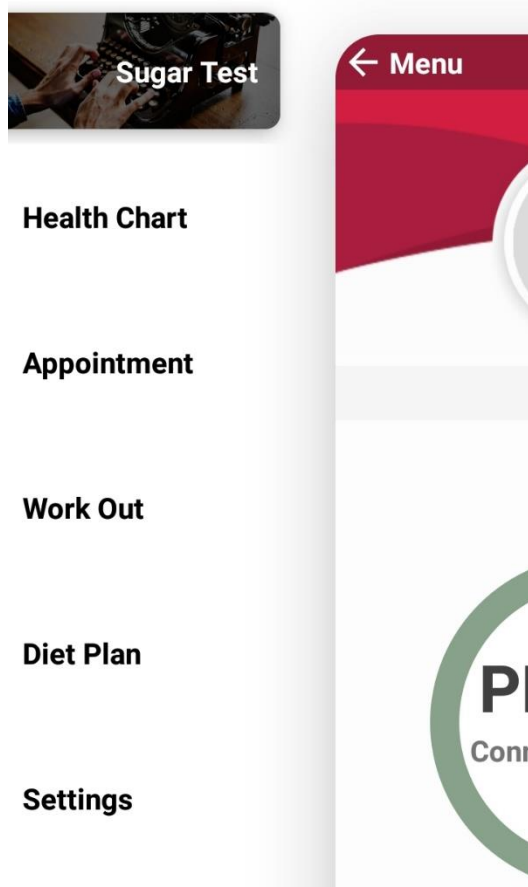


Figure 15: Application Navigation Drawer

### 5.2.4 Health Chart

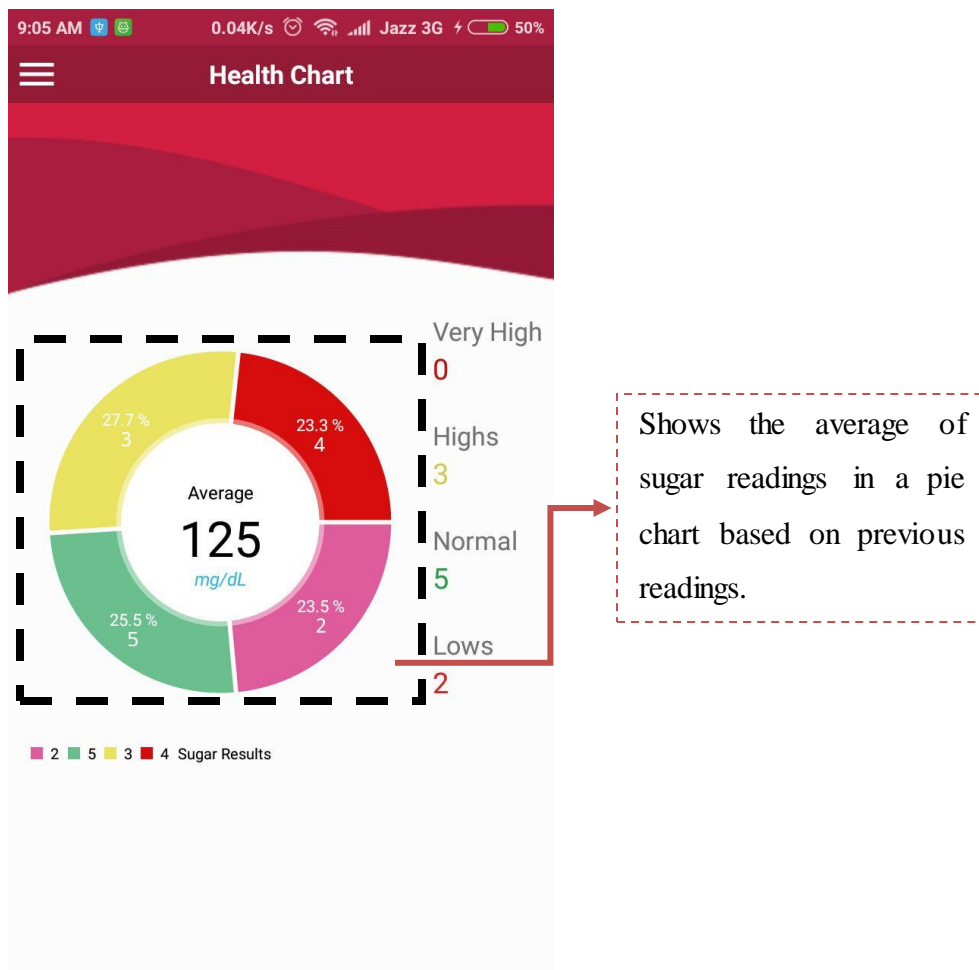


Figure 16 Health Chart

This is the menu that will be shown to the patient whenever they log in. In this menu they can select “Health Chart” to see the chart of checking blood sugar schedule. On clicking “Test Alert” user can enable or disable the notification of checking blood sugar. “Sugar Test” option will direct the user to a menu where the user can test their blood sugar levels by using the Gluco-Care device. By clicking the “Appointment” option the patient can book appointment with one of the best doctors available in our list. “Prescription” option will enable the patient to view or add the prescription.

### 5.2.5 Diet Chart

In this screen the patient can see what type of diet they should follow and the diet is recommended on the basis of the patient's previous readings.

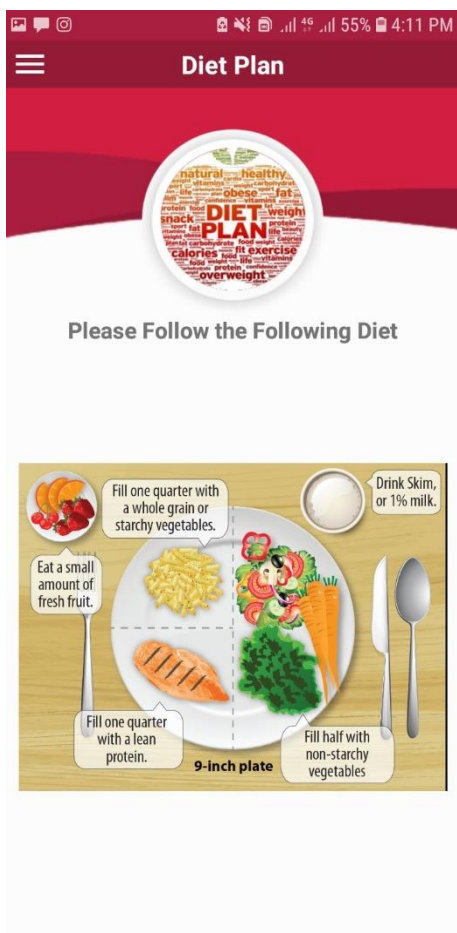


Figure 17: Diet Plan

### 5.2.6 Workout Chart

This page shows the workout which is generated through the blood sugar average readings.

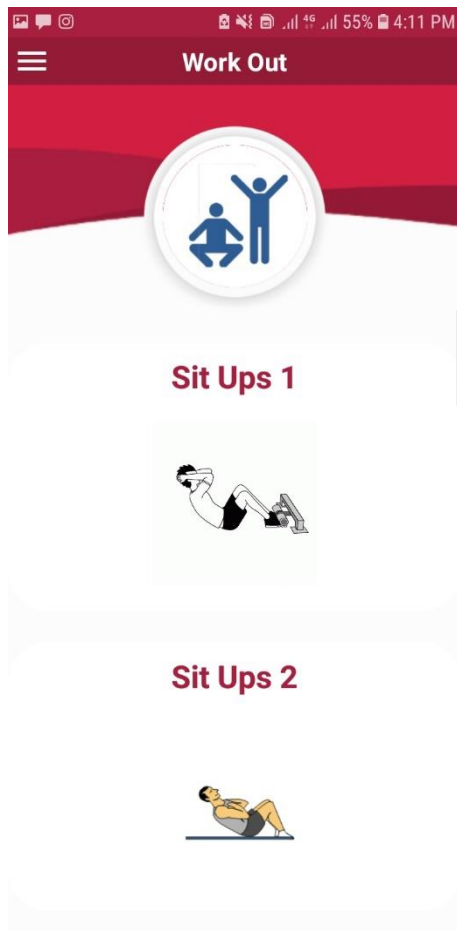


Figure 18: Workout Chart

## 5.2.7 Patient Settings

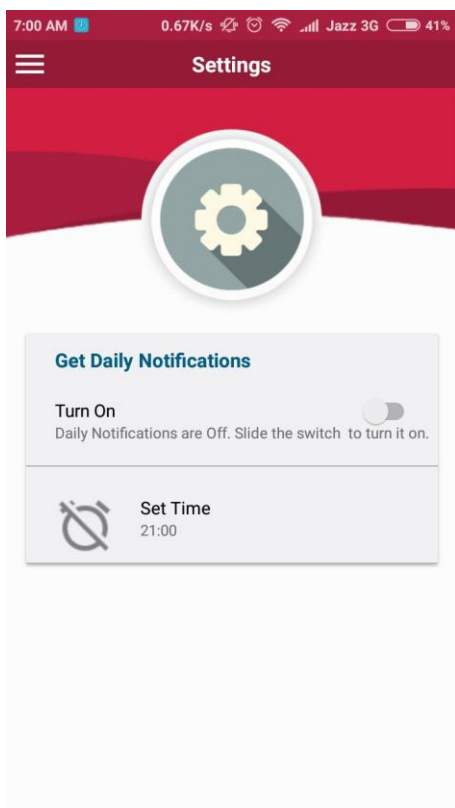


Figure 19 Patient Setting

In this menu the patient can Toggle his or her settings. They can update their account info in “My Account”, Check Prescriptions in “Prescriptions”, Delete Account in “Delete Account”, Check previously visited doctor in “Doctor’s History” and can log out.

### 5.3 Doctors End

Doctor will control his/her functions from here.

#### 5.3.1 Main

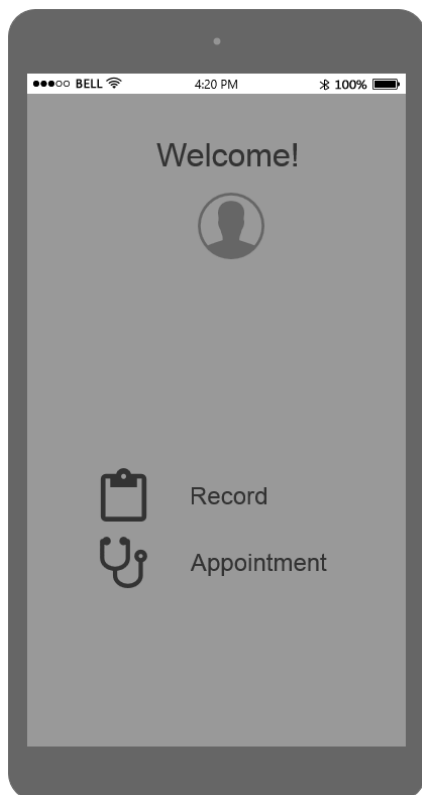


Figure 20 Doctor Main Page

In this menu the doctor will have option of “Record” where the doctor will be able to check the history of treated patients. In “Appointment” will be able to accept or reject the appointment requests by the patients.



### 5.3.2 Doctor's Settings

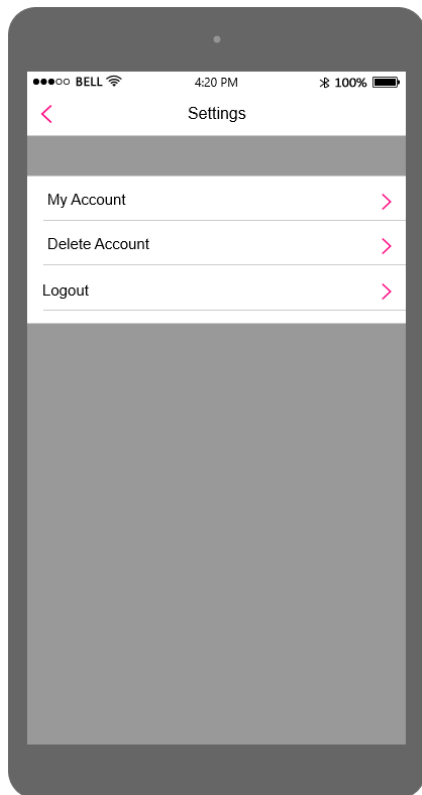


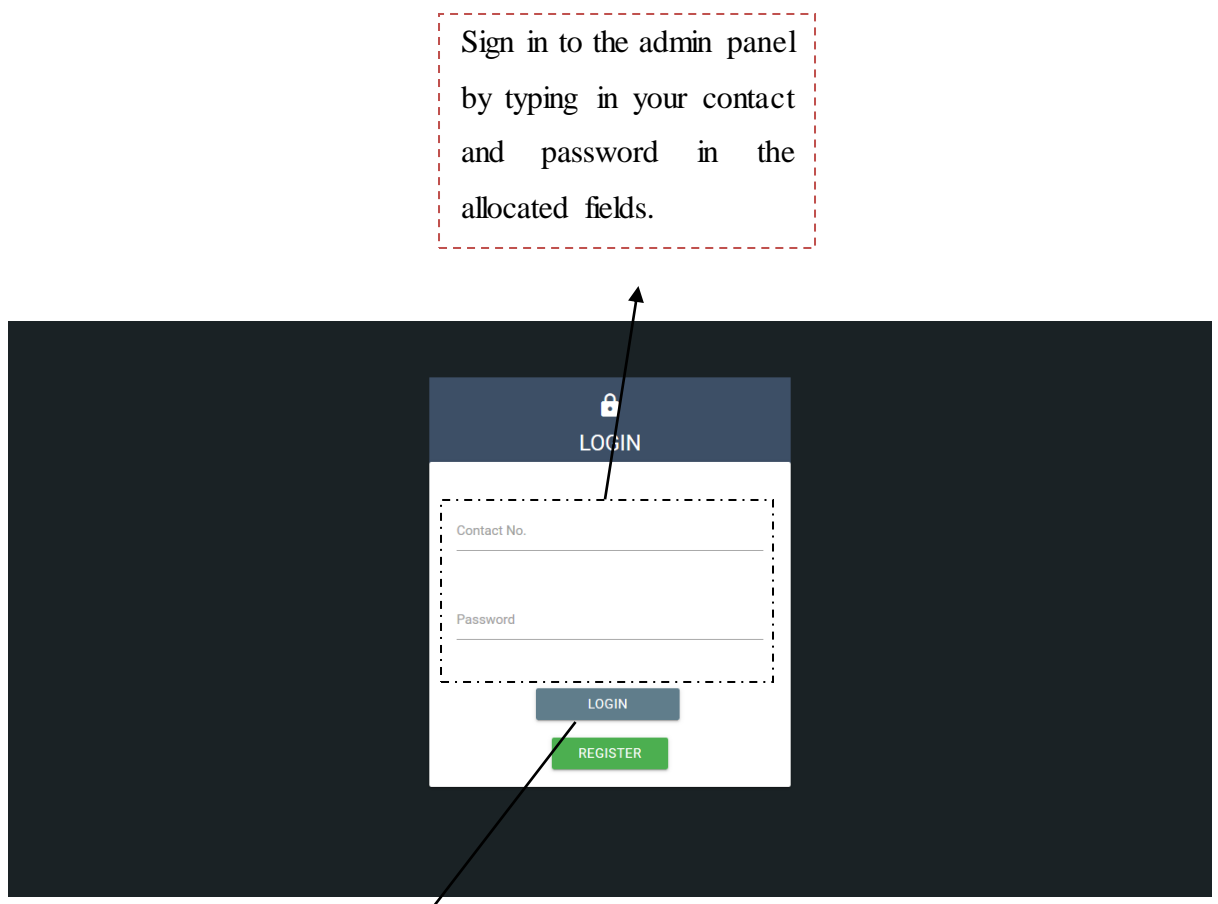
Figure 21 Doctor's Setting

In this menu the doctor can change his setting and can delete his account if he wants to.

## 5.4 Web Application (Admin Panel)

### 5.4.1 Login Page

Admin can access panel with his login credentials. Login credentials will be provided by the developers for the first-time use.



Press the “Log in” button to complete the process.

## 5.4.2 Admin panel Home Screen

Admin panel contain the information of the Stats. Overall stats as described below in figures. Admin can view the active appointments and list of Pending Doctor. Admin also has access to verify the doctors after interviewing them. Admin can also view the record of patients.

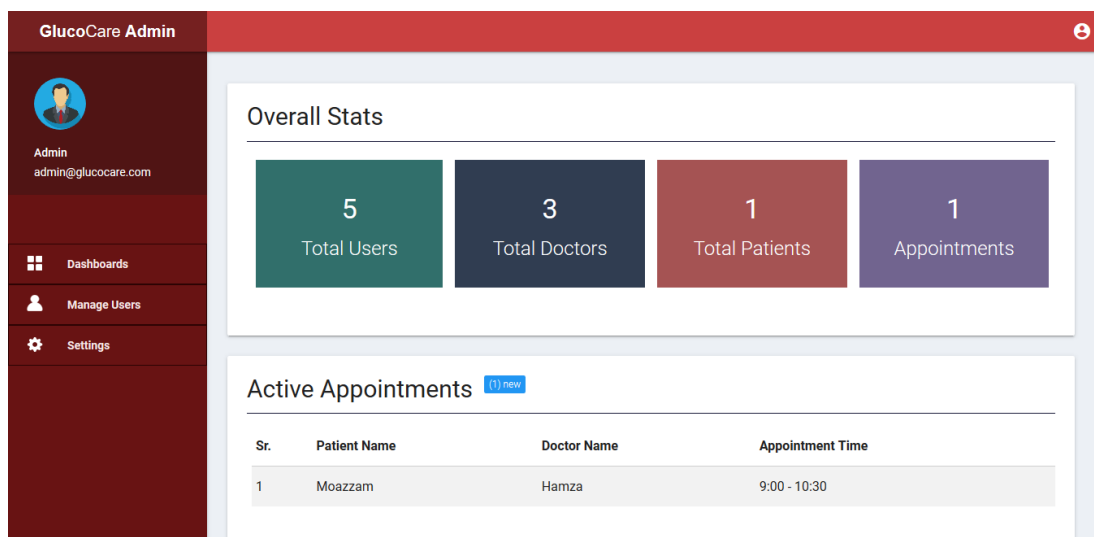


Figure 23: Home Screen

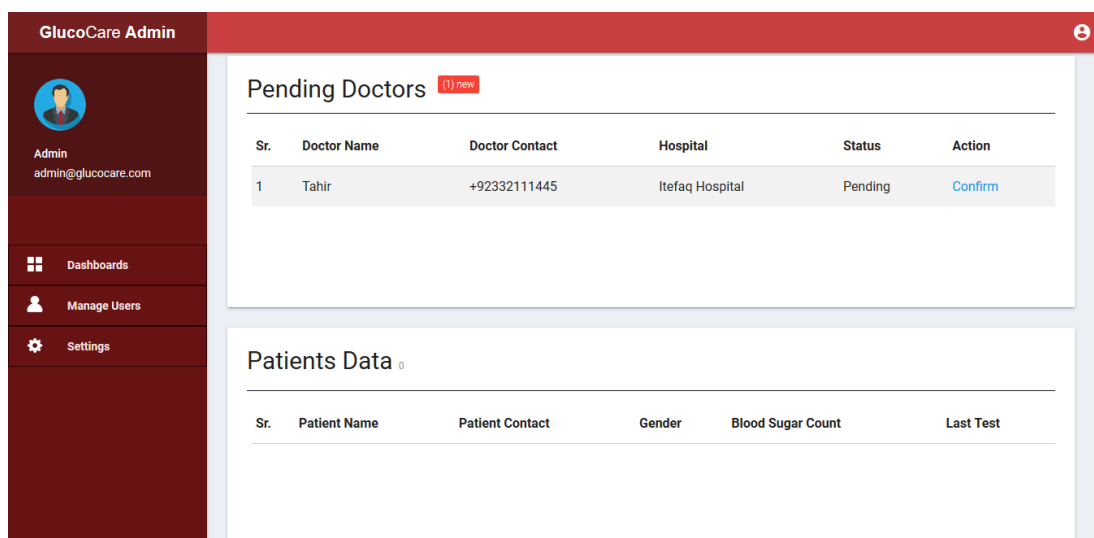


Figure 24: Pending Doctor Verification

### 5.4.3 Admin panel Home Screen

Admin panel contain the information of the Stats. Overall stats as described below in figures. Admin can view the active appointments and list of Pending Doctor. Admin also has access to verify the doctors after interviewing them. Admin can also view the record of patients.

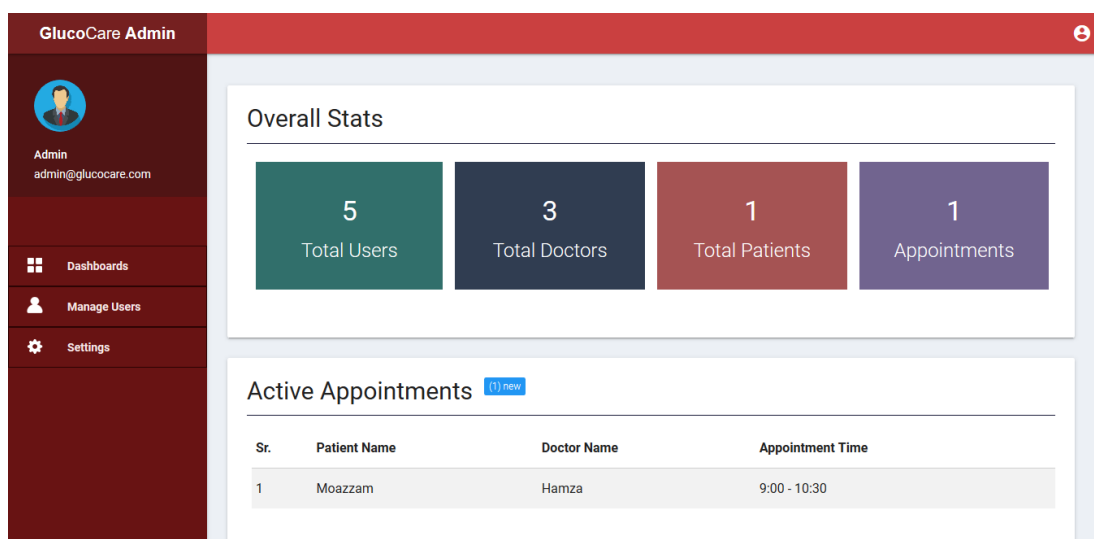


Figure 25: Home Screen

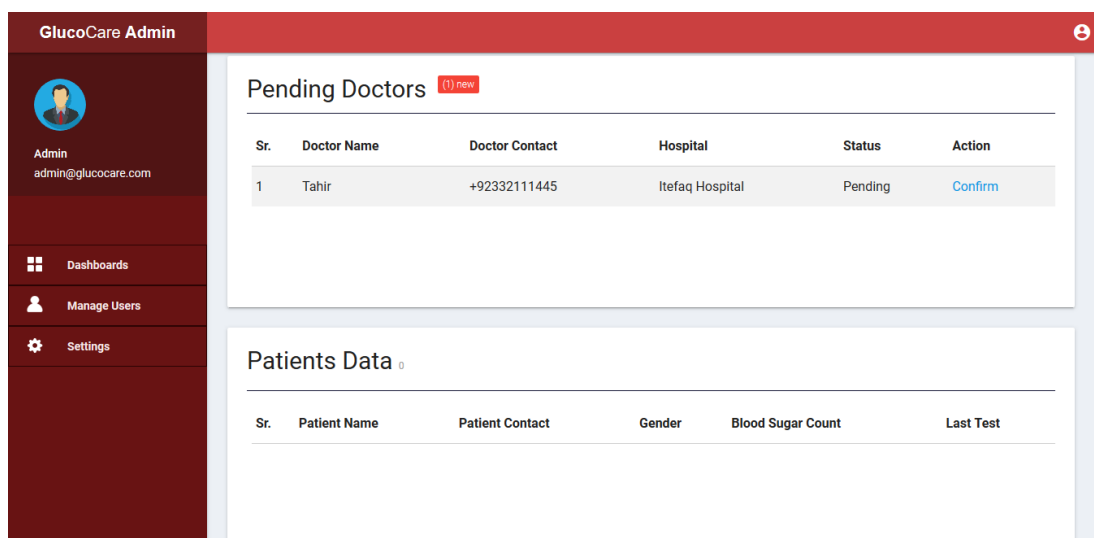


Figure 26: Pending Doctor Verification

#### 5.4.4 Entity Relationship Diagram (ERD)

Following is the ERD of the Project.

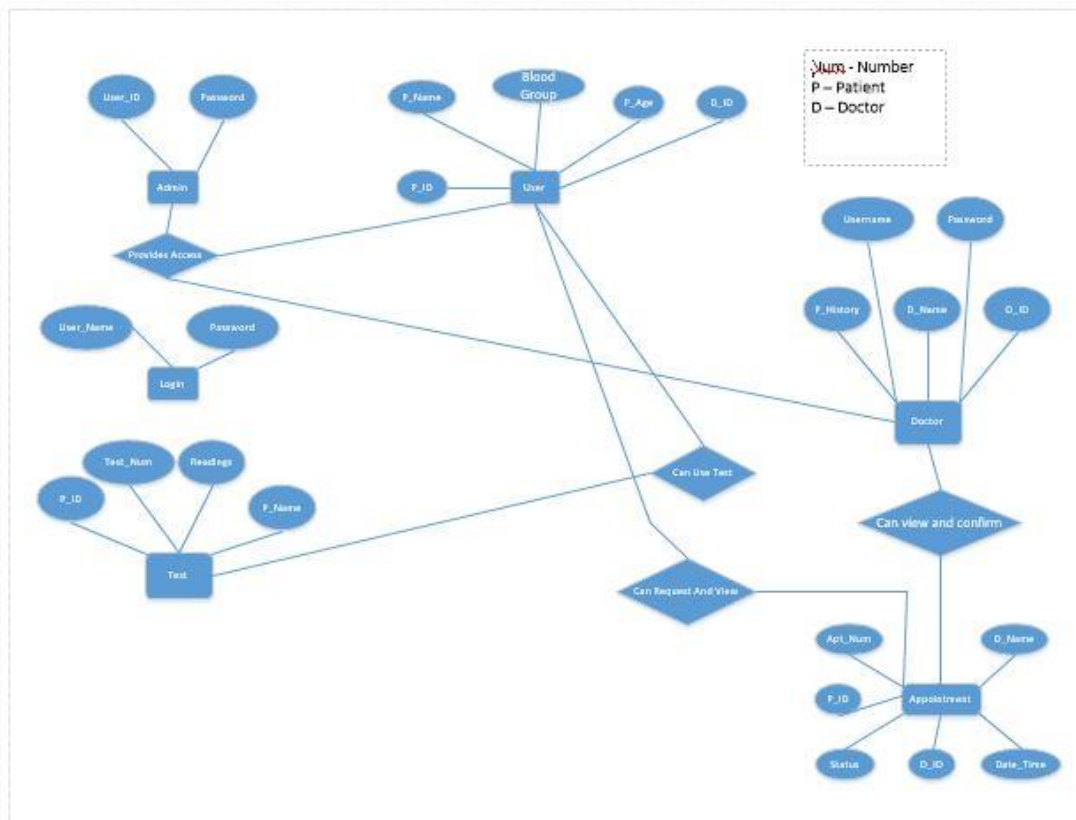


Figure 27:ERD of Project

## CHAPTER 6

### CONCLUSION AND RECOMMENDATIONS

#### 6.1 Conclusion

Diabetes is a tragic disease. It does not have a cure, but we can improve our quality of life by tracking our sugar level. Nowadays, Android applications has become an integral part of smart phones and Gluco-Care is a step towards it. Most of the people use insulin or medication but they lack in taking the precautions. Most of the diabetic patients are unaware of the diet and exercises recommended by the doctors specifically for diabetic patients. Many of them don't know the ideal conditions to start workout or the intake of diet which can maintain the sugar level throughout the day. Workout helps you control diabetes and prevent long-term complication. The more strenuous your workout, the longer your blood sugar will be affected. Low blood sugar is possible even four to eight hours affected after exercise. If your blood sugar level is less than 100 mg/dl take a carbohydrate snack prior to beginning the workout because lower than 100 mg/dl blood sugar level is too low to exercise safely.

Exercise makes it easier to control your Diabetes. If you have Type 1 diabetics avoid skipping more than 2 days of exercise in a row and get at least 150 minutes of physical activity per week. When you have Type 2 diabetes, physical activity is an important component of your treatment plan. With Type 1 diabetes, body stops making insulin. Diet and exercise play an important role in helping your blood sugar levels stable.

It's also important to have a healthy meal plan and maintain your blood glucose level instead of depending on medications and insulin, if necessary. Like many other exercises introduced in Gluco-care, Swimming is an ideal exercise for people with type 2 diabetes because it doesn't put pressure on your joints like walking and jogging.

Foods like raw, cooked or roasted vegetables, whole-grain, higher-fibre foods or protein can help to maintain your blood sugar level and make your life happy and healthy. If you do not have enough insulin available, your blood glucose levels can increase right after exercise. Check your blood sugar before and after exercise. Do not exercise if your blood sugar is over 250 mg/dl.

Gluko-Care aims at solving problems and doing much more for the patients of this disease by make them feel that they are not doing anything extra, so it remains all natural.

## **6.2 Recommendation**

### **6.2.1 PCB**

We will redesign the hardware by shifting it to PCB.

### **6.2.2 Wireless Connection**

The device that is connecting to the smartphone through OTG can be connected to the smartphone using Bluetooth. Using this method will increase the durability of the Arduino device.

### **6.2.3 Painless Testing**

Patient will use saliva to test blood glucose instead of pricking the finger for blood.