

**myDialysis:
DIALYSIS CENTER MOBILE APPLICATION**

NUR KHAIRUN SOFIA BINTI AMAN

**Bachelor of Computer Science (Software
Engineering) With Honours**

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NUR KHAIRUN SOFIA BINTI AMAN

Date of Birth : 03 JUNE 2000

Title : myDialysis: DIALYSIS CENTER APPLICATION

Academic Session : 2021/2022 SEMESTER 2

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access
(Full Text)

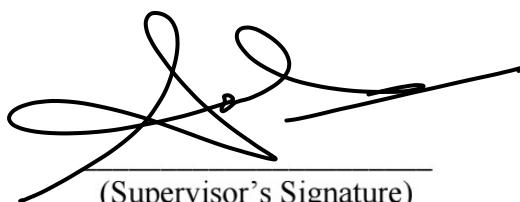
I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)



(Supervisor's Signature)

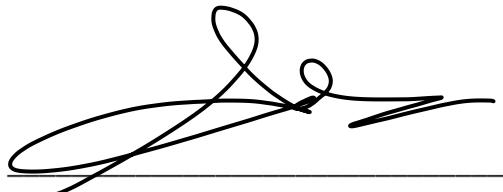
____000603-12-0328____
New IC/Passport Number
Date: 28 February 2023

DR. SALWANA MOHAMAD @ ASMARA
Name of Supervisor
Date: 1/3/2023

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor's Degree in Computer Science (Software Engineering) with Honours.

A handwritten signature in black ink, appearing to read "DR. SALWANA MOHAMAD", is written over a horizontal line.

(Supervisor's Signature)

Full Name : DR. SALWANA MOHAMAD @ ASMARA

Position : SENIOR LECTURER

Date : 1/3/2023



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to read 'Nur Khairun Sofia Binti Aman', is placed over a horizontal line.

(Student's Signature)

Full Name : NUR KHAIRUN SOFIA BINTI AMAN

ID Number : CB19115

Date : 28 February 2023

**myDialysis:
DIALYSIS CENTER MOBILE APPLICATION**

NUR KHAIRUN SOFIA BINTI AMAN

Thesis submitted in fulfilment of the requirements
for the award of the degree of
Bachelor's Degree in Computer Science (Software Engineering) with Honours

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

February 2023

ACKNOWLEDGEMENTS

First and foremost, Alhamdulillah all praise to almighty Allah S.W.T that I am really grateful for the good health and well-being that were necessary to complete my dialysis mobile application thesis.

I wish to express my sincere thanks and deepest appreciation to my project supervisor, Dr. Salwana Mohamad @ Asmara for her guidance, encouragement, and knowledge throughout the project's progress. I gained a lot of new knowledge within the development process of the project. Due to this, I have managed to finish this project's thesis this semester successfully.

I would like to take this opportunity as well to express gratitude to my parents for their unceasing encouragement, support, and attention. I am also grateful and thankful to my Undergraduate Project evaluators for their comments and guidance throughout assessing this project. In addition, I place on record, my sense of gratitude to the one and all, who directly and indirectly, have lent their hand and energy in this venture.

Thank you to all of them, may Allah bless you all.

ABSTRAK

Aplikasi mudah alih myDialysis adalah alat atau aplikasi perisian yang digunakan oleh komuniti pusat dialisis iaitu kakitangan dialisis@jururawat untuk menguruskan masa slot pesakit, rekod rawatan, pembayaran dan direktori pusat dialisis lain, pesakit pusat dialisis untuk menguruskan masa slot mereka, melihat rawatan merekod, membuat pembayaran, melihat direktori pusat dialisis lain dan menguruskan temujanji dengan pakar, dan kakitangan hospital@jururawat yang ditugaskan kepada pesakit untuk menguruskan temujanji pesakit dengan pakar. Kaedah sedia ada untuk menguruskan semua tugasan dilakukan secara manual yang tidak sistematik dan serta tidak selamat. Dengan penggunaan aplikasi ini, operasi tugas dapat dilakukan secara sistematik dan semua data seperti rekod rawatan pesakit dapat disimpan dengan selamat.

ABSTRACT

mydialysis mobile application is a tool that is used by the dialysis center community which are the dialysis staff@nurse to manage patients' slot time, treatment record, payment, and other dialysis center directories, the dialysis center's patients to manage their slot time, view treatment record, make payment, view other dialysis center directories and manage appointment with the specialist, and the hospital staff@nurse which were assigned to the patient to manage patient's appointment with the specialist. The existing method to manage all the tasks was done manually which is not systematic and secure. With the use of this application, the task operations can be performed systematically and all the data such as patients' treatment records could be stored safely.

TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS	x
ABSTRAK	xi
ABSTRACT	xii
LIST OF TABLES	xvi
LIST OF FIGURES	xvii
LIST OF ABBREVIATIONS	xxi
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope	4
1.5 Significance of the project	4
1.6 Report organization	5
CHAPTER 2 LITERATURE REVIEW	6
2.1 Introduction	6
2.2 Review of existing system	6
2.2.1 Application 1 - Fresenius myCompanion App	6
2.2.2 Application 2 - NephroPlus App	8
2.2.3 Application 3 - Freedom Dialysis App	10
2.3 Summary	12
CHAPTER 3 METHODOLOGY	14
3.1 Introduction	14
3.2 Project Management Framework	14
3.3 Project Requirement	17
3.3.1 Functional requirements:	17
3.3.2 Non-functional requirements:	18
3.3.3 Constraints and limitations	19

3.4 Proposed Design	20
3.4.1 Context Diagram	20
3.4.2 Data Flow Diagram	21
3.4.3 Use Case Diagram	22
3.4.4 Use Case Description	23
3.4.5 Data Design	38
3.4.6 Data Dictionary	38
3.5 Proof of Initial Concept	45
3.5.1 Sign Up and Login Interface	45
3.5.2 Manage Profile Interface	46
3.5.3 Manage Appointment Interface	49
3.5.4 Manage Slot Time	53
3.5.5 Manage Treatment Data	57
3.5.6 Manage Directory	59
3.5.7 Manage Payment	61
3.6 Testing Plan	64
3.7 Potential Use of Proposed Solution	66
CHAPTER 4 RESULTS AND DISCUSSION	67
4.1 Introduction	67
4.2 Development Tools	67
4.3 Implementation Process	67
4.3.1 Database Implementation	68
4.3.2 Interface Development	79
4.3.2.1 Home Page	79
4.3.2.2 Profile Page	85
4.3.2.3 Dialysis Center Directory Page	87
4.3.2.4 Treatment Record Page	90
4.3.2.5 Slot Time Page	93
4.3.2.6 Appointment Page	96
4.3.2.7 Payment Page	99
4.4 Application Testing	102
4.4.1 Proof of Testing	105

4.5 Result and Discussion	106
CHAPTER 5 CONCLUSION	110
 5.1 Introduction	110
 5.2 Limitation and Constraint	111
 5.3 Future Work	112
APPENDIX A	114

LIST OF TABLES

Table 2. 1 Comparison between the three existing app	12
Table 3. 1 Description and actor involved in each module.	22
Table 3. 2 Manage Profile Use Case Description	23
Table 3. 3 Manage Appointment Use Case Description	25
Table 3. 4 Manage Slot Time Use Case Description	28
Table 3. 5 Manage Treatment Data Use Case Description	30
Table 3. 6 Manage Directory Use Case Description	33
Table 3. 7 Manage Payment Use Case Description	35
Table 3. 8 Dialysis Staff Data Dictionary	38
Table 3. 9 Hospital Staff Data Dictionary	39
Table 3. 10 Patient Data Dictionary	40
Table 3. 11 Slot Time Data Dictionary	40
Table 3. 12 Treatment Record Data Dictionary	41
Table 3. 13 Appointment Data Dictionary	42
Table 3. 14 Bill Data Dictionary	43
Table 3. 15 Dialysis Center Data Dictionary	43
Table 3. 16 Shows the UAT Form Example	64

LIST OF FIGURES

Figure 2. 1 Sample UI 1 of Fresenius myCompanion App	7
Figure 2. 2 Sample UI 2 of Fresenius myCompanion App	8
Figure 2. 3 Sample UI 1 of NephroPlus App	9
Figure 2. 4 Sample UI 2 of NephroPlus App	9
Figure 2. 5 Sample UI 1 of Freedom Dialysis App	11
Figure 2. 6 Sample UI 2 of Freedom Dialysis App	11
Figure 3. 1 Workflow of RAD	15
Figure 3. 2 Context Diagram	20
Figure 3. 3 Data Flow Diagram	21
Figure 3. 4 Use Case Diagram	22
Figure 3. 5 Entity Relationship Diagram	38
Figure 3. 6 Register Interface	46
Figure 3. 7 Login Interface	46
Figure 3. 8 Patient Homepage Interface	47
Figure 3. 9 Profile Page Interface	47
Figure 3. 10 Edit Profile Interface	48
Figure 3. 11 View Profile Interface	48
Figure 3. 12 Patient View Appointment Interface	50
Figure 3. 13 Reschedule Appointment Form	50
Figure 3. 14 Hospital Staff Appointment Interface	51
Figure 3. 15 Add Appointment Interface	51
Figure 3. 16 Reschedule Interface	52
Figure 3. 17 Slot Time Interface	54
Figure 3. 18 Request Slot Time Interface	54
Figure 3. 19 View Slot Time Interface	55
Figure 3. 20 Add Slot Time Interface	55
Figure 3. 21 Request Slot Time Interface	56

Figure 3. 22 Patient Treatment Record Interface	57
Figure 3. 23 Patient Treatment Record Interface	57
Figure 3. 24 Add Treatment Record	58
Figure 3. 25 Edit Treatment Record Interface	58
Figure 3. 26 Directory Interface	59
Figure 3. 27 Manage Directory Interface	59
Figure 3. 28 Add Directory Interface	60
Figure 3. 29 Payment Page Interface	61
Figure 3. 30 PayBill Interface	61
Figure 3. 31 Staff Payment Page Interface	62
Figure 3. 32 Add Bill Interface	62
Figure 3. 33 View Payment Interface	63

Figure 4. 1 <i>pubspec.yaml</i> file	69
Figure 4. 2 Database Helper (initialize database)	70
Figure 4. 3 Database Helper (creating database table)	71
Figure 4. 4 Database Helper (CRUD Operation)	72
Figure 4. 5 Model Class	73
Figure 4. 6 User Model Class Coding	74
Figure 4. 7 Slot Model Class Coding	74
Figure 4. 8 Treatment Record Model Class Coding	75
Figure 4. 9 Appointment Model Class Coding	76
Figure 4. 10 Payment Model Class Coding	77
Figure 4. 11 Directory Model Class Coding	78
Figure 4. 12 Home Page Coding	79
Figure 4. 13 Patient Home Page Interface	80
Figure 4. 14 Dialysis Staff Home Page Coding	81
Figure 4. 15 Dialysis Staff Home Page Interface	82
Figure 4. 16 Hospital Staff Home Page Coding	83
Figure 4. 17 Hospital Staff Home Page Interface	84
Figure 4. 18 Profile Page Coding	85
Figure 4. 19 Profile Page Interface	86
Figure 4. 20 Directory Coding for Patient and Hospital Staff	87
Figure 4. 21 Add Directory Function	88
Figure 4. 22 Dialysis Staff Directory Coding	88
Figure 4. 23 Directory Page Interfaces	89
Figure 4. 24 Treatment Record Page Coding	90
Figure 4. 25 Add Treatment Record Function	91
Figure 4. 26 Treatment Record Page Interfaces	92
Figure 4. 27 Add Slot Time Function	93
Figure 4. 28 Slot Time Page Coding	94
Figure 4. 29 Slot Time Page Interfaces	95
Figure 4. 30 Add Appointment Function	96
Figure 4. 31 Appointment Page Coding	97
Figure 4. 32 Appointment Page Interfaces	98
Figure 4. 33 Add Bill Function	99

Figure 4. 34 Payment Page Coding	100
Figure 4. 35 Payment Page Interfaces	101
Figure 4. 36 UAT Form (Patient Section)	102
Figure 4. 37 UAT Form (Dialysis Staff Section)	103
Figure 4. 38 UAT Form (Hospital Staff Section)	104
Figure 4. 39 User Testing on Sign Up Function	105
Figure 4. 40 User Testing on Login Function	105
Figure 4. 41 User Testing as Patient	105
Figure 4. 42 User Testing as Dialysis Staff	105
Figure 4. 43 User Testing as Hospital Staff	105
Figure 4. 44 Name of the UAT respondents	106
Figure 4. 45 UAT Result Graph (Patient Section)	107
Figure 4. 46 UAT Result Graph (Dialysis Staff Section)	108
Figure 4. 47 UAT Result Graph (Hospital Staff Section)	109

LIST OF ABBREVIATIONS

UI	User Interface
RAD	Rapid Application Development
UAT	User Acceptance Testing
UML	Unified Modelling Language
UAT	User Acceptance Test

CHAPTER 1

INTRODUCTION

1.1 Introduction

Dialysis treatment is one of the most important and high-demand treatments in Malaysia since its rate in all states in Malaysia has exceeded 100 per million state population. The dialysis treatment has shown overnight growth and in the past 10 years, the number of dialysis patients has tripled, as stated by Datuk Dr. Noor Hisham Abdullah (Medical Development Division, Ministry of Health Malaysia, 2010, 10). myDialysis is a mobile application that helps dialysis patients to organize their dialysis treatment time slot and review their treatment records. Dialysis patients will also be able to view their monthly appointment with the doctor. This app will be used by dialysis center staff as well. As for dialysis center staff, they are able to key in patients' treatment records and manage patients' treatment time slots.

With the modernization and technology, we have today, most dialysis centers still implement the traditional way to record and store patients' data which is by using sheets and files. This will lead to a few problems such as missing files, misplaced files, time-consuming searching for patients' files, and lack of file storing place. Furthermore, these physical files are easily destroyed by unavoidable casualties such as fire, floods, and earthquakes. myDialysis will be convenient for the dialysis center staff and patients as it helps to store treatment data more securely and systematized so that the data can be reached at speed and with less time consumption.

myDialysis app is aimed to make it convenient for dialysis center staff, patients, and hospital staff to record, store and review patients' treatment data. This app also will have information about other dialysis centers such as their location and contact number for patients' reference.

1.2 Problem Statement

Dialysis center patients face difficulties when they want to review their treatment records. There is no platform provided for the patients to review their treatment records and they did not have a copy of their treatment records. If they need it, they have to contact the dialysis center which is very inconvenient for them and the dialysis center staff as they have to search for the file while patients will have to wait for them to give the information. The treatment slot is arranged by the dialysis staff and patients have to do the treatment 3 times a week. However, most patients are employees and sometimes they have to change their treatment slot due to their work time that requires them to go outstation or work course. The dialysis center allows them to change their treatment slot for a particular day but they have to inform and book early for that slot which is also in a traditional and manual way where patients have to contact the center to ask for an available slot for that day. Moreover, some patients might need to do the treatment at other dialysis centers but there is also no platform where patients can get the other dialysis centers' information easily. On top of that, dialysis patients usually have an appointment with specialist doctors every 3 months to monitor their blood and other related health issues. Nonetheless, there is no platform for the patient manage their appointment with the specialist.

1.3 Objectives

1. To study the existing application and determine how to develop dialysis management application.
2. To design and develop an application that helps dialysis patients to view their treatment records, manage their dialysis slot time and organize their specialist appointment conveniently.
3. To evaluate the effectiveness and functionality of the application in solving the problem statement.

1.4 Scope

1. User scope
 - a. Dialysis patients
 - b. Dialysis center staffs@nurse
 - c. Hospital staffs@nurse

Area

- a. Malaysia only

2. System scope

Cover modules:

- a. Manage slot time
- b. Record treatment data
- c. Manage appointments
- d. Manage directory
- e. Manage user profile
- f. Manage payment

3. Development scope

- a. Contains multimedia elements such as graphics and texts.
- b. Using Android Studio, SQLite, Dart Programming Language, Flutter Framework

1.5 Significance of the project

myDialysis is beneficial to the Dialysis Center community as it is a platform for making it convenient to manage simple things such as adding and changing treatment time slot. Hospital staff also will be able to set patients appointment with specialist easily. Moreover, patients can make payment for their treatment at ease and online. In fact, the patients' families will also become aware of the patients' condition by viewing at the patients treatment records.

1.6 Report organization

This thesis consists of five chapters of the report. Chapter 1 explained the introduction of the project such as problem statements, objectives of the project, project scope, and significance of the project. Chapter 2 discuss the literature review on three existing and similar application that consists of a description and comparison of the application. Chapter 3, focuses on the methodology that will be used for this project development. Chapter 4, briefly describes the implementation, and all results obtained during this project development. Chapter 5 will sum up the results obtained from this project and also explain the limitations, and constraints of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

There is no specific dialysis application in Malaysia that helps patients to rearrange their treatment slots and staff to record patients' treatment data. As mentioned in chapter 1, the process to record patients' treatment data and health data is still applying the traditional method which is recording it manually in files. Nevertheless, Malaysia has one application that helps dialysis patients to discover the dialysis center all over Malaysia, called *Pusat Dialisis Malaysia* and it involves both private center and free center. Dialysis centers' founders can register their location and related information on the app to help the patients find them easily (*Pusat Dialisis Malaysia - Apps on Google Play*, 2021). This is a good initiative, but it does not efficient for the patients as they need to install another application to manage their dialysis treatment data, slot time and payment. Therefore, implementing the features which helps patients to view their treatment record, manage treatment slot time, manage payment, and view other dialysis center directory in one application is an effective solution for this project.

2.2 Review of existing system

Dialysis center in Malaysia does not have a well-organized existing system yet. However, there are three existing and similar apps to this app that will be developed. The three existing apps that will be reviewed are Fresenius myCompanion, NephroPlus, and Freedom Dialysis and these apps have their own features, advantages, and disadvantages.

2.2.1 Application 1 - Fresenius myCompanion App

(*Fresenius MyCompanion - Apps on Google Play*, n.d.)

Fresenius myCompanion is an app developed by Fresenius Medical Care Deutschland GmbH and it was first released on 22nd December 2021. This app is mainly

for hemodialysis patients in NephroCare clinics only. Patients will be able to view their treatment summary, health info, and any medication needed to be taken. Furthermore, this app also includes a feature that provides patients with useful tips for dialysis patients and nutrient-rich food recipes. Authorized patients can get access to this app easily with the help of the clinic staff. With the latest update on this app, patients can easily view their treatment data on the dashboard. Patients also will be able to customize the focus area on their dashboard. This app has a simple and minimalist UI design which is very suitable for dialysis patients that consists of different age levels. The disadvantage of this app is there is no feature for patients to rearrange their treatment time slot and appointment with specialists. Moreover, it does not provide patients with other dialysis centers' information for patients' treatment transit, and it is not a centralized database that other dialysis centers could access to patients' treatment data.

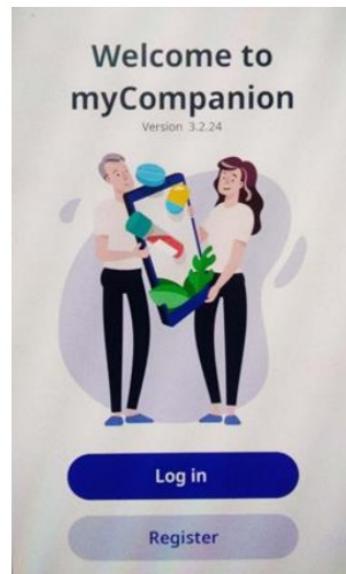


Figure 2. 1 Sample UI 1 of Fresenius myCompanion App



Figure 2. 2 Sample UI 2 of Fresenius myCompanion App

2.2.2 Application 2 - NephroPlus App

(*NephroPlus - Apps on Google Play*, n.d.)

NephroPlus is an application to help improve the quality of care in hemodialysis patients. It was released on 26th October 2021 by India's largest network of dialysis centers, NephroPlus. It provides features for patients to book their dialysis session slots, access their lab test results, and get more information about kidney disease and dialysis care. This app also offers patients to consult with nephrologists or dieticians via video consults. Moreover, patients can pay for their treatment on this app so that they do not have to bring cash to the dialysis center and the transaction history can be viewed by patients. The patients also are able to download and share copies of the bills. Furthermore, this app provides patients with unlimited free lifetime access to the largest directory of Renal Diet recipes. The unique feature of this app is patients can book a holiday dialysis session s that patients can enjoy their holiday without worrying about not getting good quality dialysis. Once the patients book a holiday dialysis session, the NephroPlus team will find their nearest dialysis center and arrange the dialysis slot for them (NephroPlus, n.d.). This app can be used by NephroPlus patients. Not NephroPlus patients can also use this app but only certain features are accessible such as browsing the recipes, clinical tips,

booking holiday dialysis, and finding the nearest NephroPlus Center. The downside of this app is it does not provide patients with specialist appointments to monitor patient's health conditions and only provides video consultations. In addition, this app is specifically for NephroPlus Centers only and does not provide information about other private dialysis centers.



Login

Let's start with your
Mobile number

+91 ▾

Enter mobile number

I accept the [Terms & Conditions](#)

Get OTP

Skip

Figure 2. 3 Sample UI 1 of NephroPlus App

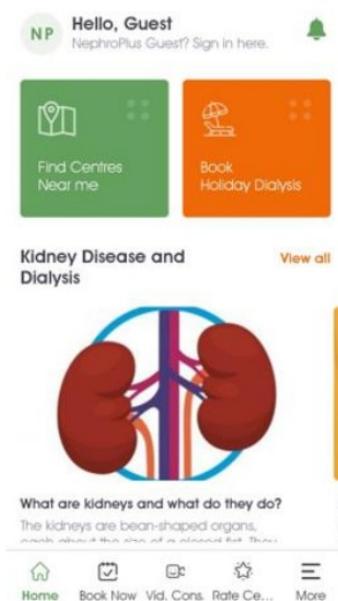


Figure 2. 4 Sample UI 2 of NephroPlus App

2.2.3 Application 3 - Freedom Dialysis App

(Freedom Dialysis - Apps on Google Play, 2022)

Freedom Dialysis is an application developed by Epic Medical Company that can be used by dialysis patients to book a dialysis appointment. It was released on 27th December 2021. It offers the simplest way to search for the nearest dialysis center around the patient and book an appointment. This app provides patients with over 1000 dialysis centers in more than 30 big cities around the world and it makes it more convenient for patients who are traveling. In addition, this app has a multi-payment methods option that patients can choose to pay for the treatment because Freedom Dialysis offers this facility to private dialysis centers only and it comes with instant booking information. The interesting part of this app is they provide a QR code for the patients to open the dialysis center's door. The QR code will be available 5 minutes before the treatment appointment start time and can be used for 1 time only. When patients have successfully opened the door, the QR code will be invalid. It makes it easy for the patients to get inside the center without having to wait for the staff to open the door for them. Furthermore, there is a feature that offers patients with transportation service which is very effortless for patients who need transportation to go to the dialysis centers. However, this app is precisely for booking an appointment and it does not have a feature where patients can review their treatment data for that appointment. Patients can only view the history of the booked appointment without any further information. Moreover, it does not have a centralized database to record and store patients' treatment data for the dialysis centers to access.



Figure 2. 5 Sample UI 1 of Freedom Dialysis App



Figure 2. 6 Sample UI 2 of Freedom Dialysis App

2.3 Summary

In Malaysia, there is no existing application that helps dialysis patient to review their treatment data and other features provided on the NephroPlus app. Besides, based on the released date of the three existing applications, we know that this application facility for dialysis patients still does not get the notice or awareness of how necessary it is for dialysis patients in Malaysia as we know that the dialysis patients in Malaysia have increased. Furthermore, these three apps are not widely used in Malaysia. For Fresenius myCompanion app, it is widely used in the United States, German, and China (Fresenius Medical Care, n.d.). NephroPlus is located in India and Freedom Dialysis app is widely used also in the United States.

Table 2. 1 Comparison between the three existing app

Application	Fresenius myCompanion	NephroPlus	Freedom Dialysis
Functions	<ol style="list-style-type: none">1. View treatment data2. View health info3. View medication needs to be taken	<ol style="list-style-type: none">1. View treatment data2. View lab result3. Book dialysis slot	<ol style="list-style-type: none">1. Book dialysis slot2. Provide other dialysis centers information
Advantages	<ol style="list-style-type: none">1. Provide recipes for dialysis patients2. Patients get authorized easily3. Customize dashboard	<ol style="list-style-type: none">1. Provide holiday dialysis sessions2. Provide payment feature3. Provide recipes4. Have a video consult	<ol style="list-style-type: none">1. Provide QR code to enter the dialysis center2. Provide payment feature3. Provide transportation service

Disadvantages	<ol style="list-style-type: none"> 1. No feature for patients to rearrange their treatment time slot and appointment with specialists 2. Does not provide patients with other dialysis centers' information 	<ol style="list-style-type: none"> 1. Does not provide patients with specialist appointments 2. Does not provide information about other private dialysis centers 	<ol style="list-style-type: none"> 1. Does not display treatment data 2. Does not provide appointments with specialists
---------------	---	---	---

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, we will discuss the methodology that is going to be implemented in this project as well as the detailed requirements, design, testing plan, and the potential of the project.

3.2 Project Management Framework

The methodology that is going to be implemented is the Rapid Application Development (RAD) methodology which does not consume a lot of time or resources on planning and instead uses prototyping techniques to introduce the application. RAD is also a development lifecycle designed to give speedy development and higher-quality results than those implemented with the traditional methods. It is designed to take maximum advantage of powerful development software that has evolved (Daud et al., 2010). RAD consists of 4 stages which are Requirements Planning, User Design, Rapid Construction, and Cutover. The implementation will be easier by using RAD as the development focuses on each requirement developed at one time. It also improves user satisfaction since the user will be involved throughout the development of the system.

Rapid Application Development (RAD)

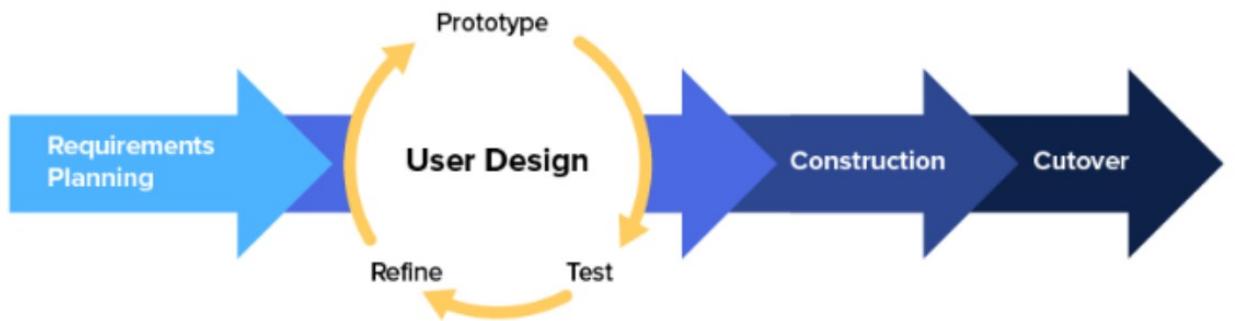


Figure 3. 1 Workflow of RAD

The workflow or stages in RAD methodology are shown in figure 3.1 above. In the study, this method is compatible with the application as it can produce a better application within time constraints.

The first phase is the requirements planning which is to establish a general understanding of the business operation of the Dialysis Center, observing the current operation which is the flow of inserting data manually, and to identify the business process that will be supported by myDialysis application. At this phase, the meeting between the developer and the users or clients will be held to mutual understanding of the objectives of the project development and general idea about the application. The meeting will discuss on the research of current situation, defining and finalizing the requirements.

Second phase is the application design which is to analyze in detail about the Dialysis Center activities and data associated with the proposed application area, and to develop the structure in terms of interface and database. Appropriate approach to the application will be selected and a work plan defining the steps for the transition of the application will be prepared. The scheduled done need to be followed as guidance to ensure the completeness of the application. UML will be created for the better understanding of myDialysis application. The context diagram and data flow diagram

will be constructed to study the interaction between use case, object and the flow of the application.

Third phase is development process which is to complete the detailed design of myDialysis application. In this phase, the application will be created and tested by the end-user to proof the validation of the interface and the functionality of the application. The interaction between main function and subfunction of the application need to be valid in order to prevent errors. When there is no error in this phase, the next phase can be proceed without issue.

The last phase of RAD is cutover which is the delivery of the developed application to its client or user. Cutover can be involved with some other activities such as implementation, system testing, user training and dealing with myDialysis application. The objective of this phase is to install the application in the production operation with minimal disruption of usual business activity in the Dialysis Center. Furthermore, it can maximize the effectiveness of the application in supporting the operation of Dialysis Center and easily identify the potential of failure. The client or users will be trained to operate the new application that will be install and provide support to resolve any problem that might arise.

3.3 Project Requirement

3.3.1 Functional requirements:

1. The application should allow user to sign up.
2. The application should allow the user to login by using their registered email and password.
3. The application should allow the user to update their profile.
4. The application should allow dialysis staff to add, update and delete patients' treatment records.
5. The application shall allow dialysis staff to add, update and delete other dialysis center directory.
6. The application should allow dialysis staff to add, update and delete patients' treatment slot time.
7. The application should allow dialysis staff to approve and reject patient slot time change request.
8. The application should allow dialysis staff to update and delete patients profile.
9. The application should allow dialysis staff to add, update and delete patient treatment bill.
10. The application should allow patients to request changes for their treatment slot time.
11. The application should patients to view their treatment records.
12. The application should allow patients to reschedule their appointment time.
13. The application should allow patients to make payments through various payment methods.
14. The application should allow hospital staff to add, update and delete the appointment dates for the patient.
15. The application should allow hospital staff to approve and reject patient appointment reschedule request.
16. The application should search and display the other dialysis centers' information.
17. The application should search and display patient profile.

18. The application should search and display patient upcoming and completed slot time.
19. The application should search and display patient upcoming and completed appointment details.
20. The application should search and display patient treatment records.
21. The application should search and display patient treatment bill.

3.3.2 Non-functional requirements:

1. The application should display an alert dialog if the user enters incorrect password or email.
2. The application should validate all required text form field before adding into the database to ensure that the crucial data is not null.
3. The application should display an alert dialog when data has successfully been added into database.
4. The application should be easy to navigate to accommodate all the users.
5. The application should have a low number of failures to prove that the application operates properly.
6. The application must support all versions of OS on android devices.

3.3.3 Constraints and limitations

There are a few constraints for developing the application. The constraints are in terms of hardware equipment and skill considering the low quality, out-of-date, and lack of sufficient space for the hardware equipment can be a drawback for the project and will drag the time of the development to be completed. Besides, not enough skills and experience in developing a mobile application could be the time constraint for developing the application where a lot of new knowledge such as mobile application framework and programming language need to be learnt.

Other limitations of the application are in terms of the Internet connectivity. Even though this application is a mobile application based and wireless network had been widely used, however the way we approach towards the new technology might be vary. Some of the user of this application might have trouble finding internet connectivity in their place. Even if in today's world are in the process of evolving the 5G internet connection, there must be some places especially in the rural area might face the limitation to access it. Other than that, since it is a mobile application-based system and the user are mostly elderly, device screen size also one of the limitations for this application. Elderly patient might have trouble while going through the application on their smartphone devices due to the age and health factor that might have affect their vision.

3.4 Proposed Design

3.4.1 Context Diagram

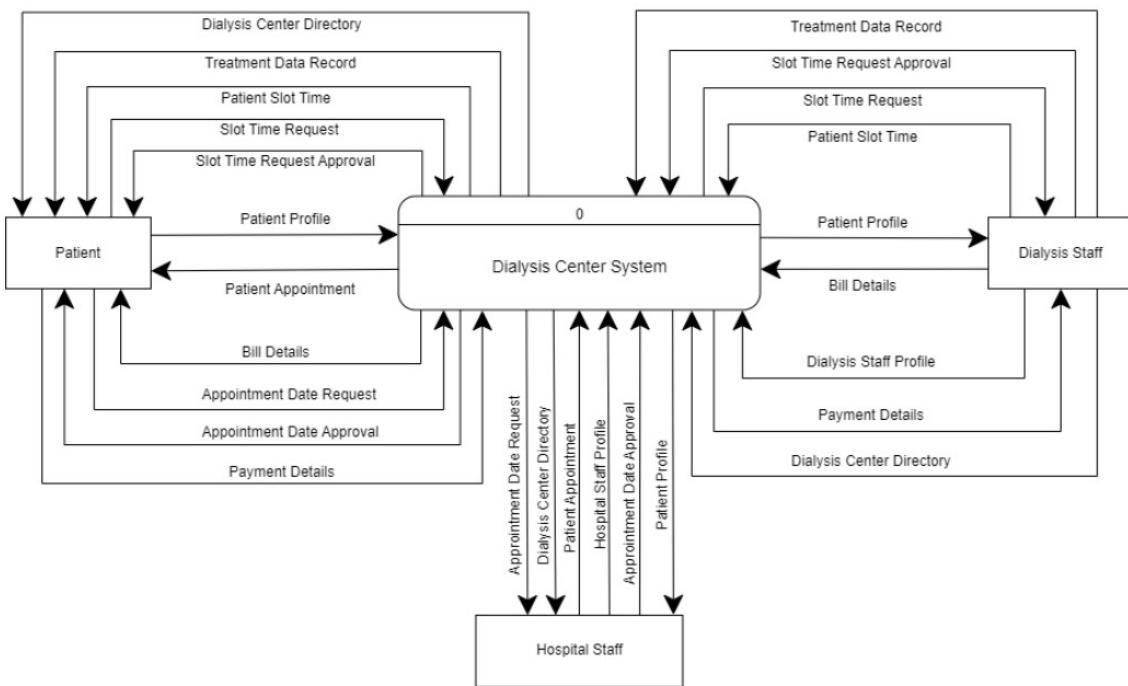


Figure 3. 2 Context Diagram

myDialysis is an application which helps dialysis center's patients to manage their dialysis schedule and specialist appointment easily. This application also helps dialysis staff to record patients' treatment data and manage patients' dialysis schedules. Users are required to sign up into the application with a special code given by the dialysis center staff which indicates that they are authorized users of the application. The code will be given manually during the patient's first visit to the dialysis center along with their specialist nurse's code. The dialysis staff can set patients' dialysis slot time schedules, add treatment records, add other dialysis center information, add patients treatment bill, and approve or reject patients' slot time requests. The patients can view their dialysis slot time, request slot time changes, view treatment records, view appointments, request appointment date changes, view the directory, and make payments. The hospital staff can set an appointment, approve or reject appointment date changes, view patients profile, and view other dialysis center directory.

3.4.2 Data Flow Diagram

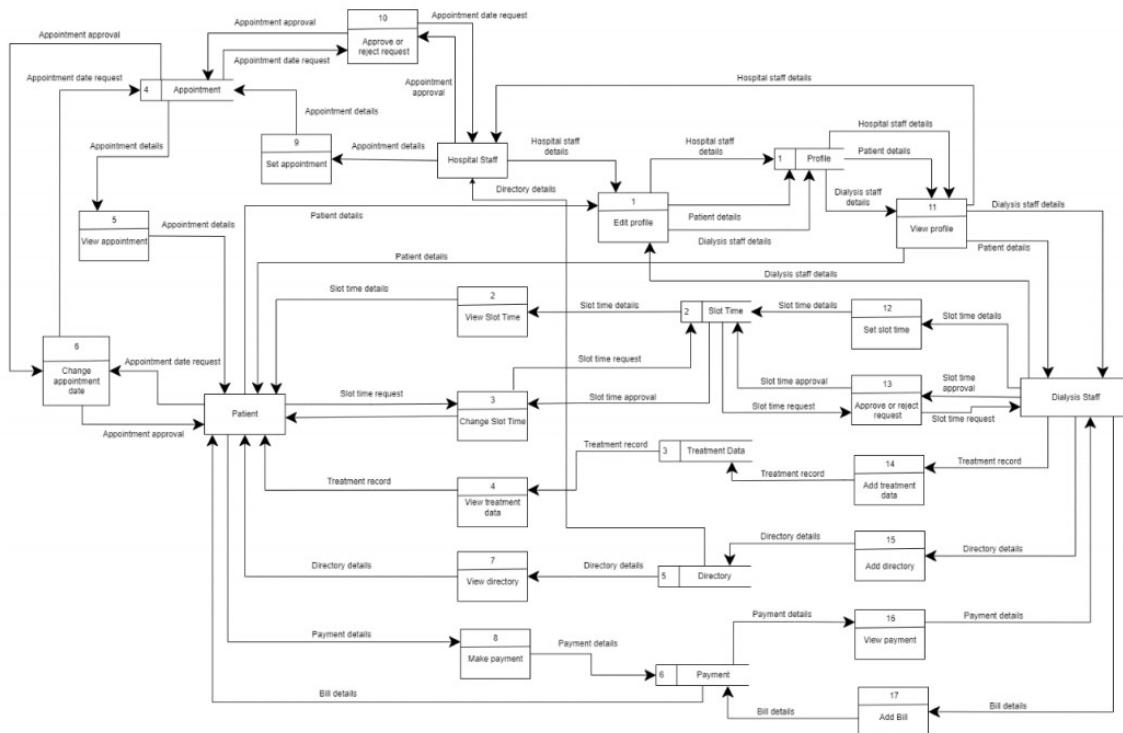


Figure 3. 3 Data Flow Diagram

3.4.3 Use Case Diagram

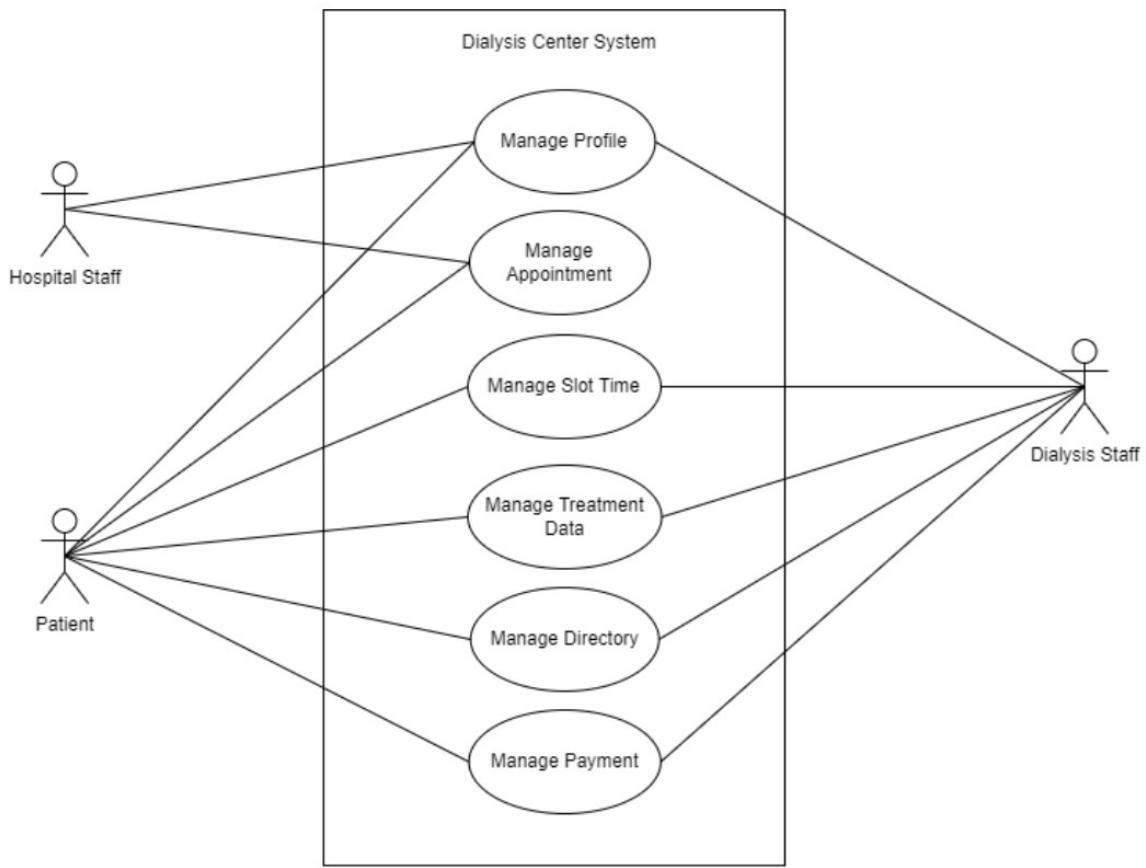


Figure 3. 4 Use Case Diagram

This application contains six modules which are Manage Slot Time, Manage Treatment Data, Manage Appointment, Manage Directory, Manage Profile, and Manage Payment. The description and actors involved in each module are shown in the table below.

Table 3. 1 Description and actor involved in each module.

Module	Description	Actor Involve
Manage Slot Time	This module is for dialysis center staff to add dialysis slot time for patients. Patients can view their slot time and request to change the slot time.	Dialysis staff and patient

Manage Treatment Data	This module is for dialysis staff to record patients' treatment data and patients can view their treatment records in this module.	Dialysis staff and patient
Manage Appointment	This module allows hospital staff to set a specialist appointment for the patients. Patients can request to change the appointment date and view appointment details in this module.	Hospital staff and patient
Manage Directory	This module allows dialysis staff to add, update and delete other dialysis center's information. Patients and hospital staff can view the other dialysis center details.	Dialysis staff, patient, and hospital staff
Manage Profile	This module is for users to edit and view their profiles. Dialysis staff and hospital staff also can view patients profile.	Dialysis staff, hospital staff, and patient
Manage Payment	This module is for patients to make payments, and dialysis staff can add patients bill and view the payment details and records.	Dialysis staff and patient

3.4.4 Use Case Description

3.4.4.1 Manage Profile

Table 3. 2 Manage Profile Use Case Description

Use Case Name	Manage Profile
Use Case ID	myDialysis-001
Brief Description	This use case allows users to edit and view their profile.
Actor	Dialysis Staff, Hospital Staff, and Patient

Pre-Conditions	The users must have registered and logged in to the application.
Basic Flow	<ol style="list-style-type: none"> 1. The use case starts when the user goes to the Profile Page. 2. The system will retrieve user profile from the database and display it to the user with the edit profile button. 3. User clicks on the “Edit” button. 4. The system will retrieve the existing user profile from the database and display an editable user profile form. 5. User will edit their information on the editable user profile form and click on the “Save Changes” button. 6. The system will validate the form to make sure it has been completed and save the updated user profile into the database. [A1: Incomplete form] 7. The system will display a message after the updated user profile is saved successfully. 8. The use case end.
Alternative Flow	<p>[A1: Incomplete form]</p> <ol style="list-style-type: none"> 1. User profile form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow to complete the form.
Exception Flow	None
Post-Conditions	User profile is successfully edited or displayed; the database is updated.
Rules	None
Constraints	Profile cannot be deleted.

3.4.4.2 Manage Appointment

Table 3. 3 Manage Appointment Use Case Description

Use Case Name	Manage Appointment
Use Case ID	myDialysis-002
Brief Description	This use case allows Hospital Staff to set an appointment, edit an appointment, and approve or reject appointment date changes request. This use case also allows Patients to view appointments and reschedule appointments.
Actor	Hospital Staff, and Patient
Pre-Conditions	The users must have logged in to the system.
Basic Flow	<p>Hospital Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Appointment Page. 2. System will retrieve the appointment details from the database and display them to the user with the “Set Appointment” button, “Edit” button, and “Request” button. [A1: Edit Appointment] [A2: Appointment Request] 3. User clicks on the “Set Appointment” button. 4. System will display an appointment form. 5. User will fill in the details for the appointment and click on the “Save” button. 6. The system will validate the form to make sure it has been completed and save the appointment details into the database. [A3: Incomplete form] 7. The system will display a message after the appointment is saved successfully. 8. The use case end.

	<p>Patient:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Appointment Page. 2. The system will retrieve the user appointment details from the database and display all the user appointment lists with the “Reschedule” button. 3. User clicks on the “Reschedule” button. 4. System will display a request date form. 5. User fills in the request form and clicks on the “Submit” button. 6. The system will validate the form to make sure it has been completed and save the request details into the database. [A1: Incomplete form] 7. The system will display a message after the request is saved successfully. 8. The use case end.
Alternative Flow	<p>Hospital Staff:</p> <p>[A1: Edit Appointment]</p> <ol style="list-style-type: none"> 1. User clicks on the “Edit” button. 2. System will retrieve the appointment details from the database and display an editable appointment form. 3. User will fill in the details for the appointment and click on the “Save Changes” button. 4. The system will validate the form to make sure it has been completed and save the appointment details into the database. <p>[A3: Incomplete form]</p> <ol style="list-style-type: none"> 5. The system will display a message after the appointment is saved successfully. 6. The use case end. <p>[A2: Appointment Request]</p>

	<ol style="list-style-type: none"> 1. User clicks on the “Request” button. 2. System retrieves the request details from the database and displays them to the user with the “Approve” button and the “Reject” button. 3. User clicks on either the “Approve” button to approve the appointment date request or the “Reject” button to decline the appointment date request. 4. System will display a message after the approval status is saved into the database successfully. 5. The use case end. <p>[A3: Incomplete form]</p> <ol style="list-style-type: none"> 1. Appointment form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow to complete the form. <p>Patient:</p> <p>[A1: Incomplete form]</p> <ol style="list-style-type: none"> 1. The request date form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 3 in the alternative flow to complete the form.
Exception Flow	None
Post-Conditions	Patient appointment is created and displayed successfully.
Rules	None
Constraints	None

3.4.4.3 Manage Slot Time

Table 3. 4 Manage Slot Time Use Case Description

Use Case Name	Manage Slot Time
Use Case ID	myDialysis-003
Brief Description	This use case allows dialysis staff to add patient dialysis slot time and approve or reject the patient slot time request. This use case also allows patients to view their slot time and request new dialysis slot time.
Actor	Dialysis Staff, and Patient
Pre-Conditions	The users must have logged in to the system.
Basic Flow	<p>Dialysis Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Slot Time Page. 2. System will retrieve the slot time details from the database and display them to the user with the “Add Slot Time” button and “Slot Time Request” button. [A1: User chooses the “Slot Time Request” button] 3. User clicks on the “Add Slot Time” button. 4. System will display a slot time form. 5. User fills in the slot time form and clicks on the “Save” button. 6. The system will validate the form to make sure it has been completed and save the slot time details into the database. [A2: Incomplete form] 7. The system will display a message after the slot time is saved successfully. 8. The use case end.

	<p>Patient:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Slot Time Page. 2. System will retrieve the slot time details from the database and display them to the user with the “Request Slot Time” button. 3. User clicks on the “Request Slot Time” button. 4. System will display a request slot time form. 5. User fills in the slot time request form and clicks on the “Submit” button. 6. The system will validate the form to make sure it has been completed and save the slot time request into the database. <p>[A1: Incomplete form]</p> <ol style="list-style-type: none"> 7. The system will display a message after the slot time request is saved successfully. 8. The use case end.
Alternative Flow	<p>Dialysis Staff:</p> <p>[A1: User chooses the “Slot Time Request” button]</p> <ol style="list-style-type: none"> 1. User clicks on the “Slot Time Request” button. 2. System will retrieve slot time requests from the database and display them to the user with the “Approve” button and the “Reject” button. 3. User clicks on either the “Approve” button to approve the slot time request or the “Reject” button to decline the slot time request. 4. System will display a message after the approval status is saved into the database successfully. 5. The use case end. <p>[A2: Incomplete form]</p>

	<ol style="list-style-type: none"> 1. The slot time form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow to complete the form. <p>Patient:</p> <p>[A1: Incomplete form]</p> <ol style="list-style-type: none"> 1. The slot time request form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 3 in the alternative flow to complete the form.
Exception Flow	None
Post-Conditions	Patients' slot time is added and saved successfully.
Rules	None
Constraints	None

3.4.4.4 Manage Treatment Data

Table 3. 5 Manage Treatment Data Use Case Description

Use Case Name	Manage Treatment Data
Use Case ID	myDialysis-004
Brief Description	This use case allows dialysis staff to record and view patients' treatment data and allows patients to view their treatment data.

Actor	Dialysis Staff, and Patient
Pre-Conditions	The users must have logged in to the system.
Basic Flow	<p>Dialysis Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Treatment Record Page. 2. System retrieves the treatment data from the database and displays them with Add Treatment button, and the Edit Treatment Data button [A1: User chooses to Edit Treatment Data] 3. User clicks on the “Add Treatment Data” button. 4. System will display a treatment data form. 5. User fills in the treatment data form and clicks on the “Save” button. 6. The system will validate the form to make sure it has been completed and save the treatment data into the database. [A2: Incomplete form] 7. The system will display a message after the treatment data is saved successfully. 8. The use case end. <p>Patient:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Treatment Record Page. 2. System will retrieve the treatment data record from the database and display them to the user. 3. The use case end.
Alternative Flow	<p>Dialysis Staff:</p> <p>[A1: User chooses to Edit Treatment Data]</p> <ol style="list-style-type: none"> 1. User clicks on the “Edit Treatment Data” button.

	<ol style="list-style-type: none"> 2. System retrieves the treatment data from the database and displays an editable treatment data form to the user. 3. User fills in the editable treatment data form and clicks on the “Save Changes” button. 4. The system will validate the form to make sure it has been completed and save the treatment data into the database. [A3: Incomplete form] 5. The system will display a message after the treatment data is saved successfully. 6. The use case end. <p>[A2: Incomplete form]</p> <ol style="list-style-type: none"> 1. The treatment data form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow for Add Treatment Data and from step 3 in the alternative flow for Edit Treatment Data to complete the form.
Exception Flow	None
Post-Conditions	Treatment data is successfully added, edited, and displayed.
Rules	None
Constraints	None

3.4.4.5 Manage Directory

Table 3. 6 Manage Directory Use Case Description

Use Case Name	Manage Directory
Use Case ID	myDialysis-005
Brief Description	This use case allows dialysis staff to add, edit and view other dialysis center information and allows patients to view the directory.
Actor	Dialysis Staff, Hospital Staff, and Patient
Pre-Conditions	The users must have logged in to the system.
Basic Flow	<p>Dialysis Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Directory Page. 2. System will retrieve the directory data from the database and show them to the user with the “Add Directory” button, and the “Edit Directory” button [A1: User chooses Edit Directory] 3. User clicks on the “Add Directory” button. 4. System will display a directory form to the user. 5. User fills in the directory form and clicks on the “Save” button. 6. System will validate the form to make sure it has been completed and save the directory data into the database. [A2: Incomplete form] 7. The system will display a message after the directory data is saved successfully. 8. The use case end.

	<p>Patient and Hospital Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Directory Page. 2. System will retrieve the directory data from the database and display them to the user. 3. The use case end.
Alternative Flow	<p>Dialysis Staff:</p> <p>[A1: User chooses Edit Directory]</p> <ol style="list-style-type: none"> 1. User clicks on the “Edit Directory” button. 2. System retrieves the directory data from the database and displays an editable directory form to the user. 3. User fills in the editable directory form and clicks on the “Save Changes” button. 4. The system will validate the form to make sure it has been completed and save the directory data into the database. [A2: Incomplete form] 5. The system will display a message after the directory data is saved successfully. 6. The use case end. <p>[A2: Incomplete form]</p> <ol style="list-style-type: none"> 1. The directory form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow for Add Directory and from step 3 in the alternative flow for Edit Directory to complete the form.
Exception Flow	None
Post-Conditions	Directory is added, edited, and displayed successfully.
Rules	None
Constraints	None

3.4.4.6 Manage Payment

Table 3. 7 Manage Payment Use Case Description

Use Case Name	Manage Payment
Use Case ID	myDialysis-006
Brief Description	This use case allows dialysis staff to add bills and view payment details and also allows patients to view billing details and make payments.
Actor	Dialysis Staff, and Patient
Pre-Conditions	The users must have logged in to the system.
Basic Flow	<p>Dialysis Staff:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Payment Page. 2. System will retrieve billing details from the database and display them to the user with the “Add Bill” button, “Edit Bill” button, and “View Payment” button. [A1: User chooses the “Edit Bill” button] [A2: User chooses the “View Payment” button] 3. User clicks on the “Add Bill” button. 4. System will display a billing form. 5. User fills in the billing form and clicks on the “Save” button. 6. The system will validate the form to make sure it has been completed and save the billing details into the database. [A3: Incomplete form] 7. The system will display a message after the billing details are saved successfully. 8. The use case end.

	<p>Patient:</p> <ol style="list-style-type: none"> 1. The use case starts when the user goes to the Payment Page. 2. System will retrieve the billing details from the database and display them to the user with the “Pay Bill” button. 3. User clicks on the “Pay Bill” button. 4. System will display a payment form to the user. 5. User fills in the payment form and clicks on the “Pay” button. 6. The system will validate the form to make sure it has been completed and navigate the user to the payment gateway [A1: Incomplete form] 7. User will make a payment transaction. 8. System will validate the transaction and save the payment details into the database. [E1: Failed Transaction] 9. The system will display a message after the payment details are saved successfully. 10. The use case end.
Alternative Flow	<p>Dialysis Staff:</p> <p>[A1: User chooses the “Edit Bill” button]</p> <ol style="list-style-type: none"> 1. User clicks on the “Edit Bill” button. 2. System will retrieve billing details from the database and display an editable billing form. 3. User fills in the editable billing form and clicks on the “Save Changes” button. 4. The system will validate the form to make sure it has been completed and save the billing details into the database. [A3: Incomplete form] 5. The system will display a message after the billing details are saved successfully. 6. The use case end.

	<p>[A2: User chooses the “View Payment” button]</p> <ol style="list-style-type: none"> 1. User clicks on the “View Payment” button. 2. System retrieves the payment details from the database and displays them to the user. 3. The use case end. <p>[A3: Incomplete form]</p> <ol style="list-style-type: none"> 1. The billing form is not complete. 2. The system will display an error message. 3. User clicks on the “OK” button and continues from step 5 in the basic flow for Add Bill or from step 3 in the alternative flow for Edit Bill to complete the form. <p>Patient:</p> <p>[A1: Incomplete form]</p> <ol style="list-style-type: none"> 1. The payment form is not complete. 2. The system will display an error message. 3. Continue from step 5 in the basic flow to complete the form.
Exception Flow	<p>Dialysis Staff: None</p> <p>Patient:</p> <p>[E1: Failed Transaction]</p> <ol style="list-style-type: none"> 1. User payment transaction has failed. 2. System display an error message. 3. User clicks on the “OK” button. 4. The use case end.

Post-Conditions	None
Rules	None
Constraints	None

3.4.5 Data Design

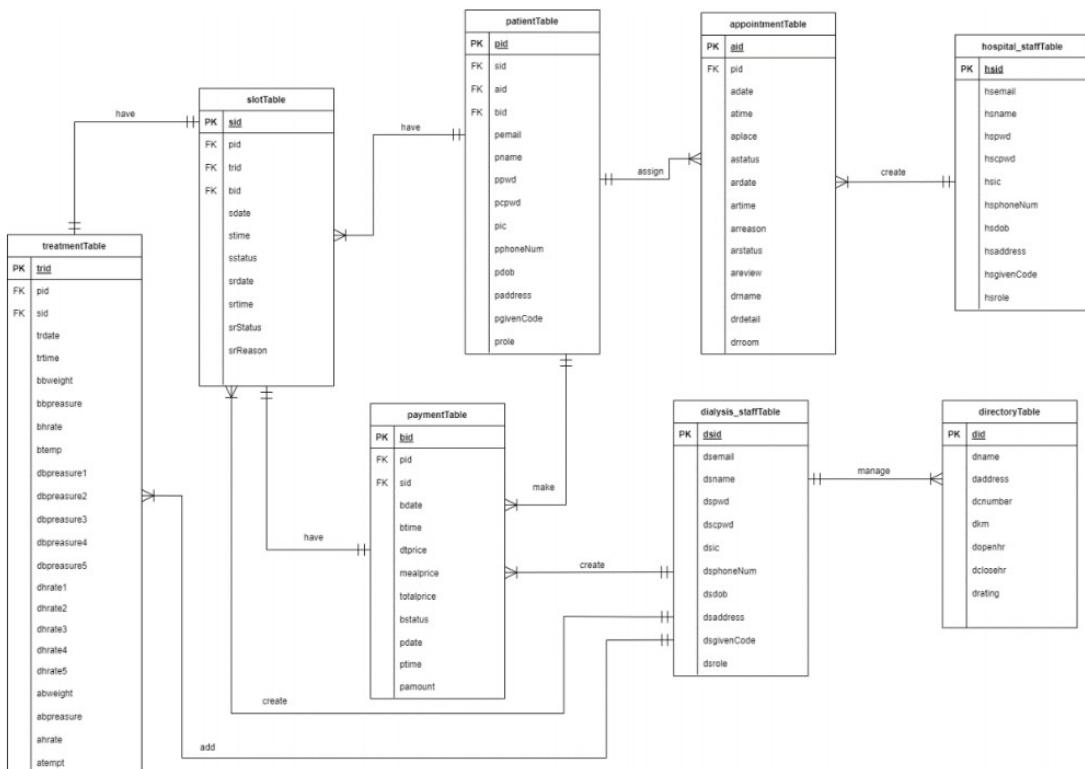


Figure 3. 5 Entity Relationship Diagram

3.4.6 Data Dictionary

1. Dialysis Staff Table

Table 3. 8 Dialysis Staff Data Dictionary

Field Name	Description	Data Type	Constraint
dsid	Dialysis staff unique ID	VARCHAR(50)	PK

dsemail	Dialysis staff email	VARCHAR(50)	
dsname	Dialysis staff name	VARCHAR(12)	
dspwd	Dialysis staff password	VARCHAR(50)	
dscpwd	Dialysis staff confirm password	VARCHAR(50)	
dsic	Dialysis staff IC number	VARCHAR(20)	
dsphoneNum	Dialysis staff phone number	VARCHAR(12)	
dsdob	Dialysis staff date of birth	DATE	
dsaddress	Dialysis staff address	VARCHAR(50)	
dsgivenCode	Dialysis staff given code	VARCHAR(50)	
dsrole	Dialysis staff role	VARCHAR(50)	

2. Hospital Staff

Table 3. 9 Hospital Staff Data Dictionary

Field Name	Description	Data Type	Constraint
hsid	Hospital staff unique ID	VARCHAR(50)	PK
hsemail	Hospital staff email	VARCHAR(50)	
hsname	Hospital staff name	VARCHAR(12)	
hspwd	Hospital staff password	VARCHAR(50)	
hscpwd	Hospital staff confirm password	VARCHAR(50)	
hsic	Hospital staff IC number	VARCHAR(20)	
hsphoneNum	Hospital staff phone number	VARCHAR(12)	
hsdob	Hospital staff date of birth	DATE	
hsaddress	Hospital staff address	VARCHAR(50)	
hsgivenCode	Hospital staff given code	VARCHAR(50)	
hsrole	Hospital staff role	VARCHAR(50)	

3. Patient

Table 3. 10 Patient Data Dictionary

Field Name	Description	Data Type	Constraint
pid	Patient staff unique ID	VARCHAR(50)	PK
sid	Slot time unique ID	VARCHAR(50)	FK
aid	Appointment unique ID	VARCHAR(50)	FK
bid	Bill unique ID	VARCHAR(50)	FK
pemail	Patient staff email	VARCHAR(50)	
pname	Patient staff name	VARCHAR(12)	
ppwd	Patient staff password	VARCHAR(50)	
pcpwd	Patient staff confirm password	VARCHAR(50)	
pic	Patient staff IC number	VARCHAR(20)	
pphoneNum	Patient staff phone number	VARCHAR(12)	
pdob	Patient staff date of birth	DATE	
paddress	Patient staff address	VARCHAR(50)	
pgivenCode	Patient staff given code	VARCHAR(50)	
prole	Patient staff role	VARCHAR(50)	

4. Slot Time

Table 3. 11 Slot Time Data Dictionary

Field Name	Description	Data Type	Constraint
sid	Slot ID number	VARCHAR(50)	PK
sdate	Patient dialysis treatment session date	DATE	
stime	Patient dialysis treatment session time	TIME	
trid	Patient email	VARCHAR(50)	FK
bid	Request ID number	VARCHAR(15)	FK
sstatus	Slot time status	VARCHAR(50)	

srdate	Proposed date for slot time request	DATE	
srttime	Proposed time for slot time request	TIME	
srStatus	Slot time request status	VARCHAR(50)	
srReason	Reason for slot time request	VARCHAR(50)	

5. Treatment Record

Table 3. 12 Treatment Record Data Dictionary

Field Name	Description	Data Type	Constraint
trid	Treatment ID number	VARCHAR(15)	PK
trdate	Treatment record date	DATE	
trtime	Treatment record time	TIME	
bbweight	Patient body weight before treatment	VARCHAR(15)	
bbpressure	Patient blood pressure before treatment	VARCHAR(15)	
bhrate	Patient heart rate before treatment	VARCHAR(15)	
btemp	Patient body temperature before treatment	VARCHAR(15)	
dbpressure1	Patient blood pressure on the first hour of treatment	VARCHAR(15)	
dbpressure2	Patient blood pressure on the second hour of treatment	VARCHAR(15)	
dbpressure3	Patient blood pressure on the third hour of treatment	VARCHAR(15)	
dbpressure4	Patient blood pressure on fourth hour of treatment	VARCHAR(15)	
dbpressure5	Patient blood pressure on fifth hour of treatment	VARCHAR(15)	
dhrate1	Patient heart rate on first hour of treatment	VARCHAR(15)	
dhrate2	Patient heart rate on second hour of treatment	VARCHAR(15)	
dhrate3	Patient heart rate on third hour of treatment	VARCHAR(15)	
dhrate4	Patient heart rate on fourth hour of treatment	VARCHAR(15)	

dhrate5	Patient heart rate on fifth hour of treatment	VARCHAR(15)	
abweight	Patient body weight after treatment	VARCHAR(15)	
abpreasure	Patient blood pressure after treatment	VARCHAR(15)	
ahrate	Patient heart rate after treatment	VARCHAR(15)	
atemp	Patient body temperature after treatment	VARCHAR(15)	
sid	Patient email address	VARCHAR(15)	FK
pid	Patient unique ID	VARCHAR(15)	FK

6. Appointment

Table 3. 13 Appointment Data Dictionary

Field Name	Description	Data Type	Constraint
aid	Appointment ID number	VARCHAR(15)	PK
aplace	Place of the appointment	VARCHAR(50)	
atime	Time of the appointment	TIME	
adate	Date of the appointment	DATE	
astatus	Appointment status	VARCHAR(15)	
ardate	Appointment reschedule date	DATE	
artime	Appointment reschedule time	TIME	
arreason	Appointment reschedule reason	VARCHAR(15)	
arstatus	Appointment reschedule status	VARCHAR(15)	
areview	Appointment review	VARCHAR(50)	
drname	Doctor's name	VARCHAR(15)	
drdetail	Doctor's expertise	VARCHAR(15)	
drroom	Doctor's room details	VARCHAR(15)	
pid	Patient unique ID	VARCHAR(15)	FK

7. Bill

Table 3. 14 Bill Data Dictionary

Field Name	Description	Data Type	Constraint
bid	Bill ID number	VARCHAR(15)	PK
pid	Patient unique ID	VARCHAR(15)	FK
sid	Slot ID number	VARCHAR(15)	FK
bdate	Date of the bill been released	DATE	
btime	Time of the bill been released	TIME	
dtprice	Price of the treatment service	VARCHAR(15)	
mealprice	Price of the meal service	VARCHAR(15)	
totalprice	Total amount to be paid for the bill	VARCHAR(15)	
pdate	Date of the payment made	DATE	
ptime	Time of the payment made	TIME	
pamount	Amount paid by patient	VARCHAR(15)	
pmethod	Method used to make payment	VARCHAR(50)	
bstatus	Status for the payment	VARCHAR(15)	

8. Dialysis Center Table

Table 3. 15 Dialysis Center Data Dictionary

Field Name	Description	Data Type	Constraint
did	Directory ID number	VARCHAR(15)	PK
dname	Dialysis center name	VARCHAR(50)	
daddress	Dialysis center address	VARCHAR(50)	
dcnumber	Dialysis center contact number	VARCHAR(12)	
dkm	Dialysis center kilometre	VARCHAR(15)	
dopenhr	Dialysis center open hour	TIME	

dclosehr	Dialysis center close hour	TIME	
drating	Dialysis center rating	VARCHAR(15)	

3.5 Proof of Initial Concept

The prototype for this application is created by using a prototyping tool which focalize on user interface and user experience design called Figma. It is a suitable tool to build the prototype as this application has a mobile version which allow the user to view and interact with the prototypes.

3.5.1 Sign Up and Login Interface

Sign Up Interface is an interface for new users to create their accounts. The users will be given a unique code that will be used to register in the application. This is to prevent unauthorized users from taking advantage of the system and its data. Users will be required to enter their name, email address, password, and given code. If the user has already signed up for this system, then the user can access this application by logging in at the login interface. This system required the user's email address and password to log in to the system.

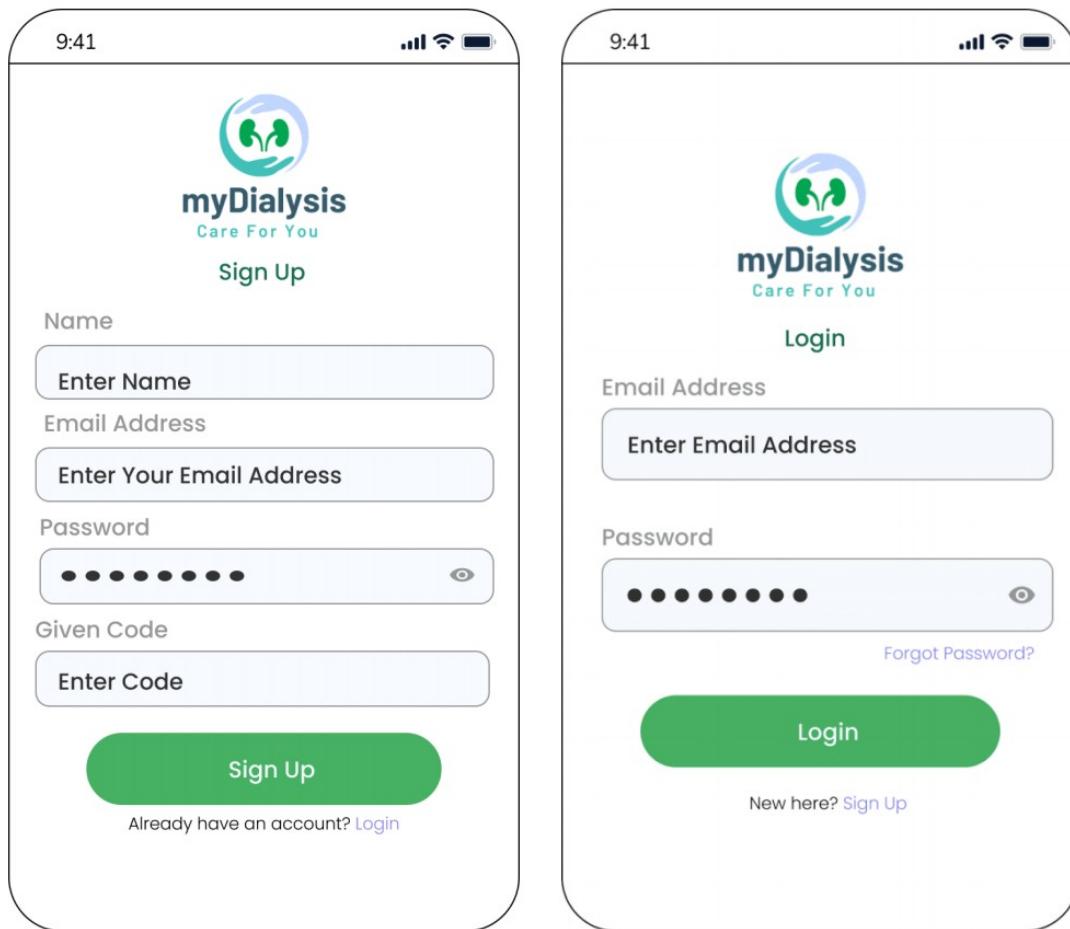


Figure 3. 6 Register Interface

Figure 3. 7 Login Interface

3.5.2 Manage Profile Interface

Once the user has registered or logged in to the system, the user can see the Profile button at the bottom right corner of the Homepage in Figure 3.8 Patient Homepage Interface. When the users click on the profile button, the system will show them an interface where users can choose either to edit their profile or view their profile as in Figure 3.9 Profile Page Interface. Figure 3.10 Edit Profile Interface shows the editable profile form when users want to edit their profile. The last figure, Figure 3.11 View Profile Interface display the user profile interface if they wish to view their profile by clicking the view profile in the profile page interface.

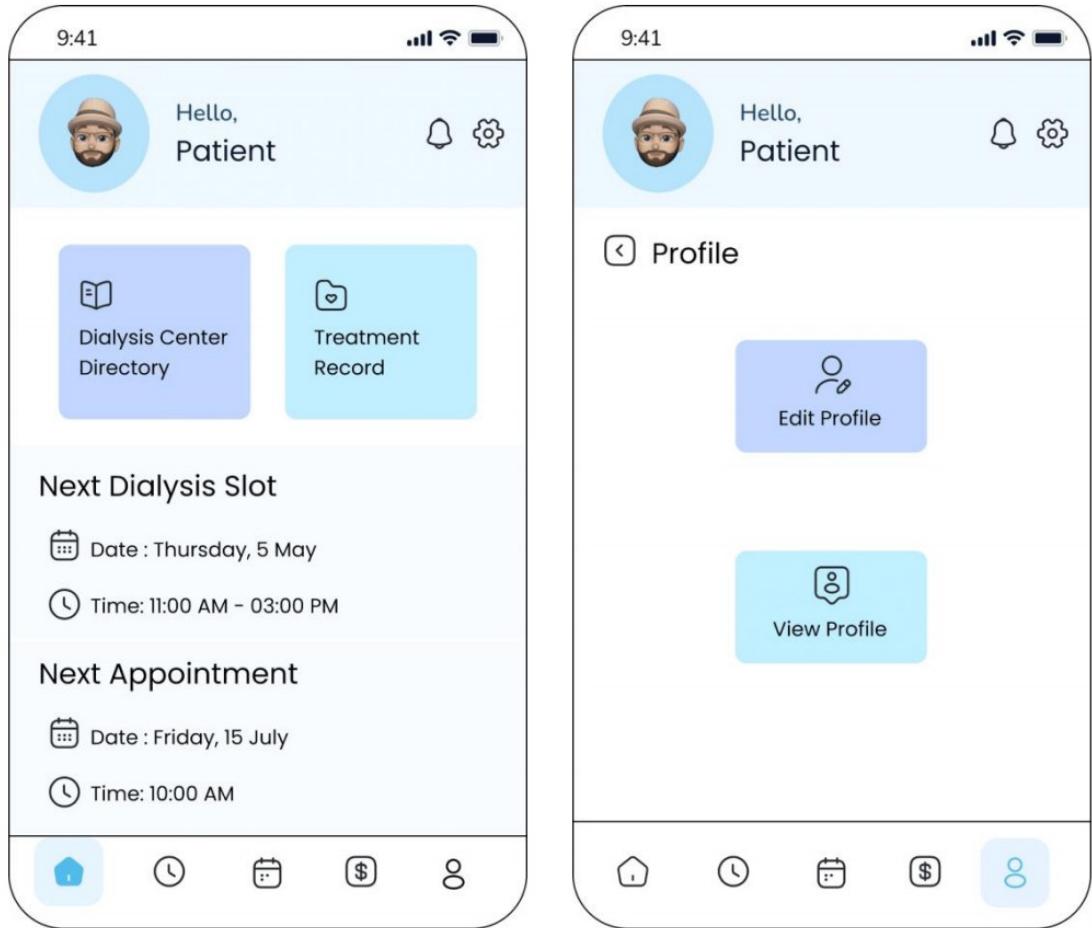


Figure 3.8 Patient Homepage Interface

Figure 3.9 Profile Page Interface

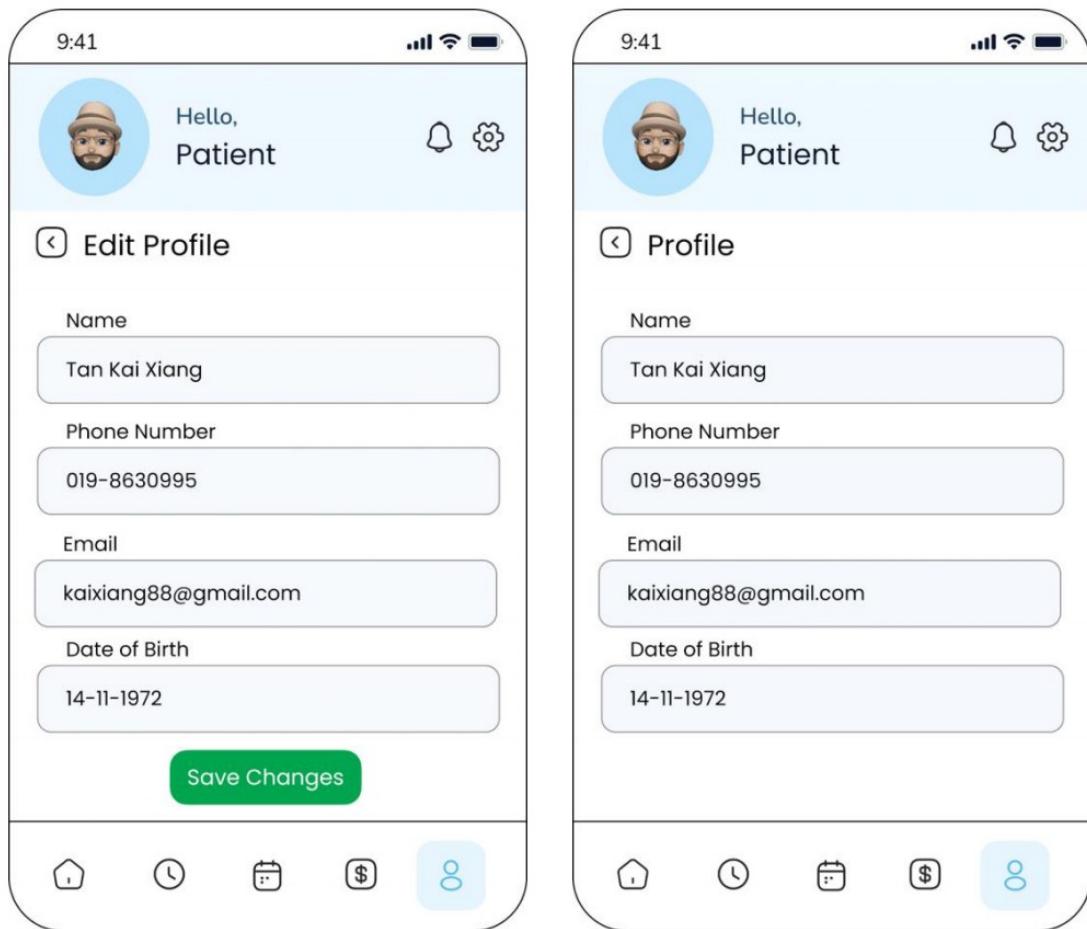
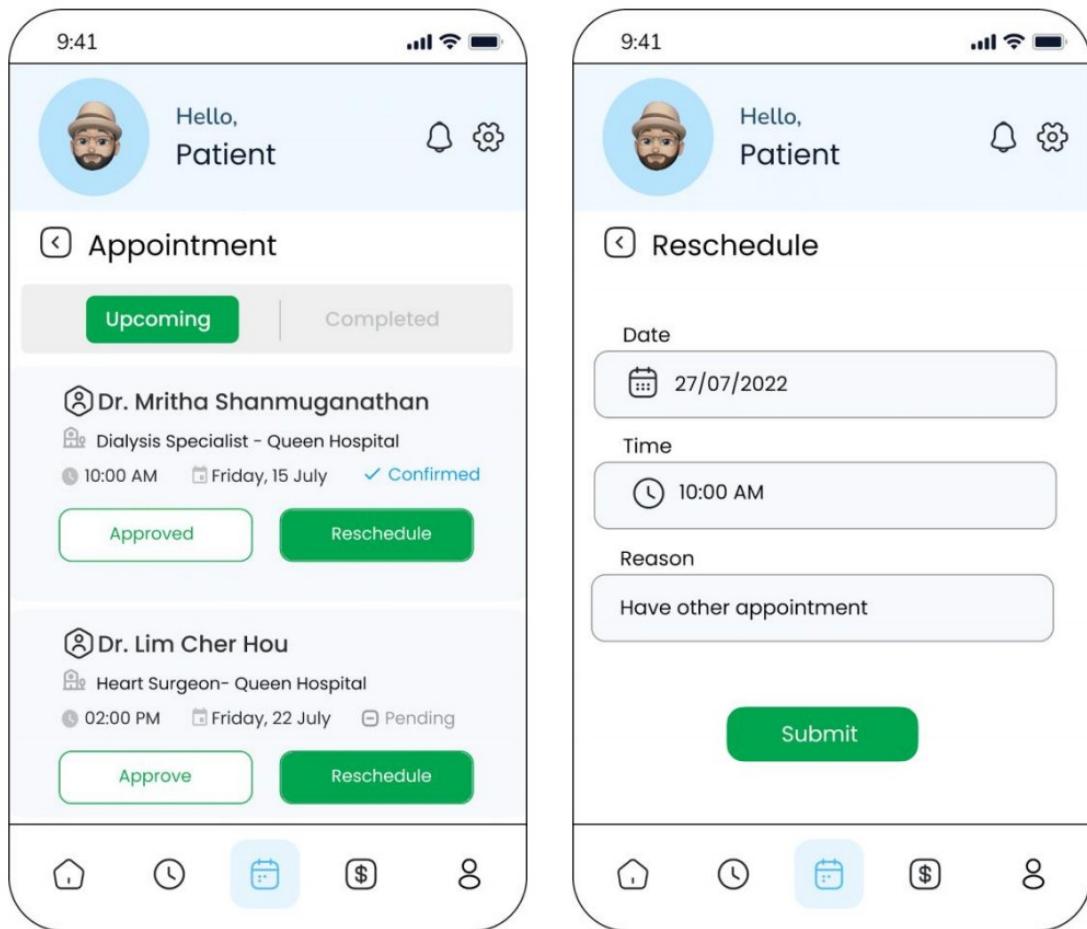


Figure 3. 10 Edit Profile Interface

Figure 3. 11 View Profile Interface

3.5.3 Manage Appointment Interface

The Patient View Appointment Interface in Figure 3.12 shows what patients will see when patients click on the appointment button at the bottom of the interface. Patients can view the appointment set by the hospital staff at this interface. Patients also can approve or request to reschedule the appointment to the desired day by clicking on the approve button or reschedule button. If the patient has approved the appointment, the approval status will be sent to the hospital staff and a ‘confirmed’ text will be displayed. Else, it will show a ‘pending’ text. The Reschedule Appointment Form as in Figure 3.13 will be shown if the patient clicks on the reschedule button on Figure 3.12 Interface. Patients are required to fill in the form before clicking on the submit button. Once done, the patients will get notified if the hospital staff approves or decline the rescheduling. Figure 3.14 is for the hospital staff to view the patients' appointments and patients' approval status. Hospital staff can edit the appointment details, when necessary, by clicking the edit button beside the approval status. Hospital staff also can add an appointment by clicking the set appointment button and the system will show an appointment form for the user to fill in as in Figure 3.15 and view the rescheduling appointment request in the reschedule interface, Figure 3.16. Hospital staff also can approve or decline the request by clicking on the approve button or the reject button.



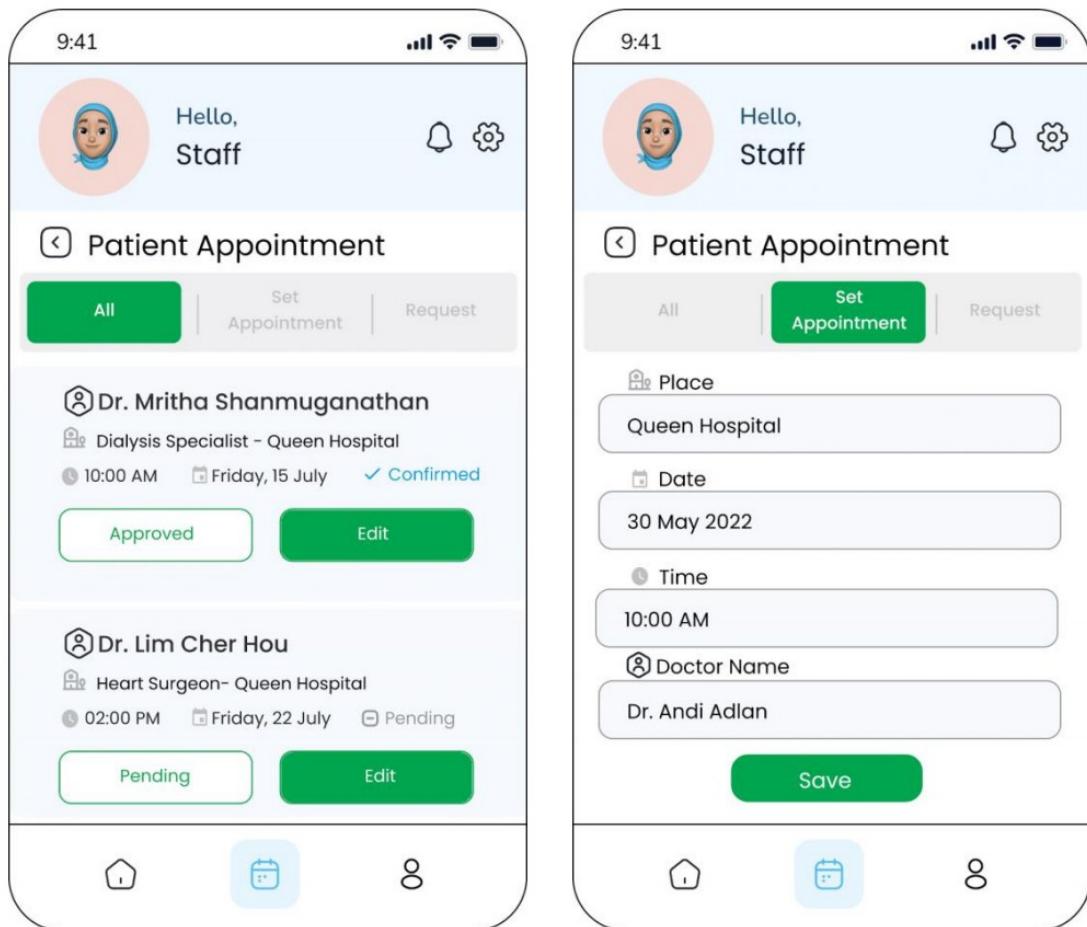


Figure 3. 14 Hospital Staff Appointment Interface

Figure 3. 15 Add Appointment Interface

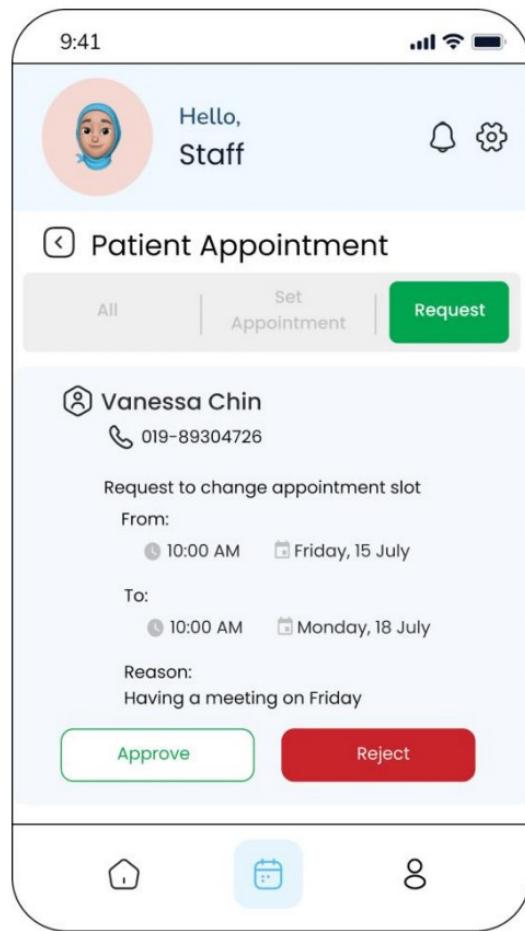


Figure 3.16 Reschedule Interface

3.5.4 Manage Slot Time

Figure 3.17 shows the patient slot time interface where patients can view their upcoming dialysis slot time. Patients also can approve the slot time to indicate that they are confirmed to come to the dialysis slot. If patients cannot make it on that day, patients can request to change the slot time by clicking the request slot time button. If patients click on the request slot time button, the interface in Figure 3.18 will be displayed and patients need to fill in the form and click on submit button to send the request to the dialysis centre. Dialysis staff can view the patients' slot time in View Slot Time Interface, Figure 3.19. The patients' confirmation status also will be displayed on that interface to indicate that they are coming to their dialysis session. Dialysis staff also can add the slot time by clicking on the add slot time button. Once the dialysis staff clicks on the add slot button, the interface in Figure 3.20 will be displayed and dialysis staff can add patients slot time. Dialysis can view the request slot time changes made by patients in Request Slot Time Interface, Figure 3.21 with the details of the changes slot time and can approve the request or decline the request by clicking on the button

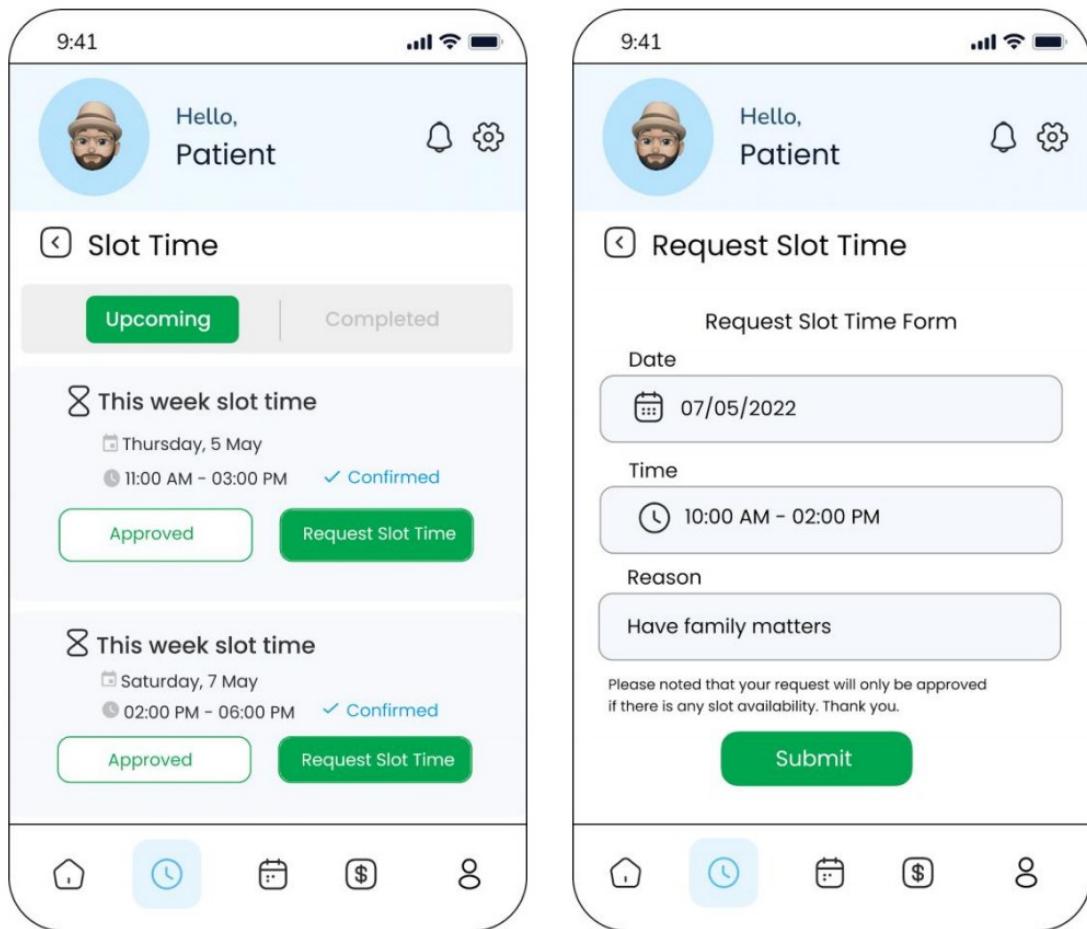


Figure 3. 17 Slot Time Interface

Figure 3. 18 Request Slot Time Interface

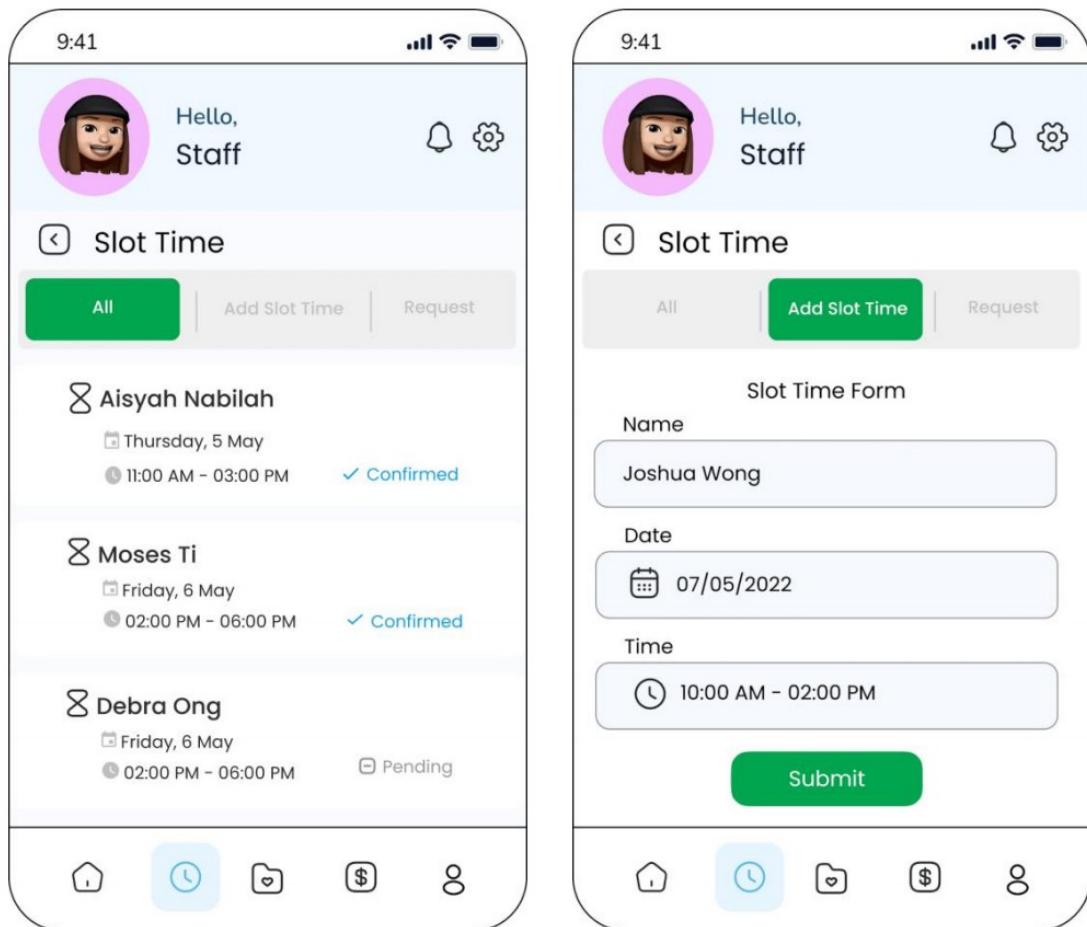


Figure 3. 19 View Slot Time Interface

Figure 3. 20 Add Slot Time Interface

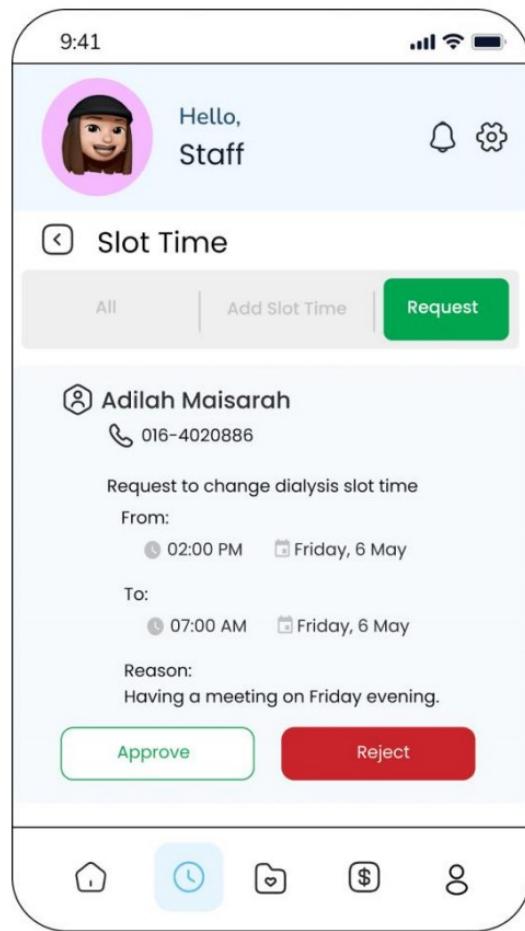


Figure 3. 21 Request Slot Time Interface

3.5.5 Manage Treatment Data

Patients can view all their dialysis treatment records in Patient Treatment Record Interface, Figure 3.22. Dialysis staff can view the patients' treatment records in Patient Treatment Record Interface, Figure 3.23 and record the patients' treatment data in Add Treatment Record, Figure 3. 24 every time the patients come to the dialysis session so that patients can always monitor their health. Dialysis staff can always edit the patients' treatment data in Edit Treatment Record Interface Figure 3. 25.

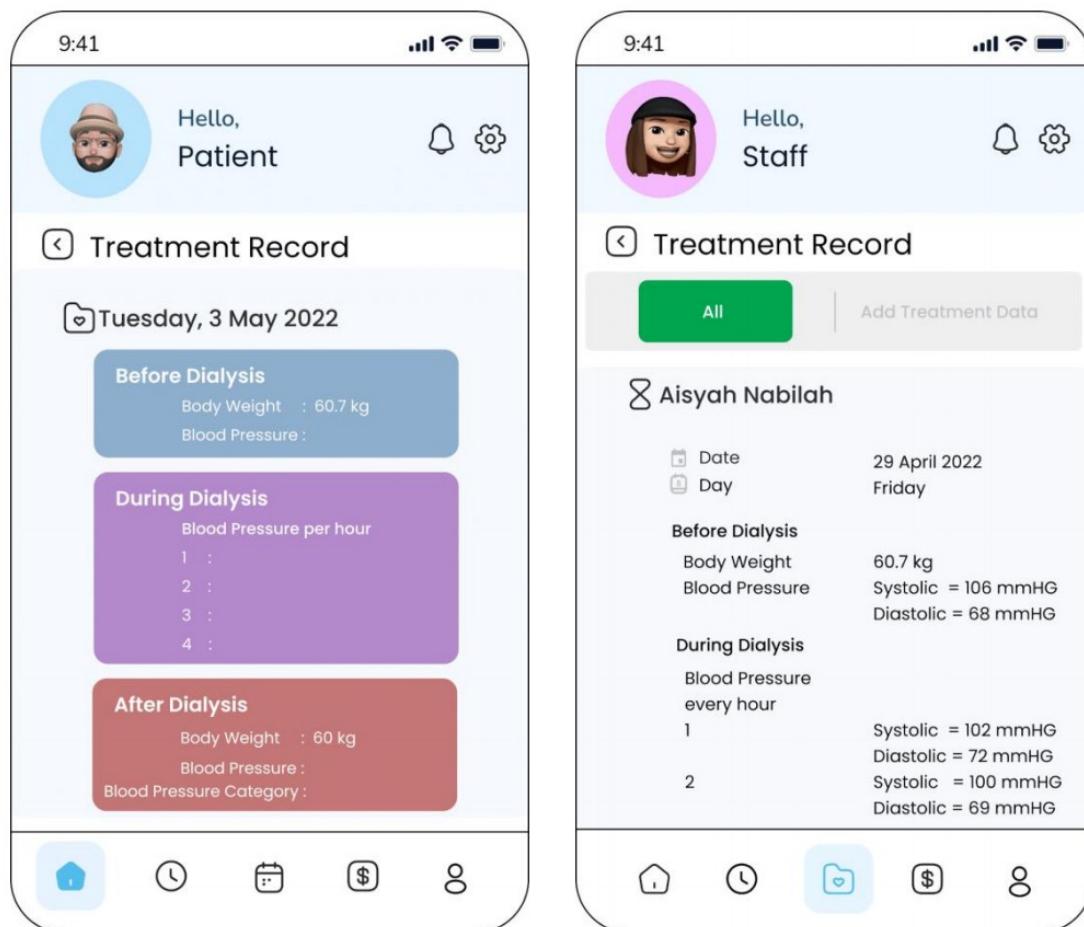


Figure 3. 22 Patient Treatment Record Interface

Figure 3. 23 Patient Treatment Record Interface

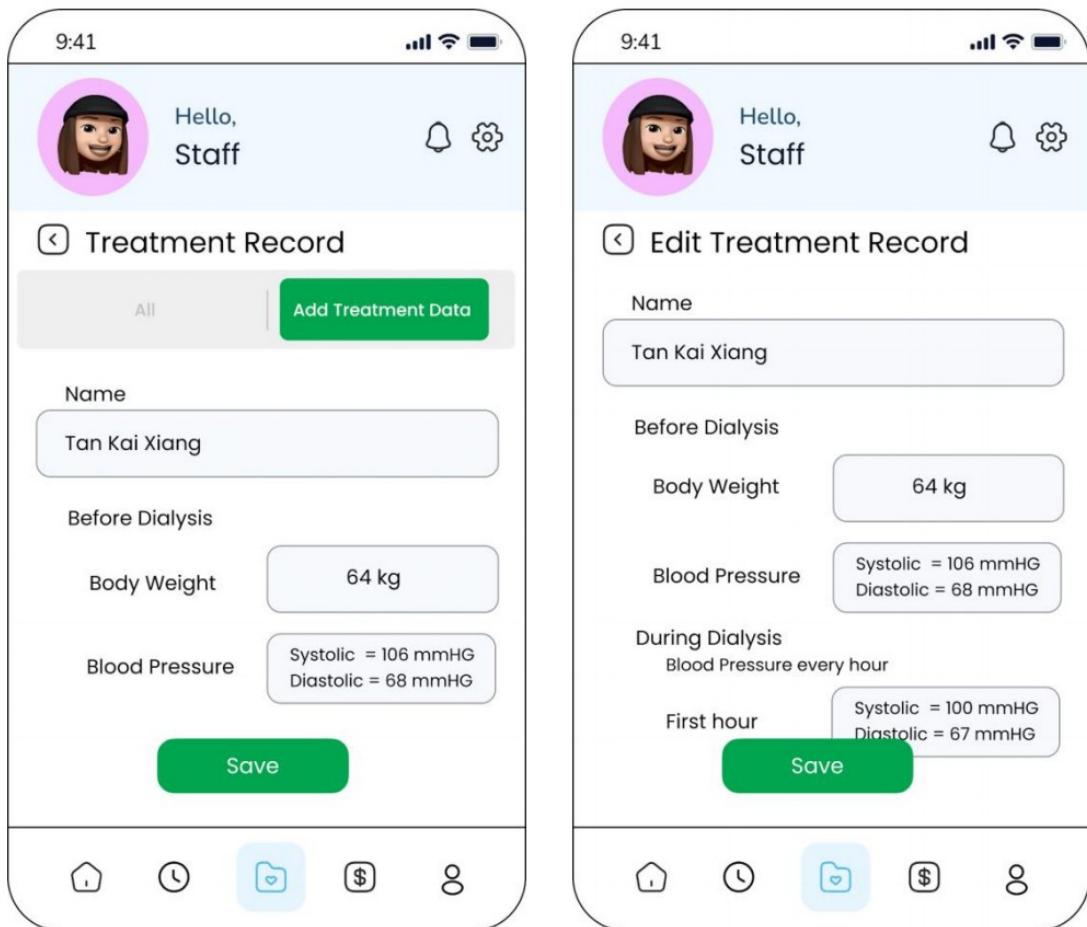


Figure 3. 24 Add Treatment Record

Figure 3. 25 Edit Treatment Record Interface

3.1.6 Manage Directory

Patients can view other dialysis center information in Directory Interface as in Figure 3.26 to ease the patients finding the nearest dialysis center when the patients go on holiday or have to work outstation. Figure 3.27 shows the Manage Directory Interface which Dialysis staff can edit the other dialysis center information. Dialysis staff also can add a new directory by clicking the Add Directory button and the Add Directory Interface will be displayed which allows staff to fill in the dialysis center information as in Figure 3.28.

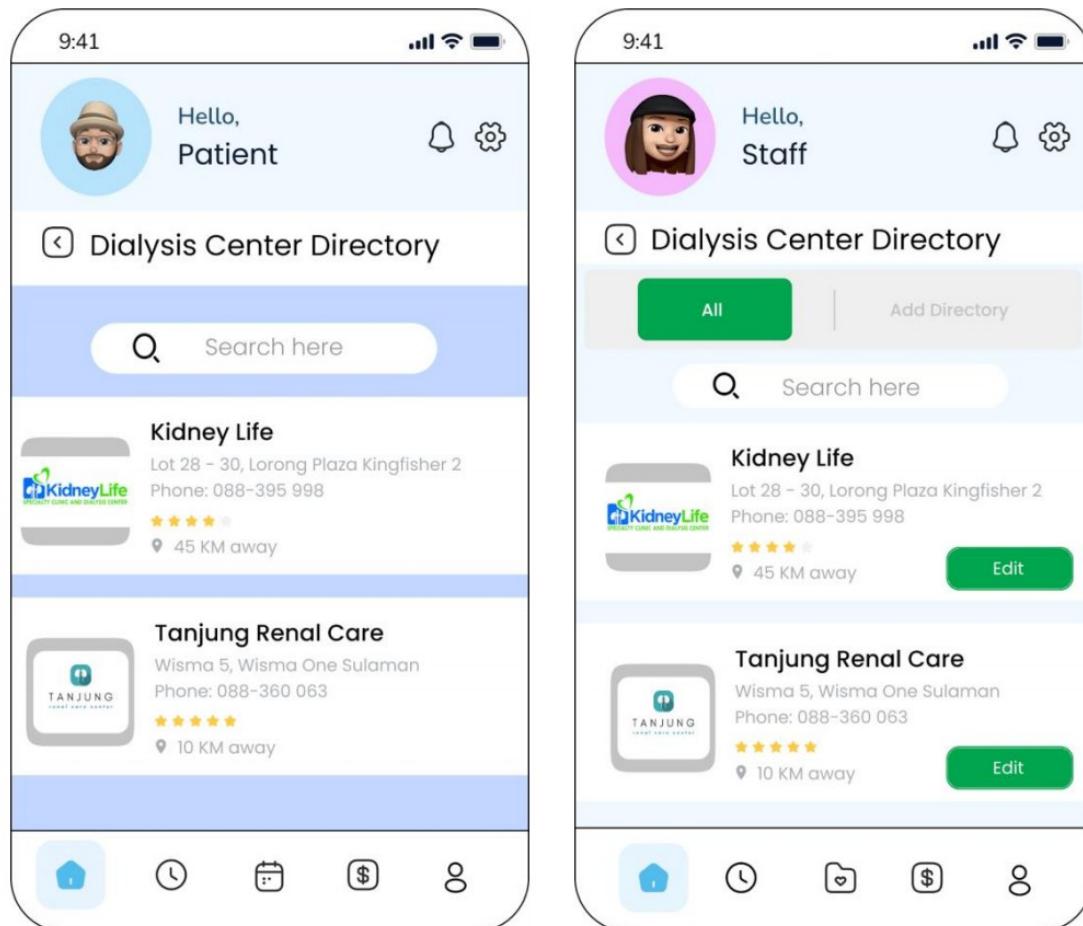


Figure 3. 26 Directory Interface

Figure 3. 27 Manage Directory Interface

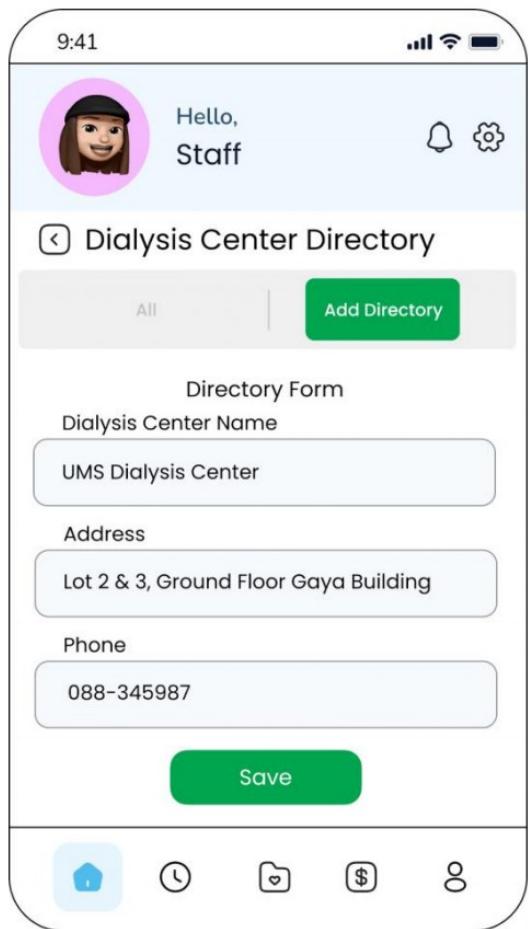


Figure 3. 28 Add Directory Interface

3.5.7 Manage Payment

Patients can view their dialysis slot bill in Payment Page Interface and also can make payments by clicking on the Pay Bill button as shown in Figure 3.29. Once the patients click on the Pay Bill button, PayBill Interface will be displayed and patients can make payment for the bill easily. There are three payment method option provided that patients can select. For dialysis staff, they can view all the patients' bills in Staff Payment Page Interface, refer Figure 3.31 as well as edit the bill by clicking the Edit button shown in the interface. Dialysis staff also can add patients' bills by clicking the Add Bill button and Add Bill Interface as in Figure 3.32 will be shown. The dialysis staff will fill in the bill details and click on the Save button to add the bill and send it to the patient Figure 3.33 shows the View Payment Interface for the dialysis staff which allows dialysis staff to view all the patients' bill payments that have been paid.

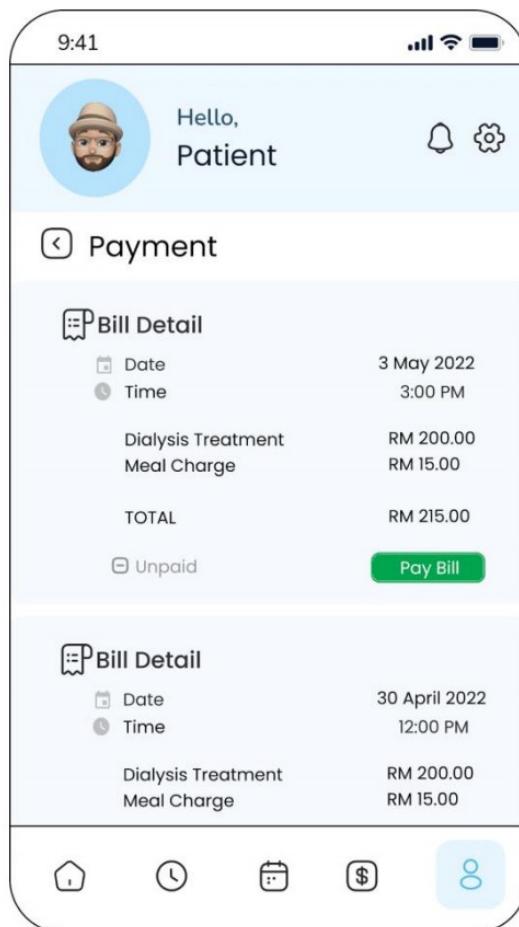


Figure 3. 29 Payment Page Interface

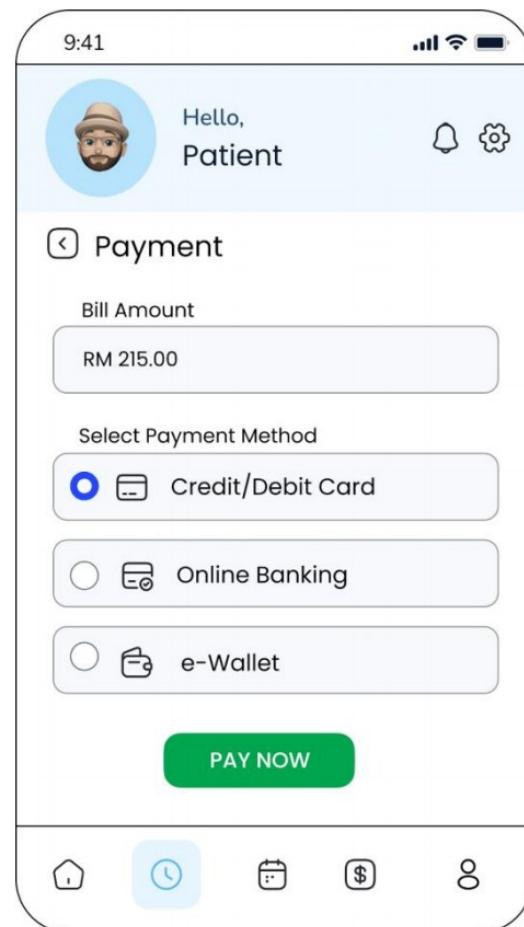


Figure 3. 30 PayBill Interface

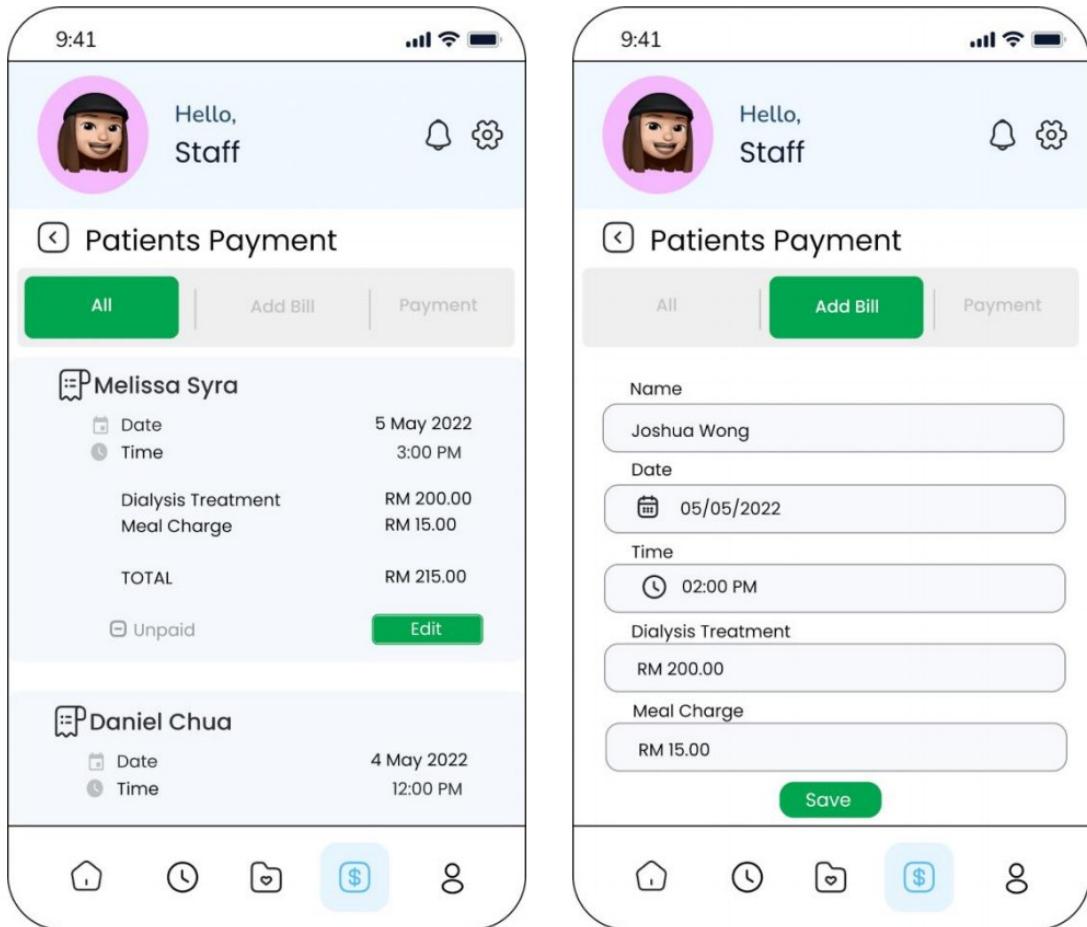


Figure 3. 31 Staff Payment Page Interface

Figure 3. 32 Add Bill Interface

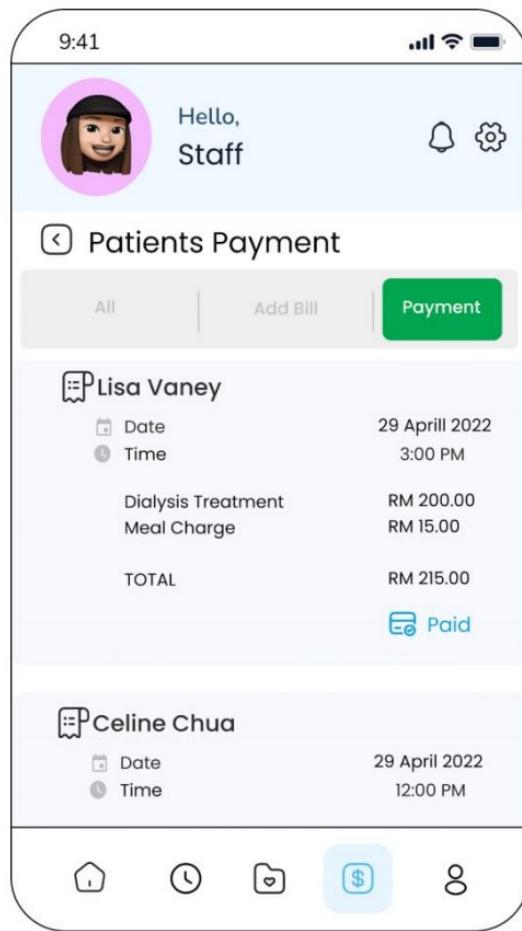


Figure 3. 33 View Payment Interface

3.6 Testing Plan

Testing is a crucial phase in the software development to ensure that the system is free from any defects and errors. It also helps developer in detect and eliminate risks and problems earlier so that the system can be deliver in good quality and ensure user satisfaction with the UI Testing. The type of testing will be used is the User Acceptance Testing (UAT). This testing type will be done by the end users to check if the system is well enough before the final deliverables.

Table 3. 16 Shows the UAT Form Example

No	Module	Activities	Status		Comments
			Success	Fail	
1	Profile	Edit profile			
2		View profile			
3	Appointment	Set appointment			
4		View appointment			
5		Approve appointment			
6		Request to reschedule appointment			
7		Approve or reject request			

8	Slot Time	Add slot time			
9		View slot time			
10		Approve slot time			
11		Request slot time			
12		Approve or reject request			
13	Directory	Search directory			
14		View directory			
15	Treatment Record	Insert treatment data			
16		Edit treatment data			
17		View treatment data			
18	Payment	Add bill			
19		View bill			
20		Make payment			
21		View payment			

This test has been performed by:

Name:

Signature:

Date:

3.7 Potential Use of Proposed Solution

This application is going to be developed to enhance efficiency of the dialysis centre management in handling the patients. Based on the user experience and the manual way of storing patients' treatment data, there are a number of requirements given to give these applications a chance in helping them to manage the dialysis operations well and also provide convenience to the kidney patients.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

Chapter 4 discussing on the development, implementation, and testing of myDialysis Application. This application is implemented for dialysis center staff, hospital staff, and dialysis center patients. Visual Studio Code and Android Studio are the tools used to develop this application.

4.2 Development Tools

No	Tool	Purpose
1	Visual Studio Code	For scripting, to create interfaces
2	Android Studio	To run the mobile emulator
3	DB Browser for SQLite	To view the database stored

Table 4.1 Tools used to develop the application.

Table 4.1 shows the tools used in the development of the myDialysis application.

4.3 Implementation Process

Implementation is a process to record all the steps in developing the application. This application is using mobile as a device. Therefore, creating and setting up a mobile emulator on Android Studio is vital to view and run the application while under the development. Before the development process takes part, the mobile emulator created on Android Studio will be connected to Visual Studio Code where all the coding for the application will be created by using a dart programming language.

4.3.1 Database Implementation

Setting up backend services such as connecting the project to the database is necessary for any software project development. This application uses SQLite to store the database. SQLite is an in-process library that implements a self-contained, zero-configuration, and transactional SQL database engine. It means that SQLite has minimal support from external libraries and no installation and setup needed, it just needs to add its dependencies to the project which is easy to get done by copying and pasting it. Moreover, SQLite is a serverless database that store, read, and writes data directly from the database files on disk which is convenient for the client because there is no need to set up any client-server connection, and all the data in SQLite are stored locally on the device.

As mentioned above, SQLite needs zero configuration. Hence, an SQLite plugin is necessary to be added to the project as a dependency under *pubspec.yaml* files which allow to access SQLite database in both Android and iOS operating systems. SQLite also requires a model class that represents the data object, a database helper class to perform CRUD operation, and a path provider plugin that helps to find commonly used locations on the file system of the device.

```
pubspec.yaml M X databaseHelper.dart userModel.dart
pubspec.yaml
  Camel Debug with JBang

  1 name: mydialysis_app
  2 description: A new Flutter project.

  3
  4 publish_to: 'none' # Remove this line if you wish to publish to pub.dev
  5 # The following defines the version and build number for your application.
  6 version: 1.0.0+1

  7
  8 environment:
  9   sdk: '>=2.18.4 <3.0.0'

 10
 11 dependencies:
 12   flutter:
 13     sdk: flutter
 14   cupertino_icons: ^1.0.2
 15   sqflite: ^2.2.0+3
 16   firebase_auth: ^4.1.3
 17   firebase_core: ^2.3.0
 18   cloud_firestore: ^4.1.0
 19   google_nav_bar: ^5.0.6
 20   intl: ^0.18.0
 21   path_provider: ^2.0.11
 22   shared_preferences: ^2.0.17
 23   wc_form_validators: ^1.0.0
 24   toast: ^0.3.0

 25
 26 dev_dependencies:
 27   flutter_test:
```

PROBLEMS 314 OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE

PS C:\Users\User\Flutter Project\mydialysis-app-new-sqlite>

Figure 4. 1 *pubspec.yaml* file

Figure 4.1 shows all the dependencies added to the project in *pubspec.yaml* file.

```
databaseHelper.dart pubspec.yaml userModel.dart
lib > db > databaseHelper.dart > DatabaseHelper > insertUser
1 // ignore_for_file: prefer_conditional_assignment, unused_field, unused_local_variable, prefer_collection_literals
2
3 import 'package:sqflite/sqflite.dart';
4 import 'dart:async';
5 import 'dart:io';
6 import 'package:path_provider/path_provider.dart';
7 import 'package:mydialysis_app/model/userModel.dart';
8
9 class DatabaseHelper {
10     static DatabaseHelper? _databaseHelper; //singleton dbhelper
11     static Database? _database;
12
13     String userTable = 'user_table';
14     String coluid = 'uid';
15     String coluname = 'uname';
16     String colupwd = 'upwd';
17     String colucpwd = 'ucpwd';
18     String coluphoneNum = 'uphoneNum';
19     String coluemail = 'uemail';
20     String coludob = 'udob';
21     String coluaddress = 'uaddress';
22     String colugivenCode = 'ugivenCode';
23     String colurole = 'urole';
24
25     DatabaseHelper._createInstance(); //named constructor to create instance of dbhelper
26
27     factory DatabaseHelper() {
28         if (_databaseHelper == null) {
29             _databaseHelper =
30                 DatabaseHelper._createInstance(); //executed only once, singleton obj
31         }
32         return _databaseHelper!;
33     }
34
35     Future<Database> get database async {
36         if (_database == null) {
37             _database = await initializeDatabase();
38         }
39         return _database;
40     }
41
42     Future<Database> initializeDatabase() async {
43         //get directory path
44         Directory directory = await getApplicationDocumentsDirectory();
45         String path = directory.path + '/myDialysis.db'; //db name
46         print(path);
47
48         //open/create db at given path
49         var myDatabase = await openDatabase(path, version: 1, onCreate: _createDB);
50         return myDatabase;
51     }
}
```

Figure 4. 2 Database Helper (initialize database)

Figure 4.2 shows the coding to initialize the SQLite database in *databaseHelper.dart* file.

```

141  //creating db(table and fields)
142  void _createDB(Database db, int newVersion) async {
143    //user table
144    String table1 =
145      'CREATE TABLE $userTable($coluid INTEGER PRIMARY KEY AUTOINCREMENT, $colname TEXT NULL, $colnic TEXT UNIQUE , $colupwd TEXT NULL, $colcpwd
146      TEXT NULL, $colphonenumber TEXT NULL, $colemail TEXT NULL UNIQUE, $coludob TEXT NULL, $coluaddress TEXT NULL, $colugivenCode TEXT NULL, $colurole
147      TEXT NULL)';
148    //slot table
149    String table2 =
150      'CREATE TABLE $slotTable($colsid INTEGER PRIMARY KEY AUTOINCREMENT, $coldate DATE NULL, $colstime TIME NULL, $colsstatus TEXT NULL, $colsrdate
151      DATE NULL, $colsrtime TIME NULL, $colsstatus TEXT NULL, $colsReason TEXT NULL, $colsppname TEXT NULL)';
152    //treatment table
153    String table3 =
154      'CREATE TABLE $treatmentTable($coltrid INTEGER PRIMARY KEY AUTOINCREMENT, $colbw TEXT NULL, $colbp TEXT NULL, $colbhr TEXT NULL, $colbtemp
155      TEXT NULL, $colbpi TEXT NULL, $colbp2 TEXT NULL, $colbp3 TEXT NULL, $colbp4 TEXT NULL, $colbp5 TEXT NULL, $colbri TEXT NULL, $colbhr2 TEXT
156      NULL, $colbhr3 TEXT NULL, $colbhr4 TEXT NULL, $colbhr5 TEXT NULL, $colbw TEXT NULL, $colbp TEXT NULL, $colbhr TEXT NULL, $colbtemp TEXT NULL,
157      $colbpname TEXT NULL, $coltrdate DATE NULL, $coltrtime TIME NULL)';
158    //payment table
159    String table4 =
160      'CREATE TABLE $paymentTable($colbid INTEGER PRIMARY KEY AUTOINCREMENT, $colbdate DATE NULL, $colbtime TIME NULL, $coldtpice TEXT NULL,
161      $colmealprice TEXT NULL, $coltotalprice TEXT NULL, $colbstatus TEXT NULL, $colbdate DATE NULL, $colbtime TIME NULL, $colpamount FLOAT NULL,
162      $colppname TEXT NULL)';
163    //directory table
164    String table5 =
165      'CREATE TABLE $directoryTable($coldid INTEGER PRIMARY KEY AUTOINCREMENT, $coldname TEXT NULL, $coldaddress TEXT NULL, $coldcnumber TEXT NULL,
166      $coldlogo VARCHAR(50) NULL, $coldka TEXT NULL, $coldopenhr TEXT NULL, $coldclosehr TEXT NULL, $coldrating TEXT NULL)';
167    //appointment table
168    String table6 =

```

Figure 4. 3 Database Helper (creating database table)

Figure 4.3 shows the coding to create table in Sqlite database in *databaseHelper.dart* file.

The screenshot shows a code editor with the file `databaseHelper.dart` open. The code implements a Database Helper class with methods for creating the database, fetching users, logging in users, inserting users, and updating users.

```
53 //creating db(table and fields)
54 void _createDB(Database db, int newVersion) async {
55   await db.execute(
56     'CREATE TABLE $userTable($coluid INTEGER PRIMARY KEY AUTOINCREMENT, $coluname TEXT NULL, $colupwd
57     TEXT NULL, $colucpwd TEXT NULL, $coluphoneNum TEXT NULL, $coluemail TEXT NULL, $coludob TEXT NULL,
58     $coluaddress TEXT NULL, $colugivenCode TEXT NULL, $colurole TEXT NULL)');
59
60 //fetch operation: get all data from db
61 Future<List<UserModel>> getAllUser() async {
62   final data = await _database!.query(userTable);
63   List<UserModel> result = data.map((e) => UserModel.fromJson(e)).toList();
64   return result;
65 }
66
67 //fetch one user only
68 Future<List<Map<String, dynamic>>> getUserbyId(int uid) async {
69   final result = await _database!
70     .rawQuery('SELECT * FROM $userTable WHERE $coluid = $uid');
71   return result;
72 }
73
74 //fetch one user only by email
75 Future<List<Map<String, dynamic>>> getUserByEmail(String uemail) async {
76   final result = await _database!
77     .rawQuery('SELECT * FROM $userTable WHERE $coluemail = $uemail');
78   return result;
79 }
80
81 //fetch one user only by email and password
82 Future<UserModel?> getLoginUser(
83   String uemail, String upwd) async {
84   final result = await _database!.rawQuery(
85     'SELECT * FROM $userTable WHERE $coluemail = $uemail AND $colupwd = $upwd');
86   if (result.isNotEmpty) {
87     return UserModel.fromJson(result.first);
88   }
89   return null;
90 }
91
92 //insert operation: insert data obj from db
93 Future<int> insertUser(Map<String, dynamic> row) async {
94   final result = await _database!.insert(userTable, row);
95   return result;
96 }
97
98 //update operation: update data obj from db & save to db
99 Future<int> updateUser(int uid, Map<String, dynamic> row) async {
100   final result = await _database!
101     .update(userTable, row, where: '$coluid = ?', whereArgs: [uid]);
102   return result;
103 }
```

At the bottom of the screen, the terminal shows the command: `PS C:\Users\User\Flutter Project\mydialysis-app-new-sqlite>`.

Figure 4.4 Database Helper (CRUD Operation)

Figure 4.4 shows the coding of the CRUD operation of the database in `databaseHelper.dart` file.

Below are the coding for the model class of this application that represents the variables and objects for all the table in database. Figure 4.4 shows all the model class files included in the project.

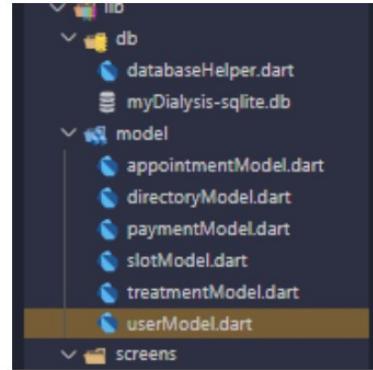


Figure 4. 5 Model Class

```

import 'dart:core';

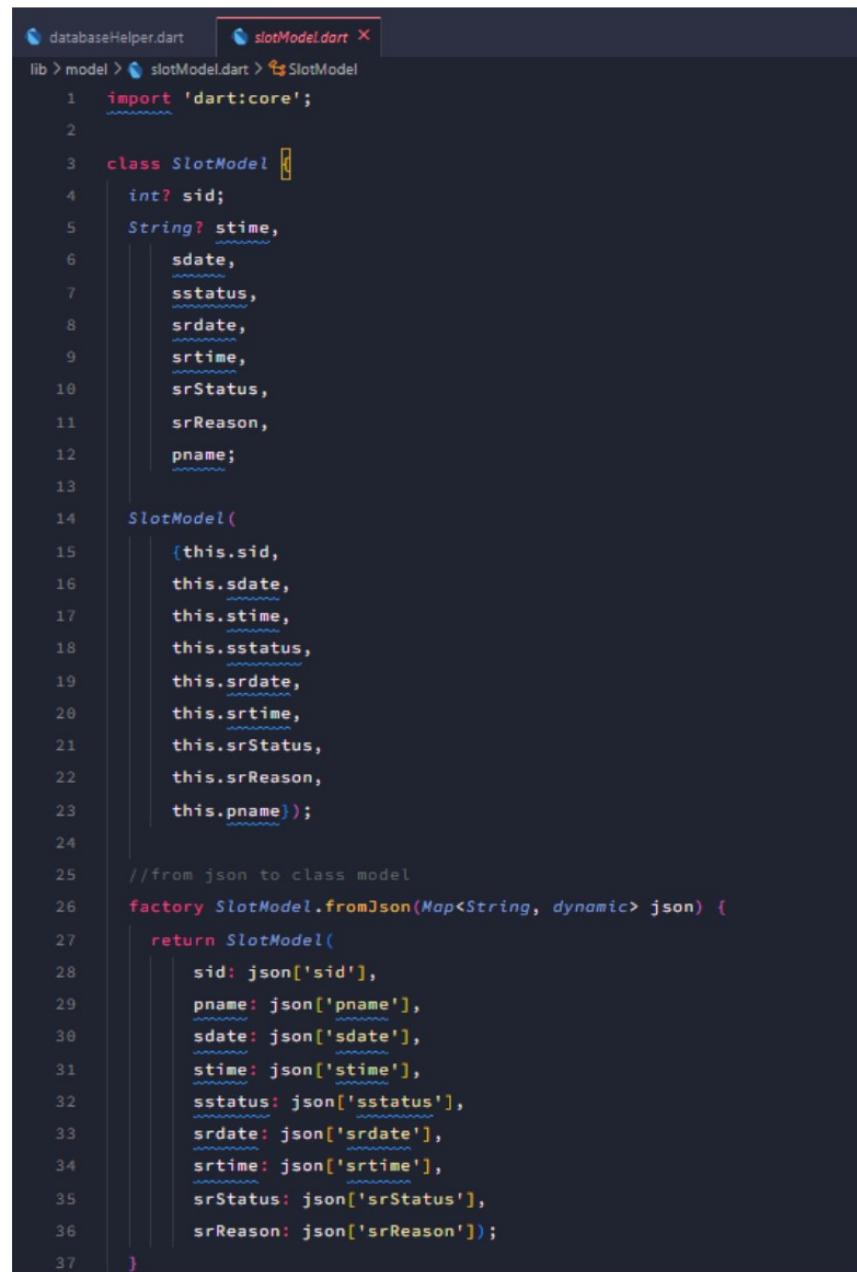
class UserModel {
  String? uid,
  uname,
  upwd,
  ucpwd,
  uphoneNum,
  uemail,
  udob,
  uaddress,
  ugivencode,
  urole;

  UserModel(
    this.uid,
    this.uname,
    this.upwd,
    this.ucpwd,
    this.uphoneNum,
    this.uemail,
    this.udob,
    this.uaddress,
    this.ugivencode,
    this.ulrole);

  //from json to class model
  factory UserModel.fromJson(Map<String, dynamic> json) {
    return UserModel(
      uid: json['uid'],
      uname: json['uname'],
      upwd: json['upwd'],
      ucpwd: json['ucpwd'],
      uphoneNum: json['uphoneNum'],
      uemail: json['uemail'],
      udob: json['udob'],
      uaddress: json['uaddress'],
      ugivencode: json['ugivencode'],
      urole: json['ulrole']);
  }
}

```

Figure 4. 6 User Model Class Coding



The screenshot shows a code editor with a dark theme. The file is named `slotModel.dart`. The code defines a class `SlotModel` with properties for sid, sdate, sstatus, srdate, srtime, srStatus, srReason, and pname. It includes a constructor and a factory method for creating objects from JSON data.

```
1 import 'dart:core';
2
3 class SlotModel {
4   int? sid;
5   String? sdate,
6   sstatus,
7   srdate,
8   srtime,
9   srStatus,
10  srReason,
11  pname;
12
13
14  SlotModel(
15    this.sid,
16    this.sdate,
17    this.sstatus,
18    this.srdate,
19    this.srtime,
20    this.srStatus,
21    this.srReason,
22    this.pname);
23
24
25 //from json to class model
26 factory SlotModel.fromJson(Map<String, dynamic> json) {
27   return SlotModel(
28     sid: json['sid'],
29     pname: json['pname'],
30     sdate: json['sdate'],
31     sstatus: json['sstatus'],
32     srdate: json['srdate'],
33     srtime: json['srtime'],
34     srStatus: json['srStatus'],
35     srReason: json['srReason']);
36
37 }
```

Figure 4. 7 Slot Model Class Coding

The screenshot shows the Visual Studio Code interface with the following details:

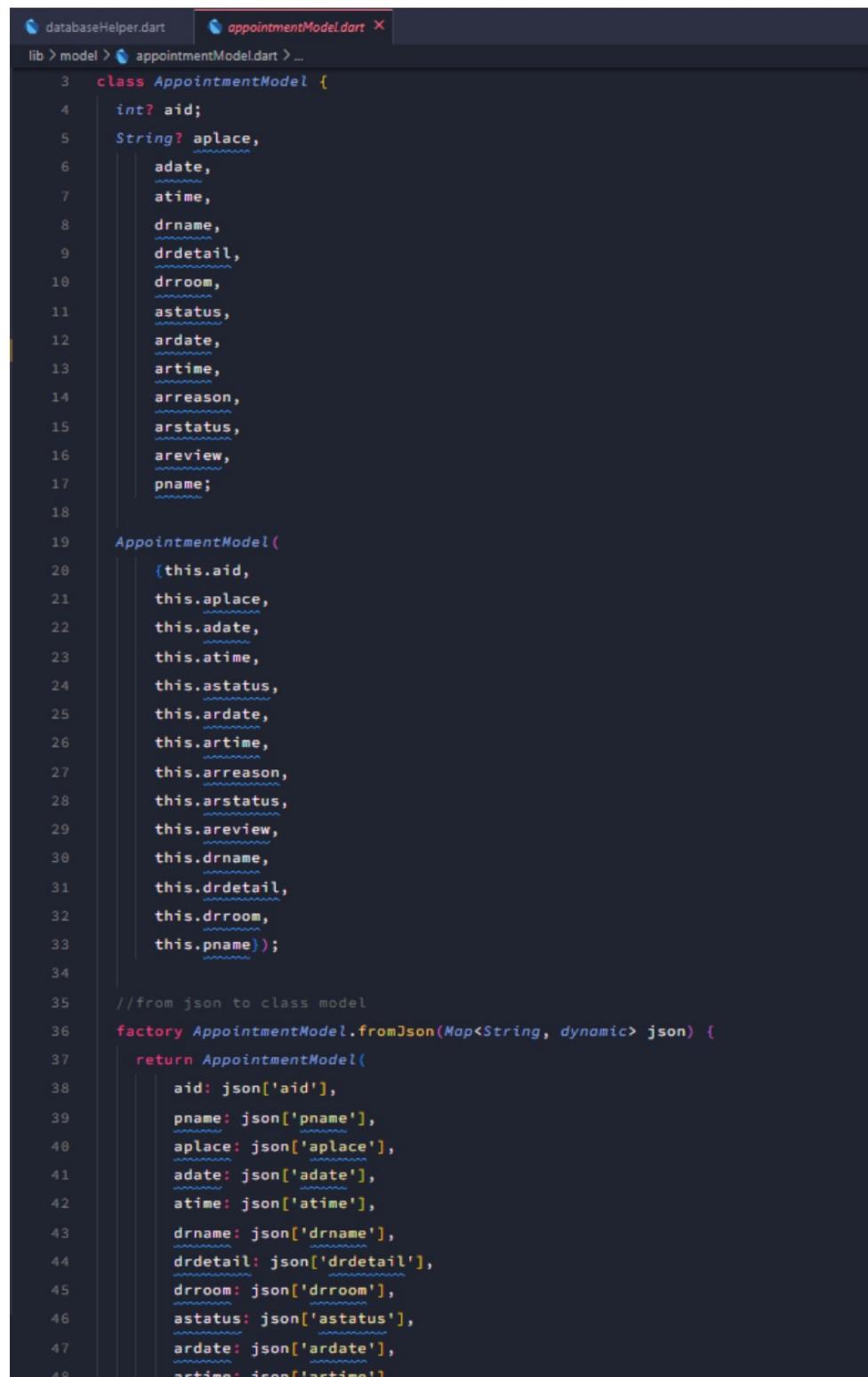
- EXPLORER** panel on the left displays the project structure of "MYDIALYSIS-APP-NEW-SQLITE". It includes folders for .dart_tool, .idea, .vscode, android, build, images, ios, lib (containing db, model, screens, dialysis staff, directory ds, payment ds, profile ds, slot time ds, tabs ds, treatment record ds, widget ds), patient, services, test, web, windows, and various configuration files like .gitignore, .metadata, analysis_options.yaml, mydialysis_app.iml, pubspec.lock, and pubspec.yaml.
- Treatment Model Class Coding** in the center shows the code for the `treatmentModel.dart` file. The code defines a class `TreatmentModel` with properties: `trid`, `bbweight`, `bbpressure`, `bhrate`, `btemp`, `dbpressure1`, `dbpressure2`, `dbpressure3`, `dbpressure4`, `dbpressure5`, `dhrate1`, `dhrate2`, `dhrate3`, `dhrate4`, `dhrate5`, `abweight`, `abpressure`, `ahrate`, `atemp`, `pname`, `trdate`, and `trtime`. The constructor `TreatmentModel` initializes these properties.

```
import 'dart:core';

class TreatmentModel {
    int? trid;
    String? bbweight,
        bbpressure,
        bhrate,
        btemp,
        dbpressure1,
        dbpressure2,
        dbpressure3,
        dbpressure4,
        dbpressure5,
        dhrate1,
        dhrate2,
        dhrate3,
        dhrate4,
        dhrate5,
        abweight,
        abpressure,
        ahrate,
        atemp,
        pname,
        trdate,
        trtime;

    TreatmentModel(
        {this.trid,
        this.bbweight,
        this.bbpressure,
        this.bhrate,
        this.btemp,
        this.dbpressure1,
        this.dbpressure2,
        this.dbpressure3,
        this.dbpressure4,
        this.dbpressure5,
        this.dhrate1,
        this.dhrate2,
        this.dhrate3,
        this.dhrate4,
        this.dhrate5,
        this.abweight,
        this.abpressure,
        this.ahrate,
        this.atemp,
        this.pname,
        this.trdate,
        this.trtime});
}
```

Figure 4. 8 Treatment Record Model Class Coding



The screenshot shows a code editor with two tabs: `databaseHelper.dart` and `appointmentModel.dart`. The `appointmentModel.dart` tab is active, displaying the following Dart code:

```
lib > model > appointmentModel.dart ...
3   class AppointmentModel {
4     int? aid;
5     String? aplace,
6         adate,
7         atime,
8         drname,
9         drdetail,
10        drroom,
11        astatus,
12        ardate,
13        artime,
14        arreason,
15        arstatus,
16        areview,
17        pname;
18
19   AppointmentModel(
20     {this.aid,
21      this.aplace,
22      this.adate,
23      this.atime,
24      this.astatus,
25      this.ardate,
26      this.artime,
27      this.arreason,
28      this.arstatus,
29      this.areview,
30      this.drname,
31      this.drdetail,
32      this.drroom,
33      this.pname});
34
35 //from json to class model
36 factory AppointmentModel.fromJson(Map<String, dynamic> json) {
37   return AppointmentModel(
38     aid: json['aid'],
39     pname: json['pname'],
40     aplace: json['aplace'],
41     adate: json['adate'],
42     atime: json['atime'],
43     drname: json['drname'],
44     drdetail: json['drdetail'],
45     drroom: json['drroom'],
46     astatus: json['astatus'],
47     ardate: json['ardate'],
48     artime: json['artime'],
```

Figure 4. 9 Appointment Model Class Coding



The screenshot shows a code editor with a dark theme. The title bar displays 'paymentModel.dart' and 'databaseHelper.dart'. The main area contains the code for the 'PaymentModel' class.

```
1 import 'dart:core';
2
3 class PaymentModel {
4     int? bid;
5     String? bdate,
6         btime,
7         dtprice,
8         mealprice,
9         totalprice,
10    bstatus,
11    pdate,
12    ptime,
13    pamount,
14    pname;
15
16 PaymentModel(
17     {this.pname,
18     this.bid,
19     this.bdate,
20     this.btime,
21     this.dtprice,
22     this.mealprice,
23     this.totalprice,
24     this.bstatus,
25     this.pdate,
26     this.ptime,
27     this.pamount,});
28
29 //from json to class model
30 factory PaymentModel.fromJson(Map<String, dynamic> json) {
31     return PaymentModel(
32         pname: json['pname'],
33         bid: json['bid'],
34         bdate: json['bdate'],
35         btime: json['btime'],
36         dtprice: json['dtprice'],
37         mealprice: json['mealprice'],
38         totalprice: json['totalprice'],
39         bstatus: json['bstatus'],
40         pdate: json['pdate'],
41         ptime: json['ptime'],
42         pamount: json['pamount'],
43     );
44 }
45 }
```

Figure 4. 10 Payment Model Class Coding



The screenshot shows a code editor with a dark theme. At the top, there are two tabs: 'databaseHelper.dart' and 'directoryModel.dart'. The 'directoryModel.dart' tab is active, showing the following Dart code:

```
lib > model > directoryModel.dart > DirectoryModel
1 import 'dart:core';
2
3 class DirectoryModel {
4     int? did;
5     String? dname, daddress, dcnumber, dkm, dlogo, dopenhr, dclosehr, drating;
6
7     DirectoryModel(
8         {this.did,
9          this.dname,
10         this.daddress,
11         this.dcnumber,
12         this.dlogo,
13         this.dkm,
14         this.dopenhr,
15         this.dclosehr,
16         this.drating});
17
18 //from json to class model
19 factory DirectoryModel.fromJson(Map<String, dynamic> json) {
20     return DirectoryModel(
21         did: json['did'],
22         dname: json['dname'],
23         daddress: json['daddress'],
24         dcnumber: json['dcnumber'],
25         dlogo: json['dlogo'],
26         dkm: json['dkm'],
27         dopenhr: json['dopenhr'],
28         dclosehr: json['dclosehr'],
29         drating: json['drating']);
30     }
31 }
32 }
```

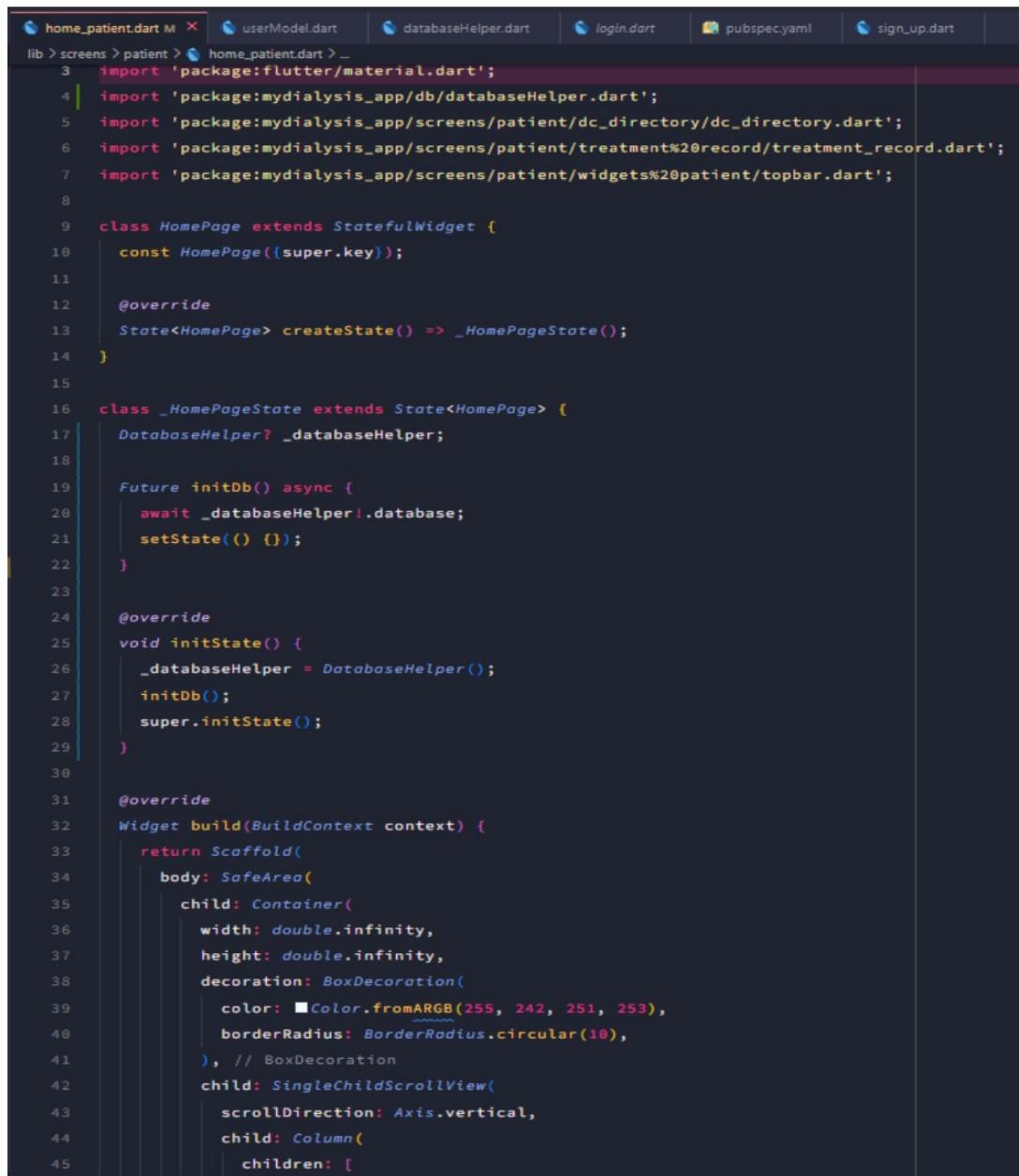
Figure 4. 11 Directory Model Class Coding

4.3.2 Interface Development

Once the database implementation is done, the development of the user interface and connection for the database and UI can be started.

4.3.2.1 Home Page

Home Page for Patient



```
home_patient.dart M userModel.dart databaseHelper.dart login.dart pubspec.yaml sign_up.dart
lib > screens > patient > home_patient.dart > ...
3 import 'package:flutter/material.dart';
4 import 'package:mydialysis_app/db/databaseHelper.dart';
5 import 'package:mydialysis_app/screens/patient/dc_directory/dc_directory.dart';
6 import 'package:mydialysis_app/screens/patient/treatment%20record/treatment_record.dart';
7 import 'package:mydialysis_app/screens/patient/widgets%20patient/topbar.dart';
8
9 class HomePage extends StatefulWidget {
10   const HomePage({super.key});
11
12   @override
13   State<HomePage> createState() => _HomePageState();
14 }
15
16 class _HomePageState extends State<HomePage> {
17   DatabaseHelper? _databaseHelper;
18
19   Future initDb() async {
20     await _databaseHelper!.database;
21     setState(() {});
22   }
23
24   @override
25   void initState() {
26     _databaseHelper = DatabaseHelper();
27     initDb();
28     super.initState();
29   }
30
31   @override
32   Widget build(BuildContext context) {
33     return Scaffold(
34       body: SafeArea(
35         child: Container(
36           width: double.infinity,
37           height: double.infinity,
38           decoration: BoxDecoration(
39             color: Color.fromRGBO(255, 242, 251, 253),
40             borderRadius: BorderRadius.circular(10),
41           ), // BoxDecoration
42           child: SingleChildScrollView(
43             scrollDirection: Axis.vertical,
44             child: Column(
45               children: [
46                 Container(
47                   padding: EdgeInsets.all(10),
48                   child: Row(
49                     mainAxisAlignment: MainAxisAlignment.spaceBetween,
50                     children: [
51                       Text("Treatment Record"),
52                       Text("Treatment Record"),
53                     ],
54                   ),
55                 ),
56                 Container(
57                   padding: EdgeInsets.all(10),
58                   child: Row(
59                     mainAxisAlignment: MainAxisAlignment.spaceBetween,
60                     children: [
61                       Text("Treatment Record"),
62                       Text("Treatment Record"),
63                     ],
64                   ),
65                 ),
66               ],
67             ),
68           ),
69         ),
70       ),
71     );
72   }
73 }
```

Figure 4. 12 Home Page Coding

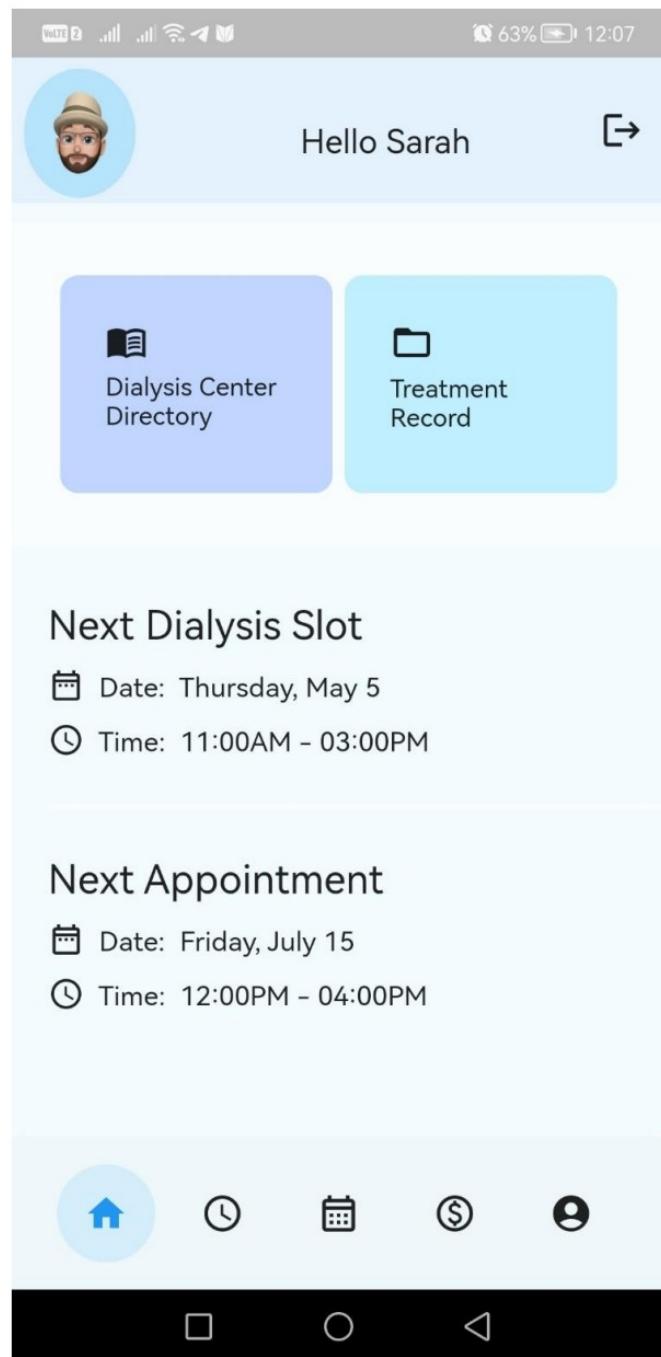
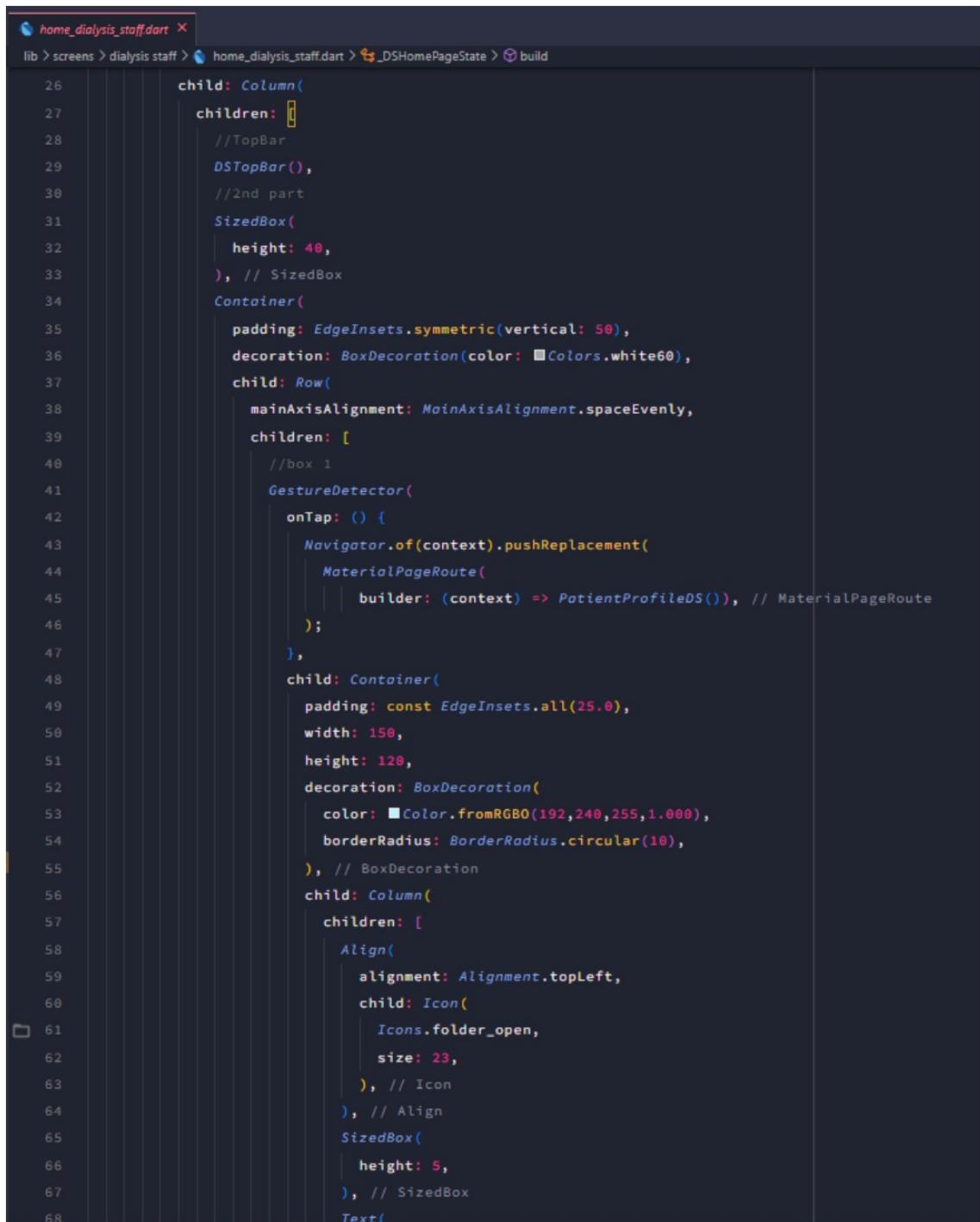


Figure 4. 13 Patient Home Page Interface

This page shows the patient's next dialysis slot and next appointment date and time. The patient also can access other dialysis center directory and their dialysis treatment record by clicking the button as shown in Figure 4.13.

Home Page for Dialysis Staff



The screenshot shows a code editor with the file `home_dialysis_staff.dart` open. The code is a Dart class for a home page screen. It uses the `Column`, `Row`, `Container`, `GestureDetector`, and `Navigator` widgets from the Flutter framework. The code defines a main layout with a top bar, a central container with a rounded rectangle, and a bottom section with an icon and text.

```
lib > screens > dialysis staff > home_dialysis_staff.dart > _DSHomePageState > build
26     child: Column(
27       children: [
28         //TopBar
29         DSTopBar(),
30         //2nd part
31         SizedBox(
32           height: 40,
33         ), // SizedBox
34         Container(
35           padding: EdgeInsets.symmetric(vertical: 50),
36           decoration: BoxDecoration(color: Colors.white60),
37           child: Row(
38             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
39             children: [
40               //box 1
41               GestureDetector(
42                 onTap: () {
43                   Navigator.of(context).pushReplacement(
44                     MaterialPageRoute(
45                       builder: (context) => PatientProfileDS()), // MaterialPageRoute
46                     );
47                 },
48                 child: Container(
49                   padding: const EdgeInsets.all(25.0),
50                   width: 150,
51                   height: 120,
52                   decoration: BoxDecoration(
53                     color: Color.fromRGBO(192, 240, 255, 1.000),
54                     borderRadius: BorderRadius.circular(10),
55                   ), // BoxDecoration
56                   child: Column(
57                     children: [
58                       Align(
59                         alignment: Alignment.topLeft,
60                         child: Icon(
61                           Icons.folder_open,
62                           size: 28,
63                         ), // Icon
64                         ), // Align
65                         SizedBox(
66                           height: 5,
67                         ), // SizedBox
68                         Text(
69                           "Patient Profile"
70                         ),
71                       Align(
72                         alignment: Alignment.topLeft,
73                         child: Icon(
74                           Icons.add,
75                           size: 28,
76                         ), // Icon
77                         ), // Align
78                         SizedBox(
79                           height: 5,
80                         ), // SizedBox
81                         Text(
82                           "Add New Patient"
83                         )
84                     ],
85                   ),
86                 ),
87               ),
88             ],
89           ),
90         ),
91       ],
92     ),
93   ),
94 }
```

Figure 4. 14 Dialysis Staff Home Page Coding

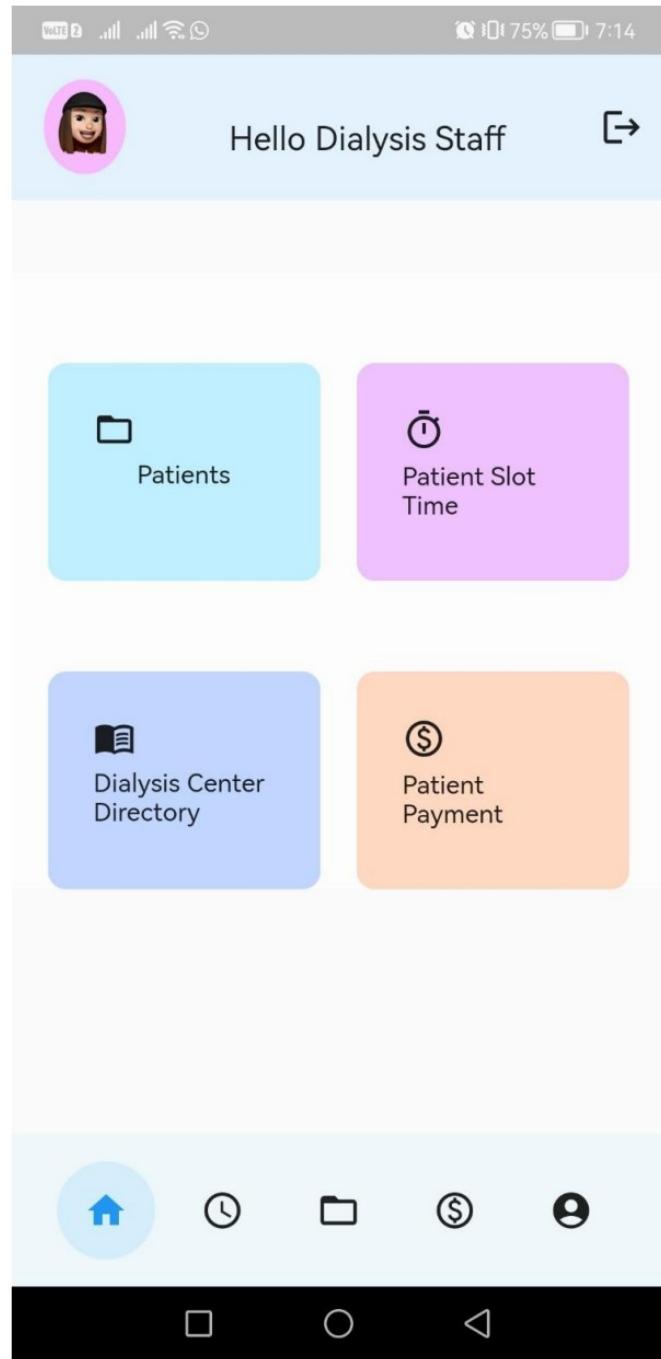
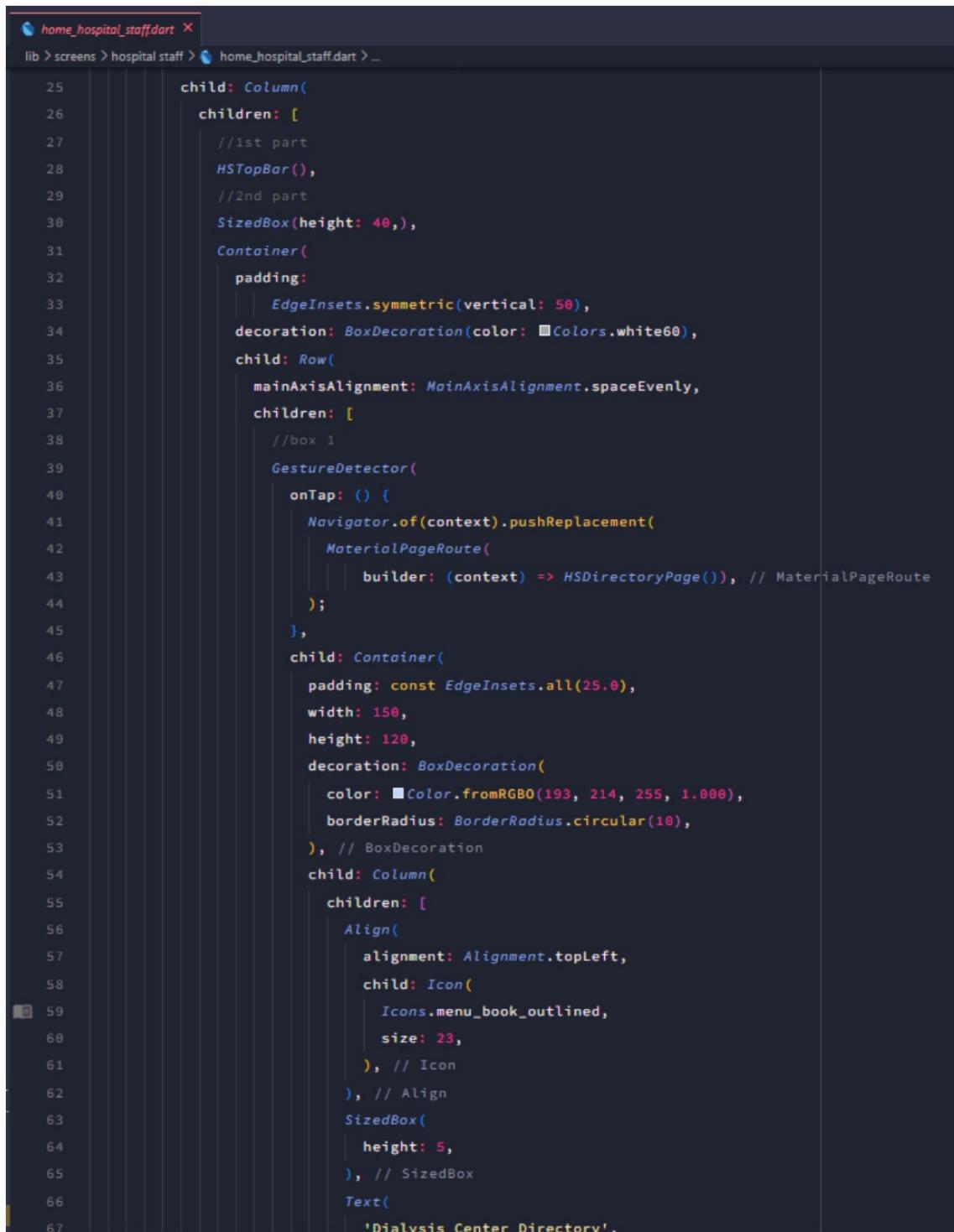


Figure 4. 15 Dialysis Staff Home Page Interface

This page shows Patients Profile button, Patient Slot Time button, Directory button, Payment button, and bottom bar button which user can access to the function/interface by clicking the button as shown in Figure 4.15.

Home Page for Hospital Dialysis



The screenshot shows a code editor with the file `home_hospital_staff.dart` open. The code is written in Dart and defines the UI for a hospital staff home page. It uses a `Column` widget as the main container, containing a `HSTopBar`, a `SizedBox` for height, and a `Container` with padding and a white decoration. Inside this container is a `Row` with space-between alignment, containing a `GestureDetector` that onTap navigates to `HSDirectoryPage`. The `Container` also contains a `Column` with an `Align` widget that aligns its child icon to the top-left and has a `SizedBox` with height 5, followed by a `Text` widget with the text 'Dialysis Center Directory'.

```
25     child: Column(
26       children: [
27         //1st part
28         HSTopBar(),
29         //2nd part
30         SizedBox(height: 40,),
31         Container(
32           padding: EdgeInsets.symmetric(vertical: 50),
33           decoration: BoxDecoration(color: Colors.white60),
34           child: Row(
35             mainAxisAlignment: MainAxisAlignment.spaceEvenly,
36             children: [
37               //box 1
38               GestureDetector(
39                 onTap: () {
40                   Navigator.of(context).pushReplacement(
41                     MaterialPageRoute(
42                       builder: (context) => HSDirectoryPage(), // MaterialPageRoute
43                     );
44                 },
45               ),
46               child: Container(
47                 padding: const EdgeInsets.all(25.0),
48                 width: 150,
49                 height: 120,
50                 decoration: BoxDecoration(
51                   color: Color.fromRGBO(193, 214, 255, 1.000),
52                   borderRadius: BorderRadius.circular(10),
53                 ), // BoxDecoration
54                 child: Column(
55                   children: [
56                     Align(
57                       alignment: Alignment.topLeft,
58                       child: Icon(
59                         Icons.menu_book_outlined,
60                         size: 23,
61                       ), // Icon
62                     ), // Align
63                     SizedBox(
64                       height: 5,
65                     ), // SizedBox
66                     Text(
67                       'Dialysis Center Directory'.

```

Figure 4. 16 Hospital Staff Home Page Coding

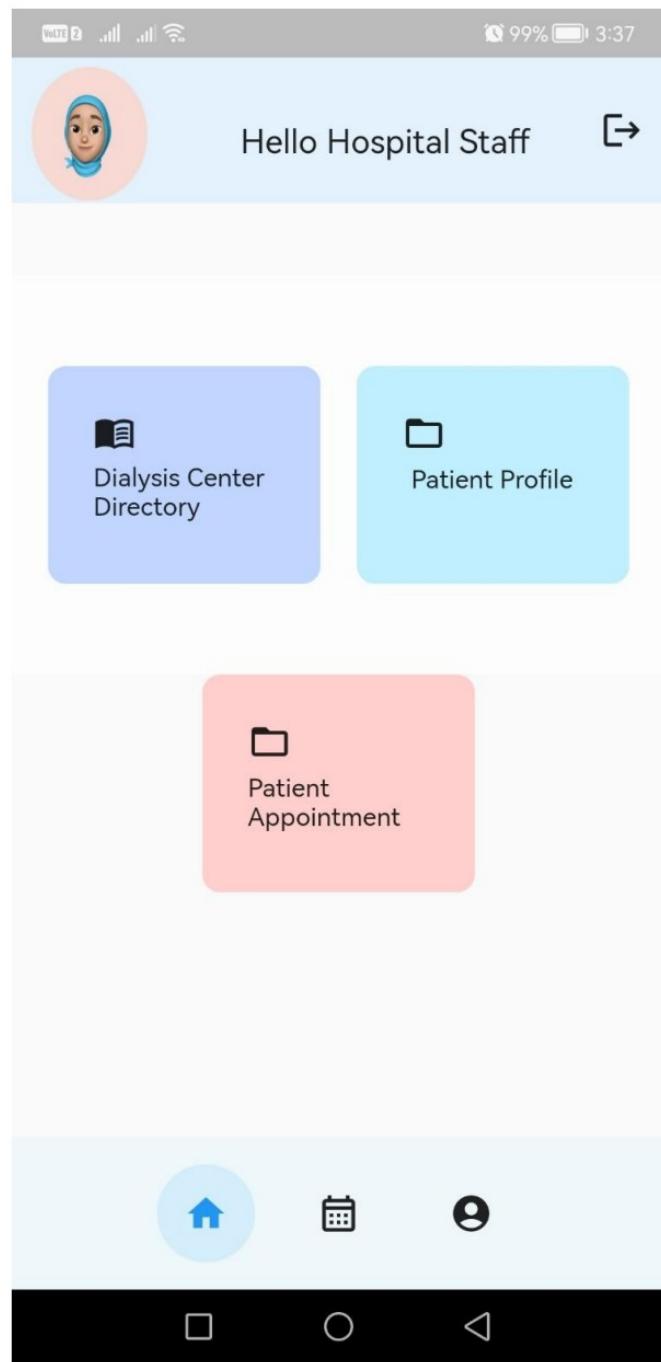


Figure 4. 17 Hospital Staff Home Page Interface

This page shows Patients Profile button, Directory button, and bottom bar button which user can access to the function/interface by clicking the button as shown in Figure 4.17.

4.3.2.2 Profile Page

Profile page interface for all users are similar except for the data the application retrieved from the database is based on the user who logged in into the application.



```
profile_patient.dart
lib > screens > patient > profile > profile_patient.dart > _ProfilePageState > build

69           // Name part
70           Row(
71             children: [
72               Text(
73                 'Name: ',
74                 style: TextStyle(fontSize: 18),
75               ), // Text
76               SizedBox(
77                 width: 80,
78               ), // SizedBox
79               Container(
80                 padding: EdgeInsets.all(8.0),
81                 child: Text('$currentUser',style: TextStyle(fontSize: 16)),
82               ), // Container
83             ],
84           ), // Row
85           SizedBox(
86             height: 20,
87           ), // SizedBox

88           // Phone number
89           Row(
90             children: [
91               Text(
92                 'Phone Number: ',
93                 style: TextStyle(fontSize: 18),
94               ), // Text
95               SizedBox(
96                 width: 8,
97               ), // SizedBox
98               Container(
99                 padding: EdgeInsets.all(8.0),
100                child: Text('$phoneNumber',style: TextStyle(fontSize: 16)),
101              ), // Container
102            ],
103          ), // Row
104          SizedBox(
105            height: 20,
106          ), // SizedBox
```

Figure 4. 18 Profile Page Coding

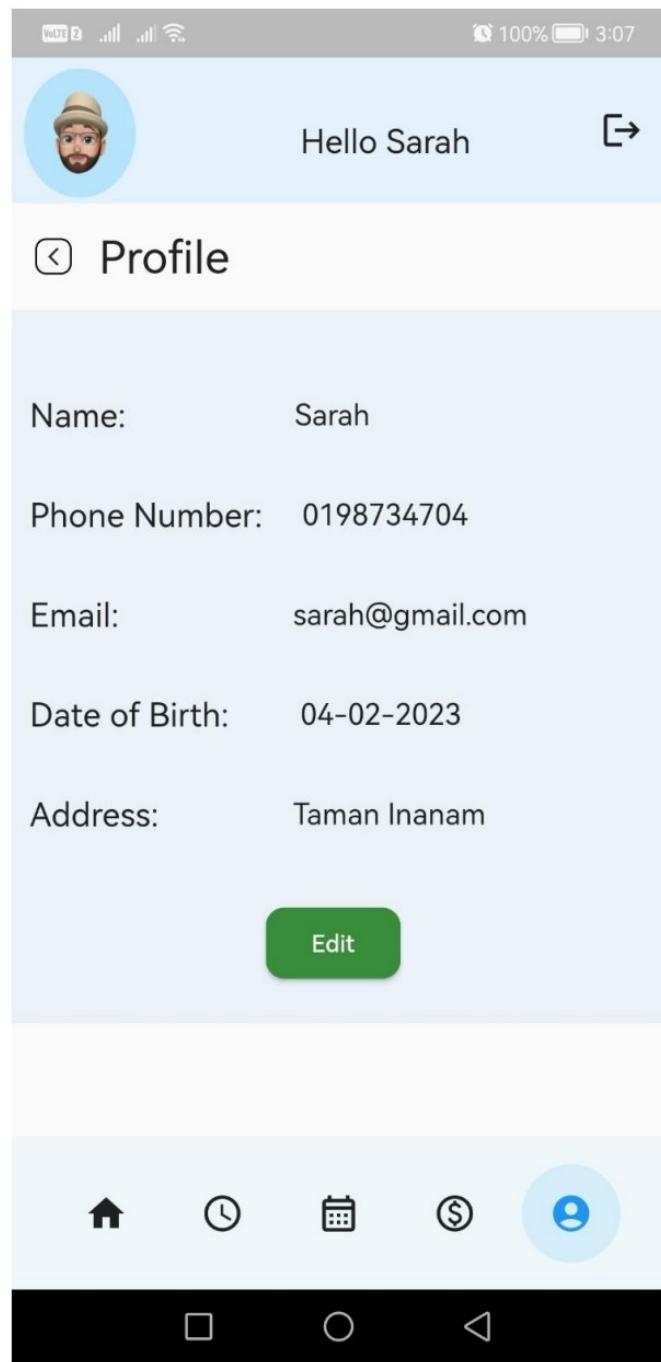
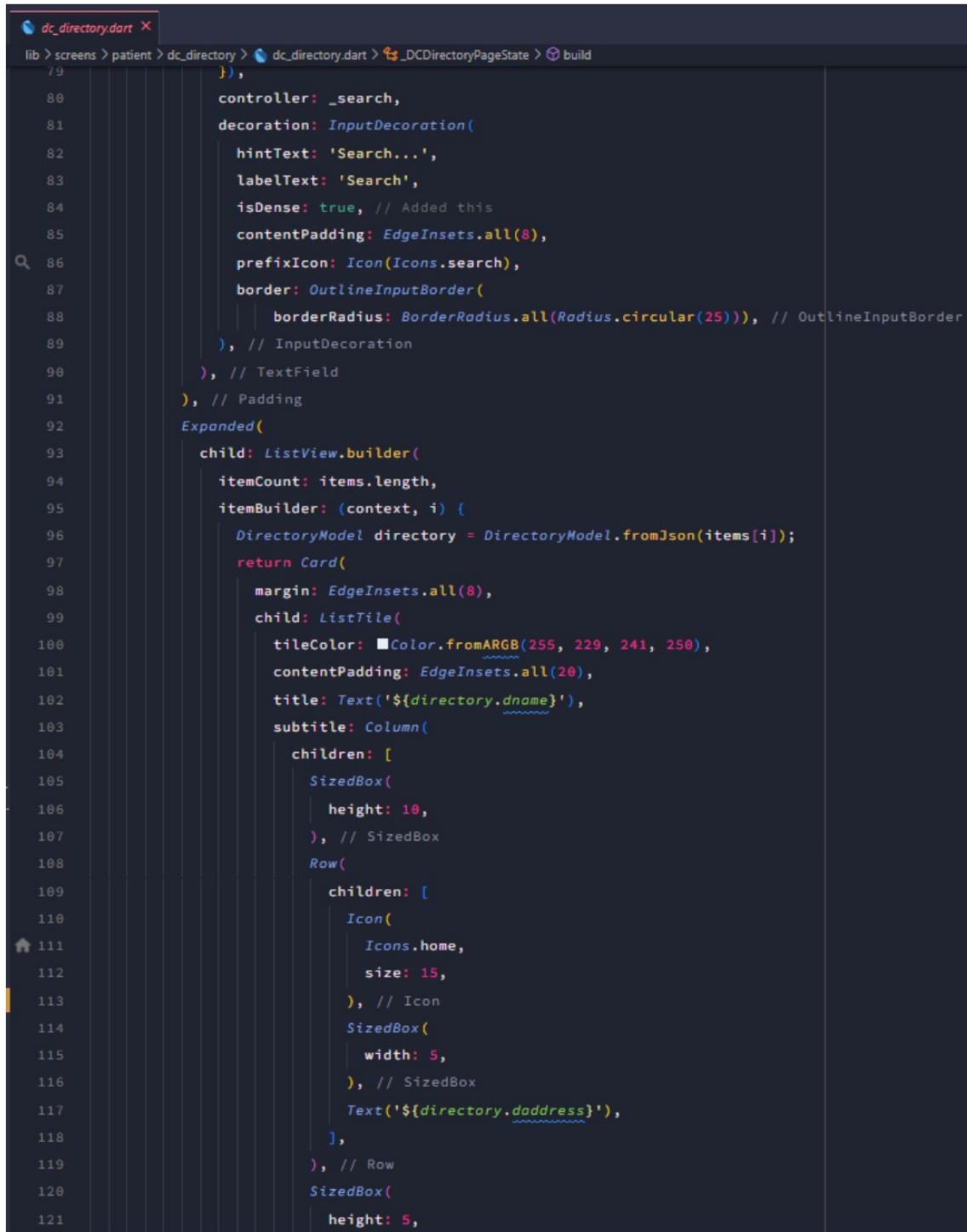


Figure 4. 19 Profile Page Interface

4.3.2.3 Dialysis Center Directory Page

Directory page for Patient and Hospital Staff are similar, only Dialysis Staff's directory page interface has little differences due to the add, update, and delete directory function.



```
dc_directory.dart
lib > screens > patient > dc_directory > dc_directory.dart > _DCDirectoryPageState > build
  ...
    ...
    controller: _search,
    decoration: InputDecoration(
      hintText: 'Search...',
      labelText: 'Search',
      isDense: true, // Added this
      contentPadding: EdgeInsets.all(8),
      prefixIcon: Icon(Icons.search),
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(25)), // OutlineInputBorder
    ), // InputDecoration
  ), // Padding
  Expanded(
    child: ListView.builder(
      itemCount: items.length,
      itemBuilder: (context, i) {
        DirectoryModel directory = DirectoryModel.fromJson(items[i]);
        return Card(
          margin: EdgeInsets.all(8),
          child: ListTile(
            tileColor: Color.fromRGBO(255, 229, 241, 250),
            contentPadding: EdgeInsets.all(20),
            title: Text('${directory.dname}'),
            subtitle: Column(
              children: [
                SizedBox(
                  height: 10,
                ), // SizedBox
                Row(
                  children: [
                    Icon(
                      Icons.home,
                      size: 15,
                    ), // Icon
                    SizedBox(
                      width: 5,
                    ), // SizedBox
                    Text('${directory.daddress}'),
                  ],
                ), // Row
                SizedBox(
                  height: 5,
                ),
              ],
            ),
          ),
        );
      }
    )
  )
}
```

Figure 4. 20 Directory Coding for Patient and Hospital Staff

```
lib > screens > dialysis staff > tabs ds > directory tabs > AddDirectory.dart > AddDirectoryTabBar > AddDirectoryTabBar
41
42     Future saveDirectory() async{
43         await _databaseHelper!.insertDirectory({
44             'dname': _dcnameCon.text.trim(),
45             'daddress': _dcaddressCon.text.trim(),
46             'dcnumber': _dcnumberCon.text.trim(),
47             'dkm': _dckmCon.text.trim(),
48             'dopenhr': _dcopenhCon.text.trim(),
49             'dclosehr': _dcclosehCon.text.trim(),
50             'drating': _dcratingCon.text.trim(),
51         });
52         showDialog<String>(
53             context: context,
54             builder: (BuildContext context) => AlertDialog(
55                 title: const Text('Directory Data added Successfully'),
56                 actions: <Widget>[
57                     TextButton(
58                         onPressed: () => Navigator.of(context).push(
59                             MaterialPageRoute(builder: (context) => DSDirectoryPage()),
60                         ),
61                         child: const Text('OK'),
62                     ), // TextButton
63                 ], // <Widget>[]
64             ), // AlertDialog

```

Figure 4. 21 Add Directory Function

```
lib > screens > dialysis staff > tabs ds > directory tabs > AllDirectory.dart > _AllDirectoryTabBarState > initState
36     void filterSearch(String query) async {
37         var pSearchList = allDirectory;
38         if (query.isNotEmpty) {
39             var pListdata = [];
40             pSearchList.forEach((items) {
41                 var directory = DirectoryModel.fromJson(items);
42                 if (directory.dname!.toLowerCase().contains(query.toLowerCase())) {
43                     pListdata.add(items);
44                 }
45             });
46             setState(() {
47                 items = [];
48                 items.addAll(pListdata);
49             });
50         } else {
51             setState(() {
52                 items = [];
53                 items = allDirectory;
54             });
55         }
56     }
57
58     Future deleteData(int did) async {
59         await _databaseHelper!.deleteDirectory(did);
60         setState(() {});
61     }

```

Figure 4. 22 Dialysis Staff Directory Coding

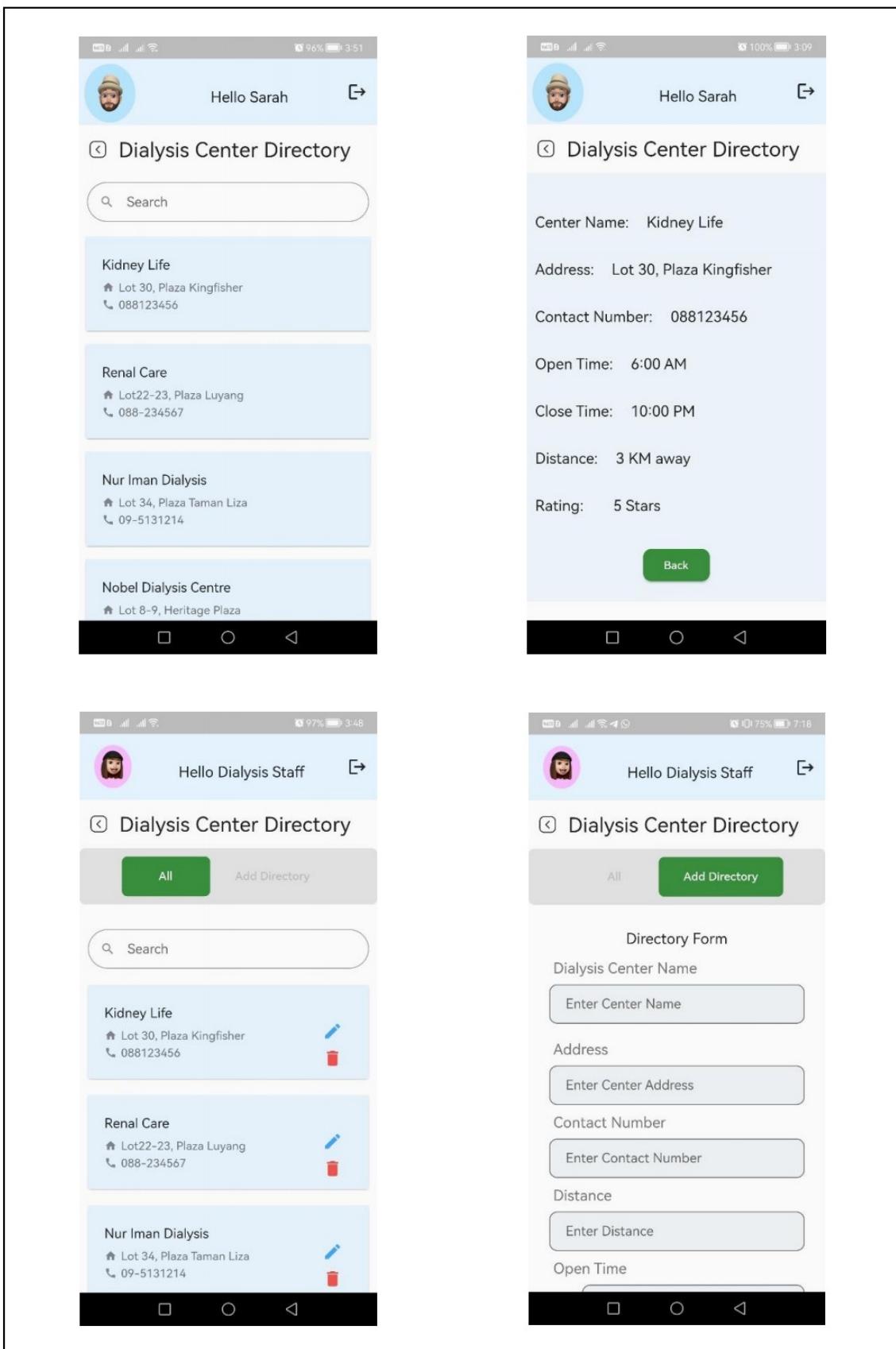
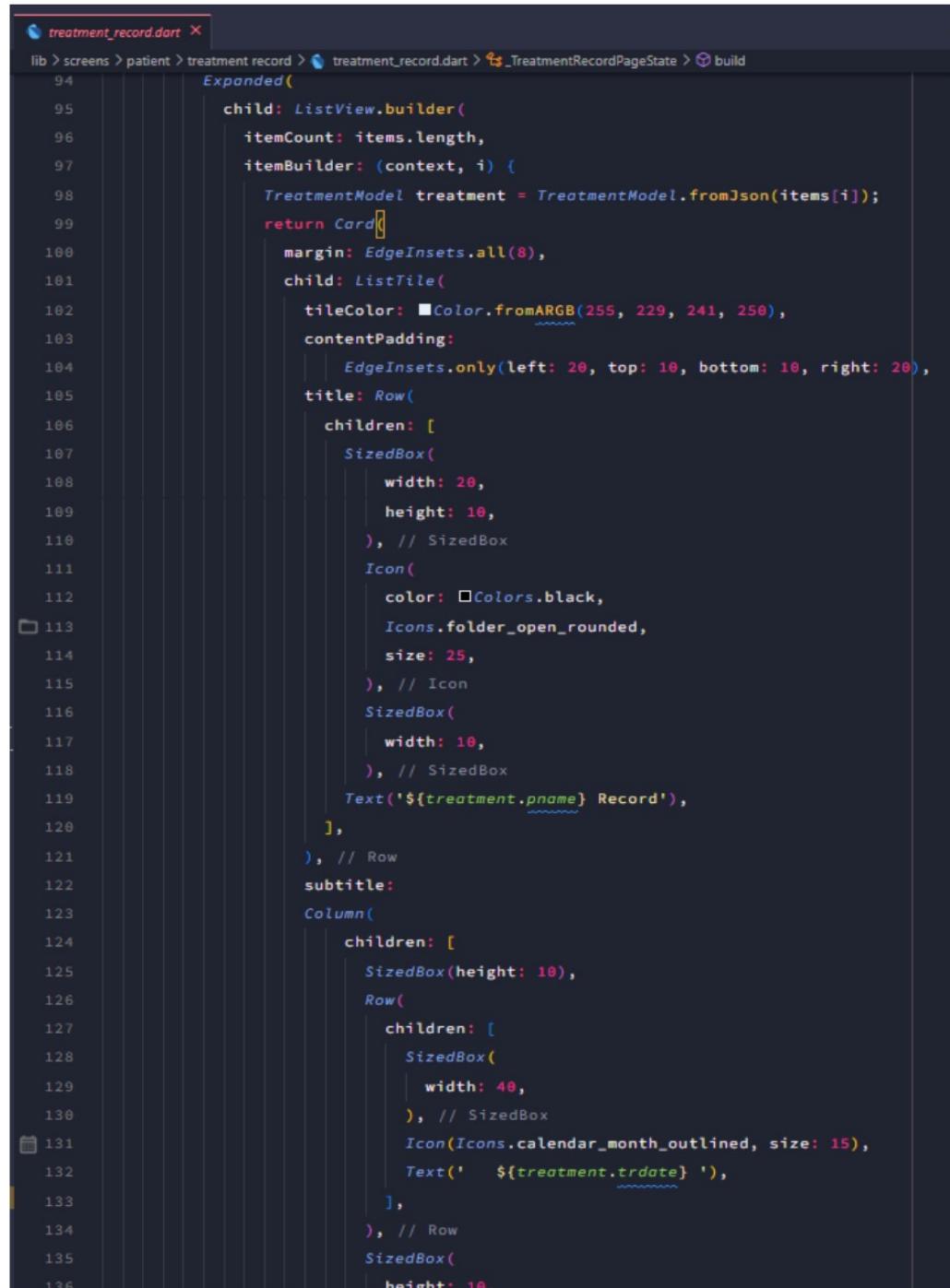


Figure 4.23 Directory Page Interfaces

4.3.2.4 Treatment Record Page

Treatment record interface for patient and dialysis staff a bit similar. The only difference is that dialysis staff have the function to add, update, and delete the treatment record while patients can only view the details of the treatment record.



The screenshot shows a code editor with the file 'treatment_record.dart' open. The code is a Flutter widget definition for a treatment record page. It uses a ListView.builder to build a list of cards, each representing a treatment record. Each card has a title (containing the patient's name and 'Record') and a subtitle (containing a date). The code includes imports for 'lib', 'screens', 'patient', 'treatment record', 'TreatmentRecordPageState', and 'build'. The main structure is an Expanded child of a ListView.builder.

```
lib > screens > patient > treatment record > treatment_record.dart > _TreatmentRecordPageState > build
94     Expanded(
95       child: ListView.builder(
96         itemCount: items.length,
97         itemBuilder: (context, i) {
98           TreatmentModel treatment = TreatmentModel.fromJson(items[i]);
99           return Card(
100             margin: EdgeInsets.all(8),
101             child: ListTile(
102               tileColor: Color.fromARGB(255, 229, 241, 250),
103               contentPadding:
104                 EdgeInsets.only(left: 20, top: 10, bottom: 10, right: 20),
105               title: Row(
106                 children: [
107                   SizedBox(
108                     width: 20,
109                     height: 10,
110                   ), // SizedBox
111                   Icon(
112                     color: Colors.black,
113                     Icons.folder_open_rounded,
114                     size: 25,
115                   ), // Icon
116                   SizedBox(
117                     width: 10,
118                   ), // SizedBox
119                   Text('${treatment.pname} Record',
120                 ],
121               ), // Row
122               subtitle:
123               Column(
124                 children: [
125                   SizedBox(height: 10),
126                   Row(
127                     children: [
128                       SizedBox(
129                         width: 40,
130                       ), // SizedBox
131                       Icon(Icons.calendar_month_outlined, size: 15),
132                       Text(' ${treatment.trdate} '),
133                     ],
134                   ), // Row
135                   SizedBox(
136                     height: 10,
```

Figure 4. 24 Treatment Record Page Coding

```
60 Future<TreatmentModel?> addTreatment() async {
61   await _databaseHelper!.insertData({
62     'bbweight': _bbwCon.text.trim(),
63     'bbpreasure': _bbpCon.text.trim(),
64     'bhrate': _bhrCon.text.trim(),
65     'btemp': _btempCon.text.trim(),
66     'dbpreasure1': _dbp1Con.text.trim(),
67     'dbpreasure2': _dbp2Con.text.trim(),
68     'dbpreasure3': _dbp3Con.text.trim(),
69     'dbpreasure4': _dbp4Con.text.trim(),
70     'dbpreasure5': _dbp5Con.text.trim(),
71     'dhrate1': _dhr1Con.text.trim(),
72     'dhrate2': _dhr2Con.text.trim(),
73     'dhrate3': _dhr3Con.text.trim(),
74     'dhrate4': _dhr4Con.text.trim(),
75     'dhrate5': _dhr5Con.text.trim(),
76     'abweight': _abwCon.text.trim(),
77     'abpreasure': _abpCon.text.trim(),
78     'ahrate': _ahrCon.text.trim(),
79     'atemp': _atempCon.text.trim(),
80     'pname': _pnameCon.text.trim(),
81     'trdate': _trdateCon.text.trim(),
82     'trtime': _trtimeCon.text.trim(),
83   });
84   showDialog<String>(
85     context: context,
86     builder: (BuildContext context) => AlertDialog(
87       title: const Text('Treatment Data added Successfully'),
88       actions: <Widget>[
89         TextButton(
90           onPressed: () => Navigator.of(context).push(
91             MaterialPageRoute(builder: (context) => DSTreatmentRecordPage()),
92           ),
93           child: const Text('OK'),
94         ), // TextButton
95       ], // <Widget>[]
96     ), // AlertDialog
97   );
98 }
```

Figure 4. 25 Add Treatment Record Function

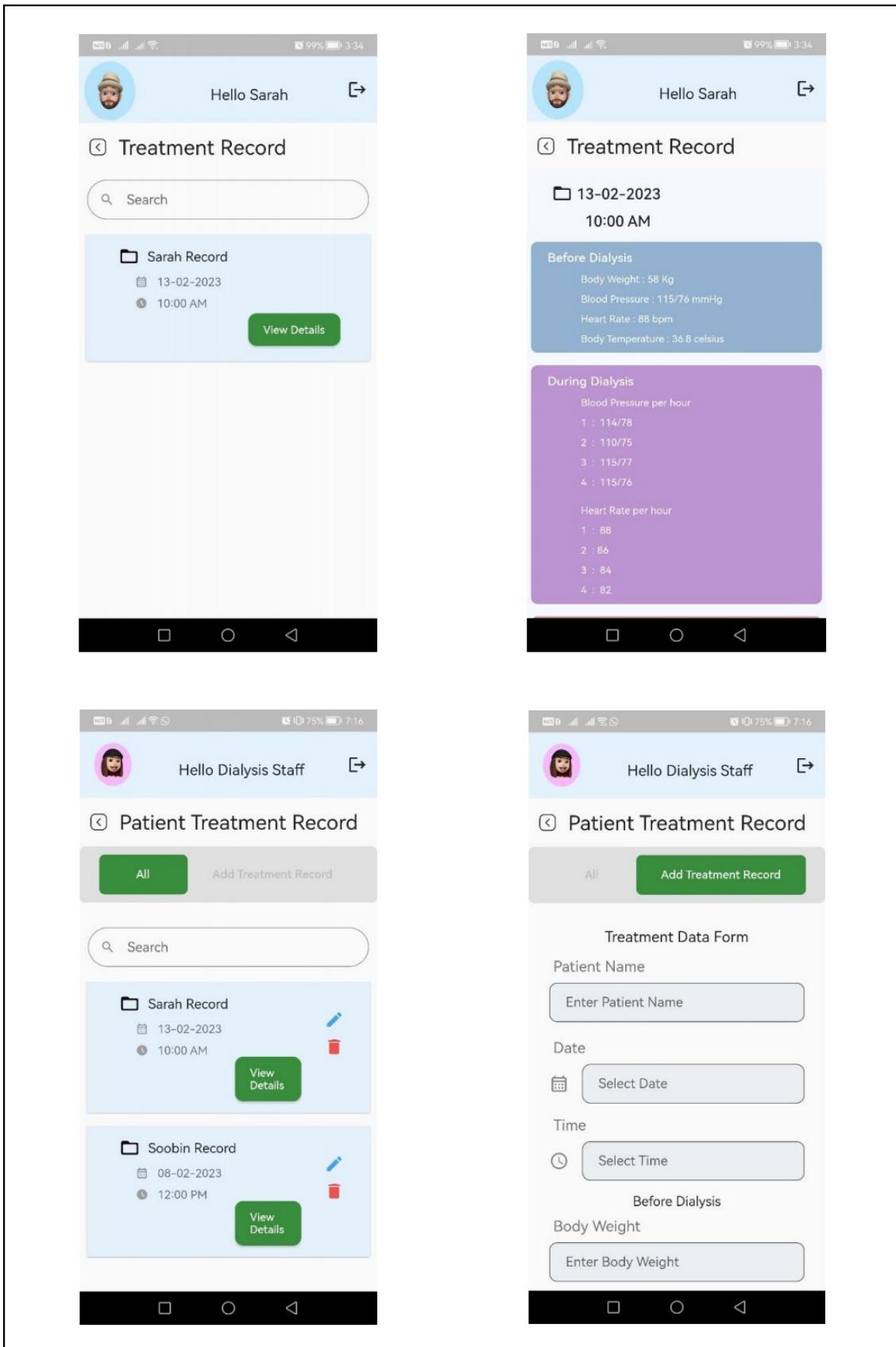
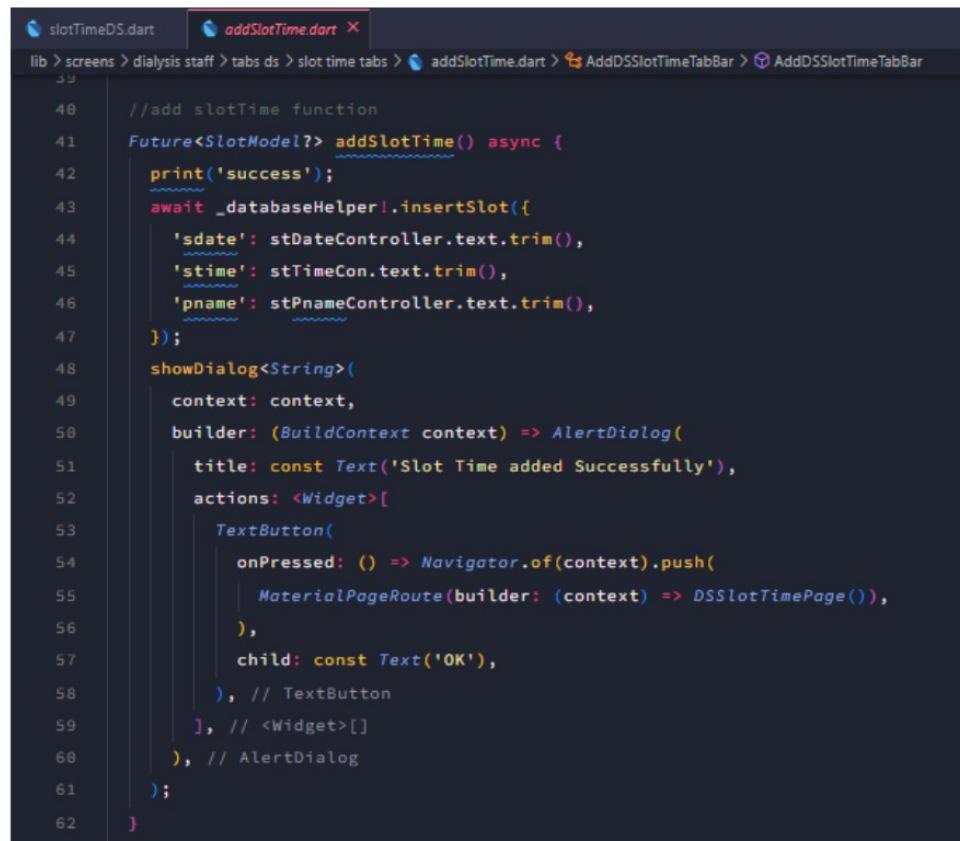


Figure 4. 26 Treatment Record Page Interfaces

4.3.2.5 Slot Time Page

Patient's slot time interface shows their dialysis treatment slot time that has been assigned by the dialysis staff, the completed slot time and the button for patient to make request slot time changes. Dialysis staff's slot time interface shows the patients slot time that has been added, slot time form and patient slot time request.



```
lib > screens > dialysis staff > tabs ds > slot time tabs > addSlotTime.dart > AddDSSlotTimeTabBar > AddDSSlotTimeTabBar

40 //add slotTime function
41 Future<SlotModel?> addSlotTime() async {
42   print('success');
43   await _databaseHelper!.insertSlot({
44     'sdate': stDateController.text.trim(),
45     'stime': stTimeCon.text.trim(),
46     'pname': stPnameController.text.trim(),
47   });
48   showDialog<String>(
49     context: context,
50     builder: (BuildContext context) => AlertDialog(
51       title: const Text('Slot Time added Successfully'),
52       actions: <Widget>[
53         TextButton(
54           onPressed: () => Navigator.of(context).push(
55             MaterialPageRoute(builder: (context) => DSSlotTimePage()),
56           ),
57           child: const Text('OK'),
58         ), // TextButton
59       ], // <Widget>[]
60     ), // AlertDialog
61   );
62 }
```

Figure 4. 27 Add Slot Time Function

```
time_slot_patient.dart X userModel.dart databaseHelper.dart pubspec.yaml sign_up.dart
lib > screens > patient > time slot > time_slot_patient.dart > ...
42     ), // BoxDecoration
43     child: Center(
44       child: TabBar(
45         labelColor: Colors.white,
46         unselectedLabelColor: Colors.grey.shade400,
47         isScrollable: true,
48         labelPadding: EdgeInsets.only(
49           left: 40.0, right: 40.0, bottom: 0.2, top: 0.2), // EdgeInsets.only
50         indicator: BoxDecoration(
51           border: Border.all(color: Colors.green.shade700),
52           borderRadius: BorderRadius.circular(7),
53           color: Colors.green.shade700,
54         ), // BoxDecoration
55         tabs: [
56           Tab(
57             text: 'Upcoming',
58           ), // Tab
59           Tab(
60             text: 'Completed',
61           ), // Tab
62         ],
63       ), // TabBar
64     ), // Center
65   ), // Container
66   SizedBox(
67     height: 20,
68   ), // SizedBox
69   Expanded(
70     child: TabBarView(
71       children: [
72         PatientSTTabBarUpcoming(),
73         PatientSTTabBarCompleted(),
74       ],
75     ), // TabBarView
76   ), // Expanded
77 ].
```

Figure 4. 28 Slot Time Page Coding

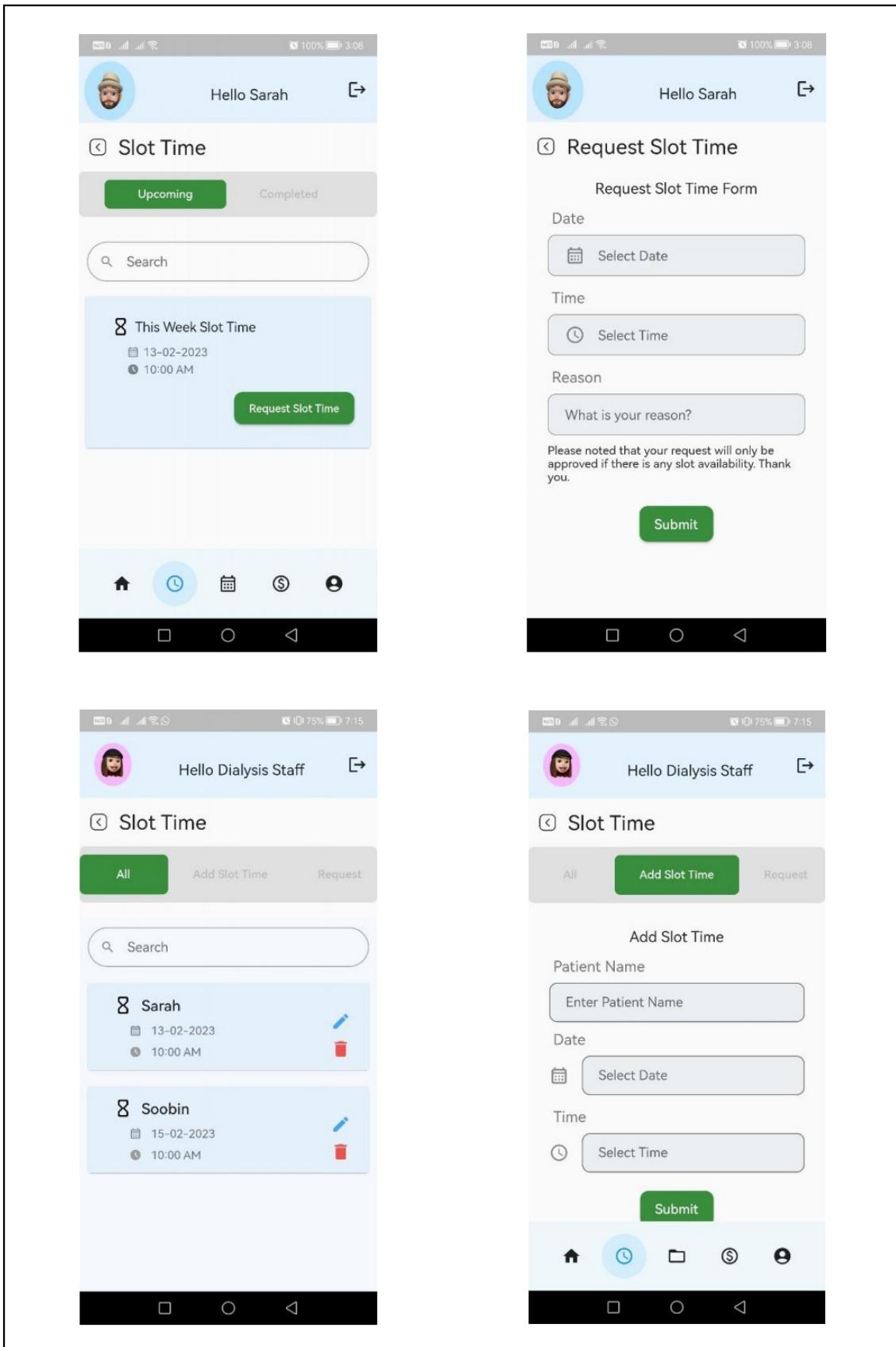
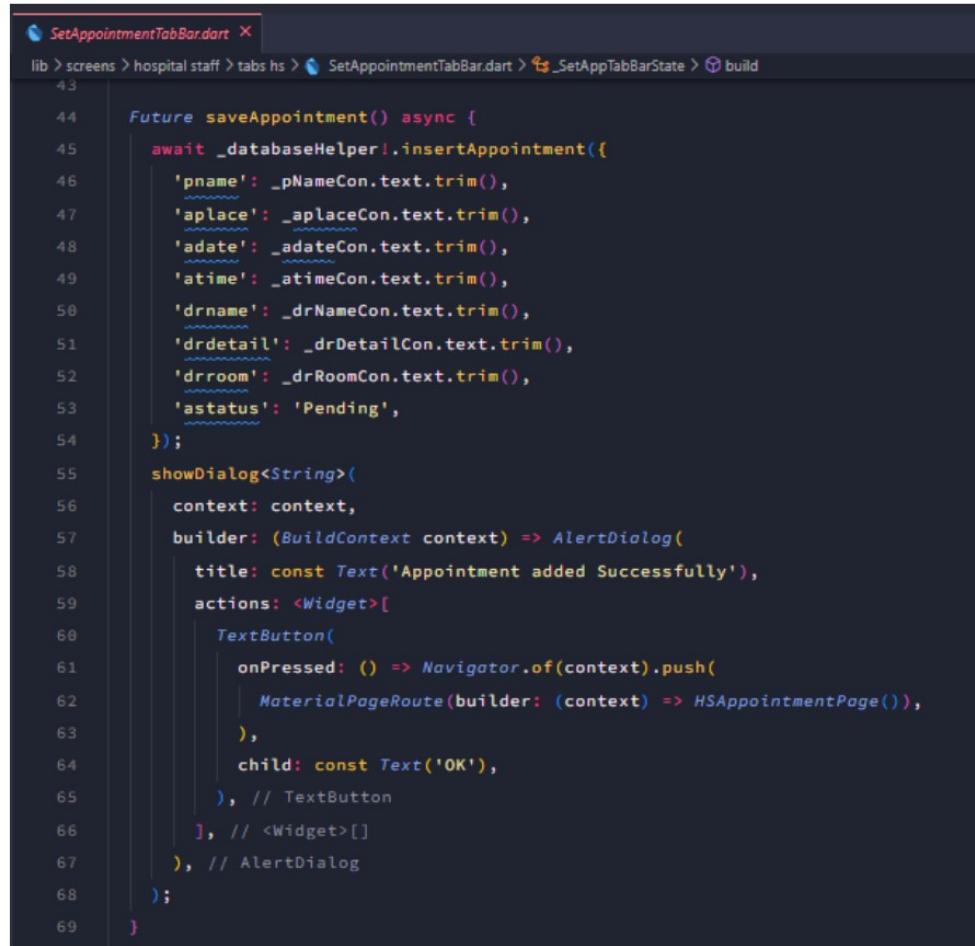


Figure 4.29 Slot Time Page Interfaces

4.3.2.6 Appointment Page

Patient appointment page interface shows the patient upcoming appointment information with specialists of a hospital where they refer to which has been assigned by the hospital staff, the completed appointment and the button for patient to make request for reschedule their appointment date and time. Hospital staff appointment page interfaces shows all the appointment that they have created, appointment form and patient reschedule request.



```
lib > screens > hospital staff > tabs hs > SetAppointmentTabBar.dart > _SetAppBarState > build
43
44     Future saveAppointment() async {
45         await _databaseHelper!.insertAppointment({
46             'pname': _pNameCon.text.trim(),
47             'aplace': _aplaceCon.text.trim(),
48             'adate': _adateCon.text.trim(),
49             'atime': _atimeCon.text.trim(),
50             'drname': _drNameCon.text.trim(),
51             'drdetail': _drDetailCon.text.trim(),
52             'drroom': _drRoomCon.text.trim(),
53             'astatus': 'Pending',
54         });
55         showDialog<String>(
56             context: context,
57             builder: (BuildContext context) => AlertDialog(
58                 title: const Text('Appointment added Successfully'),
59                 actions: <Widget>[
60                     TextButton(
61                         onPressed: () => Navigator.of(context).push(
62                             MaterialPageRoute(builder: (context) => HSArrangementPage()),
63                         ),
64                         child: const Text('OK'),
65                     ), // TextButton
66                 ], // <Widget>[]
67             ), // AlertDialog
68         );
69     }

```

Figure 4. 30 Add Appointment Function

```
appointment_patient.dart userModel.dart databaseHelper.dart pubspec.yaml sign_up.dart
lib > screens > patient > appointment > appointment_patient.dart > ...
  ...
  42           ), // BorderRadius.circular
  43       ), // BoxDecoration
  44     child: Center(
  45       child: TabBar(
  46         labelColor: Colors.white,
  47         unselectedLabelColor: Colors.grey.shade400,
  48         isScrollable: true,
  49         labelPadding: EdgeInsets.only(
  50           left: 40.0, right: 40.0, bottom: 0.2, top: 0.2), // EdgeInsets.only
  51         indicator: BoxDecoration(
  52           border: Border.all(color: Colors.green.shade700),
  53           borderRadius: BorderRadius.circular(7),
  54           color: Colors.green.shade700,
  55         ), // BoxDecoration
  56         tabs: [
  57           Tab(
  58             text: 'Upcoming',
  59           ), // Tab
  60           Tab(
  61             text: 'Completed',
  62           ), // Tab
  63         ],
  64       ), // TabBar
  65     ), // Center
  66   ), // Container
  67   SizedBox(
  68     height: 20,
  69   ), // SizedBox
  70   Expanded(
  71     child: TabBarView(
  72       children: [
  73         PatientAppBarTabBarUpcoming(),
  74         PatientAppBarTabBarCompleted(),
  75       ],
  76     ), // TabBarView
  77   ), // Expanded
  78 ], // Column
  79 ), // Scaffold
  80   ), // SafeArea
  81   ); // DefaultTabController
  82 }
  83 }
```

Figure 4. 31 Appointment Page Coding

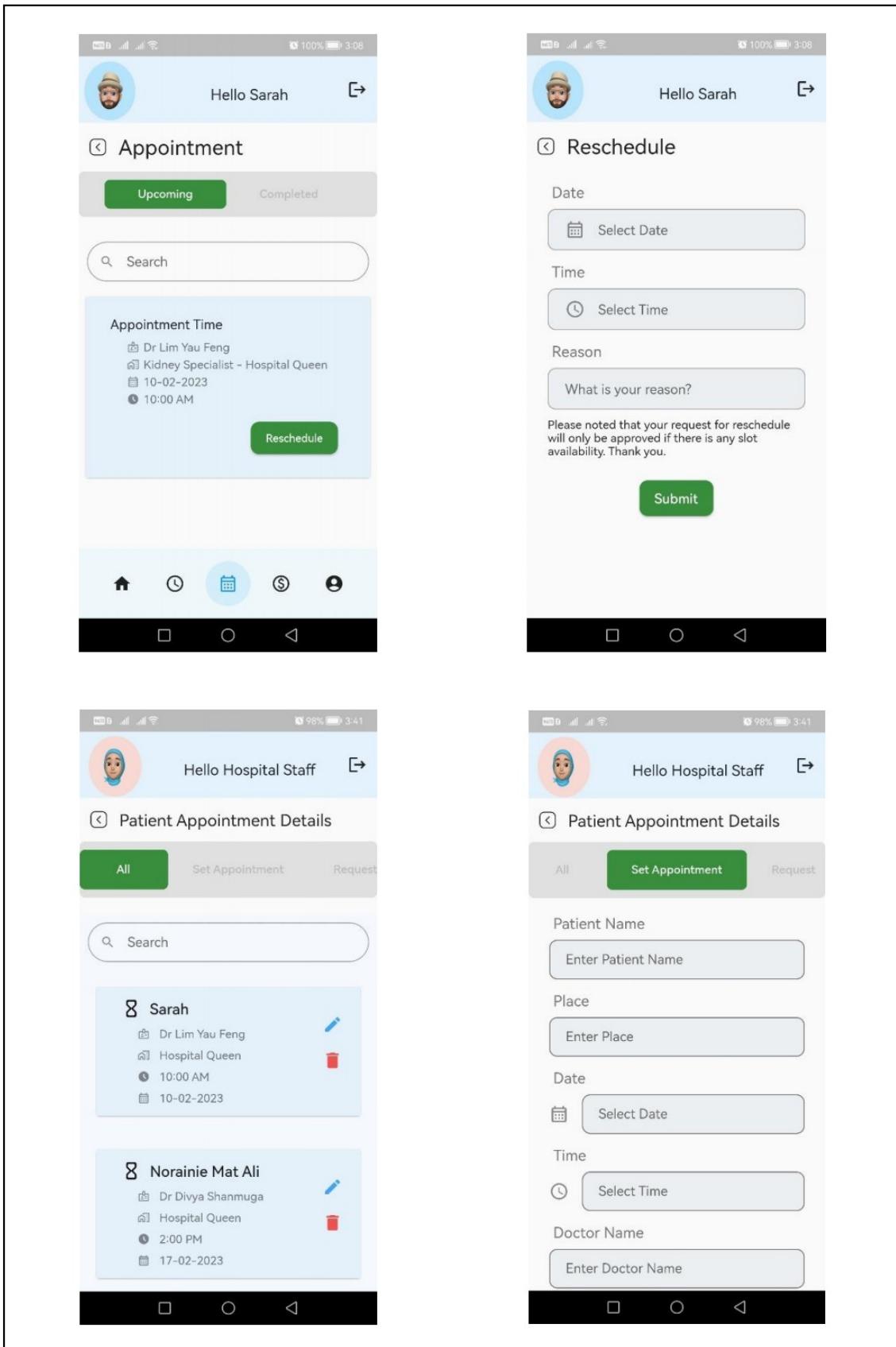
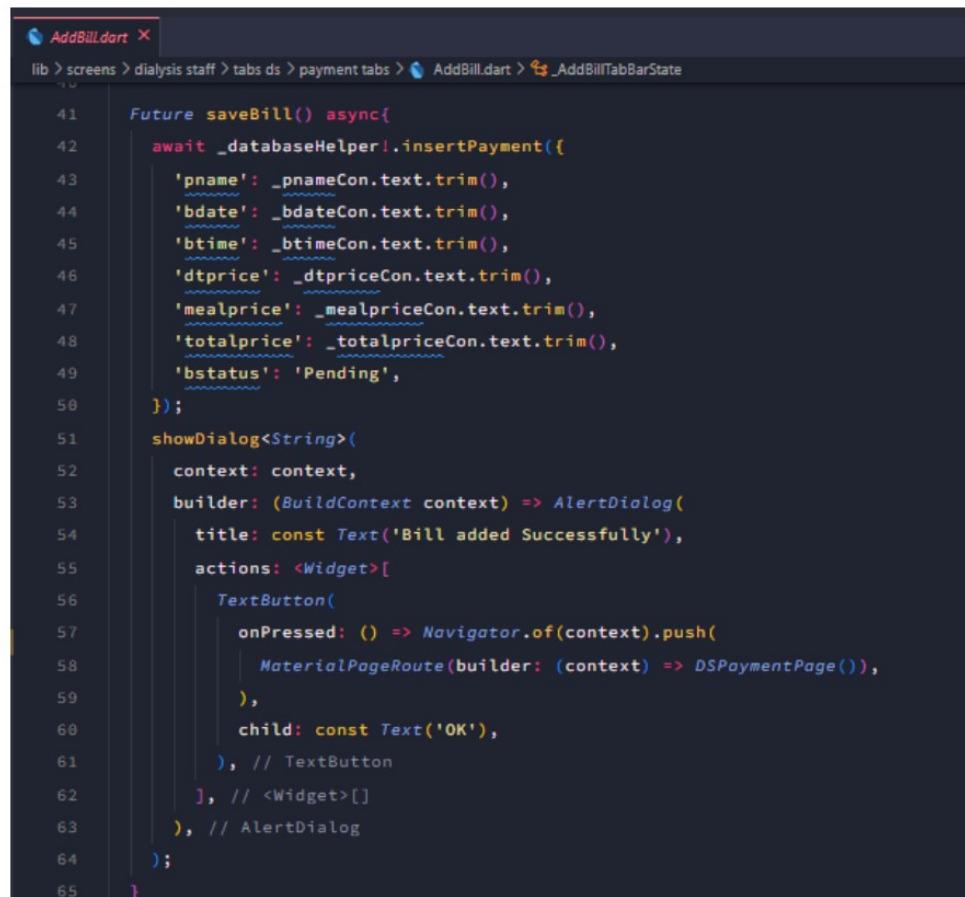


Figure 4. 32 Appointment Page Interfaces

4.3.2.7 Payment Page

Patient payment page the patient's dialysis treatment bill and the patient can pay the bill by clicking on the “Pay Bill” button using the payment method they prefer. Dialysis staff payment page shows all the patient treatment bill that have been created, bill form, and patient payment information.



```
lib > screens > dialysis staff > tabs ds > payment tabs > AddBill.dart > _AddBillTabBarState

41 Future saveBill() async{
42     await _databaseHelper!.insertPayment({
43         'pname': _pnameCon.text.trim(),
44         'bdate': _bdateCon.text.trim(),
45         'btime': _btimeCon.text.trim(),
46         'dtprice': _dtpriceCon.text.trim(),
47         'mealprice': _mealpriceCon.text.trim(),
48         'totalprice': _totalpriceCon.text.trim(),
49         'bstatus': 'Pending',
50     });
51     showDialog<String>(
52         context: context,
53         builder: (BuildContext context) => AlertDialog(
54             title: const Text('Bill added Successfully'),
55             actions: <Widget>[
56                 TextButton(
57                     onPressed: () => Navigator.of(context).push(
58                         MaterialPageRoute(builder: (context) => DSPaymentPage()),
59                     ),
60                     child: const Text('OK'),
61                 ), // TextButton
62             ], // <Widget>[]
63         ), // AlertDialog
64     );
65 }
```

Figure 4. 33 Add Bill Function

```
payment_patient.dart
lib > screens > patient > payment > payment_patient.dart > _PaymentPageState > build
92     Expanded(
93         child: ListView.builder(
94             itemCount: items.length,
95             itemBuilder: (context, i) {
96                 PaymentModel payment = PaymentModel.fromJson(items[i]);
97                 return Card(
98                     margin: EdgeInsets.all(20),
99                     child: ListTile(
100                         tileColor: Color.fromARGB(255, 229, 241, 250),
101                         contentPadding:
102                             EdgeInsets.only(left: 30, top: 20, bottom: 20),
103                         title: Row(
104                             children: [
105                                 Icon(Icons.receipt_long_rounded),
106                                 size: 25, color: Colors.black), // Icon
107                                 SizedBox(
108                                     width: 5,
109                                 ), // SizedBox
110                                 Text('Bill Detail'),
111                             ],
112                         ), // Row
113                         subtitle: Column(
114                             children: [
115                                 SizedBox(
116                                     height: 10,
117                                 ), // SizedBox
118                                 Row(
119                                     children: [
120                                         SizedBox(
121                                             width: 20,
122                                         ), // SizedBox
123                                         Icon(
124                                             Icons.calendar_month_outlined,
125                                             size: 15,
126                                         ), // Icon
127                                         SizedBox(
128                                             width: 5,
129                                         ), // SizedBox
130                                         Text('${payment.bdate}'),
131                                     ],
132                                 ),
133                             ],
134                         ),
135                     );
136                 }
137             );
138         );
139     );
140 }
```

Figure 4. 34 Payment Page Coding

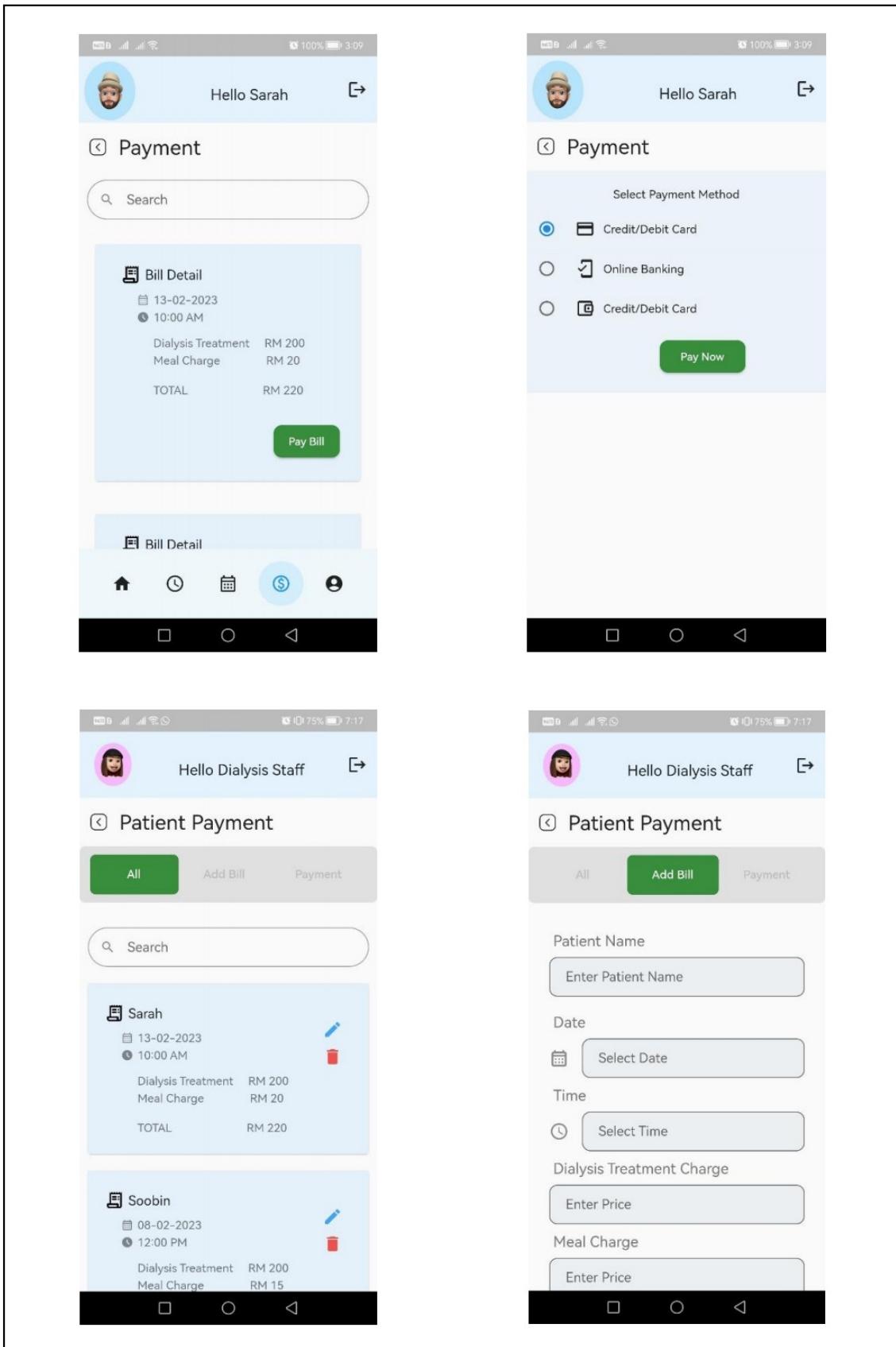


Figure 4. 35 Payment Page Interfaces

4.4 Application Testing

User Acceptance Test (UAT) is used to test every function of the application. The application was being tested as an alpha version application which is an early version of the application where it is still being designed and built and unstable but have all the useful functions to show the functionality of the application. The UAT form is produced using Google Form which can help to generate the result. The link of the UAT form will be given to the testers through *WhatsApp* or email which the tester prefers before the testing begin.

The screenshot shows a Google Form titled "myDialysis Application UAT Form for User". It includes the following sections:

- Account Information:** Email: khairunsfaman@gmail.com (not shared), Switch accounts, *Required.
- User Acceptance Testing (UAT) Form for Patient:** Instructions: In this section, you are required to test the application as dialysis center patient. The given code on the sign-up page is capital P with any number.
- Activities and status ***: A table with two columns: Success and Fail.

	Success	Fail
User able to sign up.	<input type="radio"/>	<input type="radio"/>
User able to login into the application	<input type="radio"/>	<input type="radio"/>
User able to view their profile.	<input type="radio"/>	<input type="radio"/>
User able to edit their profile.	<input type="radio"/>	<input type="radio"/>
User able to view upcoming slot time.	<input type="radio"/>	<input type="radio"/>
User able to view completed slot time.	<input type="radio"/>	<input type="radio"/>
User able to send slot time request.	<input type="radio"/>	<input type="radio"/>
- Test Requirements:** A list of 14 items, each with two radio button options: Success () and Fail ().
 - User able to send slot time request.
 - User able to view upcoming appointment.
 - User able to view completed appointment.
 - User able to send reschedule appointment request.
 - User able to search treatment record.
 - User able to view treatment record details.
 - User able to view bill detail on payment page.
 - User able to select payment method.
 - User able to make payment.
 - User able to view dialysis center directory.
 - User able to search for dialysis center name on directory page.
 - User able to view dialysis center details.
 - User able to logout from the application
- Comment or feedback on the application:** A text area labeled "Your answer".
- Navigation:** Back, Next, Clear form.

Figure 4. 36 UAT Form (Patient Section)

The screenshot shows the 'User Acceptance Testing (UAT) Form for Dialysis Staff' section of the myDialysis application. The form includes a header with the myDialysis logo and a sign-in area. Below the header, the title 'myDialysis Application UAT Form for User' is displayed. A note indicates that users must sign up with the email 'khairunsfaman@gmail.com' and a password consisting of capital 'D' followed by any number.

User Acceptance Testing (UAT) Form for Dialysis Staff

In this section, you are required to test the application as a dialysis center staff. The given code on the sign-up page is capital D with any number.

Activities and status *

	Success	Fail
User able to sign up.	<input type="radio"/>	<input checked="" type="radio"/>
User able to login into the application	<input checked="" type="radio"/>	<input type="radio"/>
User able to view their profile.	<input type="radio"/>	<input checked="" type="radio"/>
User able to edit their profile.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view patients profile.	<input checked="" type="radio"/>	<input type="radio"/>
User able to view patient slot time.	<input type="radio"/>	<input checked="" type="radio"/>
User able to add patient slot time.	<input checked="" type="radio"/>	<input type="radio"/>

User able to view patient slot time request.	<input type="radio"/>	<input checked="" type="radio"/>
User able to approve patient slot time request.	<input type="radio"/>	<input checked="" type="radio"/>
User able to reject patient slot time request.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view patient treatment record.	<input type="radio"/>	<input checked="" type="radio"/>
User able to add patient treatment record.	<input type="radio"/>	<input checked="" type="radio"/>
User able to edit patient treatment record details.	<input type="radio"/>	<input checked="" type="radio"/>
User able to delete patient treatment record.	<input type="radio"/>	<input checked="" type="radio"/>
User able to add treatment bill.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view treatment bill.	<input type="radio"/>	<input checked="" type="radio"/>
User able to edit treatment bill.	<input type="radio"/>	<input checked="" type="radio"/>
User able to delete treatment bill.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view treatment payment details.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view dialysis center directory.	<input type="radio"/>	<input checked="" type="radio"/>
User able to search for dialysis center name on directory page.	<input type="radio"/>	<input checked="" type="radio"/>
User able to view dialysis center details.	<input type="radio"/>	<input checked="" type="radio"/>
User able to add dialysis center details.	<input type="radio"/>	<input checked="" type="radio"/>

User able to edit dialysis center details.	<input type="radio"/>	<input checked="" type="radio"/>
User able to delete dialysis center details.	<input type="radio"/>	<input checked="" type="radio"/>
User able to logout from the application	<input type="radio"/>	<input checked="" type="radio"/>

Comment or feedback on the application.

Your answer

Back Next Clear form

Figure 4. 37 UAT Form (Dialysis Staff Section)

myDialysis Application UAT Form for User

khairunsfaman@gmail.com (not shared) Switch accounts

*Required

User Acceptance Testing (UAT) Form for Hospital Staff

In this section, you are required to test the application as a hospital staff. The given code on the sign-up page is capital H with any number.

Activities and status *

	Success	Fail
User able to sign up.	<input type="radio"/>	<input type="radio"/>
User able to login into the application	<input type="radio"/>	<input type="radio"/>
User able to view their profile.	<input type="radio"/>	<input type="radio"/>
User able to edit their profile.	<input type="radio"/>	<input type="radio"/>
User able to search patient profile	<input type="radio"/>	<input type="radio"/>
User able to view patients profile.	<input type="radio"/>	<input type="radio"/>

User able to add patient appointment.

User able to edit patient appointment.

User able to delete patient appointment.

User able to view patient appointment reschedule request.

User able to approve patient appointment reschedule request.

User able to reject patient appointment reschedule request.

User able to view dialysis center directory.

User able to search for dialysis center name on directory page.

User able to view dialysis center details.

User able to logout from the application

Comment or feedback on the application.

Your answer

Figure 4. 38 UAT Form (Hospital Staff Section)

4.4.1 Proof of Testing

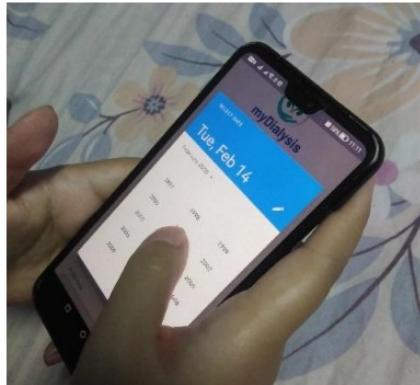


Figure 4. 39 User Testing on Sign Up Function

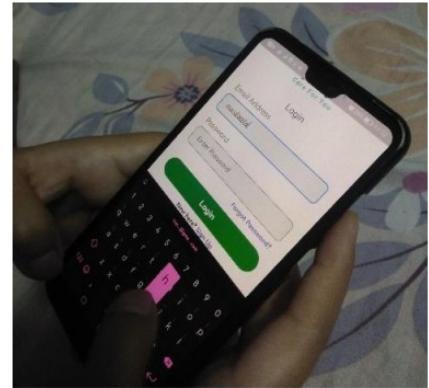


Figure 4. 40 User Testing on Login Function

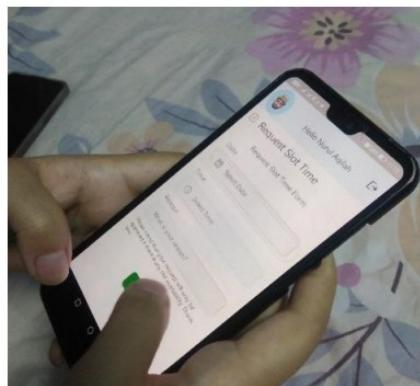


Figure 4. 41 User Testing as Patient

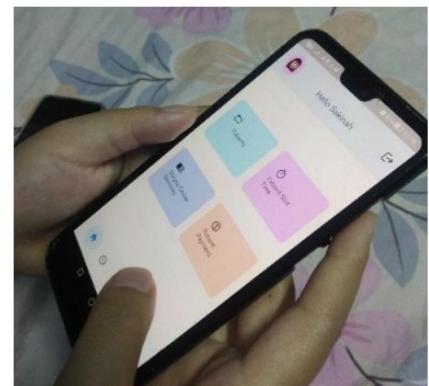


Figure 4. 42 User Testing as Dialysis Staff

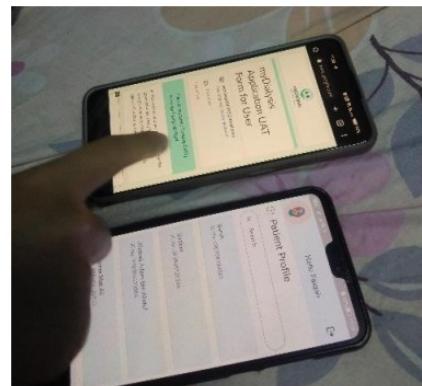


Figure 4. 43 User Testing as Hospital Staff

4.5 Result and Discussion

The testing was conducted by 4 users which all of them test the application as patient, dialysis staff and hospital staff point of view to inspect whether the application has met the requirements and is well enough before the final deliverables. Moreover, feedback will be collected from the tester which can help to improve the application. Based on the UAT form that have been responded by the testers on, mostly all the functions of the application have operated successfully. However, there are a few functions that have failed the test which is the edit functions on all interfaces including to update the slot time and appointment request status by approving and rejecting the request as well as the update for the deleted data . Based on the comments received, the delete function is successful, however the deleted data are not updated and removed from the interface unless the user logout from the application and login again, then the data will be updated and removed from the display or interface. The full UAT result graph can be refer on Appendix A.

Name:
4 responses
NURUL AQILAH
Mahligai Bt Muhamad Ikhsan
Abdullah Muhammed B Aswaf
Nur Balqis Binti Omar

Figure 4. 44 Name of the UAT respondents



Figure 4. 45 UAT Result Graph (Patient Section)



Figure 4. 46 UAT Result Graph (Dialysis Staff Section)



Figure 4. 47 UAT Result Graph (Hospital Staff Section)

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 discussing the summary of the development of the myDialysis application for dialysis center patient and staff, and hospital staff in order to achieve the objectives and solve the problem. In traditional way of storing data, it is mainly rely on sheets and files which most dialysis center are still using this method to store patients data. This will lead to a few problems such as missing files, misplaced files, time-consuming searching for patients' files, lack of file storing place, and destroyed by unavoidable casualties such as fire, floods, and earthquakes. This application will be convenient for the dialysis center staff and patients as it helps to store treatment data more securely and systematized. The data can be reached at speed and with less time consumption. The software used to develop this application is Visual Studio Code by using Dart programming language, Flutter as its framework and Android Studio to run the mobile emulator. This application is implemented and evaluated by 4 users that test the application as patients, dialysis staff, and hospital staff point of view to test the effectiveness and the functionality of the application.

5.2 Limitation and Constraint

The limitation and constraint during the development of the application:

i. Limitation of time

Time is crucial in the development of an application. Due to the limited time given in developing this application, there are few functionalities of the application modules that are not developed fully and efficient enough.

ii. Inadequate experience

Experience is expensive and it makes one work almost perfect. Due to no experience in developing mobile applications before, it leads to this application was not developed well enough and not operating smoothly. It still has a lot of improvement that can be done.

iii. Coding error

Coding is vital in developing the application. Errors such as logic errors and syntax errors can sometimes occur in the coding script, which can cause the application to be incapable of being executed and tested. The errors need to be solved in order to run the application smoothly, but it is time-consuming considering the lack of knowledge and experience in mobile application development.

5.3 Future Work

There are several functions and features that can be improved and added into the myDialysis application. Some functions to be improved are based on the UAT result.

- i. Edit/update functions for all modules can be fixed to complete the CRUD operation of the application.
- ii. Alert dialog can be added when the delete button is clicked to ensure that the data is not being deleted accidentally.
- iii. The issue where the form will be resetting when the application encounters an error on validation coding will be fixed to ensure user satisfaction so that user doesn't have to redo the form many times.
- iv. Notification feature can be added to remind the patients about their dialysis slot time and appointments schedule.
- v. Settings feature can be added for users to customize the language of the application such as Bahasa Melayu and Mandarin language.
- vi. Feature to get the location of the other dialysis center in the directory module can be added to make it easy for patients to locate the location such as linking the application with Google maps application.
- vii. A slot treatment schedule can be added in Slot Time Module and can be displayed to patients for them to easily view the slot time availability and organize the changes of their slot time based on the availability displayed. Same improvement for the Appointment Module as well.

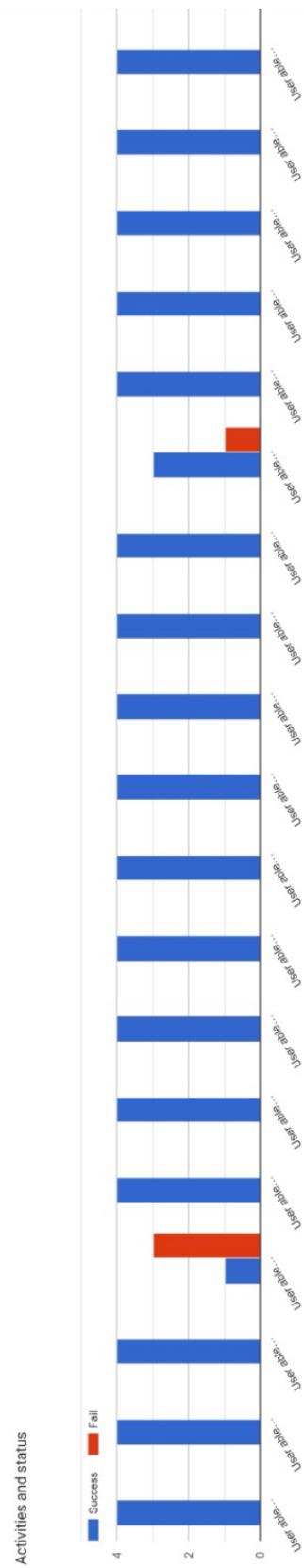
REFERENCES

- Medical Development Division, Ministry of Health Malaysia. (2010, November). *Nephrology Services Operational Policy*. Retrieved from <https://www.moh.gov.my/moh/images/gallery/Polisi/NEPHROLOGY.pdf>
- Daud, N. M. N., Bakar, N. A. A. A., & Rusli, H. M. (2010). *Implementing Rapid Application Development (RAD) methodology in developing practical training application system*. Retrieved from <https://doi.org/10.1109/ITSIM.2010.5561634>
- Fresenius Medical Care. (n.d.). *Fresenius Medical Care Company Profile*. Fresenius Medical Care. Retrieved from <https://www.freseniusmedicalcare.asia/en/about-us/company-profile>
- Fresenius myCompanion - Apps on Google Play*. (n.d.). Google Play. <https://play.google.com/store/apps/details?id=com.fresenius.MyCompanionNephroCare>
- NephroPlus. (n.d.). *Holiday Dialysis*. NephroPlus.com. <https://www.nephroplus.com/services/holiday-dialysis/>
- NephroPlus - Apps on Google Play*. (n.d.). Google Play. <https://play.google.com/store/apps/details?id=com.nephroplus.nephroplus>
- Freedom Dialysis - Apps on Google Play*. (2022, January 21). Google Play. <https://play.google.com/store/apps/details?id=com.techvify.freedomdialysis>
- Pusat Dialisis Malaysia - Apps on Google Play*. (2021, June 1). Google Play. <https://play.google.com/store/apps/details?id=com.triple3AAA.pusatdialis&hl=en&gl=US>
- Wikipedia Contributors. (2021, February 7). *Figma (software)*. Retrieved from Wikipedia website: [https://en.wikipedia.org/wiki/Figma_\(software\)](https://en.wikipedia.org/wiki/Figma_(software))
- Tohid, M. N. A., & Ibrahim, N. (2021). *Hemodialysis Patient Management System Using Mobile Application*. 2(2), 668–681. Retrieved from <https://doi.org/https://doi.org/10.30880/aitcs.2021.02.02.044>

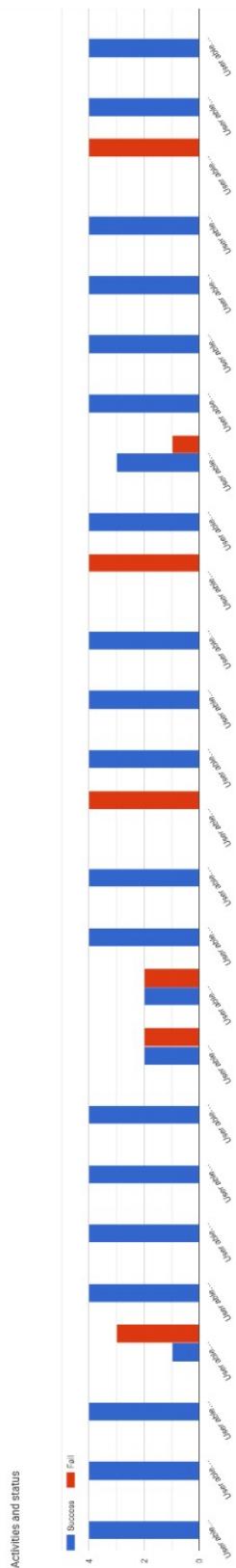
APPENDIX A

UAT Result (Graph)

UAT Result Graph for Patient



UAT Result Graph for Dialysis Staff



UAT Result Graph for Hospital Staff

