Medical Insulin (Med.In) Android Application

By
Ammar Zulfikar
001201400003

A Thesis
Submitted to the Faculty of Computing
President University
In Partial Fulfillment of the Requirements
For the Degree of Bachelor of Science
In Information Technology

Cikarang, Bekasi, Indonesia

May 2019

Medical Insulin (Med.In) Android Application

By

Ammar Zulfikar

Approved:

Rosalina, M.Kom
Thesis Advisor

Nur Hadisukmana, M.Sc
Head of Information Technology

Rila Mandala, Ph.D
Dean of Faculty of Computing

Medical Insulin (Med.In) Android Application

By

Ammar Zulfikar

Approved:

Rosalina, M.Kom.	Nur Hadisukmana, M.Sc.
Thesis Advisor	Head of Information Technology

Rila Mandala, Ph.D.
Dean of Faculty of Computing

Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other University or Institution.

I have read the Thesis Regulations and I am aware of the potential consequences of any breach of them.

Cikarang, May 2019

Ammar Zulfikar

Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee, and that this thesis has not been submitted for a higher degree to any other University or Institution.

I have read the Thesis Regulations and I am aware of the potential consequences of any breach of them.

Cikarang, May 2019

Ammar Zulfikar

ABSTRACT

Nowadays, technology become the important role for our life. Every industries start to implement technology as crucial role that has a function to make easier work. Healthcare is definitely one of the most important industry that using technology for better action to help patient. This merger is responsible for improving and saving countless lives all around the world.

Many hospitals and practices using medical technology like mobile devices on the job, physicians can now have access to any type of information they need – from drug information, research and studies, patient history or records, and more – within mere seconds. And, with the ability to effortlessly carry these mobile devices around with them throughout the day, they are never far from the information they need. Here in this thesis, trying to develop an application which remind the user to take the insulin

This thesis aims to remining the user who has diabetes for taking insulin. This thesis uses Ionic Angularfire computer language.

DEDICATION

I would like to dedicate this thesis to my parents who always pray, support and encourage me to finish my thesis. Thank you so much.

ACKNOWLEDGEMENTS

I would like to express my gratitude towards Allah SWT. This research is finished to fulfill the requirements to graduate and receive Bachelor of Science degree in President University.

I would also like to express my gratitude to the following:

1. My family, especially my father, mother, brothers and sister who always support me and encourage me in every situation I may be in
2. Mrs. Rosalina, M.Sc. as my thesis advisor for giving me advices and support during the working of this thesis
3. Dean, lecturers and all of the Faculty of Computing staff who contributed on giving me knowledge and experience throughout my study in the faculty
4. My fellow colleagues in Computing batch 2014 who always support me and fight alongside with me
5. Friends that support my thesis, Nisa and Hashfi. Thank you for the support and advise through the hard situation
6. Thanks to Stackoverflow, IonicForum, Github, FirebaseForum and other sites that give me the knowledge about programming to finish this thesis

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

The Introduction is explaining about the Background, Problem Statement, Research Objectives, Scope and Limitation, Thesis Methodology, and Thesis Outline.

1.1 Background

In this era, technology has become crucial part of health care. Technology has been impinging on human health since the origins of human kind. The development of tools, weapons for hunting, clothing, controllable fire, language, money, boats and containers for storing and carrying things are all technologies that played an important role in sustaining and advancing the spread of people around the globe.

Consideration of technology is important in any examination of the organization and functioning of health care services and systems. Major regulatory frameworks and institutions exist solely to manage the introduction and use of safe, effective and efficient technology in health care. Technology may be developed in response to a specific health problem. Many technological developments result from clinicians in hospitals discussing their needs with technology developers. New technology may come from basic research into a health or biological problem. Many of application have been invented to help the doctor and patient. One of health problem is Diabetes. This disease will make a huge impact for the patient if the patient forget to consume the insulin.

This thesis will develop an android application which function is to remind the diabetes patient for intake the insulin on the right time which is also the patient could maintain the blood sugar levels. This application also have healthy lifestyle tips to make the patient living on the right path.

1.2 Problem Statement

Many patient that has diabetes sometime forgot to take the insulin on the right time. The reason is the patient forgot to make a reminder to take the insulin. The diabetes patient should be often checking the blood sugar of their body. On the other hand, the patient also has to consume healthy food and healthy lifestyle.

Specifically, this research aims to answer the following question:

- How to remind the patient to use the insulin on the right time
- How to help the diabetes patient to control the blood sugar levels of patient
- How to remind the patient to have healthy lifestyle.
- How to see the list daily calorie that must be passed the body mass index.

1.3 Thesis Objective

This thesis has an objective which are to:

- Create an application that allows the user to has reminder for take the insulin on the right time
- Show the graphic of every blood sugar data based on the data

1.4 Scope and Limitation

This thesis is focused on developing an application which are able to:

1. Replace the old ways of monitoring the blood sugar level.

2. Provide an easier and more efficient way to have reminder of take insulin.

However, this application has some limitations which are listed below:

1. This application is only application in outline without finishing the UI and UX development that means it is not ready for publish to the public.
2. This application does not have security system so it will be easy to hack.

1.5 Thesis Methodology

The thesis development methodology used to create medical Insulin (Med.In) android is Rapid Application Development (RAD). In software development, RAD is a concept that was created out of frustration of waterfall software design approach which often resulted in products that were out of date or inefficient by the time they were actually released. The term was inspired by James Martin and is a software development methodology that uses minimal planning in favor of rapid prototyping. RAD has four distinct phases as shown in Figure 1.1.

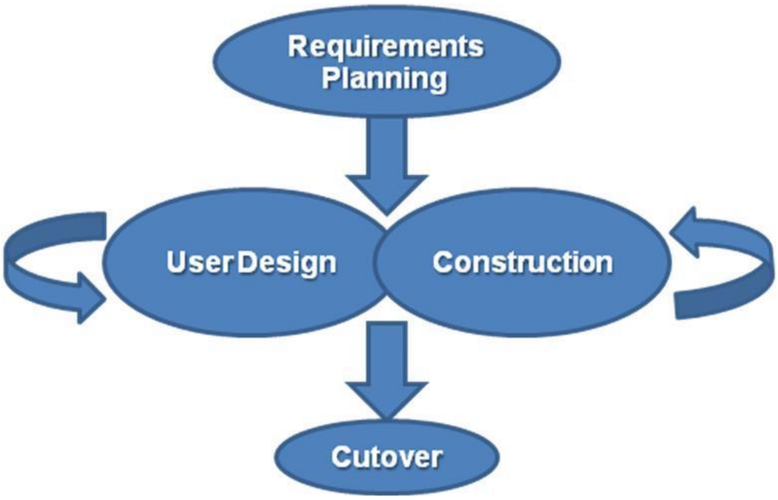


Figure 1.1 RAD Method Life Cycle by James Martin [1]

The phases of RAD model:

1. Planning Requirements Phase

The purpose of this phase is to gather every related literatures and scientific articles as the preliminary investigation to establish requirements, constraints and scope of the thesis.

2. User Design Phase

During this phase, analyzing the gathered data is performed in order to visualize how the program work and the processing of input to output. On this phase, prototypes are developed to represent all system processes, inputs, and outputs.

3. Construction Phase

This phase focuses on program and application development task. However, changes and improvements may still be done on the application as the application developed. The tasks on this phase are programming and application development, coding, unit-integration and system testing.

4. Cut-Over Phase

The purpose of the final phase of the RAD model are to test the features, methods or performances of the programs using a set of predefined test cases. Developers should ensure the main algorithm perform properly towards some special cases of input. The main algorithm's performance needs to be evaluated and compared with others algorithm that perform the same task.

1.6 Thesis Outline

The structure of this thesis consists of seven chapters which are:

1. Chapter I: Introduction

The introduction chapter consists of background, problem statement, thesis objective, scope and limitation, thesis methodology, and thesis outline.

2. Chapter II: Literature Study

The literature study chapter explains the theoretical supports and methods in making this application. This chapter contains explanations about the concept of QR code payment transaction such as QR code, AES algorithm, Online Transaction Processing, Top Up and Balance.

3. Chapter III: System Analysis

The system analysis chapter consists of system overview, comparison overview, functional analysis, the development processes analysis, hardware and software requirement, use-case diagram, use-case diagram with the narrative description, activity diagram, and sequence diagram.

4. Chapter IV: System Design

The system design chapter is discussed in a hierarchical process. This chapter includes the steps which take the elements from the analysis model (Chapter III). This chapter consists of component-level design, interface design, and physical design.

5. Chapter VI: System Implementation

The system implementation reveals the development process of the application. This chapter consists of source code, and user interface screenshots. This chapter includes user interface development and application details.

6. Chapter VI: System Testing

This chapter takes part after the development. It is done to encounter the troubleshooting of some weaknesses and mistakes. This chapter includes testing environment and testing scenario (Functionality, Performance, and Environment).

7. Chapter VI: Conclusion and Future Work

The conclusion and future work chapter gives a summary of the entire thesis obtained from the result of application testing. This chapter also contains some future plans for improving the application.

CHAPTER II

LITERATURE STUDY

The literature study is containing of all the knowledge and method that related to this research. The literature study is consist of Diabetes, Insulin and Blood Sugar Level.

2.1 Diabetes

Diabetes is the increasingly growing metabolic threat of our contemporary era. Diabetes was first described in an Egyptian manuscript from 1500 BC, mentioning “too great emptying of the urine”. Later on, Indian physicians described also the disease and classified it as honey urine by the fact that ants were attacked by patient’s urine. The term “diabetes” or “to pass through” was first used in 250 BC by the Greek Apollonius of Memphis[21].

Diabetes is the condition in which the body does not properly process food for use as energy. Most of the food we eat is turned into glucose, or sugar, for our bodies to use for energy. The pancreas, an organ that lies near the stomach, makes a hormone called insulin to help glucose get into the cells of our bodies. When you have diabetes, your body either doesn't make enough insulin or can't use its own insulin as well as it should. This causes sugars to build up in your blood. This is why many people refer to diabetes as “sugar[4]. Diabetes mellitus is a chronic endocrine disorder, characterized by hyperglycaemia resulting from absolute or relative insulin deficiency[5].

Long-term complications of diabetes include retinopathy with potential loss of vision; nephropathy leading to renal failure; peripheral neuropathy with risk of foot ulcers, amputations, and Charcot joints; and autonomic neuropathy causing gastrointestinal, genitourinary, and cardiovascular symptoms and sexual dysfunction. Patients with diabetes have an increased incidence of atherosclerotic cardiovascular, peripheral arterial, and cerebrovascular disease. Hypertension and abnormalities of lipoprotein metabolism are often found in people with diabetes[8].

The American Diabetes Association's (ADA's) "Standards of Medical Care in Diabetes," referred to as the Standards of Care, is intended to provide clinicians, patients, researchers, payers, and other interested individuals with the components of diabetes care, general treatment goals, and tools to evaluate the quality of care. The Standards of Care recommendations are not intended to preclude clinical judgment and must be applied in the context of excellent clinical care, with adjustments for individual preferences, comorbidities, and other patient factors[9].

The origin and etiology of DM can vary greatly but always include defects in either insulin secretion or response or in both at some point in the course of disease. Mostly patients with diabetes mellitus have either type 1 diabetes (which is immune-mediated or idiopathic) Type 2 DM (formerly known as non-insulin dependent DM) is the most common form of DM characterized by hyperglycemia, insulin resistance, and relative insulin deficiency[11]. And also diabetes type 3 that has related to reflect the newly identified pathogenic mechanism of neurodegeneration[13].

Type 1 diabetes is a disease in which autoimmune destruction of pancreatic β-cells leads to insulin deficiency. Controlling blood glucose with an acceptable range is a major goal of therapy. Measurements of hemoglobin A1c and blood glucose levels are used for both the diagnosis and the long-term management of the disease. This chapter briefly describes the pathophysiology, diagnosis, and management of type 1 diabetes[14**]**.

Type 2 diabetes mellitus (DM) is a chronic metabolic disorder in which prevalence has been increasing steadily all over the world. As a result of this trend, it is fast becoming an epidemic in some countries of the world with the number of people affected expected to double in the next decade due to increase in ageing population, thereby adding to the already existing burden for healthcare providers, especially in poorly developed countries.[15]

There is another diabetes which is Alzheimer's Disease that Being Called Type 3 Diabetes because "It's really more of a research term, rather than a medical term," says Guojun Bu, PhD, a professor of neuroscience who is associate director of the Center for Regenerative Medicine at the Mayo Clinic in Jacksonville, Florida. It's a way to identify the growing body of research into the relationships between insulin resistance in the brain and neurodegenerative conditions that can result in cognitive decline, Alzheimer's disease, or other types of dementia.[16]

With the soaring number of patients reflecting population aging, diabetes demands broader involvement of non-specialist physicians than other diseases. Earlier intervention and continued treatment are the keys to achieving the treatment goals. The importance of close collaboration between specialists and non-specialist physicians continues to increase.[17]

2.2 Insulin

Insulin is a peptide hormone secreted by the β cells of the pancreatic islets of Langerhans and maintains normal blood glucose levels by facilitating cellular glucose uptake, regulating carbohydrate, lipid and protein metabolism and promoting cell division and growth through its mitogenic effects[18].

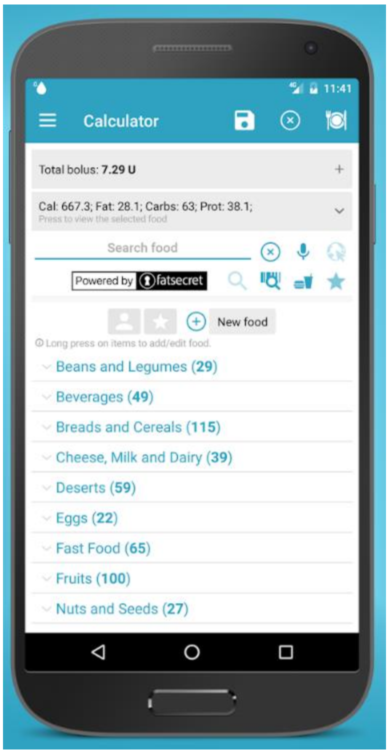
Insulin resistance, i.e., the reduced sensitivity of tissues to insulin mediated biologic activity, has been an important clinical and research interest in adult medicine for decades. Even though it was quickly recognized by endocrinologists as being intimately related to the development of type 2 diabetes, interest in insulin resistance greatly expanded among a wide range of medical specialties after it was shown to be potentially etiologically associated, in general, with cardiovascular risk (CV) and, in particular, with obesity, hypertension and dyslipidemia.[19]

Besides obesity, another important habitual factor that affects the development of insulin resistance and type2 diabetes is sedentary lifestyle. Moreover, sedentary lifestyle is one of the modifiable risk factors of type 2 diabetes and the value of exercise to improve insulin signalling and glucose metabolism cannot be overemphasized [20]

2.3 Related Work

2.3.1 Diabetes :M

This application can find the trends in blood glucose levels and allows the user to calculate normal and prolonged insulin boluses using its highly effective, top-notch bolus calculator. It also has a vast nutrition database, to help the user keep track of food intake and nutrition information, as well as exercise time. Never forget another check with our simple but powerful reminders system. Diabetes:M can analyze the values from the imported data from various glucometers and insulin pumps via the exported files from their respective diabetes management software systems[7].



Total bolus: 7.29 U

Cal: 667.3; Fat: 28.1; Carbs: 63; Prot: 38.1;
Press to view the selected food

Long press on items to add/edit food.

Figure 2.1 Screenshot of Diabetes : M Application

2.3.2 Diabetes Pal

Diabetes Pal helps track, analyze, and share blood glucose, medication, and food data manually and automatically. Diabetes Pal enables to track user progress in real time. With the Telcare BGM®, the app will automatically receive the blood glucose readings. Manual entry modes allow blood glucose storage for non-Telcare BGM users, too!®,Manually enter logs of blood glucose, medication, or food in just a few seconds using our optimized “Add Log” interface. The user can even attach a note to your entries. This application also comes the ability to see automatically generated graphs of user trends over the period of time of user choice[6].

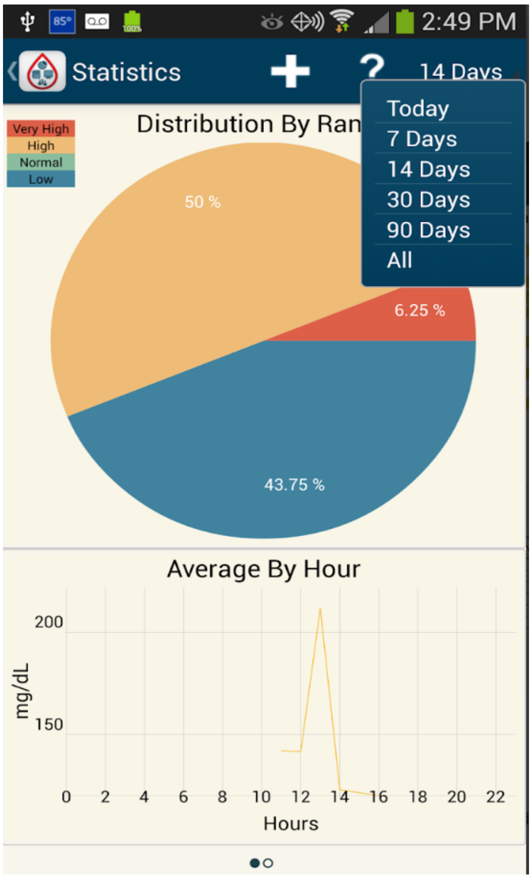


Figure 2.2 Screenshot of Diabetes Pal Application

2.4 Comparison Overview

Listed in Table 2.1 below are the several features that compared from related work with Medical Insulin (Med.In) Android Application.

Table 2.1 Medical Insulin (Med.In) Android Application Comparison Overview

Features	Diabetes :M	Diabetes Pal	Medical Insulin

Insulin Reminder	Yes	No	Yes
Blood Sugar Monitor	Yes	Yes	Yes
Blood Sugar Data	Yes	No	No
Recommendation for healthy lifestyle	Yes	No	Yes

CHAPTER III

SYSTEM ANALYSIS

The system analysis is explaining about the process analysis of system procedure. The content of this chapter are System Overview, Functional Analysis, Development Process Analysis, Hardware and Software Requirements, Use-case Diagram, Use-case Narrative, Swim Lane Diagram, and Sequence Diagram.

3.1 System Overview

This application is intended to help the diabetes patient to monitoring the blood sugar level from table or graphic. This application also provided the recommendation for healthy lifestyle and also the reminder for take insulin on the right time.

3.2 Functional Analysis

There are several functions that provided by the Medical Insulin Android Application. Table 3.1 below will describe this application functions.

Table 3.1 Function Description of Medical Insulin.

No	Function Description
1	Allow user to see reminder for take the insulin
2	Allow user to monitoring the blood sugar level
3	Allow user to input blood sugar level
4	Allow user to see recommendation for healthy lifestyle

5	Allow user to see the blood sugar level on the table

3.3 Hardware and Software Requirements

This thesis development, Med-In Medical Insulin requires following hardware and software in application development process. Hardware needed are listed below:

- Personal Computer

  Personal computer is the workstation for this application in the development process.

- Smartphone

  Smartphone is used as a tool to run and test the application in Android platform.

Software needed are listed below:

- Cordova

  Cordova wraps your HTML/JavaScript app into a native container which can access the device functions of several platforms. These functions are exposed via a unified JavaScript API, allowing you to easily write one set of code to target nearly every phone or tablet on the market today and publish to their app stores. Cordova also act as a compiler in this application.

- Microsoft Office

  Microsoft Office used to document the whole development process of this application.

- Firebase

  Firebase is built on Google infrastructure and scales automatically, for even the largest apps. Firebase is the server, the API and the datastore, all written so generically that user can modify it to suit most needs.

- Web Browser

Web browser is used to check the database as well as the PHP syntax. The preferred web browsing is Google Chrome version 74.0.3729.169 (32-bit).

3.4 Use-Case Diagram

Use case diagrams represent use case, actors and the relationship between the use cases and the actors. The diagram defining the system's limits and the relationships between the system and environments. Use-case diagram for Medical Insulin Android Application can be seen in Figure 3.1.

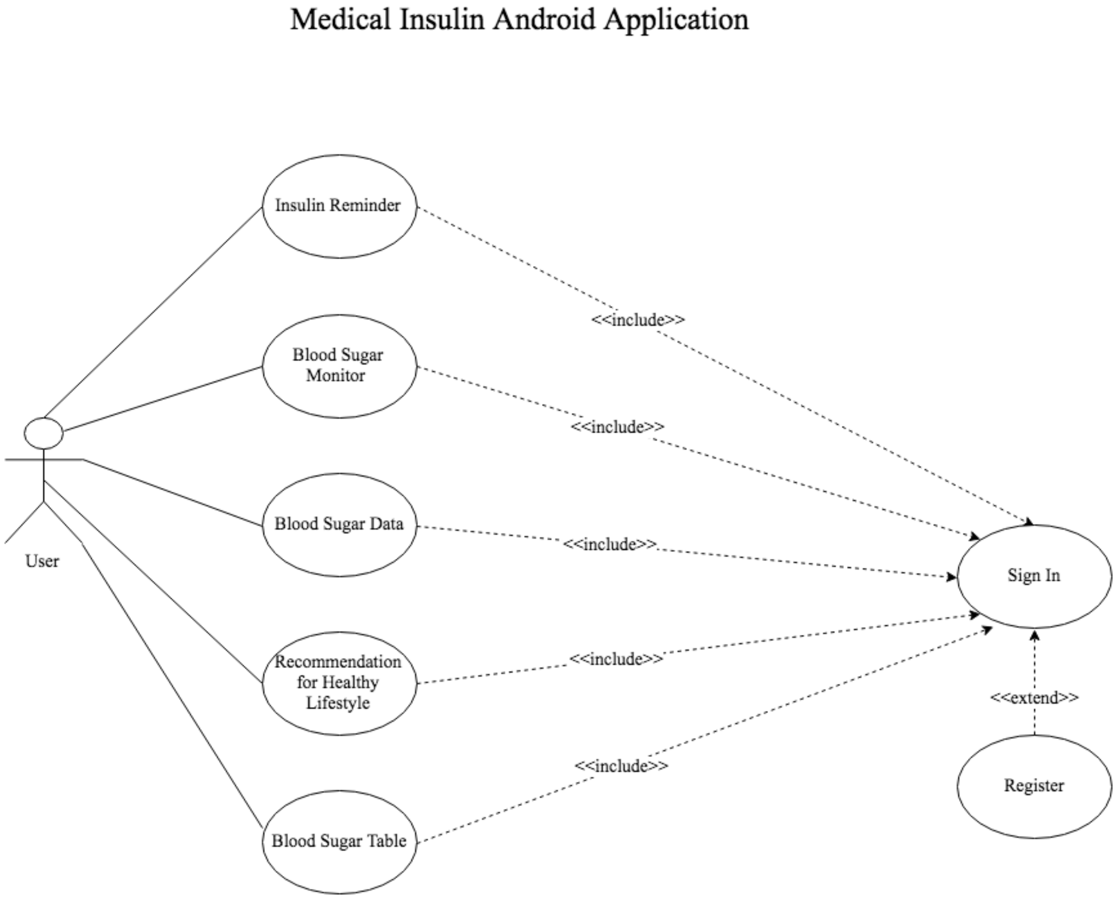


Figure 3.1 Use-Case Diagram of Medical Insulin Android Application

3.5 Use Case Narrative

Use-case diagram with narrative description is the structured textual description of a use case that describes how the user interacts with system. The narrative description describes the detail of use cases such as name, description case, precondition, alternate courses, conclusion, and post condition. The purpose of use-case narrative is helping to identify possible misunderstanding during very early stages.

Table 3.2 Use-Case Narrative for User Sign-in to Application.

Point	Narrative	
Use Case Name	Sign In	
Use Case ID	001	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows registered user to fully use the application by click the sign in button in sign in form	
Precondition	The actor has been registered in the system.	
Typical Course of Events	Actor Action	System Response
	Step 1: User open medical insulin application	Step 2: Display sign in form
	Step 3: Insert e-mail address in the e-mail field	
	Step 4: Insert password in the password field	
	Step 5: Click button Sign In	Step 6: Checking the e-mail and password in the database
		Step 7: Direct the valid user into the main page

Alternate Courses	If the actor inserts an invalid e-mail and/or password, the system will notify that the sign in failed
Conclusion	This use case shows the steps of how the actor sign in to the system
Post Condition	If the use case was successful, the actor will be able fully access the application

Table 3.3 Use-Case Narrative for User Register to Application

Point	Narrative	
Use Case Name	Register	
Use Case ID	002	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows new user to register into the application by click the register button in login form	
Precondition	The actor has not been registered in the system.	
	The actor click button register in sign in form	
	Actor Action	System Response
	Step 1: User click the button Register	Step 2: Show registration form
	Step 3: Input profile data	
	Step 4: Input treatment data	
	Step 5: Click Create Register Button	Step 6: Verify all the data
		Step 7: Save data on database
		Step 8: Directly to sign in page
		Step 5.1: Verify does not success
		Step 5.2: Show the failed message

Alternate Courses	The actor must input all the required field on register page if not the system will notify the register process is failed
Conclusion	This use case shows the steps of how the actor register into the system
Post Condition	If the use case was successful, the actor will be able to login

Table 3.4 Use-Case Narrative for Insulin Reminder

Point	Narrative	
Use Case Name	Insulin Reminder	
Use Case ID	003	
Priority	Low	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows registered user to view the insulin reminder schedule	
Precondition	The actor wants to check insulin reminder schedule	
Typical Course of Events	Actor Action	System Response
	Step 1: User click insulin reminder button	Step 2: Redirect to insulin reminder page
		Step 3: Retrieve the user information from the database
		Step 4: Display all of the insulin reminder information
Alternate Courses	None	
Conclusion	This use case shows the steps of how the actor want to see insulin reminder schedule	
Post Condition	If the user case was successful, the actor information will be displayed for the actor to see	

Table 3.5 Use-Case Narrative for Blood Sugar Monitor

Point	Narrative	
Use Case Name	Blood Sugar Monitor	
Use Case ID	004	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows the user to see the graphic of all insulin type	
Precondition	The actor wants to monitoring the blood sugar of user by seeing from insulin type	
Typical Course of Events	Actor Action	System Response
	Step 1: User click blood sugar monitor button	Step 2: Redirect to blood sugar monitor page
	Step 3: User choose one of insulin type	Step 4: Retrieve the blood data
		Step 5: Display graphic of blood sugar based on the insulin type
Alternate Courses	The system will not show the graphic if the user not input the insulin type data	
Conclusion	This use case shows the steps of how the actor wants to see the graphic of all insulin type	
Post Condition	If the user case was successful, the actor will not be able to see the graphic of all insulin type	

Table 3.6 Use-Case Narrative for Add Blood Sugar Data

Point	Narrative	
Use Case Name	Add Blood Sugar Data	
Use Case ID	005	
Priority	High	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows the user to add blood sugar data	
Precondition	The actor wants to add of the blood sugar data	
Typical Course of Events	Actor Action	System Response

	Step 1: User click blood sugar data button	Step 2: Redirect to blood sugar data page
	Step 3: User choose one of blood sugar data type	
	Step 4: User choose date when checking of blood sugar data	
	Step 5: User input result of blood sugar data	Step 6: Retrieve the blood sugar data type
		Step 7: Retrieve the date blood sugar data
		Step 8: Retrieve the result blood sugar data
Alternate Courses	The actor must input all the required field on add blood sugar data, if not the system will notify the add blood sugar data process is failed	
Conclusion	This use case shows the steps of input blood sugar data	
Post Condition	If the user case was successful, the actor has finish input all the blood sugar data	

Table 3.7 Use-Case Narrative for Recommendation for Healthy Lifestyle

Point	Narrative	
Use Case Name	Recommendation for healthy lifestyle	
Use Case ID	006	
Priority	Medium	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows user to see the recommendation for healthy lifestyle	
Precondition	The actor has sign in to application.	
Typical Course of Events	Actor Action	System Response

	Step 1: The user click recommendation for healthy lifestyle button	Step 2: Retrieve the all data user based on user BMI
		Step 3: Display the healthy lifestyle tips based on user BMI
Alternate Courses	None	
Conclusion	This use case shows the tips of healthy lifestyle user based on the BMI	
Post Condition	If the user case was successful, the actor could see the tips of healthy lifestyle.	

Table 3.8 Use-Case Narrative for Blood Sugar Table

Point	Narrative	
Use Case Name	Blood Sugar Table	
Use Case ID	007	
Priority	Medium	
Primary Business Actor	User	
Primary System Actor	System	
Description	This use case allows user to see the blood sugar from table	
Precondition	The actor has sign in to application.	
Typical Course of Events	Actor Action	System Response
	Step 1: User click table of sugar monitor button	Step 2: Redirect to table blood sugar monitor page
		Step 3: Retrieve the blood data
		Step 4: System display table of blood sugar
Alternate Courses	None	
Conclusion	This use case shows the table of blood sugar	
Post Condition	If the user case was successful, the actor could see the tips of table of blood sugar.	

3.6 Swim Lane Diagram

A swim lane diagram is a process flowchart diagram that provide richer information on who does what. Swim lane flowchart is separated visually by putting the process in lanes. Parallel lines divide the chart into lanes. User lane defines what user does, system defines what program performs in the process, Server defines what Server does when transaction happen. The Swim Lane Diagram of QR Code Payment Transaction can be seen in Figure 3.2 up to 3.9

Figure 3.2 shows that when user open the application. The application will display login page means that user needs to input email and password for login to the application. If the user success to login application, the system will display home page of the application.

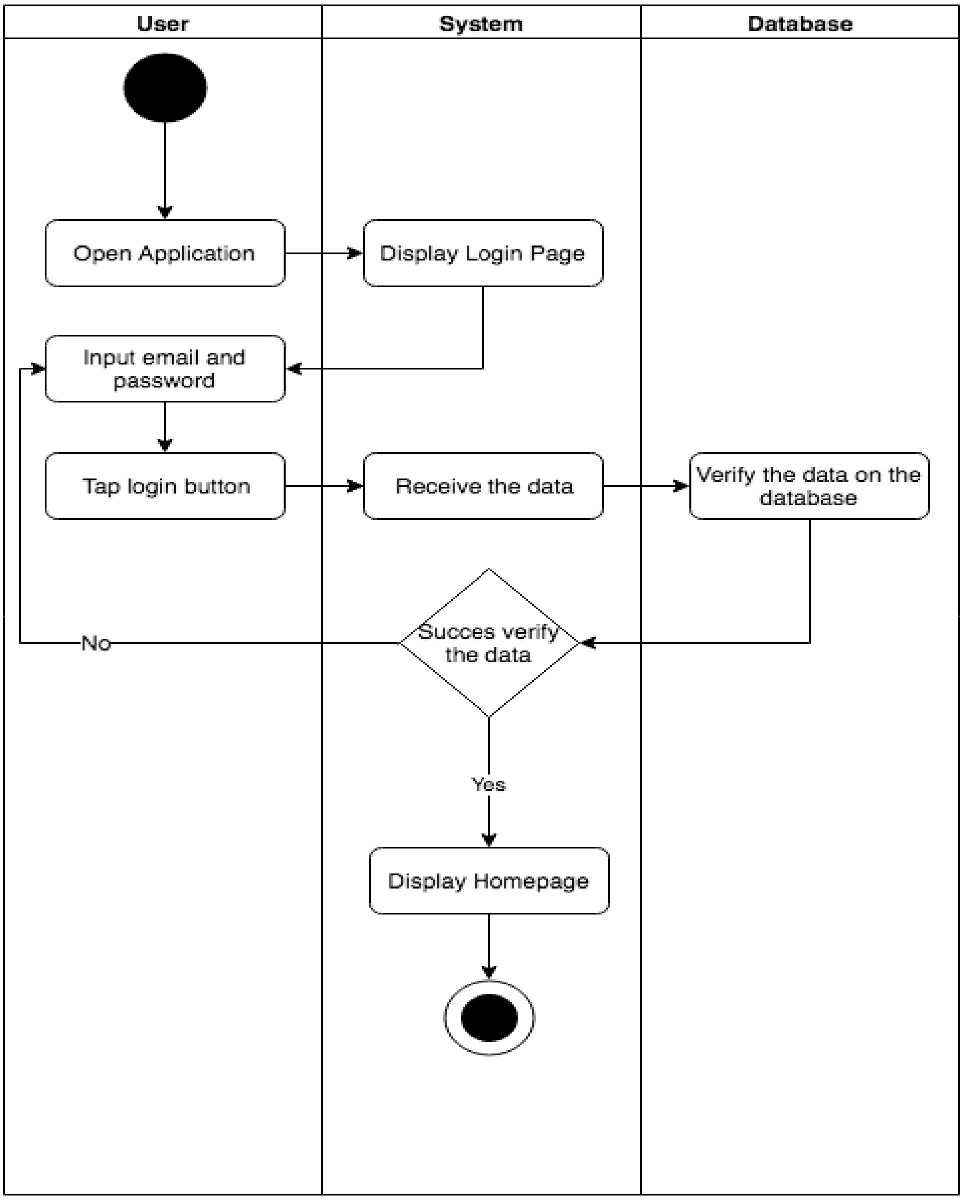


Figure 3.2 Login

Figure 3.3 shows when user in register page. The user needs to input all the fields that required in register page. Then, the system will restore the data to database.

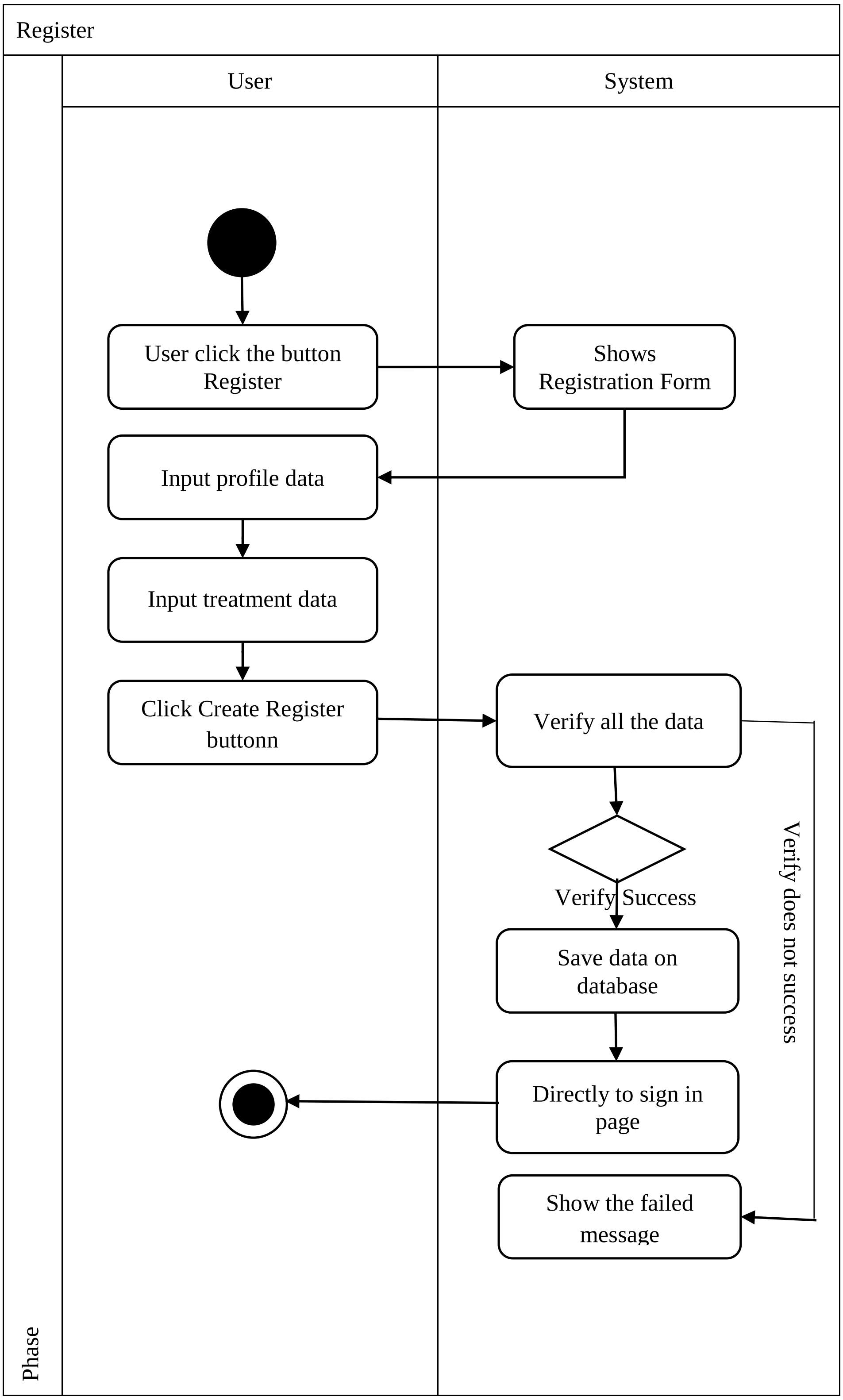


Figure 3.3 Register

Figure 3.4 shows when user click insulin reminder, then system will display the insulin reminder. This page is allowing the user to see the time for taking the insulin.

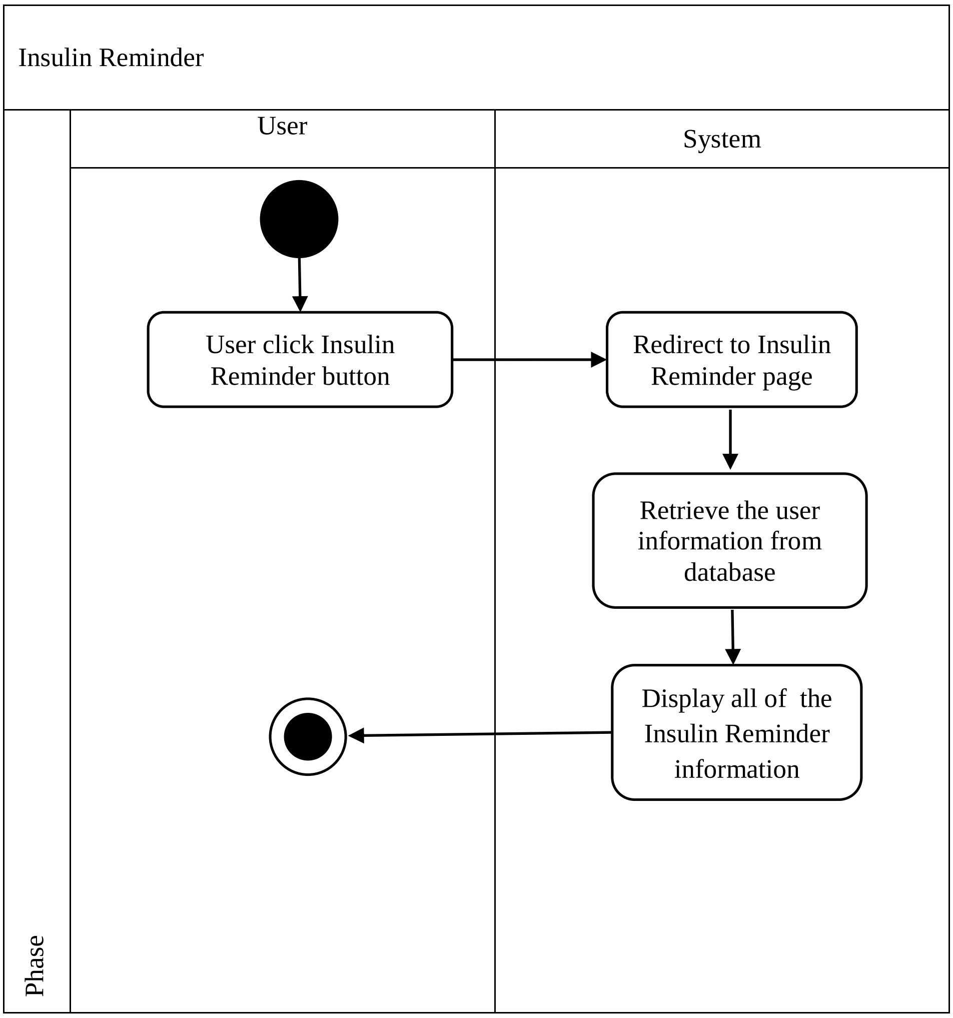

Insulin Reminder
User
System
Phase
User click Insulin Reminder button
Redirect to Insulin Reminder page
Retrieve the user information from database
Display all of the Insulin Reminder information

Figure 3.4 Insulin Reminder

Figure 3.5 shows when user click button on blood sugar monitor button, the system will display the list option of insulin type. After user choose the insulin type, the graphic of blood sugar user will display.

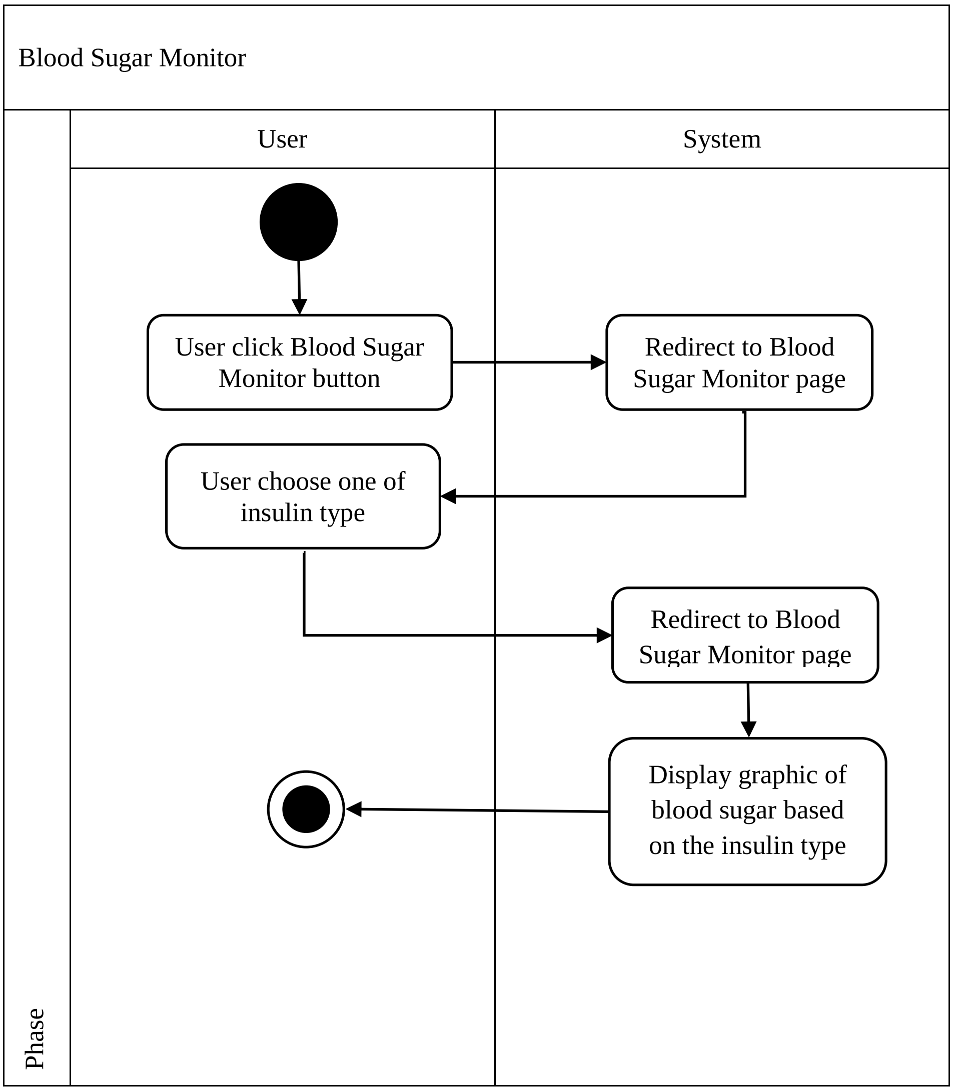

Blood Sugar Monitor
User
System
Phase
User click Blood Sugar Monitor button
Redirect to Blood Sugar Monitor page
User choose one of insulin type
Redirect to Blood Sugar Monitor page
Display graphic of blood sugar based on the insulin type

Figure 3.5 Blood Sugar Monitor

Figure 3.6 shows when user click blood sugar data button, then system will display the input blood sugar data page. After user input insulin type that user wants, date and the result of blood sugar.

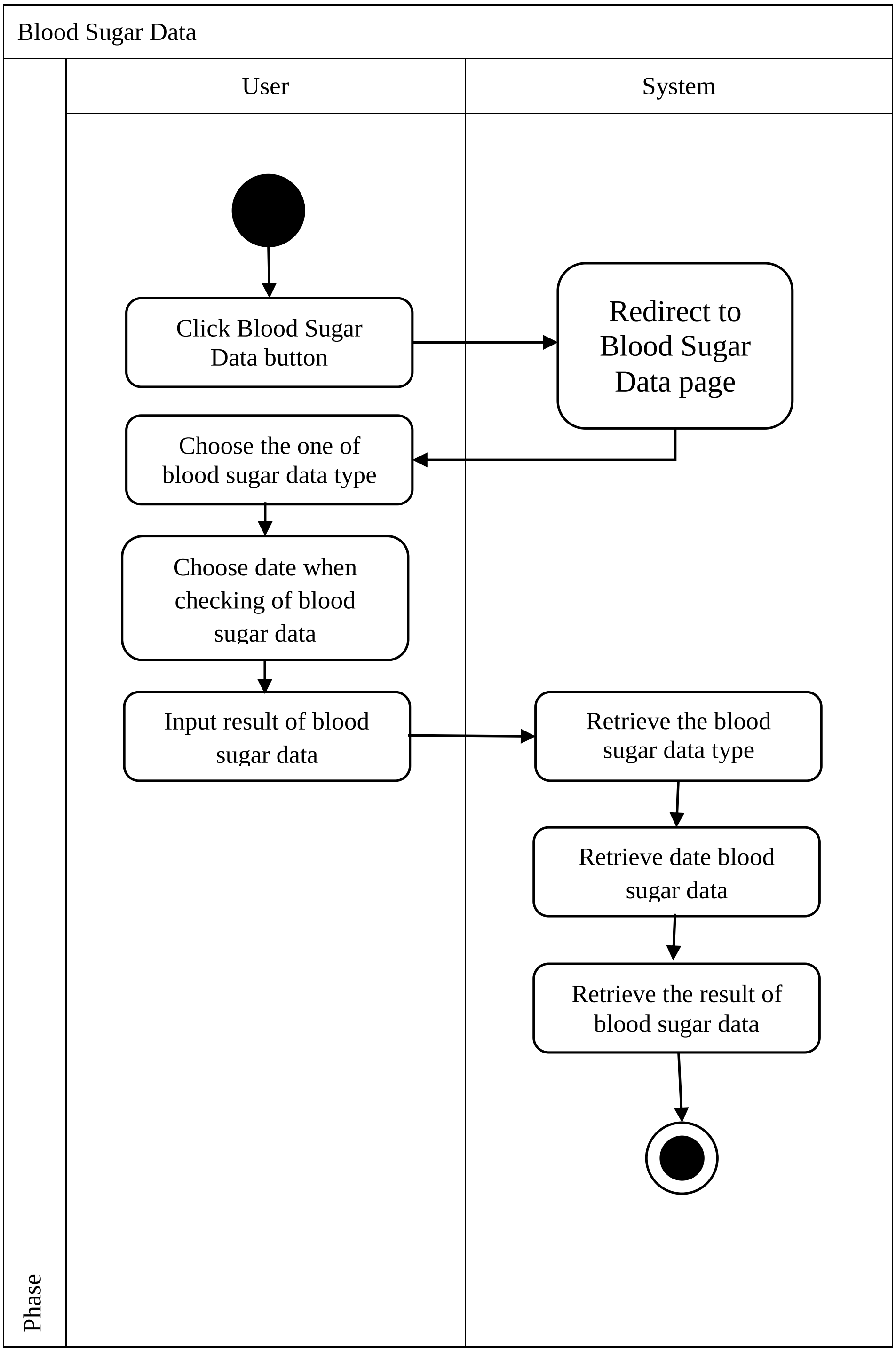

Blood Sugar Data
User
System
Phase
Click Blood Sugar Data button
Redirect to Blood Sugar Data page
Choose the one of blood sugar data type
Choose date when checking of blood sugar data
Input result of blood sugar data
Retrieve the blood sugar data type
Retrieve date blood sugar data
Retrieve the result of blood sugar data

Figure 3.6 Blood Sugar Data

Figure 3.7 shows when the user click recommendation for healthy lifestyle button, then system will retrieve all the information from database and display the recommendation for healthy lifestyle based on BMI user.

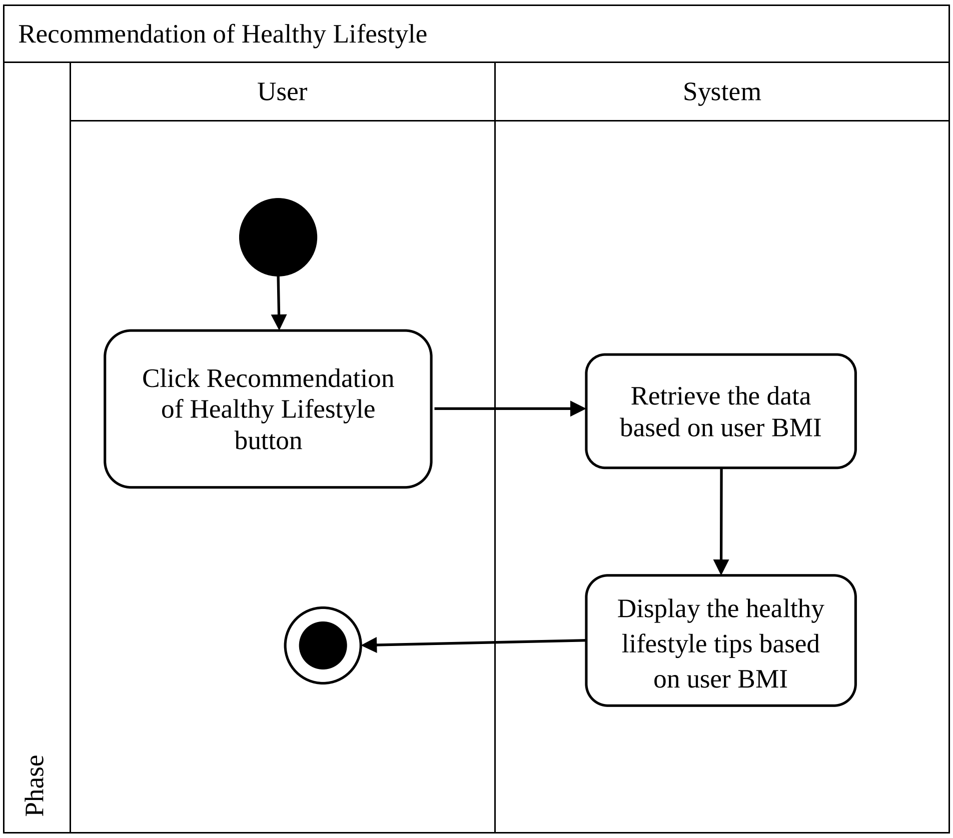

Recommendation of Healthy Lifestyle
User
System
Phase
Click Recommendation of Healthy Lifestyle button
Retrieve the data based on user BMI
Display the healthy lifestyle tips based on user BMI

Figure 3.7 Recommendation for Healthy Lifestyle

Figure 3.8 shows when the user click table of blood sugar monitor button, then system will display the table of blood sugar monitor. This page has a function to see the blood sugar data on table.

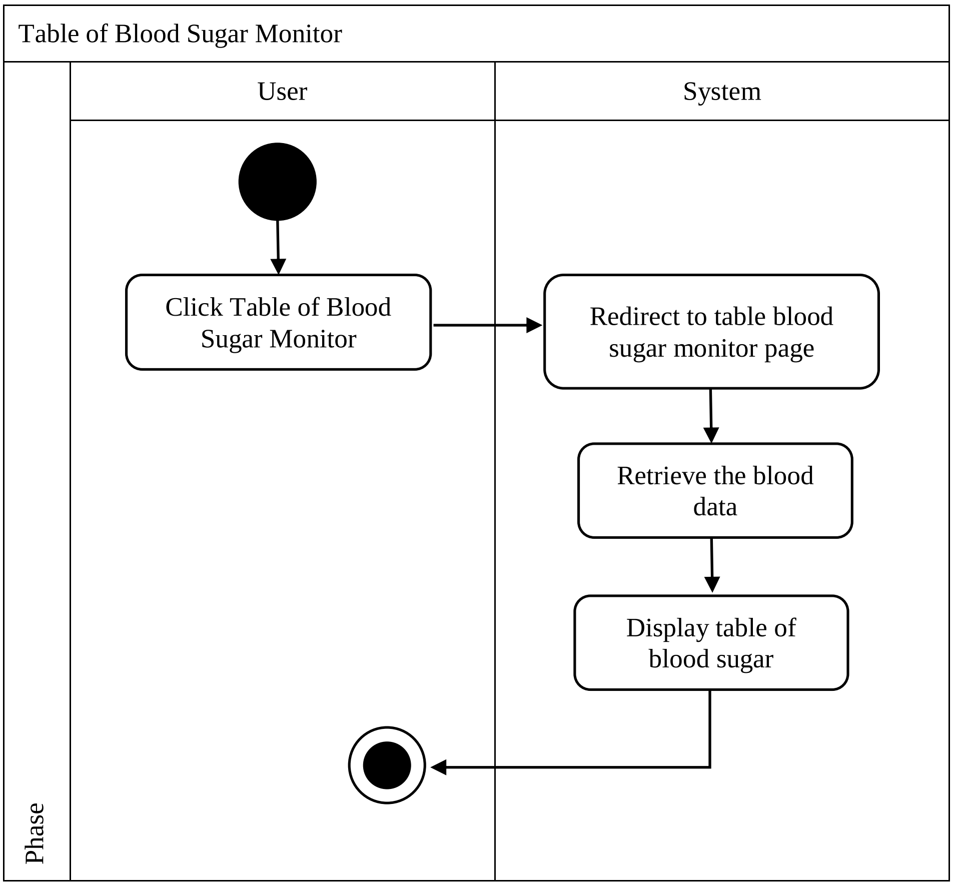


Table of Blood Sugar Monitor
User
System
Phase
Click Table of Blood Sugar Monitor
Redirect to table blood sugar monitor page
Retrieve the blood data
Display table of blood sugar

Figure 3.8 Table of Blood Sugar Monito

CHAPTER IV

SYSTEM DESIGN

Systems design also defining the process elements of a system development such as modules, architecture, components, interfaces, class diagram and data for a system analysis which based on the specified requirements. Systems design implementing the systematic approach to the design of a system. System design are divided into three sections such as user interface design, physical design, database design, and class diagram.

4.1 User Interface Design

The user interface one of the most important parts of any program. The goal of the user interface design is the design of the application that has focus on usability and efficiency. This section is show the interaction between user and the program when the user using the application. The details of the features on the screen is explained on this section. The User Interface (UI) design is created by Cordova application can be seen in Figure 4.1 until Figure 4.10

4.1.1 Login

Login page is the first page that are shown to the user after they enter the application. There are two field texts and two buttons. The field texts are to input email and the passwords. And the other button are for login and register. There is content for description of the application in the login page. Figure 4.1 below shows it and Table 4.1 describe the figure.

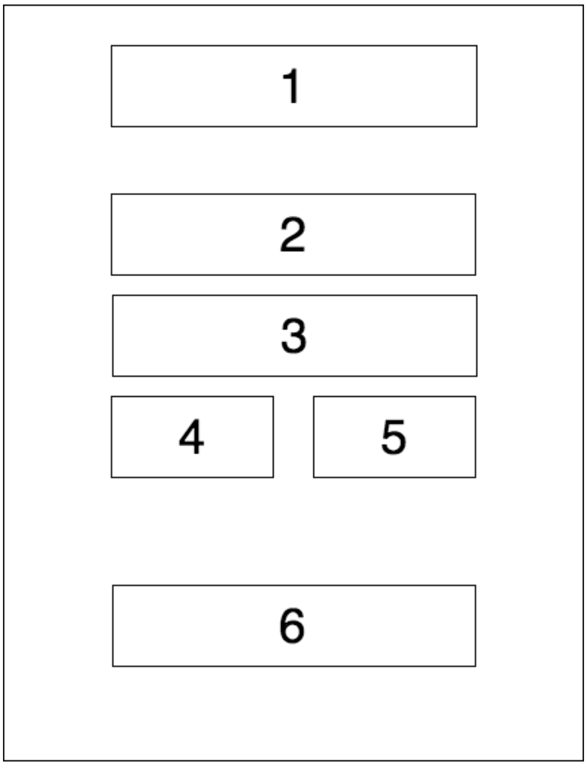


Figure 4.1 Login Interface

Table 4.1 Login Interface Description

No	Description
1	Name of Page
2	E-Mail
3	Password
4	Login Button
5	Register Button

4.1.2 Register

The register page displays the part that user needs to input e-mail, password, confirm password. There is confirm button to store all the information. Figure 4.2 below shows it and Table 4.2 describe the figure.

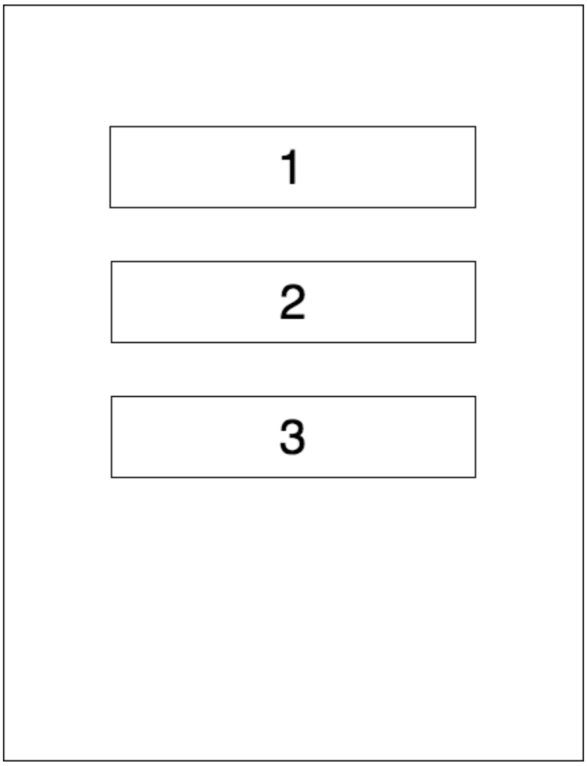


Figure 4.2 Register Interface

Table 4.2 Register Page Interface Description

No	Description
1	Name of Page
2	Input E-Mail
3	Input Password
4	Input Confirm Password
5	Confirm Button

4.1.3 Insulin Reminder

Insulin Reminder is where the user can see the reminder time for taking the insulin. Figure 4.3 below shows it and Table 4.3 describe the figure.

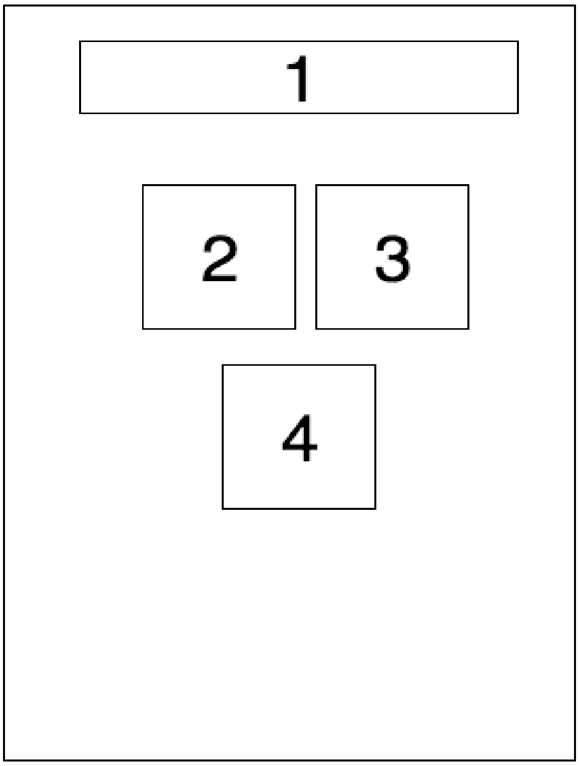


Figure 4.3 Insulin Reminder Interface

Table 4.3 Insulin Reminder Interface Description

No	Description
1	Morning Reminder
2	Afternoon Reminder
3	Evening Reminder

4.1.4 Blood Sugar Monitor

This page is to monitoring the blood sugar level of the user that will display on graphic. Figure 4.4 below shows it and Table 4.4 describe the figure.

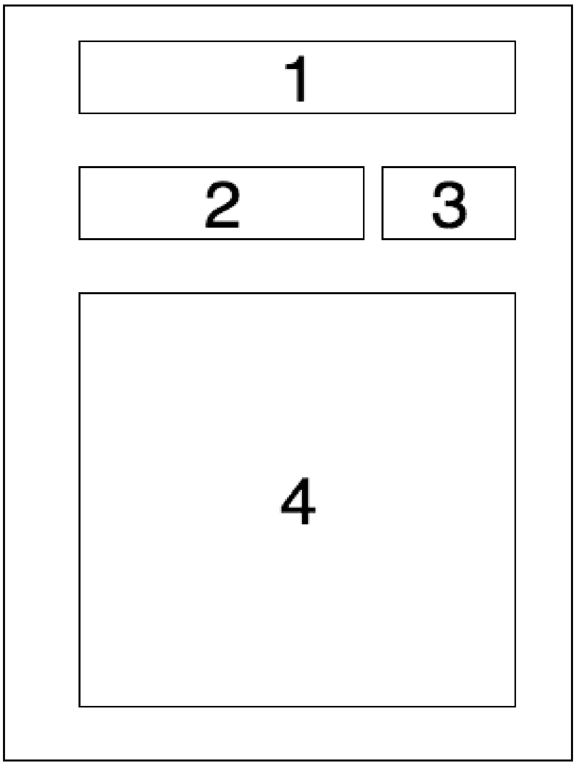


Figure 4.4 Blood Sugar Monitor Interface

Table 4.4 Blood Sugar Monitor User Interface

No	Description
1	Name of Page
2	Insulin type option
3	Check Button
4	Graphic Insulin

4.1.5 Blood Sugar Data

This page is to add data of blood sugar that will be used on show the blood sugar level on graphic. Figure 4.5 below shows it and Table 4.5 describe the figure.

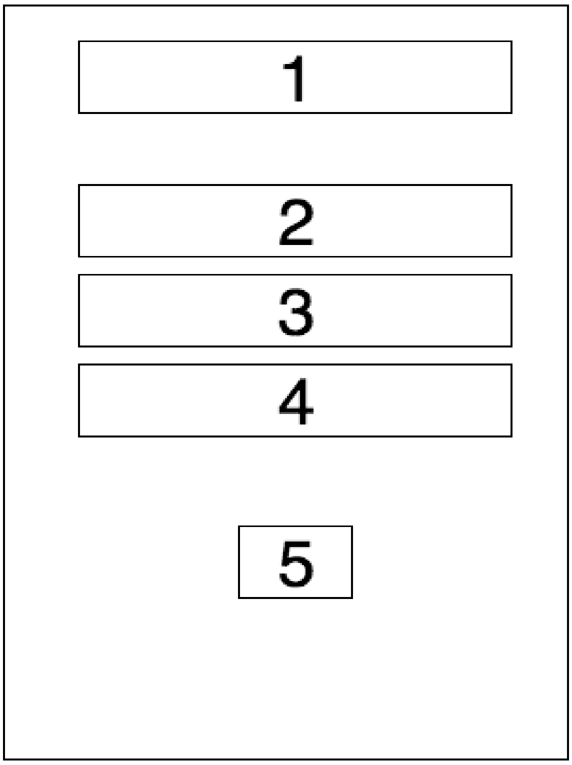


Figure 4.5 Blood Sugar Data Interface

Table 4.5 Blood Sugar Data Interface Description

No	Description
1	Name of Page
2	Input Insulin Type Option
3	Input Date
4	Input Result
5	Add Button

4.1.6 Recommendation for Healthy Lifestyle

This page is to show the tips for healthy lifestyle based on BMI user. Figure 4.6 below shows it and Table 4.6 describe the figure.

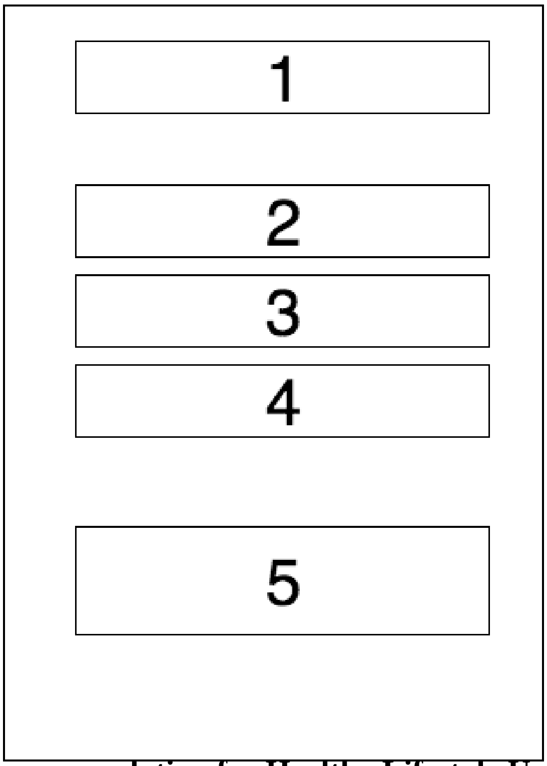


Figure 4.6 Recommendation for Healthy Lifestyle User Interface

Table 4.6 Recommendation for Healthy Lifestyle Interface Description

No	Description
1	Name of Page
2	Name
3	Age
4	BMI
5	Tips Healthy Lifestyle

4.1.7 Blood Sugar Table

This page is to show the table of blood sugar based on blood sugar data. Figure 4.7 below shows it and Table 4.7 describe the figure.

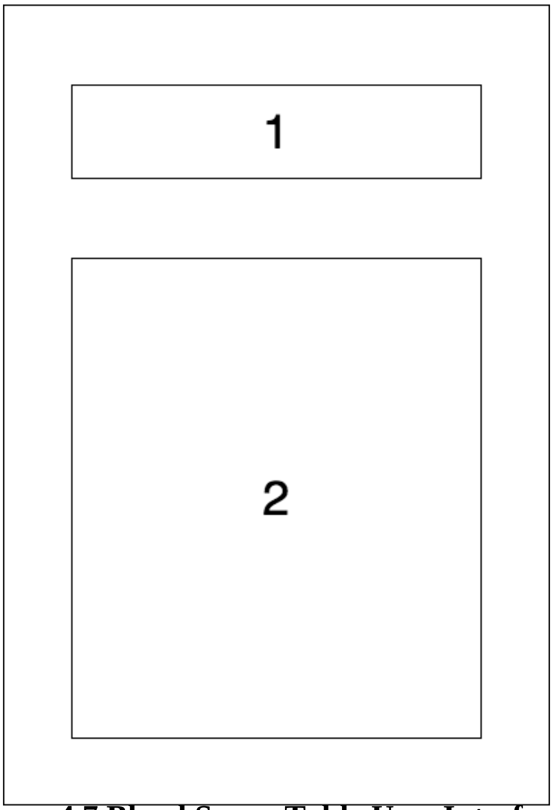


Figure 4.7 Blood Sugar Table User Interface

Table 4.7 Blood Sugar Table Interface Description

No	Description
1	Name of Page
2	Table

4.2 Physical Design

The physical design phase includes the software and hardware which are required during the Medical Insulin development process. This section will be explain about the development tools and the minimum specification to maximize the application performance. The tools are in form of software which used to develop the application and hardware supposed exceed the minimum requirements

4.2.1 Software

Table 4.12 Medical Insulin Software Requirements

No	Field	Description
1	Operating System	Windows 10 64bits.
2	Programming Language	The programming language used for the application development is Ionic Angularfire.
3	Program Development	Cordova is used for the programming environment because it is easier and faster to develop an android application.
4	Microsoft Office	Microsoft Office used as a documentation tools which help the creation of thesis document including the diagrams and tables

4.2.2 Hardware

Table 4.13 Medical Insulin Hardware Requirements

No	Hardware	Description
1	Smartphone Operating System	Minimum requirement is Android Jelly Bean 4.3 or above is recommended for running Medical Insulin Android Application
2	Memory	Minimum requirement is 1024 MB RAM or above is recommended for running Medical Insulin Android Application.
3	Hard Drive	Minimum requirement is 50 MB of storage free size or above is recommended Medical Insulin Android Application.

4.3 Database Design

Database is a collection of information that is organized so that it can be easily accessed, managed and updated. There are three tables at database as shown at

Figure 4.4 which are users, data and blood. Table of users has seven attributes that contains userId as Primary Key, age, height, weight, patientname, total, username. Table of data has five attributes which are userId as Primary Key, dosage, frequenceofUssage, insulintype, and nameInsulin.Table of blood has four attributes that consist of userId as Primary Key, checkingtype, date, and result

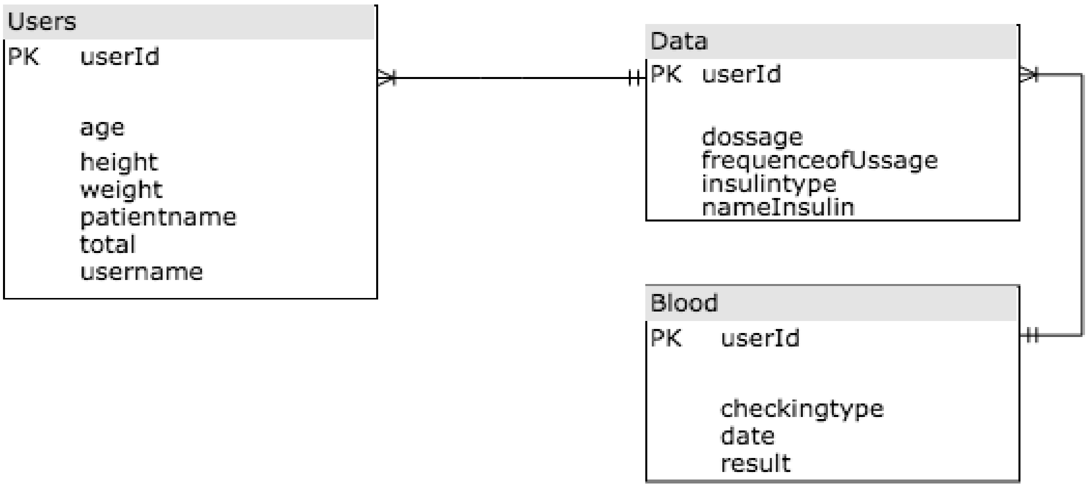


Figure 4.8 Database Diagram of Medical Insulin

4.4 Class Diagram

Class diagram is the description of system structure by showing all system’s classes. The attributes, operations, and the relationships is shown into static diagram. Class diagrams are used for data modelling, its classes represent both the main objects, interaction of each classes to be programmed. There are six classes diagram which are login, register, blood sugar data, blood sugar monitor, and notification ionic. The picture of class diagram can be seen in Figure 4.9

Login is a class that handle to login to the application. The user has to input username and password.

Register is class that handle the unregistered user to register before login to application. For every user that has been register to the application will have own userId.

Blood Sugar Data is class that handle the input data of blood sugar that use on display the graphic blood sugar monitor.

Blood Sugar Monitor is class that handle to display the graphic based on the blood sugar data that input by user.

Notification Ionic is a class that handle the notification of the application.

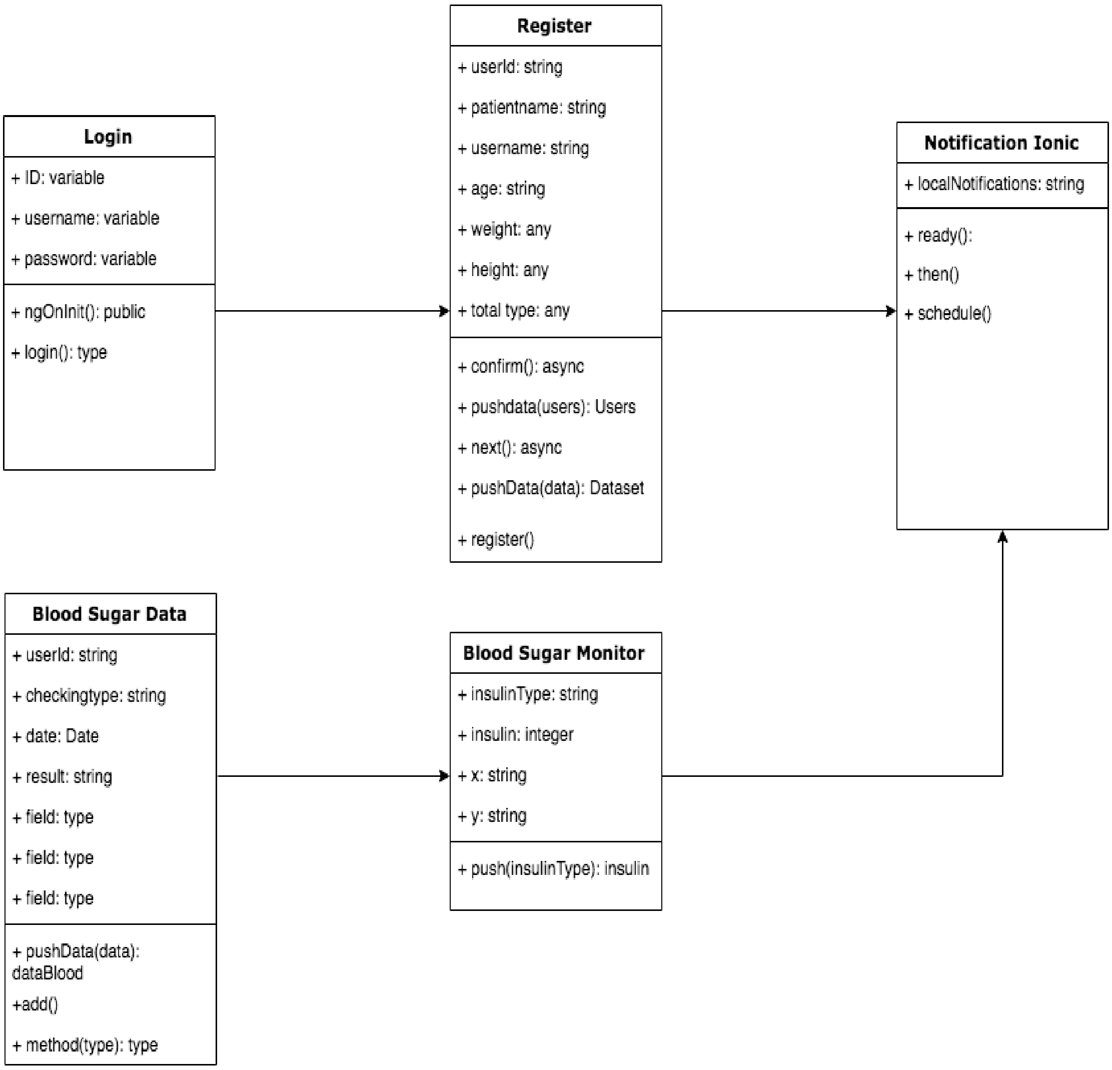


Figure 4.9 Class Diagram of Medical Insulin Android Application

CHAPTER V

SYSTEM IMPLEMENTATION

The system implementation chapter is explaining the system development android application. The system development of this application was developed using some tools, which are Cordova as the compiler that using Ionic Angularfire as the programming language, and Firebase as the real time database. This chapter is containing of User Interface Development and Coding Structure. User Interface Development explaining about every component in the user interface of this application and for coding structure explaining about the important part of source code on the application.

5.1. User Interface Development

The interface of QR Code Payment Transaction is containing of several Interfaces. There are Login Interface, Register Interface, Insulin Reminder Interface, Blood Sugar Monitor Interface, Blood Sugar Data Interface, Recommendation for Healthy Lifestyle Interface, and Blood Sugar Table Interface. The details of user interfaces Medical Insulin Android Application describe in Figure 5.1 up to Figure 5.7

5.1.1 Login Interface

When the user accesses the application, the login Interface will be . The registered user can type the email and password account for login in this application

. There is register button for the unregistered user for register to the application that directly go to register page. Figure 5.1 below shown the login Interface of the application.

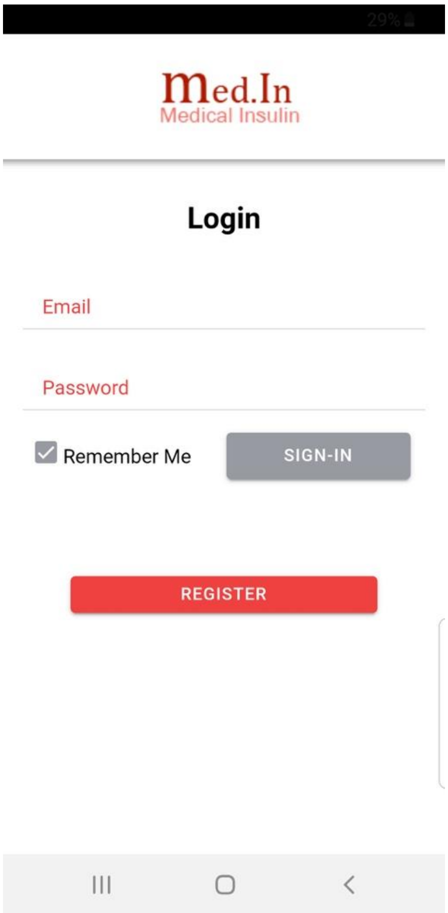


Figure 5.1 Login Page

5.1.2 Register Interface

The register Interface is display all the part that user needs to fill the requirements. The parts that user needs to input are e-mail, phone number, e-mail, password, and confirm password. There is here is confirm button to store all the user information into database. Figure 5.2 below shown the screen shot of register interface of the application.

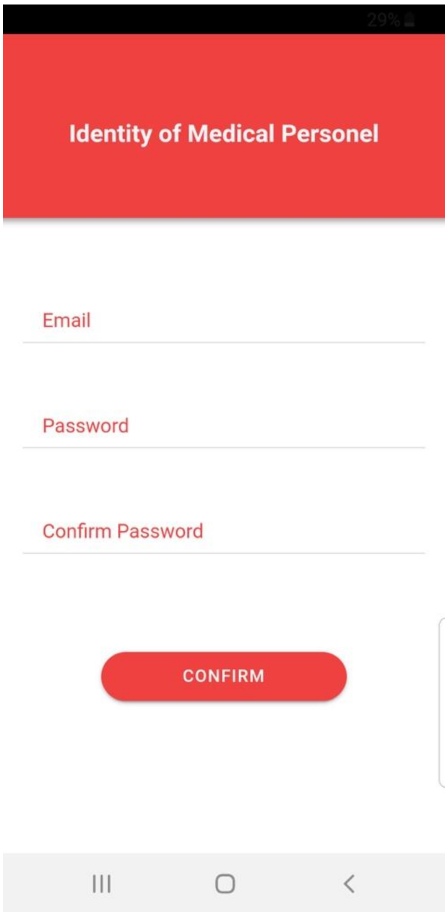
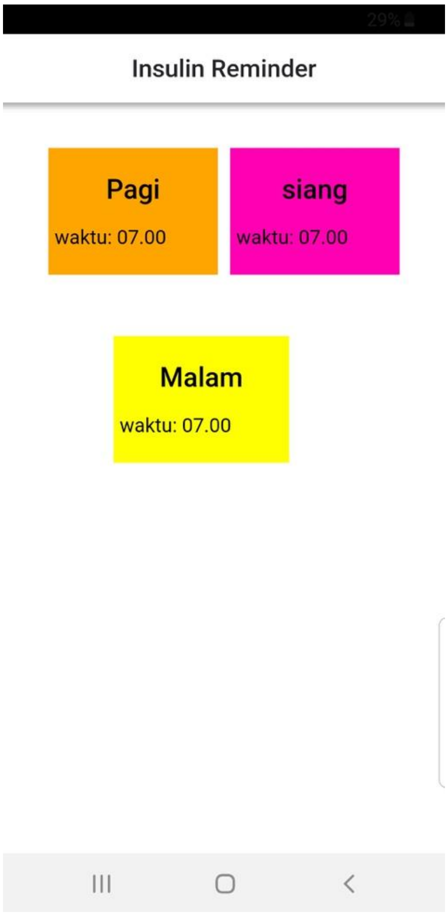


Figure 5.2 Register Page

5.1.3 Insulin Reminder Interface

This interface is display the reminder for user to taking the insulin. That containing of morning reminder, afternoon reminder, and evening reminder. Figure 5.3 below shown the screen shot of insulin reminder interface of the application.



Insulin Reminder

Pagi

waktu: 07.00

siang

waktu: 07.00

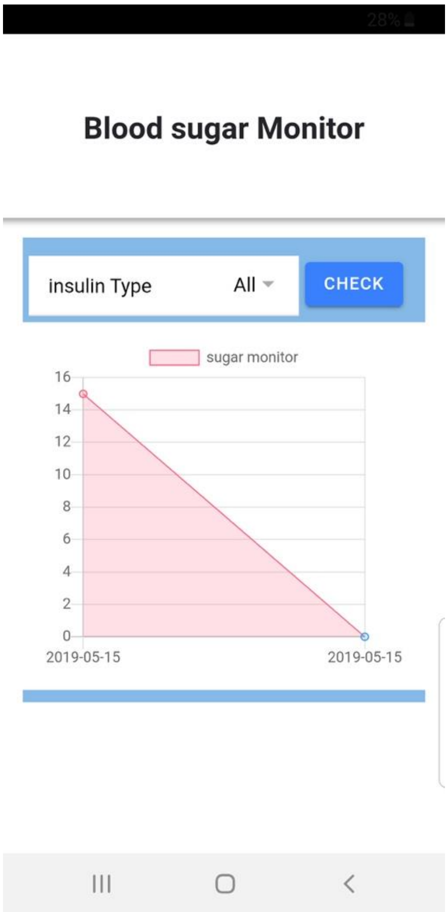
Malam

waktu: 07.00

Figure 5.3 Insulin Reminder Page

5.1.4 Blood Sugar Monitor Interface

This interface is display the blood sugar monitor user Interface. This page has function to display the blood sugar on the graphic based on the insulin type. The user could change the insulin type on the option section. Figure 5.4 below shown the screen shot of blood sugar monitor interface of the application.



insulin Type All

Figure 5.4 Blood Sugar Monitor Page

5.1.5 Blood Sugar Data Interface

This interface is display the blood sugar data user interface. The interface is displaying the input for blood sugar data that consist of insulin type, date and result of the insulin type. Figure 5.5 below shown the screen shot of blood sugar data user interface on the application.

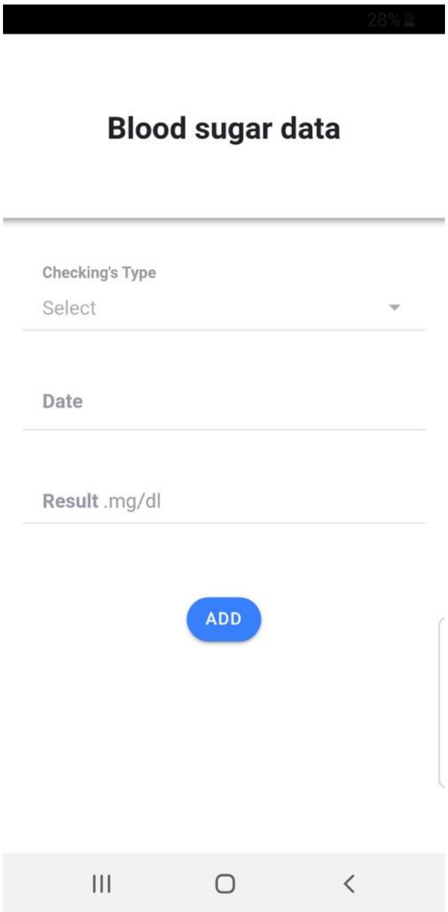
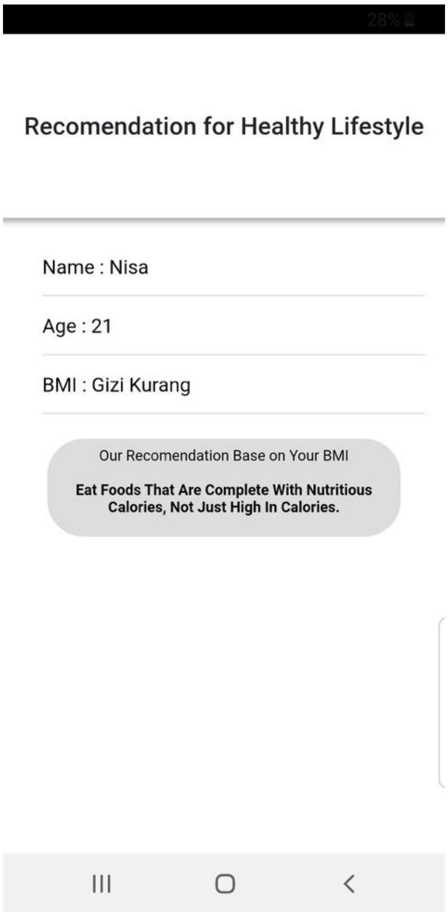


Figure 5.5 Blood Sugar Data Page

5.1.6 Recommendation for Healthy Lifestyle Interface

This interface is display the recommendation for healthy lifestyle user interface. This page will display the tips for healthy lifestyle based on BMI user. Figure 5.6 below shown the screen shot of recommendation for healthy lifestyle user interface of the application.



Recomendation for Healthy Lifestyle

Name : Nisa

Age : 21

BMI : Gizi Kurang

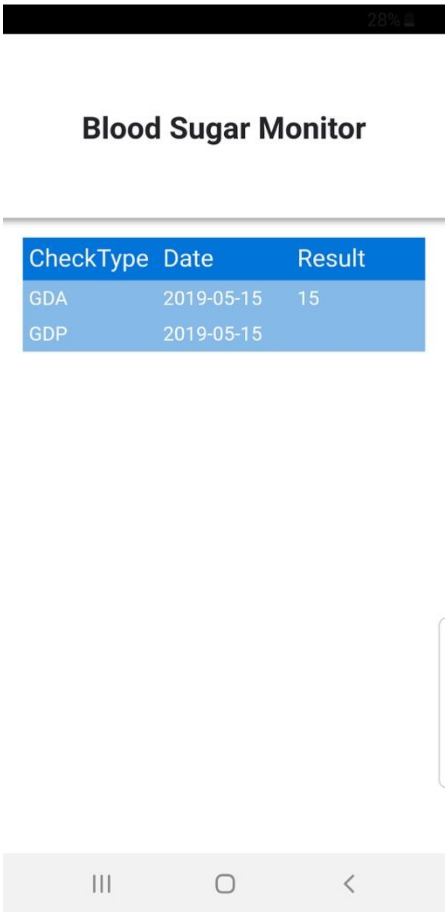
Our Recomendation Base on Your BMI

Eat Foods That Are Complete With Nutritious Calories, Not Just High In Calories.

Figure 5.6 Recommendation for Healthy Lifestyle Page

5.1.7 Blood Sugar Table Interface

This interface is display table of blood sugar monitor feature. This page is show the table of blood sugar based on blood sugar data. Figure 5.7 below shown the screen shot of blood sugar table interface of the application.



Blood Sugar Monitor

CheckType	Date	Result
GDA	2019-05-15	15
GDP	2019-05-15	

Figure 5.7 Blood Sugar Table Page

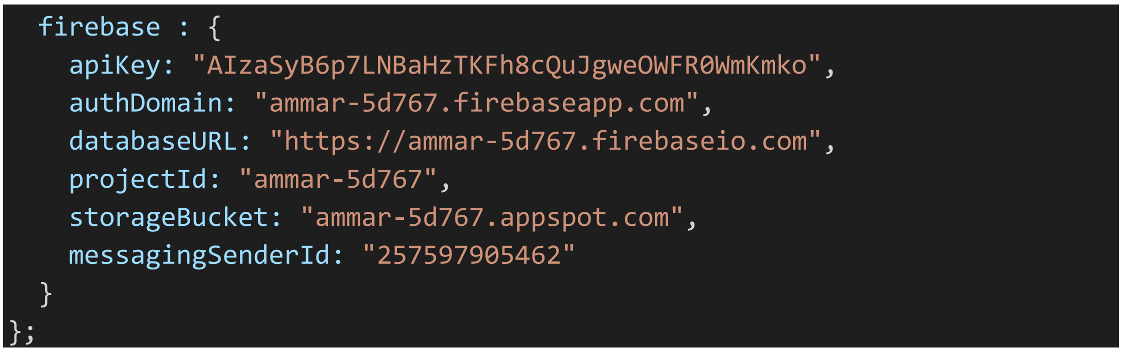
5.2. Coding Structure

Coding structure explaining the technical code that used on development of application. The functionality and logic of the script will be describing in this section

5.2.1 Firebase Credential

This application is using Firebase to handle the backend. The database of this app uses firestore. Moreover, at the first time before creating the firebase project, user need to called the credential function from firebase. Below is this project credential of firebase. Firebase credential used for set the key of each project. It means every project will have different firebase credentials. It will be shown in Figure 5.12.

```
export const environment = {
  production: false,
```



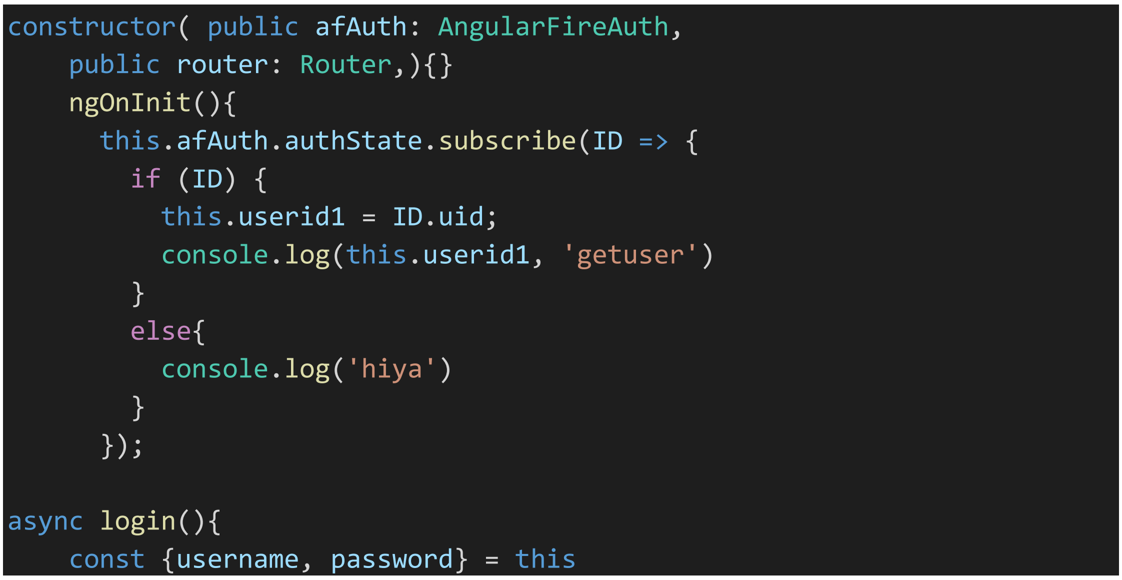
```
  firebase : {
    apiKey: "AIzaSyB6p7LNBaHzTKFh8cQuJgweOWFR0WmKmko",
    authDomain: "ammar-5d767.firebaseapp.com",
    databaseURL: "https://ammar-5d767.firebaseio.com",
    projectId: "ammar-5d767",
    storageBucket: "ammar-5d767.appspot.com",
    messagingSenderId: "257597905462"
  }
};
```

Environment.ts

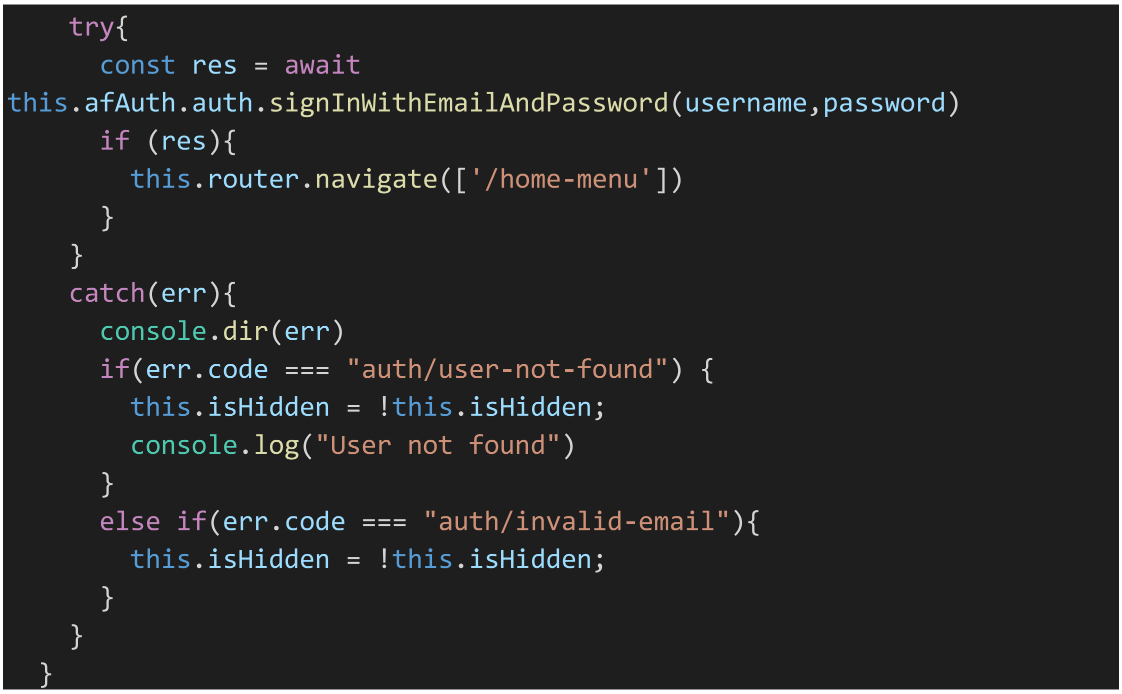
Figure 5.8 Firebase Credential.xml

5.2.2 Login Activity

The login using firebase algorithm **signInWithEmailAndPassword**. Bellow shows the codes for login, user is required to fill the field to be able to use the feature of the application. So, the first step is user needs to enter the username and password and then click the Login button. To execute the **signInWithEmailAndPassword** it needs to call the angularfire authentication that already create in constructor which is afAuth. Then the customerLogin Function run the sign in function by using the email and password that already inputted in string. If the sign in function success it will redirect to customer home page. However, if the sign in failed, there will be error displayed in the page. The input data code is shown on Figure 5.13



```
constructor( public afAuth: AngularFireAuth,
    public router: Router,){}
    ngOnInit(){
      this.afAuth.authState.subscribe(ID => {
        if (ID) {
          this.userid1 = ID.uid;
          console.log(this.userid1, 'getuser')
        }
        else{
          console.log('hiya')
        }
      });

async login(){
    const {username, password} = this
```



```
  try{
   const res = await
this.afAuth.auth.signInWithEmailAndPassword(username,password)
   if (res){
    this.router.navigate(['/home-menu'])
   }
  }
  catch(err){
   console.dir(err)
   if(err.code === "auth/user-not-found") {
    this.isHidden = !this.isHidden;
    console.log("User not found")
   }
   else if(err.code === "auth/invalid-email"){
    this.isHidden = !this.isHidden;
   }
  }
 }
```

Home.page.ts

Figure 5.9 Login Activity Code

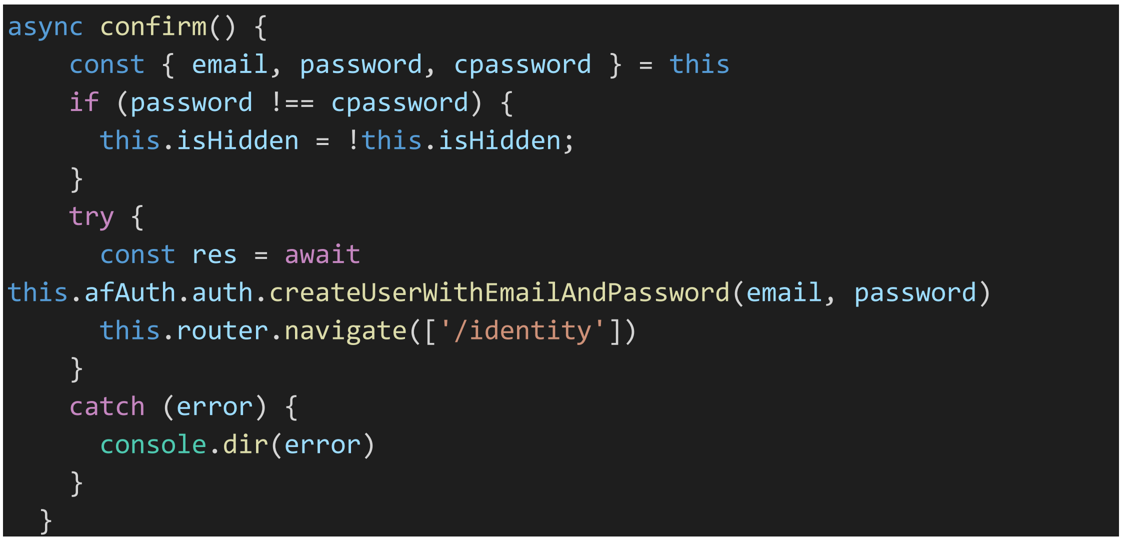
5.2.3 Register Activity

There are three steps to register. First of all is registering the Email and Password that will be store in Firebase authentication. Then the user need to input the user data. After that in the next page the user must input the insulin data and the register will be done.

The storing data to register is simple, all the string only need to store using CreateUserWithEmailAndPassword. After it works the page will redirect to next page which is to input user data. However when the register is failed, the error console will be shown.

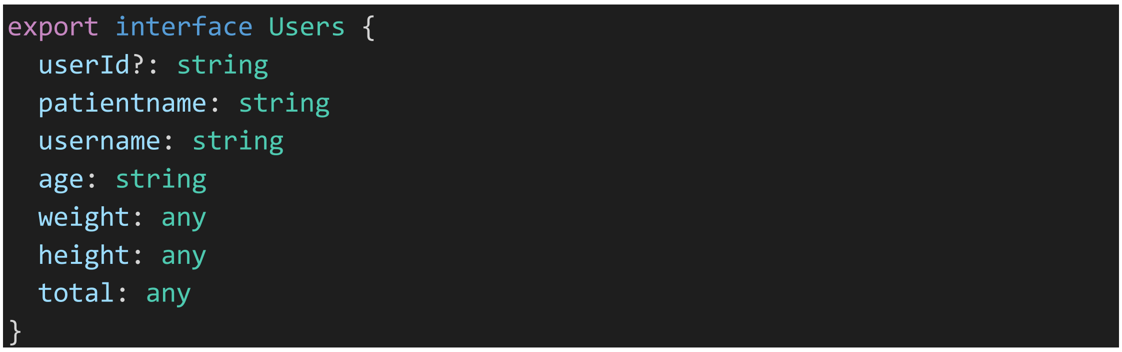
After register the email and password the apps will be redirect to the patient identity page and the user must input the data and the data of user will be handle in user.service.ts. the user.service.ts used as database handler for ‘user’ database which will call in the identity page to push database in firestore. In identity.page.ts the string of object will be send to the data from the user.service after that the auth uses to check if there is any login then it will be get the userId value. Next the userId value will be store in the database to set as foreign key. After that, the count function uses to count the BMI value which is Height divide by width. All the string value that already input will be store to firestore database as a object of ‘Users’.

After register the profile data the apps will continue the process into the treatment data. Actually in the treatment.data.ts there are 2 main algorithm which is input data and Ionic Local Notification. However, in this part it only describe about pushing the data. This algorithm is almost same with identity.page.ts the only different is it push the data to ‘data’ database. Data.service.ts uses to handle database. After that it will store to treatment.data.ts then check the AuthId after all finish then push the data into firestore.. The input data code is shown on Figure 5.15

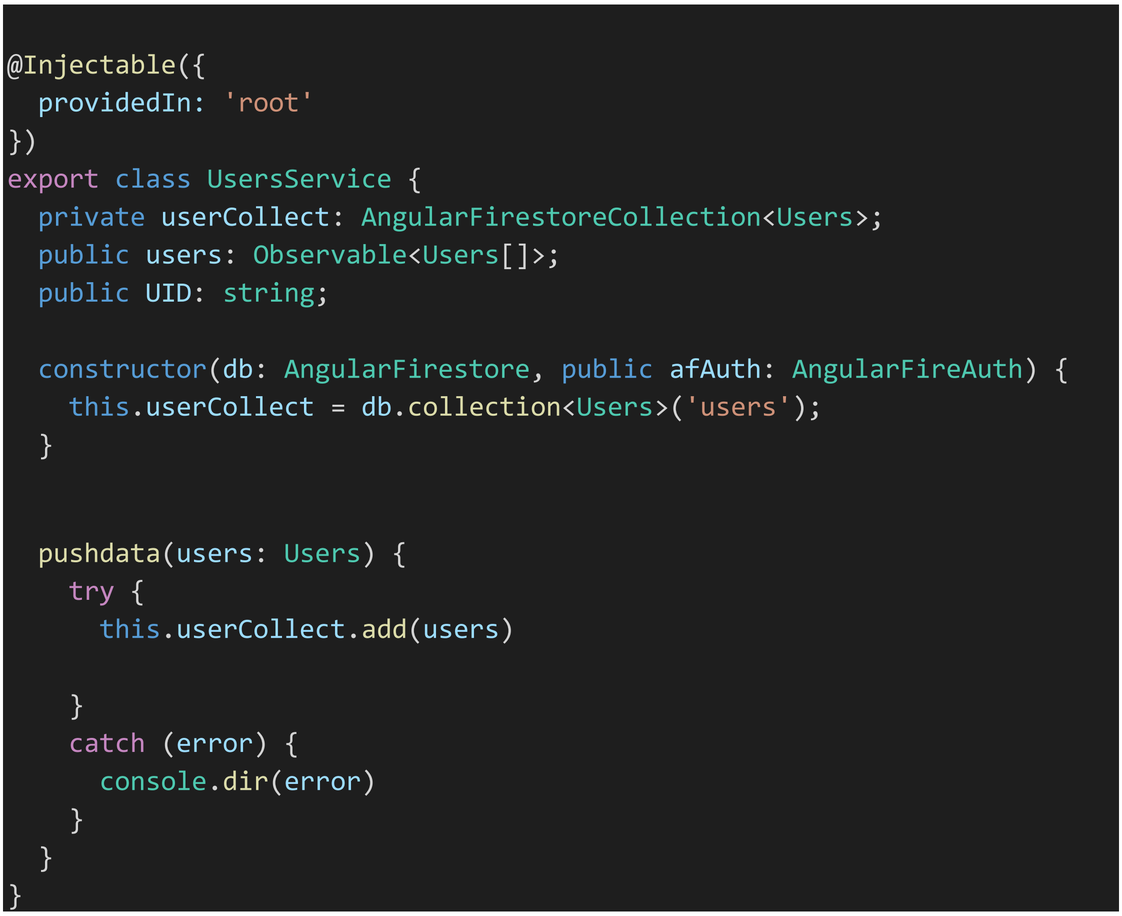


```
async confirm() {
    const { email, password, cpassword } = this
    if (password !== cpassword) {
      this.isHidden = !this.isHidden;
    }
    try {
      const res = await
this.afAuth.auth.createUserWithEmailAndPassword(email, password)
      this.router.navigate(['/identity'])
    }
    catch (error) {
      console.dir(error)
    }
  }
```

Register.page.ts

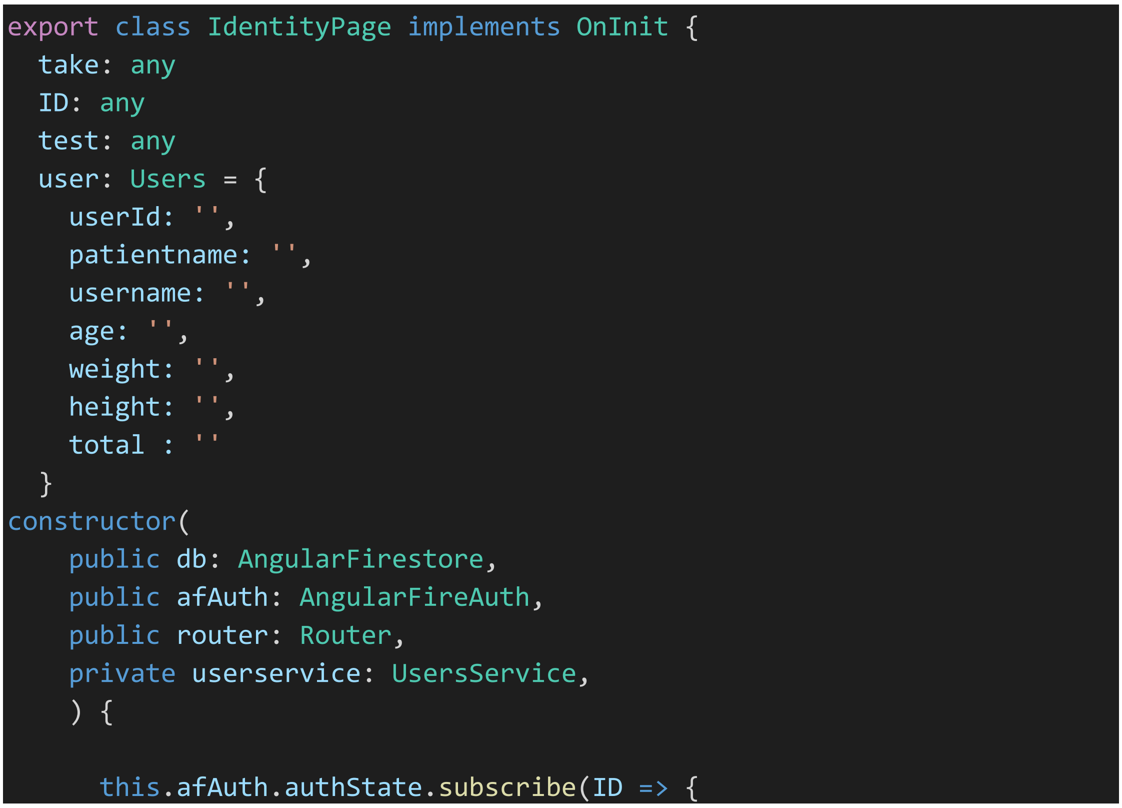


```
export interface Users {
  userId?: string
  patientname: string
  username: string
  age: string
  weight: any
  height: any
  total: any
}
```

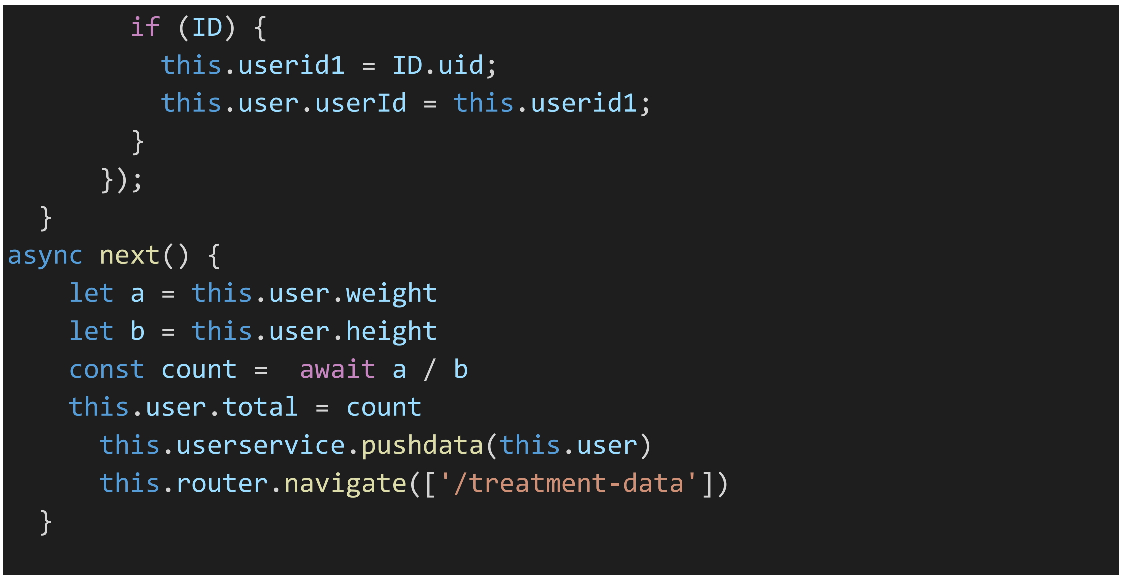


```
@Injectable({
  providedIn: 'root'
})
export class UsersService {
  private userCollect: AngularFirestoreCollection<Users>;
  public users: Observable<Users[]>;
  public UID: string;

  constructor(db: AngularFirestore, public afAuth: AngularFireAuth) {
    this.userCollect = db.collection<Users>('users');
  }


  pushdata(users: Users) {
    try {
      this.userCollect.add(users)

    }
    catch (error) {
      console.dir(error)
    }
  }
}
```

User.service.ts

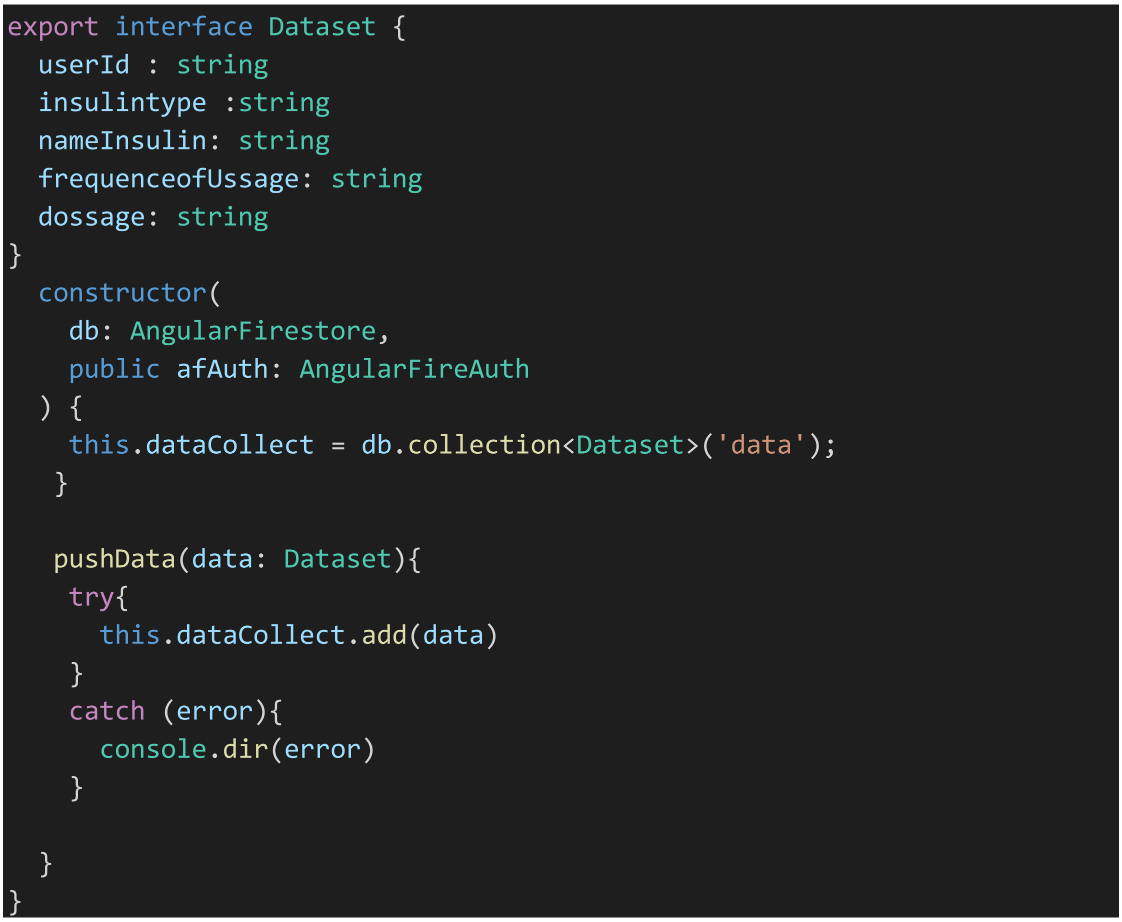


```
export class IdentityPage implements OnInit {
  take: any
  ID: any
  test: any
  user: Users = {
    userId: '',
    patientname: '',
    username: '',
    age: '',
    weight: '',
    height: '',
    total : ''
  }
constructor(
    public db: AngularFirestore,
    public afAuth: AngularFireAuth,
    public router: Router,
    private userservice: UsersService,
    ) {

      this.afAuth.authState.subscribe(ID => {
```



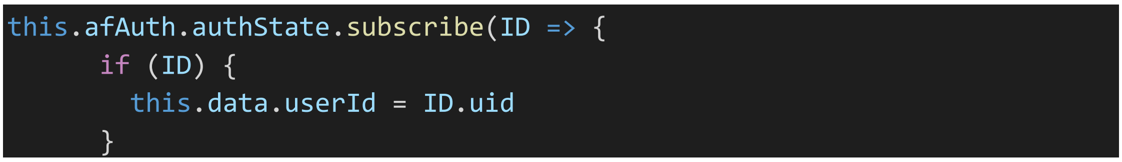
```
    if (ID) {
     this.userid1 = ID.uid;
     this.user.userId = this.userid1;
    }
   });
 }
async next() {
  let a = this.user.weight
  let b = this.user.height
  const count =  await a / b
  this.user.total = count
   this.userservice.pushdata(this.user)
   this.router.navigate(['/treatment-data'])
 }
```

Identity.page.ts



```
export interface Dataset {
 userId : string
 insulintype :string
 nameInsulin: string
 frequenceofUssage: string
 dossage: string
}
 constructor(
  db: AngularFirestore,
  public afAuth: AngularFireAuth
 ) {
  this.dataCollect = db.collection<Dataset>('data');
  }

  pushData(data: Dataset){
  try{
   this.dataCollect.add(data)
  }
  catch (error){
   console.dir(error)
  }

 }
}
```

Data.service.ts



```
this.afAuth.authState.subscribe(ID => {
   if (ID) {
    this.data.userId = ID.uid
   }
```

```
register() {
    this.dataservice.pushData(this.data);
    this.router.navigate(['/home-menu']);
}
```

Treatment-data.page.ts

Figure 5.10 Register Activity Code

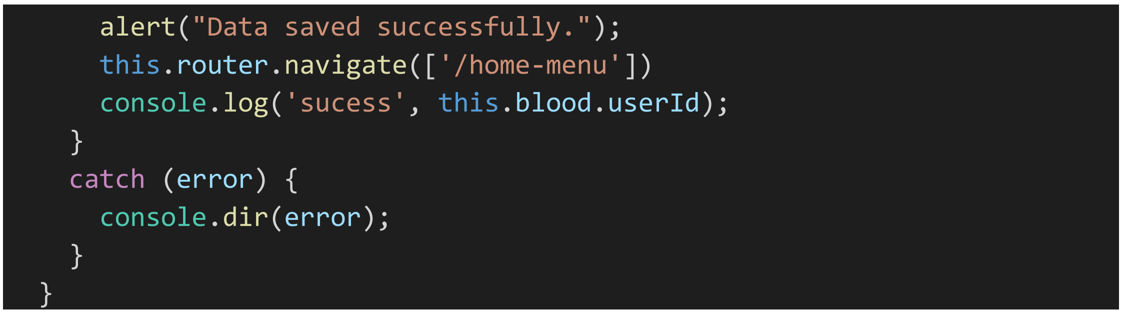
5.2.4 Input Blood Sugar Data Activity

The database of this app is using firestore database. All the data will handle by firestore. In blood sugar data the database will be push from ‘blood’ database. The code flow is. First the system will check if it is login or not from angularfireauth. If Login is true then it will take UserId. After that the string data that already inputted will be store together with userId string as an object.. The code is shown on Figure 5.17

```
export interface dataBlood {
  userId: string
  checkingtype: string
  date: Date
  result: string
}

  constructor(db: AngularFirestore,
    public afAuth: AngularFireAuth) {
    this.bloodDataCollect = db.collection<dataBlood>('blood');


  }
  pushData(data: dataBlood) {
    try {
      this.bloodDataCollect.add(data)
    }
    catch (error) {
      console.dir(error)
    }

  }
```

Blood.service.ts

```
add(){
    try {
      this.bloodservice.pushData(this.blood);
```



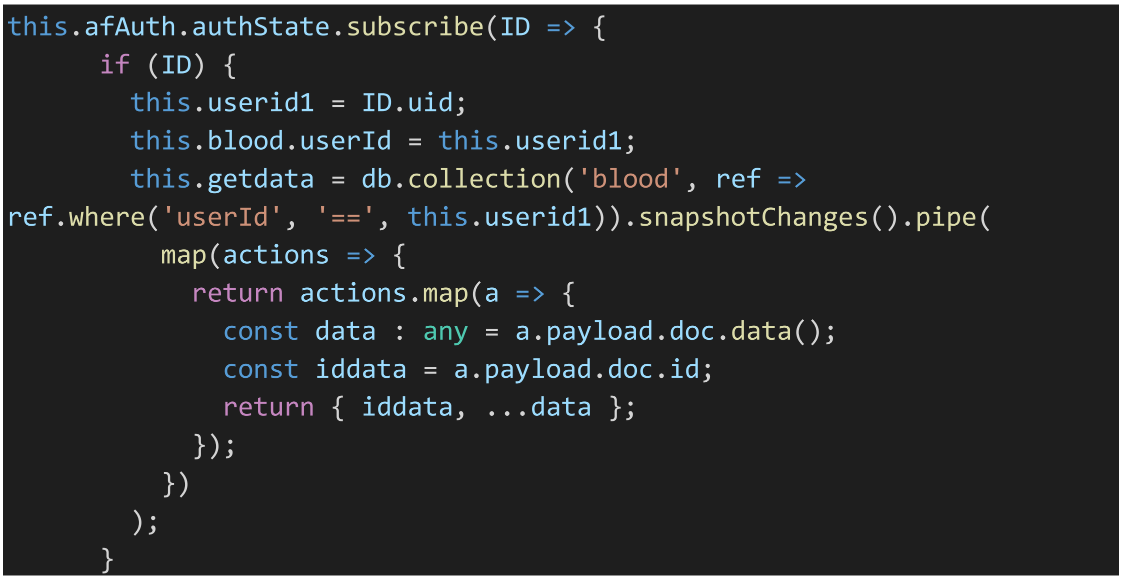
```
   alert("Data saved successfully.");
   this.router.navigate(['/home-menu'])
   console.log('sucess', this.blood.userId);
  }
  catch (error) {
   console.dir(error);
  }
 }
```

Bloodsugar.page.ts

Figure 5.11 Input Blood Sugar Data Activity Code

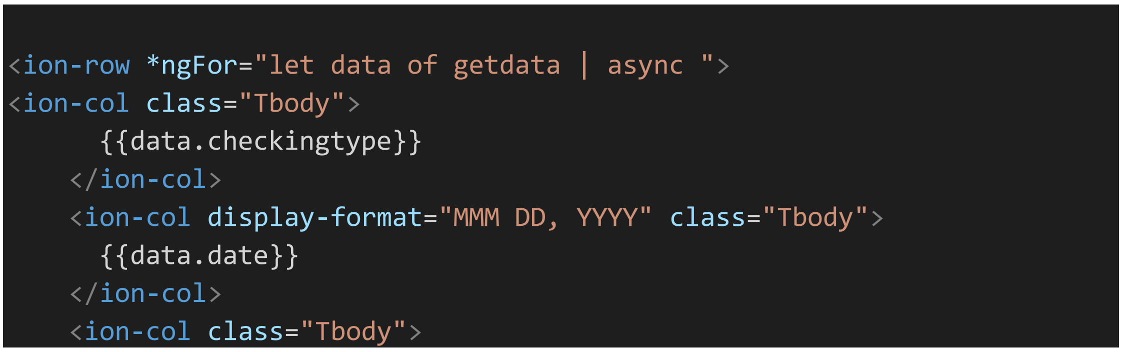
5.2.5 Read Blood Sugar Data as a Table

The blood sugar data that already input will show as a table. The system will read data where the userId of the data is same with login userId. After that all the data will be retrieve in the Blood-table.page.html page looping as table. The code is shown on Figure 5.18 and Figure 5.19.

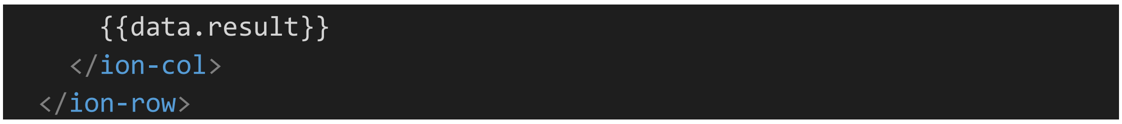


```
this.afAuth.authState.subscribe(ID => {
   if (ID) {
    this.userid1 = ID.uid;
    this.blood.userId = this.userid1;
    this.getdata = db.collection('blood', ref =>
ref.where('userId', '==', this.userid1)).snapshotChanges().pipe(
     map(actions => {
      return actions.map(a => {
       const data : any = a.payload.doc.data();
       const iddata = a.payload.doc.id;
       return { iddata, ...data };
      });
     })
    );
   }
```

Blood-table.page.ts



```

<ion-row *ngFor="let data of getdata | async ">
<ion-col class="Tbody">
   {{data.checkingtype}}
  </ion-col>
  <ion-col display-format="MMM DD, YYYY" class="Tbody">
   {{data.date}}
  </ion-col>
  <ion-col class="Tbody">
```



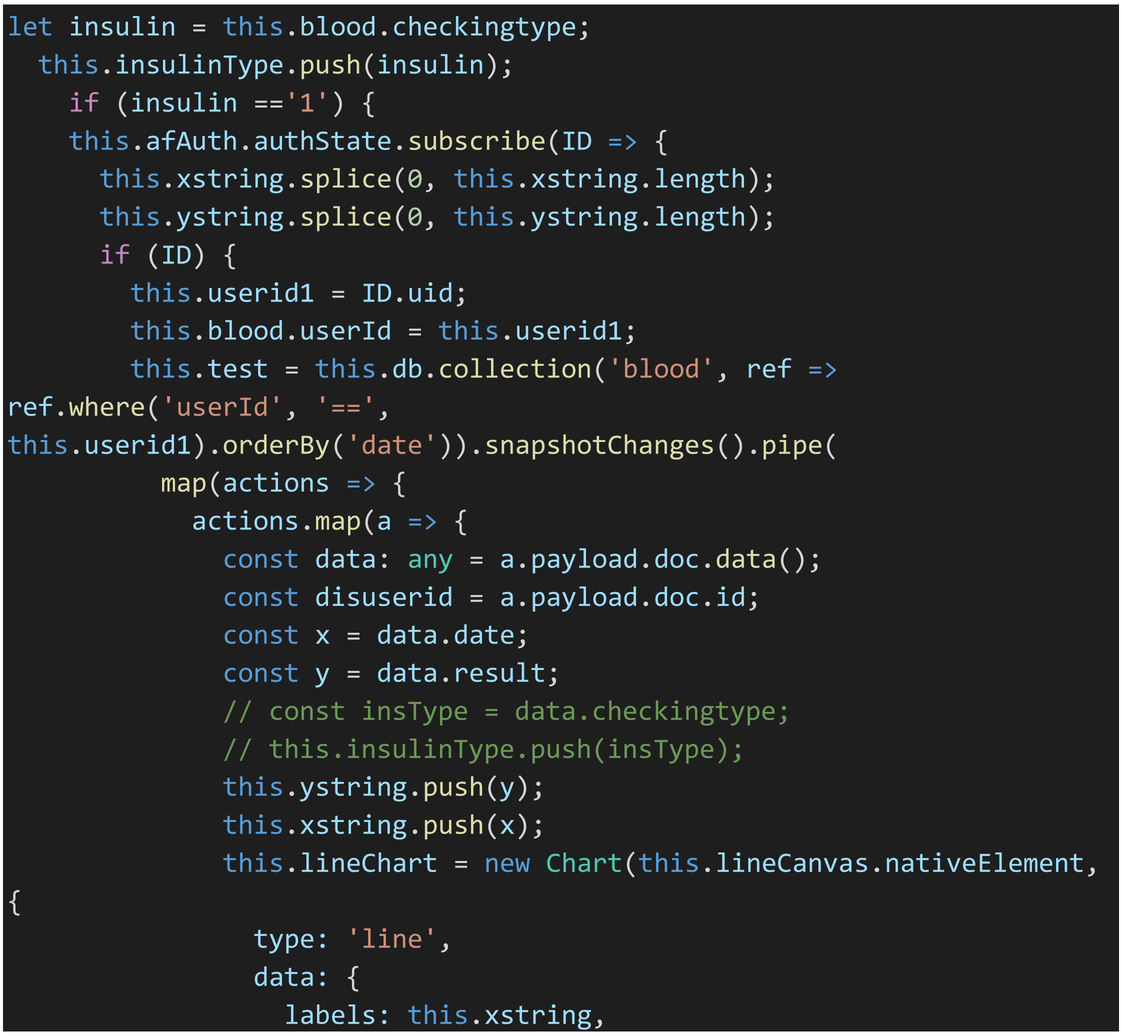
```
      {{data.result}}
    </ion-col>
  </ion-row>
```

Blood-table.page.html

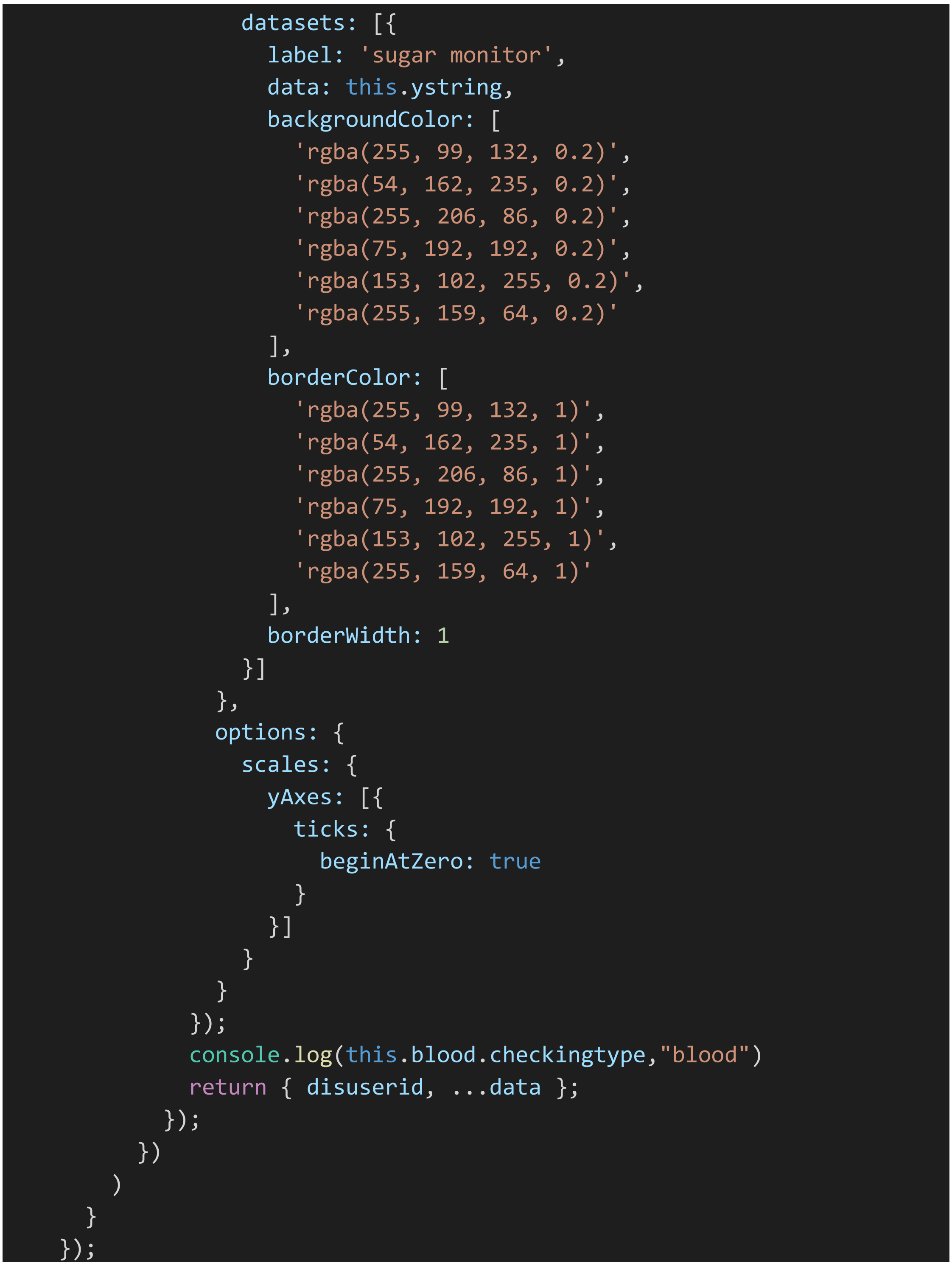
Figure 5.12 Read Blood Sugar Data as a Table Activity Code

5.2.6 Read Blood Sugar Data as a Chart using ChartJS Activity

The blood sugar data that already input will show as a table. The system will read data where the userId of the data is same with login userId. After that all the data will be retrieve in the Blood-table.page.html page looping as chart. the system will store only 'date' and 'result' data as the x and y. after that the ChartJS change the data into Line Chart and display it to the front page by calling the "new Chart(this.lineCanvas.nativeElement" function. The code is shown on Figure 5.21



```
let insulin = this.blood.checkingtype;
  this.insulinType.push(insulin);
    if (insulin =='1') {
    this.afAuth.authState.subscribe(ID => {
      this.xstring.splice(0, this.xstring.length);
      this.ystring.splice(0, this.ystring.length);
      if (ID) {
        this.userid1 = ID.uid;
        this.blood.userId = this.userid1;
        this.test = this.db.collection('blood', ref =>
ref.where('userId', '==',
this.userid1).orderBy('date')).snapshotChanges().pipe(
          map(actions => {
            actions.map(a => {
              const data: any = a.payload.doc.data();
              const disuserid = a.payload.doc.id;
              const x = data.date;
              const y = data.result;
              // const insType = data.checkingtype;
              // this.insulinType.push(insType);
              this.ystring.push(y);
              this.xstring.push(x);
              this.lineChart = new Chart(this.lineCanvas.nativeElement,
{
                type: 'line',
                data: {
                  labels: this.xstring,
```



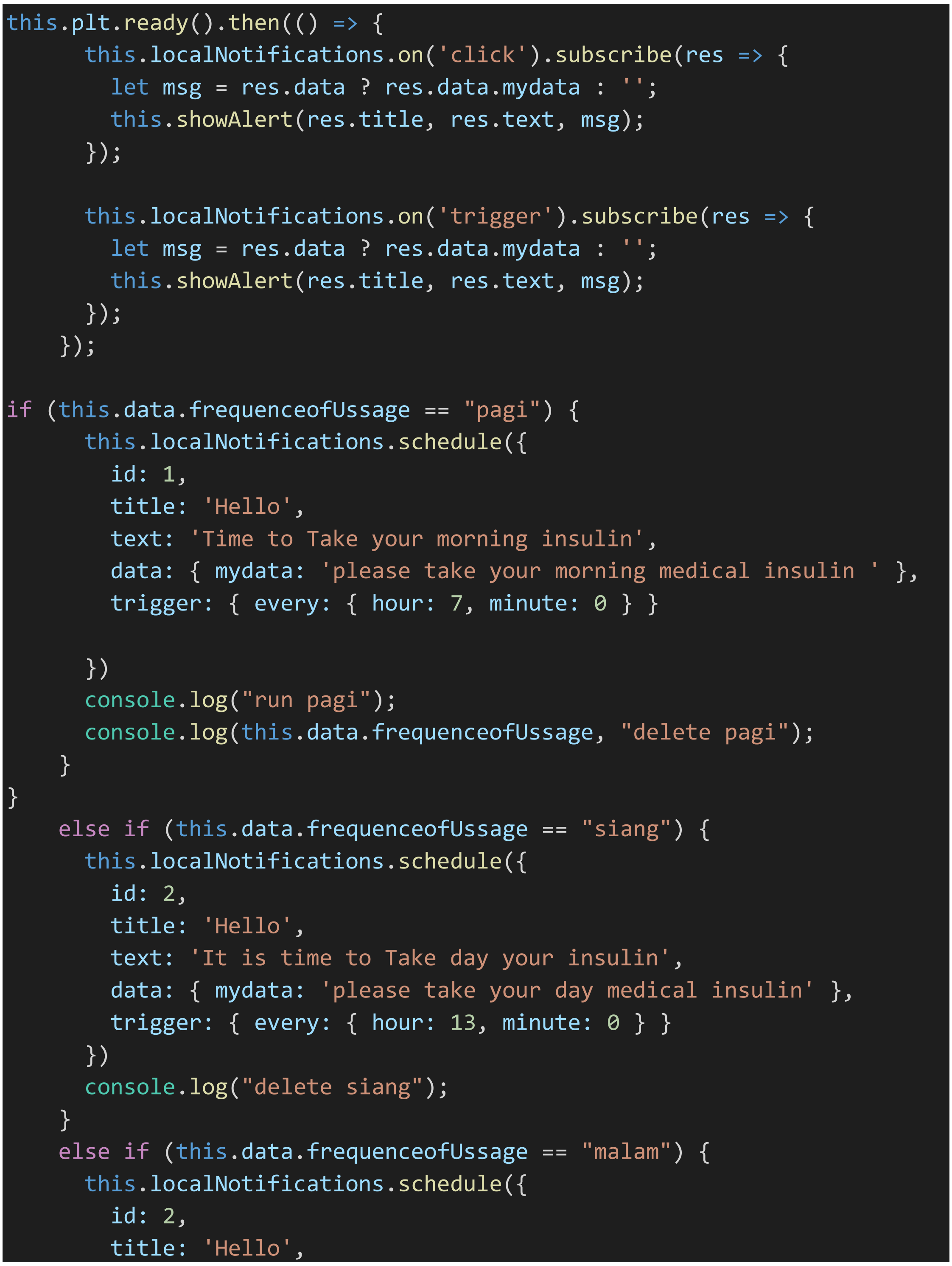
```
              datasets: [{
                label: 'sugar monitor',
                data: this.ystring,
                backgroundColor: [
                  'rgba(255, 99, 132, 0.2)',
                  'rgba(54, 162, 235, 0.2)',
                  'rgba(255, 206, 86, 0.2)',
                  'rgba(75, 192, 192, 0.2)',
                  'rgba(153, 102, 255, 0.2)',
                  'rgba(255, 159, 64, 0.2)'
                ],
                borderColor: [
                  'rgba(255, 99, 132, 1)',
                  'rgba(54, 162, 235, 1)',
                  'rgba(255, 206, 86, 1)',
                  'rgba(75, 192, 192, 1)',
                  'rgba(153, 102, 255, 1)',
                  'rgba(255, 159, 64, 1)'
                ],
                borderWidth: 1
              }]
            },
            options: {
              scales: {
                yAxes: [{
                  ticks: {
                    beginAtZero: true
                  }
                }]
              }
            }
          });
          console.log(this.blood.checkingtype,"blood")
          return { disuserid, ...data };
        });
      })
    )
  }
});
```

Sugarmonitor.page.ts

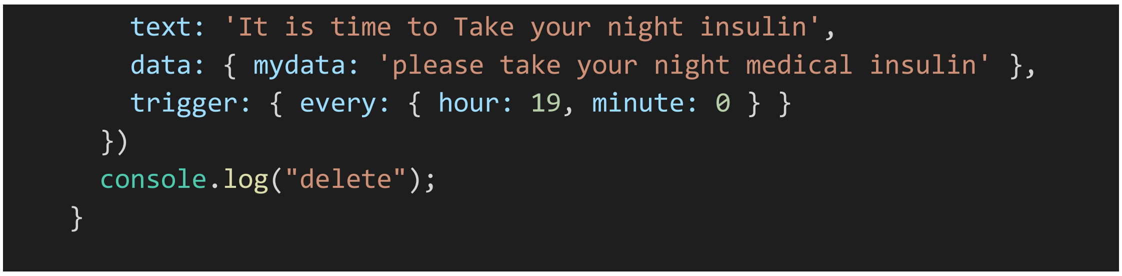
Figure 5.13 Read Blood Sugar Data as a Chart using ChartJS Activity Code

5.2.7 Local Notification Ionic

The notification data is handle by Ionic native which is new function from Ionic. The notification data divided by three part Morning, Afternoon and night. All the notification data is based on user input time data. The notification itself will be triggered at certain time based on the data. At the first line of code shown below is to run the notification. If the platform is ready the notification that already click will be shown also the notification that already get triggered by the time will be show base on the time.



```
this.plt.ready().then(() => {
      this.localNotifications.on('click').subscribe(res => {
        let msg = res.data ? res.data.mydata : '';
        this.showAlert(res.title, res.text, msg);
      });

      this.localNotifications.on('trigger').subscribe(res => {
        let msg = res.data ? res.data.mydata : '';
        this.showAlert(res.title, res.text, msg);
      });
    });

if (this.data.frequenceofUssage == "pagi") {
      this.localNotifications.schedule({
        id: 1,
        title: 'Hello',
        text: 'Time to Take your morning insulin',
        data: { mydata: 'please take your morning medical insulin ' },
        trigger: { every: { hour: 7, minute: 0 } }

      })
      console.log("run pagi");
      console.log(this.data.frequenceofUssage, "delete pagi");
    }
}
    else if (this.data.frequenceofUssage == "siang") {
      this.localNotifications.schedule({
        id: 2,
        title: 'Hello',
        text: 'It is time to Take day your insulin',
        data: { mydata: 'please take your day medical insulin' },
        trigger: { every: { hour: 13, minute: 0 } }
      })
      console.log("delete siang");
    }
    else if (this.data.frequenceofUssage == "malam") {
      this.localNotifications.schedule({
        id: 2,
        title: 'Hello',
```



```
        text: 'It is time to Take your night insulin',
        data: { mydata: 'please take your night medical insulin' },
        trigger: { every: { hour: 19, minute: 0 } }
      })
      console.log("delete");
    }
```

Treatment.data.ts

Figure 5.13 Local Notification Ionic Activity Code

CHAPTER VI

SYSTEM TESTING

The system testing chapter is described the testing of the program to evaluate all the features that has specified requirements. The system testing is to ensure if all the function of application will work as expected. There are two sections that will be described in this chapter such as Testing Environment and Testing Scenario.

6.1 Testing Environment

Testing environment is used to test Medical Insulin Android application features and measure the capability of the program to run in order and properly within the scope and limitation system. The testing will be done by all possibilities scenarios in order to get great testing result.

The application will be tested by the following hardware and software specification as follows:

1. Oppo F7, Oreo 8.1 OS (Android API 21)

6.2 Software testing

Software testing is a process of confirming the software has been manufactured by the developer that achieve the quality product and assure manufactured product is working according to the specification and satisfying the user needs[3]. The software testing that been used in this thesis is black box testing. Black Box Testing is a testing technique without reference to the internal structure of the component or system. In Black Box Testing it is not necessary for a tester to have good programming knowledge, since it only examines the fundamental aspects of the system without going into detail. One of the type of black box testing that has been used on this thesis is using user acceptance test as the functional test for the application[2]. This testing is the last step before the application goes to live. The user acceptance test makes sure the application is able to fulfill the user needs and also establishing confidence in the user that the application is fit for purpose.

6.3 Usability Testing

Testing scenario describes the activities that have to be tested to ensure the system and application work in a proper way. The testing scenario of Medical Insulin application is divided into several sections which are:

6.3.1 Login Section

The first page that the user find after splash page. The user needs to fill in the field in order to access the homepage.

Table 6.1 Sign-In Section Testing Scenario

No	Scenario	Expected Result	Result
1	Valid e-mail address and password	Directly to the home page	As expected
2	Invalid e-mail address and password	Display invalid input message	As expected
3	Tap Register button	Directly to the sign up page	As expected

6.3.2 Register Section

After user tap register text on login page, the application will be directly display the register page. After that, the user input all the data requirements on the fields.

Table 6.2 Sign-Up Section Testing Scenario

No	Scenario	Expected Result	Result
1	Fill all the required field	Update the database, redirect to the homepage	As expected
2	Fields are not completely filled	Display error message	As expected

6.3.3 Insulin Reminder Section

This testing scenario is expected to display reminder schedule for user to take the insulin.

Table 6.3 Insulin Reminder Section Testing Scenario

No	Scenario	Expected Result	Result
1	Display the time for take insulin	Retrieve the data from the database and display it	As expected

6.3.4 Blood Sugar Monitor Section

This testing scenario is use to testing the blood sugar monitor page that display the graphic of blood sugar. Start from input choose the insulin type. This section is expected to display the blood sugar graphic based on the input of insulin type.

Table 6.4 Blood Sugar Monitor Section Testing Scenario

No	Scenario	Expected Result	Result

1	Tap Insulin type option	Display all the insulin type option	As expected
2	Tap check button	Retrieve the data from the database and display graphic based on insulin type	As expected

6.3.5 Blood Sugar Data Section

This testing scenario is use to testing the blood sugar data page. Start from input insulin type, date and result of insulin type. This section is expected to restore the data from user input that will be display on blood sugar monitor and blood sugar table.

Table 6.5 Blood Sugar Data Section Testing Scenario

No	**Scenario**	**Expected Result**	**Result**
1	Tap Checking's type option	Display the insulin type option	As expected
2	Input date	Display date	As expected
3	Input result	Display the result of	As expected
4	Tap add button	Store the data into database	As expected

6.3.6 Recommendation for Healthy Lifestyle Section

In this section, the program is expected to display the recommendation for healthy lifestyle based on user BMI.

Table 6.6 Recommendation for Healthy Lifestyle Section Testing Scenario

No	Scenario	Expected Result	Result
1	Load tips healthy lifestyle based on BMI User	Retrieve all the data from database and display it	As expected

6.3.7 Blood Sugar Table Section

This testing scenario is use to blood sugar table. The program is expected to display the blood sugar monitor data on the table.

Table 6.7 Blood Sugar Table Section Testing Scenario

No	Scenario	Expected Result	Result
1	Tap top up balance	Display the top up balance page	As expected
2	User input amount and tap top up balance	Store the data to database then the balance is updated	As expected

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

There are several conclusions that could be obtained from thesis, which are:

1) This application could become one of prevention the diabetes patient to relapsed.

2) This application helps users to take insulin on the right time.

3) This application diabetes patient to control the blood sugar level.

7.2 Future Work

There are several features that could be improved in this research for the further development such as:

- Better User Interface (UI) of the application

  Improving the user interface (UI) of the application to be easy-friendly style, so that the user will be easier to understand each feature of this application.

- Create the detail and finishing of the application

  Improving the detail and the finishing of the application to create better UI and UX in order to developed in the daily life.

REFERENCES

[1] "RAD Model," [Online].

[2] Manish Kumar, Santosh Kumar Singh, Dr. R. K. Dwivedi, "A Comparative Study of Black Box Testing and White Box," *International Journal of Advance Research in Computer Science and Management Studies,* pp. 32-44, 2015.

[3] R. Yackel, "Functional Testing Types – 25 Best Practices, Tips & More!," 22 February 2017. [Online]. Available: https://www.qasymphony.com/blog/functional-testing-types/.

[4] A. Trikkalinou, A. K Papazafiropoulou and A. Melidonis, "Type 2 diabetes and quality of life," *Worlf J Diabetes,* pp. 120-129, 2017.

[5] TUE Physician Guidelines Medical Information to Support the Decisions of TUE Committees Diabetes Mellitus , "Diabetes Melitus," *World Anti-Doping Program ,* vol. 3, 2015.

[6] "Diabetes M".

[7] "Diabetes Pal".

[8] J. Kelly, "Diabetes," *National Diabetes Education Program CDC National Center for Chronic Disease Prevention and Health Promotion.*

[9] American Diabetes Association, "Diagnosis and Classification of Diabetes Mellitus," *Diabetes Care,* vol. 37, p. 81, 2014.

[10] Standards of Medical Care, American Diabetes Association, 2018.

[11] Habtamu Wondifraw Baynest, "Classification, Pathophysiology, Diagnosis and Management of Diabetes Mellitus," *Journal of Diabetes and Metabolism,* vol. 6, no. 5, 2015.

[12] Eric Steen, Benjamin M. Terry, Enrique J. Rivera, Jennifer L. Cannon, Thomas R. Neely, Rose Tavares, X. Julia Xu, Jack R. Wands and Suzanne M. de la Monte, "Impaired insulin and insulin-like growth factor expression and signaling mechanisms in Alzheimer's disease – is this type 3 diabetes?," *Journal of Alzheimer's Disease 7 ,* p. 63, 2005.

[13] Suzanne M. de la Monte and Jack R. Wands, "Alzheimer's Disease Is Type 3 Diabetes–Evidence Reviewed," *Journal of Diabetes Science Technology,* 2008.

[14] Lindy Kahanovitz, Patrick M. Sluss and Steven J. Russell, "Type 1 Diabetes – A Clinical Perspective," *HHS Public Access,* 2017.

[15] Abdulfatai B. Olokoba, Olusegun A. Obateru and Lateefat B. Olokoba, "Type 2 Diabetes Mellitus: A Review of Current Trends," *Oman Medical Journal,* 2012.

[16] S. H. Salomon, "Everyday Health," [Online]. Available: https://www.everydayhealth.com/diabetes/alzheimers-disease-actually-type-diabetes/.

[17] Kohei KAKU, "Pathophysiology of Type 2 Diabetes and Its Treatment Policy," *Research and Reviews.*

[18] G. Wilcox, "Insulin and insulin resistance.," *The clinical biochemist reviews,* 2005.

[19] "Insulin Resistance".

[20] A. R. Sinaiko, "Insulin Resistance," *HHS Public Access.*

[21] A. V. a. S. R. Joseph Fomusi Ndisang, "Insulin Resistance, Type 1 and Type 2 Diabetes, and Related Complications 2017," *Journal of Diabetes Research,* 2017.