



# **SugarSync**

**Monika Kumar**

Submitted in partial fulfilment of the requirements  
for the degree of

**BSc in Business Computing**

Technological University Dublin (City Campus)

May, 2024

Supervisor: Audrey Jennings

## **Declaration**

This is an original work. All references and assistance are acknowledged.

Signed: Monika Kumar

Date : 26 / 04 / 2024

## Acknowledgements

I would like to give a huge thanks to my family, friends, lecturers and my supervisors for all the endless support. I would have not gotten to the finish line of my degree and my final year project without all their support and help.

I would like to begin and thank my mom and dad for always being there for me. It's been a huge help knowing there is always someone standing behind me cheering me on. My mum and dad are my biggest support and my whole degree is for them. I would also like to thank my brother, Suraj, for always listening to me and giving me advice even though he didn't understand anything I was studying. I appreciate all the small things. I would also like to thank my grandparents, for always asking me how my studies were going and motivating me.

Secondly, I would like to thank my lecturers I have had in the past four years and my supervisors. I have been grateful to learn so much from them from every module they have taught. All of this knowledge contributed to my final year project.

I want to give a huge thanks to my supervisor Audrey Jennings for all her continuous support during my final year project. She had given me all the help I required to complete my final year project and I am very grateful to have had her as my supervisor. I would also like to thank my second reader Jenny Munnely for her valuable suggestions and input. Both of my supervisors gave me the confidence to complete my project to the best of my degree.

Finally, I would like to thank my friends in class who have had a really positive impact on me and helped me with such a great college experience. They made my college time an unforgettable time. Specifically, I would like to thank Najla Alsafadi and Alicja Sokalska.

## **Abstract**

SugarSync is a mobile application aimed at patients with any form of diabetes. The main aim of the app is to be able to **save and monitor glucose levels, diet and exercise to keep their diabetes in control and in range**. The application **helps diabetes patients be able to predict whether their sugar levels are too high or too low** and to have their **records saved on their phones for easy access**.

This report will display all the functions of this app and how these features were designed to make it easier for diabetes patients. This includes all the technologies being used, design features and any challenges that occurred during the duration of creating this application.

GitHub Link: <https://github.com/MK250/Final-Year-Project.git>

# Table of Contents

<b>Chapter 1 - Introduction .....</b>	<b>7</b>
Introduction to Project.....	7
Objectives of Project .....	10
Business Case of Project .....	12
Overview of Technologies Used.....	13
<b>Chapter 2 – Requirements Capture and Analysis .....</b>	<b>16</b>
Business Requirements.....	16
Non – Functional Requirements .....	19
Requirements Analysis.....	21
Use Case .....	22
<b>Chapter 3 – Design .....</b>	<b>25</b>
System Architecture .....	25
Data Model Design .....	28
Database Object Summary .....	29
Software Design .....	32
User Interface Design .....	34
<b>Chapter 4 - Implementation .....</b>	<b>36</b>
<i>Issues and Resolution with Implementation .....</i>	<i>52</i>
<b>Chapter 5 – Test Plan.....</b>	<b>53</b>
Admin Tests .....	53
<b>Chapter 6 – Conclusion .....</b>	<b>58</b>
Final Conclusions.....	58
<b>References.....</b>	<b>59</b>

# Chapter 1 - Introduction

## Introduction to Project

My family has a history of diabetes and the one thing they always struggled with was keeping count of their sugar levels. They used scraps of paper to write it down. The finger prick devices never saved more than seven days' worth of data. There was never a proper way to monitor sugar levels and keep track of past history records.

Once I had begun my final year of college and had to think of an idea for my final year project, this diabetes problem came to my mind. Thus began the project for SugarSync.

I began research in my house and started testing out all the finger prick devices I had in my house. I tested each one to see how much data each device holds and how far back each data goes. I realized that all of these devices that are recommended by doctors are very basic and very few have proper apps connected to monitor sugar levels. With this research, I also realized, that any device that has an app, it was only connected via Bluetooth and the device had to be near the phone to be able to refresh with new data to get proper data analysis.

My main aim for my project was to make an easy-to-use app for diabetes patients to be able to manage and monitor their sugar levels. This application focuses on **sugar levels, diets and exercise of patients so it makes it easier for the user to monitor how high or low their sugar levels** are and how important it is to **control diet and exercise** and the impact it has on their glucose levels. This app also allows the patient to be able to **scan and capture their finger prick test results** which then saves it on the app and gives the patient a quicker and easy way to input their glucose levels without having the device near them to refresh data. This way the user always has their glucose levels with them. This is also an efficient way to show their doctors the user's past glucose level history.

The patient can register and login to the application. While the user is registering to the app, they have the chance to **save any diabetes-related information to the app** so that the user has all this data in one place. Once registered, the user is displayed with the home dashboard in which they can see the latest check on which date and time. In

the latest check, the user is displayed with the latest amount of sugar intake, glucose level and the amount of exercise. This gives the user a quick overview of all the important aspects of their health and what the user needs to focus on.

The user is presented with the **last seven days' trend with bar charts** for blood glucose levels, sugar intake and exercise. The user can also **check the last thirty days of their data along with the default seven days**. The user has the option to select what **target range** they are looking for in their blood glucose whether it is **doctor-recommended or personal** in which the user inputs their age, type of diabetes, weight and average glucose reading - this does a **small calculation** in which the app calculates what the target range should be for that user.

The user also has the opportunity to calculate the bolus level recommended for them by clicking on the 'Bolus Advisor' button. This directs them to another Screen in which the user can input the blood glucose level, the carbohydrate intake and the insulin sensitivity factor. All these factors **calculate the bolus unit recommended for the user**. Bolus is an insulin some diabetes patients take and this calculator advises them either before or after food is advised for the patients. The user is also directed to a quick YouTube video explaining what Bolus insulin is and how the calculation is done.

The user can navigate to other screens which include a glucose screen in which the user can capture the glucose level data. **Once the data is captured, the user is displayed with the current glucose level and simultaneously the user can see on the graph in the screen how high or low their glucose level is to the target normal range of 5.0**. The user also has an option to **manually add their glucose level** to the date and time and save the data. The glucose screen checks if the glucose level is higher than the target range then it checks the database and sees if the sugar level is too high and exercise is low, the app shows an alert dialogue warning the user and it does the same if the glucose level is too low. The user then has the option to view all their past glucose from the start.

Other functionalities of this app include **inputting diet (breakfast, lunch and dinner)**. The user can also get the **nutritional information of each food item** before submitting it. The nutritional information includes the total calories, protein, carbohydrates, fat and total sugar. Once the user inputs their diet meals, the user is automatically

**displayed with the amount of sugar they have intake with a progress bar** and the user can **track their daily sugar intake** against the total amount of sugar intake a diabetic patient can have (30g daily). The user can also **check all the past history meals** along with the date and time submitted,

The user can also **input their exercise details** (type of exercise and time taken for the exercise). Once submitted, the user is displayed with the **amount of time the user has exercised of the day in the progress bar** against the total amount of exercise a diabetic patient should be doing (30 mins daily). The user has the option to be able to **see all their past exercise data** along with the data submitted and their data. The user can also **view YouTube videos playlist which displays exercises diabetics patients can do at home**. The user can also **view their profile with all their details** which include their emails and the diabetic information the user registers with (type of diabetes, what sort of insulin / therapy they take, any pills the user takes, the units of blood sugar and their target range).

All these functionalities help diabetic patients monitor their sugar levels, diet and exercise to stay on top of their health and allow the patient to have all their data in one place in their phones.

## Objectives of Project

The objective of creating this project was to create an application which will aid diabetic patients in monitoring and analyzing glucose levels, diet and exercise. Patients who get diagnosed with diabetes don't properly realize the implications of this chronic disease on their lifestyle, and how the patients have to fully change their diet and start exercising. Both diet and exercise go hand in hand to keep the glucose levels in control. This app helps diabetic patients keep track of everything that is required to keep themselves healthy and helps them see what needs to be changed and how all these factors come in handy to keep diabetes in control.

The overall objective of this project is to have all of the relevant needs of a diabetic patient in one place, with the application being easy to use.

Users should be able to:

- Register and login to their account
- Logout of their account
- View the latest check of the app in the home dashboard
- View bar charts of data on glucose levels, sugar intake and exercise time
- Calculate bolus insulin levels
- Capture glucose level data
- Add glucose level manually (date, time and glucose level)
- View all past glucose levels
- Edit glucose levels
- Delete glucose levels
- View nutritional information for each meal
- Add diet meals (Breakfast, lunch, dinner)
- View all past diet meals
- Edit diet meals
- Add exercise data (Exercise Type and time taken for exercise)
- View all past exercise data
- Edit exercise data

- View the YouTube video playlist for at-home exercises
- View profile information
- Edit profile information
- Email themselves a CSV copy of the glucose data

The system should be able to authenticate users for registration and login to the app and allow users to access core functionality.

## Business Case of Project

In my opinion, this project will benefit all diabetic patients whether newly diagnosed or from birth. For diabetic patients, they need to realize how diet and exercise play together to keep glucose levels in control. Most diabetic patients don't know that keeping track of sugar levels is very important and should have all the old data in hand. A diabetic patient needs to change their lifestyle so they can keep their sugar levels in range. Some patients don't exercise as often or lose weight to keep themselves healthy. A healthy body is essential for a diabetic patient.

Traditional methods of tracking these parameters can be inconvenient and often lack the necessary insights for effective management. These methods are direct to the point with no proper explanation to the patient. A finger prick test device just displays the exact glucose level and whether it is in range. There is no mention of what the target range should be or what track the patient is at for that current testing.

Another challenge the patient faces is that the finger prick devices don't tell the patient the possible reason as to why the glucose level is too high or low. The patient doesn't know what to avoid or to do more of.

Therefore, this application will help diabetic patients how to stay organized to keep themselves healthy by monitoring their glucose levels and displaying the reasons why their glucose is up and down by their diet and exercise input. This app seeks to address these challenges mentioned above with a better and more intuitive platform to monitor and manage their condition more effectively.

### Business Rules

- The patient must register to access their account in the application
- To use the app, the patient must be logged in
- The patient needs to permit their phone access to the camera and notification to use camera features to scan and capture glucose levels and to receive any notifications from the application.
- To use the app, the patient can only have one account associated with their email

## Overview of Technologies Used

### Android Studio

#### Android Studio



I decided to use Android Studio for my project as this is the easiest way for the user to capture their data and have it with them all the time. I learnt how to use this application from Thoa Pham in Mobile Applications. This was such a great module as it gave me the starting point on how to use this app to create an Android phone application. As patients always have their phones, developing a phone application made more sense as the users can show all their data to the doctors anytime required.

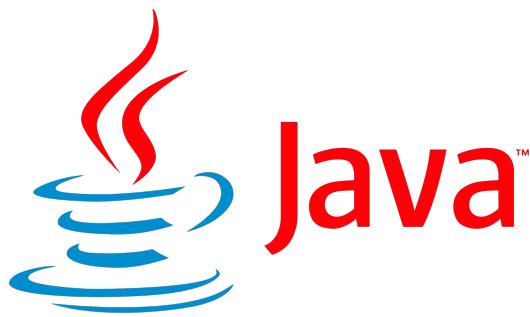
### Firebase



I went with Firebase as my database for the application as it is cloud-based and also has Realtime database. This suited my application as the users always need to input

data and it had to be saved in real time. I also used authentication to provide a secure way to implement user authentication and authorization. Using Firebase provided me with a strong backend functionality that can increase the development process. It enhanced the overall application performance and it was a convenient way to save data.

## Java



I chose Java as the programming language for my project as I had the most experience using that. I have spent the last four years learning to use this language. Java was also the first programming language I learnt. I felt confident enough to develop a whole stack application to the best ability. I felt like using Java would give me a greater benefit and I would be able to develop complex features better than using a language I didn't have much experience with. Using Java gave me another benefit if I was having any troubleshooting errors, I could always go to one of my lecturers and see where I was going wrong. Plus with all the experience I had using Java, I had a better idea of how to use it to my best ability.

## XML



```
<?xml version="1.0" encoding="UTF-8"?>
<foo>Hello World!</foo>
```

I used XML in various aspects of my project. I used XML in developing the UI (user interface) of my application. I also used it in the Android Manifest file. XML was used in the layout files as they defined the layout of my activities. XML came hand in hand with Java used in Android Studio. It made creating the front end of my application in a better way. I was able to create the UI on how it suited my application.

## GitHub



I used GitHub as a way to track my progress and version control throughout the duration of developing the application. It backed up all my code and it was a convenient place to have all my backup in one place so I always had something to go back. GitHub came in handy when my laptop had gotten broken and was gone for repair for 2 weeks. I was able to work on it in a different device as I had all my code.

# Chapter 2 – Requirements Capture and Analysis

## Business Requirements

Actors:

User (patient) - Only one type of actor is used in this application.

User:

When the user has registered on the application, this user is then stored in the database. When the user is registering, the user must provide their email and password. Along with login details, the user must also answer some diabetes-related questions upon registration (type of diabetes, what form of insulin the user takes, if the user takes any pills, units of blood they monitor and the target range of glucose level). After the user is successfully registered, the user is then able to login with the email and password they provided.

The user is directly navigated to the home dashboard once logged in. This is where the user is provided with all of their health-related details when the user once starts to input their data.

The user can see all their data in the form of bar charts with target lines for the user to analyze their data. In the home dashboard, the user can also check the bolus insulin advised. Once clicked, the user is brought to another Screen where the user inputs the average glucose level, carbohydrates and insulin sensitivity factor, with all these three factors, a recommended bolus unit is calculated for the user.

There is a bottom navigation provided in the home dashboard where the user can flick from and go to glucose, diet, exercise and profile Screens. The user can start scanning and inputting their glucose levels, add in their diet meals and add in any forms of exercise the user has done.

- **Glucose Screen:**

The user is presented with two buttons on the Screen. The ‘capture’ button opens a camera for which the **user can scan their finger prick test results** from the device. The screen will then display the user the glucose level results and the results will also be simultaneously shown on the graph below. The user also has the option to manually add their glucose results if they wish and those results will also be displayed on the graph too. Once the results are captured, **the app will look through the database and check the diet sugar intake and exercise time to tell the user why the results are too high or low through an alert dialogue**. On the top corner, the user will see a menu option button **to view all past results** if they wish. This will open another screen where all the past results are shown. They are displayed from oldest to latest order from the date submitted.

- **Diet Screen:**

The user **can input their diet meals (breakfast, lunch and dinner)**. The user can input all three meals at the same time if they wish but they also have the chance to submit them at different interval times throughout the day. The user is also simultaneously **displayed with the amount of sugar they have consumed** with a **progress bar (limit 30g daily)**. The user can also **check the nutritional information** of each meal or all three before they submit. This gives the user a chance to see which meals have more sugar and can change throughout. The diet screen also has a menu option in the top corner in which the user is directed to a separate screen which displays all the **past diet meals** and the user can see from oldest to latest from the date and time submitted.

- **Exercise Screen:**

The user can **input the exercise** they have done for the day ( type of exercise and time taken for the exercise). Once submitted, the user can instantly see the **amount of time they have taken on the progress bar** against the daily amount of time a diabetic patient should take (**30 mins daily**). The user can also view a YouTube video playlist in which they are at home exercises the user can wish to do if they can’t go outside. The user is presented with a menu option button in the top corner after which

clicking the user is brought to another screen. The user then can **view all the past exercises** they have undertaken. They are all displayed from oldest to latest from the date they have submitted.

- **Profile Screen:**

The user **can view all the information they have provided when registering**. They also have the **option to edit** any information and save the changes.

## Non – Functional Requirements

### Security

- To allow access to the application, the user must be authenticated when they are logging in to their account. This makes sure that the user's data is protected and only the account holder can use all the application's functionality.

### Theme

- When accessing different screens of the application, the user must be presented with the same theme. For user experience, the application should have a common colour, text style formatting and design. Any lists or dialogues should be presented in the same manner and design. This brings consistency to the application and enhances the user experience.

### Simplicity

- The screens of the application should be direct to the point to make it easier for the user to use. There should be no hidden buttons or any non-relevant information presented to the user when accessing the application. Any names used in the application must be simple and understandable. Users should be able to use any buttons or features easily on hand.

### Availability

- The user's device must be connected to the internet to gain their data and application's functionality.

## **Usability**

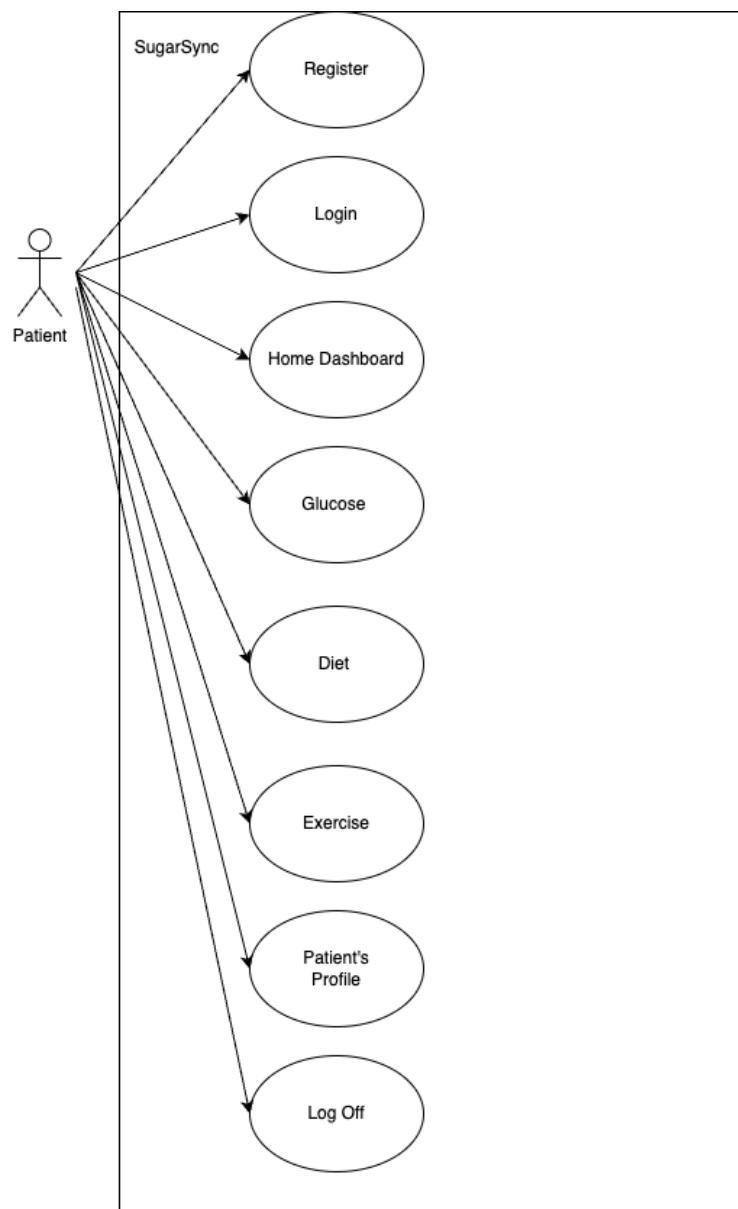
- The user interface should be easy to use for the user. They should be able to navigate from one screen to another with ease. The application should open for the user within seconds and any buttons being used for any sort should be doing their functionality within seconds too so as not to make the user wait for long.

## Requirements Analysis

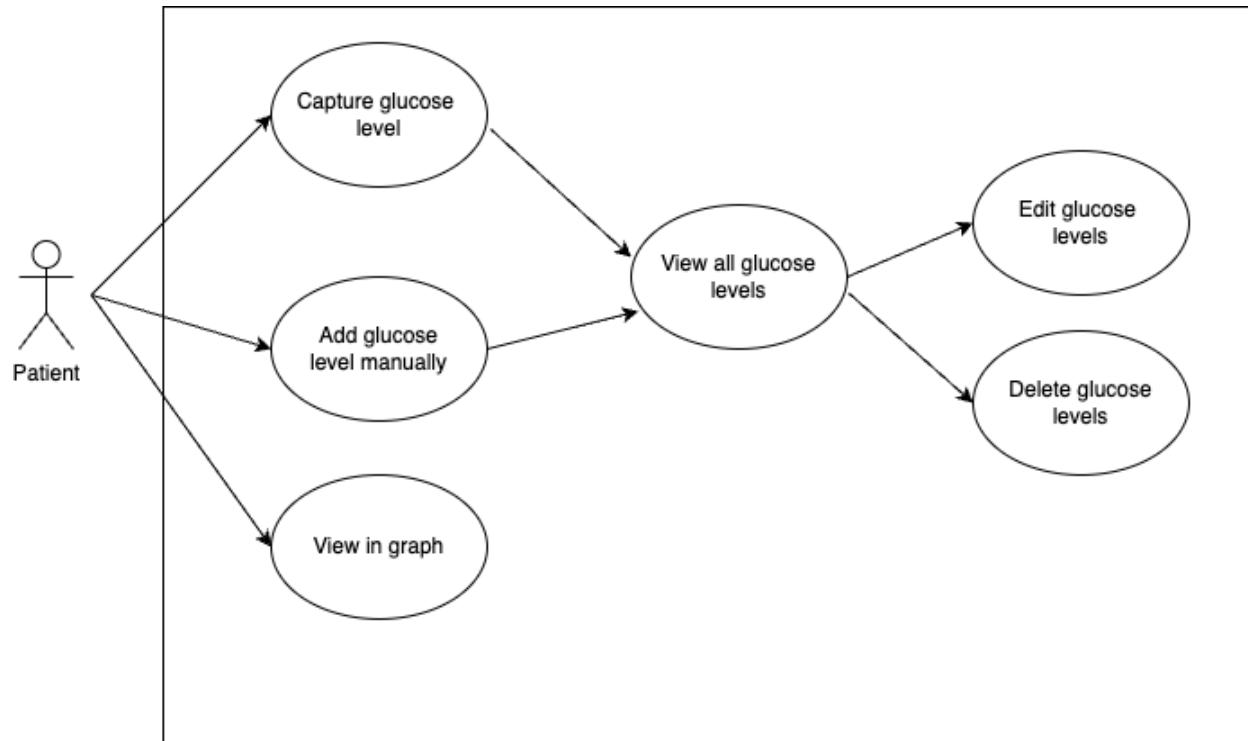
To research for this project, I reached out to my family and friends that have diabetes or know anyone that has diabetes. This helped me a lot for my analysis as I was able to find out what sort of an app they would like that would make it easier for them to monitor their diabetes. I was also able to test out some of the finger prick test devices that helped me how the device sorts the results. The knowledge of the monitoring of the diabetes results came in handy as I have family members at home that deal with diabetes. I was able to question and come up with some features that would prove to be useful for them. After the questioning, I was able to brainstorm some ideas and discuss with both family and friends and I was able to have a proper direction as to how I would like my app to look like and how the application's features can benefit the diabetic patients. It was also really helpful that my family member's tested the app step by step when I was developing the app so I could figure out which features worked better for them.

# Use Case

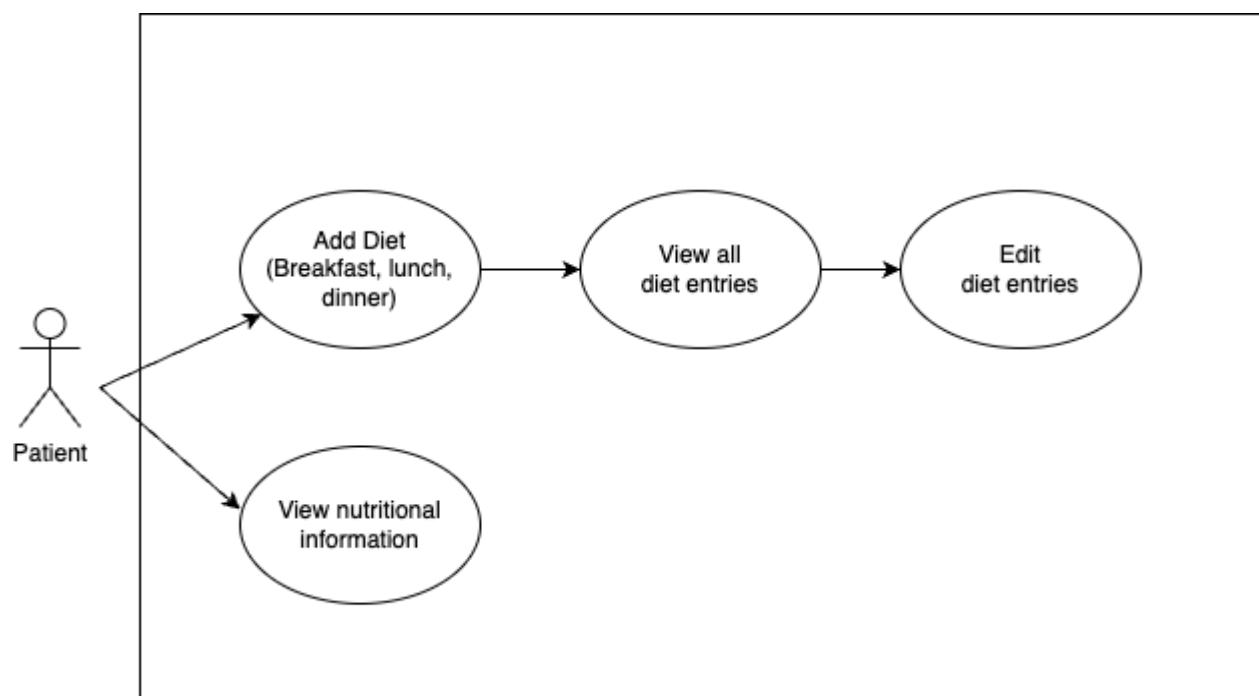
## High-Level Analysis



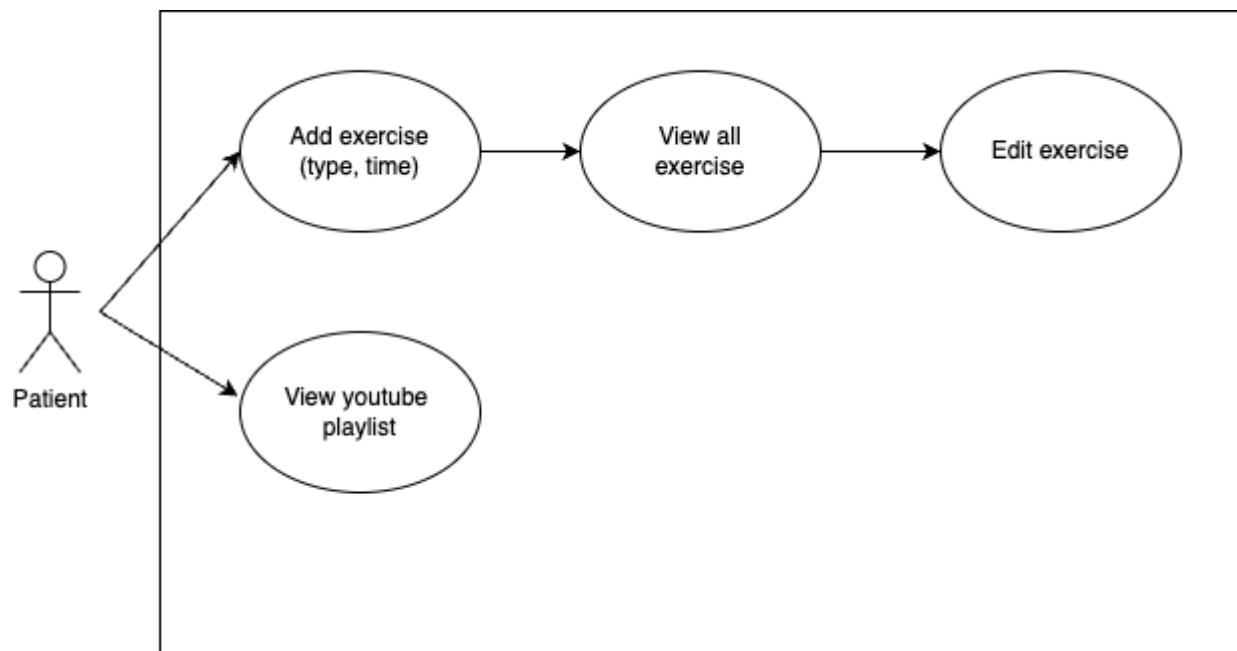
## Glucose



## Diet

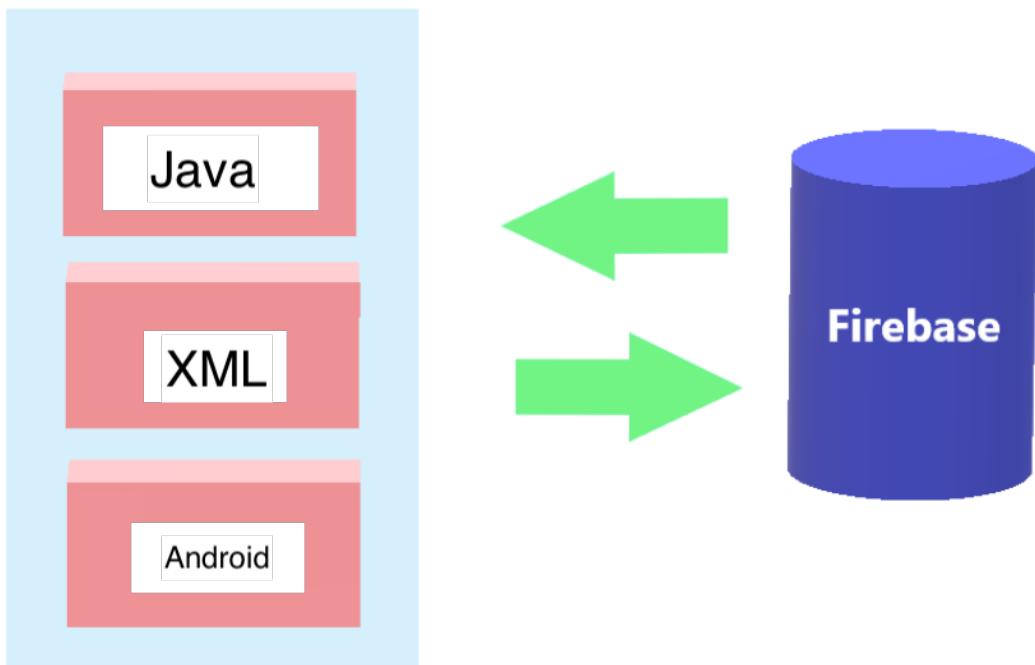


## Exercise



# Chapter 3 – Design

## System Architecture



SugarSync is an Android-based application using Firebase as a database. The system architecture follows as:

### **Presentation Layer:**

This is the outer layer which includes activities and fragments. The presentation layer is for handling the UI of the application. This is the layer of how the user views it.

### **Application Layer:**

All of the business logic is handled in this layer. Firebase backend comes in use and carries out user authentication. Data input occurs in this layer. The user inputs and scans their data and it's then added and captured.

**Data Layer:**

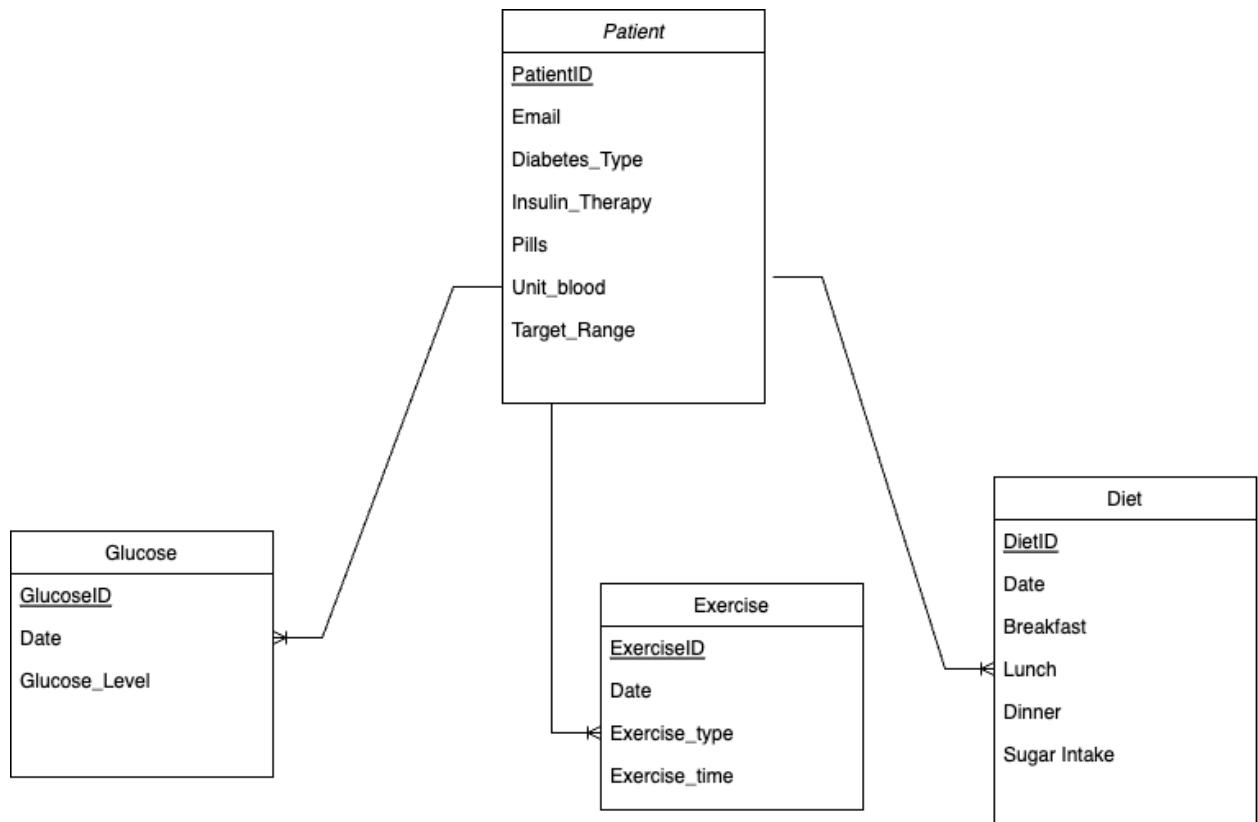
Real-time database is used in this layer. This layer handles all the data persistence of the application. All the necessary data needed by the Application layer is provided from this layer. All data inputted and captured are handled in this layer. All the APIs are also handled in this layer.

**Firebase Backend:**

Firebase backend is a cloud-based framework which includes Realtime database and Authentication for my project. This framework provides me with all the services and APIs communication required for my application.

## Data Model Design

As my database layer, I used Firebase, a mobile and web application development application that provides various tools and services to create an application easily and efficiently. My application used the Firebase Realtime Database and Firebase Authentication. These are two big important features carried out by Firebase.



## Database Object Summary

### Patient

<i>Patient</i>
<u>PatientID</u>
Email
Diabetes_Type
Insulin_Therapy
Pills
Unit_blood
Target_Range

The Patient table contains all the patient's personal information relating to themselves and their diabetic health. The table also includes the unique identifier (primary key) which is PatientID.

The email of the patient is entered when they are registering.

Other information the table includes is:

- Type of Diabetes the patient has.
- The type of insulin therapy the patient takes
- The pills the patient takes
- Patient's preference for the blood sugar level unit
- The blood sugar level target range

These details are saved once the user is successfully registered. The user has the chance to edit any details at any given time.

### Glucose

<i>Glucose</i>
<u>GlucoseID</u>
Date
Glucose_Level

The Glucose tables contain all the information relevant to capturing the glucose level of a patient.

The primary key is GlucoseID.

This table holds the following information about the patient's glucose:

- Date the patient captured the glucose level
- The actual glucose level figure

## Diet

Diet	
<u>DietID</u>	
Date	The Diet table holds all the data that the patient inputs every day which consists of all the diet meals throughout the day.
Breakfast	
Lunch	
Dinner	
Sugar Intake	The primary key is DietID.

The Diet table holds all the data that the patient inputs every day which consists of all the diet meals throughout the day.

The primary key is DietID.

The Diet table includes the following:

- The date the patient inputs the diet meals
- Patient's breakfast
- Patient's lunch
- Patient's dinner

## **Exercise**

Exercise
<u>ExerciseID</u>
Date
Exercise_type
Exercise_time

The Exercise table contains all the details that the patient inputs throughout the day when they exercise.

The primary key is ExerciseID.

The Exercise table holds the following information:

- The date the patient inputs exercise details
- The type of exercise the patient does
- The time taken for the exercise taken

## Software Design

SugarSync's software design consists of the Model View Controller (MVC architecture).

The application backend is Firebase SDK.

Software Design High-Level Overview:

### **Activities:**

- I used many activities in my application. These activities acted as the different screens in the application. These were used for the login, registration and the view of all glucose, diet and exercise and act as a communicator between the other screens in the app.

### **Fragment:**

- My application holds five fragments which are home fragment, glucose fragment, diet fragment and profile fragment. These fragments are reusable components that are used to represent a segment of the user interface. Fragments help with the navigation from one screen to another.

### **Adapter:**

- Adapter classes were used in my application to hold data from the model classes into the views of the application. They were used as a way to create views for model item classes and to add data from the matching model class. They acted as a way to respond to the model class.

**Model Class:**

- In my application, model classes were used to define data objects such as Glucose, Diet, NutritionResponse and Exercise. The behaviour and structure of these objects were defined in these classes and the data retrieved from Firebase were also represented.

The MVC architecture assisted me during the development of this application as I was able to maintain and manage the code easily. Each component (model, view and controller) had a separate responsibility and it didn't interfere with each other.

## User Interface Design

While working on the design of the user interface, certain components needed more priority such as the ease of use for the user and the application's consistency.

User Interface Design:

### **Layout:**

- The application was to be clear, concise and organised in such a manner that the user wouldn't have any difficulties using it. I focused on the simplicity of the application and making sure that the user won't have any problems. I made sure of that by having a common theme throughout the app so that all the buttons and components had the same colour and font.

### **Navigation**

- For ease of use, I included a bottom navigation in my app so that there was a quick way of going from one screen to another for the user. By using bottom navigation the user can access the home dashboard, glucose fragment, diet fragment, exercise fragment and profile fragment. In the glucose, diet and exercise fragments, I also included menu option to view all the past entries. I wanted to make the navigation for the user as easy and efficient as possible. Including all these components, users should have no problem navigating from the one screen to another.

**Libraries:**

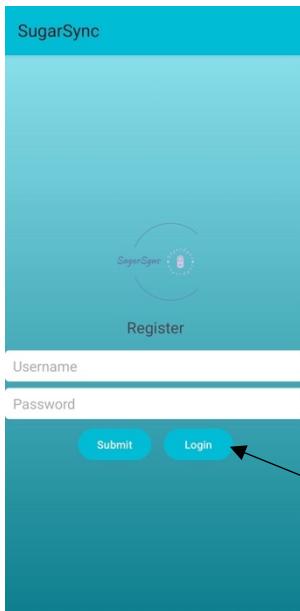
- I took the help of some libraries to enhance the user experience while using the application. I used MP Android Chart library for the user to be able to view the bar charts for all their data. I also used the YouTube library for the user to be able to view the YouTube playlist be it to for the at home - exercise or understanding what a Bolus insulin is.

**Android Controls:**

- My application includes multiple text field boxes and buttons. All of these components are well defined in the app in a clear and controlled way with everything labelled to make it easier and understandable for the user.

## Chapter 4 - Implementation

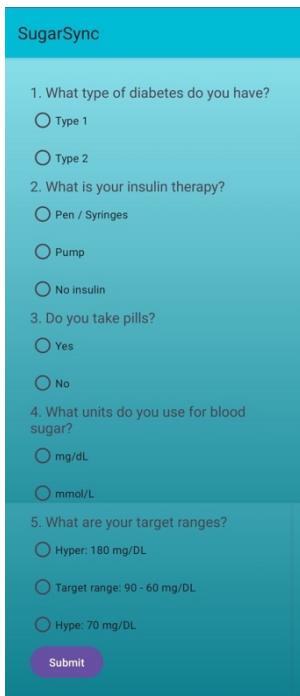
- **Main Activity Screen ( Registration Screen)**



On the main activity Screen, the user has the option to either register to a new account or login to their account by using the 'Submit' and 'Login' buttons.

Component: com.example.SugarSync.MainActivity

- **Registration Questions Screen**



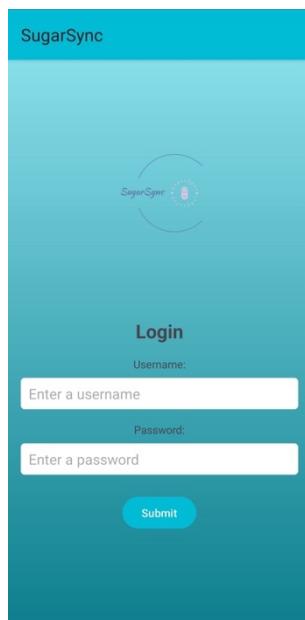
When the user is registering, they are directed to another Screen of questions related to their diabetic health once clicking Submit.

The user can this way save their important personal details regarding their health.

Once all these questions are answered, the user is then successfully registered and brought to the home dashboard.

Component: com.example.SugarSync.Question\_Activity

- **Login Screen**



Once the user is successfully registered, the user can login with their email and password at any time. All of this data is authenticated with Firebase so their account is protected and private. Only the account holder knowing the email and password can login.

Component: com.example.SugarSync.LoginActivity

- **Home Dashboard**

The home dashboard is a personalized health dashboard for the user. At one glance the user can view all the details of their health. On the home dashboard, user will find the bar charts of data with the target lines and can view the bottom navigation bar to the other screens of the application. The home dashboard has nested scroll view so the user can scroll and check all the features with ease.

Component: com.example.SugarSync.Fragment.HomeFragment



## SugarSync

Blood Glucose Level (mg/dL)

Enter your blood glucose level

Carbohydrate Intake (g)

Enter carbohydrate intake

Insulin Sensitivity Factor (ISF)

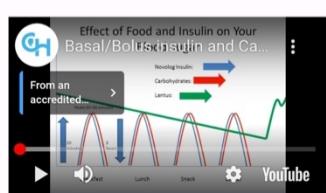
Enter your ISF value

**Calculate Bolus**

## The Bolus Advisor Screen

This Screen calculates the recommended units a diabetic patient can take. The user inputs the average blood glucose level, carbohydrate intake and Insulin sensitivity factor. Afterwards, this calculates the recommended unit.

Bolus is an insulin some diabetic patients take either before or after food.

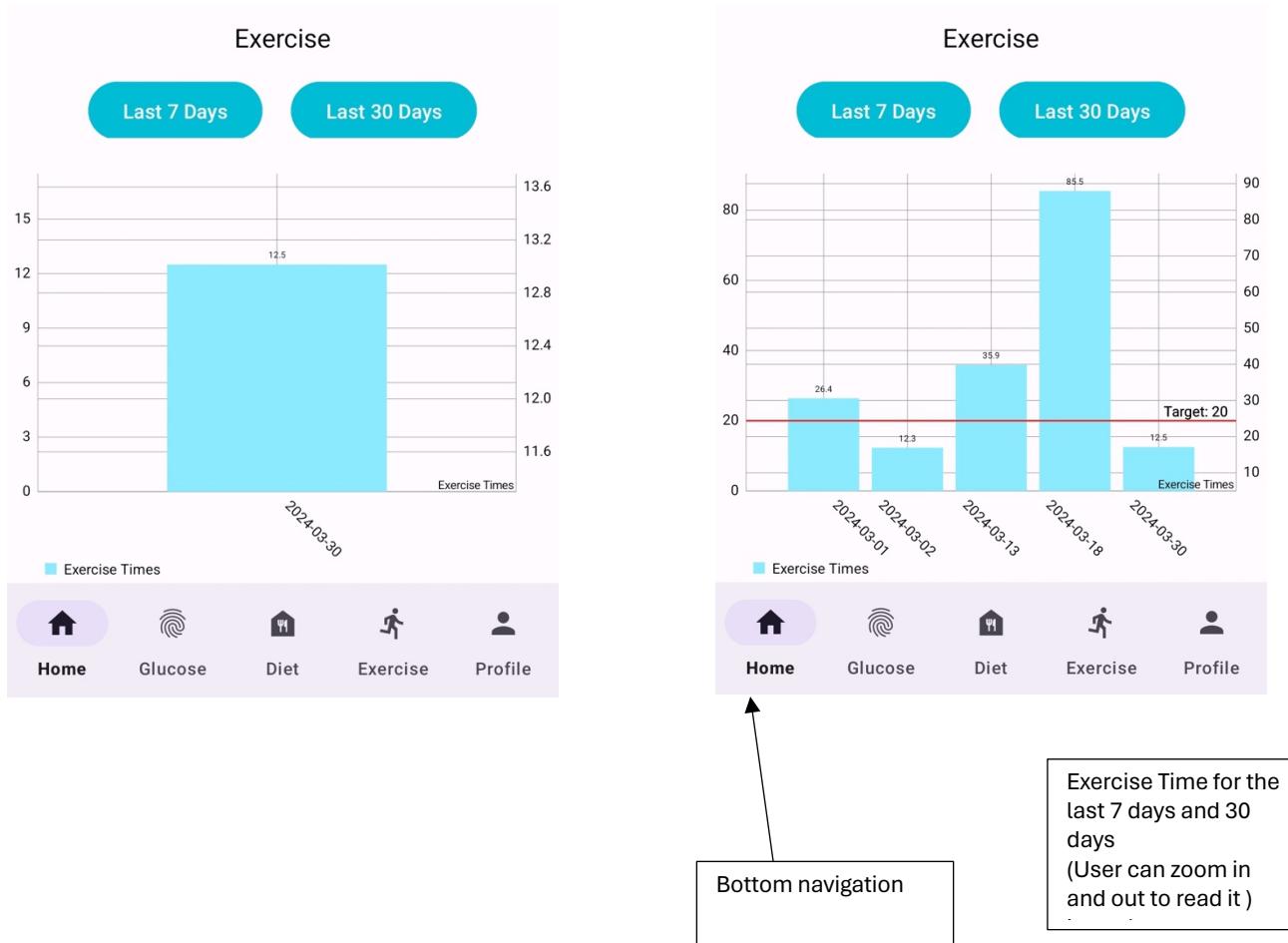


YouTube video explaining Bolus Insulin and calculation.



Sugar Intake for the last 7 days and 30 days  
(User can zoom in and out to read it)

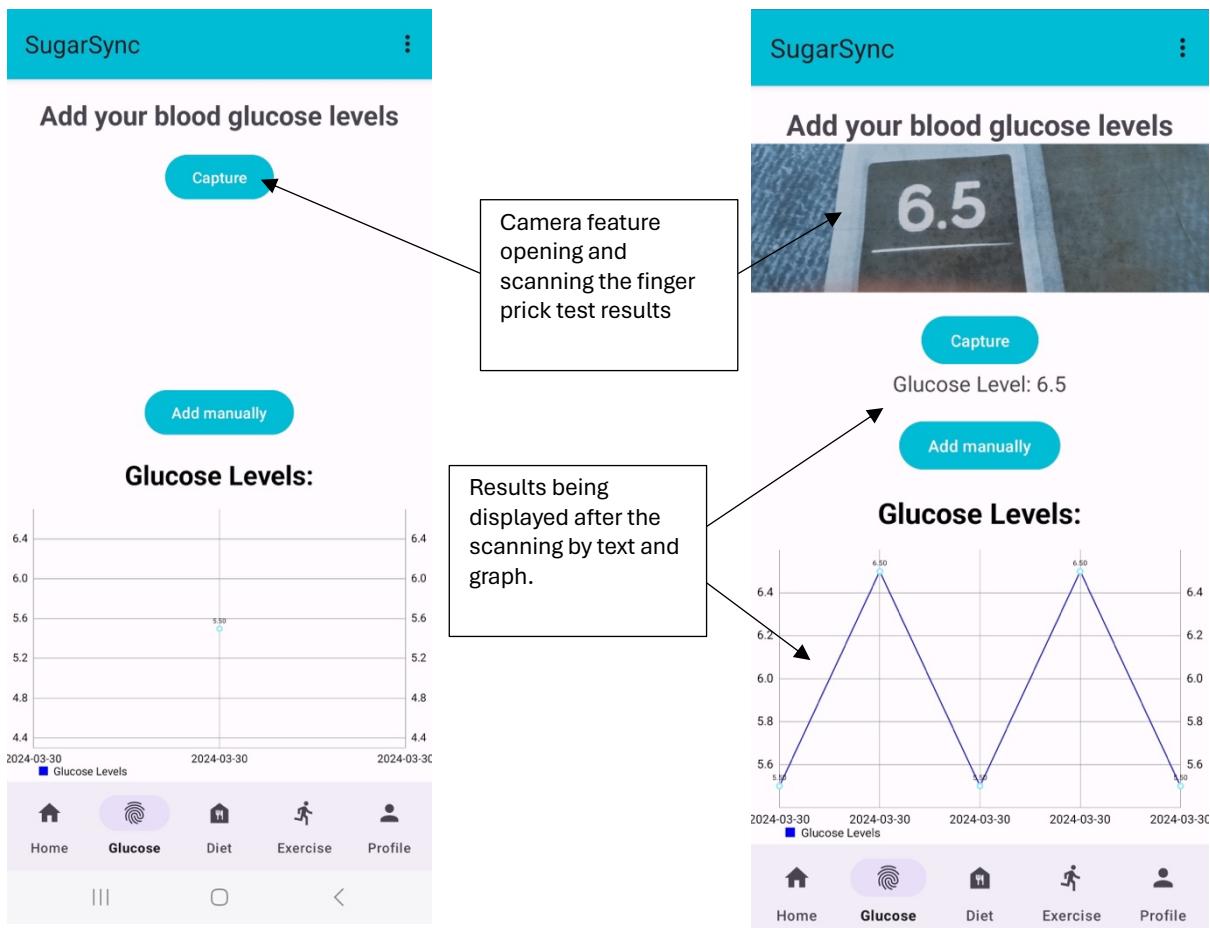




Component: com.example.SugarSync.Fragment.HomeFragment  
Bottom Navigation: bottom\_navigation.xml

- **Glucose Screen**

The glucose screen is directed when the user clicks the Glucose button in the bottom navigation. The user can scan and capture the glucose level when clicking the capture button. The user can also add their glucose levels using the add manually button too.

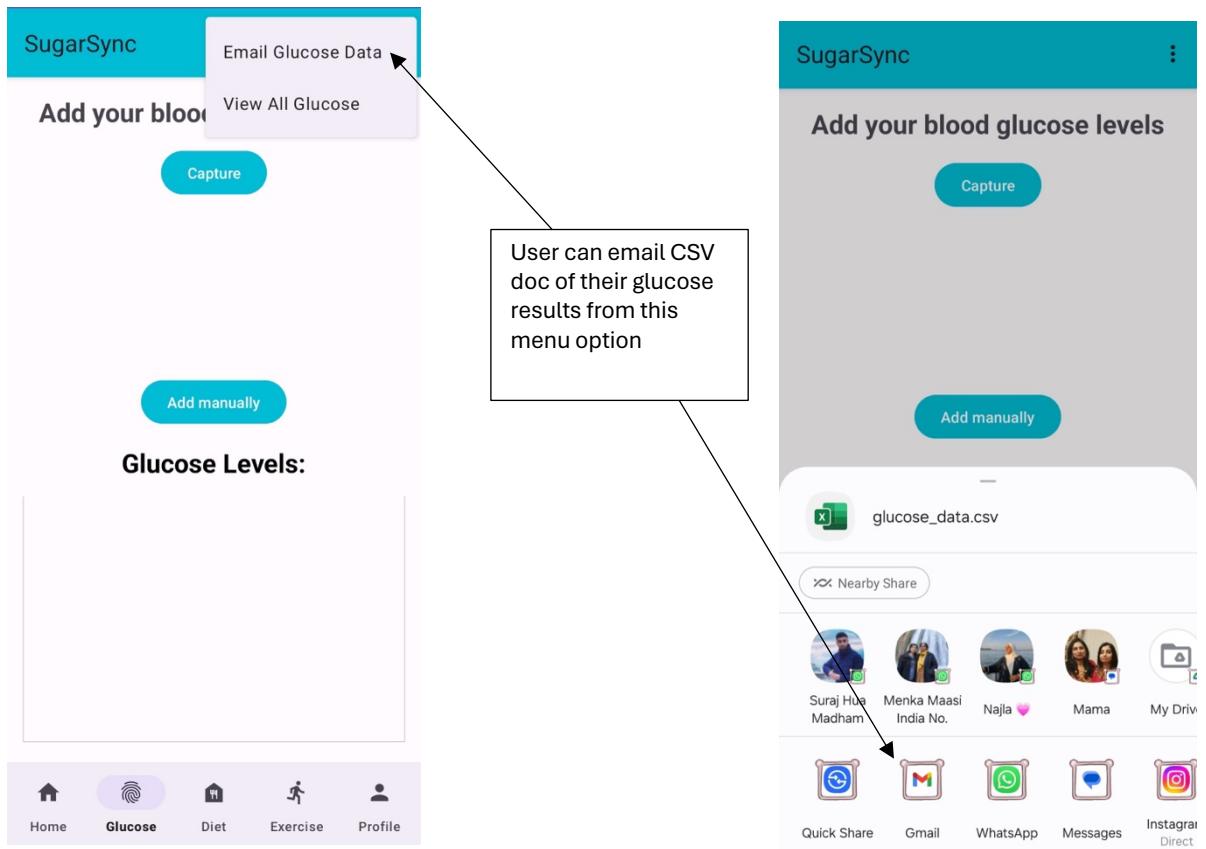


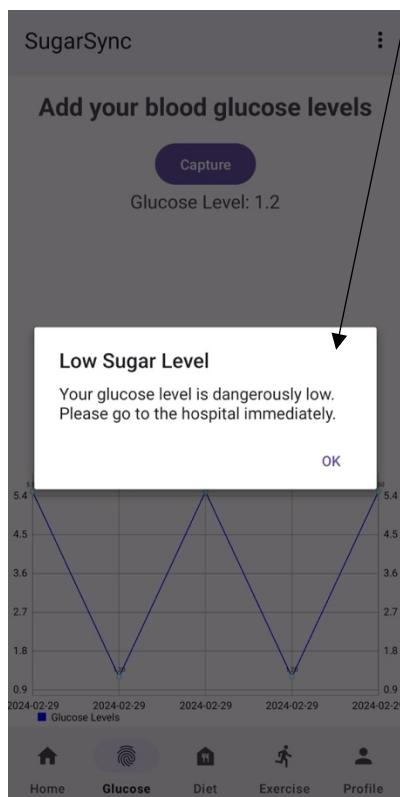
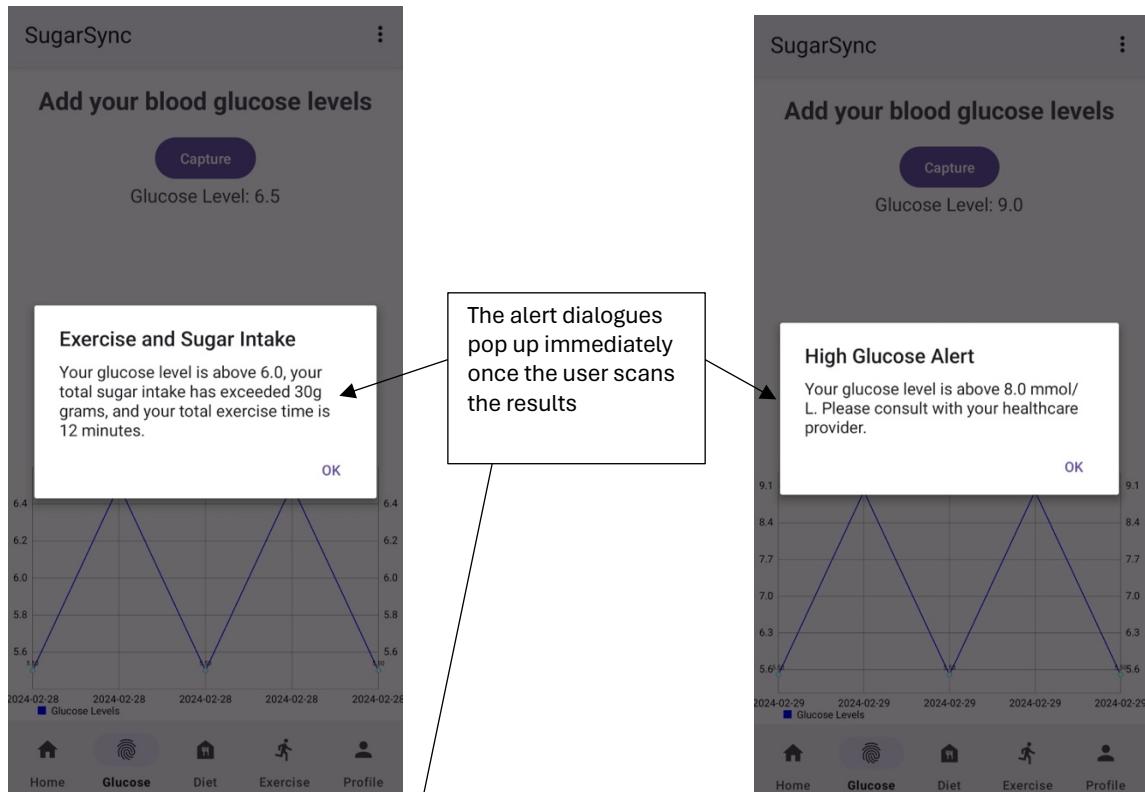
The user can add their glucose levels manually too. There is a date picker calendar and the user adds in the time and the glucose level figure. After clicking add. The glucose levels are saved and the user can view them.

Component:

com.example.SugarSync.Fragment.GlucoseFragment







These Alert dialogues pop up when the user is scanning their results. If the results are too high or low than the recommended amount, the application will look into the database and check sugar intake and exercise time and give you reasons as to why is it too high or low and what your next steps should be.

- View All Glucose Screen

SugarSync

View All Glucose

Add your blood glucose levels

Capture

Add manually

Glucose Levels:

Glucose Levels

Date	Time	Glucose Level
01/01/1970	00:59:59	6.0
02/21/2024	19:12:11	6.5
02/22/2024	10:47:44	6.5
02/22/2024	10:47:44	6.5
02/22/2024	11:15:25	6.5
02/22/2024	11:15:25	6.5
02/22/2024	11:26:55	6.5

Home   Glucose   Diet   Exercise   Profile

A menu option button to view all past glucose entries

List of all glucose level entries from oldest to latest.

SugarSync

Edit Glucose Level

9.5

Save Changes

User can edit the glucose levels results and save them

- Diet Screen

SugarSync

Dietary Needs

Breakfast  
100g of cornflakes

Lunch  
Enter lunch

Dinner  
Enter dinner

Submit

Get Nutritional Info

2.67/30g

The user can immediately see their sugar intake from the progress bar.

In the diet screen, the user can input their meals throughout the day. The users have to be very specific with all their meals which includes adding the amount of grams of food eaten and what sort of food. This way the correct sugar intake is saved for the user.

Component:  
`com.example.SugarSync.Fragment.DietFragment`

SugarSync

Dietary Needs

Breakfast  
100g of cornflakes

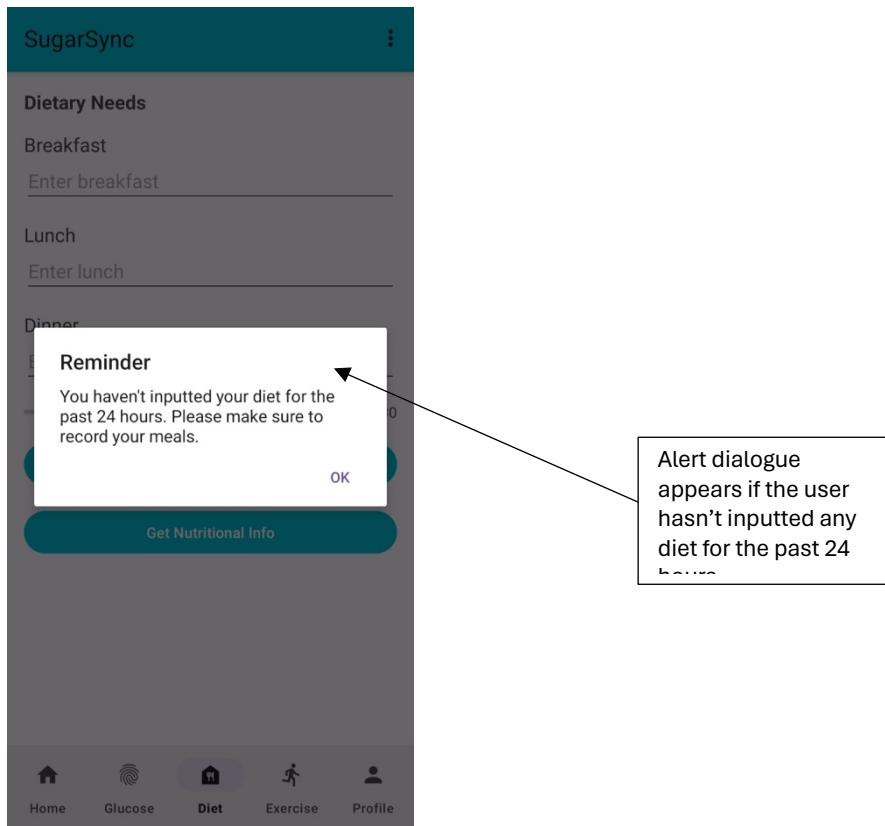
Lunch  
Enter lunch

Nutritional Information

Calories: 357.0 kcal  
Protein: Protein: 7.5 g  
Carbs: Carbohydrates (net): 80.8 g  
Fat: Total lipid (fat): 0.4 g  
Sugar: Sugars, total including NLEA: 9.5 g

OK

The user can check the nutritional info of a meal before submitting.



- View all diet screen

SugarSync

**Dietary Needs**

Breakfast  
Enter breakfast

Lunch  
Enter lunch

Dinner  
Enter dinner

Submit

Get Nutritional Info

Home Glucose Diet Exercise Profile

View All Diets

Menu options button to view all the past diet meals.

- 
- SugarSync
- Diet**
- Date: 2024-03-26 12:23:19  
Lunch: 200g of pasta
- Date: 2024-03-26 12:23:39  
Dinner: 200g of cornflakes
- Date: 2024-03-26 12:24:01  
Breakfast: 100g of cornflakes  
Lunch: 100g of cornflakes  
Dinner: 100g of cornflakes
- Date: 2024-03-26 12:24:57  
Dinner: 200g of pasta
- Date: 2024-03-27 19:37:43  
Lunch: 100g of pasta
- Date: 2024-03-30 10:51:24  
Dinner: 100g of pasta
- List of all diet meals from oldest to latest

SugarSync

**Diet**

Date: 2024-03-26 12:16:38  
Breakfast: 100g of cornflakes

Lunch: 100g of cornflakes

Dinner: 100g of cornflakes

Edit dinner  
200g of cornflakes

Cancel Save

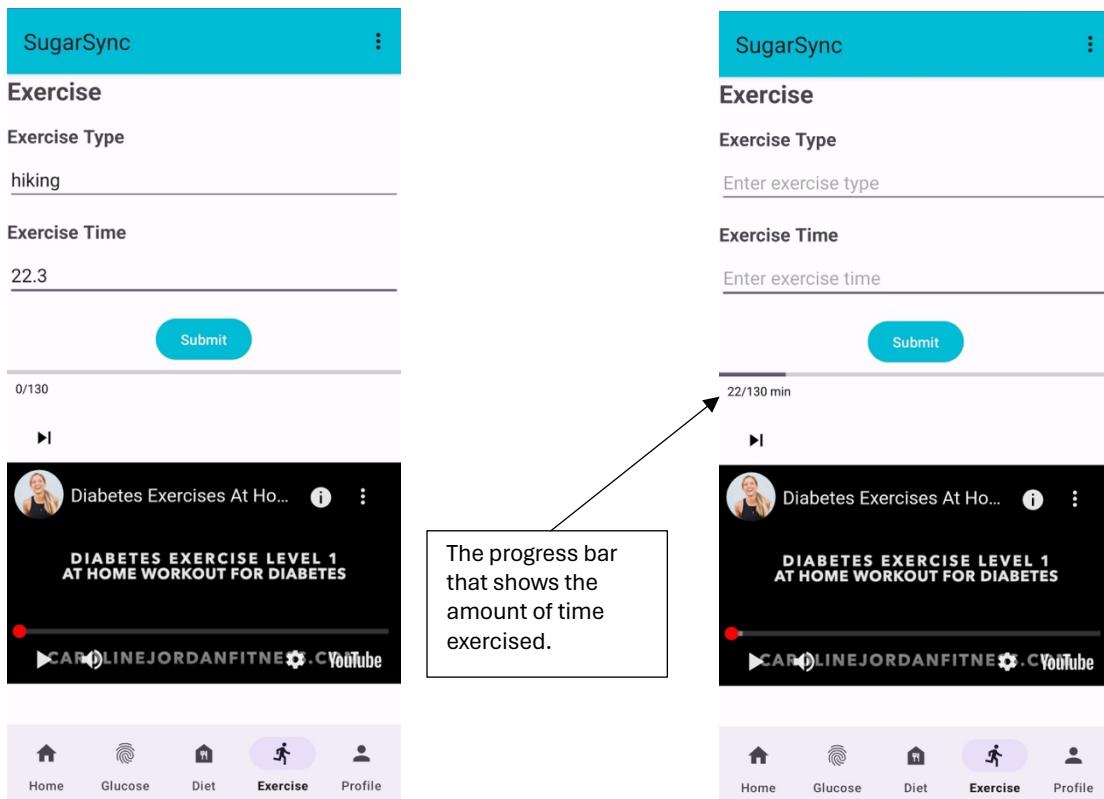
Date: 2024-03-26 12:20:05  
Dinner: 200g of cornflakes

Date: 2024-03-26 12:20:18  
Dinner: 100g of pasta

Date: 2024-03-30 10:51:00  
Lunch: 100g of pasta

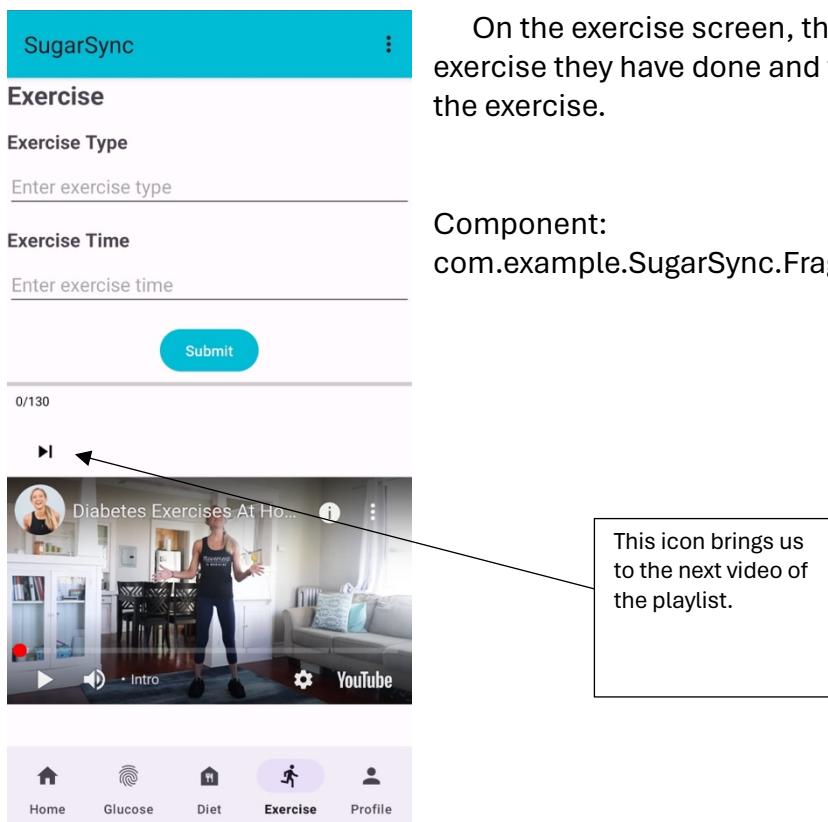
The user has the option to edit the diet meals and save them.

- **Exercise Screen**



On the exercise screen, the user can input the type of exercise they have done and the time taken to complete the exercise.

Component:  
com.example.SugarSync.Fragment.ExerciseFragment



- View all Exercise Screen

**SugarSync**

**Exercise**

**Exercise Type**  
Enter exercise type

**Exercise Time**  
Enter exercise time

**Submit**

0/130

▶

Diabetes Exercises At Ho... i :

DIABETES EXERCISE LEVEL 1  
AT HOME WORKOUT FOR DIABETES

CARDLINEJORDANFITNESS.C YouTube

**Home** **Glucose** **Diet** **Exercise** **Profile**

**SugarSync**

**Exercise List**

Date	Exercise Type	Exercise Time
2024-03-01	hiking	10.2
2024-03-06	walking	12.5
2024-03-20	hiking	23.5
2024-03-30	hiking	22.3

**SugarSync**

**Exercise List**

Date	Exercise Type	Exercise Time
2024-03-13	hiking	12.3
2024-03-18	hiking mountains	12.4
2024-03-30	walking	12.5

**Edit Exercise Type**  
hiking mountains

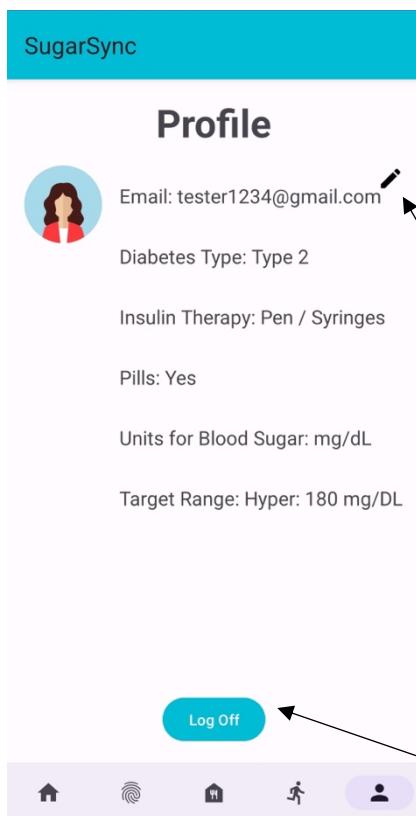
**Edit Exercise Time**  
12.4

**SugarSync**

**Exercise List**

Date	Exercise Type	Exercise Time
2024-03-13	hiking	12.3
2024-03-18	hiking mountains	12.4
2024-03-30	walking	12.5

- **Profile Screen**

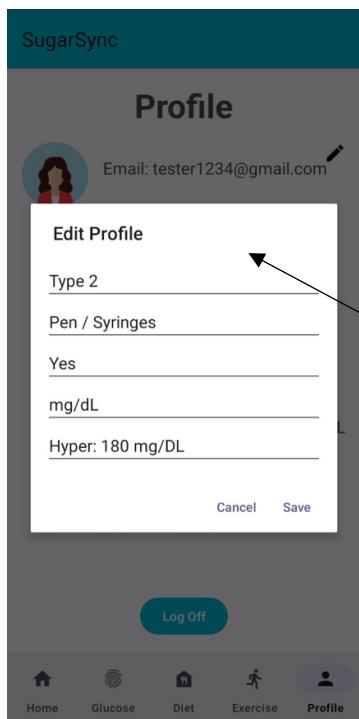


On the profile screen, the user will see all the information they have registered with. Along with the information, the user can log off from their account.

**Component:**  
`com.example.SugarSync.Fragment.ProfileFragment`

Edit icon to edit any details and save them

Log off button which brings the user to the login Screen



Users can edit any information and save them.

## Issues and Resolution with Implementation

- The first issue I had at the start of the project was trying to understand how Firebase worked as I had not worked with that before. The resolution came up with was to watch YouTube tutorials and research how it worked and how it was used.
- The next error I had with the implementation was the Android Studio application was too slow to run. Everything I did on the application, it ran too slow. The typing and the deployment took too long. I then researched and found out that other people were having the same issue as me. I had to follow a small tutorial and install the application again to run at a higher speed.
- When I was trying to deploy my application on the Android Studio emulator, it took too long to run and sometimes it would fail to launch. I then decided to use my own Android phone and connect it to my laptop to run. The deployment then started to launch quicker than the emulator.
- One of the big issues I had with the implementation was that I didn't realise that using the latest build Gradle for Android Studio would make it difficult to implement some API and libraries. I struggled a lot to make the MP Android Chart API run in my application. I had to follow many tutorials to get the API to implement. I had to keep testing to make it work.
- I struggled with understanding how the Firebase Real-Time database worked. I tried saving data but I could not get it to work. I had to research on the internet and look at some of the examples of how it worked and ask the lecturer in college to see where I was going wrong.
- I had trouble with figuring out why my graph wouldn't display all the data from the real time firebase database. That took me so long as the data would always have some problems. The solution was to make sure that my code pathway to the firebase should be correct and not to miss any node in the database so that the graphs would work.

## Chapter 5 – Test Plan

### Admin Tests

Test Name	Test Scenario	Result Expected	Actual Result	Test Result
Intro to application	The user should be able to view the register Screen where they have the option to login or register.	The introduction screen is displayed to the user with the options	As expected	Pass
Registration	The user should be able to with a working email and password with the conditions.	The user is registered and saved in Firebase database	As expected	Pass
Registration Questions	The user should be directed to the diabetic related questions after submitting email and password.	User is brought to the register questions and the information is saved to database	As expected	Pass
Login	The user should be able to login to their registered account using the email and password.	User is logged in their registered account authenticated by Firebase.	As expected	Pass
Logout	The user should be able to log out from their account	User is logged out and brought to the login Screen	As expected	Pass
Home Dashboard	The user should be able to view all the home dashboard features including the latest check and bar charts.	The user is able to view all the home dashboard features	As expected	Pass
Bolus Advisor	The user should be able to input their details necessary to calculate the bolus insulin.	The user can see the recommended bolus insulin recommended after inputting the details	As expected	Pass

Test Name	Test Scenario	Results Expected	Actual Results	Test Result
Setting Target Range	The user can set a target range for the blood sugar bar chart. It can be doctor recommended or personal.	The user is able to set a target range and its saved in firebase database. The target line is visible on the bar chart.	As expected	Pass
Blood glucose historical data	The user should be able to view the last seven days and last thirty days of the past glucose data.	The user can view all the past historical data on the bar charts by clicking the buttons.	As expected	Pass
Sugar Intake historical data	The user should be able to view the last seven days and last thirty data of the past sugar intake data	The user is able to view all the past data on the bar chart by clicking the buttons	As expected	Pass
Exercise Time historical data	The user should be able to view the last seven days and last thirty days of data of the exercise time	The user can see all the past historical data on the bar charts by clicking on the data	As expected	Pass
Blood glucose Scan	The user should be able to open the camera feature and scan the finger prick test results	The user is able to view the glucose result from the scan.	As expected	Pass
Add manually blood glucose	The ability to add the glucose level data manually.	The user is able to add the glucose data by adding in the date, time and glucose level figure. Data is saved in firebase	As expected	Pass
View glucose on graph	The ability to view the scan and manually added results on the graph.	The user is able to see the glucose level figure on the graph immediately after the submission.	As expected	Pass

Test Name	Test Scenario	Result Expected	Actual Result	Test Result
View past glucose data	The user should be able to view all the past data from oldest to latest in a list	The user can see all their past glucose data from oldest to latest in a list	As expected	Pass
Edit glucose data	The user should have the ability to edit the glucose data	The user is able to edit the glucose data and changed edits are saved in firebase	As expected	Pass
Delete glucose data	The ability to delete glucose data	The user is able to delete the glucose data. The data is also deleted from the list and firebase	As expected	Pass
Too high glucose alert	The user should be able to view an alert dialogue when the results are too high	The user can able to view the alert dialogue immediately after scanning the results. The alert dialogues checks the sugar intake and exercise.	As expected	Pass
Too low glucose alert	The user should be able to view an alert dialogue when the glucose results are too low.	The user can view the alert dialogue immediately after scanning the results. The alert dialogues checks the sugar intake and exercise.	As expected	Pass
In the middle glucose alert	The user should be able to view an alert dialogue when the glucose results are low but the sugar intake is too high and exercise is too low.	The user can view the alert dialogue immediately after scanning the results. The alert dialogues checks the sugar intake and exercise.	As expected	Pass
Add diet meals	The ability to add diet meals throughout the day .	The user is able to add all the diet meals. The diet meals are saved in firebase.	As expected	Pass

Test Name	Test Scenario	Expected Results	Actual Result	Test Result
Sugar intake progress bar	The ability to view the immediate sugar intake after adding diet meals	The user can see the immediate change in the sugar intake progress bar indicating how much sugar the user has intake.	As expected	Pass
View nutritional information	The ability to view the nutritional information of a meal before inputting	The user can see the nutritional information of the meal as a sugar intake checker	As expected	Pass
View past diet meals	The ability to view all the past diet meals in a list	The user is able to view all the past diet history in a list from oldest to latest	As expected	Pass
Edit diet meals	The user should have the ability to edit diet meals	The user can edit their diet meals. The saved changes are saved in firebase and displayed in the list	As expected	Pass
Add exercise	The user should have the ability to add the exercise they have done for the day	The user is able to add the exercise for the day. The exercise entry is saved to firebase	As expected	Pass
Exercise time progress bar	The user should be able to see the immediate progress bar update	The user can see the immediate progress bar update of the exercise time the user has done	As expected	Pass
Exercise at home YouTube playlist	The user should be able to view the YouTube playlist and go next to the video	The user can see the YouTube playlist and is able to move next to the next video	As expected	Pass
View past exercise history	The user should have the ability to view the past exercise data	The user can see the past exercise history details from oldest to latest	As expected	Pass

Test Name	Test Scenario	Expected Result	Actual Result	Test Result
Edit exercise	The user should be able to edit the exercise details	The user can edit the exercise details and the changes are saved to firebase	As expected	Pass
Profile Dashboard	The user should be able to view the profile features	The user can see the profile dashboard features	As expected	Pass
Glucose Dashboard	The user should be able to view glucose dashboard features	The user can see the glucose dashboard features	As expected	Pass
Diet Dashboard	The user should be able to view the diet dashboard features	The user can see the diet dashboard features	As expected	Pass
Exercise Dashboard	The user should be able to view the exercise dashboard features	The user can see exercise dashboard features	As expected	Pass
Edit profile information	The user should be able to edit the profile information	The user can edit the profile information. The changes are saved to the firebase.	As expected	Pass
Diet Alert Dialogue	The user should be able to view an alert dialogue if they did not input any diet meals for 24 hours	The alert dialogue popped up when the user didn't input any diet data for 24 hours.	As expected	Pass
Glucose Email Menu Option	The user should be able to email csv format of glucose data	The user was able to successfully email csv format of their glucose data.	As expected	Pass

## Chapter 6 – Conclusion

### Final Conclusions

Throughout my developing my application, I have learnt so much. I feel like at the start I was very doubtful as to how I was going to create an application from scratch and how I was going to get my idea from paper to actual reality but I have surprised myself. I put in so much effort and it paid off in the end.

My coding skills have increased after the completion of this project. My outlook on coding has changed, I didn't quite enjoy it at the start but the satisfaction of completing a project and seeing it in real life has changed my mindset. This project has enhanced so many of my skills like project management as I had to work on this application consistently and manage my other assignments alongside. This project has taught me how to debug coding projects and what to do when something is not working. I have learnt time management skills and how I should be dividing work and giving time to complete it.

In the project, I was faced with many challenges. This project taught me how to deal with them. I am not the best at coding so researching and watching YouTube tutorials helped me a lot. There were many cases during my project, that my project would have so many errors and it would not run. Learning how to debug and find the errors helped a lot.

The experience of developing this project ranged from excitement and eagerness to frustration and satisfaction. It was a wild range of emotions throughout. I am very grateful for all the opportunities that came my way during the development of this project. By the time of completion of this project, I would say this has been such a learning and rewarding experience for me. I am glad I stayed through and completed this project.

In conclusion, from the project, I have learnt a lot and I am really happy with the way my project turned out. There are more features I could add but I am hoping to develop them in the future. I am thankful for all this project has given me. This was truly a great opportunity given to me.

## References

- Type 1 diabetes. (2024). Retrieved April 24, 2024, from HSE.ie website:  
<https://www2.hse.ie/conditions/type-1-diabetes/>
- Type 2 diabetes. (n.d.). Retrieved from <https://www2.hse.ie/conditions/type-2-diabetes/>
- Diabetes. (2024). Retrieved April 24, 2024, from HSE.ie website:  
<https://www2.hse.ie/conditions/diabetes/>
- CDC. (2022, November 3). Get Active! Retrieved April 24, 2024, from Centers for Disease Control and Prevention website:  
[https://www.cdc.gov/diabetes/managing/active.html#:~:text=The%20goal%20is%20to%20get,%2C%20shoulders%2C%20and%20arms\).](https://www.cdc.gov/diabetes/managing/active.html#:~:text=The%20goal%20is%20to%20get,%2C%20shoulders%2C%20and%20arms).)
- Nutrition Analysis API - Nutrition Data Service - Edamam. (2024). Retrieved April 24, 2024, from Edamam.com website: <https://developer.edamam.com/edamam-nutrition-api>
- PhilJay/MPAndroidChart: A powerful 🚀 Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, panning and animations. (2019, March 20). Retrieved April 24, 2024, from GitHub website: <https://github.com/PhilJay/MPAndroidChart>
- Text recognition v2. (2015). Retrieved April 24, 2024, from Google for Developers website: <https://developers.google.com/ml-kit/vision/text-recognition/v2>

- YouTube Data API | Google for Developers. (n.d.). Retrieved from  
<https://developers.google.com/youtube/v3>
- Firebase. (2022). Retrieved April 24, 2024, from Firebase website:  
<https://firebase.google.com/>
- Download Android Studio & App Tools - Android Developers. (2023). Retrieved April 24, 2024, from Android Developers website:  
<https://developer.android.com/studio>