

hiDoc: DOCTOR APPOINTMENT BOOKING  
WEB APPLICATION

NUR NADHIRAH BINTI YAHAYA

BACHELOR IN SOFTWARE ENGINEERING

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Nur Nadhirah Binti Yahaya

Date of Birth :

Title : Doctor Appointment Booking Web Application

Academic Session : Semester 22/23

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)

\_\_\_\_\_  
Date: 09/02/2023

\_\_\_\_\_  
Muhammad' Arif Bin Mohamad  
Date: 09/02/2023

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



## SUPERVISOR'S DECLARATION

I/We\* hereby declare that I/We\* have checked this thesis/project\* and in my/our\* opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of \*Doctor of Philosophy/ Master of Engineering/ Master of Science in Computer Science (Software Engineering) With Honours

A handwritten signature in blue ink, appearing to read 'Muhammad' Arif Bin Mohamad', is written above a horizontal line.

(Supervisor's Signature)

Full Name : Muhammad' Arif Bin Mohamad

Position : Pensyarah Kanan

Date : 09/02/2023

---

(Co-supervisor's Signature)

Full Name :

Position :

Date :



## 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 be 'Nur Nadhirah', is written above a horizontal line.

(Student's Signature)

Full Name : Nur Nadhirah Binti Yahaya

ID Number : CB19042

Date : 09/02/2023



hiDoc: DOCTOR APPOINTMENT BOOKING  
WEB APPLICATION

NUR NADHIRAH BINTI YAHAYA

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Computer Science (Software Engineering) With Honours

Faculty of Computing  
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

## **ACKNOWLEDGEMENTS**

First and foremost, praise and thanks to Allah SWT, the Almighty, for His blessings throughout my project work to successfully complete this thesis.

I'd like to express my heartfelt appreciation to my project supervisor, Dr Muhammad' Arif Bin Mohamad, for giving me the opportunity to work on this project and for providing invaluable guidance throughout. He has given me direction and advice on how to improve and increase the quality of my project. I am deeply grateful for what he has provided for me. I'd like to express my heartfelt gratitude to his wife and family for their acceptance and patience during our discussion about project work and thesis preparation.

I am grateful to my parents for their love, prayers, care, and sacrifices in educating and preparing me for the future. I am grateful to them for their understanding, patience, and ongoing support in helping me complete this project work. My heartfelt gratitude goes to my friends for their assistance and support throughout my project work and thesis preparation.

## ABSTRAK

Tempahan temu janji doktor dalam talian kini lebih mudah dan cekap daripada pergi ke klinik atau membuat panggilan telefon. Tidak banyak laman web dengan maklumat mengenai klinik kerajaan dan bukan kerajaan. Terdapat juga beberapa laman web yang tidak membenarkan pesakit membuat temujanji dengan doktor pilihan mereka. Sebagai contoh, hanya terdapat beberapa laman web yang memberikan maklumat mengenai klinik di Pekan, Pahang dan tidak ada laman web yang mengandungi maklumat mengenai klinik di Pekan yang membolehkan penduduk Pekan membuat temujanji dengan doktor. Justeru, matlamat projek ini adalah untuk membolehkan pengguna membuat temujanji dengan klinik kerajaan atau bukan kerajaan di daerah Pekan, Pahang di dalam satu laman web, dengan pilihan memilih doktor pilihan mereka. Terdapat lima modul dalam aplikasi ini: urus akaun, urus tempahan janji temu, urus rekod perubatan, urus maklumat ubat dan urus klinik. Modul-modul ini dilaksanakan berikutan penyelidikan yang meluas antara sistem sedia ada dan kertas penyelidikan untuk mewujudkan aplikasi tempahan temu janji doktor yang lebih baik yang boleh memberi manfaat kepada masyarakat.

## **ABSTRACT**

Booking a doctor's appointment online is now more convenient and efficient than going to the clinic or making phone calls. There aren't many websites with information on both government and non-government clinics. There are also a few websites that allow patients to book an appointment with the doctor of their choice. For example, there are only a few websites that provide information on clinics in Pekan, Pahang and there is no website that contains information about the clinics in Pekan that allow the Pekan residents to book an appointment with the doctor. Thus, the goal of this project is to allow users to book appointments with government or non-government clinics in district Pekan, Pahang, with the option of selecting the doctor of their choice. There are five modules in this application: manage account, manage appointment booking, manage medical records, manage medicine information, and manage clinic. These modules are implemented following extensive research between existing systems and research papers to create a better doctor appointment booking application that can benefit society.

## **TABLE OF CONTENT**

<b>DECLARATION</b>	
<b>TITLE PAGE</b>	
<b>ACKNOWLEDGEMENTS</b>	<b>ii</b>
<b>ABSTRAK</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENT</b>	<b>v</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>14</b>
1.1 Introduction	14
1.2 Problem Statement	14
1.3 Objective	15
1.4 Scope	15
1.5 Significance of Project	16
1.6 Report Organization	17
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>18</b>
2.1 Introduction	18
2.2 Literature Analysis	18
2.2.1 Literature review table for 3 articles	21
2.2.2 Conclusion based on a review of the three articles	22

2.3	Review of Existing Systems	23
2.3.1	DOC2US	23
2.3.2	DoctorOnCall	24
2.3.3	BookDoc	27
2.3.4	Comparison with Other Existing Systems	29
2.4	Proposed Application	31
2.5	Laravel Framework	32
2.6	Summary	33
<b>CHAPTER 3 METHODOLOGY</b>		<b>34</b>
3.1	Introduction	34
3.2	Project Management Framework	34
3.2.1	Requirements	35
3.2.2	Design	35
3.2.3	Development	35
3.2.4	Testing	35
3.2.5	Deployment	35
3.2.6	Review	35
3.3	Project Requirement	36
3.3.1	Functional Requirement	36
3.3.2	Non-Functional Requirement	37
3.3.3	Constraints and Limitations	38
3.4	Proposed Design	38
3.4.1	Context Diagram	39
3.4.2	Use Case Diagram	40
3.4.3	Manage Account	40

3.4.4	Manage Appointment Booking	44
3.4.5	Manage Medical Record	47
3.4.6	Manage Medicine Information	50
3.4.7	Manage Clinic	53
3.5	Data Design	56
3.5.1	ERD	56
3.5.2	Database Dictionary	57
3.6	Proof of Initial Concept	60
3.6.1	All users can access	61
3.6.2	Admin Interface	67
3.6.3	Clinic Interface	70
3.6.4	Doctor Interface	77
3.6.5	User Interface	82
3.7	Testing Plan	86
3.8	Potential Use of Proposed Solution	88
<b>CHAPTER 4 IMPLEMENTATION, RESULT AND DISCUSSION</b>		<b>89</b>
4.1	INTRODUCTION	89
4.2	DEVELOPMENT TOOLS	89
4.3	IMPLEMENTATION	89
4.3.1	Installing Software Tools	90
4.3.2	Collect Multimedia Contents	92
4.3.3	Development of Doctor Booking System	93
4.4	Testing	126
4.5	Result and discussion	127
4.5.1	Section 1: Basic Question	128

4.5.2	Section 2: Testing Results	129
4.5.3	Section 3: Comments	130
<b>CHAPTER 5 CONCLUSION</b>		<b>131</b>
5.1	Introduction	131
5.2	Research Constraint	131
5.3	Future Work	132
<b>REFERENCES</b>		<b>133</b>
<b>APPENDIX A: USER ACCEPTANCE TEST</b>		<b>135</b>
<b>APPENDIX B: SRS</b>		<b>140</b>
<b>APPENDIX C: SDD</b>		<b>198</b>



## LIST OF TABLES

Table 2.1	Literature Review for 3 Articles	21
Table 2.2	Comparison of three existing systems	29
Table 3.1	Functional Requirement	36
Table 3.2	Non-Functional Requirement	37
Table 3.3	Constraints and Limitations	38
Table 3.9	Database Dictionary for ‘users’	57
Table 3.10	Database Dictionary for ‘clinic’	57
Table 3.11	Database Dictionary for ‘doctor’	58
Table 3.12	Database Dictionary for ‘patient’	58
Table 3.13	Database Dictionary for ‘appointment’	58
Table 3.14	Database Dictionary for ‘records’	59
Table 3.15	Database Dictionary for ‘medicine’	59
Table 3.16	User Acceptance Test	86
Table 4.1	Software Development Tools	89

## LIST OF FIGURES

Figure 2.1	DOC2US Logo	23
Figure 2.2	DOC2US Interfaces	24
Figure 2.3	DoctorOnCall Logo	25
Figure 2.4	DoctorOnCall Interfaces	26
Figure 2.5	BookDoc Logo	27
Figure 2.6	BookDoc Interfaces	28
Figure 3.1	Agile System Development Process	35
Figure 3.2	Context Diagram	39
Figure 3.3	Use Case Diagram of Entire Application	40
Figure 3.4	Use Case Diagram of Manage Account	41
Figure 3.5	Sequence Diagram for Manage Account	43
Figure 3.6	Activity Diagram for Manage Account	43
Figure 3.7	Use Case Diagram for Manage Appointment Booking	44
Figure 3.8	Sequence Diagram for Manage Appointment Booking	45
Figure 3.9	Activity Diagram for Manage Appointment Booking	46
Figure 3.10	Use Case Diagram for Manage Medical Record	47
Figure 3.11	Sequence Diagram for Manage Medical Record	48
Figure 3.12	Activity Diagram for Manage Medical Record	49
Figure 3.13	Use Case Diagram for Manage Medicine Information	50
Figure 3.14	Sequence Diagram for Manage Medicine Information	51
Figure 3.15	Activity Diagram for Manage Medicine Information	52
Figure 3.16	Use Case Diagram for Manage Clinic	53
Figure 3.17	Sequence Diagram for Manage Clinic	54
Figure 3.18	Activity Diagram for Manage Clinic	55
Figure 3.19	Entity Relationship Diagram (ERD)	56
Figure 3.20	Main Page Interface	61
Figure 3.21	Main Page - When patient or user login Interface	62
Figure 3.22	Login Interface	63
Figure 3.23	Register Interface	64
Figure 3.24	List of clinics Interface	65
Figure 3.25	List of medicines Interface	66
Figure 3.26	Admin Homepage Interface	67
Figure 3.27	User Management Interface	68

Figure 3.28	Medicine Management Interface	69
Figure 3.29	Get Started Interface	70
Figure 3.30	Clinic Homepage	71
Figure 3.31	Doctor Management Interface	72
Figure 3.32	Profile Interface	73
Figure 3.33	Appointment Interface	74
Figure 3.34	Medical Record Interface	75
Figure 3.35	Patient Medical Record	76
Figure 3.36	Get Started Interface	77
Figure 3.37	Doctor Homepage	78
Figure 3.38	Profile Interface	79
Figure 3.39	Appointment Interface	80
Figure 3.40	Patient Medical Record Interface	81
Figure 3.41	User Homepage	82
Figure 3.42	Profile Interface	83
Figure 3.43	Appointment Interface	84
Figure 3.44	Patient's Medical Record Interface	85
Figure 4.1	Installing Visual Studio Code	90
Figure 4.2	Installing Xampp	91
Figure 4.3	Installing Laravel Framework	92
Figure 4.4	Composer Command	92
Figure 4.5	Multimedia Content	93
Figure 4.6	Website Main page	94
Figure 4.7	Login	95
Figure 4.8	Register	96
Figure 4.9	Search doctor	97
Figure 4.10	Manage Clinic Module (Clinic List)	98
Figure 4.11	Manage Medicine Module (User View)	99
Figure 4.12	Admin Homepage	100
Figure 4.13	Manage Account Module (User Management)	101
Figure 4.14	Manage Medicine Module (Medicine Management)	102
Figure 4.15	Manage Medicine Module (Add Medicine)	103
Figure 4.16	Manage Medicine Module (View Medicine)	104
Figure 4.17	Manage Medicine Module (Edit Medicine)	104
Figure 4.18	Clinic Get Started	105

Figure 4.19	Insert Clinic Information	106
Figure 4.20	Clinic Homepage	107
Figure 4.21	Manage Account Module (Doctor Management)	108
Figure 4.22	Clinic Profile	109
Figure 4.23	Appointment List or History	110
Figure 4.24	Update status	111
Figure 4.25	Appointment Email Cancellation	112
Figure 4.26	Medical Record Status	112
Figure 4.27	Add Medical Record	112
Figure 4.28	Medical Record Column Changes Button	113
Figure 4.29	Patient's Medical Record	113
Figure 4.30	Get Started	114
Figure 4.31	Insert Doctor Information	115
Figure 4.32	Doctor Homepage	116
Figure 4.33	Doctor Profile	117
Figure 4.34	Appointment List & History	118
Figure 4.35	Patient's Medical Record	119
Figure 4.36	User Homepage	120
Figure 4.37	User Profile	121
Figure 4.38	Appointment List & History	122
Figure 4.39	Doctor Section at Main Page	123
Figure 4.40	Make an Appointment with Doctor Option	124
Figure 4.41	Button to view clinic list	124
Figure 4.42	Clinic List	125
Figure 4.43	Doctor List in Each Clinic	125
Figure 4.44	Patient's Medical Record	126

## LIST OF ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheets
GUI	Graphical User Interface
HTML	Hypertext Markup Language
MVC	Model-View-Controller
PHP	Hypertext Pre-processor
UML	Unified Modelling Language Application Programming Interface

## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

Many fatal diseases can be avoided and treated if caught early. People, on the other hand, prefer to ignore early symptoms of an illness because they are mild. By the time they realise they have the illness, the infection has spread throughout their body. It is critical to have a regular body check-up to avoid future health problems.

Patients' hectic daily schedules make it difficult for them to go in person and book a medical visit, and they frequently forget to book through contact. This causes a delay in treatment and may result in a serious health problem later. Furthermore, if the patient does not have an appointment, they will have to wait for a long time if they want to see the doctor that day. As we all know, time is money, and waiting for your turn wastes a lot of it, which is why addressing these issues is critical. By resolving these issues, it is possible to avoid treatment delays and save many lives by defeating the disease at an early stage and preventing the patient's condition from deteriorating.

Doctor Appointment Booking Web Application (hiDoc) is a project that aims to create a website where patients can book appointments online with any clinic in the Pekan district, whether government or non-government. Users will be able to view a list of doctors with their expertise or specialisation in each clinic. Patients can then book an appointment with the doctor of their choice by specifying the preferred date and time.

#### 1.2 Problem Statement

Booking a doctor's appointment online is now faster and more efficient than visiting the clinic or making phone calls. The current doctor booking appointment only includes features such as booking an appointment based on date, time, location, and what services

they can provide. And there aren't many websites that provide information on both government and non-government clinics in the Pekan district. There are also few websites that allow patients to book an appointment with the doctor of their choice.

The application for scheduling doctor appointments for all clinics in the Pekan district has not yet been built. An application that allows patients to arrange appointments at all clinics is extremely convenient and beneficial. Patients can choose which clinic they wish to visit and do not need to search online for an appointment at that facility. They can also view the services that each clinic provides.

Furthermore, developing a website for doctor appointment booking will allow Pekan residents to receive medical treatment more quickly because they may make an appointment at any time and from any location if they have access to the internet. Nowadays, it has become increasingly difficult to obtain an appointment for a medical problem or a routine check-up (U. Chauhan,2022). Patients' hectic daily schedules nowadays make it tough for them to go and book a medical visit in person or over the phone. In addition, patients who do not have an appointment must wait a long time for their turn. They do not have to wait for a long time if they book an appointment online by selecting the date and time they choose, and they only need to be at the clinic at the appropriate time and date for the appointment.

### **1.3 Objective**

- i. To study the existing doctor appointment booking.
- ii. To develop a website for Doctor Appointment Booking Web Application using Visual Studio Code by implementing Laravel framework.
- iii. To evaluate the functionality of the proposed doctor appointment booking web application using the User Acceptance Testing (UAT) method.

### **1.4 Scope**

- i. User Scope:
  - a. Admin
  - b. Each clinic in Pekan has a staff representative.
  - c. Each clinic in Pekan has a doctor.

- d. Patient
- ii. System Scope:
  - a. Covered appointment booking at any clinic in district Pekan
- iii. Development Scope:
  - a. Contains multimedia elements such as graphic and text.
  - b. Developing a website with the Laravel framework using the Visual Studio Code software.

### **1.5 Significance of Project**

- i. Clinic Representative
  - a. Clinic representative can know the difference of services or treatments that each clinic provides. This will help the other clinic to continue striving to improve their clinic
- ii. Patient
  - a. Patients can keep up to date knowledge of any new medicine and its information
  - b. Patients do not need to wait for a long time if they already made an appointment
  - c. Patients can choose the doctor they want to make an appointment with
  - d. Patients can view their appointment history
  - e. Patient can view their medical record for each appointment session.
- iii. Doctor
  - a. Doctors can keep track of their appointments
  - b. Doctors can view their appointment history



## **1.6 Report Organization**

This thesis consists of three chapters. Chapter one discusses the introduction to the project doctor appointment booking website. The problem statement, objectives, scope, and significance of the project are also discussed in Chapter 1.

The second chapter discusses a review of the literature on three articles related to appointment booking systems, as well as a review of three existing systems related to this project.

The methodology used in this project is discussed in Chapter 3. Agile methodology is used in this project. This project's stages are planning, requirements, analysis and design, implementation, evaluation, and launch.

The results and discussions based on this project's development and testing are explained in Chapter 4. All of the project's output and results were briefly discussed in this chapter. These include the creation of software, the testing of applications, and the project's outcome.

The conclusion and summary of this project's results are described in Chapter 5. In this chapter, the limitation and upcoming works were covered in detail.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

The second chapter looks at articles and existing applications of appointment booking systems in the health care sector. There are three articles and three existing systems that are thoroughly explained and compared regarding the appointment booking system. The comparisons in this article will concentrate on approaches and outcomes, whereas the comparisons between the three existing systems will concentrate on the target audience, functions provided, clinic availability information in Pekan Pahang district, and Graphical User Interface (GUI), and advantages and disadvantages. These comparisons are conducted to ensure that the Doctor Appointment Booking Web Application developed in this project is a better version that will benefit users.

#### 2.2 Literature Analysis

##### 1) Literature review towards Get-a-Doc: A Doctor Recommender System

This article proposed a web application for finding the best match doctor for the user without having to go through a list of doctors. The user only needs to provide information about his location and the type of doctor he requires. The application will detect a doctor who meets the user's requirements. This application is proposed because it is difficult to find the right doctor for treatment due to the overwhelming number of options available through online media, and most existing applications do not assist users in finding the perfect match for a doctor based on their needs.

##### 2) Literature review towards User Interactive Hospital Management System by using a Web application

This article discusses the development of a web application for hospitals, with the author's goal of maintaining a database that can store and provide information to patients, doctors, and hospital management all in one application.

The application will provide:

- i. Registration of patients
- ii. Details sorting the patient's details
- iii. User can search about the doctor's availability to book an appointment
- iv. Provide online prescriptions including lab prescription
- v. Reminder for appointment will be sent
- vi. Lab technicians can upload medical reports online

The Hospital Management System is useful for improving hospital management in the areas of clinical process analysis and activity-based costing. This system will aid in the development of an organization by increasing the effectiveness and quality of work. ID, username, and password are required to access the hospital management system. Doctors, patients, and administrators can all use it. This proposed system will assist hospitals in completing tasks quickly and efficiently while avoiding major changes to the existing system and updating records solely through the online website.

### 3) Literature review towards Health Related Web Application: Virtual Hospital

In everyday life, people order food and taxis in minutes and occasionally watch livestock market points and sports results on their mobile phones; whereas many businesses are successful only because they can meet the needs with greater management. This technology can be used to provide people all over the world with a simple Virtual Hospital web system that helps patients, doctors, labs, and chemists by simplifying the Hospital Health Care process. This will also assist hospital personnel in reducing paperwork and administrative responsibilities. Patients can also schedule an appointment and an online examination with a doctor. This web application is easily accessible from mobile phones, laptops, and desktop computers. This web application also can be partnered with an authorized pharmacy to provide quality medicines at a discounted price. The aim is to help the elderly and the disabled and people who have simple flu or other diseases to gain treatment from doctors at low Door-step rates. This evaluation service is thought to reduce hospital admissions and general distress.

How the application works:

- i. Administrators can add or remove doctors, labs, and chemicals.
- ii. Administrators have access to the user, doctor, lab, and pharmacy information.
- iii. Administrators can archive and restore users.
- iv. Administrators can add and remove special icons, view function, and program statistics, view and send messages, update profiles, and change passwords.
- v. Patients can schedule appointments, review medical information, and review prescriptions after each appointment, as well as review medical tests.
- vi. A patient can view and send messages, generate a prescription invoice, update a profile, and change their password.
- vii. A doctor can view or generate instructions, view patient medical information, update profiles, and change passwords.

### 2.2.1 Literature review table for 3 articles

Table 2.1 Literature Review for 3 Articles

No	Title	Author	Database published	Approaches	Outcome	Year Published
1	Get-a-Doc: A Doctor Recommender System	Abhinav Singh Isha Goyal Parmeet Kaur Chetna Dabas	IEEE	Used Python, HTML, CSS. Used Django framework and Google API. Used the K-Means algorithm to retrieve a perfect match to find the doctor. Used MongoDB to store data . Used sharding to distribute data on multiple servers.	Using web application as front end. Used MongoDB as database and was sharded by having 1 config server and three sharded servers to distribute data on multiple servers.	2018
2	User Interactive Hospital Management System by using a Web application	S. Sharmila Devi J.S.Deepica K. Dharshini G. Dhivyashree	IEEE	Used Spring MVC Framework to establish servlet request mapping to controller classes. Used MySQL for database Used CSS and HTML to	Using web application as front end. Has ten modules (Login, Admin, Patients, Doctors, Appointment Bookings, Online Prescriptions and Medical History).	2021

				develop user-interactive system.	Used Spring MVC framework, MySQL as backend.	
3	Health Related Web Application: Virtual Hospital	Hiren Chafekar Amit Sundas Shabnam Sharma	IEEE	Used Python programming language. Used Heroku as a cloud service platform. Used HTML to create web pages and application. Used CSS to design web pages.	Using web application as front end	2021

### 2.2.2 Conclusion based on a review of the three articles

Among the three articles reviewed, "Get-a-Doc: A Doctor Recommender System" proposed a web application for finding the best match doctor for the user without having to go through a list of doctors. The user only needs to specify his location and the type of doctor he requires. I incorporate the idea from this article about finding the best match doctor by allowing the user to search for the doctor based on the doctor's specialization.

Furthermore, I used the idea from the article Health Related Web Application: Virtual Hospital as a requirement in the manage account module of allowing users to update their profile, clinic's staff, and doctor to view a few information of the patients. I also used the idea from this article that stated to upload a medical report, but for this system, the clinic staff will input the diagnosed from the doctor and allow the patient to view the medical report for each appointment session.

Finally, the articles above show how to build a website for a healthcare or booking application using HTML and CSS as the front-end language. The modules between the

three articles are also nearly identical. As a result, I will be developing this website using HTML, CSS, and PHP to make it more interesting and up to date with the latest trends.

## **2.3 Review of Existing Systems**

### **2.3.1 DOC2US**

DOC2US is a website that allows you to text chat with a doctor or other healthcare professional at any time and from any location. Through a simple platform, this application connects patients to a healthcare professional for immediate advice. They offer a blockchain electronic health record to users to help them manage their health records in a secure and interoperable manner.



Figure 2.1 DOC2US Logo

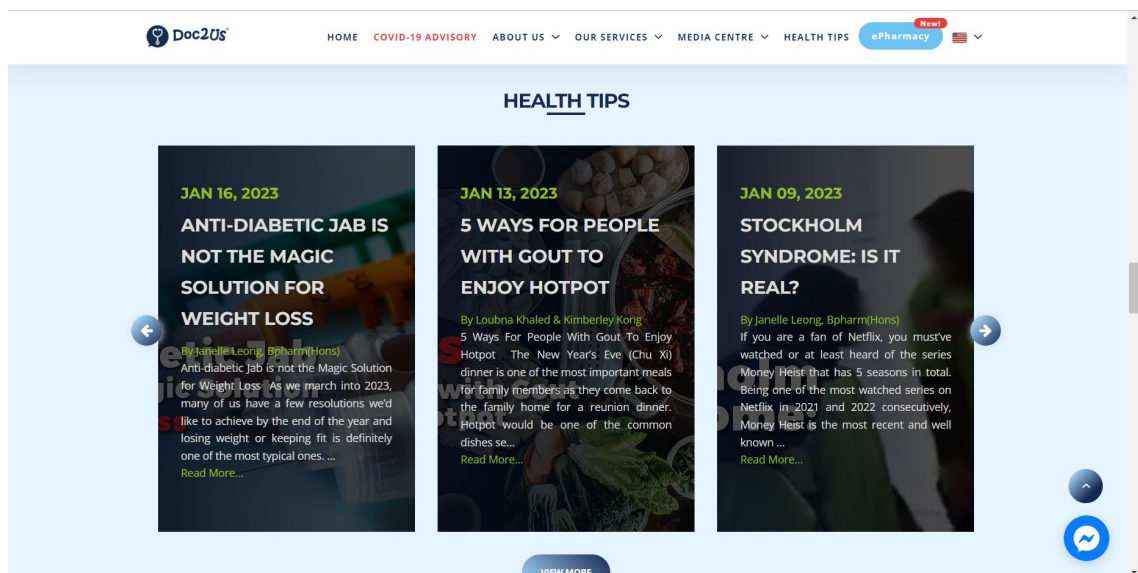
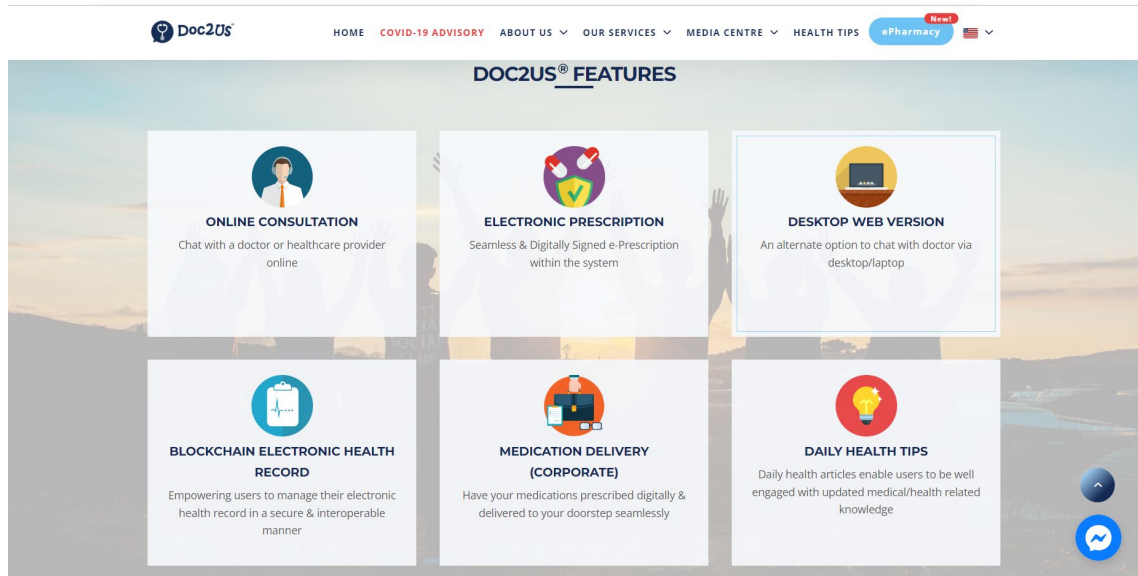


Figure 2.2 DOC2US Interfaces

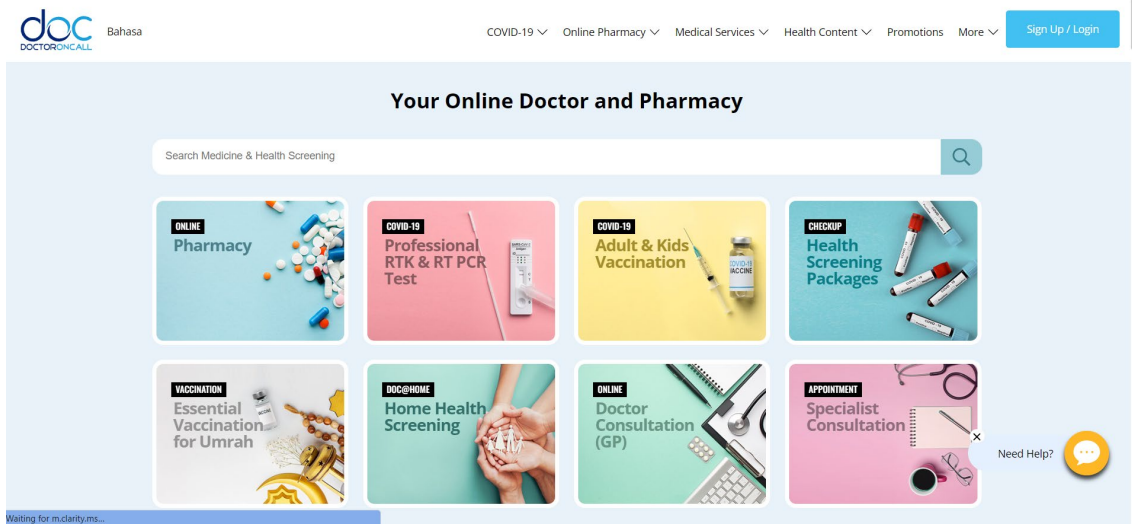
### 2.3.2 DoctorOnCall

DoctorOnCall is Malaysia's first and largest digital healthcare platform, offering more convenient and affordable medication access via their online pharmacy. They provide online pharmacies and video consultations as well as other digital health services. This app provides low prices, quick service, and ease of use.





Figure 2.3 DoctorOnCall Logo



**Dr. Tan Leng Leng**  
 24 Years Experience  
 14 Patients

**Practices**  
 Thomson Hospital Kuala Damansara

**Specialization**  
 Dermatologist

**Qualifications**  
 MBBS (UK), MRCP (UK), FRCR (Edinburgh), MR(UK), Advanced Master of Dermatology (UK)

**Languages**  
 English, Malay, Mandarin

**Conditions Consulted**  
 Hypertension, Diabetes, Psoriasis/Seborrheic Dermatitis, Tinea, Acne/Keratosis

**Services Consulted**  
 Medication Treatment, Laser Treatment For Hair Removal, Skin Exfoliation, Lipoma Treatment And Removal, Cryotherapy or Cold Therapy

**Step 1: Choose Consultation Type**  
 Choose your consultation type

**Step 2: Choose Date & Time**  
 Choose date and time from available slots

**Asumalife 1mg Tablet 1s**  
 It is advised to speak to a medical professional before consuming this product.  
 RM 0.60

**Getting your medicine from DOC is AFFORDABLE, CONVENIENT, and SAFE!**

**How to Use** | Overview | Precautions | FAQs | **Side Effects**

As with all medications, you may experience some side effects. Not everyone experiences it. This is not a cause for concern. However, it is so that you are aware of it.

Undefined  
 Drowsiness, Dizziness, Headache, Irritability, Insomnia, Flu-like symptoms, Dry mouth, Blurring of vision, Dry eyes, Burning sensation in the eyes

Return/Refund Policy

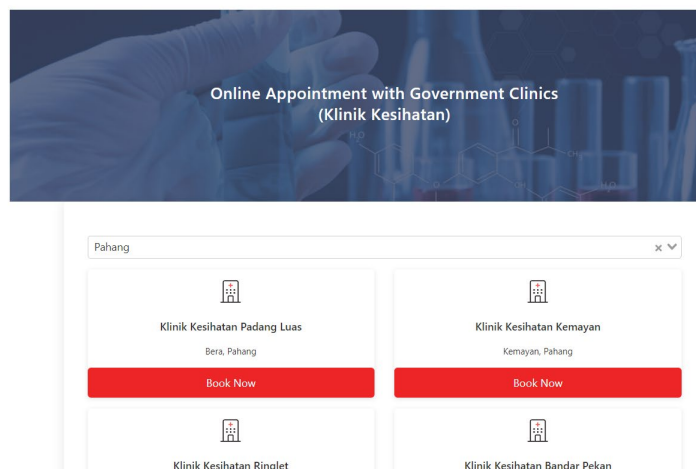
Figure 2.4 DoctorOnCall Interfaces

### 2.3.3 BookDoc

BookDoc, which operates in five countries and twenty cities, including Malaysia, Singapore, Indonesia, Hong Kong, and Thailand, is an online platform that connects patients to healthcare professionals at any time and from any location, while also incentivizing everyone to stay active.



Figure 2.5 BookDoc Logo



# Online Appointment

Home / Search & Book / Clinic

Operation Hours	
Mon	: 08:00 AM - 05:00 PM
Tue	: 08:00 AM - 05:00 PM
Wed	: 08:00 AM - 05:00 PM
Thurs	: 08:00 AM - 05:00 PM
Fri	: 08:00 AM - 05:00 PM
Sat	: Not Available
Sun	: Not Available
PH	: Not Available

## Klinik Kesihatan Peramu jaya

Pusat Kesihatan Peramu Jaya, 3, 26600 Pahang



Get Direction



Welcome back, let us know if you have any questions.

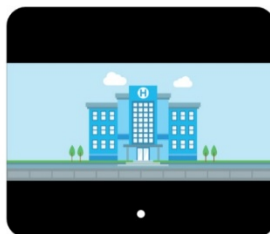
Write a message...



Home About Us Activ **Services** Newsroom Careers [Sign In](#)

# Online Appointment

Home / Search & Book / KK Peramu Jaya - Demam/Batuk/Selsema Slot 1



## KK Peramu Jaya - Demam/Batuk/Selsema Slot 1

General Practitioner (GP Doctor)

[Online Booking](#)

### Schedule

Wed, 18 Jan 2023					
08:00 AM	08:30 AM	09:00 AM	09:30 AM	10:00 AM	10:30 AM
11:00 AM	11:30 AM	02:00 PM	02:30 PM	03:00 PM	03:30 PM

**Profile** Services Hours

D  
Credentials  
Qualifications :  
Language  
Bahasa Malaysia,English

### Contact

**Klinik Kesihatan Peramu Jaya**  
Pusat Kesihatan Peramu Jaya, 3, 26600 Pahang



Get Direction

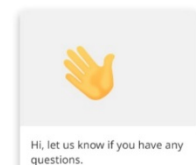


Figure 2.6 BookDoc Interfaces

### 2.3.4 Comparison with Other Existing Systems

Table 2.2 Comparison of three existing systems

	<b>DOC2US</b>	<b>DoctorOnCall</b>	<b>BookDoc</b>
<b>Graphical User Interface (GUI)</b>	The interface is simple and equipped with the information needed. The interface's colour scheme is also pleasing to the eye.	The interface is well organized but too pack with information. The interface's colour theme does not bring discomfort to the eye. Having difficulties to navigate around the interface.	It is very user-friendly. The colour scheme used for interface is also easy on the eye. It is very easy to navigate around this website.
<b>Target Audience</b>	Above 18	Above 18	Above 18
<b>District Pekan Clinic Information</b>	None	None	Yes, but only a few (government)
<b>Features</b>	<ul style="list-style-type: none"> <li>i. Map view</li> <li>ii. Online consultation</li> <li>iii. Electronic prescription</li> <li>iv. Desktop Web version</li> <li>v. Blockchain electronic health record</li> <li>vi. Medication delivery (corporate)</li> <li>vii. Daily health tips</li> </ul>	<ul style="list-style-type: none"> <li>i. Online purchase</li> <li>ii. User's review</li> <li>iii. Health screening</li> <li>iv. Consult a specialist</li> <li>v. Covid-19 testing</li> <li>vi. Online pharmacy</li> <li>vii. Talk to a doctor now (GP)</li> <li>viii. Covid-19 Vaccination and Booster Shot</li> </ul>	<ul style="list-style-type: none"> <li>i. Filter</li> <li>ii. Map view</li> <li>iii. Marketplace</li> <li>iv. Events &amp; news</li> <li>v. Health coaching</li> <li>vi. Tele-Consult</li> <li>vii. Book appointment</li> </ul>
<b>Advantage</b>	<ul style="list-style-type: none"> <li>i. Contain numerous</li> </ul>	<ul style="list-style-type: none"> <li>i. Contain diverse</li> </ul>	<ul style="list-style-type: none"> <li>i. Can book an appointment</li> </ul>

	<ul style="list-style-type: none"> <li>ii. services Can chat with a doctor via desktop, phone, or laptop</li> <li>iii. Provide daily health tips</li> <li>iv. Provide two languages (English and Malay)</li> </ul>	<ul style="list-style-type: none"> <li>ii. services (online pharmacy, book appointment, vaccine services, etc.)</li> <li>ii. Can consult with doctor or specialist via video call</li> <li>iii. Provide two languages (English and Malay)</li> </ul>	<ul style="list-style-type: none"> <li>in five different countries (Malaysia, Thailand, Singapore, Indonesia Hong Kong) based on health care type</li> <li>ii. Gps location for every clinic and pharmacy</li> <li>iii. Provide two languages (English and Malay)</li> </ul>
<b>Disadvantage</b>	<ul style="list-style-type: none"> <li>i. Cannot meet doctor face to face</li> <li>ii. Cannot book medicine for pickup</li> </ul>	<ul style="list-style-type: none"> <li>i. Does not have a map view</li> <li>ii. Cannot book medicine for pickup</li> </ul>	<ul style="list-style-type: none"> <li>i. Cannot purchase or book medicine online</li> </ul>

#### 2.3.4.1 Summary On Review Existing System

A comparison of three applications, DOC2US, DoctorOnCall, and BookDoc, reveals that each has advantages and disadvantages.

The three have good interfaces in terms of GUI, but the BookDoc app interfaces are more user-friendly and easier to navigate. The interfaces are simple, well-organized, and simple to use for a new user. The interface of DOC2US is also simple, but it provides the most valuable information on the application's front page. It describes the services that the application offers. DoctorOnCall's user interface is well-organized but too pack with information that makes it difficult to navigate around the website. The colours used in the three applications are appropriate and pleasing to the eyes.

All the applications covered topics related to website development and the target for these applications are above 18 years old. The navigation and icons in all the

applications are good, and the user can easily understand them because the application uses navigation and icons that are already well-known in other apps and daily life activities.

A significant proportion of the features shared by the three existing systems are similar, such as online consultation, language, and appointment booking. However, the difference between booking an appointment and DOC2US is that only DOC2US offers a face-to-face appointment. Each of these systems has its distinct feature, such as DOC2US, which offers daily health tips, and DoctorOnCall, which offers user reviews on their website as well as an online pharmacy. Finally, BookDoc offers health coaching. DOC2US and BookDoc both have a view map on their websites and apps, but DoctorOnCall does not. In addition, only BookDoc provides a map location for each clinic and pharmacy.

Each of these three applications has its own set of weaknesses. Users of DOC2US cannot request an appointment to meet with the doctor face to face. Users can only consult with the doctor via the internet. Next, DoctorOnCall lacks a map view that would allow the user to see where the clinic or pharmacy is located. Finally, unlike DOC2US and DoctorOnCall, BookDoc does not offer the option to purchase medications online. However, none of the three applications allow users to book medicine for pickup.

## **2.4 Proposed Application**

This project's proposed application is a doctor appointment booking website application. People over the age of 18 are the intended audience for this application. The website will be built with the Laravel framework. The development is based on an Integrated Application Development Framework course I took. To use this application, you must have an internet connection.

This proposed project will include five modules: manage account, manage clinic, manage medicine information, manage appointment booking, and manage medical records. The manage medical records module is a straightforward module that stores the patient's diagnosis during the appointment with the doctor. If the patient requests that the information be verified for medical purposes, the medical record will include a signature section for the doctor to sign. hiDoc will also provide a map for each clinic so that users

can see where the clinic is located. With this proposed application, these two features are the only ones that distinguish hiDoc from DoctorOnCall and DOC2US.

The proposed system's graphical user interface (GUI) will display all relevant information, such as service information, doctor profiles, clinic information, and so on. The GUI will be designed with enough detail to ensure that the user can easily navigate around the website. The GUI colour scheme will consist of colours that are pleasing to the eye, but not overpowering, and that complement one another. Colours that do not complement one another may cause a user to lose interest in the app. The GUI will also use appropriate and consistent icons, fonts, buttons, and font sizes so that the user can become familiar with how to use the app and ensure that the app is user-friendly. Moreover, the proposed app allows users to book appointments at any clinics in the Pekan district.

## **2.5 Laravel Framework**

The most popular method now to develop a website is by using Laravel Framework. Laravel framework is a web PHP framework that makes developer life easier to build a website. Laravel reuses existing components from various frameworks, which aids in the development of a web application. The resulting web application is more structured and pragmatic.

Laravel provides a comprehensive set of functionalities that incorporate the fundamental features of PHP frameworks. PHP (Hypertext Pre-processor) is a free and open-source general-purpose scripting language that is commonly used in web development. It is a powerful programming language used to create dynamic and interactive web applications. PHP is dominating the web development market, but to create a website, one must be familiar with the PHP language and its supporting tools, such as CSS and JavaScript (N. Yadav et al.,2019). Building a website in PHP takes time and requires a lot of technical knowledge. To solve these problems, we sometimes must write code multiple times in the same project (N. Yadav et al.,2019).

Laravel has a plenty of features that will accelerate web development. Furthermore, a Laravel-built website is secure and resistant to a variety of web attacks. Frameworks



are used to solve problems associated with the use of PHP. PHP frameworks encourage rapid application development (RAD), which saves time and aids in the development of a more stable application by reducing development time for developers (N. Yadav et al.,2019). Laravel implements its applications using the MVC method and the OOP (Object Oriented Programming) concept (Yu et al.,2010). The MVC programming pattern, which separates logic and view, makes programme code cleaner and easier to trace when troubleshooting or debugging (Yu et al.,2010).

The Laravel Framework also provides Progressive, Scalable, and Community Framework access security to information system developers. but the default system from Laravel 7 is still insufficient to provide a solution for data security and access levels for users. To overcome data security issues, access, and system damage, additional systems such as Secure Hashing Algorithm (SHA), Multi Level Access Algorithm (MLA), and Middleware Checking Access System are required (APPLICATION OF THE SECURE HASHING ALGORITHM (SHA) METHOD AND ROLE-BASED ACCESS SECURITY AUTHENTICATION IN LARAVEL 7, (n.d)). Thus, Laravel 9 will be used during the developing this project to ensure high security performance.

## **2.6 Summary**

The second chapter looked at articles and existing applications of appointment booking systems in the health care sector. There were three articles and three existing systems that were thoroughly explained and compared regarding the appointment booking system. The comparisons in this article concentrated on approaches and outcomes, whereas the comparisons between the three existing systems were concentrate on the target audience, functions provided, clinic availability information in Pekan Pahang district, Graphical User Interface (GUI), advantages and disadvantages. These comparisons were conducted to ensure that the Doctor Appointment Booking Website developed in this project is an appropriate and modern version that will benefit users. This chapter also went into detail about the Laravel framework. The Laravel framework will make hiDoc development easier. It saves time and is also safe to use because of the high security that comes when developing using Laravel..

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

Chapter 3 will be discussed the framework that will be used in this project. This chapter will also go over this project in detail, explaining its functional requirements, non-functional requirements, constraints, and limitations.

#### 3.2 Project Management Framework

The Agile methodology is used for system development in this project. Agile improve flexibility during software development, allowing the individuals involved to work more efficiently (Altameem,2015). Furthermore, because it allows developers to test the system in each iteration, agile methodology aids in the detection of faulty code in a timely manner (Altameem,2015). Agile divides system development into small incremental steps. The iteration process is applied to every build. Each iteration includes every phase of development, beginning with the requirement process, moving on to the design process, development process, testing, deployment, and review process, as shown in Figure 3.1.

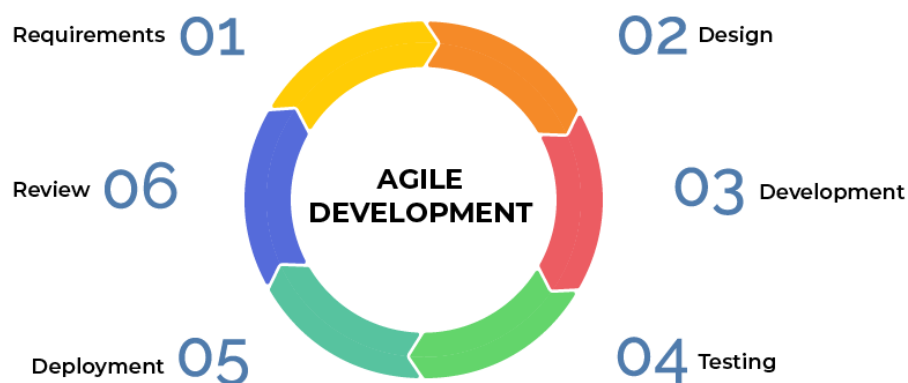


Figure 3.1 Agile System Development Process

### **3.2.1 Requirements**

System requirements are conducted in this phase to determine the system's requirements. Article research and reviews of existing applications are used to determine user requirements. System specification that has been gathered in this phase, including the software and hardware required to develop the system, has been documented. This phase establishes the project's limitations and scope.

### **3.2.2 Design**

The design phase aids in the definition of the system architecture. This phase's output is used as the input for the next phase, which is the development. The design phases' output can include Unified Modelling Language (UML), database design, and interface design.

### **3.2.3 Development**

Following that, the development phase entails the development of the product through the requirements and design phase. The functions for patients were built using Laravel Framework with MVC architecture and Java as the programming language.

### **3.2.4 Testing**

The testing phase will identify the errors and deficiencies in the system's quality that must be addressed before deployment. The method that will be used during testing is User Acceptance Testing (UAT).

### **3.2.5 Deployment**

During the deployment phase, in this phase, the working application must be presented to the proposed application's users, who are the administrator, patient, clinic representative, and doctor.

### **3.2.6 Review**

The system will be reviewed by the requirements gathered to ensure the product's usability and quality. The review phase is the final stage of the agile methodology to

ensure that the system's goals and objectives are met, and that the user is satisfied with the system. If this goal is not met, the system will have to repeat the development and testing phases until the desired outcome is achieved and ready for deployment to all users involved. Agile methodology is the best methodology for this system because agile methods are more adaptable to requested changes, there is always direct collaboration between developers and customers (Alsaqqa,2020). Agile is more effective in small and medium-sized projects like hiDoc.

### 3.3 Project Requirement

#### 3.3.1 Functional Requirement

Table 3.1 Functional Requirement

NO	FUNCTION NAME	FUNCTION DESCRIPTION
1	Login	<ul style="list-style-type: none"> <li>i. The system allows the user to login into the application by entering a email and password.</li> <li>ii. The system allows the user to re-enter the password or email if either one is entered incorrectly.</li> </ul>
2	Sign Up	<ul style="list-style-type: none"> <li>i. The system allows the users to sign up by entering a name, password, email and their role type (Clinic, Doctor, User).</li> </ul>
3	Manage Appointment Booking	<ul style="list-style-type: none"> <li>ii. The system allows the patient to book an appointment based on the preferred doctor, date, and time.</li> <li>iii. The system allows the patient to cancel their appointment.</li> <li>iv. The system allows the clinic representative to accept or reject the appointment request from the patient.</li> <li>v. The system allows the clinic representative to cancel the accepted appointment request.</li> <li>vi. The system allows the users to view their appointment history.</li> </ul>
4	Manage Clinic	<ul style="list-style-type: none"> <li>i. The system allows the patient to view the clinic information.</li> </ul>

6	Manage Account	<ul style="list-style-type: none"> <li>i. The system allows the admin to delete any user account.</li> <li>ii. The system allows the user to update or edit their account profile.</li> <li>iii. The system allows the patient to view the doctor and clinic's information.</li> <li>iv. The system allows the clinic representative and doctor to view the patient's information that make an appointment with them.</li> </ul>
6	Manage Medicine Information	<ul style="list-style-type: none"> <li>i. The system allows the administrator to add, amend, and delete medical information.</li> <li>ii. The system allows the users to view all the medicine's information.</li> </ul>
7	Manage Medical Record	<ul style="list-style-type: none"> <li>i. The system allows the clinic representative to input the patient's diagnosis information after the patient finished their appointment session.</li> <li>ii. The system allows the patient to view their medical record.</li> <li>iii. The system allows the clinic representative and doctor to view the patient's medical record that have finished their appointment.</li> </ul>
8	Logout	<ul style="list-style-type: none"> <li>i. The system allows the user to log out from the application.</li> </ul>

### 3.3.2 Non-Functional Requirement

Table 3.2 Non-Functional Requirement

NO	REQUIREMENT	DESCRIPTION
1	Performance	The system can handle multiple sessions of users in a large group.
2	Usability	The system will use command features to make it easier for both new and experienced users to use.
3	Reliability	The system probability of failure performing in 10 seconds shall be 0.001 when a different session of a user in a large number using it at the same time.
4	Maintainability	The system can be used for a long lifespan. Maintainability will be one of the priorities for the system requirements.

### 3.3.3 Constraints and Limitations

This system also includes a few constraints that limit user action when using it. A constraint is a system requirement that limits how the system can be developed. It is significant because it can assist the development team in accelerating system development by preventing problems during the development process. These are some of the constraints in this system.

Table 3.3 Constraints and Limitations

<b>NO</b>	<b>TYPE OF CONSTRAINT</b>	<b>DESCRIPTION</b>
1	Usability	The maximum allowable response time should not be more than 5 seconds.
2	Scalability	The system must be configured to work with the Internet application.
3	Maintainability	Out of office hours, the system should always be maintained.
4	Security	Users can access the system if they enter their username and password correctly.

### 3.4 Proposed Design

In this section, the context diagram, use case diagram and description, activity diagram, and sequence diagram are prepared and are shown as follows:

### 3.4.1 Context Diagram

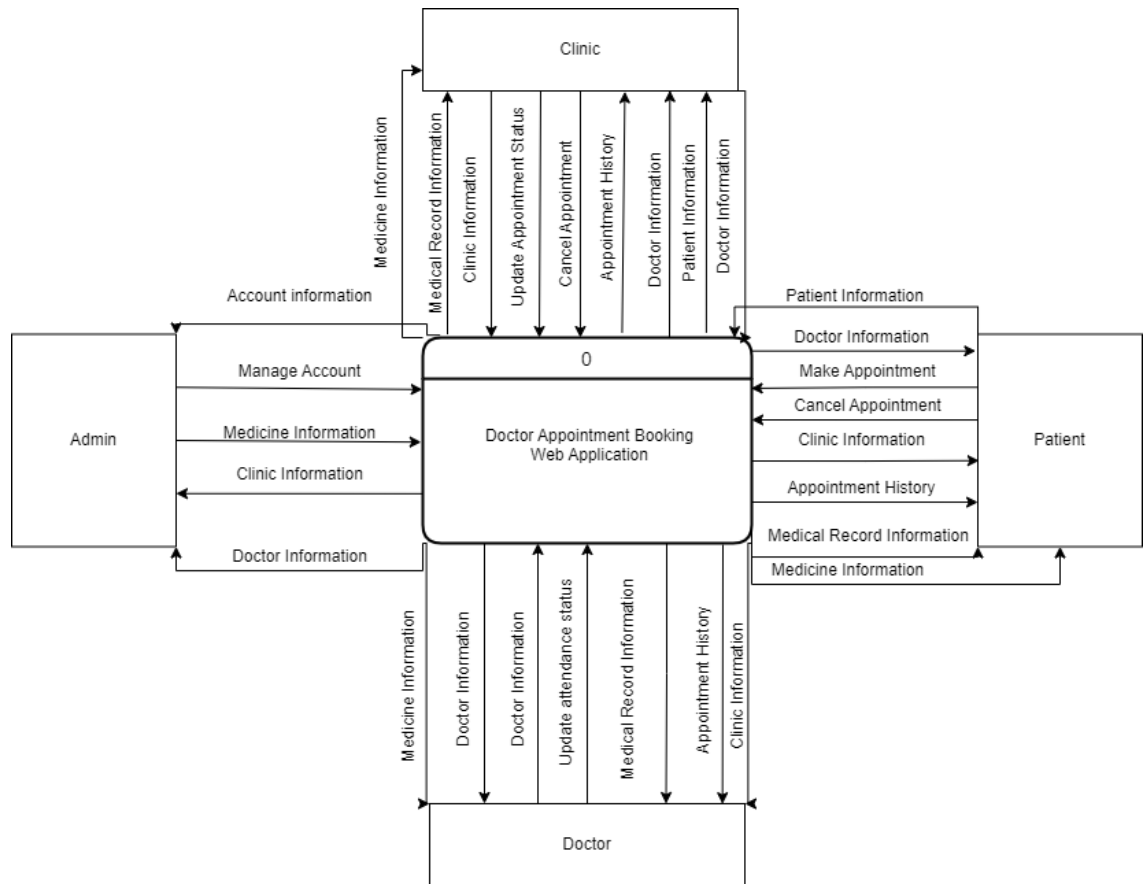


Figure 3.2 Context Diagram

The context diagram in the figure above contains four users which are admin, clinic representative, doctor, and patient. The context diagram is developed based on five modules which are manage account, manage appointment booking, manage medical records, manage medicine information, and manage clinic.

### 3.4.2 Use Case Diagram

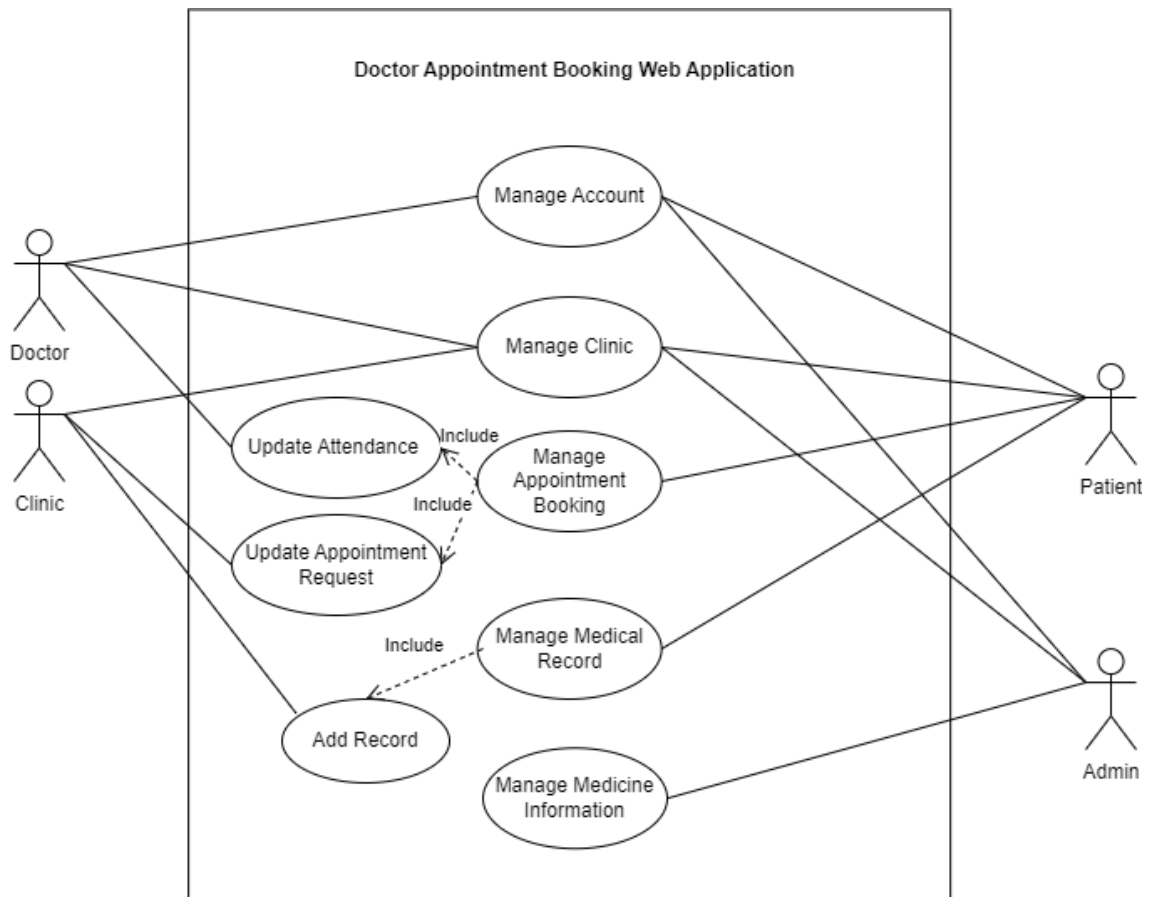


Figure 3.3 Use Case Diagram of Entire Application

The use case diagram in the figure above is for the entire application. There are four actors in the figure above: the admin, clinic representative, doctor, and patient. This application contains five modules which are manage account, manage booking appointments, manage medical record, manage medicine information, and manage clinic. Below is the use case diagram, use case description, activity diagram, and sequence diagram for all the five modules with detailed descriptions:

### 3.4.3 Manage Account

The admin, clinic representative, doctor, and patient can all access the Manage Account module. The clinic representative, the patient, and the doctor can all see and edit their profile information. The administrator can only delete and view any existing account. The use case diagram, activity diagram, and sequence diagram below provide a more complete explanation of this module:



### 3.4.3.1 Manage Account Use Case Diagram

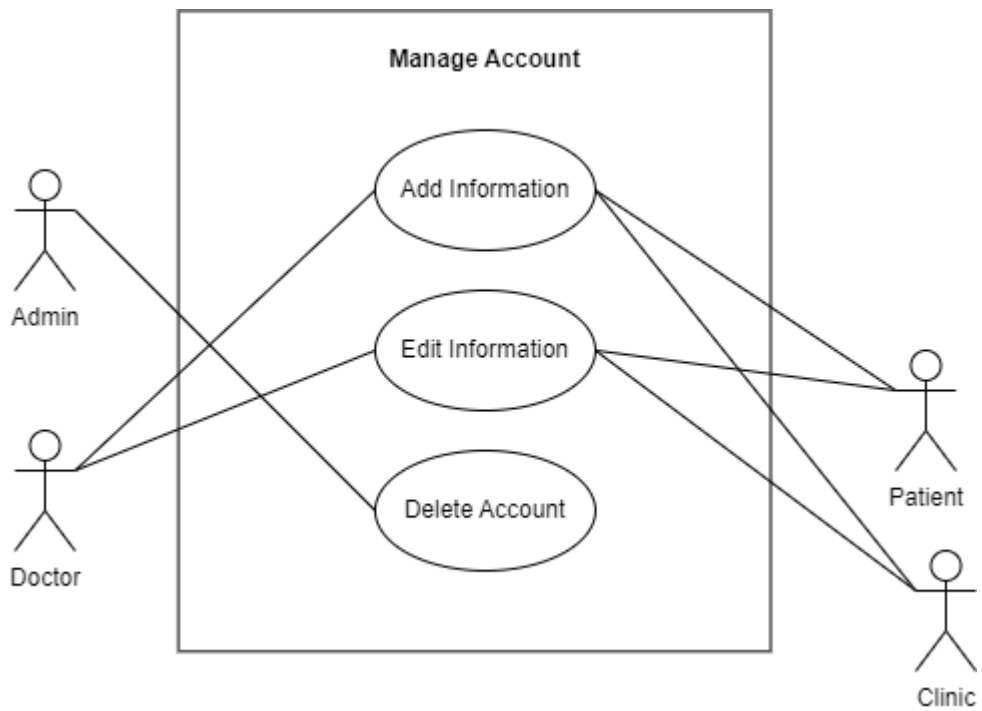
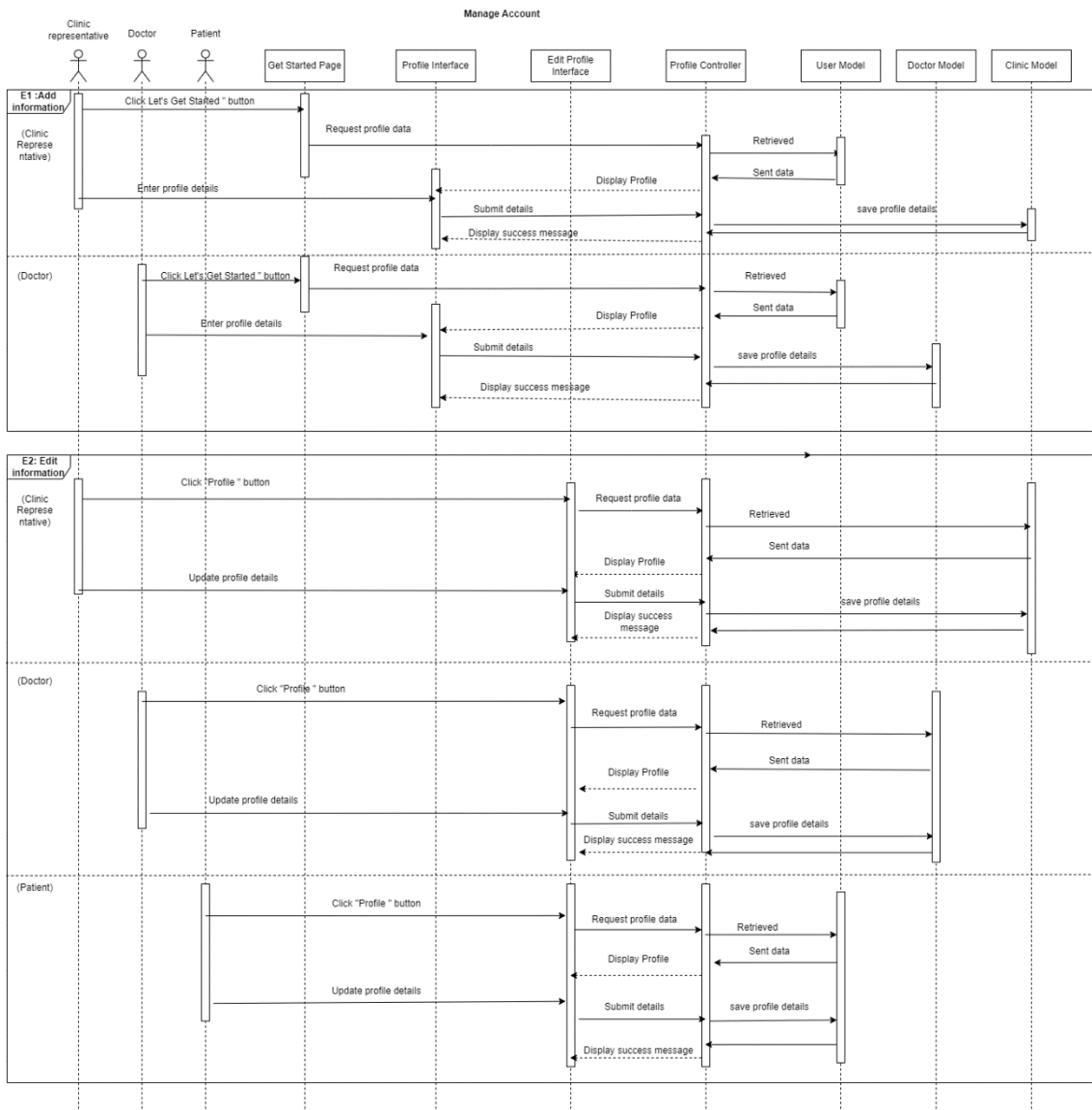


Figure 3.4 Use Case Diagram of Manage Account

### 3.4.3.2 Manage Account Sequence Diagram



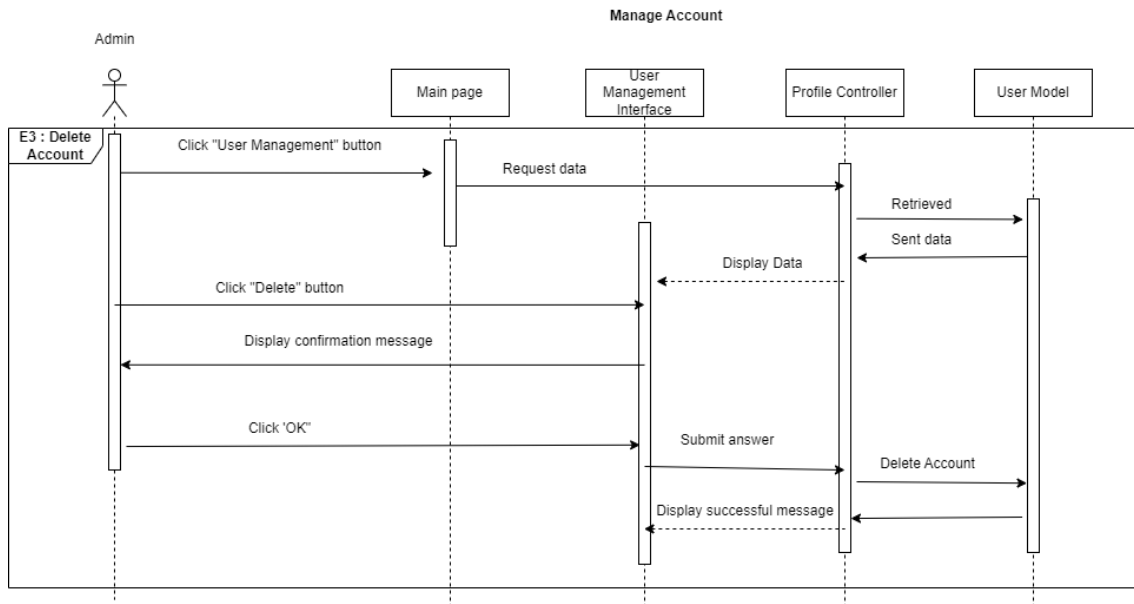


Figure 3.5 Sequence Diagram for Manage Account

### 3.4.3.3 Manage Account Activity Diagram

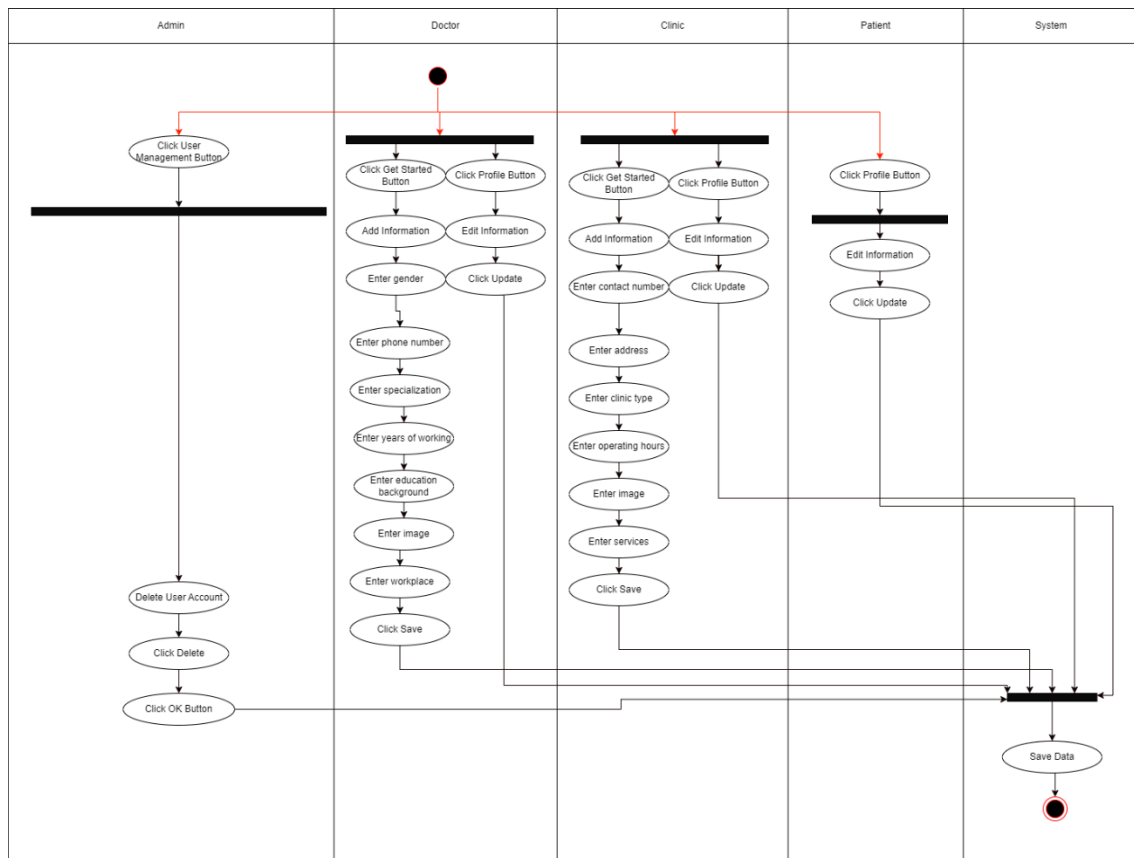


Figure 3.6 Activity Diagram for Manage Account

### 3.4.4 Manage Appointment Booking

The manage appointment booking module is accessible to the doctor, clinic representative and patient. The doctor has the authority to update the attendance status of patient while the clinic representative can approve or reject appointment requests from patient and update the attendance status. The doctor and the patient might also cancel a previously scheduled appointment. The appointment history records can be seen by the three users. This module is explained in greater detail in the use case diagram, activity diagram, and sequence diagram that follow:

#### 3.4.4.1 Manage Appointment Booking Use Case Diagram

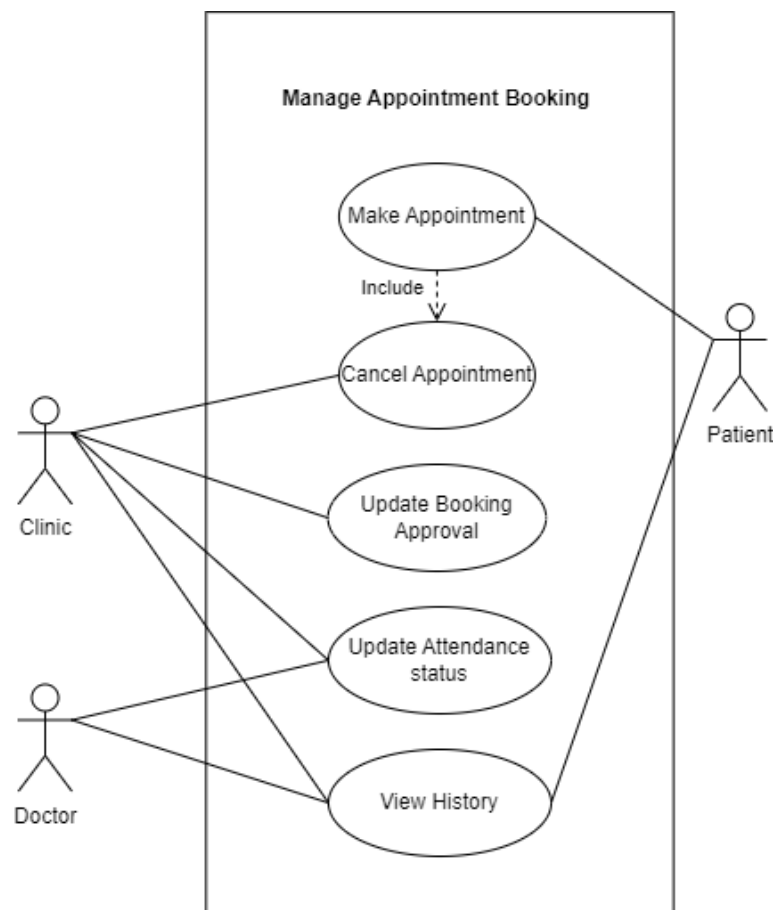


Figure 3.7 Use Case Diagram for Manage Appointment Booking

### 3.4.4.2 Manage Appointment Booking Sequence Diagram

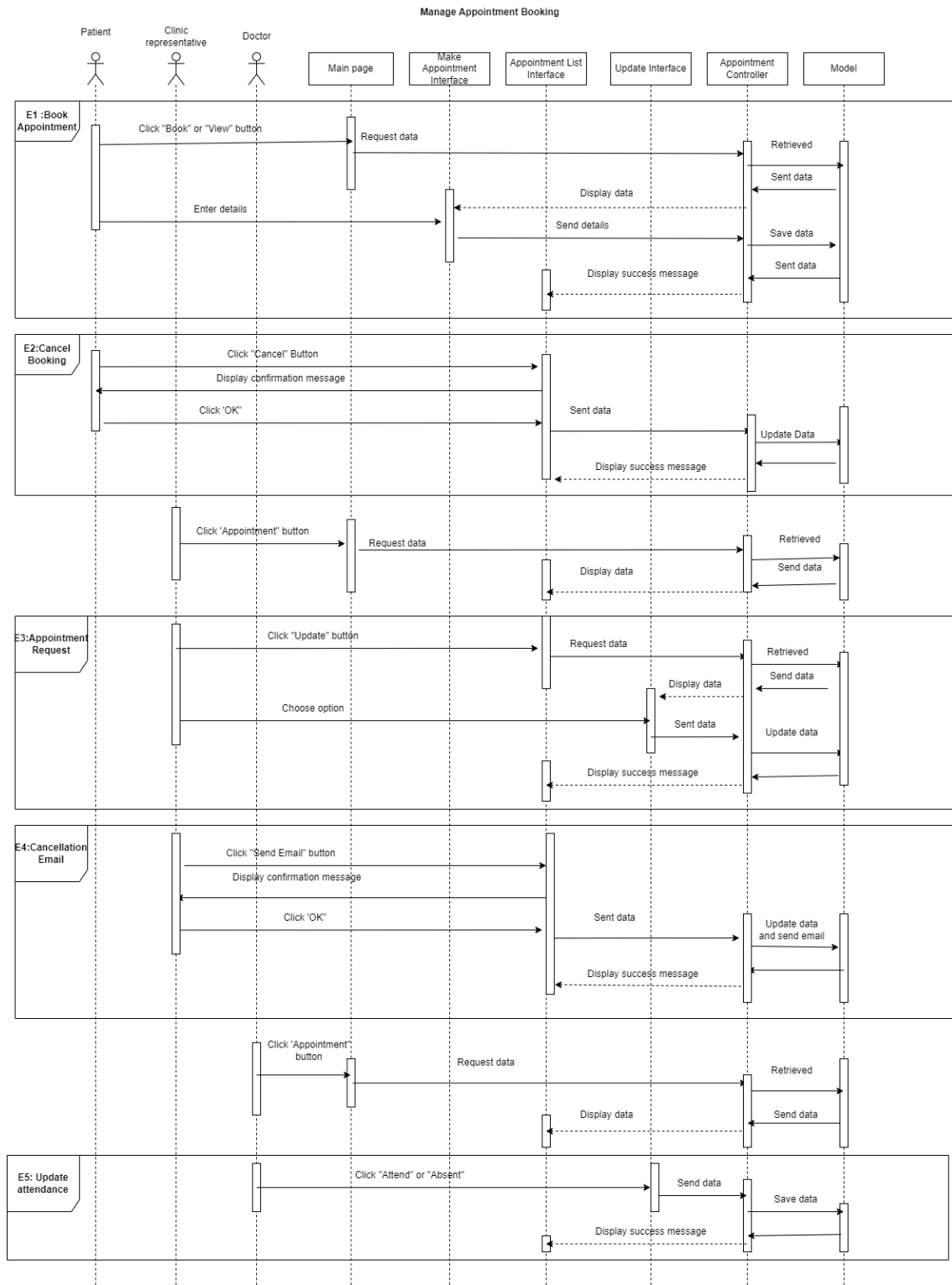


Figure 3.8 Sequence Diagram for Manage Appointment Booking

### 3.4.4.3 Manage Appointment Booking Activity Diagram

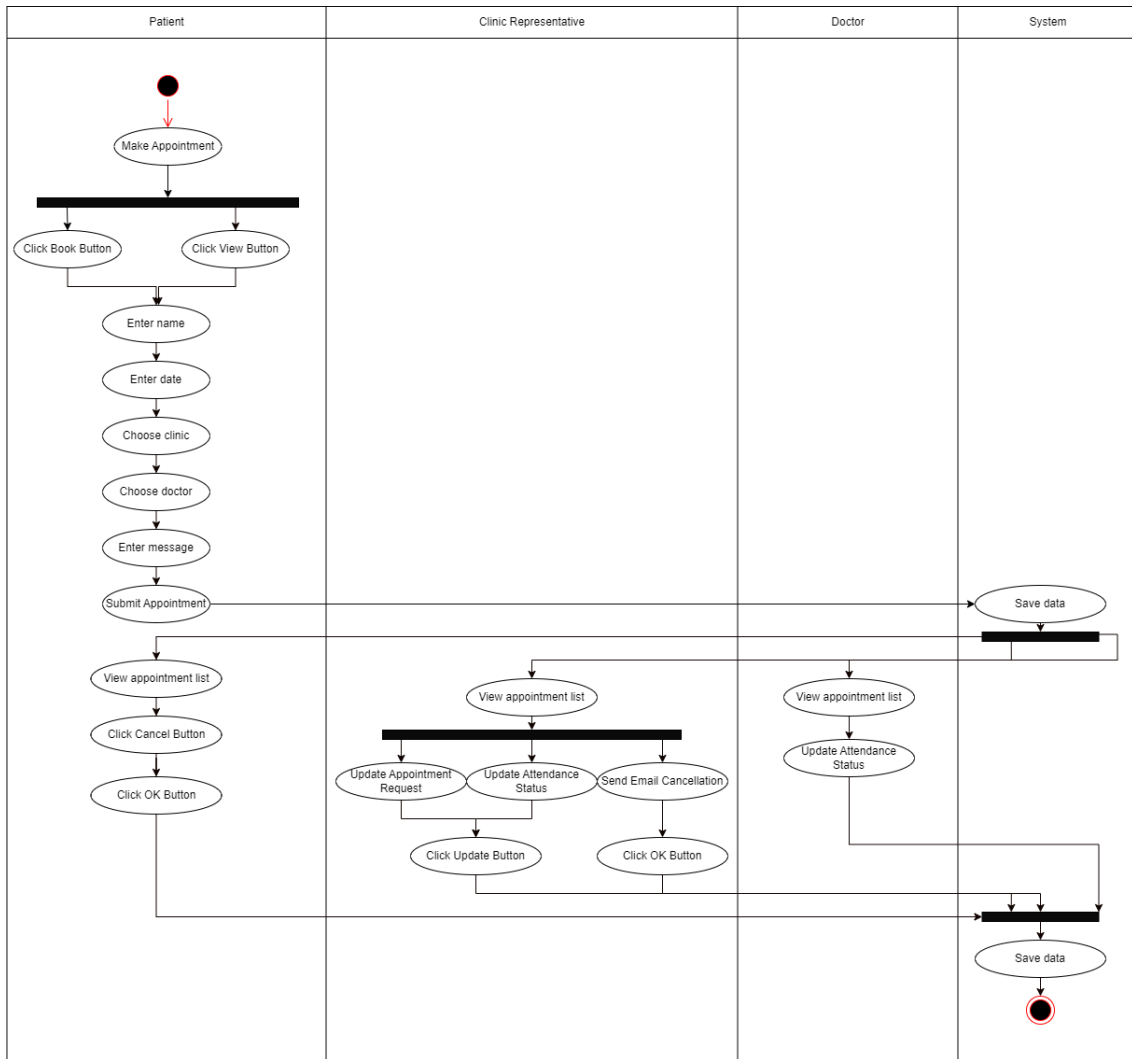


Figure 3.9 Activity Diagram for Manage Appointment Booking

### 3.4.5 Manage Medical Record

The patient, clinic staff and doctor have access to this module. The clinic staff can add medical record information from the appointment session between the patient and doctor. The doctor and patient can view the medical record. The clinic staff can also view and delete the medical record. The use case diagram, activity diagram, and sequence diagram that follow go over this module in greater depth:

#### 3.4.5.1 Manage Medical Record Use Case Diagram

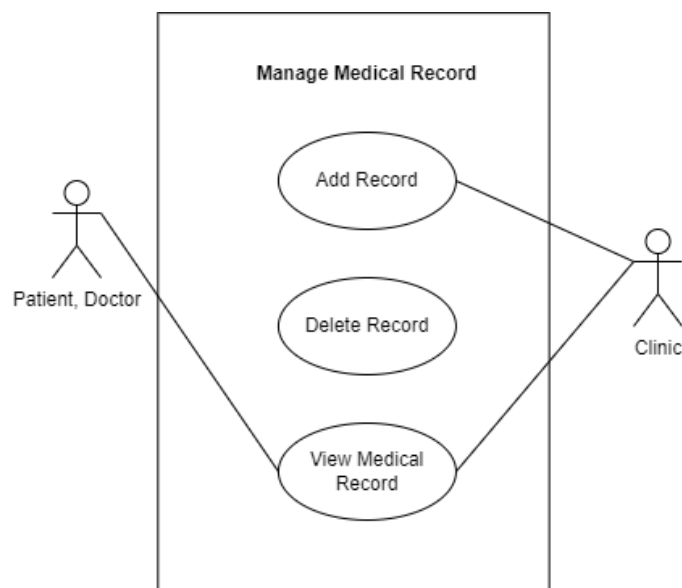


Figure 3.10 Use Case Diagram for Manage Medical Record

### 3.4.5.2 Manage Medical Record Sequence Diagram

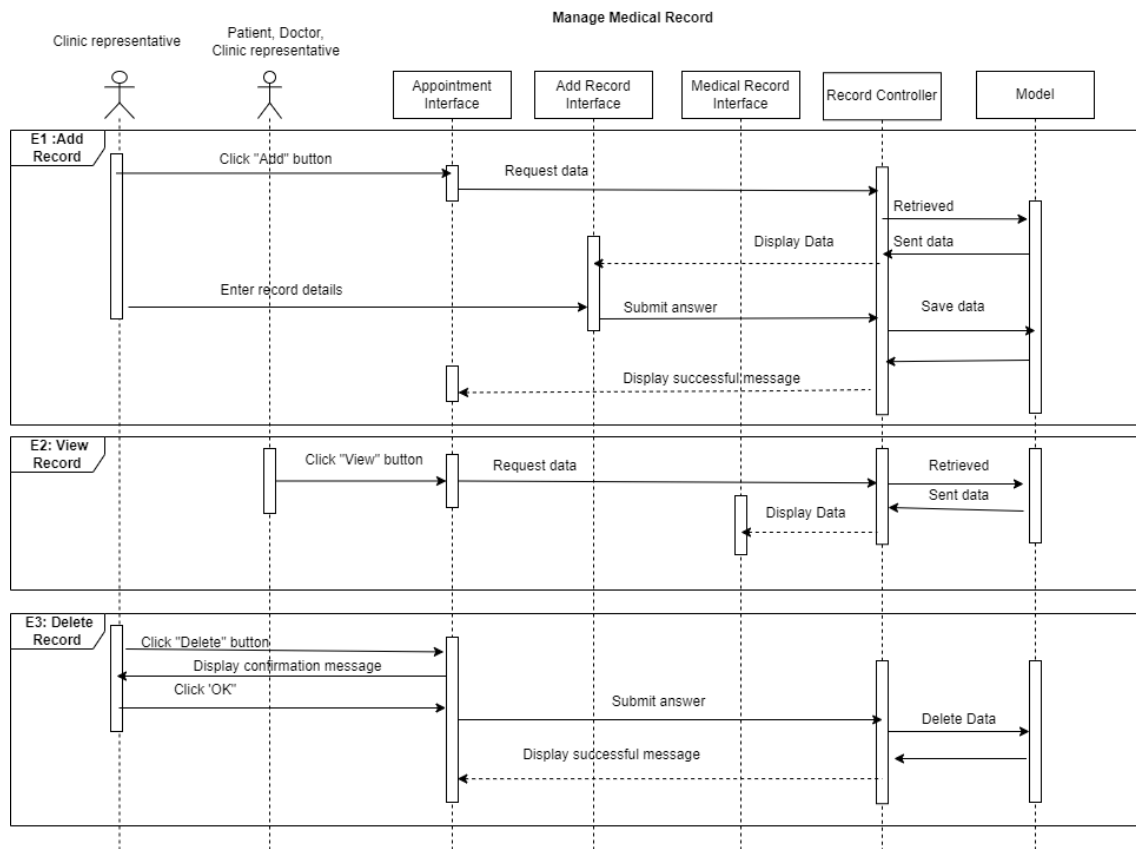


Figure 3.11 Sequence Diagram for Manage Medical Record



### 3.4.5.3 Manage Medical Record Activity Diagram

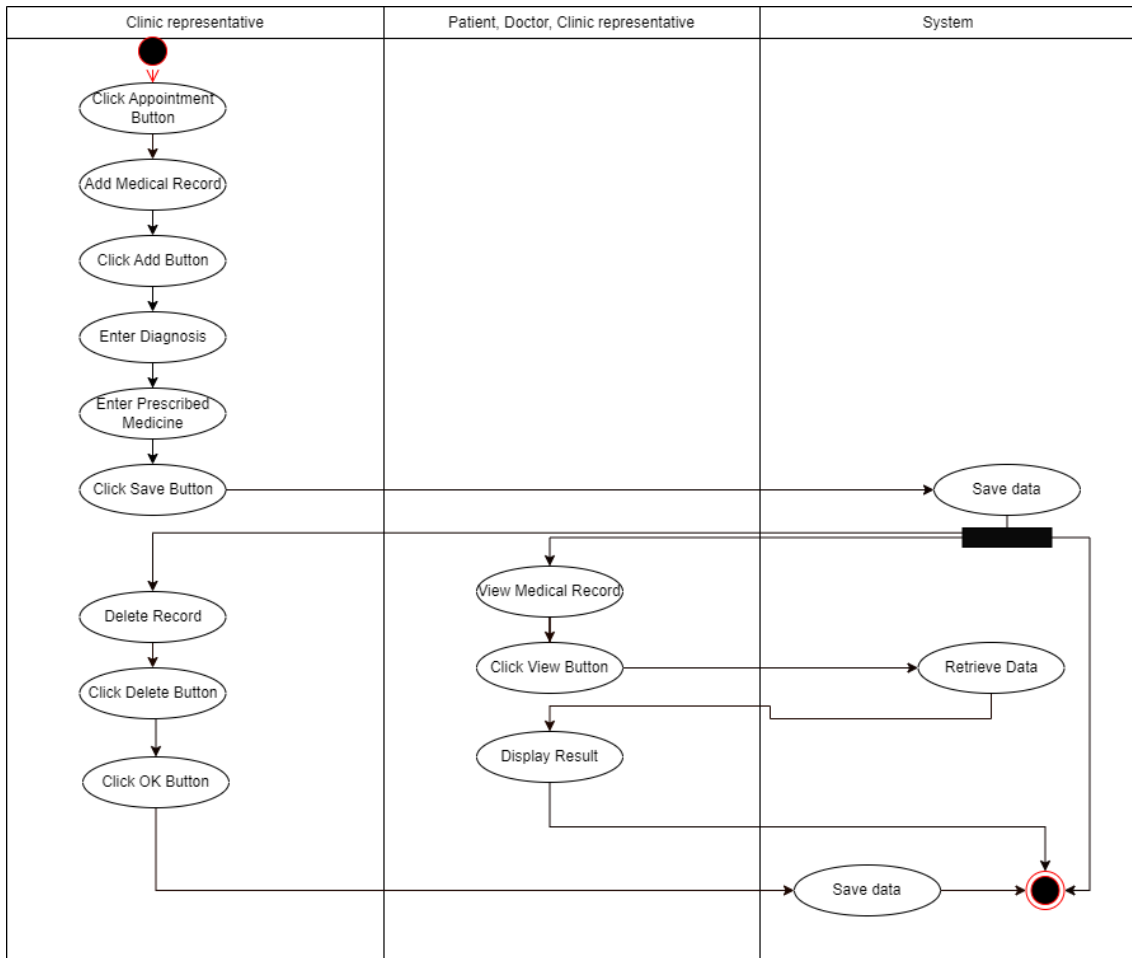


Figure 3.12 Activity Diagram for Manage Medical Record

### 3.4.6 Manage Medicine Information

Only the admin can manage medicine information section. The administrator can add new medicines, modify their details, and delete the medicine. All the users can search for the medicine at the main page of the system. This module is covered in greater detail in the use case diagram, activity diagram, and sequence diagram that follow:

#### 3.4.6.1 Manage Medicine Information Use Case Diagram

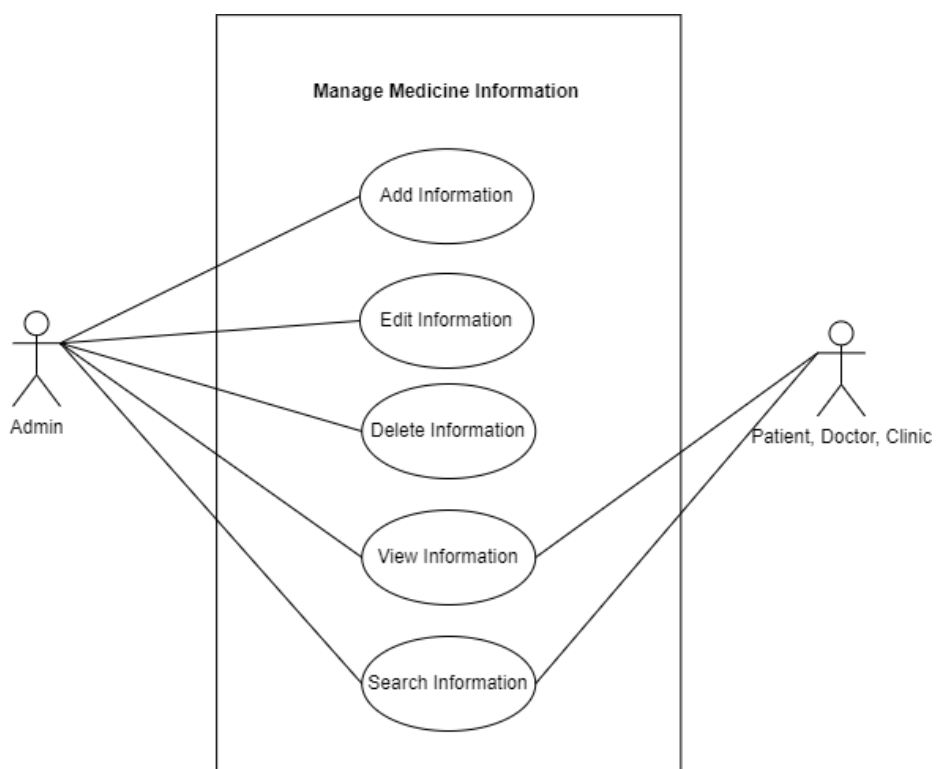


Figure 3.13 Use Case Diagram for Manage Medicine Information

### 3.4.6.2 Manage Medicine Information Sequence Diagram

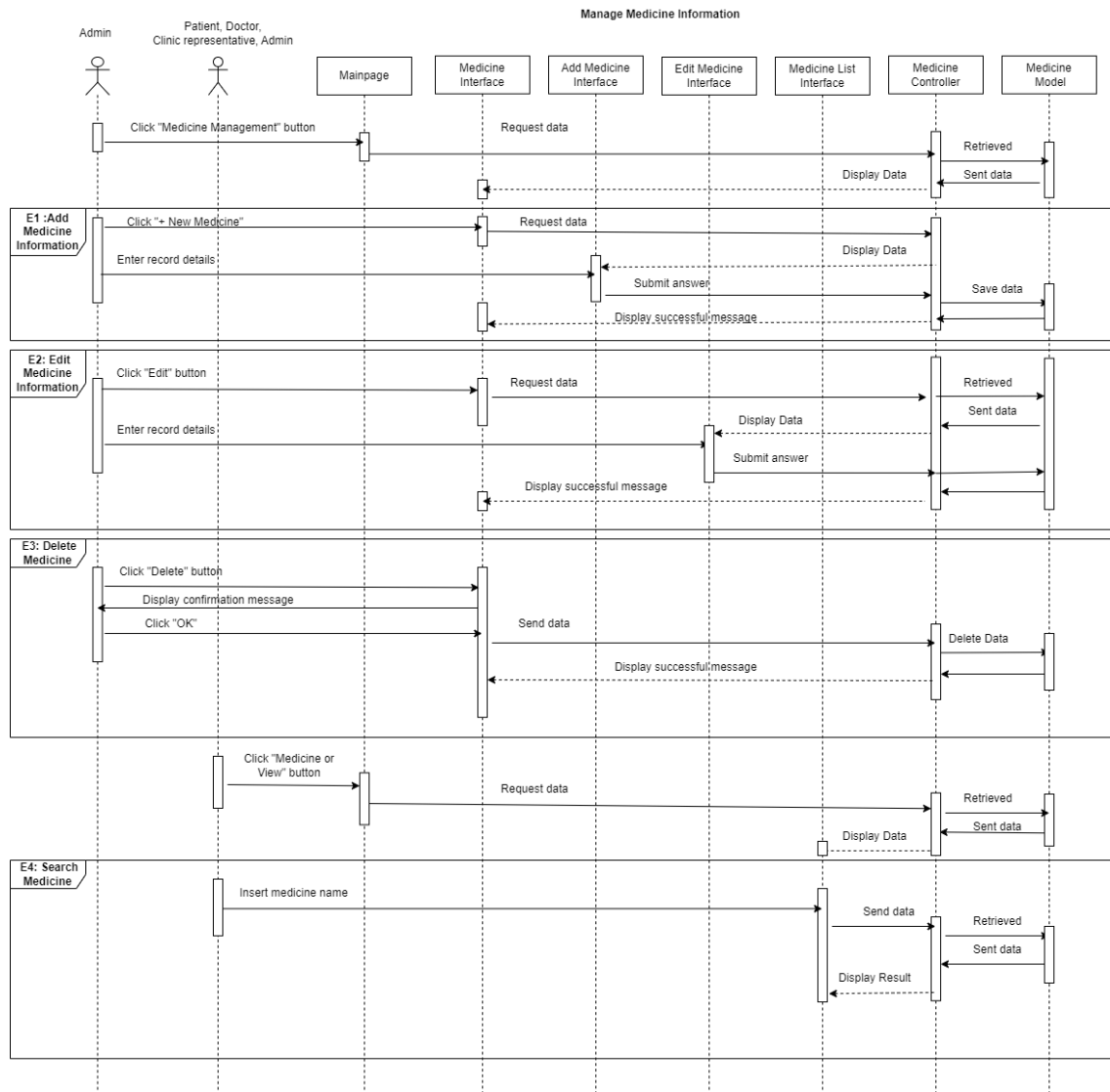


Figure 3.14 Sequence Diagram for Manage Medicine Information

### 3.4.6.3 Manage Medicine Information Activity Diagram

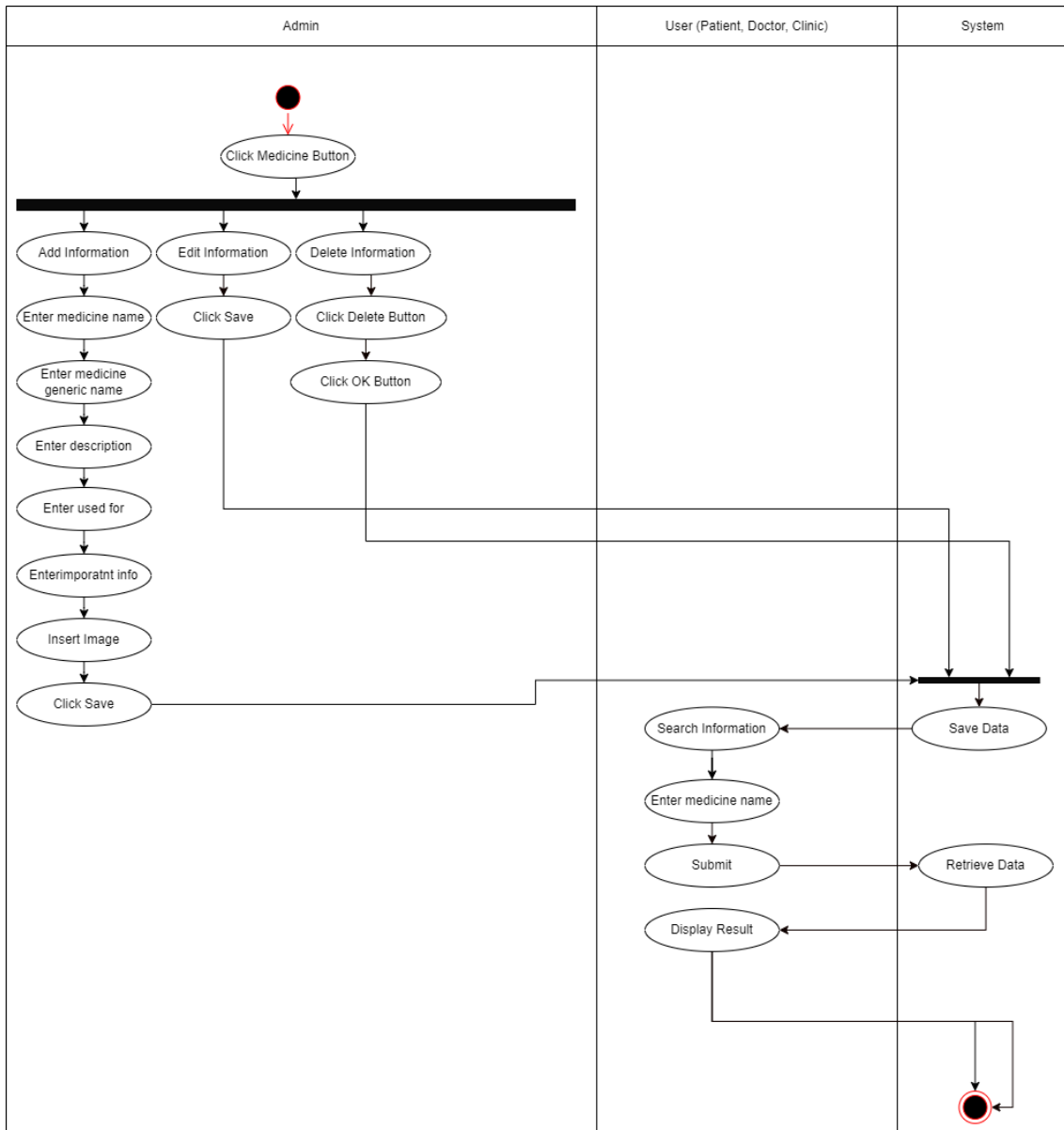


Figure 3.15 Activity Diagram for Manage Medicine Information

### 3.4.7 Manage Clinic

All users have access to the manage clinic module. Any clinic information in Pekan can be viewed by the patient. The clinic staff can also remove the doctor's account that works under them. The use case diagram, use case description, activity diagram, and sequence diagram that follow go over this module in greater depth:

#### 3.4.7.1 Use Case Diagram for Manage Clinic

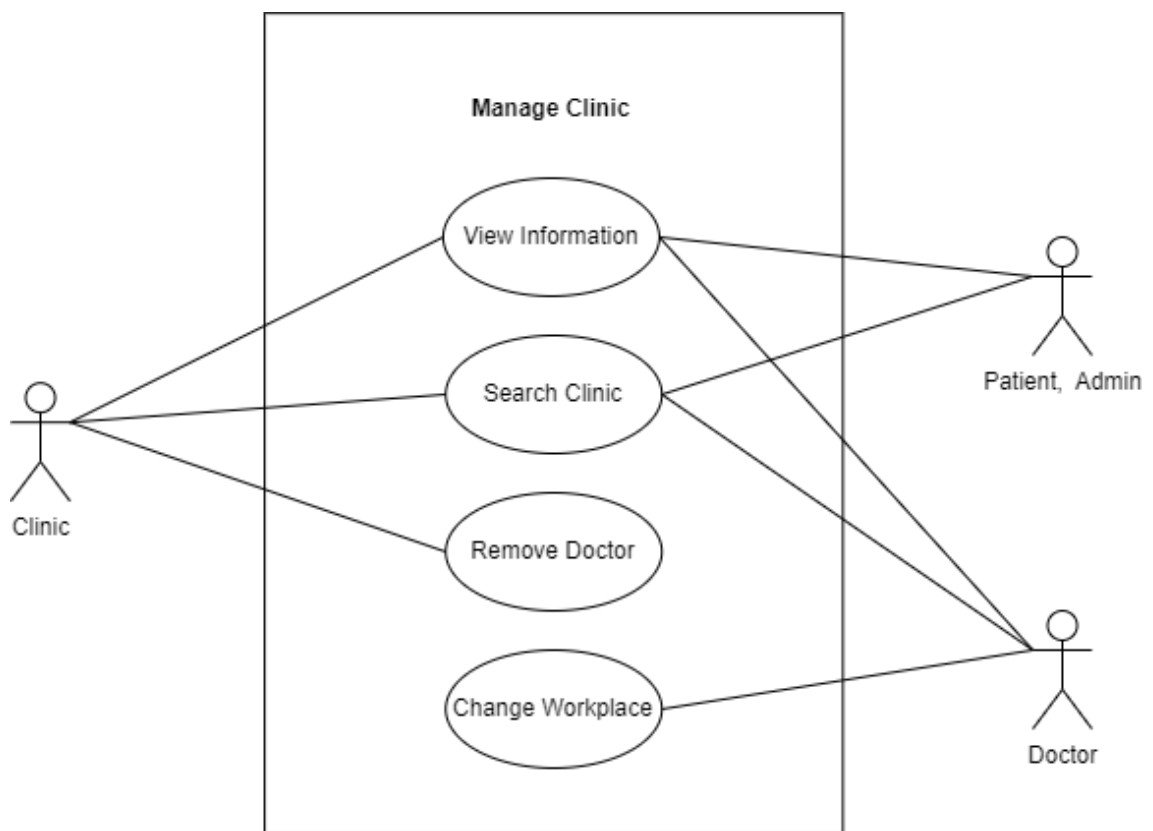


Figure 3.16 Use Case Diagram for Manage Clinic

### 3.4.7.2 Sequence Diagram for Manage Clinic

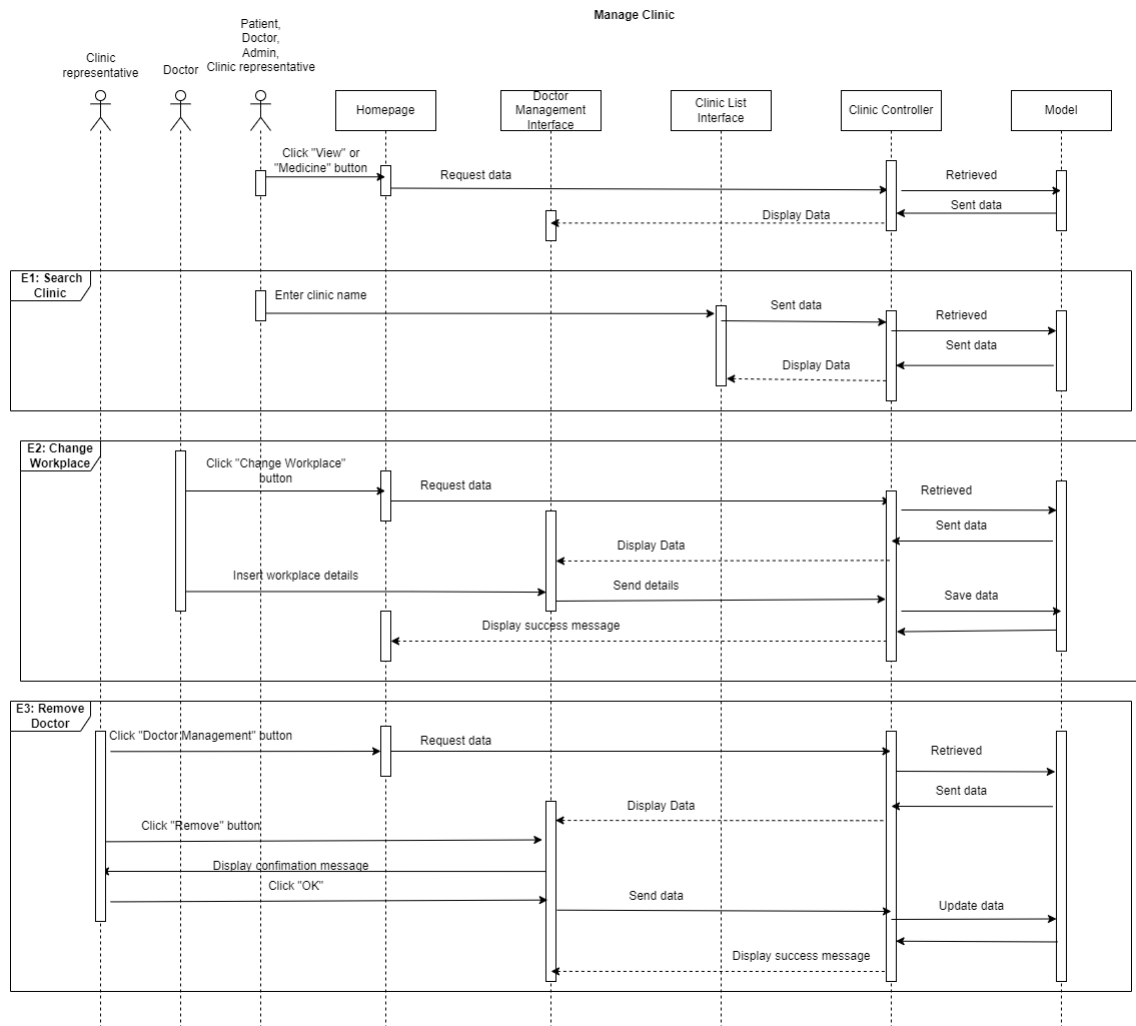


Figure 3.17 Sequence Diagram for Manage Clinic

### 3.4.7.3 Activity Diagram for Manage Clinic

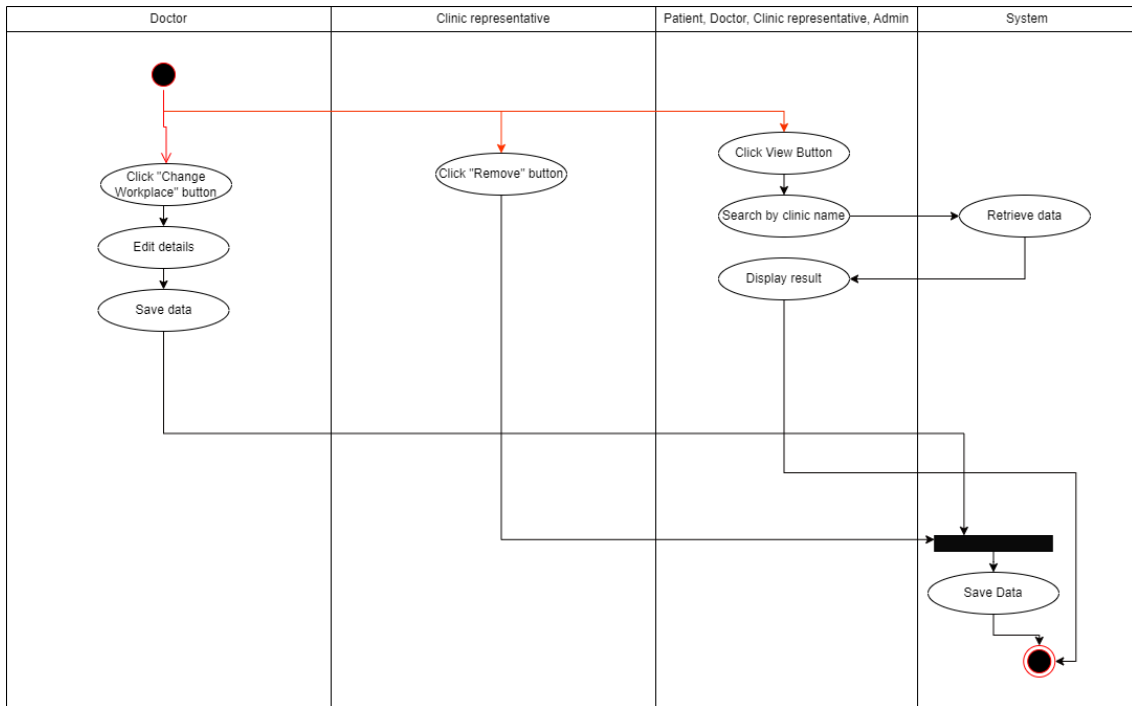


Figure 3.18 Activity Diagram for Manage Clinic

### 3.5 Data Design

In this section, an entity-relationship diagram (ERD) and data dictionary is prepared and are shown below as follows:

#### 3.5.1 ERD

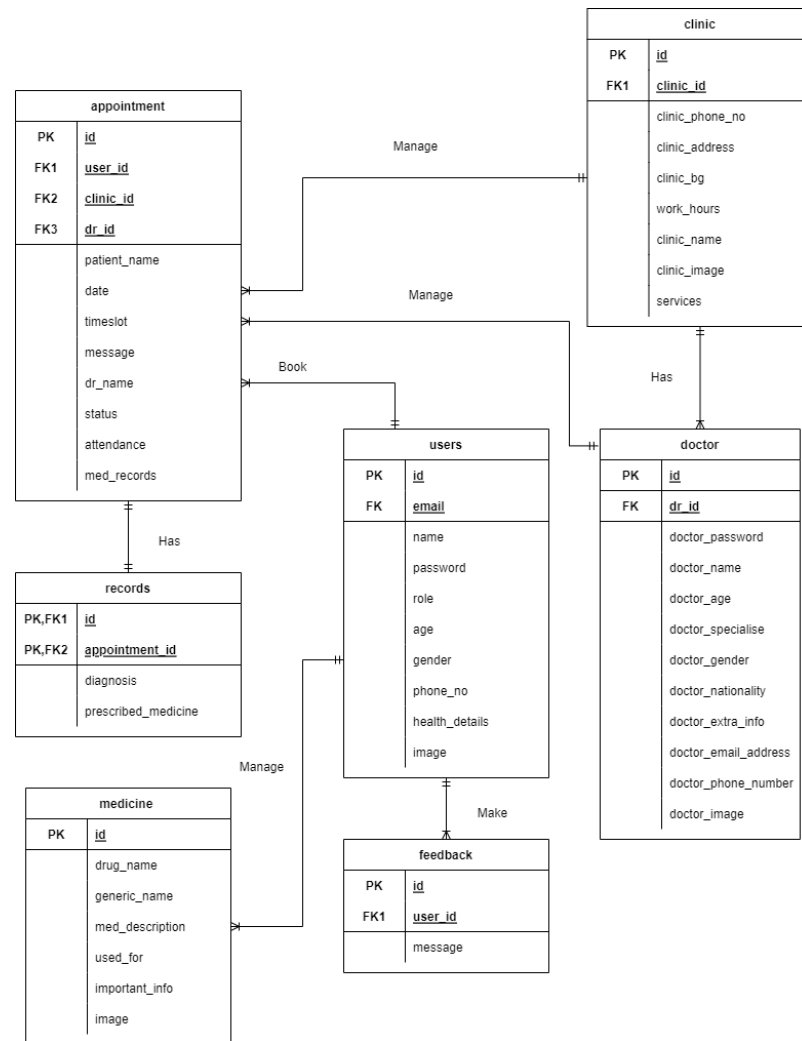


Figure 3.19 Entity Relationship Diagram (ERD)

The figure above is the entity-relationship diagram for doctor appointment booking and diabetes prediction web application. There are seven entities overall which are the 'users', 'clinic', 'doctor', 'feedback', 'appointment', 'records', and 'medicine'.



### 3.5.2 Database Dictionary

‘users’, ‘clinic’, ‘doctor’, ‘feedback’, ‘appointment’, ‘records’, and ‘medicine’, are the seven tables in this application. Each table's attribute is described in detail in the description section, including the data type, field length, and constraints.

#### 3.5.2.1 users

Table 3.4 Database Dictionary for ‘users’

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big integer	Primary Key	Login id for users
name	Varchar	Not null	Login name for users
email	Varchar	Unique key	Email of the users
password	Varchar	Not null	Password of the users
role	Varchar	Null	Role of each user
age	Int	Null	Age of patient
gender	Varchar	Null	Gender of patient
phone no	Varchar	Null	Phone number of the patient
health details	Varchar	Null	Patient's health details
image	Medium Text	Null	Patient's image

#### 3.5.2.2 clinic

Table 3.5 Database Dictionary for ‘clinic’

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for clinic
clinic id	Big Integer	Foreign Key	Login id for the clinic
clinic phone no	Varchar	Not null	Phone number of the clinic
clinic address	Varchar	Null	Address of the clinic
clinic bg	Varchar	Null	Clinic background
work_hours	Varchar	Null	Clinic working or operation hours
clinic name	Varchar	Null	Name of the clinic
clinic_image	Medium Text	Null	Image of the clinic
services	Varchar	NULL	Services that the clinic can offered

### 3.5.2.3 doctor

Table 3.6 Database Dictionary for 'doctor'

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for doctor
dr_id	Big Integer	Foreign Key	Login id for doctor
dr_name	Varchar	Not null	Name of the doctor
doctor_image	Medium Text	Not null	Image of the doctor
doctor_specialization	Varchar	Not null	The department that the doctor's expertise in
gender	Varchar	Not null	Gender of the doctor
doctor_edu_bg	Varchar	Not null	Doctor education background
doctor_years_working	Varchar	Not null	Years of working of the doctor
doctor_phone_no	Varchar	Not null	Phone number of the doctor
clinic_id	Big Integer	Not null	Workplace of the doctor

### 3.5.2.4 feedback

Table 3.7 Database Dictionary for 'patient'

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for feedback
user_id	Big Integer	Not null	Id of the person giving feedback
message	Varchar	Not null	Feedback message

### 3.5.2.5 appointment

Table 3.8 Database Dictionary for 'appointment'

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for the appointment
user_id	Big Integer	Foreign Key	Id of the patient booking the appointment
patient_name	Varchar	Not null	Name of the patient
date	Date	Not null	Appointment date
timeslot	Varchar	Not null	Appointment timeslot

message	Varchar	Null	Appointment message from patient
clinic id	Big Intger	Not null	Id of the booked clinic
dr id	Big Integer	Null	Id of the booked doctor
dr_name	Varchar	Null	Name of the booked doctor
status	Varchar	Null	Status for appointment (Accepted, Rejected)
attendance	Varchar	Null	Status for appointment attendance (Attend, Absent)
med_records	Varchar	Null	Medical record status (Exist, Null)

### 3.5.2.6 records

Table 3.9 Database Dictionary for 'records'

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id formedical record
appointment_id	Big Integer	Foreign Key	Id of the appointment
diagnosis	Varchar	Not null	Patient's diagnosis
prescribed_medicine	Varchar	Not null	Patient's prescribed medicine

### 3.5.2.7 medicine

Table 3.10 Database Dictionary for 'medicine'

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for medicine
drug_name	Varchar	Not null	Drug name of the medicine
generic_name	Varchar	Not null	Generic name of the medicine
med_description	Long Text	Not null	Description of the medicine
used_for	Long Text	Not null	Usage of the medicine
important_info	Long Text	Not null	Important info of the medicine
image	Medium Text	Null	Image of the medicine

### **3.6 Proof of Initial Concept**

These prototype interfaces are developed using appropriate font size, font style, font colour, memorable wordings, appropriate background, and button colour. The interfaces are also designed with the same font size, font style, font colour, memorable wordings, appropriate background colour, and button colour for all interfaces to ensure that the interface is easy to remember and learn even without assistance. Because of the repetitive elements implemented and the flow of interfaces for each function in the prototype, the interfaces designed for this prototype are also user-friendly. The appropriate use of colour for the design is also beneficial because the colour chosen is not distracting and will not hurt the user's eyes.

The interface navigation is also useful, allowing users to access other interfaces in the system easily. The icons used in the interfaces are also simple to understand and are widely used in other existing applications or systems. As a result, users are already familiar with the icon used in the interface of this module, such as the home icon. In this interface, the home icon in this prototype is to refer to the user's homepage. The language used in the interface is also understandable. This allows users to easily use the system and helps them achieve their goals for using the system, such as admin adding new medicine.

### 3.6.1 All users can access

#### 3.6.1.1 Main page

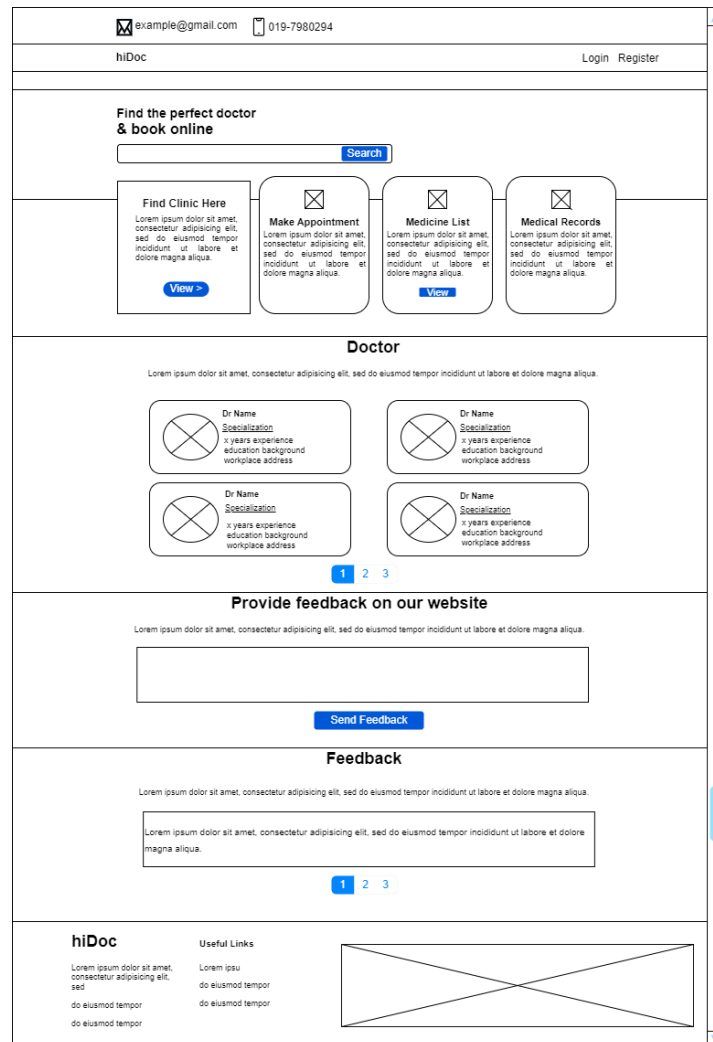


Figure 3.20 Main Page Interface

This is hiDoc's main page. Users can use this interface to search for doctors, view clinics that have registered with the system, add and view existing feedback, and view medicine lists. However, for patients who log in to the system, a button to book an appointment, such as "Book," will be available. Figure 3.21 illustrates this. Because only the patient can book an appointment for this system, the button is hidden for the other user type. The patient has the option of viewing a list of doctors who are available in each clinic.

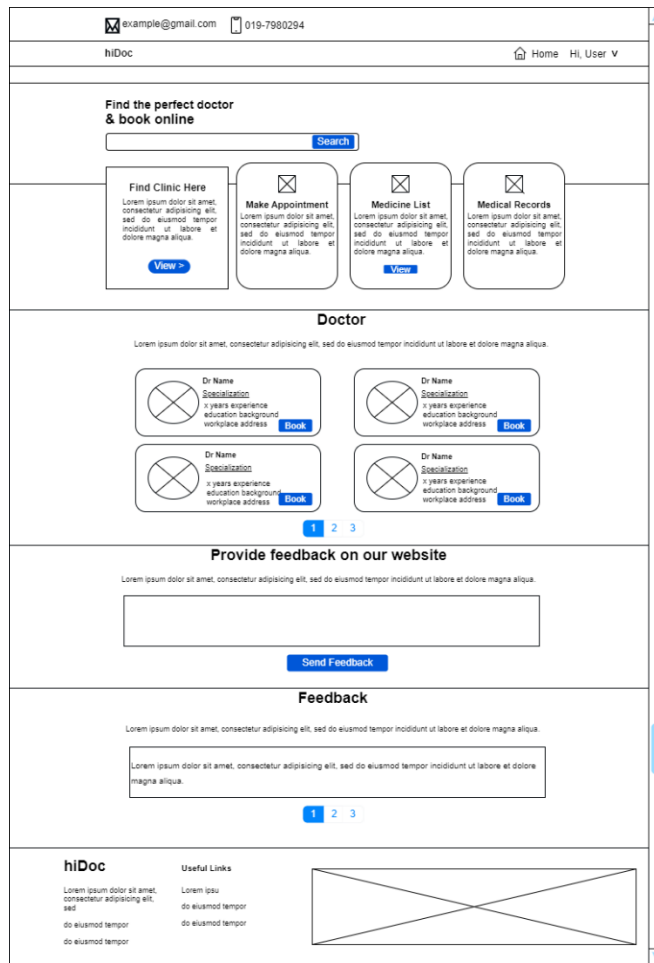
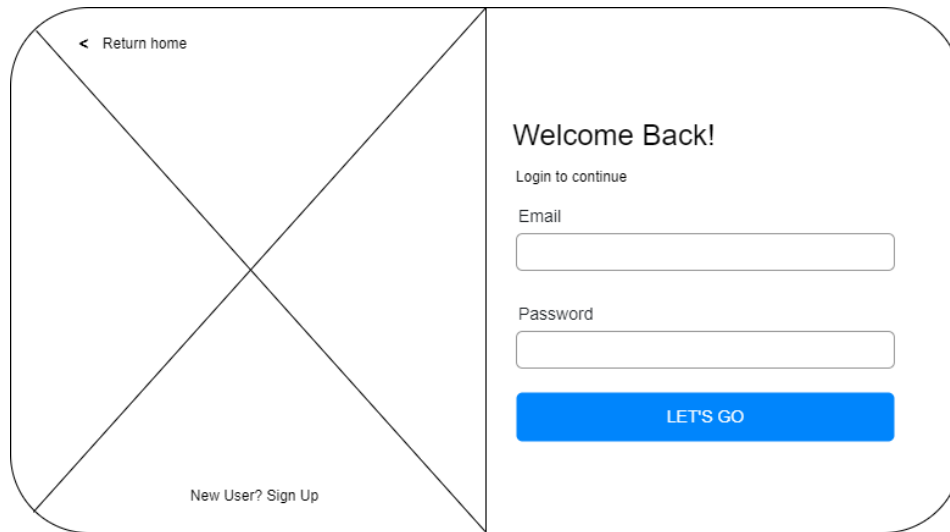


Figure 3.21 Main Page - When patient or user login Interface

### 3.6.1.2 Login

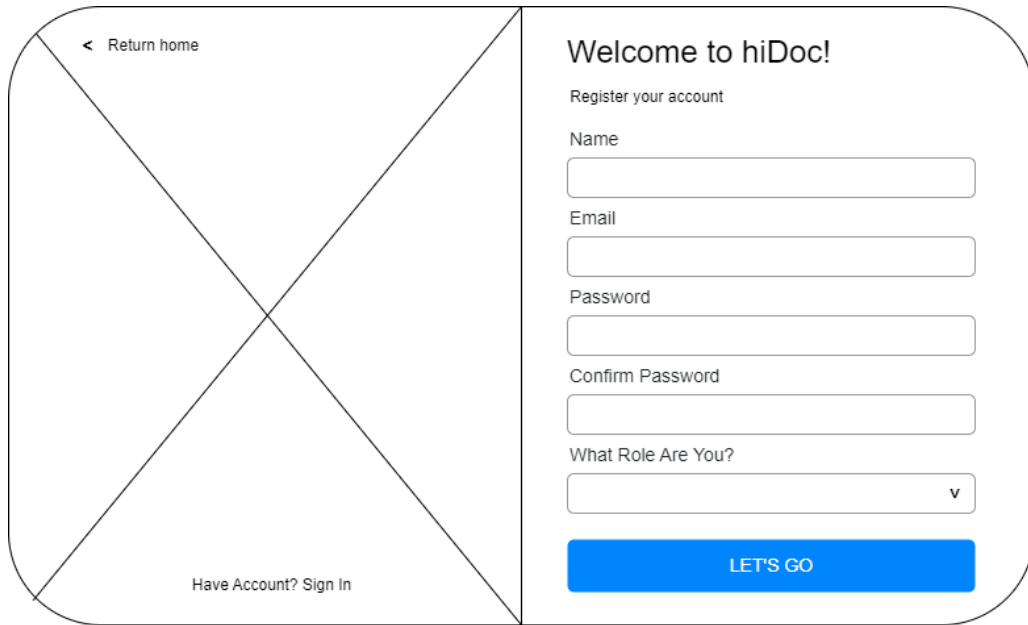


The image shows a wireframe of a login interface. It is a rounded rectangle divided into two main sections by a vertical line. The left section is mostly empty, with a back arrow and the text "< Return home" in the top-left corner and "New User? Sign Up" in the bottom-right corner. The right section contains the login form. It starts with the heading "Welcome Back!" followed by the sub-heading "Login to continue". Below this are two input fields: one for "Email" and one for "Password". At the bottom of the right section is a blue button with the text "LET'S GO".

Figure 3.22 Login Interface

This is the system's login interface. Users must enter their email address and password to login.

### 3.6.1.3 Register



The image shows a mobile application registration screen. On the left, there is a large grey area with a diagonal 'X' and a 'Return home' link at the top. On the right, the registration form is titled 'Welcome to hiDoc!' and includes fields for Name, Email, Password, Confirm Password, and a dropdown for 'What Role Are You?'. A blue 'LET'S GO' button is at the bottom right, and a 'Sign In' link is at the bottom left.

Figure 3.23 Register Interface

This is the system's registration interface. Users must enter their name, email address, password and role to register. The users also need to confirm their password.



### 3.6.1.4 Manage Clinic Module (Clinic List)

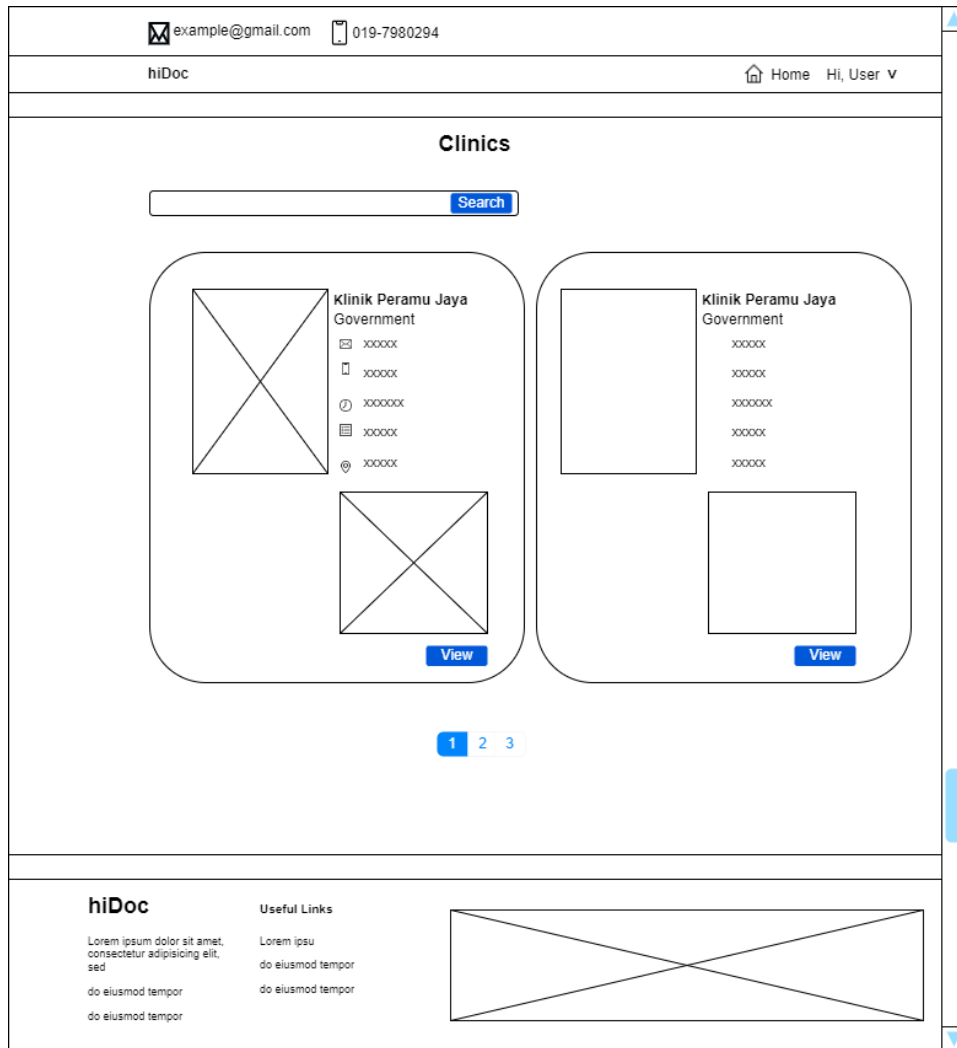


Figure 3.24 List of clinics Interface

This is the clinic list interface, where all clinic information is displayed. Users must click the "View >" button on the main page under "Find Clinic Here" to get to this page.

### 3.6.1.5 Manage Medicine Module (Medicine List)

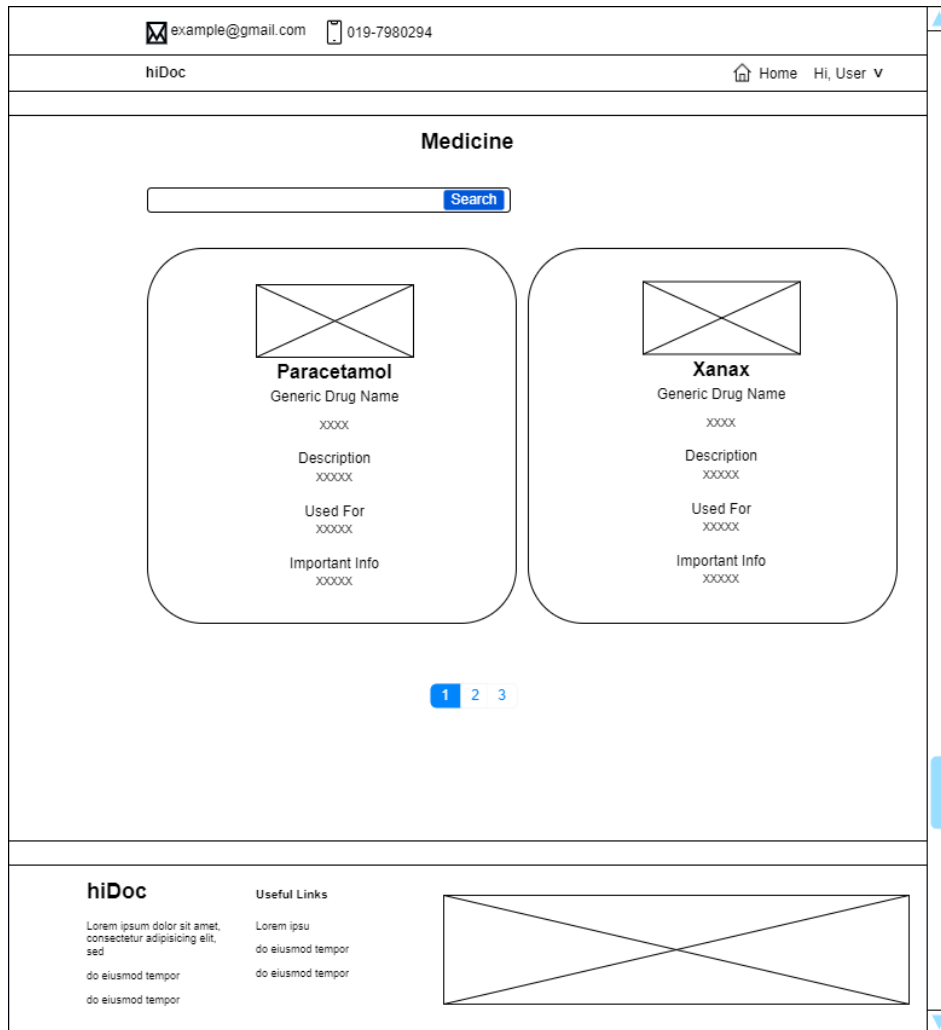


Figure 3.25 List of medicines Interface

This is the medicine list interface, where all medicine information is displayed. Users must click the "View " button on the main page under "Medicine List" to get to this page.

## 3.6.2 Admin Interface

### 3.6.2.1 Homepage

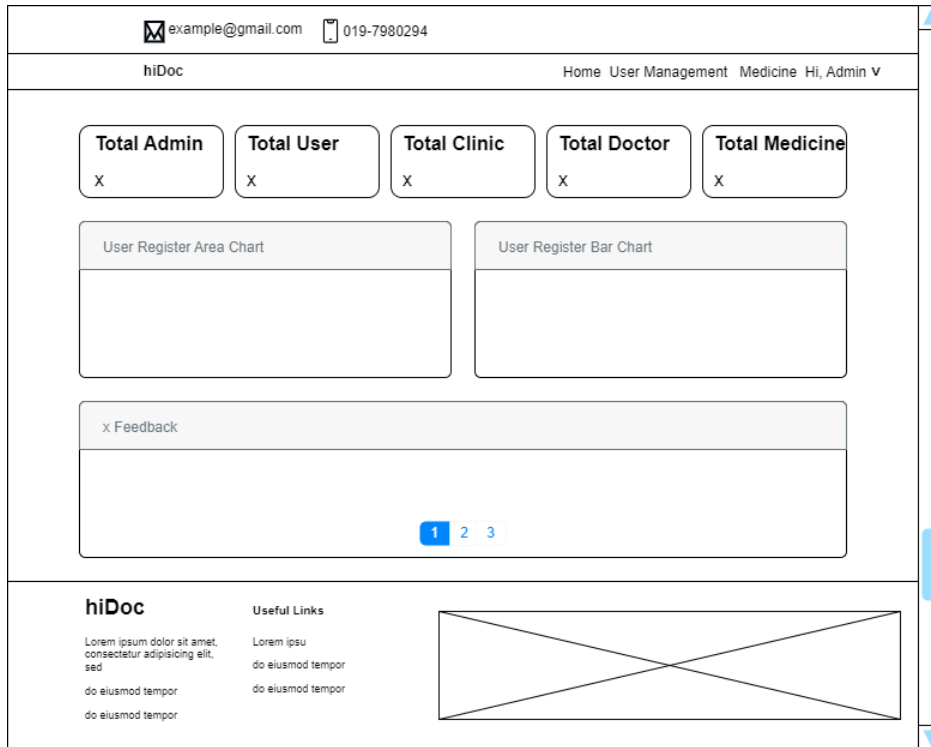


Figure 3.26 Admin Homepage Interface

This is the admin homepage. This interface displays the total number of administrators, users or patients, clinics, and doctors registered in this system. There is also a total count of the medications available in the system. The user feedback is also displayed on the admin homepage for him or her to see. There are also an area chart and bar chart that display the total count of user registration by month. There are also area and bar charts that display user registration data.

### 3.6.2.2 Manage Account Module (User Management)

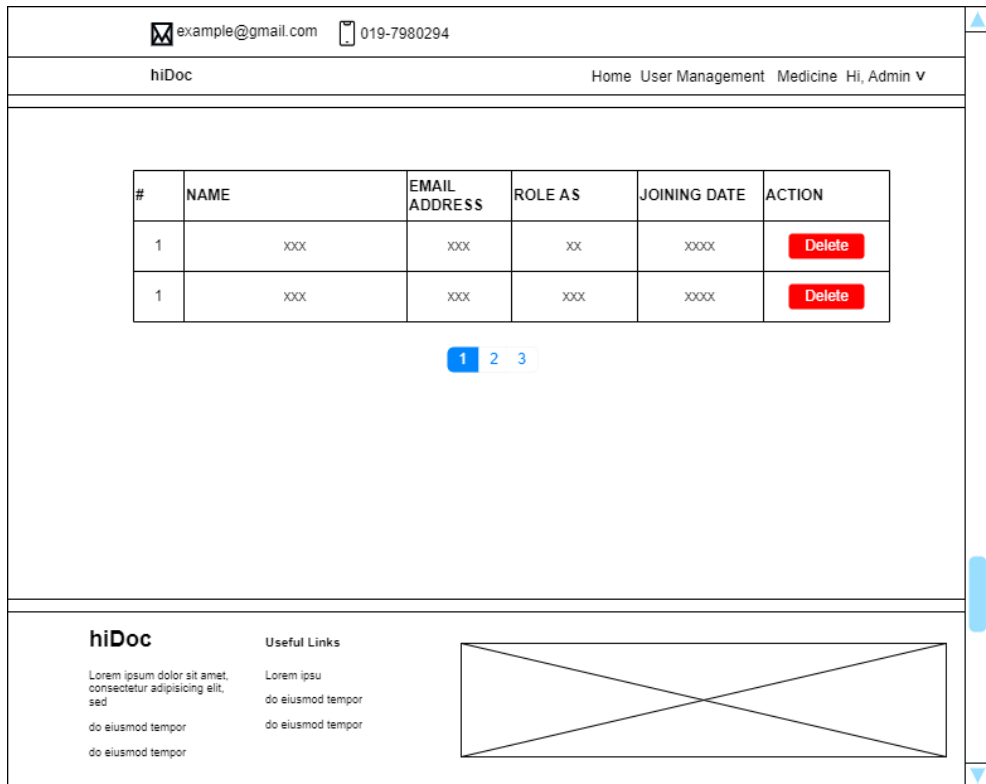


Figure 3.27 User Management Interface

This is the user management page, where the administrator can delete any account, he wants.

### 3.6.2.3 Manage Medicine Module (Medicine Management)

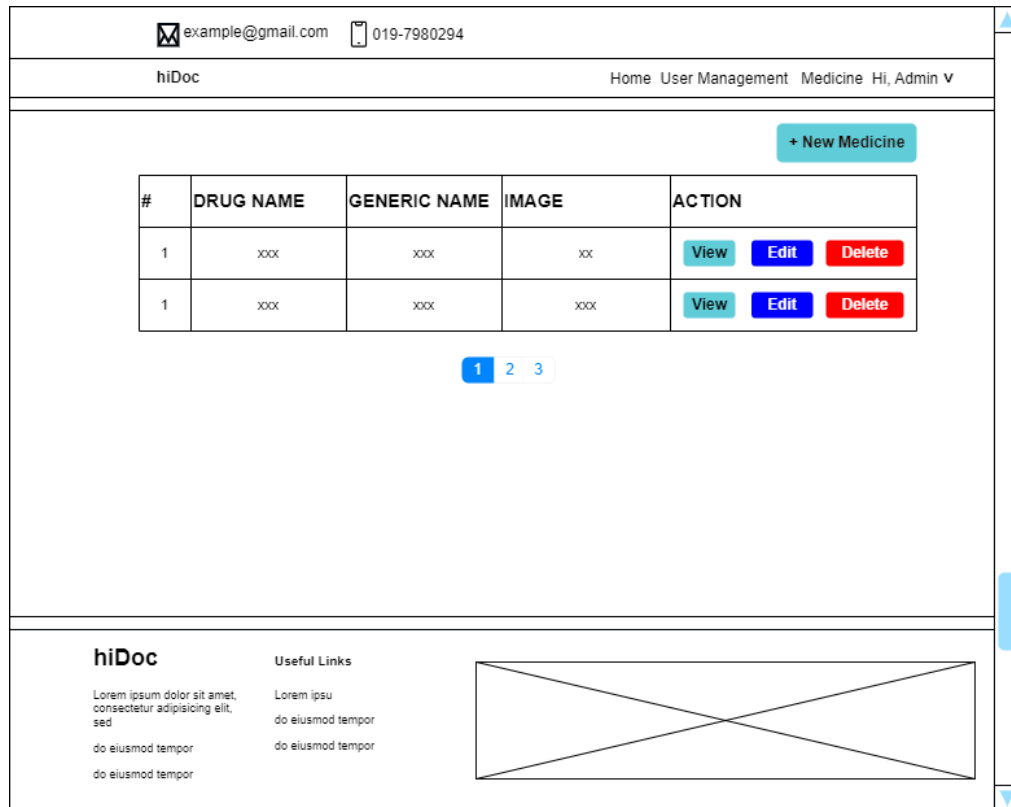


Figure 3.28 Medicine Management Interface

This is the medicine administration interface. By clicking the "view" button, the administrator can view the medication information. By clicking the "+New Medicine" button, the administrator can also add new medicine information. By clicking the "Edit" and "Delete" buttons, the administrator can also edit and delete the medicine.

### 3.6.3 Clinic Interface

#### 3.6.3.1 Get Started

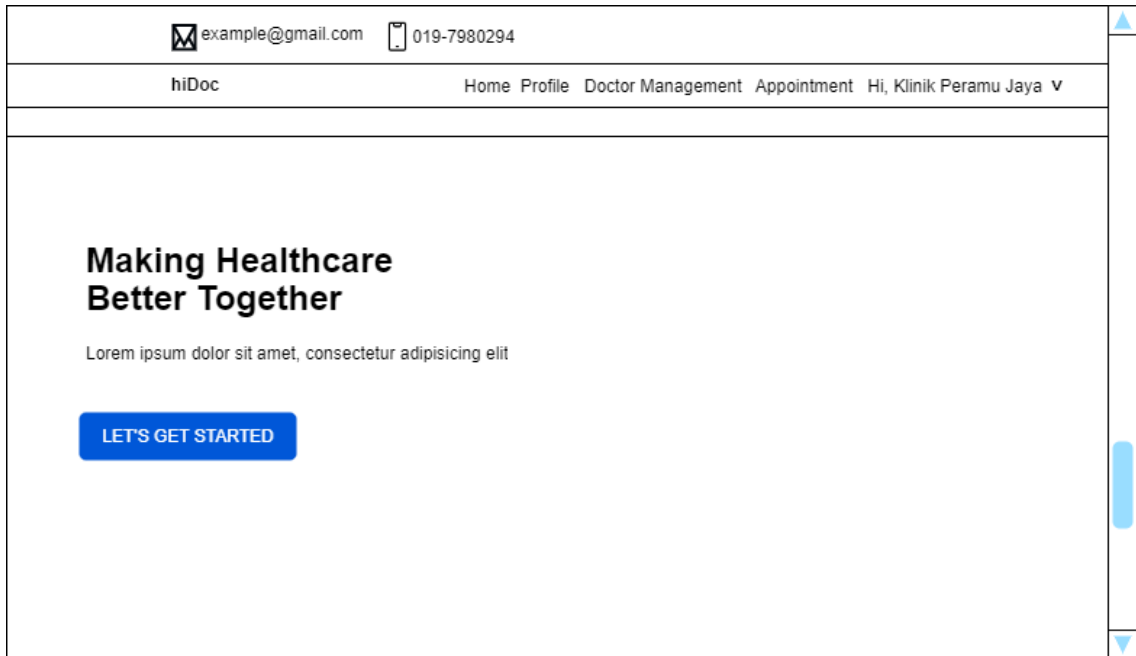


Figure 3.29 Get Started Interface

This is the get started page. This page will appear after a clinic staff member or representative has just registered in the system. The clinic staff will be directed to a page where they must enter the clinic's information after clicking the button. This data will be displayed for the users to see.

### 3.6.3.2 Homepage

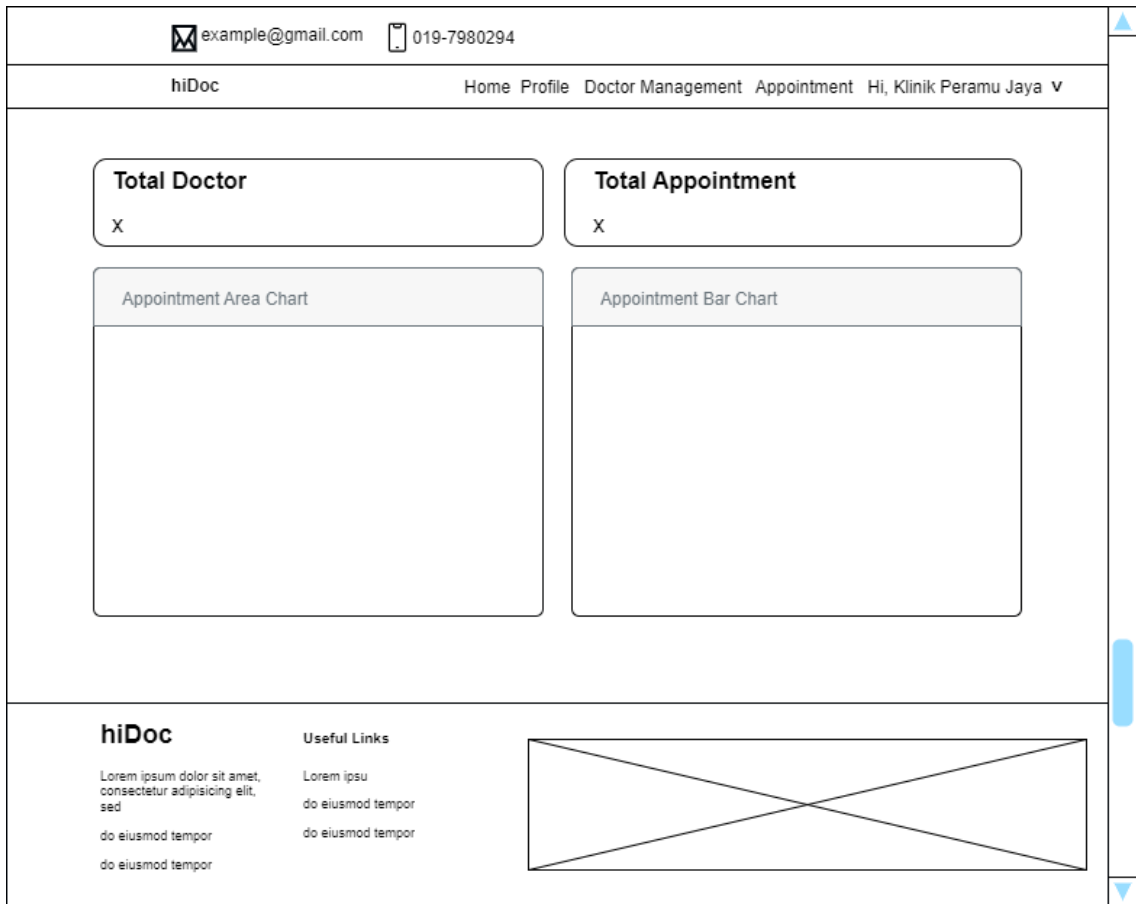


Figure 3.30 Clinic Homepage

This is the clinic's homepage. This interface displays the clinic's total number of doctors as well as the total number of appointments. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.6.3.3 Manage Account Module (Doctor Management)

example@gmail.com 019-7980294

hiDoc Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya

## Doctor Management

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	IMAGE	NAME	GENDER	PHONE NUMBER	SPECIALIZE	YEARS OF WORKING	EDU BACKGROUND	WORKPLACE	ACTION
1	xxx	xxx	xx	xxxx	xxxx	xxxx	xxxx	xxxx	Remove
1	xxx	xxx	xxx	xxxx	xxxx	xxxx	xxxx	xxxx	Remove

1 2 3

**hiDoc**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor do eiusmod tempor do eiusmod tempor

**Useful Links**

Lorem ipsu do eiusmod tempor do eiusmod tempor

Figure 3.31 Doctor Management Interface

This is the interface for doctor management. The clinic representative or staff can delete the doctor's account from this interface. This interface also displays information about the doctor.



### 3.6.3.4 Manage Account Module (Clinic Profile)

The screenshot displays the 'hiDoc' user interface for a clinic profile. At the top, the user's email 'example@gmail.com' and phone number '019-7980294' are shown. The navigation bar includes 'hiDoc', 'Home', 'Profile', 'Doctor Management', 'Appointment', and 'Hi, Klinik Peramu Jaya'. The main content area is titled 'Hi, Klinik Peramu Jaya' and contains a placeholder for a profile picture. Below this is the 'CLINIC INFORMATION' section, which includes a 'Clinic Photo' field with a 'Choose File' button and 'No file chosen' text. There are input fields for 'Clinic Name', 'Email Address', and 'Address'. The 'ABOUT CLINIC' section contains fields for 'Government/Private Clinic', 'Services available', 'Operating Hours', and 'Contact Number'. A blue 'Update Profile' button is located at the bottom of this section. The footer area includes the 'hiDoc' logo, a 'Useful Links' section with placeholder text, and a large rectangular area with a diagonal cross, likely a placeholder for an advertisement or banner.

Figure 3.32 Profile Interface

This is the profile interface for the clinic. The clinic's information can be edited here by clicking the "Update Profile" button.

### 3.6.3.5 Manage Appointment Booking Module (Appointment List & History)

✉ example@gmail.com
☎ 019-7980294

hiDoc
Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya V

#### Appointment List

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	PATIENT NAME	CONTACT NO	DATE	TIME SLOT	MESSAGE	DOCTOR SELECTED	SPECIALIZE	STATUS	ATTENDANCE	ACTION	CANCELLATION	MEDICAL RECORDS	
1	XXX	XXX	xxx	xxx	xx	xxxx	xxxx	Attend	Attend	Update	Delete	Send Email	XXXX
1	XXX	XXX	xxx	xxx	xxx	xxxx	xxxx	Pending	Attend	Update	Delete	Send Email	XXXX

1
2
3

**hiDoc**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

**Useful Links**

Lorem ipsum  
do eiusmod tempor  
do eiusmod tempor

Figure 3.33 Appointment Interface

This is the clinic's appointment list interface. This interface displays the appointment list created by users for the clinic. The clinic staff can update the status of whether the appointment request was approved or rejected, as well as whether the patient attended the appointment or not. The appointment data can also be deleted by the staff. The appointment can also be cancelled by clicking the "Send Email" button. After pressing the button, an email will be sent to the patient who scheduled the appointment.

### 3.6.3.6 Manage Medical Record Module (Add, View, Delete)

✉ example@gmail.com
☎ 019-7980294

hiDoc
Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya V

#### Appointment List

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	PATIENT NAME	CONTACT NO	DATE	TIME SLOT	MESSAGE	DOCTOR SELECTED	SPECIALIZE	STATUS	ATTEN DANCE	ACTION	CANCELLATION	MEDICAL RECORDS
1	XXX	XXX	xxx	xxx	xx	xxxx	xxxx	xxxx	<span style="background-color: green; color: white; padding: 2px 5px;">Attend</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Update</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>	<span style="background-color: orange; color: white; padding: 2px 5px;">Send Email</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">View</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>
1	XXX	XXX	xxx	xxx	xxx	xxxx	xxxx	xxxx	<span style="background-color: green; color: white; padding: 2px 5px;">Attend</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Update</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>	<span style="background-color: orange; color: white; padding: 2px 5px;">Send Email</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Add</span>

1
2
3

**hiDoc**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

**Useful Links**

Lorem ipsu  
do eiusmod tempor  
do eiusmod tempor

Figure 3.34 Medical Record Interface

This is the interface for the appointment list. This interface is linked to the medical record management module. When the attendance status is "Attend," the staff will see a "Add" button under the "MEDICAL RECORDS" column. The staff can enter the doctor's diagnosis for the user. Following the addition of data, the "Add" button will be replaced by "View" and "Delete" buttons. The "view" button allows staff to view the patient's medical record.

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxx
Diagnosis	xxxxx
Prescribed Medicine	xxxxx

\_\_\_\_\_

Authorised Signature

\_\_\_\_\_

Name

\_\_\_\_\_

Date

*Generated by,*

**hiDoc**

Figure 3.35 Patient Medical Record

This is the patient's medical record. It contains information about the appointment as well as the doctor's diagnosis.

### 3.6.4 Doctor Interface

#### 3.6.4.1 Get Started

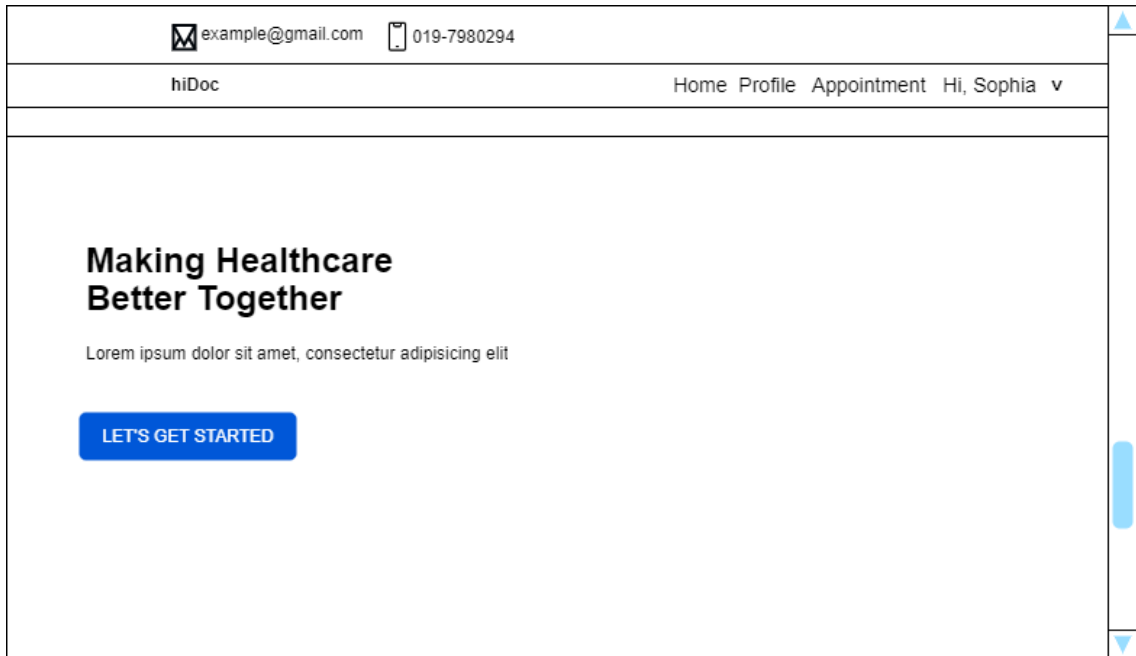


Figure 3.36 Get Started Interface

This is the get started page. This page will appear after the doctor registered in the system. The doctor will be directed to a page where they must enter their information after clicking the button. This data will be displayed for the users to see.

### 3.6.4.2 Homepage

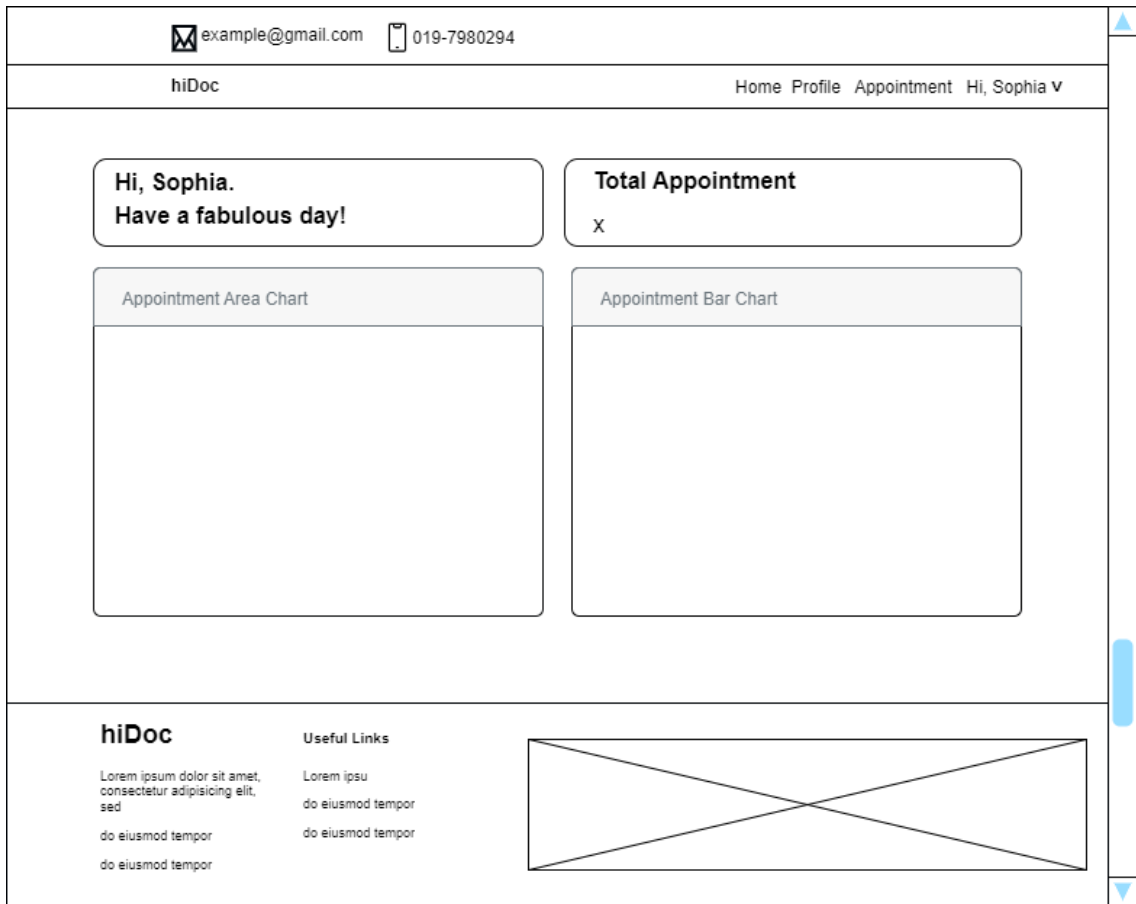


Figure 3.37 Doctor Homepage

This is the doctor's homepage. This interface displays the total number of appointments of the doctor. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.6.4.3 Manage Account Module (Doctor Profile)

example@gmail.com 019-7980294

hiDoc Home Profile Appointment Hi, Sophia v

## Hi, Sophia

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

USER INFORMATION

Profile Photo

Choose File No file chosen

Name

Gender Email Address

Phone Number Workplace

MEDICAL BACKGROUND

Specialization

Working Experience (Years)

ABOUT ME

Education Background

Update Profile

hiDoc

Useful Links

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor do eiusmod tempor do eiusmod tempor

Lorem ipsu do eiusmod tempor do eiusmod tempor

Figure 3.38 Profile Interface

This is the profile interface for the doctor. The doctor's information can be edited here by clicking the "Update Profile" button.

### 3.6.4.4 Manage Appointment Booking Module (Appointment List & History)

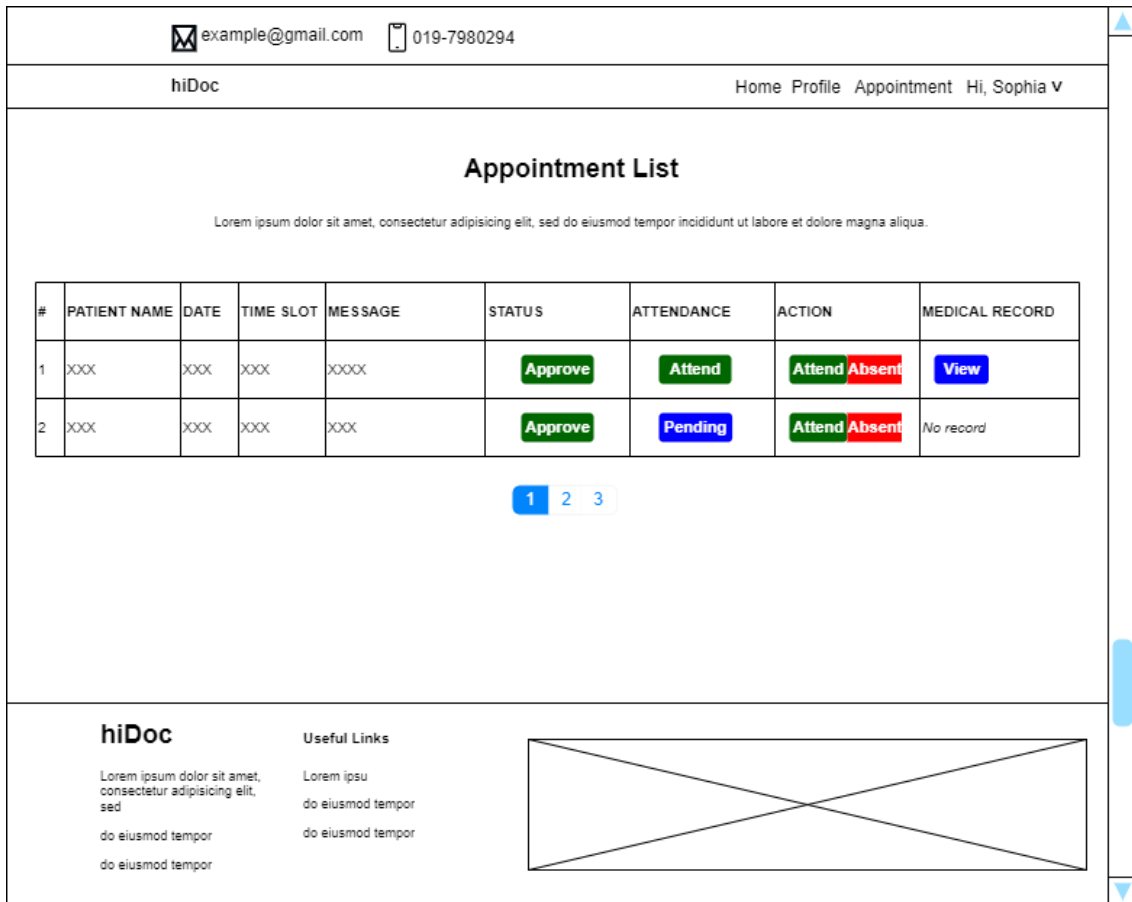


Figure 3.39 Appointment Interface

This is the interface for the doctor's appointment list. This interface displays the appointment list that users have created for the doctor. By clicking the "Attend" or "Absent" button, the doctor can update the patient's attendance for the appointment session. This interface is linked to the module manage medical records. When the clinic staff enters the medical record into the system, a "View" button appears in the interface, allowing the doctor to view the patient's medical record for that appointment session. This medical record is depicted in Figure 3.40 below.



### 3.6.4.5 Manage Medical Record Module (View)

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxx
Diagnosis	xxxxx
Prescribed Medicine	xxxxx

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

**hiDoc**

Figure 3.40 Patient Medical Record Interface

### 3.6.5 User Interface

#### 3.6.5.1 Homepage

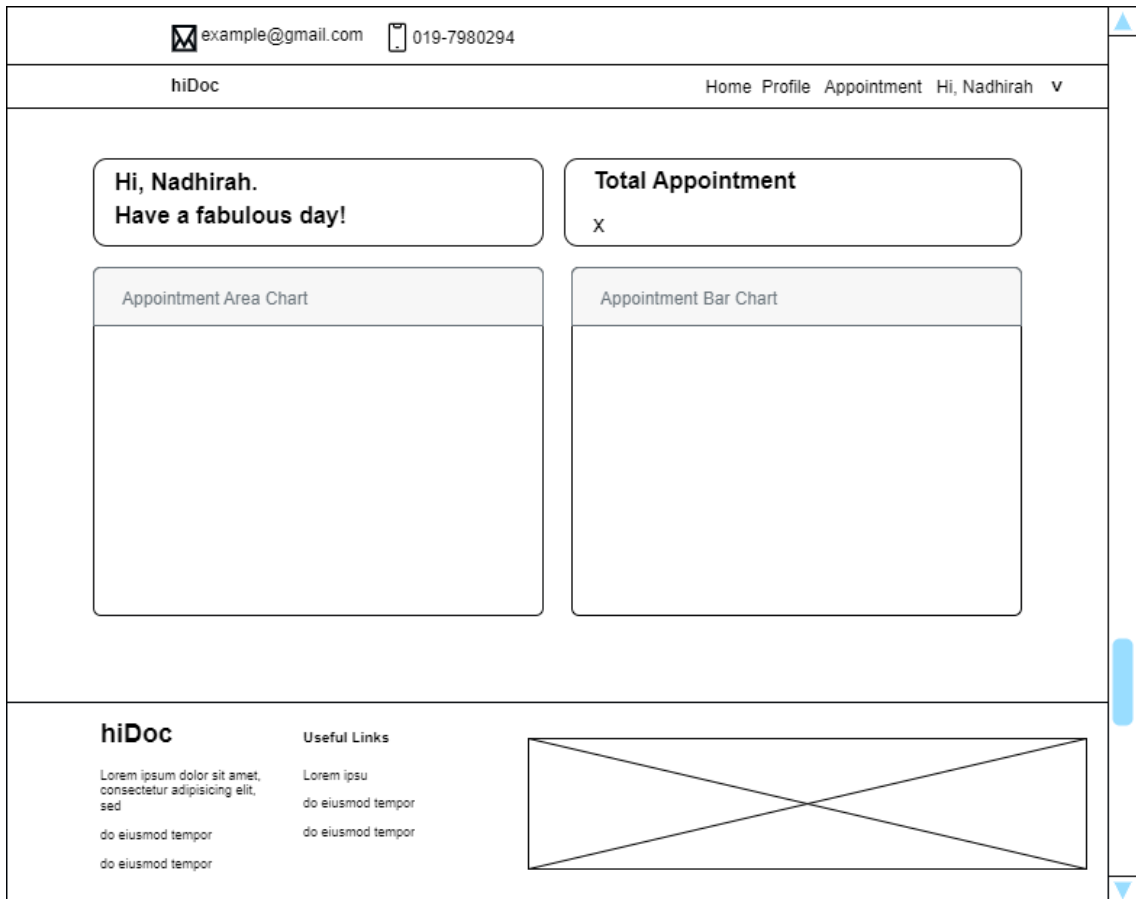
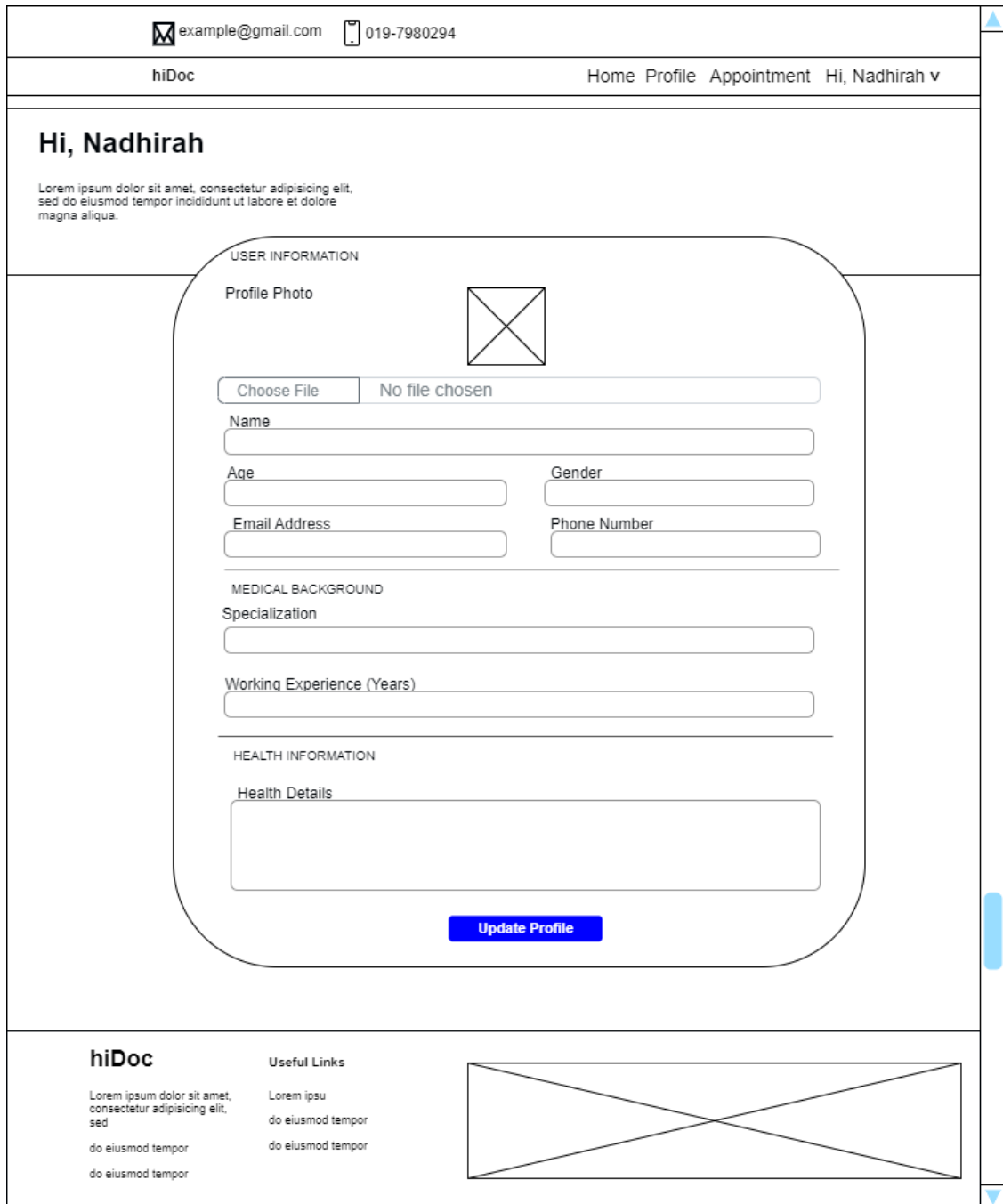


Figure 3.41 User Homepage

This is the patient's homepage. This interface displays the total number of appointments of the patient. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.6.5.2 Manage Account Module (User Profile)



The image shows a user profile interface for a patient named Nadhirah. At the top, there is a header with the email 'example@gmail.com' and phone number '019-7980294'. Below this is a navigation bar with 'hiDoc' and links for 'Home', 'Profile', 'Appointment', and 'Hi, Nadhirah v'. The main content area is titled 'Hi, Nadhirah' and includes a placeholder for a profile photo. The 'USER INFORMATION' section contains a 'Profile Photo' area with a 'Choose File' button and a 'No file chosen' message. Below this are input fields for 'Name', 'Age', 'Gender', 'Email Address', and 'Phone Number'. The 'MEDICAL BACKGROUND' section has fields for 'Specialization' and 'Working Experience (Years)'. The 'HEALTH INFORMATION' section has a 'Health Details' field. A blue 'Update Profile' button is located at the bottom of the profile form. The footer contains the 'hiDoc' logo, a 'Useful Links' section with placeholder text, and a large placeholder image.

Figure 3.42 Profile Interface

This is the profile interface for the patient. The patient's information can be edited here by clicking the "Update Profile" button.

### 3.6.5.3 Manage Appointment Booking Module (Make Appointment & Appointment History)

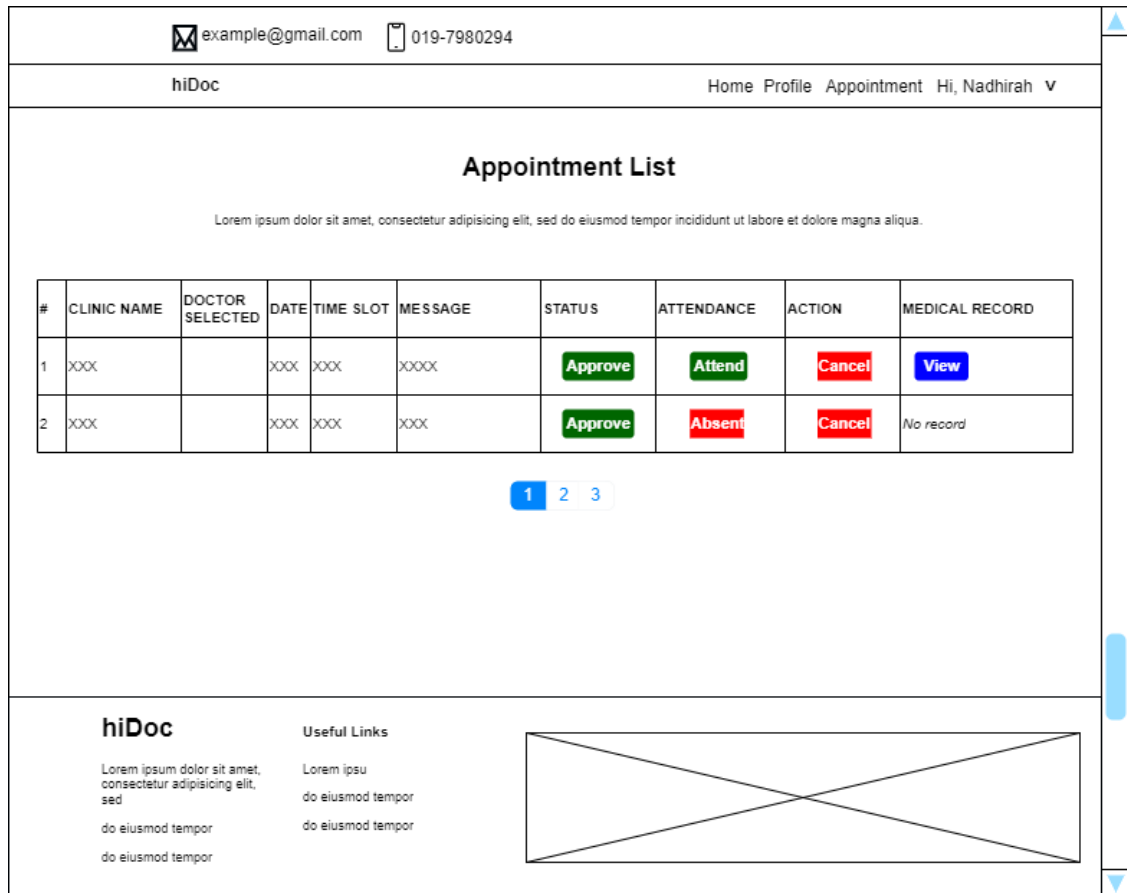


Figure 3.43 Appointment Interface

This is the appointment list interface for the patient. This interface displays the patient's appointment list. The patient can see whether their appointment request has been approved or rejected. The patient can also view their medical records by clicking the "view" button in the "MEDICAL RECORD" column. The medical record can be seen at Figure 3.44 below. However, the button will be visible only if the patient attends the appointment. By clicking the "Cancel" button, you can also cancel the appointment.

### 3.6.5.4 Manage Medical Record Module (View)

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxx
Diagnosis	xxxxx
Prescribed Medicine	xxxxx

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

**hiDoc**

Figure 3.44 Patient's Medical Record Interface

### 3.7 Testing Plan

Table 3.11 User Acceptance Test

No	Module	Activities	Status (Pass/Fail)		Comments
1	Login	User registration	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
2		User login	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
3		User logout	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
4		Get Started (clinic and doctor only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
5	Profile (except admin)	Add information	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
6		Edit profile	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
7		View own profile	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
8		View doctor profile (for clinic only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
9		View patient profile (for clinic and doctor only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
10		Delete any account (for admin only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
11	Appointment Booking	Make appointment (for patient only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
13		View doctor list (for patient only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
14		Update appointment request status (for clinic only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
15		Cancel appointment (for	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	

		patient and clinic only)			
16		View appointment history	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
17	Medical Record (except admin)	Add diagnosis (for clinic only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
18		Delete record (for clinic only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
19		View record (for patient, doctor clinic only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
20	Medicine Information	Add new medicine (For admin only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
21		Edit medicine information (For admin only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
22		Delete medicine (For admin only)	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	
23		View medicine list	Pass <input type="checkbox"/>	Fail <input type="checkbox"/>	

This test has been performed by:

Name : \_\_\_\_\_

Signature : \_\_\_\_\_

Date : \_\_\_\_\_

### **3.8 Potential Use of Proposed Solution**

There are many existing applications or systems that allow users to book an appointment to meet with a teacher, a doctor, a therapist, or to change their identification card, among other things. This type of application exists to help users reduce their workload by reducing their time spent from queuing.

The Doctor Appointment Booking Web Application allows the patient to book an appointment with any doctor at any clinic in the district of Pekan. This application will undoubtedly be beneficial to its users. This is due to the application's interesting and useful features. The admin, clinic representative, who is the clinic's representative, or staff, the doctor, and the patient are the application's users.

Patients who already have an account can book appointments on their own. Before scheduling an appointment, patients can review the doctor's profile. This will assist the patient in determining which type of doctor they prefer.

This application also includes a medical record that contains a signature section that allows the doctor to sign the medical record information if the patient's need a verified document for their health problem.

Furthermore, users can view any medicine information that they want to know. This application also includes information on all clinics in the Pekan district with its map location. Patients can schedule an appointment with any clinic they want. There will also be a list of doctors for each clinic that can allow the patient to choose the doctor that they want to schedule the appointment with.

The cancellation of appointment from the clinic, will automatically send an email to the user's email. This is to inform them about the cancellation. Contact information of the users are also available to make it easier for the clinic representative to contact the user if have any enquiry. There is also a feedback section that all users can access. The feedbacks will allow users to read other users' feedback. This feedback will also help the developer to improve the system.



## CHAPTER 4

### IMPLEMENTATION, RESULT AND DISCUSSION

#### 4.1 INTRODUCTION

The fourth chapter discusses the development, implementation, and testing of the Doctor Booking Appointment Web Application. This system has a small scope and is aimed at a healthcare institution in Pekan, Pahang, and its users. This system consists of five modules. The Laravel framework is used to create this system. It also makes use of software such as Visual Studio Code, Xampp and GitHub. This system is tested to identify any potential errors and to correct them as soon as they are discovered.

#### 4.2 DEVELOPMENT TOOLS

The software tools used in developing this system with their respective purposes are shown in Table 4.1 below.

Table 4.1 Software Development Tools

No	Software Tools	Purpose
1	Visual Studio Code	Used for scripting.
2	Xampp	To use localhost.
3	GitHub	To store the project code.

#### 4.3 IMPLEMENTATION

Documenting all the steps taken in the system's development is the goal of the implementation process. This system will be a website-based system with five modules which are: manage account, manage clinic, manage appointment booking, manage medicine information, and manage medical records.

### 4.3.1 Installing Software Tools

Software tools such as Visual Studio Code and Xampp are used in the system development. All this software must first be installed before system development can begin.

#### 4.3.1.1 Visual Studio Code

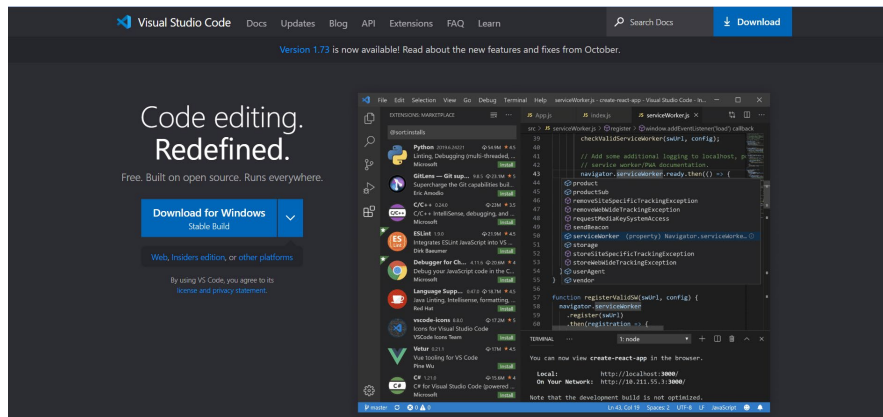


Figure 4.1 Installing Visual Studio Code

Visual Studio Code includes built-in support for JavaScript, TypeScript, and Node.js, as well as a robust ecosystem of extensions for other languages and runtimes (including C++, C#, Java, Python, PHP, Go, and .NET).

Visual Studio Code will be used for this system project programming. The system will be built with the Java programming language and the Laravel framework using MVC (Model-View-Controller) architecture. Visual Studio Code can also connect to GitHub to store project code.

### 4.3.1.2 Xampp

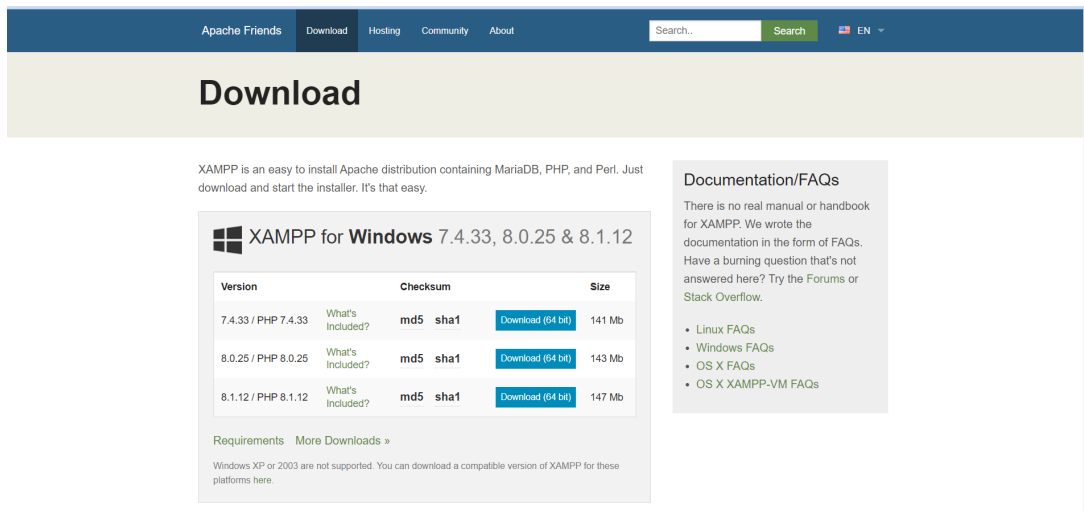


Figure 4.2 Installing Xampp

It is a free and open-source web solution that includes the Apache distribution for many servers and command-line executables, as well as modules such as Apache server, MariaDB, PHP, and Perl.

Before releasing the website to the main server, a local host or server can use XAMPP to test it on computers and laptops for this project. It is a platform that provides an appropriate environment for testing and verifying the operation of projects based on Apache, Perl, MySQL databases, and PHP through the host's system, such as Doctor Booking System.

### 4.3.1.3 Installing Laravel Framework

Laravel is a PHP-based web framework that uses significant and graceful syntaxes to build high-end web applications. It includes features from technologies such as ASP.NET MVC, CodeIgniter, Ruby on Rails, and many others. Since the architecture of this system is MVC, the Laravel framework is proved to be appropriate for its development.

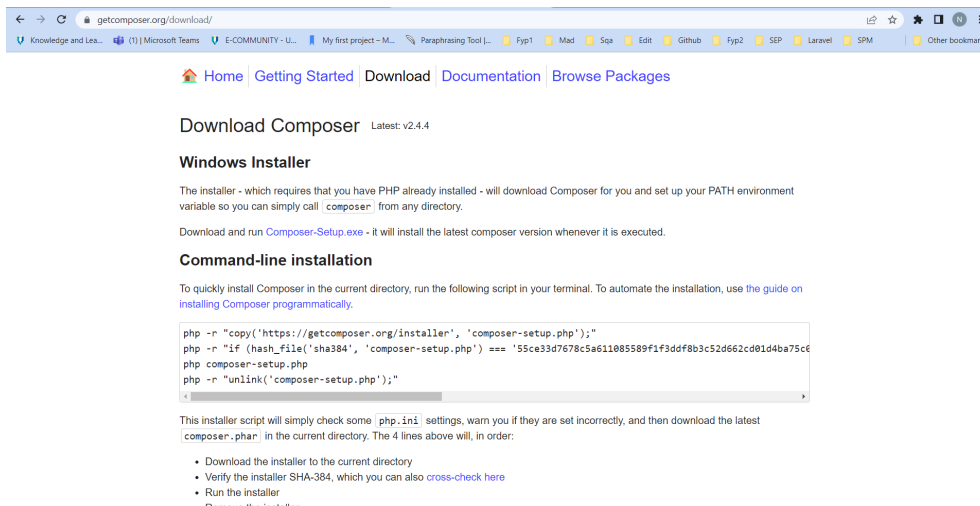


Figure 4.3 Installing Laravel Framework

Before installing Laravel, you must first install composer. After installing composer, we can install Laravel by entering the command below into the command prompt.

```
composer global require "laravel/installer"
```

Figure 4.4 Composer Command

### 4.3.2 Collect Multimedia Contents

Many multimedia contents are needed in this system such as images that will be used at main page and background.

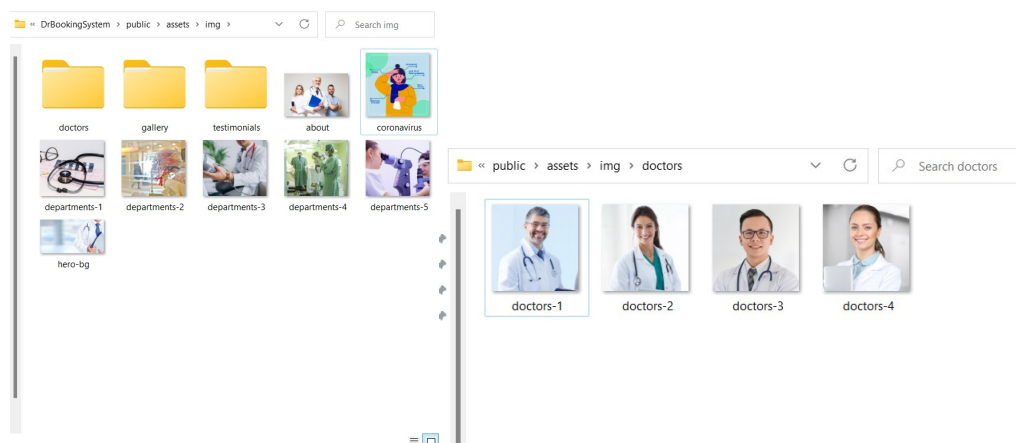


Figure 4.5 Multimedia Content

### **4.3.3 Development of Doctor Booking System**

The development phase can begin after installing the necessary software and gathering the necessary multimedia content. Visual Studio Code is used in this phase to code the user interface and functions required for this project's five modules, which are manage account, manage medicine information, manage clinic, manage appointment booking, and manage medical records. The development will be explained in five sections: All users can access, admin, clinic representative, doctor and patient or user.

### 4.3.3.1 All users can access

#### 4.3.3.1 Main page

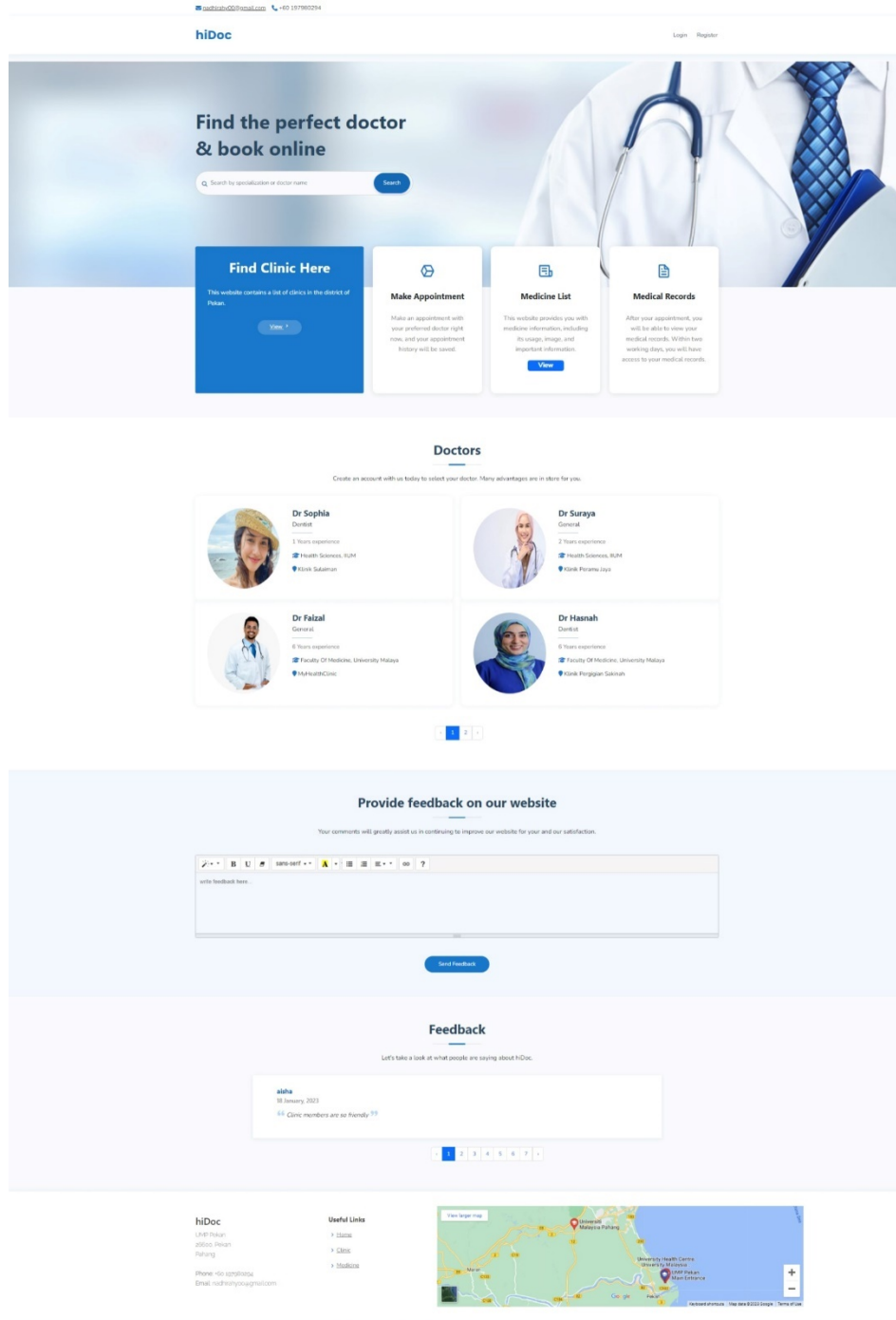


Figure 4.6 Website Main page

The main page for this project's website is depicted in [Figure 4.6](#). A few buttons on this interface can take the user to other pages. For instance, clicking "View >" will take

the user to a page that will display a list of clinics in district Pekan. A list of doctors who have already registered with the system is also displayed on this page. By filling out the feedback form, the user can also provide feedback for the page. The links, which may also lead users to other pages, are in the footer.

#### 4.3.3.2 Login

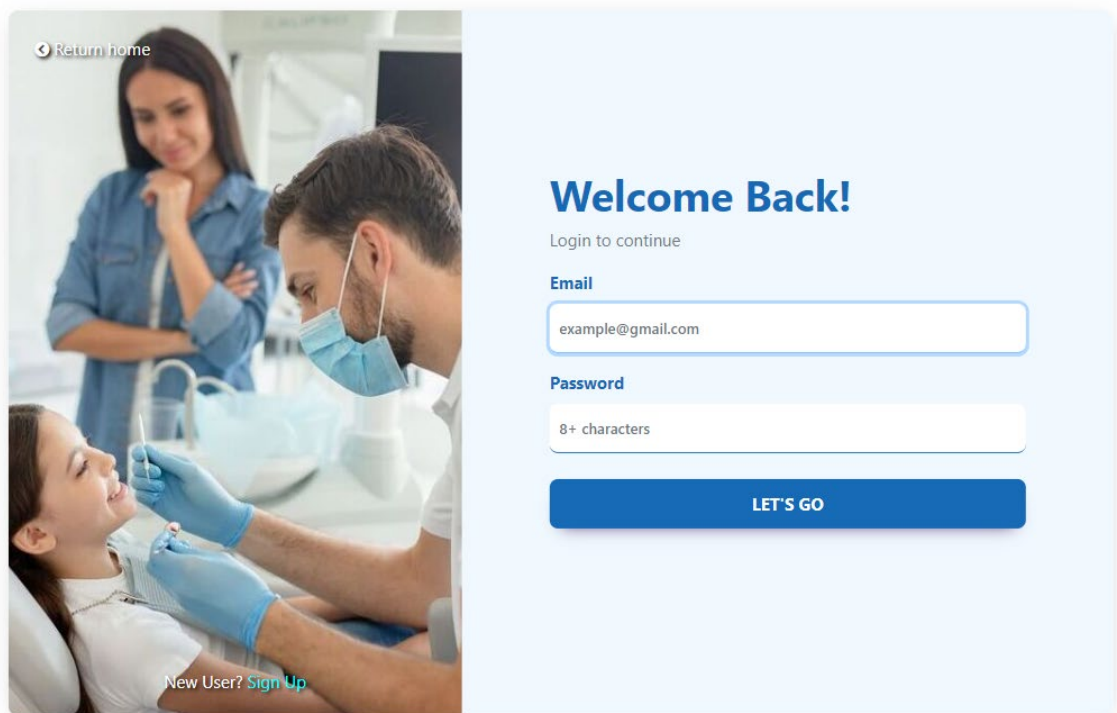


Figure 4.7 Login

The login page for all system users is shown in [Figure 4.7](#). To log into the system, the user must enter their email address and password. They will be taken to their appropriate page after logging in.

### 4.3.3.3 Register

Return home

## Welcome to hiDoc!

Register your account

**Name**

**Email**

**Password**

**Confirm Password**

**What Role Are You?**

**SIGN UP**

Have Account? [Sign In](#)

Figure 4.8 Register

The registration page shown in [Figure 4.8](#) is where new users register to access the system's features and services. The login information for new users includes their name, email address, password, and their role in this system. Additionally, they must verify their password.



#### 4.3.3.4 Search doctor

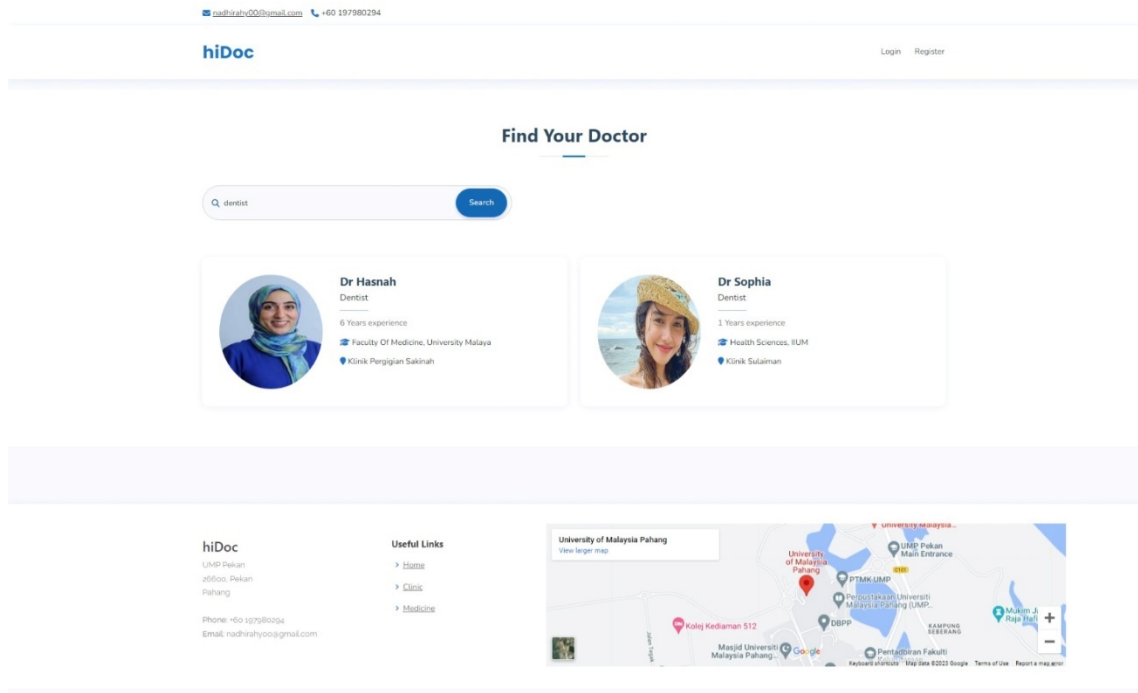


Figure 4.9 Search doctor

The figure above shows how users can search for doctors by specialisation or doctor name. If the user type is not "user" or the user has not yet logged into the system, the system will hide the book button, which will allow the user to book an appointment with the doctor. This figure can see at [Figure 4.39](#).

### 4.3.3.5 Manage Clinic Module (Clinic List)

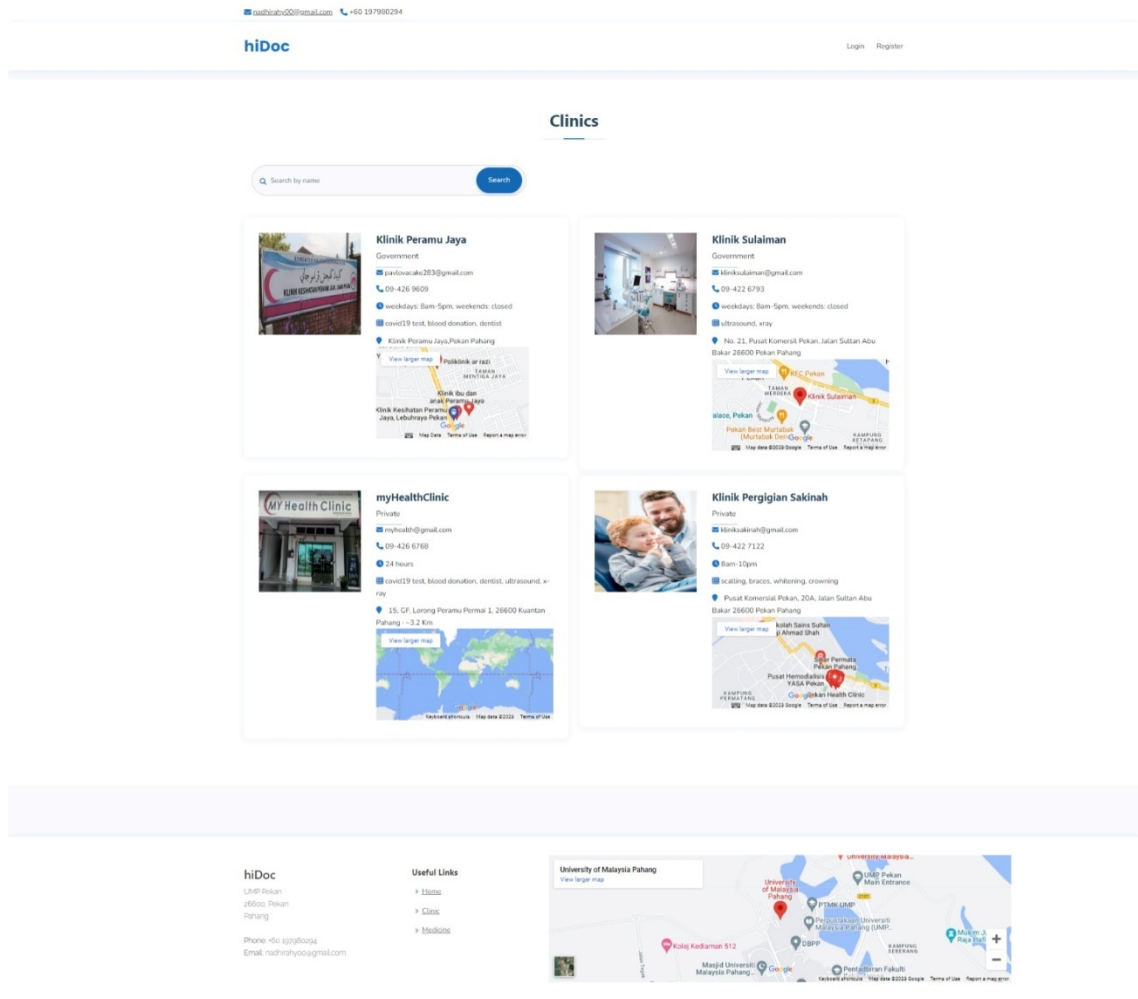


Figure 4.10 Manage Clinic Module (Clinic List)

The clinics listed in the figure above have registered with this system. This interface contains clinic information such as the clinic's name, email address, contact number, type of clinic (government or private), work or operating hours, and clinic address. The system also displays the map location of each clinic to assist the user in determining the clinic's location.

### 4.3.3.6 Manage Medicine Module (Search & View Medicine)

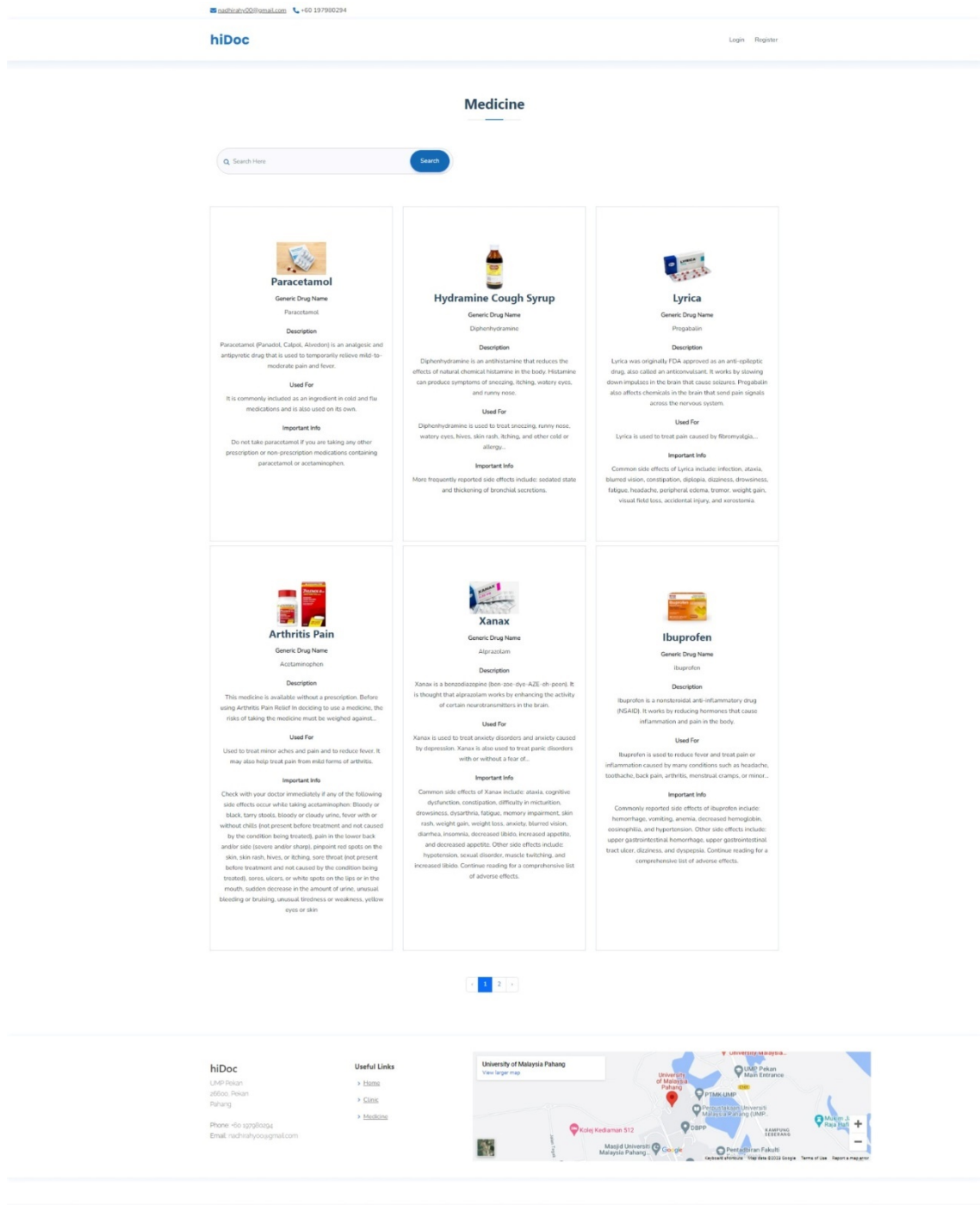


Figure 4.11 Manage Medicine Module (User View)

The figure above depicts the medicine interface, which displays a list of medicine data available in the system. The system administrator enters this information. This list is viewable by all users without logging into the system. This interface displays an image

of the medicine, its name, its generic name, what it is used for, a description, and some important information such as the potential side effects of taking the medicine.

### 4.3.3.2 Admin Interface

#### 4.3.3.1 Homepage

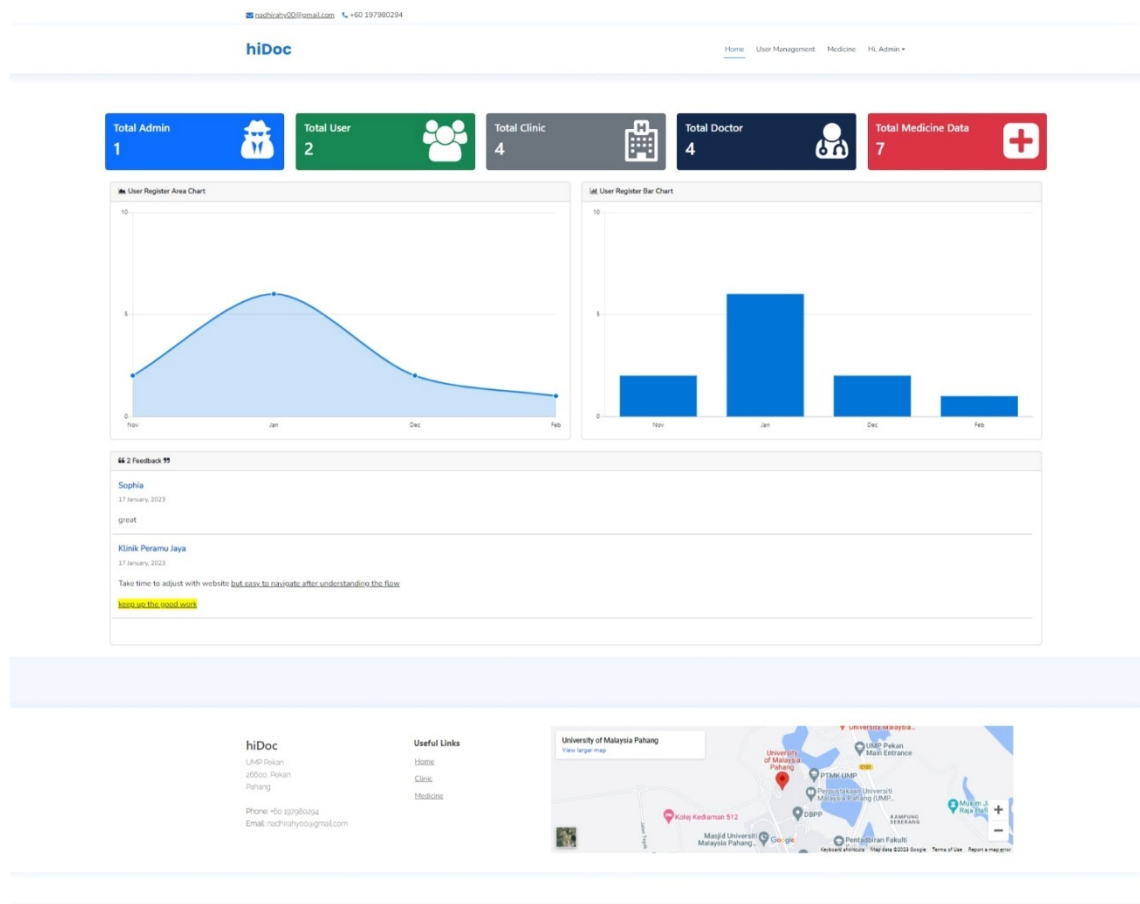


Figure 4.12 Admin Homepage

The figure above is the homepage for admin. This page displays the number of admins, users, clinics, and doctors who have registered in the system. In addition, the total number of medicine data stored in the database. The interface also includes an area and bar chart showing how many users registered in the system each month. There is also a list of user feedback for the admin to view.

### 4.3.3.2 Manage Account Module (User Management)

The screenshot displays the 'hiDoc' User Management interface. At the top, there is a navigation bar with 'Home', 'User Management', 'Medicine', and 'Hi Admin'. Below this is a table listing registered users. The table has the following columns: #, NAME, EMAIL ADDRESS, ROLE AS, JOINING DATE, and ACTION. The 'ROLE AS' column contains colored buttons: green for 'Doctor', blue for 'Clinic', and yellow for 'User'. The 'ACTION' column contains red 'Delete' buttons. Below the table is a pagination control showing '1' and '2'. At the bottom of the page, there is a footer section with 'hiDoc' contact information, 'Useful Links' (Home, Clinic, Medicine), and a map of the University of Malaysia Pahang.

#	NAME	EMAIL ADDRESS	ROLE AS	JOINING DATE	ACTION
1	Hassan	hassan@gmail.com	Doctor	18 February, 2023	Delete
2	Klinik Ponggan Sakinah	klinsakinah@gmail.com	Clinic	18 January, 2023	Delete
3	ms023	muhammadam23@gmail.com	User	17 January, 2023	Delete
4	Fauzi	fauzi@gmail.com	Doctor	17 January, 2023	Delete
5	muhathClinic	muhath@gmail.com	Clinic	17 January, 2023	Delete
6	Klinik Subman	klinsubman@gmail.com	Clinic	17 January, 2023	Delete
7	Klinik Peramu Jaya	peramujaya2023@gmail.com	Clinic	17 January, 2023	Delete
8	Surya	surya@gmail.com	Doctor	14 December, 2022	Delete
9	Sapha	sapha00@gmail.com	Doctor	14 December, 2022	Delete
10	Admin	admin@gmail.com	Clinic	18 November, 2022	Delete

Figure 4.13 Manage Account Module (User Management)

Figure 4.13 shows a list of users who have registered with the system. Following registration, the administrator will confirm that the clinic establishment information provided is correct. If the account is not legal or is a forgery, the administrator has the authority to delete it. The administrator may also delete any user account.

### 4.3.3.3 Manage Medicine Module (Medicine Management)

The screenshot displays the 'hiDoc' Manage Medicine Module. At the top, there is a navigation bar with 'Home', 'User Management', 'Medicine', and 'Hi Admin'. Below this is a table with the following columns: #, BRAND NAME, GENERIC NAME, IMAGE, and ACTION. The table contains six rows of medicine data. Each row includes a small image of the medicine packaging. The ACTION column for each row contains three buttons: 'View', 'Edit', and 'Delete'. A '+ New Medicine' button is located at the top right of the table. Below the table is a pagination control showing '1' and navigation arrows. At the bottom of the page, there is a footer section with 'hiDoc' contact information, 'Useful Links' (Home, Clinic, Medicine), and a map of the University of Malaysia Pahang.

#	BRAND NAME	GENERIC NAME	IMAGE	ACTION
1	Paracetamol	Paracetamol		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
2	Hydrantine Cough Syrup	Diphenhydramine		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
3	Lonic	Progabutin		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
4	Arthritis Pan	Acetaminophen		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
5	Xanax	Alprazolam		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
6	Risperidone	Risperidone		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

Figure 4.14 Manage Medicine Module (Medicine Management)

The list of medications that the admin has already entered the system is displayed on the page in [Figure 4.14](#). The administrator can add, view, edit and delete any existing information about medicines. A picture of the medication may also be included by the admin.

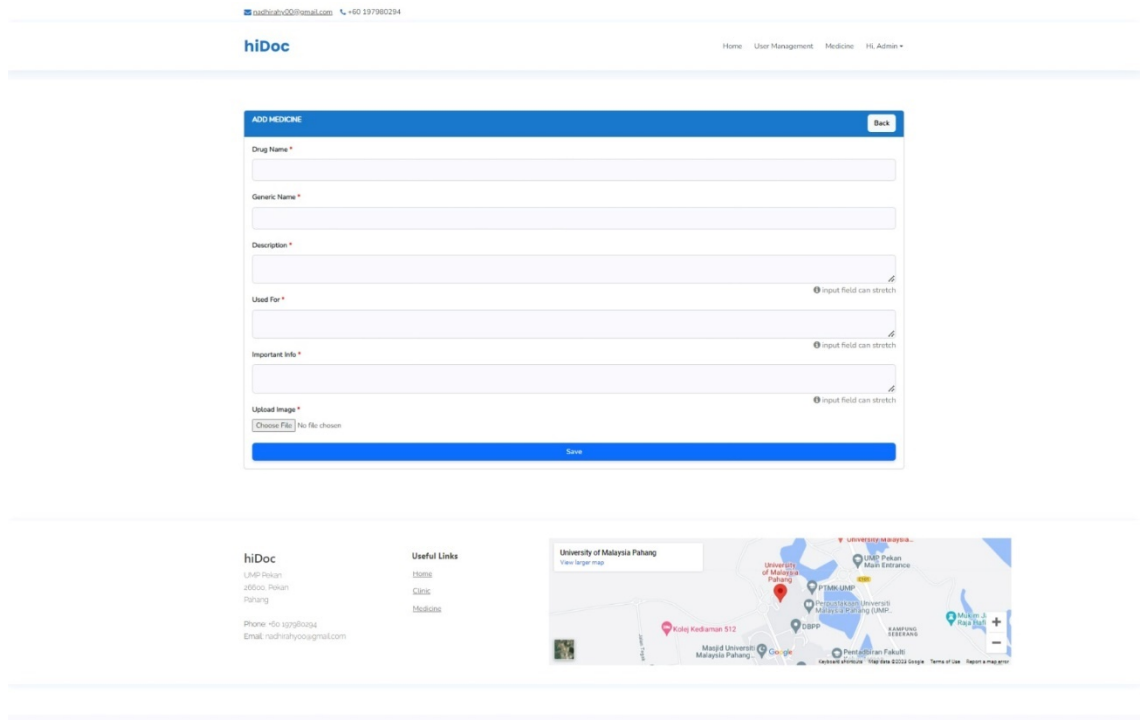


Figure 4.15 Manage Medicine Module (Add Medicine)

This is the add medicine page. When the admin clicks “+ New Medicine” button in [Figure 4.14](#). The admin will be directed to this page.

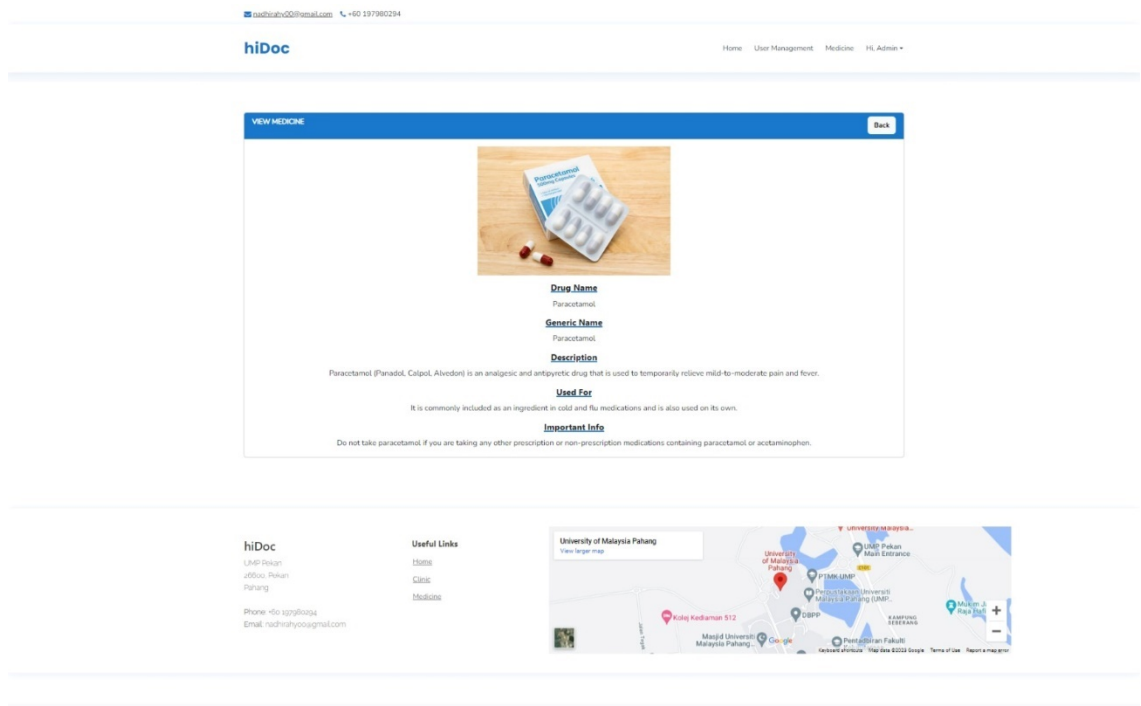


Figure 4.16 Manage Medicine Module (View Medicine)

This is the view medicine interface. When the admin clicks the “View” button, the admin will be directed to this page.

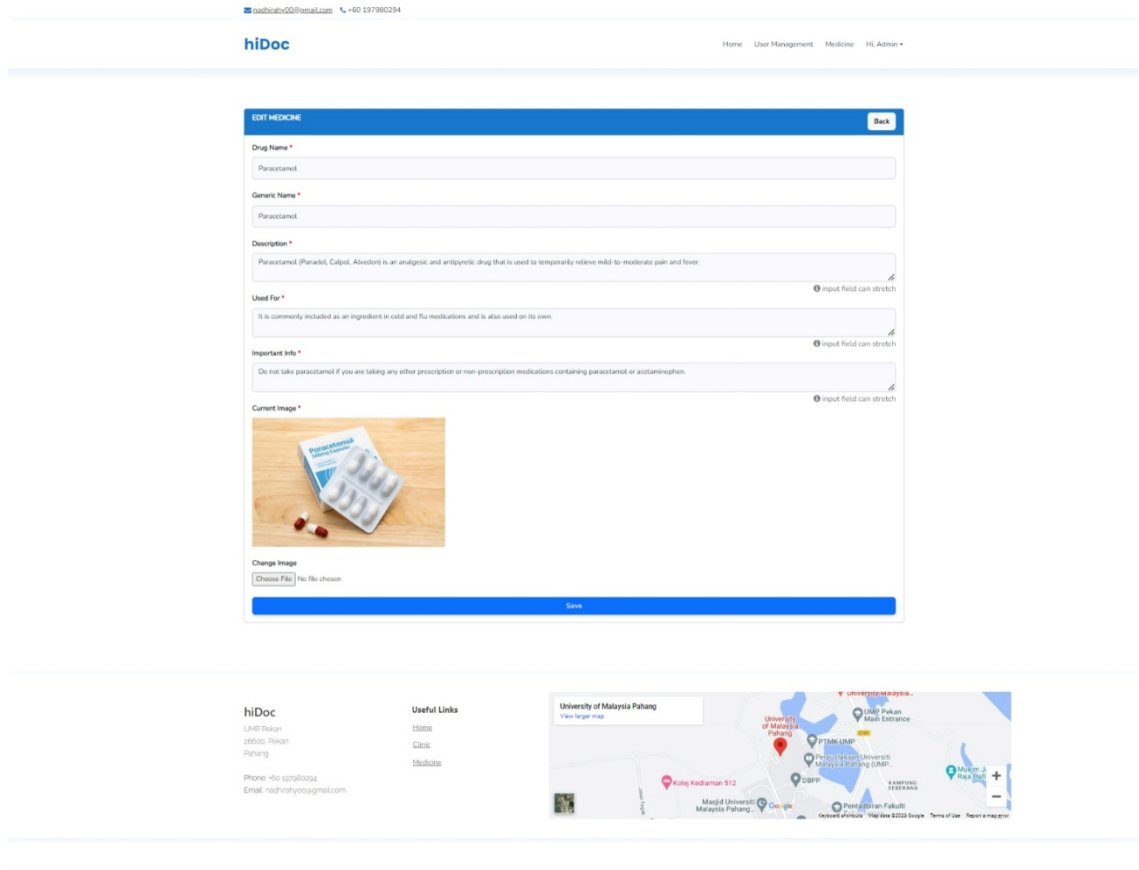


Figure 4.17 Manage Medicine Module (Edit Medicine)

The page where the administrator can edit the medication information is depicted in Figure 4.17 above. Any information can be edited by the administrator if he so chooses. The "Back" button is also present on this page to make it easier for the user to go back to the previous page.



### 4.3.3.3 Clinic Interface

#### 4.3.3.1 Get Started

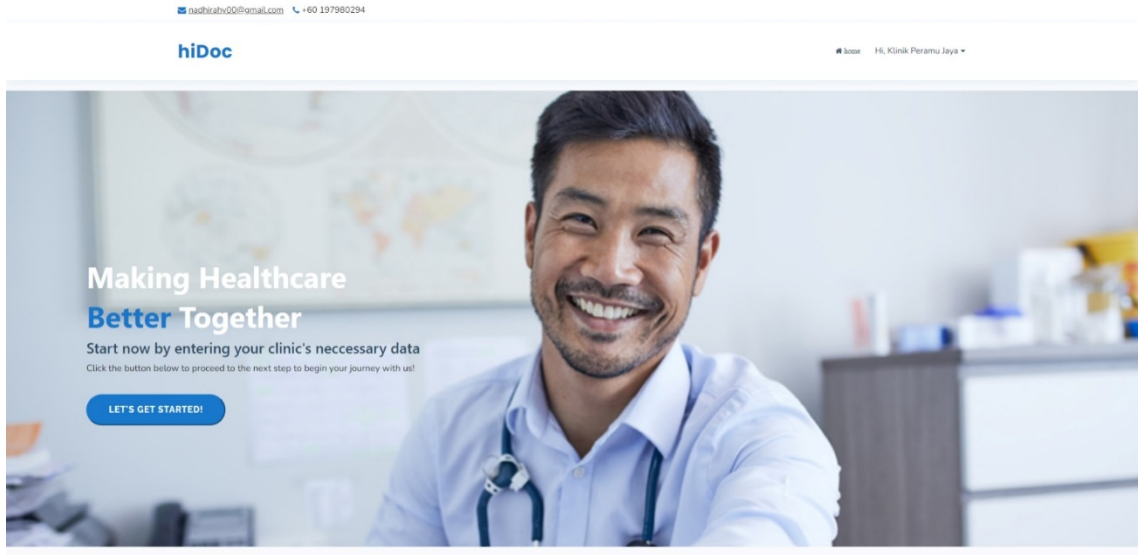


Figure 4.18 Clinic Get Started

Following registration, the clinic representative or staff will be directed to a get started page where they will be directed to a page where they must enter the clinic information by clicking the "LET'S GET STARTED" button. This page is depicted in Figure 4.19 below.

nadhirah00@gmail.com +60 197980294


hiDoc Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya

# Hi, Klinik Peramu Jaya

This is your profile page. You can update your profile information by clicking the "Update Profile" button.

**CLINIC INFORMATION**

Clinic Photo



Choose File No file chosen

Clinic Name Email Address

Klinik Peramu Jaya pavlovacake283@gmail.com

Address

Address

---

**ABOUT CLINIC**

Government/Private Clinic

Choose clinic type

Services available

Services

Operating Hours

Operating Hours

Contact Number

Contact Number

[Update Profile](#)

Made by Nur Nadhirah

**hiDoc**

UWP Pekan  
26600 Pekan  
Pahang

Phone: +60 197980294  
Email: nadhirah00@gmail.com

**Useful Links**

[Home](#)

[Appointment List](#)

[Clinic](#)

[Medicine](#)




Figure 4.19 Insert Clinic Information

The figure above depicts an interface where the newly registered clinic must enter clinic information such as contact information, work hours, services offered, clinic address, image, and whether they are a private or government clinic.

### 4.3.3.2 Homepage

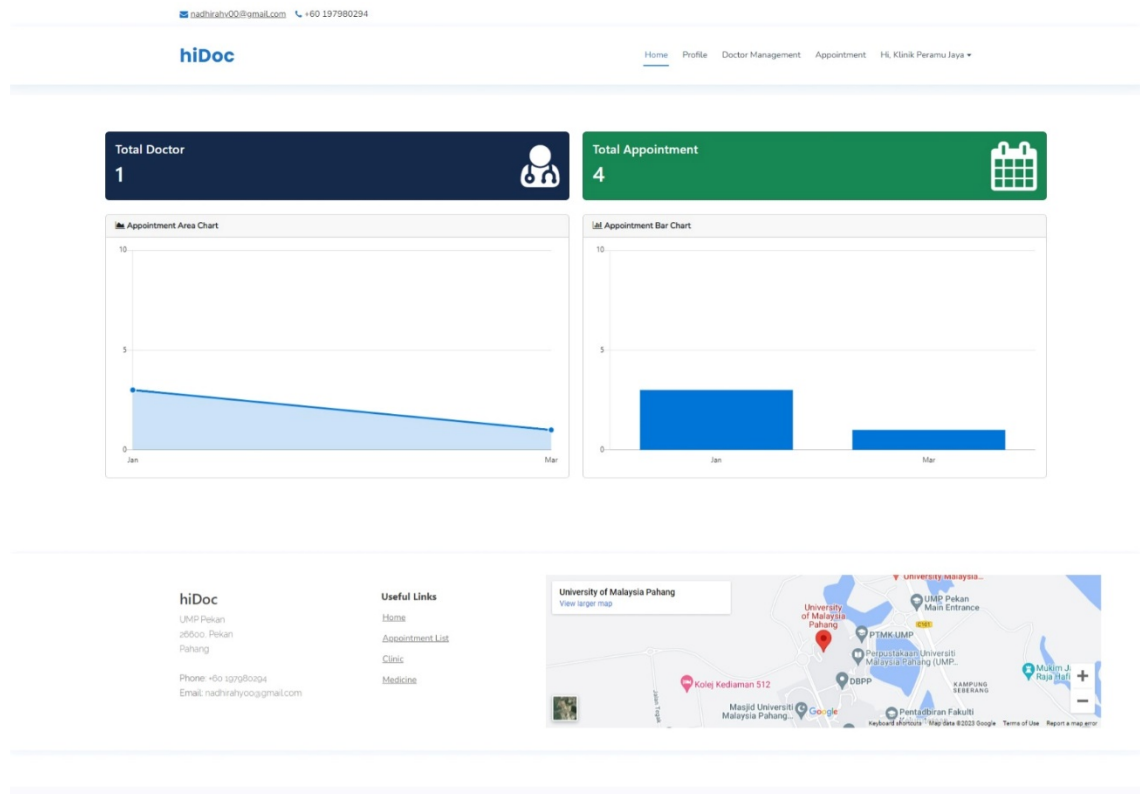


Figure 4.20 Clinic Homepage

The clinic's home page is depicted in Figure 4.20. This page lists the number of clinic's doctors, and total appointments of the clinic. Each clinic page has a unique navigation bar that is only available to clinic staff. There are also an area and bar chart that shows the total appointment for the clinic by month. Only three modules—manage accounts, manage appointment booking and manage medical record—are used by the clinic staff.

### 4.3.3.3 Manage Account Module (Doctor Management)

hiDoc

Home Profile Doctor Management Appointment Hi Klinik Peramu Jaya

### Doctor Management

This is a list of doctors who have signed up with your clinic.

#	IMAGE	NAME	GENDER	PHONE NUMBER	SPECIALIZE	YEARS OF WORKING	EDU BACKGROUND	WORKPLACE	ACTION
1		Suraya	Female	01478965	General	2	Health Sciences, IUM	Klinik Peramu Jaya	<a href="#">Remove</a>

**hiDoc**  
UMP Pekan  
26500, Pekan  
Pahang  
Phone: +60 137980294  
Email: nadhiahyoog@gmail.com

**Useful Links**  
[Home](#)  
[Appointment List](#)  
[Clinic](#)  
[Medicine](#)

University of Malaysia Pahang  
View larger map

University of Malaysia Pahang  
UMP Pekan Main Entrance  
PTMK UMP  
Pusat Universiti Malaysia Pahang UMP  
DBPP  
KAMPUNG SERANG  
Masjid Universiti Malaysia Pahang  
Dentobiran Fakulti  
Kolej Kediaman 512  
Museum J. Raja Idris

Figure 4.21 Manage Account Module (Doctor Management)

Figure above show the clinic's list of doctors. On this page, the clinic administrator or staff can remove the doctor account. The clinic administrator has account removal rights for doctors.

### 4.3.3.4 Manage Account Module (Clinic Profile)

The screenshot shows the hiDoc web application interface for managing a clinic profile. At the top, there is a header with the hiDoc logo, contact information (nadhira00@gmail.com, +60 197980294), and navigation links (Home, Profile, Doctor Management, Appointment, Hi Klinik Peramu Jaya). The main content area features a dark blue banner with the text "Hi, Klinik Peramu Jaya" and a sub-message: "This is your profile page. You can update your profile information by clicking the 'Update Profile' button." Below the banner is a white form titled "CLINIC INFORMATION". The form includes a "Clinic Photo" section with a placeholder image of the clinic sign and a "Choose File" button. The "Clinic Name" field contains "Klinik Peramu Jaya" and the "Email Address" field contains "pa-fovvcake283@gmail.com". The "Address" field contains "Klinik Peramu Jaya, Pekan Pahang". The "ABOUT CLINIC" section has a "Government/Private Clinic" dropdown set to "Government", a "Services available" field containing "covid19 test, blood donation, dentist", an "Operating Hours" field containing "weekdays: 8am-5pm, weekends: closed", and a "Contact Number" field containing "09-426 9609". A blue "Update Profile" button is at the bottom of the form. Below the form, it says "Made by Nur Nadhirah". The footer contains "hiDoc" contact info, "Useful Links" (Home, Appointment List, Clinic, Medicine), and a map of the clinic's location in UMP Pekan, Pahang.

Figure 4.22 Clinic Profile

The clinic profile is shown in the figure above. It includes the name of the clinic, its email address, the contact number, services, the clinic's address, and operation hours. The clinic staff can update their clinic profile information in this interface.

### 4.3.3.5 Manage Appointment Booking Module (Appointment List & History)

The screenshot displays the 'Appointment List' interface in the hiDoc system. At the top, there is a navigation bar with 'Home', 'Profile', 'Doctor Management', 'Appointment', and 'Hi Klinik Peramu Jaya'. The main heading is 'Appointment List', with a sub-note: 'Your clinic's appointment schedule is listed starting from the latest appointment date'.

#	PATIENT NAME	CONTACT NO	DATE	TIME SLOT	MESSAGE	DOCTOR SELECTED	STATUS	ATTENDANCE	ACTION	CANCELLATION	MEDICAL RECORD
1	Nur Nadhirah	0197980294	23 March, 2023	8:30 AM - 9:00 AM	sakit bahu	Dr Suraya	Approve	Absent	Update Delete	Send Email	No record
2	nss023	01137512383	26 January, 2023	8:00 AM - 8:30 AM	Good	NULL	Reject	Pending	Update Delete	Send Email	No record
3	nss023	01137512383	26 January, 2023	8:30 AM - 9:00 AM	I need to check my health.	Dr Suraya	Approve	Attend	Update Delete	Send Email View Delete	
4	Nur Nadhirah	0197980294	20 January, 2023	9:00 AM - 9:30 AM	back pain	Dr Suraya	Cancel	Cancel	Update Delete	Send Email	No record

Below the table, there are two sections: 'Useful Links' and a map. The 'Useful Links' section includes: Home, Appointment List, Clinic, and Medicine. The map shows the location of the University of Malaysia Pahang, with markers for 'University of Malaysia Pahang', 'PTMK UMP', 'Perpustakaan Universiti Malaysia Pahang (UMP)', 'Masjid Universiti Malaysia Pahang', 'Kolej Kediaman 512', 'DBPP', 'Pentagon Falsafi', 'KAMPUS BERSEKUTU', and 'Makam J. Raja Jait'. The map also includes a search bar and a 'View larger map' link.

Figure 4.23 Appointment List or History

Figure above is the appointment list of the clinic. It contains information on the requested and past appointment. The clinic staff can update the status of appointment to “Approve” or “Reject” and they can also update the attendance status to “Attend” or “Absent”, this can be seen at Figure 4.24 below. Attendance status is a status of whether the patient come to the appointment session or not. The clinic staff can also cancel the appointment by clicking the ‘Send Email’ button. A cancellation email will be sent to the patient’s email address. The cancellation email can be seen at Figure 4.25 below.

**UPDATE STATUS** Back

**Patient Name \***  
Nur Nadhirah

**Doctor Name**  
Dr Suraya

**Appointment Date \***  
23/03/2023

**Time Slot \***  
8:30 AM - 9:00 AM

**Message**  
sakit bahu

**Status \***  
Approve

**Attendance \***  
Absent

**Update**

**hiDoc**  
UMP Pekan  
26600, Pekan  
Pahang  
Phone: +60 197980294  
Email: nadhirah00@gmail.com

- Useful Links**
- Home
  - Appointment List
  - Clinic
  - Medicine



Figure 4.24 Update status

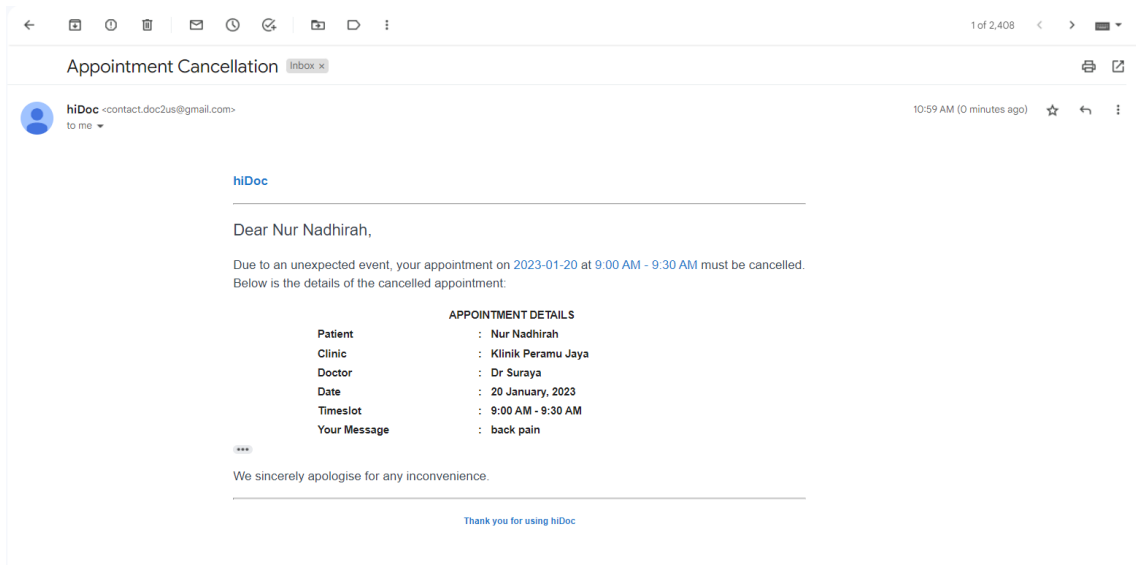


Figure 4.25 Appointment Email Cancellation

### 4.3.3.6 Manage Medical Records Module (Add, view, delete)

3	rs0023	01137512383	26 January, 2023	8:30 AM - 9:00 AM	I need to check my health.	Dr. Suniya	Approve	Attend	Update	Delete	Send Email	Add
---	--------	-------------	------------------	-------------------	----------------------------	------------	---------	--------	--------	--------	------------	-----

Figure 4.26 Medical Record Status

For this module, if the attendance status of patient is “Attend”, in the medical record column, an “Add” button will appear. The clinic staff must input the diagnosis details of the patient after they have finished their appointment session with the doctor.

In this field, enter the doctor's diagnosis for the patient.

Patient Name  
rs0023

Appointment Date  
26/01/2023

Timeslot  
8:30 AM - 9:00 AM

Patient's Message  
I need to check my health.

Diagnosis \*  
Enter patient's medical diagnosis

Prescribed Medicine \*  
Enter prescribed medicine

Save

hiDoc  
UMP Pekan  
26500, Pekan  
Rahang  
Phone: +60 197980294  
Email: nadhraty00@gmail.com

Useful Links  
Home  
Appointment List  
Clinic  
Medicine

University of Malaysia Pahang  
View larger map

University of Malaysia Pahang  
UMP Pekan Main Entrance  
PTMK UMP  
Pusat Universiti Malaysia Pahang UMP  
Masjid Universiti Malaysia Pahang  
Masjid Raja Idris  
KAMPUS BERBANG  
Pontian (Biran Fakulti)  
Kolej Kediaman 512  
Masjid Universiti Malaysia Pahang

Figure 4.27 Add Medical Record

3	rs0023	01137512383	26 January, 2023	8:30 AM - 9:00 AM	I need to check my health.	Dr. Suniya	Approve	Attend	Update	Delete	Send Email	View	Delete
---	--------	-------------	------------------	-------------------	----------------------------	------------	---------	--------	--------	--------	------------	------	--------



Figure 4.28 Medical Record Column Changes Button

After adding the medical record, the button in the medical record will change to “view” and ‘delete” button. View button is for the clinic staff to view the patient’s medical record.

MEDICAL RECORD	
Patient Name	nss023
Clinic	Klinik Peramu Jaya
Doctor	Dr Suraya
Date	26 January, 2023
Timeslot	8:30 AM - 9:00 AM
Patient's Message	I need to check my health.
Diagnosis	Fever
Prescribed Medicine	Cough syrup and panadol

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

**hiDoc**

Figure 4.29 Patient's Medical Record

The image above depicts the patient's medical record for a specific appointment. In the medical record, there is an authorised signature section for the doctor to sign. This medical record may be useful in the future if the patient needs to verify why he or she needs to bring extra medicine on the plane.

### 4.3.3.4 Doctor Interface

#### 4.3.3.1 Get Started

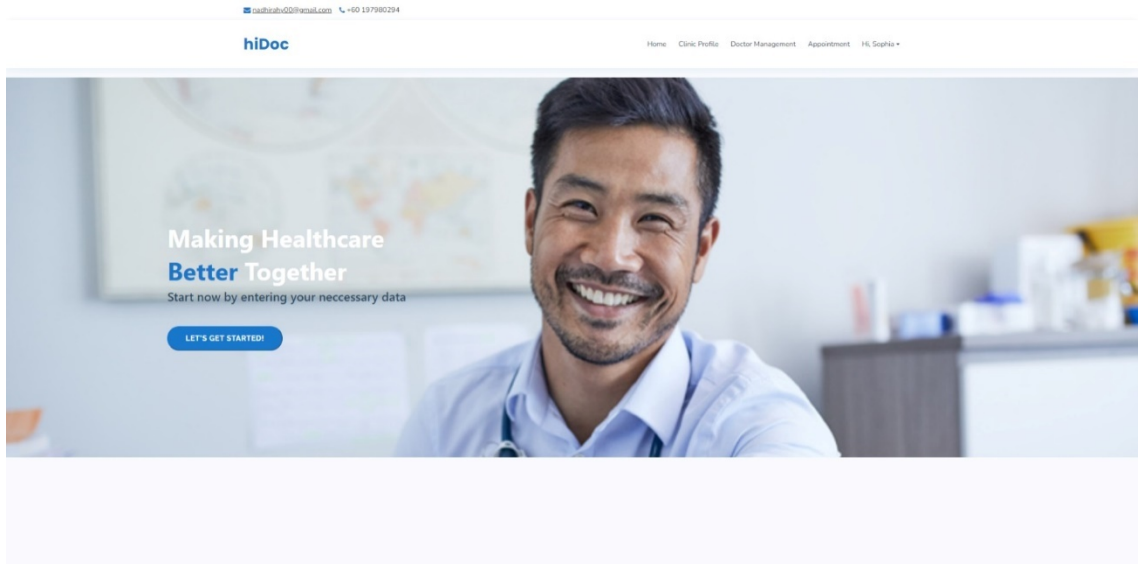


Figure 4.30 Get Started

Following registration, the doctor will be directed to a get started page where they will be directed to a page where they must enter their information by clicking the "LET'S GET STARTED" button. This page is depicted in Figure 4.31 below.

[nadhraly00@gmail.com](#) +60 19790294


**hiDoc**

[Home](#)
[Profile](#)
[Appointment](#)
[Hi, Sophia](#)

Hi,  
**Sophia**  
Get Started With Your Profile Info to Start Your Journey With Us!

**USER INFORMATION**

Profile Photo



No file chosen

Name

Gender Email Address

Phone Number Workplace

---

**MEDICAL BACKGROUND**

Specialization

Working Experience (Years)

---

**ABOUT ME**

Education Background

Made by Nur Nadhrarah

**hiDoc**

UMP Pekan  
25000, Pekan  
Pahang

Phone: +60 19790294  
Email: nadhraly00@gmail.com

**Useful Links**

[Home](#)  
[Appointment List](#)  
[Clinic](#)  
[Medicine](#)




Figure 4.31 Insert Doctor Information

The figure above depicts an interface where the newly registered doctor must enter their information such as contact information, workplace, their specialization, gender, education background, image, and years of working experience.

### 4.3.3.2 Homepage

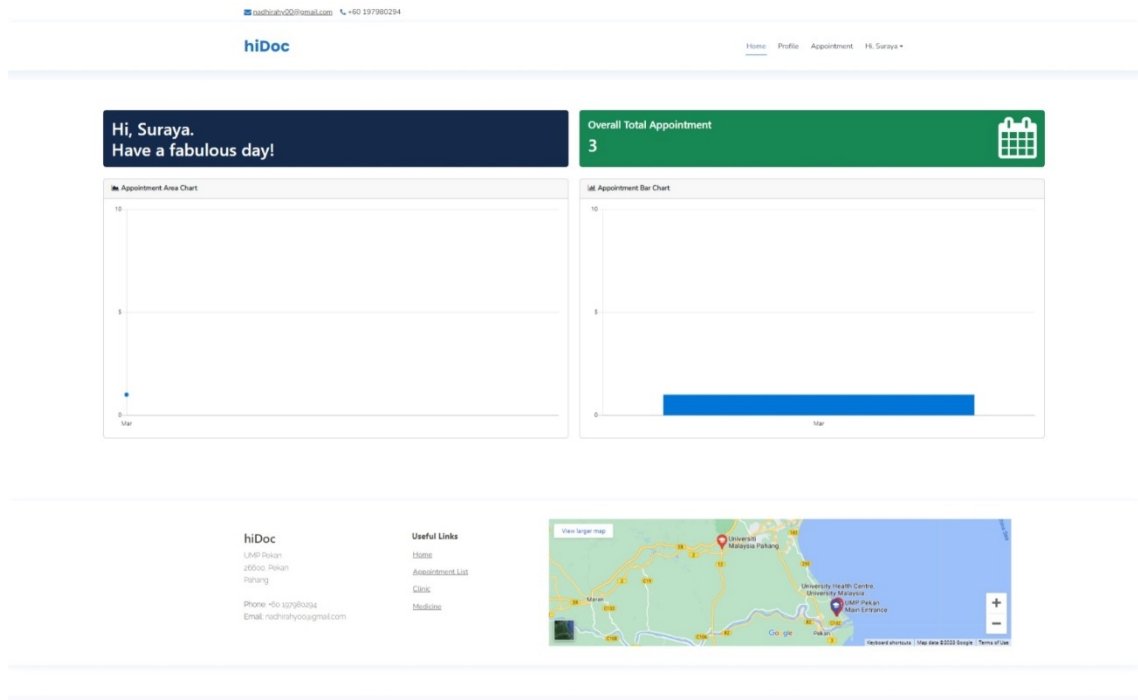


Figure 4.32 Doctor Homepage

The doctor's homepage is depicted in the figure above. It includes a count of all appointments made for the doctor. Each doctor's page has a unique navigation bar that is only available to doctors. There are also an area and bar chart that shows the total appointment for the doctor by month.

### 4.3.3.3 Manage Account Module (Doctor Profile)

The screenshot displays the 'hiDoc' web application interface for a doctor's profile. At the top, the header includes the 'hiDoc' logo, navigation links for 'Home', 'Profile', 'Appointment', and 'Hi, Suraya', and contact information: 'nadhrah@u2019hospital.com' and '+60 1979802294'. The main content area features a dark blue banner with the text 'Hi, Suraya' and a sub-message: 'This is your profile page. You can update your profile information by clicking the "Update Profile" button.' Below this is a white form titled 'USER INFORMATION 1' with a 'Profile Photo' section containing a circular image of a female doctor. The form includes several input fields: 'Choose File' (No file chosen), 'Name' (Suraya), 'Gender' (Female), 'Email Address' (suraya@gmail.com), 'Phone Number' (01478965), and 'Workplace' (Klinik Peramu Jaya). A 'MEDICAL BACKGROUND' section contains 'Specialization' (General) and 'Working Experience (Years)' (2). An 'ABOUT ME' section has an 'Education Background' field with the text 'Health Sciences, UKM'. A blue 'Update Profile' button is located at the bottom of the form. Below the form, it says 'Made by Nur Nadhirah'. The footer contains 'hiDoc' contact details, 'Useful Links' (Home, Appointment List, Clinic, Medicine), and a map of the University of Malaysia Pahang area.

Figure 4.33 Doctor Profile

The doctor's profile page is depicted in the figure above. It includes the doctor's name, email address, phone number, workplace, specialization, number of years of employment, education background, and a few other personal facts. The doctor can update their profile in this interface.

### 4.3.3.4 Manage Appointment Booking Module (Appointment List & History)

**Appointment List**

Your appointment schedule is listed starting from the latest appointment date.

#	PATIENT NAME	DATE	TIME SLOT	MESSAGE	STATUS	ATTENDANCE	ACTION	MEDICAL RECORD
1	Nur Nadhrath	23 March, 2023	8:30 AM - 9:00 AM	sakit bahu	Approve	Absent	Attend Absent	No record
2	ns023	26 January, 2023	8:30 AM - 9:00 AM	I need to check my health	Approve	Attend	Attend Absent	View
3	Nur Nadhrath	20 January, 2023	9:00 AM - 9:30 AM	back pain	Cancel	Cancel	NULL	No record

**hiDoc**  
 UMP Pekan  
 26000 Pekan  
 Pahang  
 Phone: +60 39791004  
 Email: nadhrathyoos@gmail.com

**Useful Links**  
[Home](#)  
[Appointment List](#)  
[Clinic](#)  
[Medicine](#)

**University of Malaysia Pahang**  
 View larger map

Figure 4.34 Appointment List & History

This is the interface that displays the doctor's appointment list. The doctor can change the attendance status to indicate whether the patient attended the appointment session. The doctor has the option of selecting "Attend" or "Absent". If the attendance status is "Attend," the doctor can view the patient's medical record after the clinic adds the medical record for the patient.

#### 4.3.3.5 Manage Medical Records Module (View)

MEDICAL RECORD	
Patient Name	nss023
Clinic	Klinik Peramu Jaya
Doctor	Dr Suraya
Date	26 January, 2023
Timeslot	8:30 AM - 9:00 AM
Patient's Message	I need to check my health.
Diagnosis	Fever
Prescribed Medicine	Cough syrup and panadol

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

**hiDoc**

Figure 4.35 Patient's Medical Record

The image above depicts the patient's medical record for a specific appointment. In the medical record, there is an authorised signature section for the doctor to sign. This medical record may be useful in the future if the patient needs to verify why he or she needs to bring extra medicine on the plane or needs of verification for her health problems.

### 4.3.3.5 User Interface

#### 4.3.3.1 Homepage

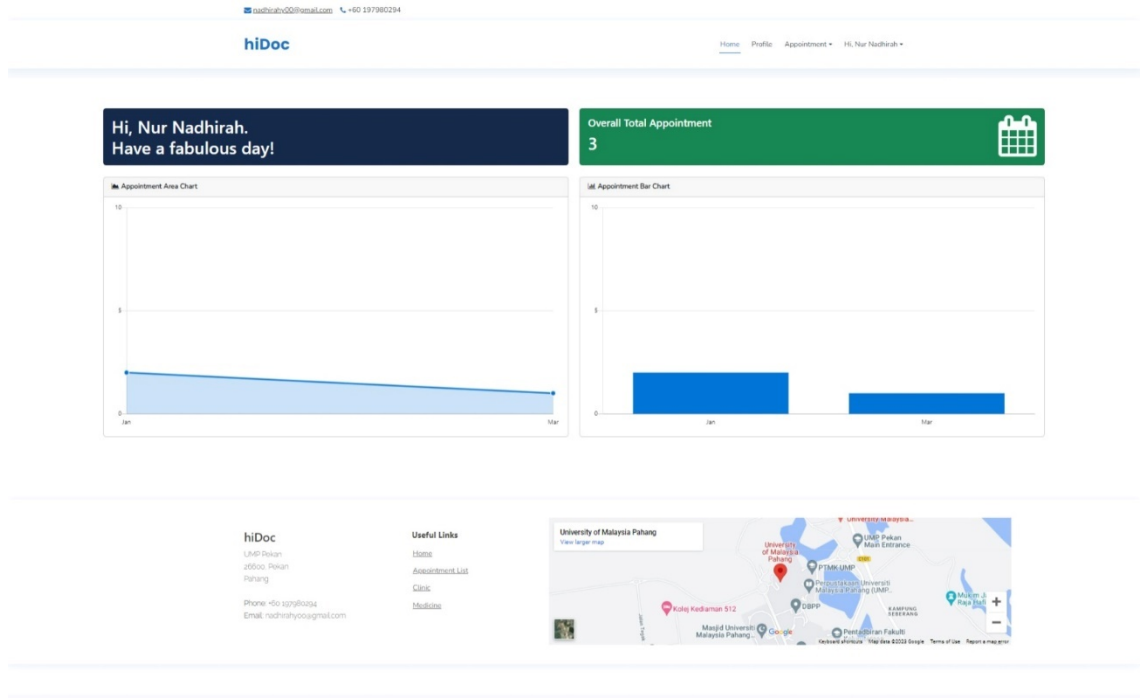


Figure 4.36 User Homepage

The user's homepage is displayed in [Figure 4.36](#). Each user page includes a unique navigation bar.



### 4.3.3.2 Manage Account Module (User Profile)

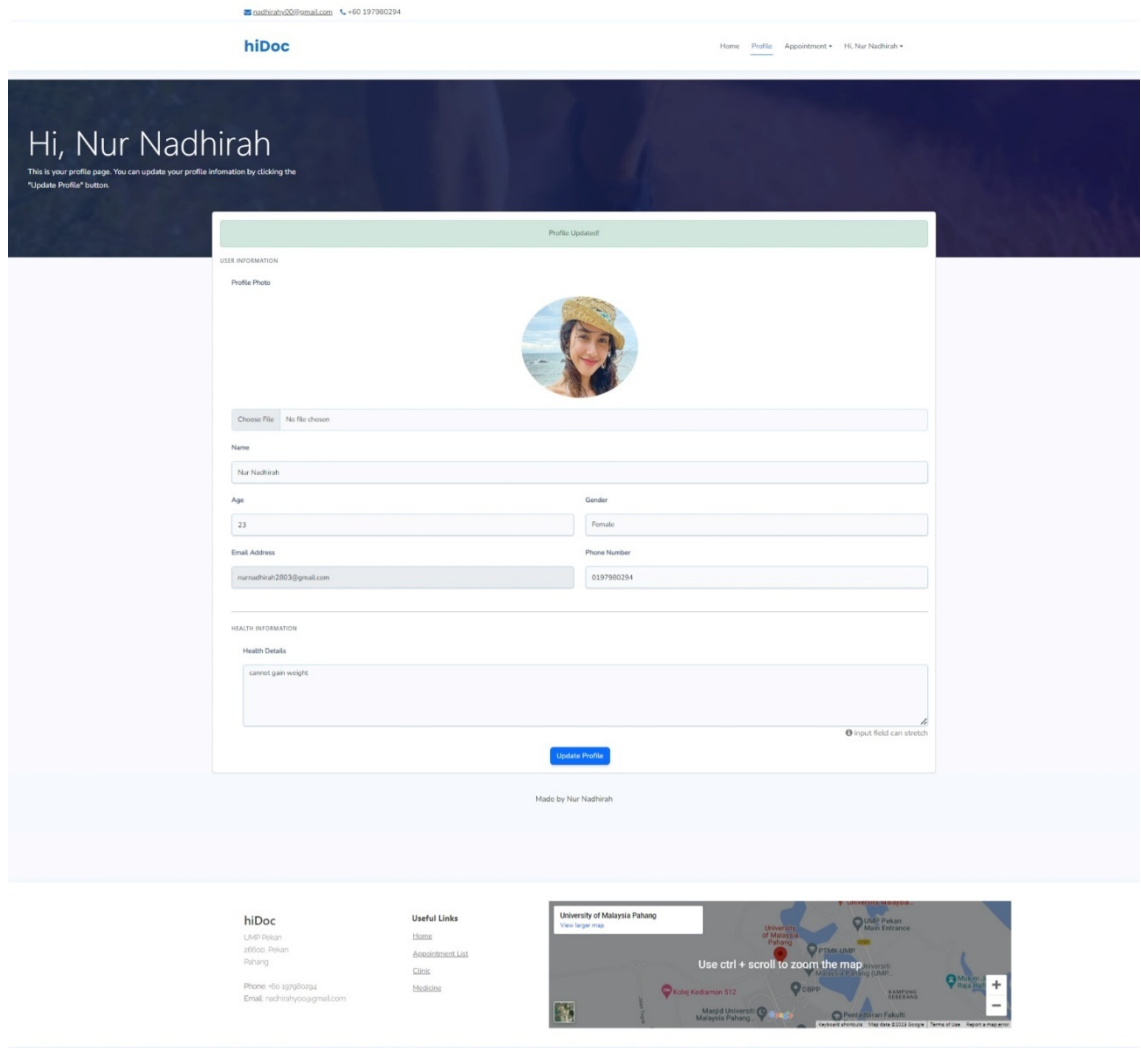


Figure 4.37 User Profile

The user profile page is depicted in the figure above. The user information on this page includes their name, email address, phone number, and any relevant health information.

### 4.3.3.3 Manage Appointment Booking Module (Make Appointment, Appointment List & History)

Appointment List

Your appointment schedule is listed starting from the latest appointment date.

#	CLINIC NAME	DOCTOR SELECTED	DATE	TIME SLOT	MESSAGE	STATUS	ATTENDANCE	ACTION	MEDICAL RECORD
1	Klinik Peramu Jaya	Dr Suraya	22 March 2023	8:30 AM - 9:00 AM	sekit batuk	Approve	Absent	Cancel	No record
2	Klinik Peramu Jaya	Dr Suraya	20 January 2023	9:00 AM - 9:30 AM	back pain	Cancel	Cancel	Cancel	No record
3	Klinik Sulaman	Dr Sophia	18 January 2023	8:00 AM - 8:30 AM	headache	Approve	Attend	Cancel	View

hiDoc  
 Klinik Peramu  
 27000 Pekan  
 Pahang  
 Phone: +60 327970294  
 Email: nadrah@yoojagmail.com

Useful Links  
 Home  
 Appointment List  
 Clinic  
 Mobile

University of Malaysia Pahang  
 View larger map

Figure 4.38 Appointment List & History

This is the interface that shows the patient's appointment schedule. The patient has the option to cancel the appointment. The patient can see whether their appointment was approved or rejected, as well as whether they attended the appointment session. If the patient attended the session, the "view" button will display a medical record from the appointment.

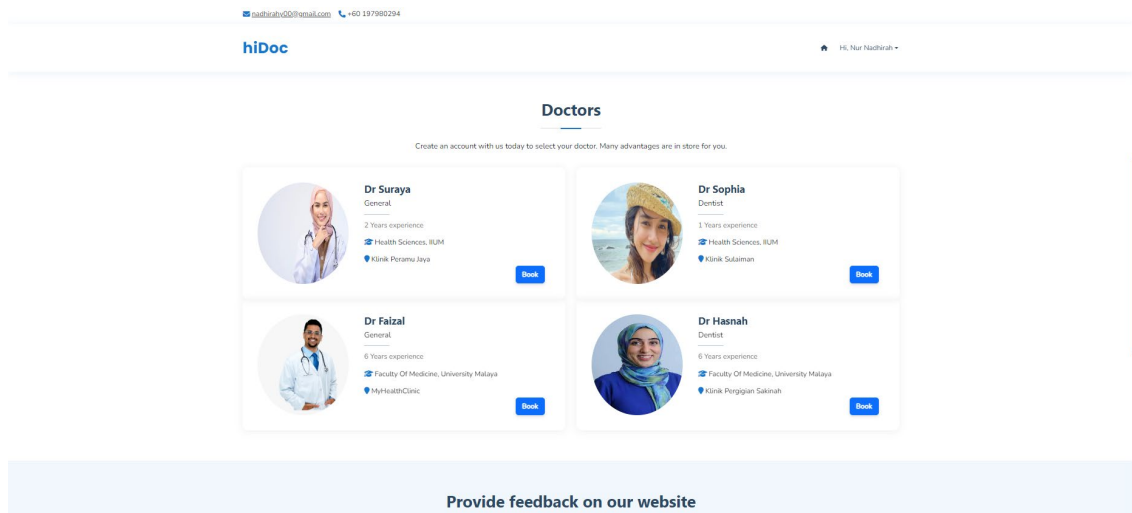


Figure 4.39 Doctor Section at Main Page

This is the list of doctors found on the main page. If the user has registered as a "User," the "Book" button will be visible to them. If they are not currently logged in or registered as a "User," the button is not visible to them. As a result, they are unable to schedule an appointment. The appointment form for this interface can be seen at Figure 4.40 below.

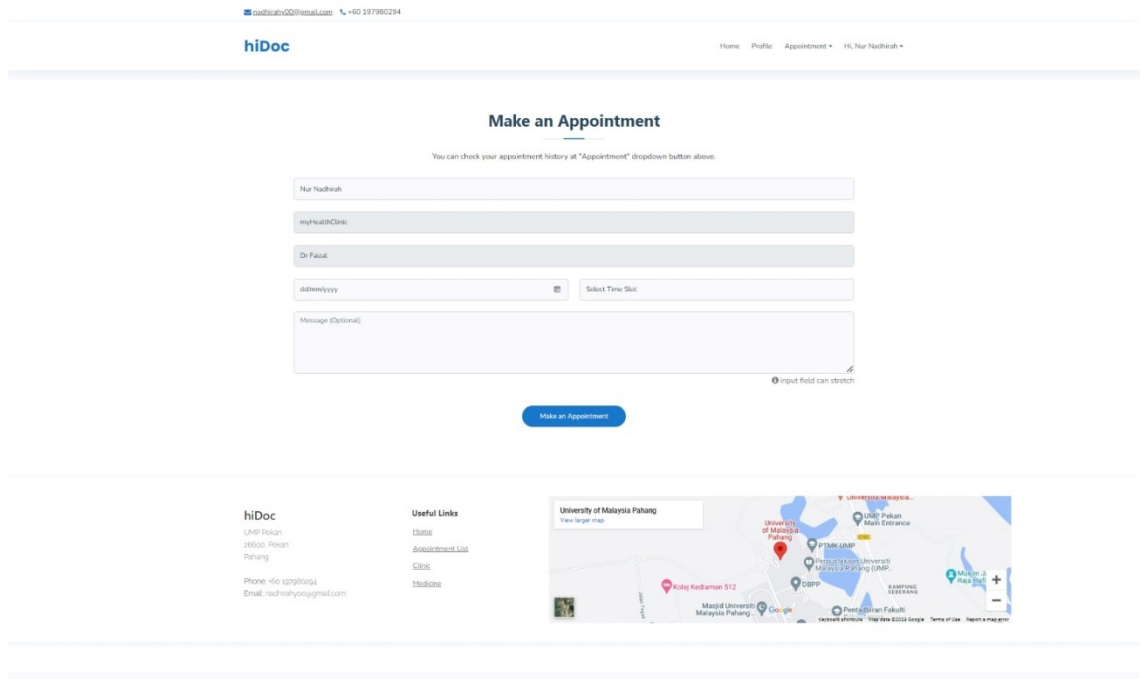


Figure 4.40 Make an Appointment with Doctor Option

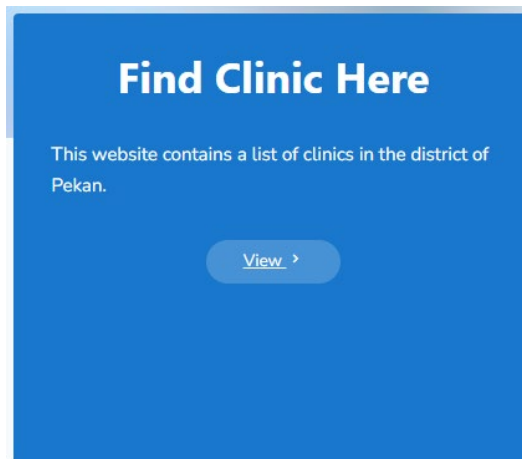


Figure 4.41 Button to view clinic list

This is a button that can be found on the main page. This button will take the user to an interface that will display a list of clinics in the district of Pekan. The clinic list interface can be seen at Figure 4.42 below.

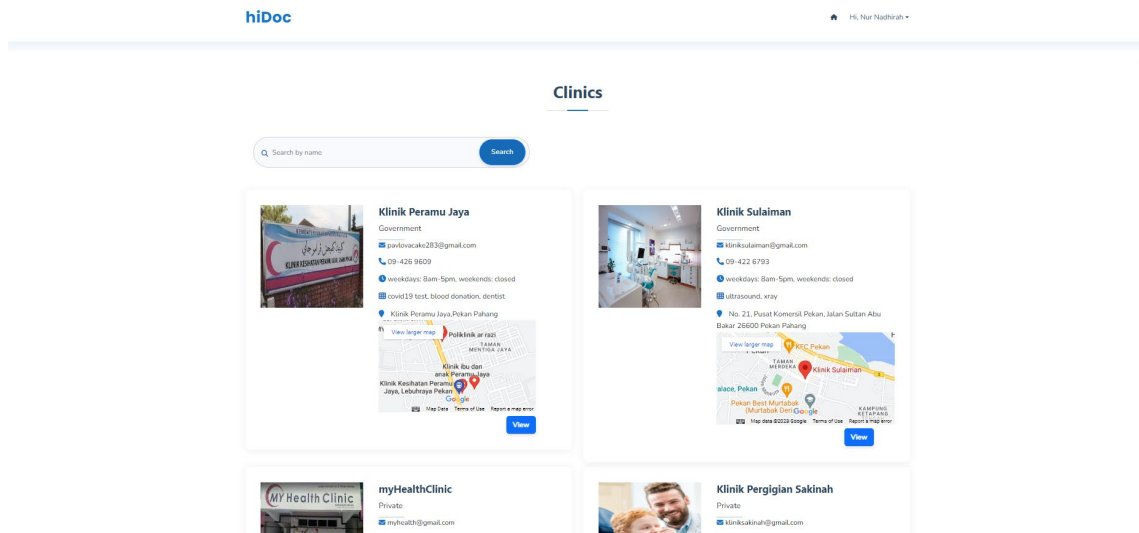


Figure 4.42 Clinic List

This page contains information about the clinic. When a user selects the "View" option. An interface displaying the clinic's doctor as well as a booking form will be displayed. This form can be seen at Figure 4.43 below. To book an appointment, the user has the option to whether choose a doctor and they must choose the date and time. Then they must wait for the approval of the booked appointment first. This status can be view at Appointment list & history interface at [Figure 4.38](#).

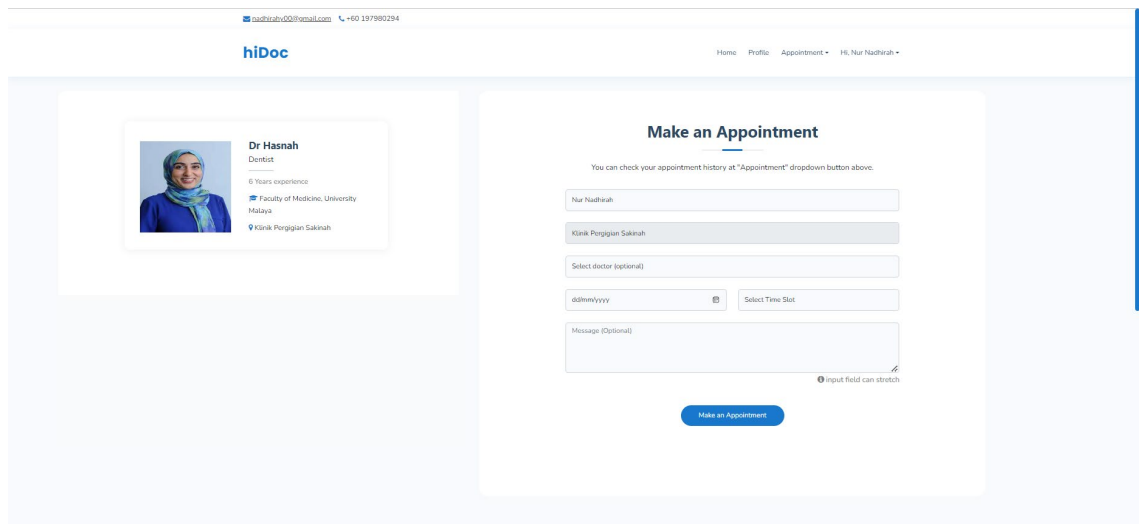


Figure 4.43 Doctor List in Each Clinic

#### 4.3.3.4 Manage Medical Records Module (View)

MEDICAL RECORD	
Patient Name	Nur Nadhirah
Clinic	Klinik Sulaiman
Doctor	Dr Sophia
Date	18 January, 2023
Timeslot	8:00 AM - 8:30 AM
Patient's Message	headache
Diagnosis	lack of sleep
Prescribed Medicine	no medicine precribed

---

Authorised Signature

---

Name

---

Date

*Generated by,*

**hiDoc**

Figure 4.44 Patient's Medical Record

The image above depicts the patient's medical record for a specific appointment. In the medical record, there is an authorised signature section for the doctor to sign. This medical record may be useful in the future if the patient needs verification for her health problems.

#### 4.4 Testing

After the development process is completed, the testing process is used to evaluate the application's usability, effectiveness, and functionality. The User Acceptance Test (UAT) Form is used to determine whether the application has been thoroughly tested. The user tests the application on a personal device (HP Laptop). The testing process was

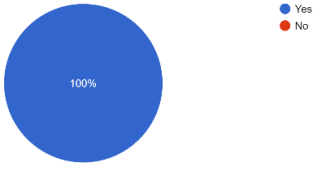
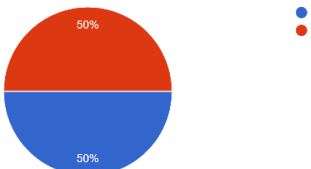
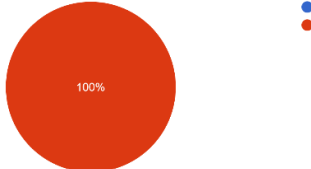
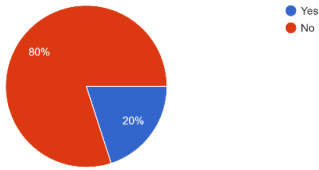
carried out on ten different users (patient, admin, clinic representative, doctor). The UAT can be seen at [APPENDIX A](#).

#### **4.5 Result and discussion**

The testing results are divided into three sections: basic questions, testing results, and tester comments. There are ten people from Pekan who are taking part in the testing. Firstly, they need to answer yes or no to the question " Do you know any website that can book an appointment with clinic that contains clinic pekan district information? Or do you know of any doctor booking appointments in Pekan?" This is evident in [Section 1](#). Second, they must test the system by answering the questions in the survey, as shown in [Section 2](#), and finally, they must write comments on system improvements or any dissatisfaction, as shown in [Section 3](#). The UAT questions can be seen at [APPENDIX A](#).

According to the first questions, 70% of those polled are unaware of any website with information on clinics in the Pekan district, while 30% are aware. And, based on the graph for testing that is available in [Section 2](#), most of the functions are successful and work perfectly, but there are a few concerns that arise during testing, such as the failure of the user to receive the cancellation email from the clinic informing the appointment cancellation and the difficulties of users in searching for a doctor. All the issues and comments from the testers have been addressed, and the hiDoc: Doctor Appointment Booking Website Application has been improved based on the testers' comments at [Section 3](#) and testing results.

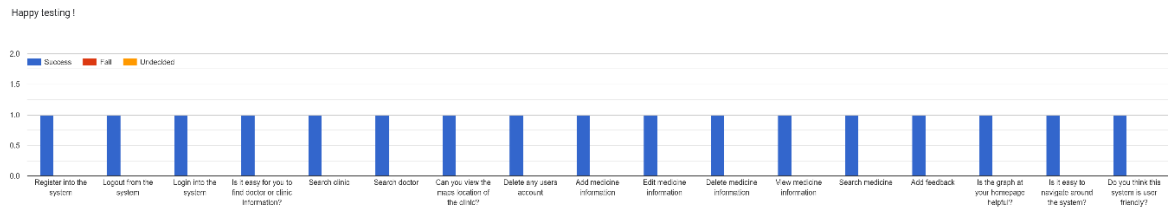
#### 4.5.1 Section 1: Basic Question

User type	Result	Total
Admin	<p>Do you know of any doctor or clinic booking appointment that contains clinic information in Pekan district?</p> <p>1 response</p>  <p>100%</p> <p>● Yes ● No</p>	<p>1 – yes</p> <p>0 - no</p>
Clinic	<p>Do you know of any doctor or clinic booking appointment that contains clinic information in Pekan district?</p> <p>2 responses</p>  <p>50%</p> <p>50%</p> <p>● Yes ● No</p>	<p>1 – yes</p> <p>1 - no</p>
Doctor	<p>Do you know of any doctor or clinic booking appointment that contains clinic information in Pekan district?</p> <p>2 responses</p>  <p>100%</p> <p>● Yes ● No</p>	<p>0 - yes</p> <p>2 - no</p>
Patient	<p>Do you know of any doctor or clinic booking appointment that contains clinic information in Pekan district?</p> <p>5 responses</p>  <p>20%</p> <p>80%</p> <p>● Yes ● No</p>	<p>1 – yes</p> <p>4 - no</p>
Summary	<p>70% of those polled are unaware of any website containing information on clinics in the district of Pekan, while 30% are aware.</p>	

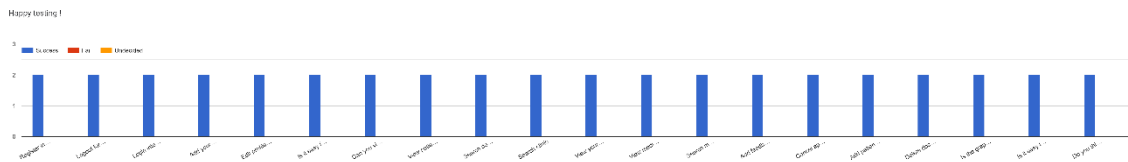


## 4.5.2 Section 2: Testing Results

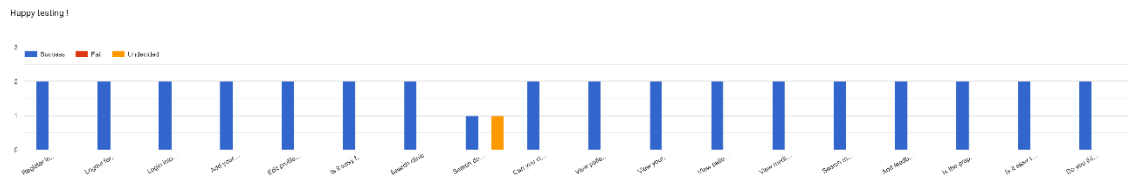
### 4.5.2.1 Admin



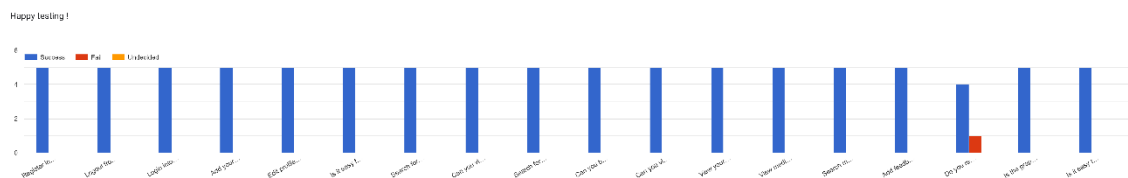
### 4.5.2.2 Clinic



### 4.5.2.3 Doctor



### 4.5.2.4 Patient



### 4.5.3 Section 3: Comments

User type	Comments
Admin	<p>Write comments to improve this system or to express your dissatisfaction.</p> <p>1 response</p> <p>Add functionality like detecting which clinic is the nearest with the user</p>
Clinic	<p>Write comments to improve this system or to express your dissatisfaction.</p> <p>2 responses</p> <p>everythings okay</p> <p>Add a comment section for each clinic</p>
Doctor	<p>Write comments to improve this system or to express your dissatisfaction.</p> <p>2 responses</p> <p>Took me a while to figure out on how to search doctor in the website.</p> <p>button view kt medicine list kene besar sikit. Tukar dari "Let's Get Started" ke "Find Clinic Here"</p>
Patient	<p>Write comments to improve this system or to express your dissatisfaction.</p> <p>5 responses</p> <p>Mungkin perlu besarkan sikit icon untuk homepage untuk user</p> <p>None</p> <p>Need to improve usability.</p> <p>Hard to find list of doctors</p> <p>Very convenient website for Pekan community.</p>

## CHAPTER 5

### CONCLUSION

#### 5.1 Introduction

Chapter 5 will summarise the findings of developing the doctor appointment booking web application for clinics in the district of Pekan, Pahang to achieve the objectives and overcome the problems stated in Chapter 1. Patients' hectic daily schedules make it difficult for them to go in person and book a medical appointment. This causes a delay in treatment and may result in a serious health problem later. Furthermore, if the patient does not have an appointment, they will have to wait for a long time if they want to see the doctor that day.

Furthermore, an application for scheduling doctor appointments for all clinics in the Pekan district has not yet been developed, and there are few existing websites that contain information on clinics in the Pekan district. This application can be a useful tool for people in Pekan who want to schedule an appointment with a clinic or doctor. As a result, hiDoc is extremely useful and convenient because it allows patients to schedule appointments at any clinic. Visual Studio Code, Xampp, and Github were used to create this application. This application was created using agile methodology, which allows for significant time and cost savings.

This application is implemented and evaluated by Pekan residents to test its effectiveness and functionality. The evaluation process yielded positive results, and this application met the application's objectives.

#### 5.2 Research Constraint

The constraint in this project is:

- i. Limitation of Time

Due to time constraints, it is not possible to develop a more high-end doctor appointment booking web application with excellent features such as a Calendar to view the appointment schedule for clinic representatives, doctors, and patients.

- ii. Inadequate requirement research

Appointment booking requirements are insufficient, such as displaying only the available slots for each day to make it easier for the patient to schedule an appointment.

### **5.3 Future Work**

There are several improvements that can be made to the Doctor Appointment Booking Web Application in the future.

- i. The Doctor Appointment Booking Web Application must be implemented with mobile platforms that support Android and iOS versions.
- ii. The developer must add more features, such as online consultation. As a result, patients can consult with the doctor without having to meet in person.
- iii. The developer must implement a date and time filtering to allow the system to display only times that have not yet been booked.
- iv. The developer must implement features to detect the distance between the clinic's location and the user's current location.
- v. The developer opens the system to other districts and implements a district-based filtering search.

## REFERENCES

- Al-Shaher, M. A., & Abdul-Wahed, A. Q. (2020). Design and Implementation of Android Application to Thi-Qar Doctors Guide and Resident Pharmacies. *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, 659–664. <https://doi.org/10.1109/ICECA49313.2020.9297498>
- Chafekar, H., Sundas, A., & Sharma, S. (2021). Health Related Web Application: Virtual Hospital. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*. <https://doi.org/10.1109/ICRITO51393.2021.9596162>
- Devi, S. S., Deepica, J. S., Dharshini, K., & Dhivyashree, G. (2021). User Interactive Hospital Management System by using Web application. *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, 1578–1585. <https://doi.org/10.1109/ICESC51422.2021.9532887>
- Singh, A., Kaur, I. G. P., & Dabas, C. (2018). Get-a-Doc: A Doctor Recommender System. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 219–223. <https://doi.org/10.1109/ICRITO.2018.8748776>
- N. Yadav, D. S. Rajpoot and S. K. Dhakad, "LARAVEL: A PHP Framework for E-Commerce Website," *2019 Fifth International Conference on Image Information Processing (ICIIP)*, Shimla, India, 2019, pp. 503-508, doi: 10.1109/ICIIP47207.2019.8985771.
- Yu, Xiaosheng, and Cai Yi. "Design and Implementation of the Website Based on PHP & MYSQL." *2010 International Conference on E-Product E-Service and E-Entertainment*. IEEE, 2010.

Altameem, E. (2015). Impact of agile methodology on software development. *Computer and Information Science*, 8(2). <https://doi.org/10.5539/cis.v8n2p9>

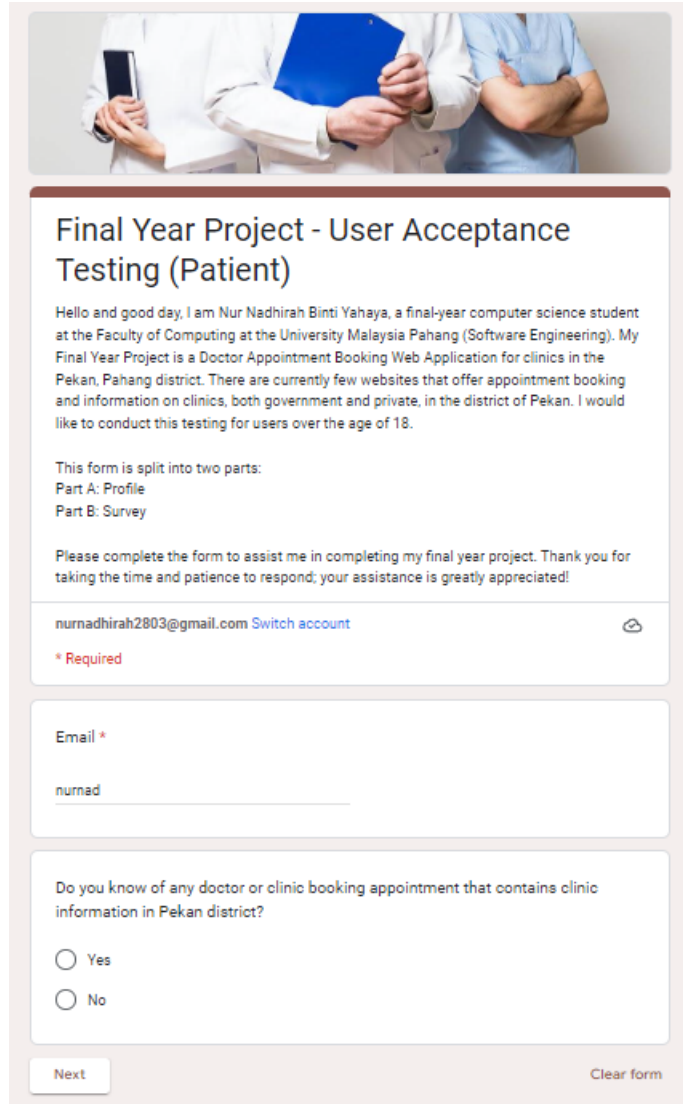
Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies (IJIM)*, 14(11), 246. <https://doi.org/10.3991/ijim.v14i11.13269>

*APPLICATION OF THE SECURE HASHING ALGORITHM (SHA) METHOD AND ROLE-BASED ACCESS SECURITY AUTHENTICATION IN LARAVEL 7 | Mary. | jurnal Edik Informatika penelitian bidang komputer sains dan pendidikan Informatika.* (n.d.). Upgrisba.Ac.Id. Retrieved January 21, 2023, from <https://ejournal.upgrisba.ac.id/index.php/eDikInformatika/article/view/4875>

U. Chauhan, H. Jha, D. Singh and S. P. S. Chauhan, "Doctor Finder and Appointment Booking Website using DJANGO," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 397-400, doi: 10.1109/ICIPTM54933.2022.9753977.

## APPENDIX A: USER ACCEPTANCE TEST

### BASIC QUESTIONS FOR ALL USERS



**Final Year Project - User Acceptance Testing (Patient)**

Hello and good day, I am Nur Nadhirah Binti Yahaya, a final-year computer science student at the Faculty of Computing at the University Malaysia Pahang (Software Engineering). My Final Year Project is a Doctor Appointment Booking Web Application for clinics in the Pekan, Pahang district. There are currently few websites that offer appointment booking and information on clinics, both government and private, in the district of Pekan. I would like to conduct this testing for users over the age of 18.

This form is split into two parts:  
Part A: Profile  
Part B: Survey

Please complete the form to assist me in completing my final year project. Thank you for taking the time and patience to respond; your assistance is greatly appreciated!

nurnadhirah2803@gmail.com [Switch account](#)

\* Required

Email \*

nurnad

Do you know of any doctor or clinic booking appointment that contains clinic information in Pekan district?

Yes

No

Next Clear form

Patient UAT, Admin UAT, Doctor UAT, and Clinic UAT will be the four groups for the User Acceptance Test (UAT). Every group has its own Google form. The basic information that is shared by all Google forms is depicted in the figure above. The user must enter their email address, gender, age, and comments about the system.

## PATIENT – GOOGLE FORM

Happy testing ! *			
	Success	Fail	Undecided
Register into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logout from the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Login into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add your profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Edit profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy for you to find doctor or clinic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search for doctor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you view the maps location of the clinic?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search for clinic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you book an appointment?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you view the list of doctor in the chosen clinic?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View your appointment history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search medicine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Do you receive any appointment cancellation email? [If appointment is cancel by the clinic]	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is the graph at your homepage helpful?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy to navigate around the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



## ADMIN – GOOGLE FORM

Admin Survey			
Happy testing ! *			
	Success	Fail	Undecided
Register into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logout from the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Login into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy for you to find doctor or clinic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search clinic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search doctor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you view the maps location of the clinic?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delete any users account	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Edit medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delete medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search medicine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is the graph at your homepage helpful?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy to navigate around the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Do you think this system is user friendly?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## CLINIC – GOOGLE FORM

	Success	Fail	Undecided
Register into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logout form the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Login into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add your profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Edit profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy for you to find doctor or clinic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you view the maps location of the clinic?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View patient information at appointment history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search doctor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search clinic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View your appointment history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search medicine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cancel appointment	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add patient's diagnosis if the attendance is 'Attend'	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delete doctor account	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is the graph at your homepage helpful?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy to navigate around the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Do you think this system is user friendly?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## DOCTOR – GOOGLE FORM

Doctor Survey			
Happy testing ! *			
	Success	Fail	Undecided
Register into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logout form the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Login into the system	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add your profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Edit profile information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy for you to find doctor or clinic information?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search clinic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search doctor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Can you view the maps location of the clinic?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View patient information at appointment history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View your appointment history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View patient's medical record if available	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
View medicine information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Search medicine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Add feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is the graph at your homepage helpful?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Is it easy to navigate around the system?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Do you think this system is user friendly?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## **APPENDIX B: SRS**

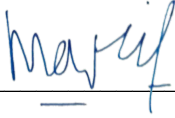
2023

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

DOCTOR APPOINTMENT BOOKING WEB  
APPLICATION



## DOCUMENT APPROVAL

	Name	Date
<b>Authenticated by:</b>  _____ Muhammad' Arif Bin Mohamad	Muhammad' Arif Bin Mohamad	09/02/2023
<b>Approved by:</b> _____ Client		

Software :

Archiving Place :

## TABLE OF CONTENT

CONTENT	PAGE
<b>DOCUMENT APPROVAL .....</b>	<b>2</b>
<b>TABLE OF CONTENT.....</b>	<b>3</b>
<b>LIST OF FIGURES .....</b>	<b>5</b>
<b>LIST OF TABLES .....</b>	<b>7</b>
<b>1.1 PROJECT DESCRIPTION.....</b>	<b>8</b>
<b>1.2 SYSTEM IDENTIFICATION .....</b>	<b>9</b>
<b>1.3 CONTEXT DIAGRAM .....</b>	<b>10</b>
<b>1.4 DATA FLOW DIAGRAM.....</b>	<b>11</b>
<b>2.1 USE CASE DIAGRAM AND DESCRIPTION .....</b>	<b>1</b>
2.1.1 Manage Account.....	1
2.1.2 Manage Appointment Booking .....	3
2.1.3 Manage Medical Record.....	6
2.1.4 Manage Medicine Information .....	8
2.1.5 Manage Clinic.....	11
<b>2.2 SEQUENCE DIAGRAM .....</b>	<b>15</b>
2.2.1 Manage Account .....	15
2.2.2 Manage Appointment Booking .....	2
2.2.3 Manage Medical Record .....	3
2.2.4 Manage Medicine Information.....	4
2.2.5 Manage Clinic.....	5
<b>3.1 INTERFACE DESIGN .....</b>	<b>6</b>
3.1.1 ALL USERS CAN ACCESS .....	6
3.1.1.1 Main Page .....	6
3.1.1.2 Register .....	8
3.1.1.3 Manage Clinic Module (Clinic List).....	9
3.1.1.4 Manage Medicine Module (Medicine List).....	10
3.1.2 ADMIN INTERFACE .....	11

3.1.2.1	Homepage .....	11
3.1.2.2	Manage Account Module (User Management).....	12
3.1.2.3	Manage Medicine Module (Medicine Management) .....	13
3.1.3	CLINIC INTERFACE.....	14
3.1.3.1	Get Started .....	14
3.1.3.2	Homepage .....	15
3.1.3.3	Manage Account Module (Doctor Management) .....	16
3.1.3.4	Manage Account Module (Clinic Profile) .....	17
3.1.3.5	Manage Appointment Booking Module (Appointment List & History) .....	18
3.1.3.6	Manage Medical Record Module (Add, View, Delete).....	19
3.1.4	DOCTOR INTERFACE .....	21
3.1.4.1	Get Started .....	21
3.1.4.2	Homepage .....	22
3.1.4.3	Manage Account Module (Doctor Profile).....	23
3.1.4.4	Manage Appointment Booking Module (Appointment List & History).....	24
3.1.4.5	Manage Medical Record Module (View) .....	25
3.1.5	PATIENT INTERFACE .....	26
3.1.5.1	Homepage .....	26
3.1.5.2	Manage Account Module (User Profile).....	27
3.1.5.3	Manage Appointment Booking Module (Make Appointment & Appointment History) 28	28
3.1.5.4	Manage Medical Record Module (View) .....	29
<b>3.2</b>	<b>HARDWARE AND SOFTWARE SPECIFICATION.....</b>	<b>30</b>



**LIST OF FIGURES**

Figure 1. 1 Context Diagram .....	10
Figure 1. 2 Data Flow Diagram .....	11
Figure 1. 3 Manage Account Use Case Diagram.....	1
Figure 1. 4 Manage Appointment Booking Use Case Diagram .....	4
Figure 1. 5 Manage Medical Record Use Case Diagram.....	7
Figure 1. 6 Manage Medicine Information Use Case Diagram.....	9
Figure 1. 7 Manage Clinic Use Case Diagram .....	12
<b>Figure 1. 8 Manage Account Sequence Diagram .....</b>	<b>15</b>
Figure 1. 9 Manage Appointment Booking Sequence Diagram.....	2
Figure 1. 10 Manage Medical Record Sequence Diagram .....	3
Figure 1. 11 Manage Medicine Information Sequence Diagram.....	4
Figure 1. 12 Manage Clinic Sequence Diagram .....	5
Figure 1. 13 Main Page Interface.....	6
Figure 1. 14 Main Page – User Interface .....	7
Figure 1. 15 Login Interface .....	8
Figure 1. 16 Register Interface.....	8
Figure 1. 17 Clinic List Interface.....	9
Figure 1. 18 Medicine List Interface.....	10
Figure 1. 19 Admin – Homepage Interface .....	11
Figure 1. 20 User Management Interface .....	12
Figure 1. 21 Medicine Management Interface .....	13
Figure 1. 22 Get Started - Clinic Interface.....	14
Figure 1. 23 Clinic – Homepage Interface.....	15
Figure 1. 24 Doctor Management Interface.....	16
Figure 1. 25 Clinic Profile Interface .....	17
Figure 1. 26 Appointment List Interface.....	18
Figure 1. 27 Medical Record Interface .....	19

---

Figure 1. 28 Medical Record Interface .....	20
Figure 1. 29 Get Started - Doctor Interface .....	21
Figure 1. 30 Doctor – Homepage Interface .....	22
Figure 1. 31 Doctor Profile Interface .....	23
Figure 1. 32 Appointment List Doctor Interface .....	24
Figure 1. 33 Medical Record Interface .....	25
Figure 1. 34 Patient - Homepage .....	26
Figure 1. 35 Patient Profile .....	27
Figure 1. 36 Appointment List Patient Interface .....	28
Figure 1. 37 Medical Record Interface .....	29

**LIST OF TABLES**

Table 1. 1 Manage Account Use Case Description .....	1
Table 1. 2 Manage Appointment Booking Use Case Description .....	4
Table 1. 3 Manage Medical Record Use Case Description .....	7
Table 1. 4 Manage Medicine Information Use Case Description.....	9
Table 1. 5 Manage Clinic Use Case Description .....	12
Table 1. 6 Hardware Specification.....	30
Table 1. 7 Software Specification.....	30

## CHAPTER 1

### 1.1 PROJECT DESCRIPTION

Doctor Appointment Booking is a web-based project in which patients can book an appointment online from any clinic in the Pekan district. There is a list of doctors with their information. Patients can then select the doctor they want to see by providing the preferred date and time.

The admin, doctor, clinic staff or representative, and patient will be the four users of this project. Manage account, manage appointment booking, manage medical record, manage medicine information, and manage clinic are the five modules in this system.

Users can access and change their profiles in the Manage Account module. On the other hand, the admin can delete any existing doctor accounts. Users can view the clinic and doctor information as well.

The Manage Appointment Booking module allows the patient to schedule an appointment with the clinic or doctor that they want. The doctor, clinic staff and patient can view their appointment history records. Furthermore, the doctor has the authority to update the attendance status while the clinic staff can also update the attendance status as well as the appointment request.

The manage medical record module can be access by the patient, clinic staff and doctor. The clinic staff will add the medical record only if the patient's attend the appointment session. The clinic staff can also view and delete the medical record. The doctor and patient can view the medical record of each appointment session.

In manage medicine information module, the admin can add a new medicine, view existing medicine, and delete any drug, while the other users can view and search about the medicine.

The manage clinic module is the final module. This is where the clinic information will be displayed at. All the users can view the clinic information. In this module also, the clinic

staff can remove the doctor account that works under them while the doctor can change their workplace details. This module is connected to the manage appointment booking module. This module will allow the users to proceed to make an appointment.

## 1.2 SYSTEM IDENTIFICATION

System Title: Doctor Appointment Booking Web Application

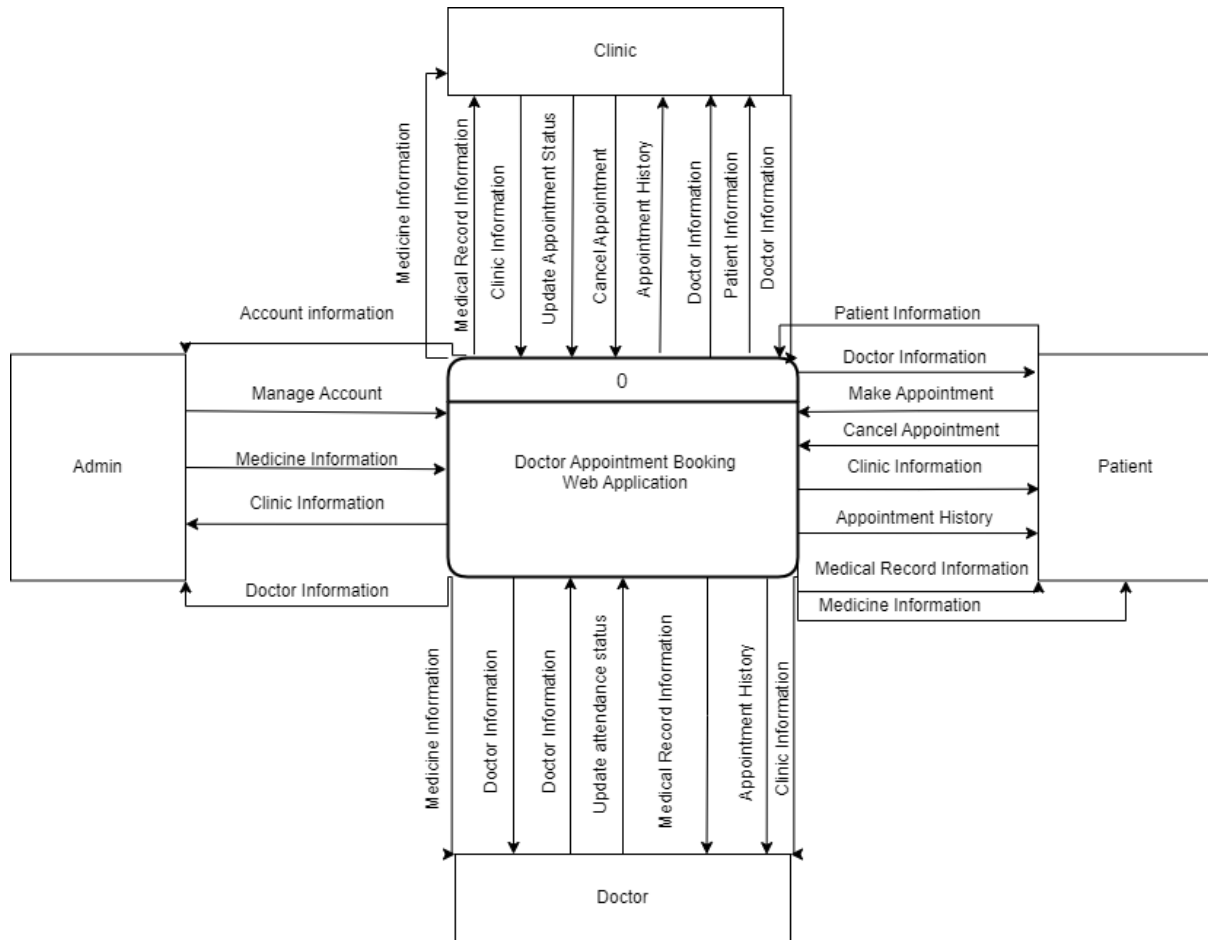
System Abbreviation: hiDoc

Year: 2023

Version: Version 1

System Identification Number: SRS-hiDoc-2023-V1

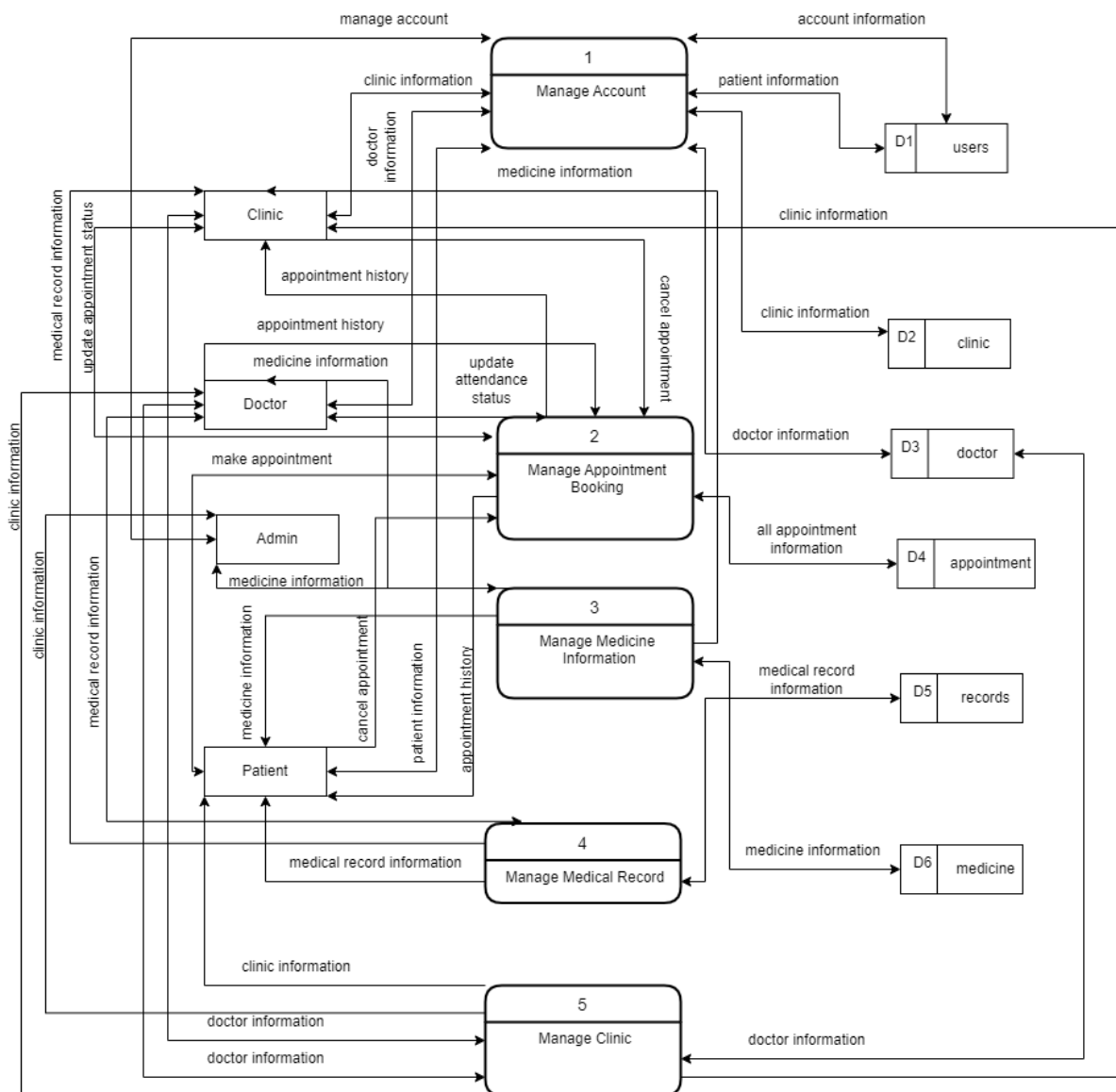
### 1.3 CONTEXT DIAGRAM



**Figure 1. 1 Context Diagram**

The context diagram in the figure above contains four users which are admin, clinic staff or representative, doctor, and patient. The context diagram is developed based on five modules which are manage account, manage appointment booking, manage medical record, manage medicine information, and manage clinic.

## 1.4 DATA FLOW DIAGRAM



**Figure 1. 2 Data Flow Diagram**

There are four external entities in the diagram above: the admin, doctor, patient, clinic staff. The admin, doctor, clinic staff, and patient are the users of this project. There are five processes in the diagram above which are the manage account, manage appointment booking, manage medical record, manage medicine information, and manage clinic. These processes are the modules in this project. There are also seven data stores which are the users, clinic, doctor, appointment, records, medicine, and feedback.

For the manage account process, patient, clinic, and doctor enter their information. The doctor's information will be stored inside the doctor data store (D3). The patient and clinic can also submit their information which will be stored inside the patient and admin data store respectively (D1,D2). Each of these users can view each other information. But only the patient can view the information for the clinic and doctor. The clinic and doctor can only view the patient's information if the patient make an appointment with the clinic or doctor.

For manage appointment booking process, the patient makes an appointment. The clinic staff will update the appointment request by accepting or rejecting the request and updating the attendance status of the patient. The doctor can also update the attendance status of patient to the appointment session. The appointment information will be stored inside the data store (D4). The clinic staff, doctor, and patient can view their appointment history records.

For manage medical records process, the clinic staff submits the diagnosis and then the patient, doctor and clinic staff can view the medical record. The clinic staff can also delete the medical record. The medical records are stored inside records data store (D5).

For manage medicine information, the admin submits medicine information. The users can view and search the medicine information. The admin can add, edit and delete the medicine. The medicine information will be stored inside data store medicine (D6).

For manage clinic process, the clinic staff can remove the doctor account that works under them while the doctor can change their workplace details. The clinic information is stored inside the data store doctor (D3).



## CHAPTER 2

### 2.1 USE CASE DIAGRAM AND DESCRIPTION

#### 2.1.1 Manage Account

The admin, clinic representative, doctor, and patient can all access the Manage Account module. The clinic representative, the patient, and the doctor can all see and edit their profile information. The administrator can only delete and view any existing account. This module is explained in greater detail in the use case diagram and description that follow:

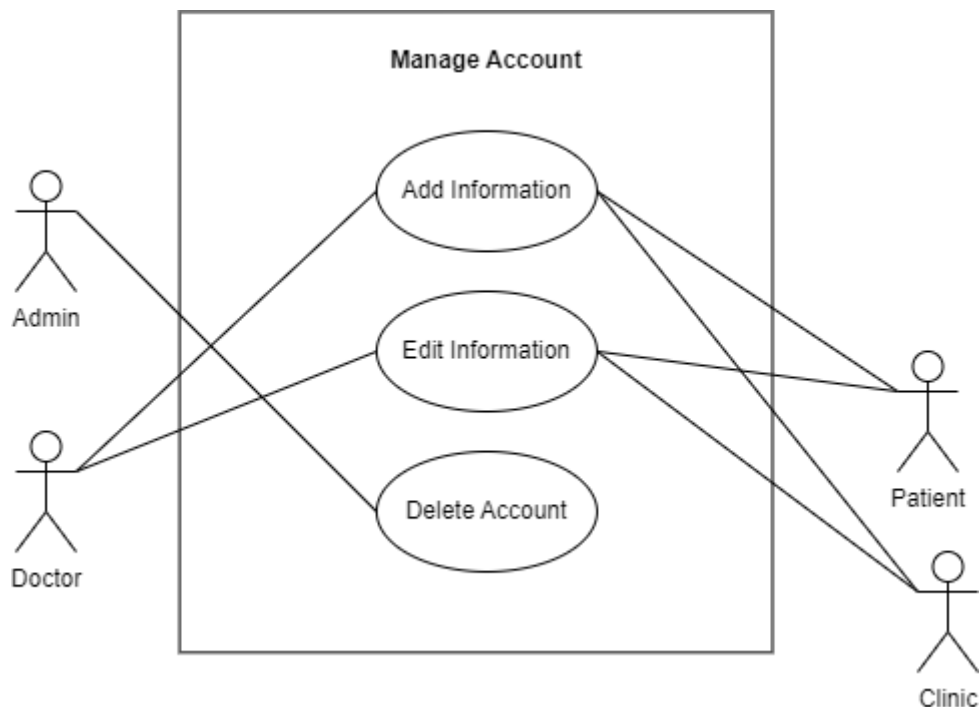


Figure 1. 3 Manage Account Use Case Diagram

Table 1. 1 Manage Account Use Case Description

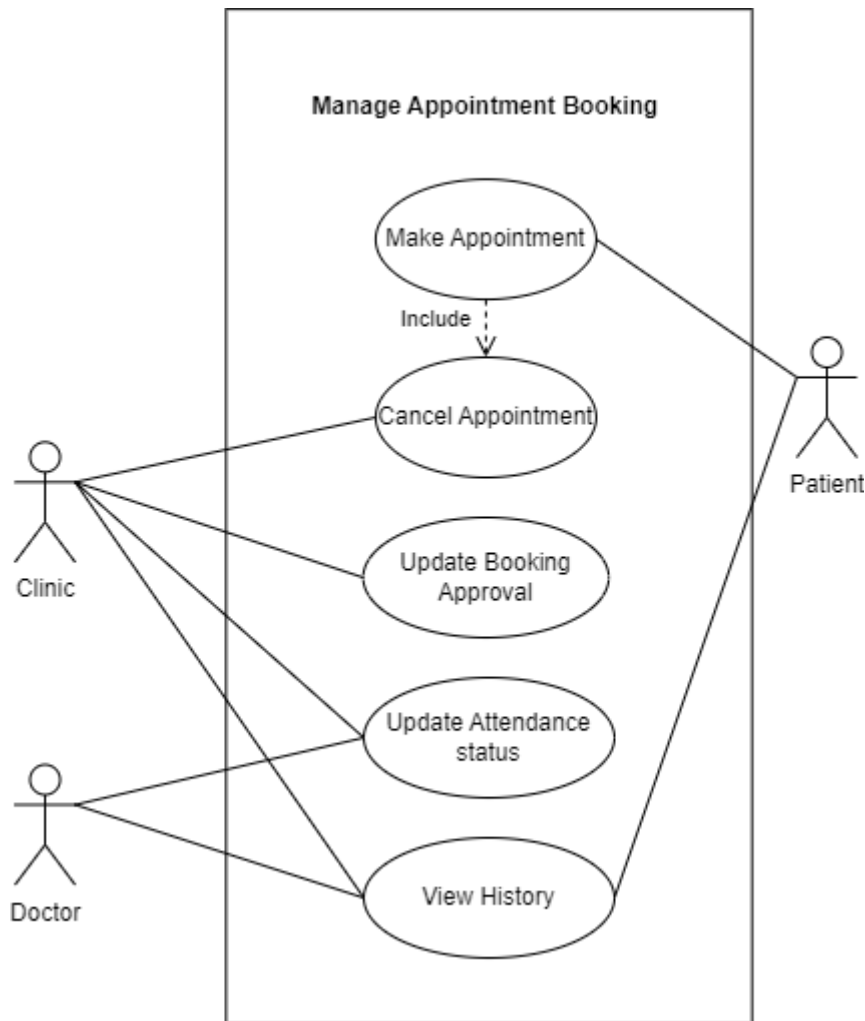
Use case name	Manage Account
Brief description	This use case is to manage the user's account and its information.
Actor	Admin, Doctor, Patient, Clinic Representative
Pre-condition	Users already have an account.
Basic flow	<b>1.</b> Doctor and clinic representative can enter their information. [E1: <b>Add Information</b> ]

	<ol style="list-style-type: none"> <li>2. Patient, doctor, and clinic can update their profile information [E2: Edit Information]</li> <li>3. System allows the admin to delete any account. [E3: Delete Account]</li> <li>4. The use case ends.</li> </ol>
Alternative flow	None
Exception flow	<p><b>E1: Add Information</b></p> <ol style="list-style-type: none"> <li>1. Doctor and clinic representative clicks the “Let’s Get Started” button.</li> <li>2. System allows doctors, and clinic representative to add information.</li> <li>3. Doctor enters gender, phone number, specialization, years of working, education background, image and workplace.</li> <li>4. Clinic representative enters contact number, address, clinic type, operating hours, image, and services that they offered.</li> <li>5. Doctor or clicks click the “Save” button.</li> <li>6. Continue with step number 1 from the basic flow.</li> </ol> <p><b>E2: Edit Information</b></p> <ol style="list-style-type: none"> <li>1. Doctor, patient, or admin clicks the “Profile” button.</li> <li>2. System allows doctor, patient, and admin to edit any information.</li> <li>3. Doctor, patient, and admin click the “Update Profile” button.</li> <li>4. Continue with step number 3 from the basic flow.</li> </ol> <p><b>E3: Delete Account</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks the “User Management” button.</li> <li>2. Admin clicks the “Delete” button on the profile that he wants to delete.</li> <li>3. System allows the admin to delete the any account.</li> </ol>

	<ol style="list-style-type: none"><li>4. System displays a pop-up message “Are you sure you want to delete this data?”</li><li>5. Admin clicks the “OK” or “Cancel” button.</li><li>6. Continue with step number 4 from the basic flow.</li></ol>
Constraints	None

### 2.1.2 Manage Appointment Booking

The manage appointment booking module is accessible to the doctor, clinic representative and patient. The doctor has the authority to update the attendance status of patient while the clinic representative can approve or reject appointment requests from patient and update the attendance status. The doctor and the patient might also cancel a previously scheduled appointment. The appointment history records can be seen by the three users. This module is explained in greater detail in the use case diagram and description that follow:



**Figure 1. 4 Manage Appointment Booking Use Case Diagram**

**Table 1. 2 Manage Appointment Booking Use Case Description**

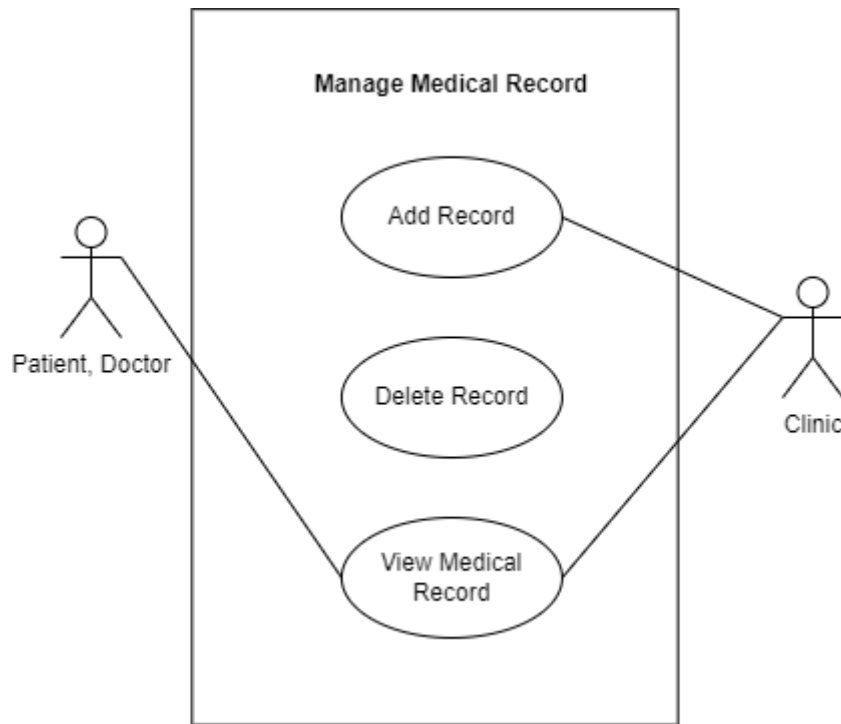
Use case name	Manage appointment booking
Brief description	This use case is to manage appointment bookings from the patients.
Actor	Doctor, Patient, Clinic representative
Pre-condition	Patient, doctors, and clinic representative need to have an account to use this module.
Basic flow	<ol style="list-style-type: none"> <li>1. Patient can make an appointment after clicking “View” or “Book”.             <ol style="list-style-type: none"> <li>a) Make an appointment [<b>E1: Book appointment</b>]</li> <li>b) Cancel appointment [<b>E2: Cancel</b>]</li> </ol> </li> <li>2. Patient, doctor, and clinic representative can view their appointment list, latest and past by</li> </ol>

	<p>clicking appointment button for doctor and clinic while patient clicks the “Appointment History” button.</p> <ol style="list-style-type: none"> <li>3. Clinic representative views appointment requests. <ol style="list-style-type: none"> <li>a) Accepts or rejects the appointment request. <b>[E3: Appointment Request]</b></li> <li>b) Send email cancellation to cancel appointment <b>[E4: Cancellation Email]</b></li> </ol> </li> <li>4. Doctor and clinic representative update the patient’s attendance for the appointment session. <b>[E5: Update attendance]</b></li> <li>5. The use case ends.</li> </ol>
Alternative flow	None
Exception flow	<p><b>E1: Book appointment</b></p> <ol style="list-style-type: none"> <li>1. Patient clicks “Book” button at doctor section in main page or click “View’ button at clinic list page.</li> <li>2. System allows patient to book an appointment with any doctor or clinic.</li> <li>3. Patient enters name, date, timeslot, choose doctor name and message about the appointment.</li> <li>4. Patient clicks “Make Appointment” button.</li> <li>5. Continue with step number 1 from the basic flow.</li> </ol> <p><b>E2: Cancel booking</b></p> <ol style="list-style-type: none"> <li>1. Patient clicks the “Cancel” button.</li> <li>2. System allows patients to cancel appointment.</li> <li>3. System displays “Are you sure you want to cancel?”</li> <li>4. Patient or doctor click “OK” or “Cancel”.</li> <li>5. Continue with step number 2 from the basic flow.</li> </ol> <p><b>E3: Appointment Request</b></p> <ol style="list-style-type: none"> <li>1. Clinic representative clicks the “Update” button at appointment list page.</li> </ol>

	<ol style="list-style-type: none"> <li>2. System allows the clinic representative to approve or reject the appointment request and the appointment attendance of patient.</li> <li>3. Choose between the option available.</li> <li>4. Clicks the “Update” button.</li> <li>5. Continue with step number 2 from the basic flow.</li> </ol> <p><b>E4: Cancellation Email</b></p> <ol style="list-style-type: none"> <li>1. Clinic representative clicks the “Send Email” button at appointment list page.</li> <li>2. System allows patients to an email to the patient to inform the appointment cancellation.</li> <li>3. System displays “Are you sure you want to cancel?”</li> <li>4. Click “OK” or “Cancel”.</li> <li>5. Continue with step number 3 from the basic flow.</li> </ol> <p><b>E5: Update attendance</b></p> <ol style="list-style-type: none"> <li>1. Doctor clicks “Attend” or “Absent” button at appointment list page.</li> <li>2. System allows doctor to update the attendance status.</li> <li>3. Continue with step number 5 from the basic flow.</li> </ol>
Constraints	None

### 2.1.3 Manage Medical Record

The patient, clinic staff and doctor have access to this module. The clinic staff can add medical record information from the appointment session between the patient and doctor. The doctor and patient can view the medical record. The clinic staff can also view and delete the medical record. The use case diagram and description that follow go over this module in greater depth:



**Figure 1. 5 Manage Medical Record Use Case Diagram**

**Table 1. 3 Manage Medical Record Use Case Description**

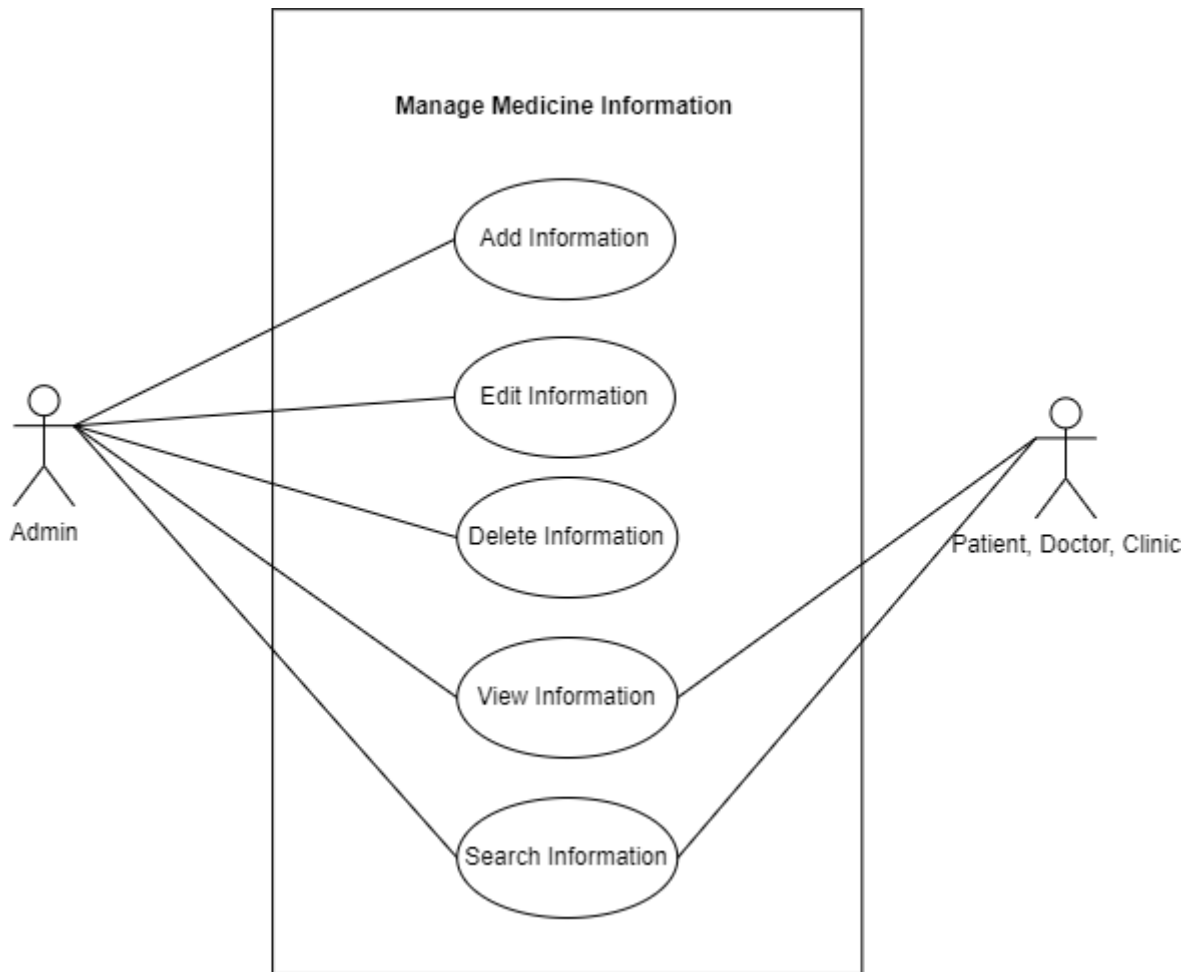
Use case name	Manage Medical Record
Brief description	This use case is to manage medical record of the patient.
Actor	Patient, Doctor, and Clinic representative
Pre-condition	<ol style="list-style-type: none"> <li>1. The patient must have an account to use this module.</li> <li>2. Patients attend the appointment session.</li> </ol>
Basic flow	<ol style="list-style-type: none"> <li>1. At appointment list page, clinic representative clicks the 'Add' button. <b>[E1: Add Record]</b></li> <li>2. Clinic representative clicks the "Delete" button. <b>[E2: Delete Record]</b></li> <li>3. The patient, doctor and clinic representative can view the records. <b>[E3: View Record]</b></li> <li>4. The use case end.</li> </ol>
Alternative flow	None

Exception flow	<p><b>E1: Add Record</b></p> <ol style="list-style-type: none"> <li>1. Clinic representative clicks “Add” button.</li> <li>2. System allows the clinic representative to input the details.</li> <li>3. Then input the diagnosis from the appointment session between the patient and the doctor with the prescribed medicine.</li> <li>4. Click “Save”</li> <li>5. Continue with step number 2 from the basic flow.</li> </ol> <p><b>E2: Delete Record</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks the “Delete” button.</li> <li>2. System allows the admin to delete the record.</li> <li>3. System displays a pop-up message “Are you sure to delete this data?”.</li> <li>4. Admin clicks the “OK” or “Cancel” button.</li> <li>5. Continue with step number 3 from the basic flow.</li> </ol> <p><b>E3: View Record</b></p> <ol style="list-style-type: none"> <li>1. Patient, clinic representative and doctor click the “view” button.</li> <li>2. System allows the three users to view the record details.</li> <li>3. Continue with step number 4 from the basic flow.</li> </ol>
Constraints	None

#### 2.1.4 Manage Medicine Information

Only the admin can manage medicine information section. The administrator can add new medicines, modify their details, and delete the medicine. All the users can search for the medicine at the main page of the system. This module is covered in greater detail in the use case diagram and description that follow:





**Figure 1. 6 Manage Medicine Information Use Case Diagram**

**Table 1. 4 Manage Medicine Information Use Case Description**

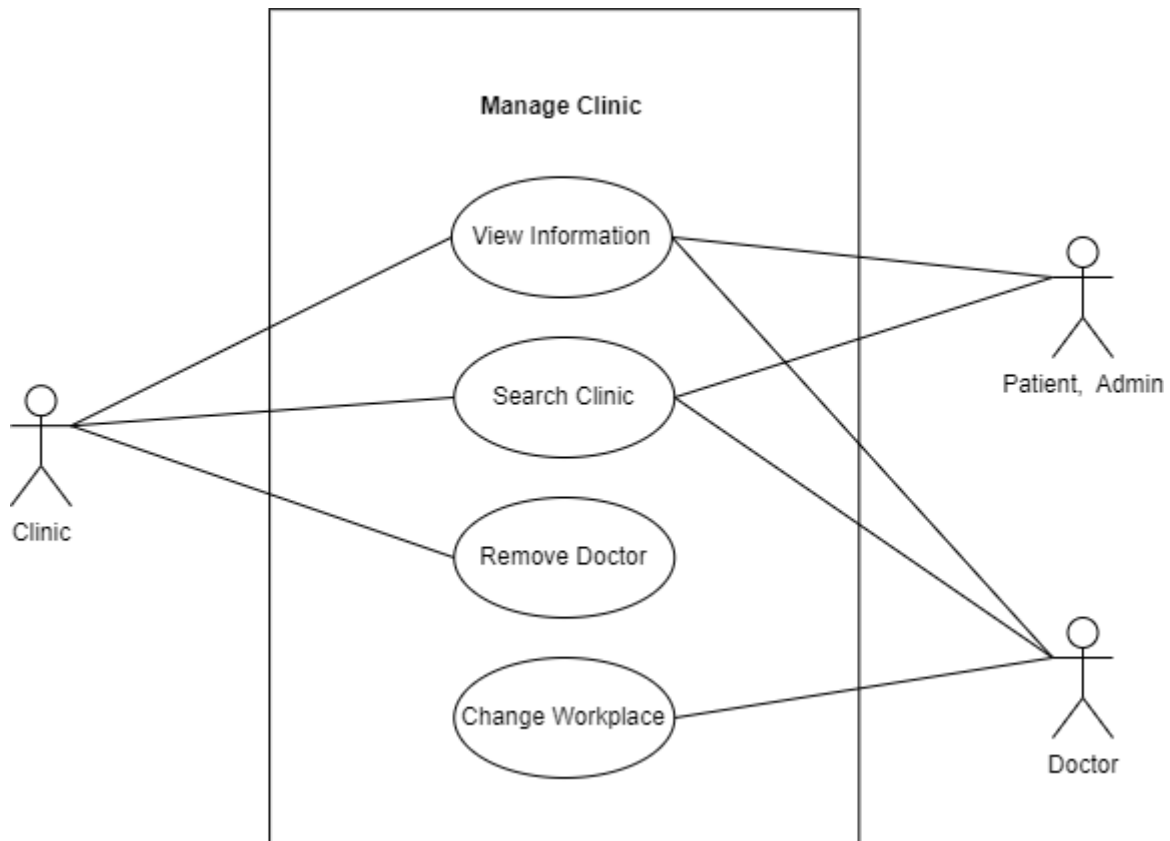
Use case name	Manage Medicine Information
Brief description	This use case is to manage medicine information.
Actor	Admin, Patient, Doctor, Clinic Representative
Pre-condition	Users need to have an account.
Basic flow	<ol style="list-style-type: none"> <li>1. Admin clicks “Medicine Management” button at navigation bar.</li> <li>2. Admin enter medicine information <ol style="list-style-type: none"> <li>a) Add information [<b>E1: Add Medicine Information</b>]</li> </ol> </li> </ol>

	<ul style="list-style-type: none"> <li>b) Edit Information [<b>E2: Edit Medicine Information</b>]</li> <li>c) Delete Information [<b>E3: Delete Medicine</b>]</li> </ul> <ol style="list-style-type: none"> <li>3. Users click ‘Medicine’ or “View” button to see the list of medicine.</li> <li>4. All users can search for the medicine by entering the medicine name. [<b>E4: Search Medicine</b>]</li> <li>5. The use case ends.</li> </ol>
Alternative flow	None
Exception flow	<p><b>E1: Add Medicine Information</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks the “+ New Medicine” button.</li> <li>2. System allows the admin to add medicine information.</li> <li>3. Admin enters medicine drug name, generic name, description, what is it used for, important information, and upload image.</li> <li>4. Admin clicks the “Save” button.</li> <li>5. Continue with step number 2 from the basic flow.</li> </ol> <p><b>E2: Edit Medicine Information</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks the “Edit” button.</li> <li>2. System allows the admin to edit any medicine information.</li> <li>3. Admin clicks the “Save” button.</li> <li>4. Continue with step number 2 from the basic flow.</li> </ol> <p><b>E3: Delete Medicine</b></p> <ol style="list-style-type: none"> <li>1. Admin clicks the “Delete” button on the medicine that he wants to delete.</li> <li>2. System allows the admin to delete the medicine.</li> <li>3. System displays a pop-up message “Are you sure to delete this data?”.</li> </ol>

	<ol style="list-style-type: none"> <li>4. Admin clicks the “OK” or “Cancel” button.</li> <li>5. Continue with step number 3 from the basic flow.</li> </ol> <p><b>E4: Search Medicine</b></p> <ol style="list-style-type: none"> <li>1. System allows the users to search medicine’s name.</li> <li>2. Users clicks search button.</li> <li>3. System displays medicine’s information.</li> <li>4. Continue with step number 5 from the basic flow</li> </ol>
Constraints	None

### 2.1.5 Manage Clinic

All users have access to the manage clinic module. Any clinic information in Pekan can be viewed by the patient. The clinic staff can also remove the doctor’s account that works under them. The use case diagram and description that follow go over this module in greater depth:



**Figure 1. 7 Manage Clinic Use Case Diagram**

**Table 1. 5 Manage Clinic Use Case Description**

Use case name	Manage Clinic
Brief description	This use case is to manage clinic information
Actor	Admin, Patient, Clinic representative, Doctor
Pre-condition	<ol style="list-style-type: none"> <li>1. All users already have an account.</li> <li>2. Only patient can see the “view” and “book” button to book an appointment.</li> </ol>

Basic flow	<ol style="list-style-type: none"> <li>1. Users click “View” button at main page.</li> <li>2. Users can view the clinic’s information by entering the clinic name.</li> <li>3. Users can search for the clinic. [E2: Search Clinic]</li> <li>4. System allows the doctor to change their workplace details. [E3: Change Workplace]</li> <li>5. System allows the clinic representative to remove any doctor account from their clinic [E4: Remove Doctor]</li> <li>6. The use case end.</li> </ol>
Alternative flow	None
Exception flow	<p><b>E1: Search Clinic</b></p> <ol style="list-style-type: none"> <li>1. Users enter clinic name.</li> <li>2. System displays the result.</li> <li>3. Patient can click “view” button to book an appointment.</li> <li>4. Continue with step number 3 from the basic flow.</li> </ol> <p><b>E2: Change Workplace</b></p> <ol style="list-style-type: none"> <li>1. Doctor clicks the “Change Workplace” button.</li> <li>2. System allows the doctor to edit the information.</li> <li>3. Doctor clicks the “save” button.</li> <li>4. Continue with step number 4 from the basic flow.</li> </ol> <p><b>E3: Remove Doctor</b></p> <ol style="list-style-type: none"> <li>1. Clinic representative clicks the “Doctor Management” button.</li> <li>2. Clinic representative clicks the “Remove” button on the profile that he wants to remove.</li> <li>3. System allows the clinic representative to remove the account.</li> <li>4. System displays a pop-up message “Are you sure you want to delete this data?”.</li> </ol>

---

	<ul style="list-style-type: none"><li>5. Clinic representative clicks the “OK” or “Cancel” button.</li><li>6. Continue with step number 5 from the basic flow.</li></ul>
Constraints	Only patient can see the “view” button to book an appointment with the clinic.

## 2.2 SEQUENCE DIAGRAM

### 2.2.1 Manage Account

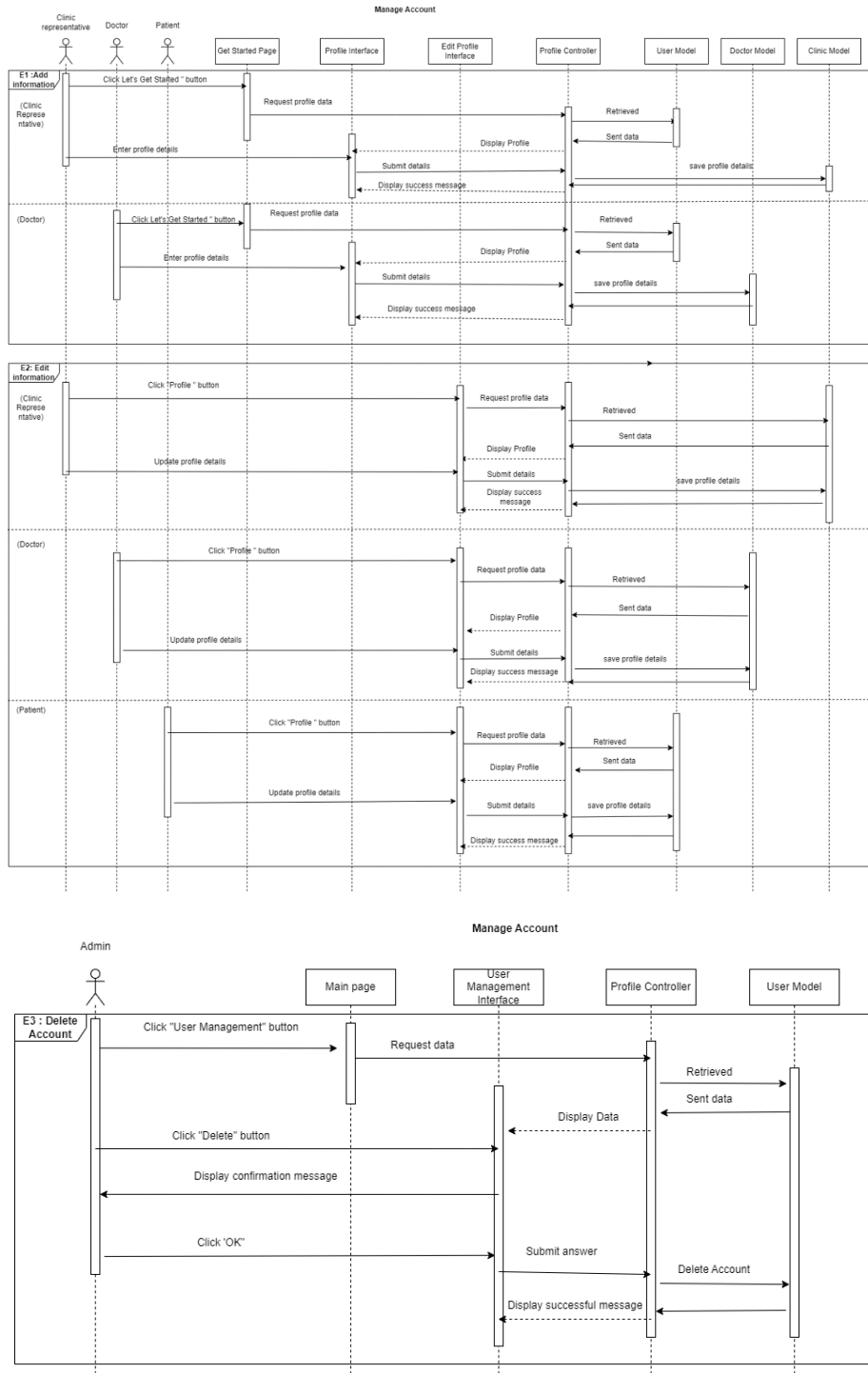


Figure 1. 8 Manage Account Sequence Diagram

The manage account sequence diagram shows that the module can add information, edit information, and delete the account. The sequence diagram has four actors which are the admin, patient, clinic, and doctor. As shown in the sequence diagram above, the doctor, patient and clinic can add their personal information while the admin cannot but all of them can edit their information. The admin had the authority to delete any account.



### 2.2.2 Manage Appointment Booking

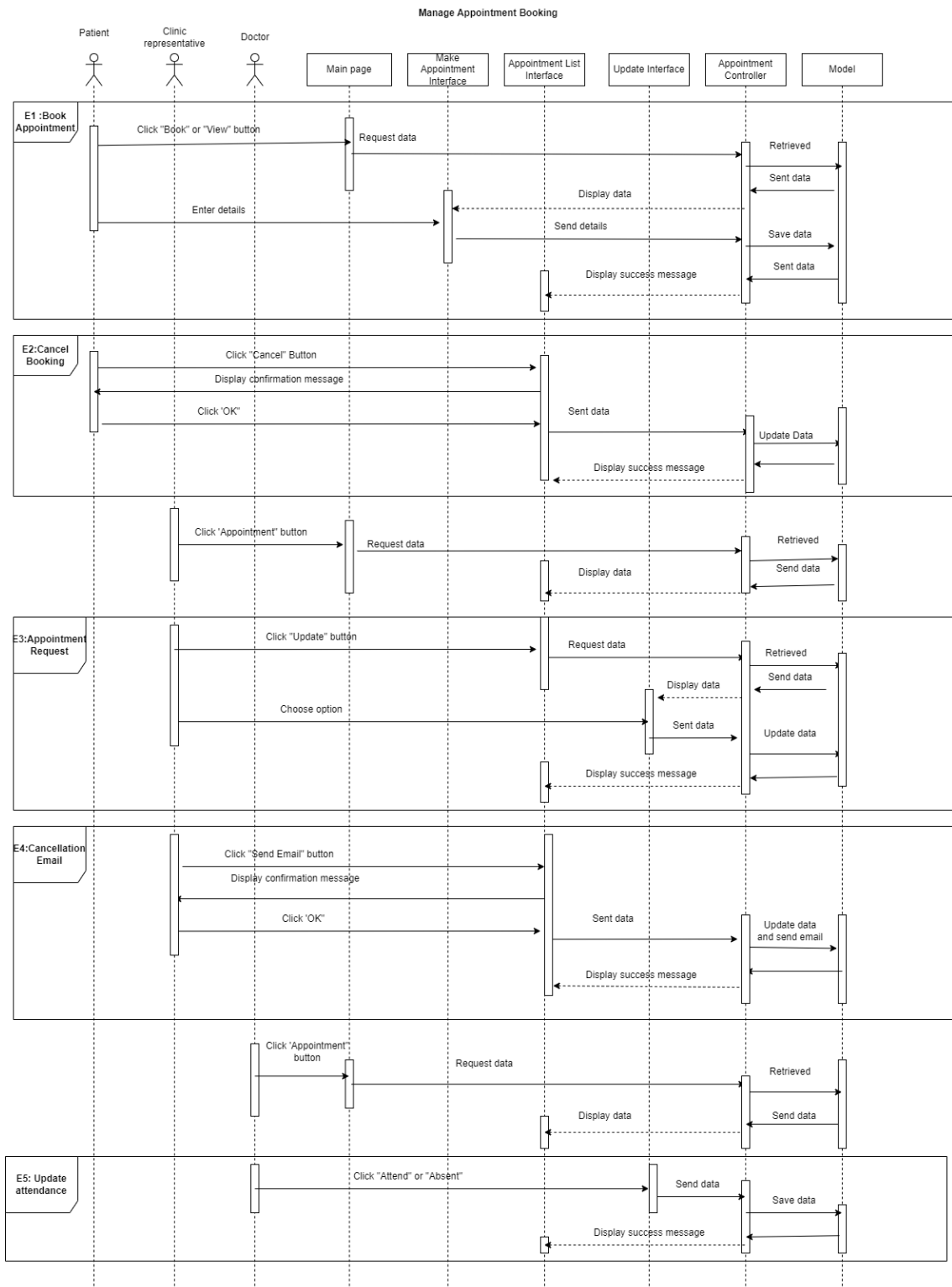
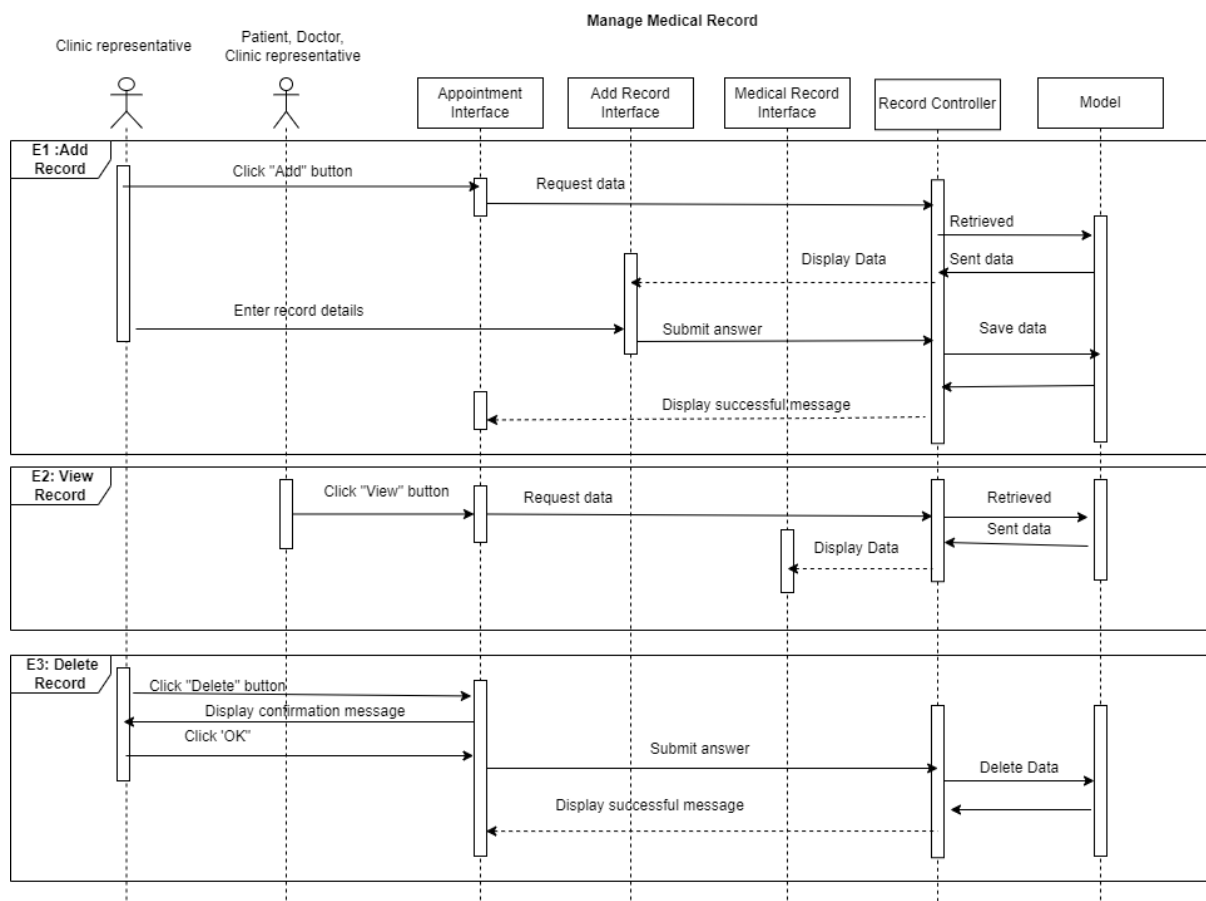


Figure 1. 9 Manage Appointment Booking Sequence Diagram

The figure above shows that the manage appointment booking has three actors which are the clinic, patient, and doctor. This module can book an appointment, cancel the booking, update appointment requests, view history, and search for the doctor by name or specialization. As shown in the sequence diagram above, the patient can book and cancel an appointment while the doctor can update the attendance status. The clinic staff can approve or reject the appointment request from the patient. Besides the patient, the clinic staff can also cancel the appointment. When the staff cancel the appointment, a cancellation email will be sent to the patient. Moreover, the clinic, patient, and doctor can view the appointment history records.

### 2.2.3 Manage Medical Record

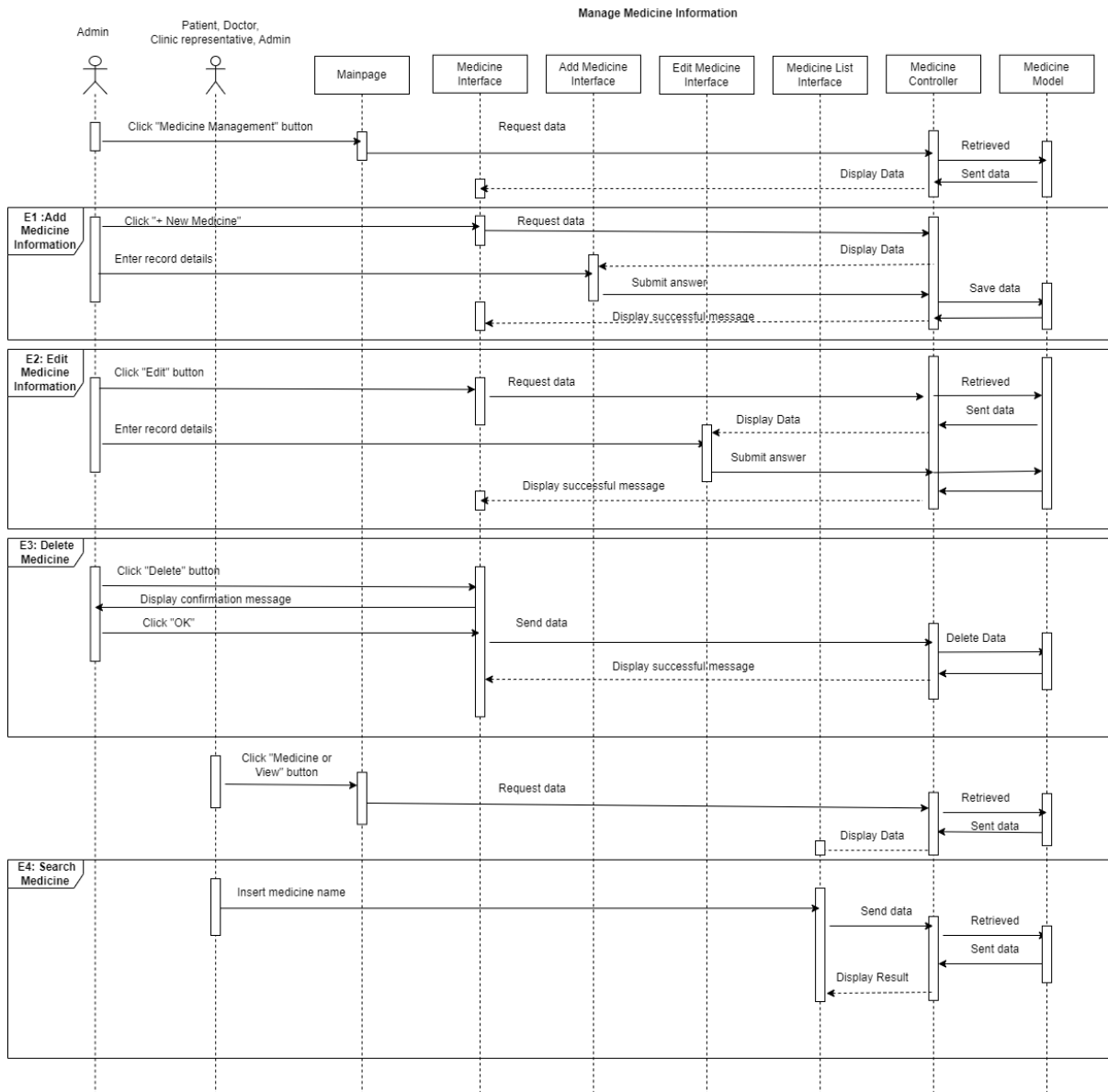


**Figure 1. 10 Manage Medical Record Sequence Diagram**

The figure above is the sequence diagram for the manage medical record module. There are three actors which are the patient, clinic staff and doctor. As shown in the sequence diagram

above, the clinic staff can add, view and delete the medical record while the patient and doctor only can view the medical record.

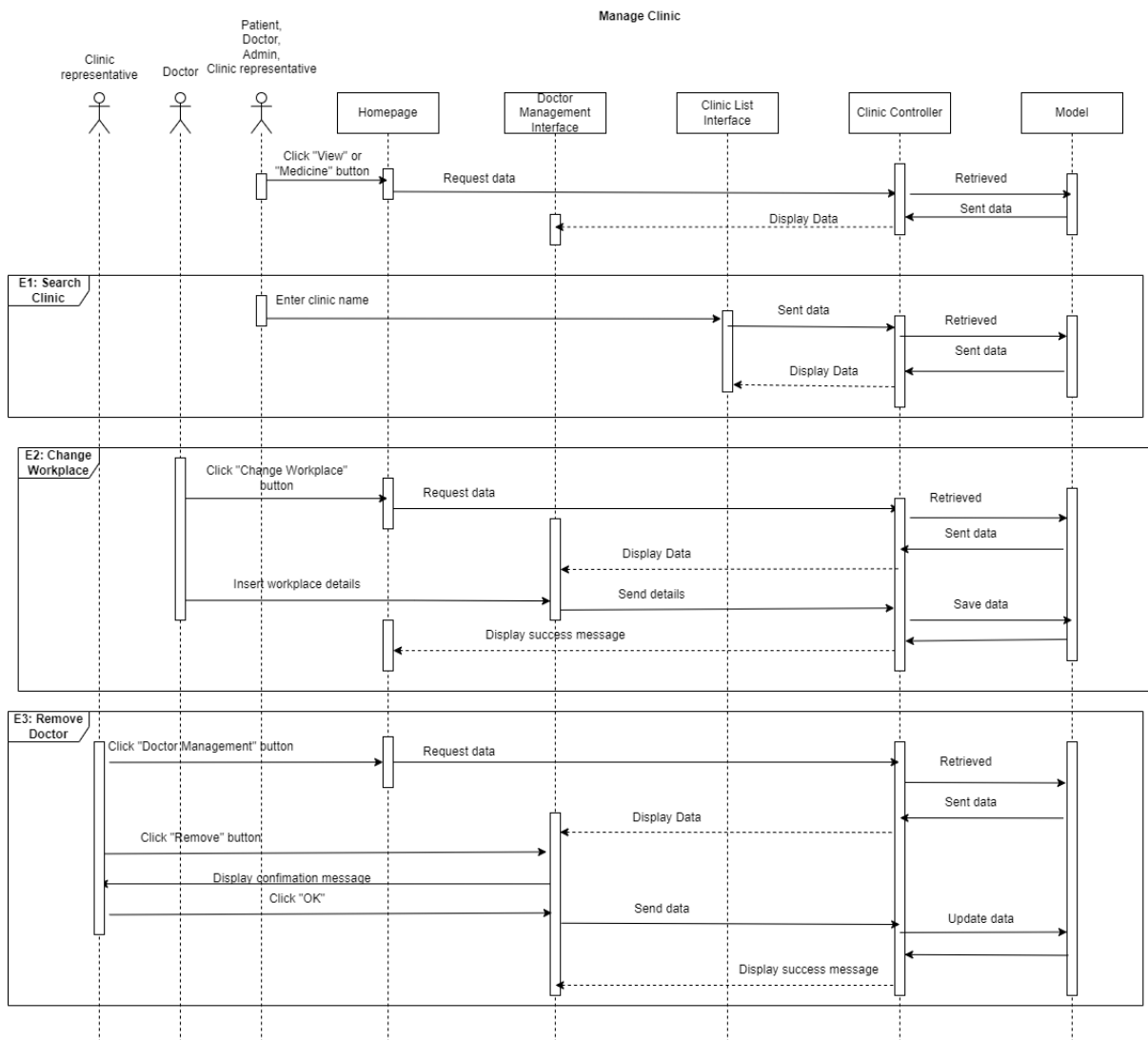
### 2.2.4 Manage Medicine Information



**Figure 1. 11 Manage Medicine Information Sequence Diagram**

The figure above is the sequence diagram for the manage medicine information module. All the users of this project are involve in this module. The sequence diagram above shows that this module can add medicine information, edit medicine information, delete medicine by the admin. The other users can only view and search the medicine information.

### 2.2.5 Manage Clinic



**Figure 1. 12 Manage Clinic Sequence Diagram**

The figure above is the sequence diagram for the manage clinic module. This module involves all the actors. As shown in the sequence diagram above, all the actors can view the clinic information. The clinic staff can remove the doctor account that works under them while the doctor can change her workplace details.

## CHAPTER 3

## 3.1 INTERFACE DESIGN

## 3.1.1 ALL USERS CAN ACCESS

## 3.1.1.1 Main Page

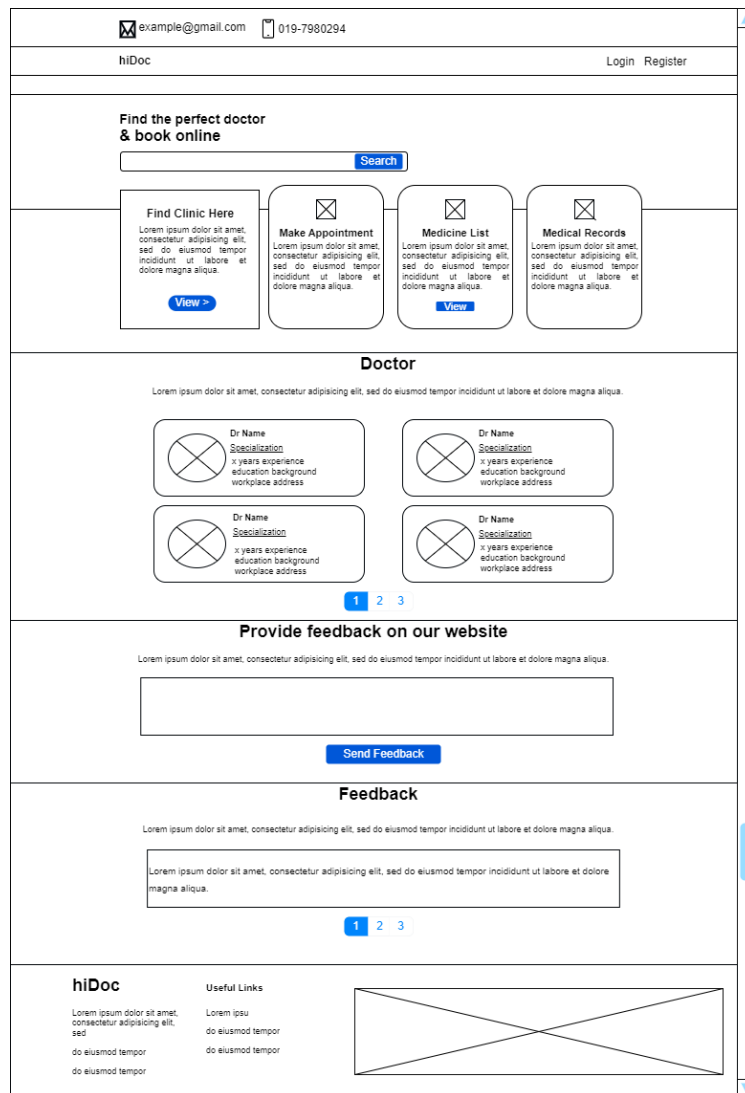


Figure 1. 13 Main Page Interface

This is hiDoc's main page. Users can use this interface to search for doctors, view clinics that have registered with the system, add and view existing feedback, and view medicine lists. However, for patients who log in to the system, a button to book an appointment, such as "Book," will be available. Figure 1.14 illustrates this. Because only

the patient can book an appointment for this system, the button is hidden for the other user type. The patient has the option of viewing a list of doctors who are available in each clinic.

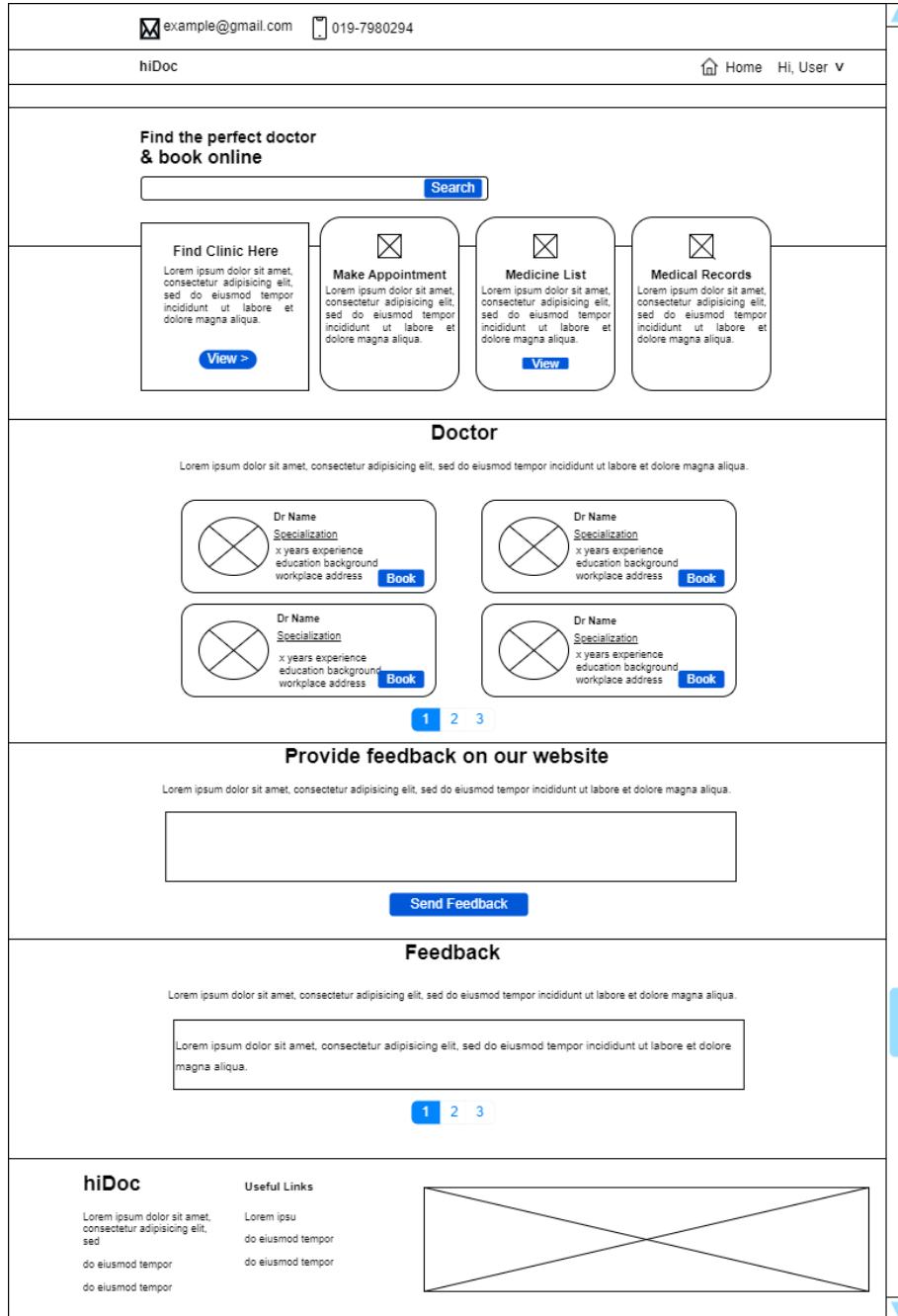
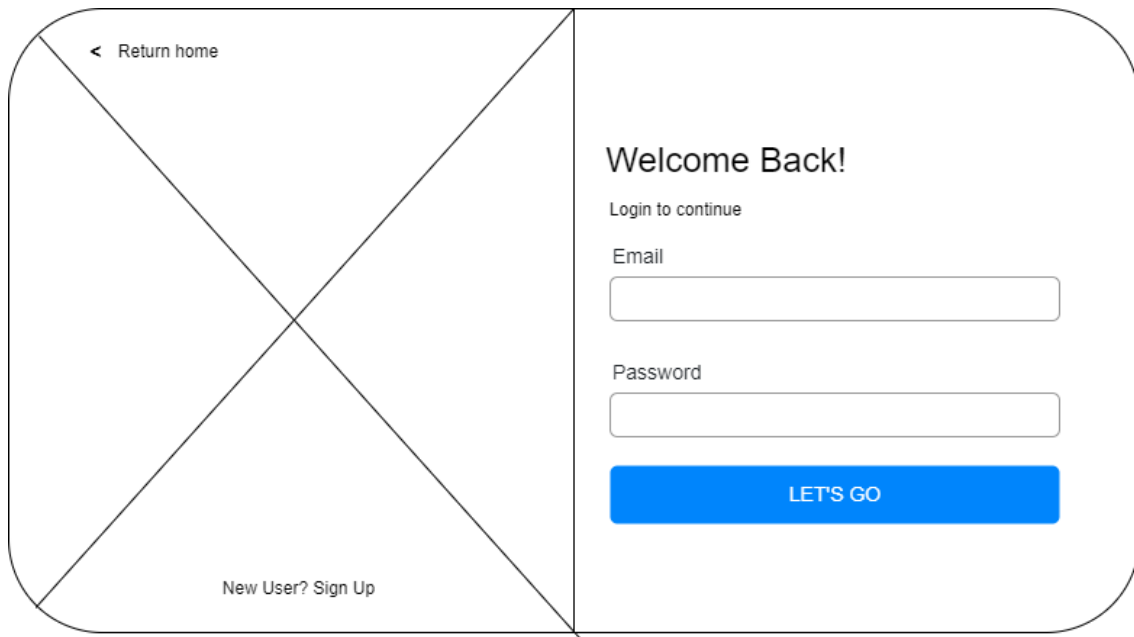


Figure 1. 14 Main Page – User Interface

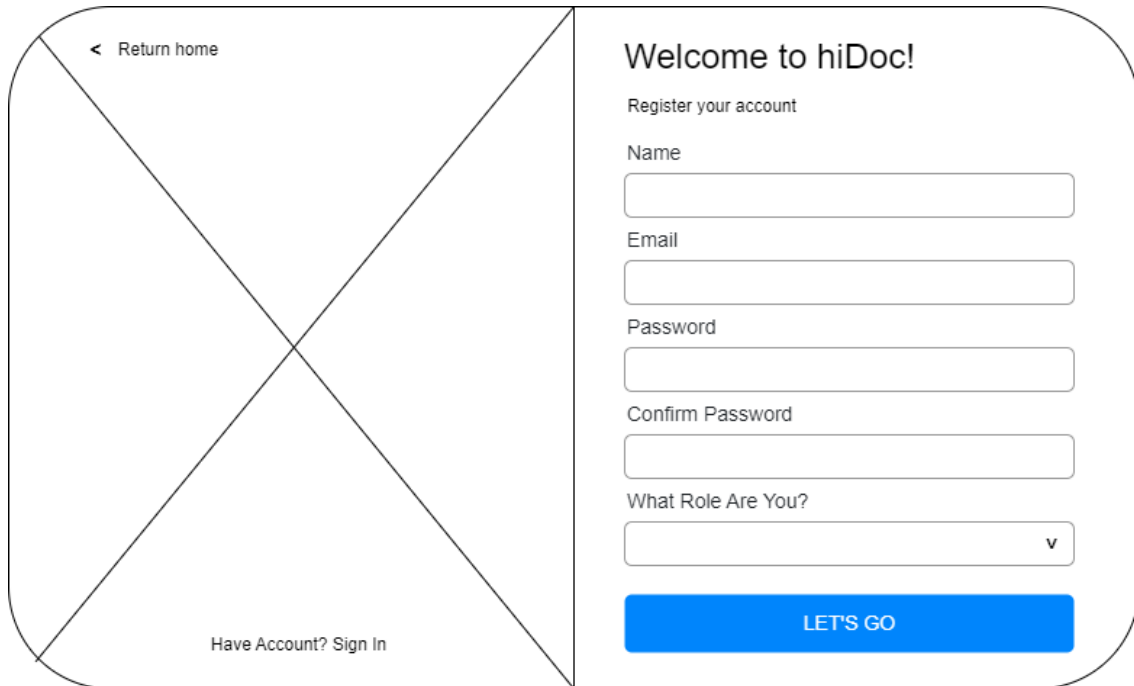


The login interface wireframe is contained within a rounded rectangle. On the left side, there is a large 'X' shape. In the top-left corner of this 'X' area, there is a back arrow icon followed by the text '< Return home'. In the bottom-right corner of the 'X' area, there is the text 'New User? Sign Up'. On the right side of the interface, the heading 'Welcome Back!' is displayed. Below the heading is the text 'Login to continue'. There are two input fields: the first is labeled 'Email' and the second is labeled 'Password'. Below these fields is a blue button with the text 'LET'S GO' in white capital letters.

**Figure 1. 15 Login Interface**

This is the system's login interface. Users must enter their email address and password to login.

### 3.1.1.2 Register



The register interface wireframe is contained within a rounded rectangle. On the left side, there is a large 'X' shape. In the top-left corner of this 'X' area, there is a back arrow icon followed by the text '< Return home'. In the bottom-right corner of the 'X' area, there is the text 'Have Account? Sign In'. On the right side of the interface, the heading 'Welcome to hiDoc!' is displayed. Below the heading is the text 'Register your account'. There are five input fields: the first is labeled 'Name', the second is labeled 'Email', the third is labeled 'Password', the fourth is labeled 'Confirm Password', and the fifth is a dropdown menu labeled 'What Role Are You?' with a downward arrow icon. Below these fields is a blue button with the text 'LET'S GO' in white capital letters.

**Figure 1. 16 Register Interface**

This is the system's registration interface. Users must enter their name, email address, password, and role to register. The users also need to confirm their password.

### 3.1.1.3 Manage Clinic Module (Clinic List)

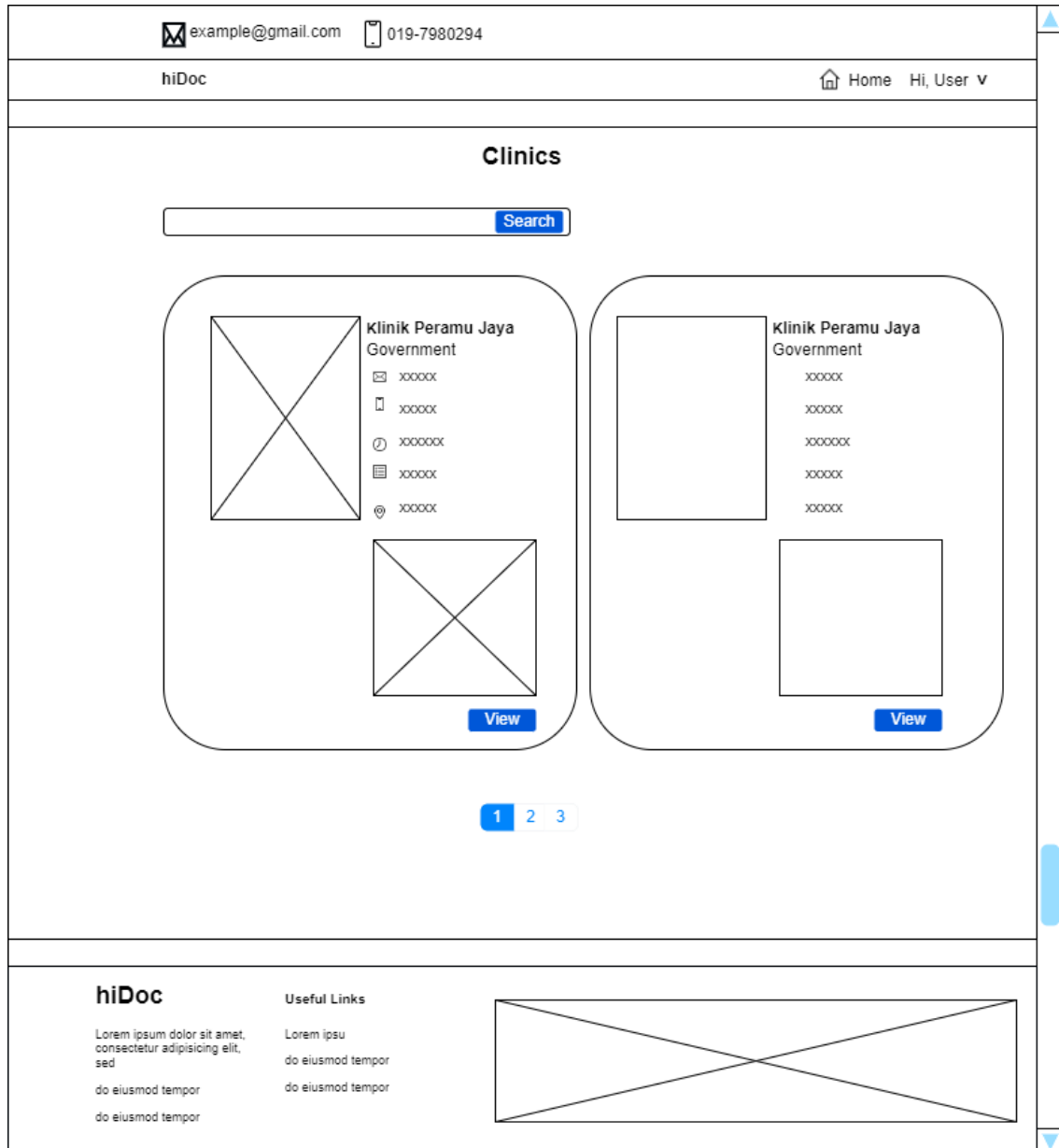


Figure 1. 17 Clinic List Interface



This is the clinic list interface, where all clinic information is displayed. Users must click the "View >" button on the main page under "Find Clinic Here" to get to this page.

### 3.1.1.4 Manage Medicine Module (Medicine List)

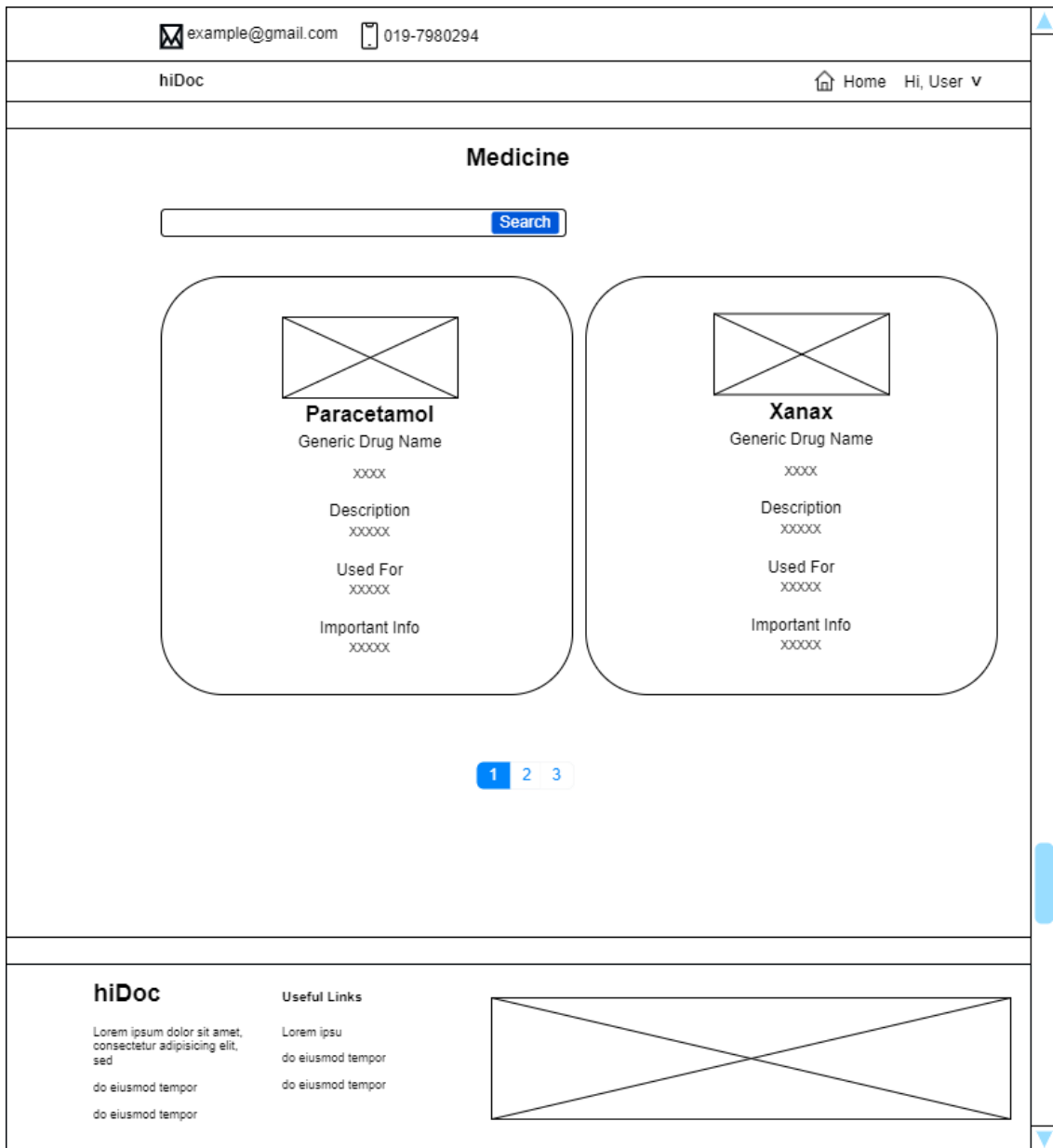
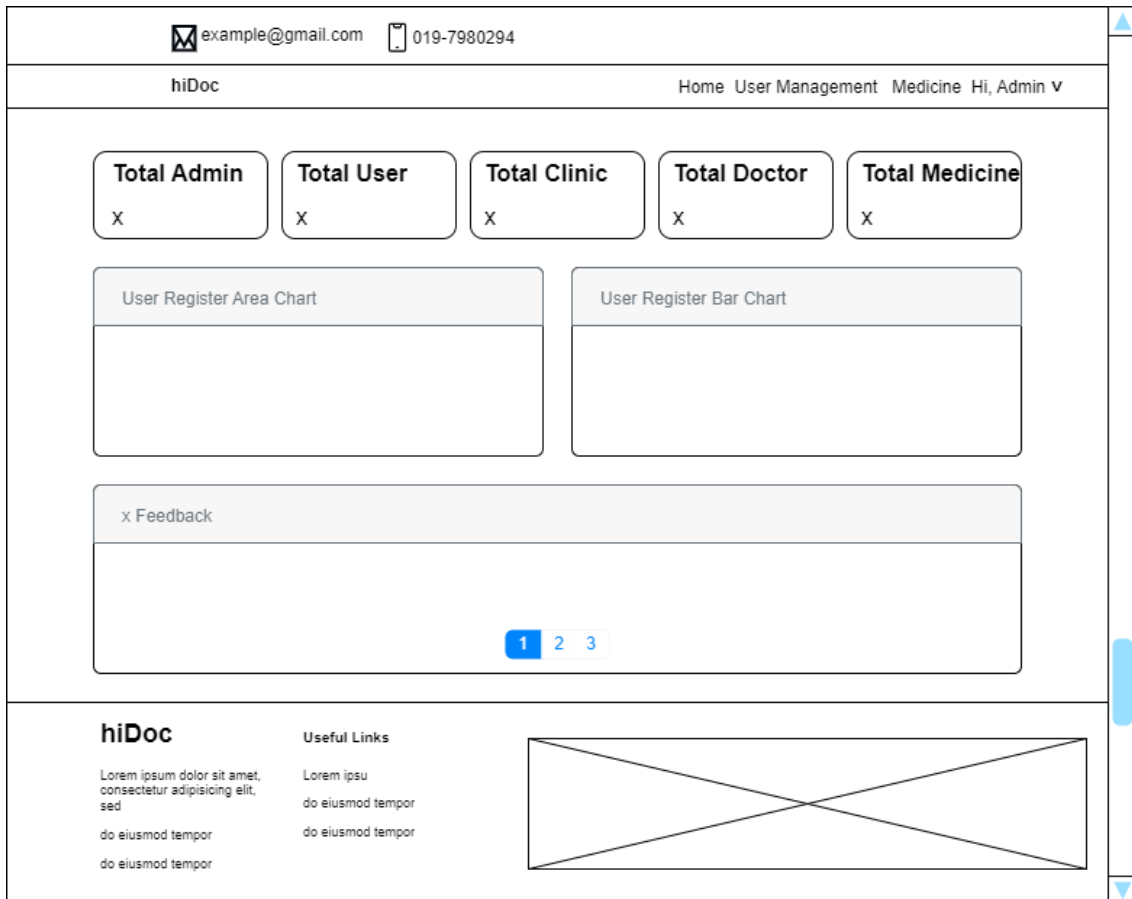


Figure 1. 18 Medicine List Interface

This is the medicine list interface, where all medicine information is displayed. Users must click the "View " button on the main page under "Medicine List" to get to this page.

### 3.1.2 ADMIN INTERFACE

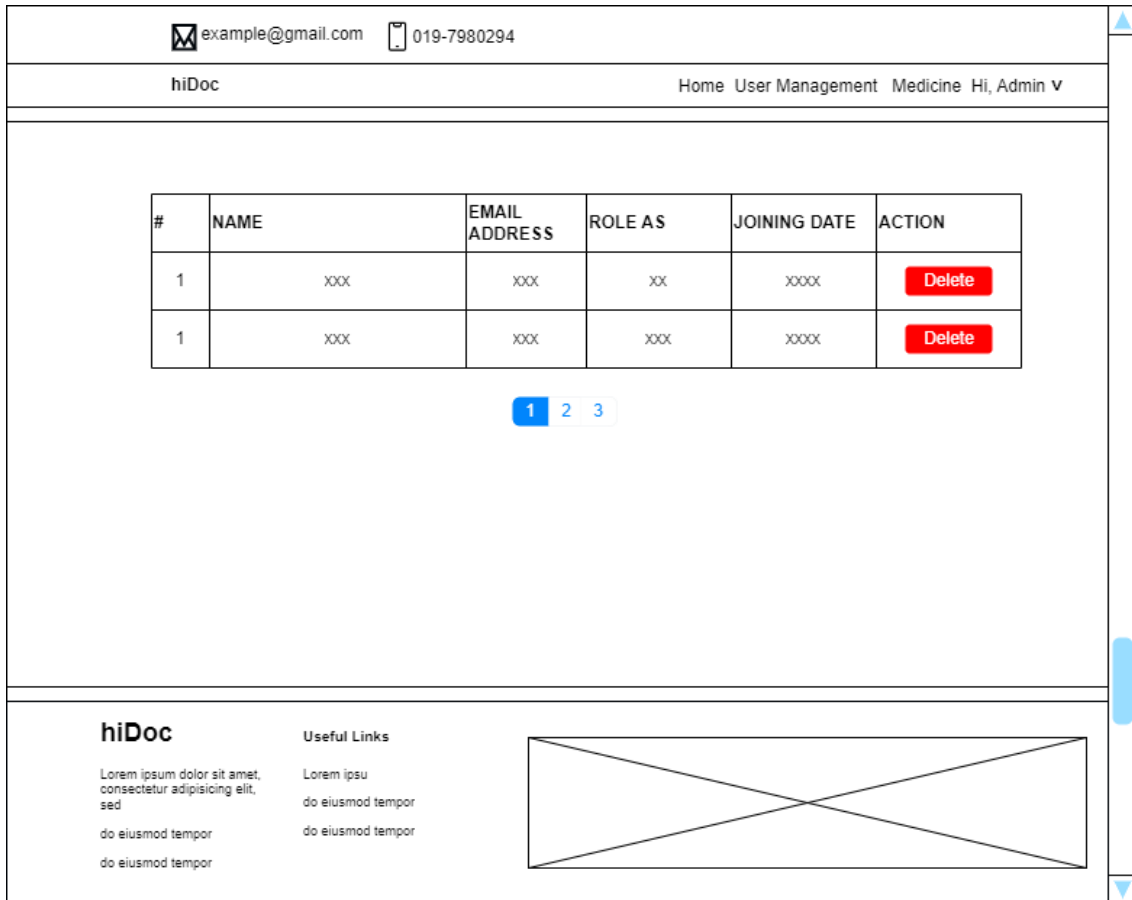
#### 3.1.2.1 Homepage



**Figure 1. 19 Admin – Homepage Interface**

This is the admin homepage. This interface displays the total number of administrators, users or patients, clinics, and doctors registered in this system. There is also a total count of the medications available in the system. The user feedback is also displayed on the admin homepage for him or her to see. There are also an area chart and bar chart that display the total count of user registration by month. There are also area and bar charts that display user registration data.

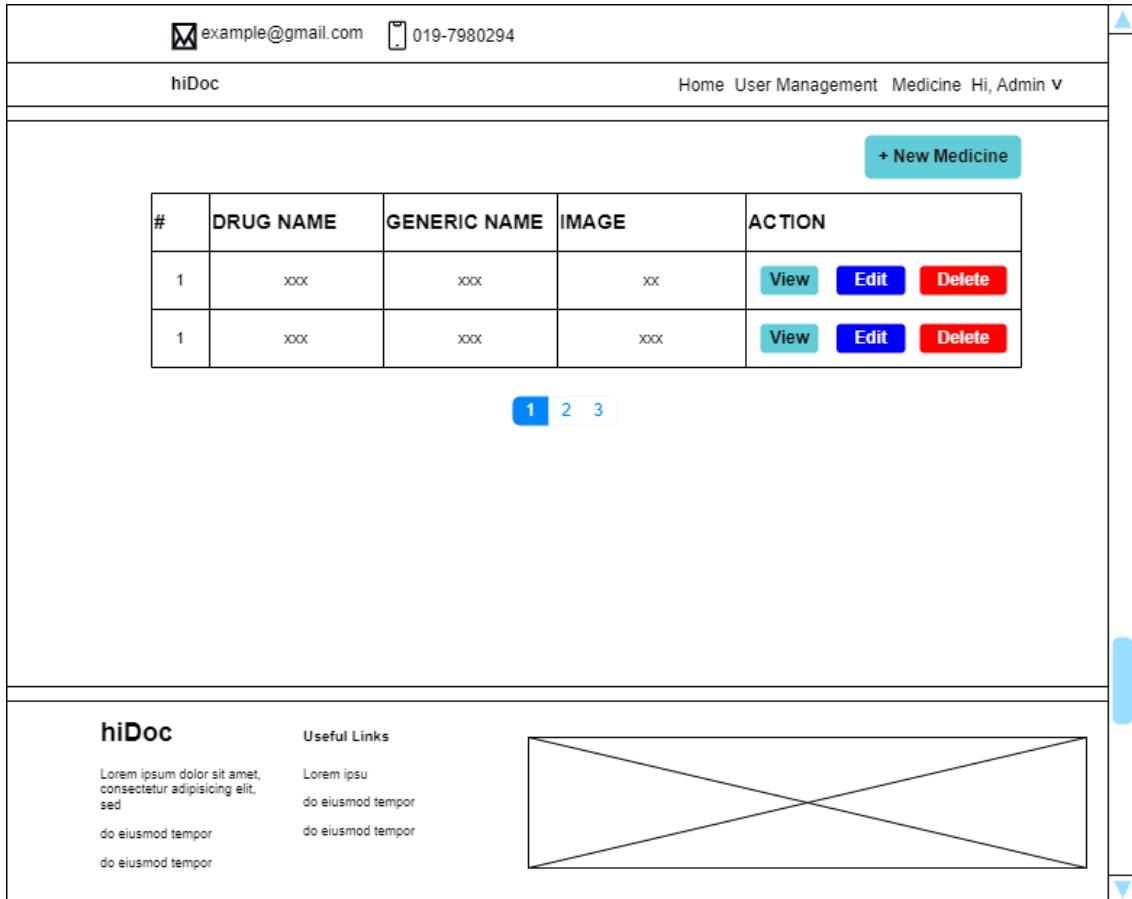
### 3.1.2.2 Manage Account Module (User Management)



**Figure 1. 20 User Management Interface**

This is the user management page, where the administrator can delete any account, he wants.

### 3.1.2.3 Manage Medicine Module (Medicine Management)

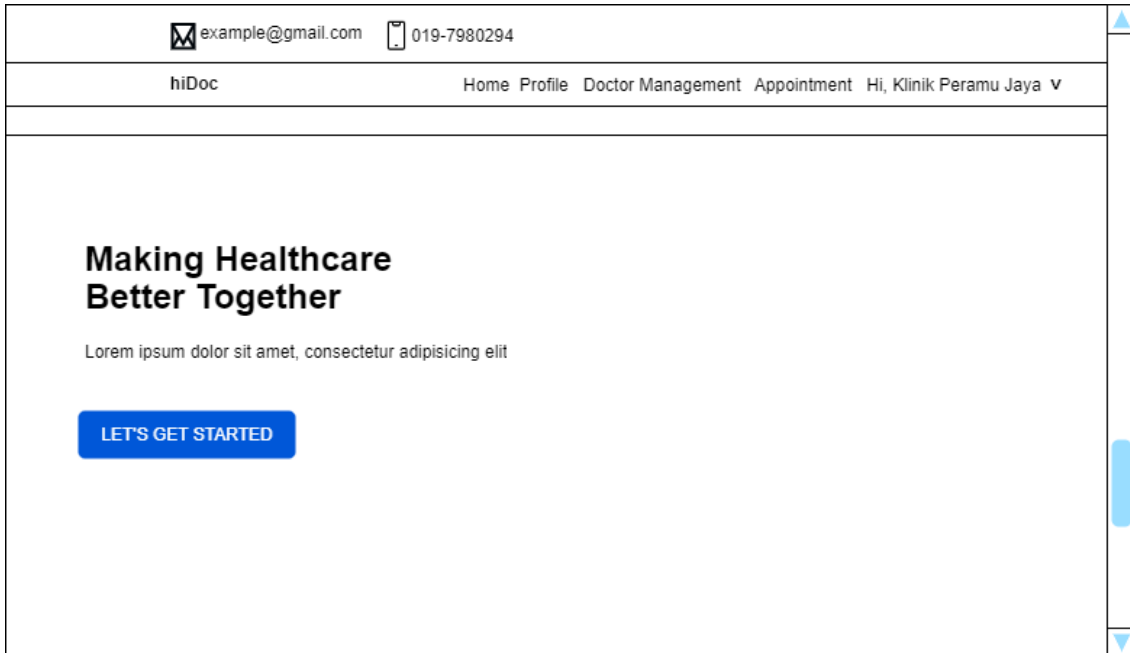


**Figure 1. 21 Medicine Management Interface**

This is the medicine administration interface. By clicking the "view" button, the administrator can view the medication information. By clicking the "+ New Medicine" button, the administrator can also add new medicine information. By clicking the "Edit" and "Delete" buttons, the administrator can also edit and delete the medicine.

### 3.1.3 CLINIC INTERFACE

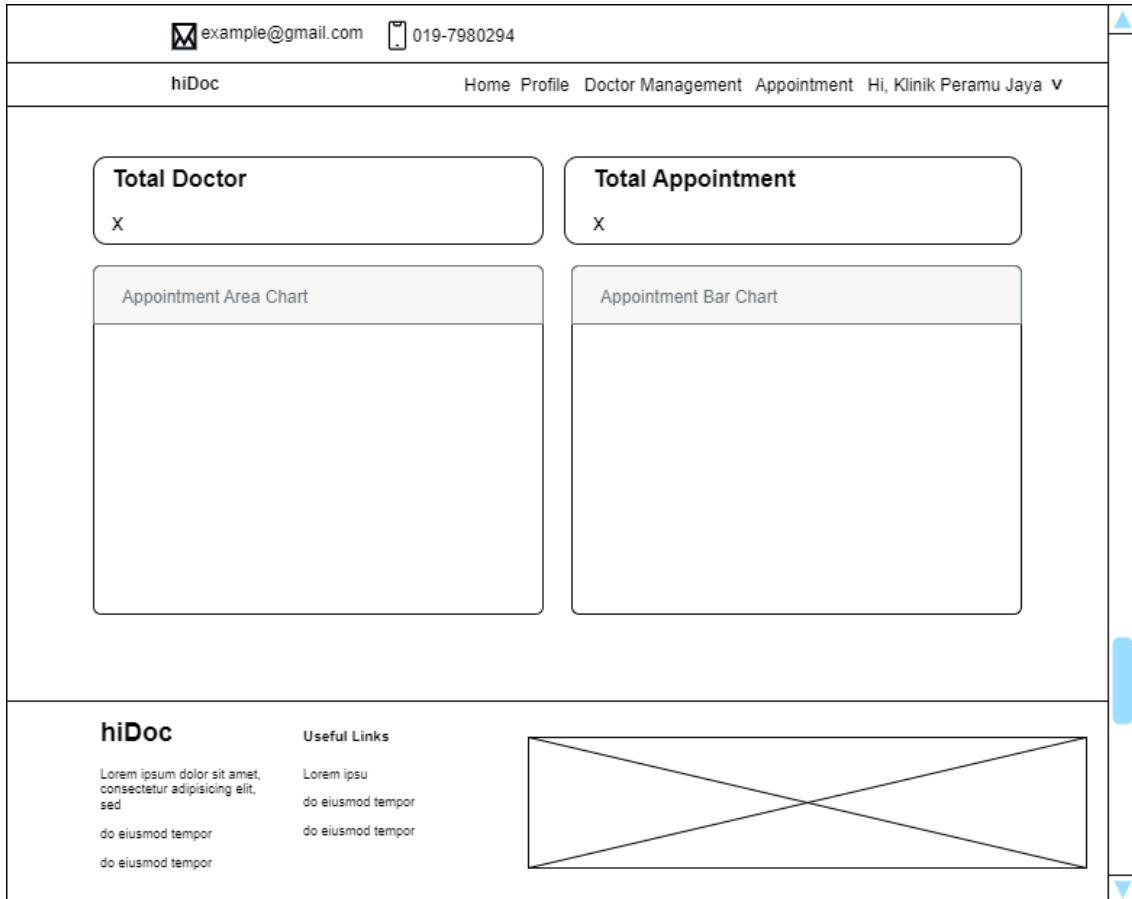
#### 3.1.3.1 Get Started



**Figure 1. 22 Get Started - Clinic Interface**

This is the get started page. This page will appear after a clinic staff member or representative has just registered in the system. The clinic staff will be directed to a page where they must enter the clinic's information after clicking the button. This data will be displayed for the users to see.

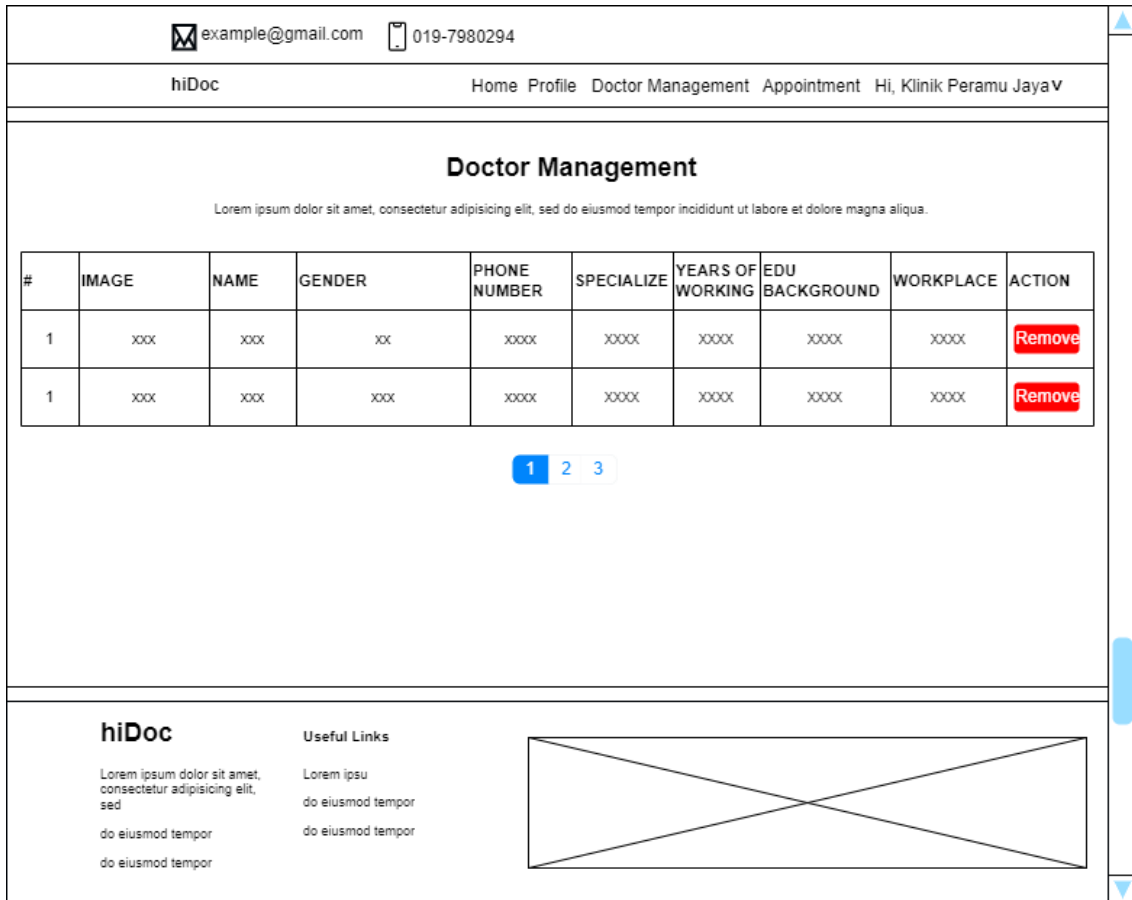
### 3.1.3.2 Homepage



**Figure 1. 23 Clinic – Homepage Interface**

This is the clinic's homepage. This interface displays the clinic's total number of doctors as well as the total number of appointments. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.1.3.3 Manage Account Module (Doctor Management)



**Figure 1. 24 Doctor Management Interface**

This is the interface for doctor management. The clinic representative or staff can delete the doctor's account from this interface. This interface also displays information about the doctor.

### 3.1.3.4 Manage Account Module (Clinic Profile)

The screenshot displays the 'Clinic Profile' interface within the hiDoc application. At the top, the user's email 'example@gmail.com' and phone number '019-7980294' are shown. The navigation bar includes 'hiDoc', 'Home', 'Profile', 'Doctor Management', 'Appointment', and 'Hi, Klinik Peramu Jaya'. The main heading is 'Hi, Klinik Peramu Jaya' with a placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.' The central form is titled 'CLINIC INFORMATION' and contains several sections: 'Clinic Photo' with a placeholder image and a 'Choose File' button; 'Clinic Name' and 'Email Address' input fields; an 'Address' input field; an 'ABOUT CLINIC' section with fields for 'Government/Private Clinic', 'Services available', 'Operating Hours', and 'Contact Number'. A blue 'Update Profile' button is located at the bottom of the form. The footer contains the hiDoc logo, placeholder text, and a 'Useful Links' section with a placeholder image.

**Figure 1. 25 Clinic Profile Interface**

This is the profile interface for the clinic. The clinic's information can be edited here by clicking the "Update Profile" button.



### 3.1.3.5 Manage Appointment Booking Module (Appointment List & History)

example@gmail.com 019-7980294

hiDoc Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya

#### Appointment List

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	PATIENT NAME	CONTACT NO	DATE	TIME SLOT	MESSAGE	DOCTOR SELECTED	SPECIALIZE	STATUS	ATTENDANCE	ACTION	CANCELLATION	MEDICAL RECORDS
1	XXX	XXX	xxx	xxx	xxx	xxxx	xxxx	Attend	Attend	Update Delete	Send Email	XXXX
1	XXX	XXX	xxx	xxx	xxx	xxxx	xxxx	Pending	Attend	Update Delete	Send Email	XXXX

1 2 3

**hiDoc**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor do eiusmod tempor do eiusmod tempor

**Useful Links**

Lorem ipsu do eiusmod tempor do eiusmod tempor

**Figure 1. 26 Appointment List Interface**

This is the clinic's appointment list interface. This interface displays the appointment list created by users for the clinic. The clinic staff can update the status of whether the appointment request was approved or rejected, as well as whether the patient attended the appointment or not. The appointment data can also be deleted by the staff. The appointment can also be cancelled by clicking the "Send Email" button. After pressing the button, an email will be sent to the patient who scheduled the appointment.

### 3.1.3.6 Manage Medical Record Module (Add, View, Delete)

✉ example@gmail.com
☎ 019-7980294

hiDoc
Home Profile Doctor Management Appointment Hi, Klinik Peramu Jaya ▾

#### Appointment List

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	PATIENT NAME	CONTACT NO	DATE	TIME SLOT	MESSAGE	DOCTOR SELECTED	SPECIALIZE	STATUS	ATTEN DANCE	ACTION	CANCELLATION	MEDICAL RECORDS
1	XXX	XXX	xxx	xxx	xx	xxxx	xxxx	xxxx	<span style="background-color: green; color: white; padding: 2px 5px;">Attend</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Update</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>	<span style="background-color: orange; color: white; padding: 2px 5px;">Send Email</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">View</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>
1	XXX	XXX	xxx	xxx	xxx	xxxx	xxxx	xxxx	<span style="background-color: green; color: white; padding: 2px 5px;">Attend</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Update</span> <span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>	<span style="background-color: orange; color: white; padding: 2px 5px;">Send Email</span>	<span style="background-color: blue; color: white; padding: 2px 5px;">Add</span>

1 2 3

**hiDoc**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor do eiusmod tempor do eiusmod tempor

**Useful Links**

Lorem ipsu do eiusmod tempor do eiusmod tempor

**Figure 1. 27 Medical Record Interface**

This is the interface for the appointment list. This interface is linked to the medical record management module. When the attendance status is "Attend," the staff will see a "Add" button under the "MEDICAL RECORDS" column. The staff can enter the doctor's diagnosis for the user. Following the addition of data, the "Add" button will be replaced by "View" and "Delete" buttons. The "view" button allows staff to view the patient's medical record.

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxx
Diagnosis	xxxxx
Prescribed Medicine	xxxxx

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

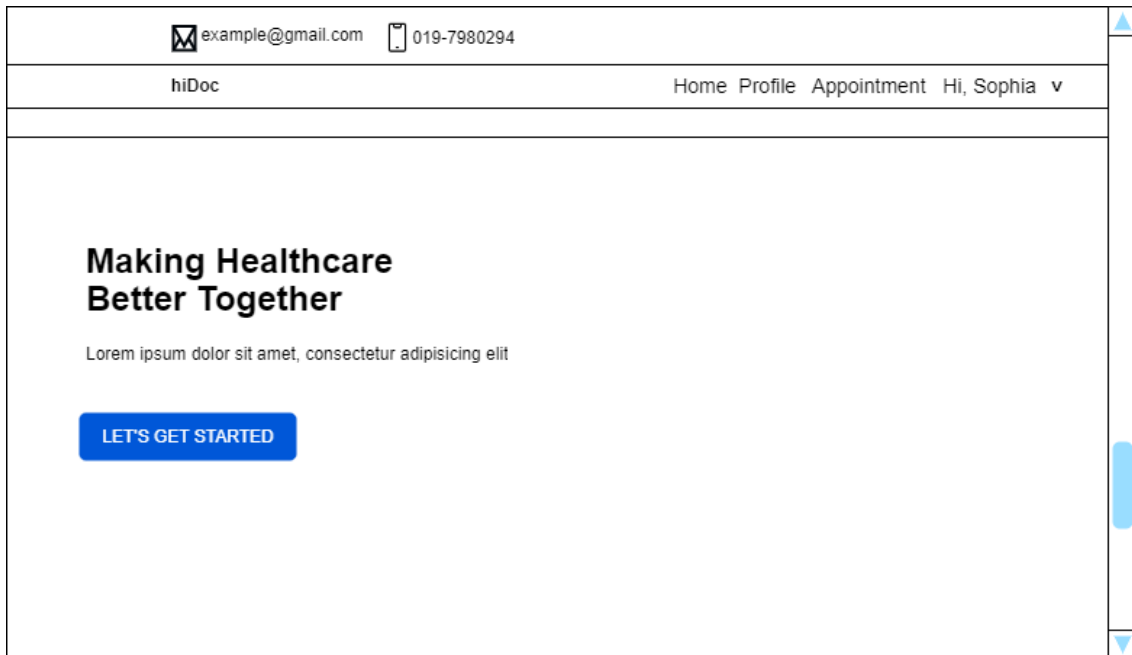
**hiDoc**

**Figure 1. 28 Medical Record Interface**

This is the patient's medical record. It contains information about the appointment as well as the doctor's diagnosis.

### 3.1.4 DOCTOR INTERFACE

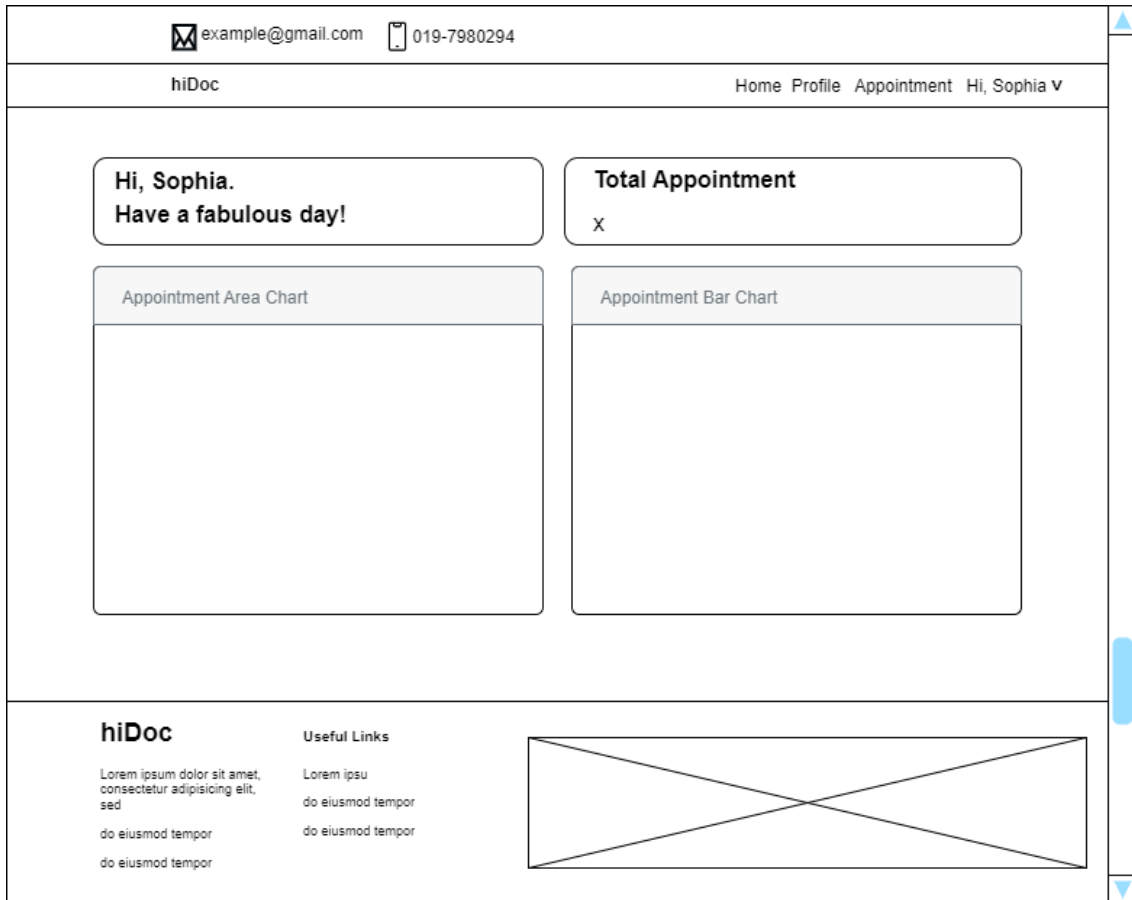
#### 3.1.4.1 Get Started



**Figure 1. 29 Get Started - Doctor Interface**

This is the get started page. This page will appear after the doctor registered in the system. The doctor will be directed to a page where they must enter their information after clicking the button. This data will be displayed for the users to see.

### 3.1.4.2 Homepage



**Figure 1. 30 Doctor – Homepage Interface**

This is the doctor's homepage. This interface displays the total number of appointments of the doctor. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.1.4.3 Manage Account Module (Doctor Profile)

The screenshot displays the 'Doctor Profile' interface within the 'hiDoc' application. At the top, the user's email 'example@gmail.com' and phone number '019-7980294' are shown. The navigation bar includes 'hiDoc', 'Home', 'Profile', 'Appointment', and 'Hi, Sophia v'. The main content area is titled 'Hi, Sophia' and contains a placeholder for a profile photo. Below this is the 'USER INFORMATION' section, which includes a 'Profile Photo' area with a 'Choose File' button and a 'No file chosen' message. The 'Name' field is followed by 'Gender' and 'Email Address' fields. Below these are 'Phone Number' and 'Workplace' fields. The 'MEDICAL BACKGROUND' section contains 'Specialization' and 'Working Experience (Years)' fields. The 'ABOUT ME' section features an 'Education Background' text area. A blue 'Update Profile' button is located at the bottom of the profile form. The footer contains the 'hiDoc' logo, 'Useful Links' with placeholder text, and a large placeholder image.

**Figure 1. 31 Doctor Profile Interface**

This is the profile interface for the doctor. The doctor's information can be edited here by clicking the "Update Profile" button.

### 3.1.4.4 Manage Appointment Booking Module (Appointment List & History)

example@gmail.com 019-7980294

hiDoc Home Profile Appointment Hi, Sophia v

### Appointment List

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

#	PATIENT NAME	DATE	TIME SLOT	MESSAGE	STATUS	ATTENDANCE	ACTION	MEDICAL RECORD
1	XXX	XXX	XXX	XXXX	Approve	Attend	Attend Absent View	
2	XXX	XXX	XXX	XXX	Approve	Pending	Attend Absent	No record

1 2 3

**hiDoc**

Useful Links

do eiusmod tempor do eiusmod tempor do eiusmod tempor

**Figure 1. 32 Appointment List Doctor Interface**

This is the interface for the doctor's appointment list. This interface displays the appointment list that users have created for the doctor. By clicking the "Attend" or "Absent" button, the doctor can update the patient's attendance for the appointment session. This interface is linked to the module manage medical records. When the clinic staff enters the medical record into the system, a "View" button appears in the interface, allowing the doctor to view the patient's medical record for that appointment session. This medical record is depicted in Figure 1.33 below.

3.1.4.5 Manage Medical Record Module (View)

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxx
Diagnosis	xxxxx
Prescribed Medicine	xxxxx

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

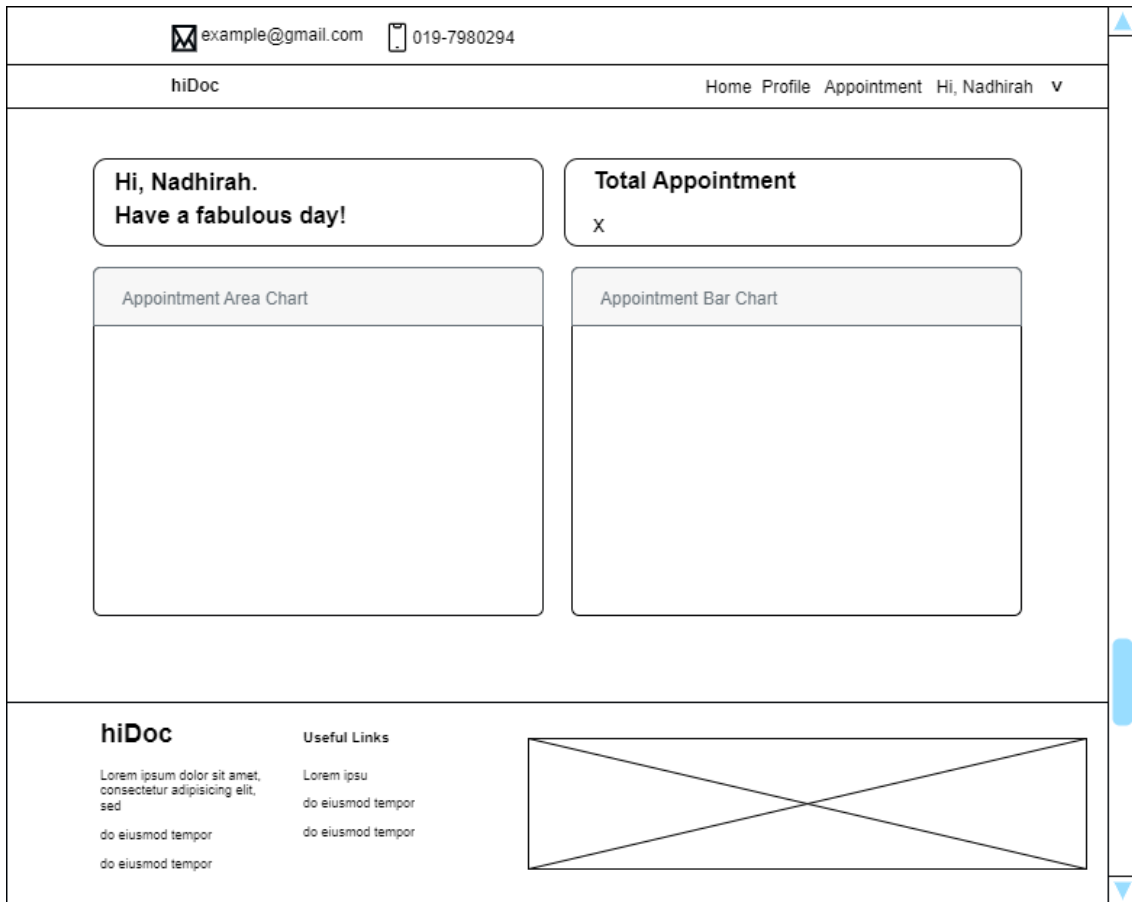
**hiDoc**

Figure 1. 33 Medical Record Interface



### 3.1.5 PATIENT INTERFACE

#### 3.1.5.1 Homepage



**Figure 1. 34 Patient - Homepage**

This is the patient's homepage. This interface displays the total number of appointments of the patient. There is also an area chart and a bar chart that show the total number of appointments by month.

### 3.1.5.2 Manage Account Module (User Profile)

The screenshot displays the 'Manage Account Module (User Profile)' interface. At the top, the user's email 'example@gmail.com' and phone number '019-7980294' are shown. The navigation bar includes 'hiDoc', 'Home', 'Profile', 'Appointment', and 'Hi, Nadhirah v'. The main content area is titled 'Hi, Nadhirah' and contains a placeholder for a profile photo. Below the photo is a 'Choose File' button and a 'No file chosen' message. The form includes input fields for 'Name', 'Age', 'Gender', 'Email Address', and 'Phone Number'. A 'MEDICAL BACKGROUND' section contains 'Specialization' and 'Working Experience (Years)' fields. A 'HEALTH INFORMATION' section has a 'Health Details' field. A blue 'Update Profile' button is located at the bottom of the form. The footer contains 'hiDoc' branding, 'Useful Links' with placeholder text, and a large placeholder for an image or logo.

**Figure 1. 35 Patient Profile**

This is the profile interface for the patient. The patient's information can be edited here by clicking the "Update Profile" button.

### 3.1.5.3 Manage Appointment Booking Module (Make Appointment & Appointment History)

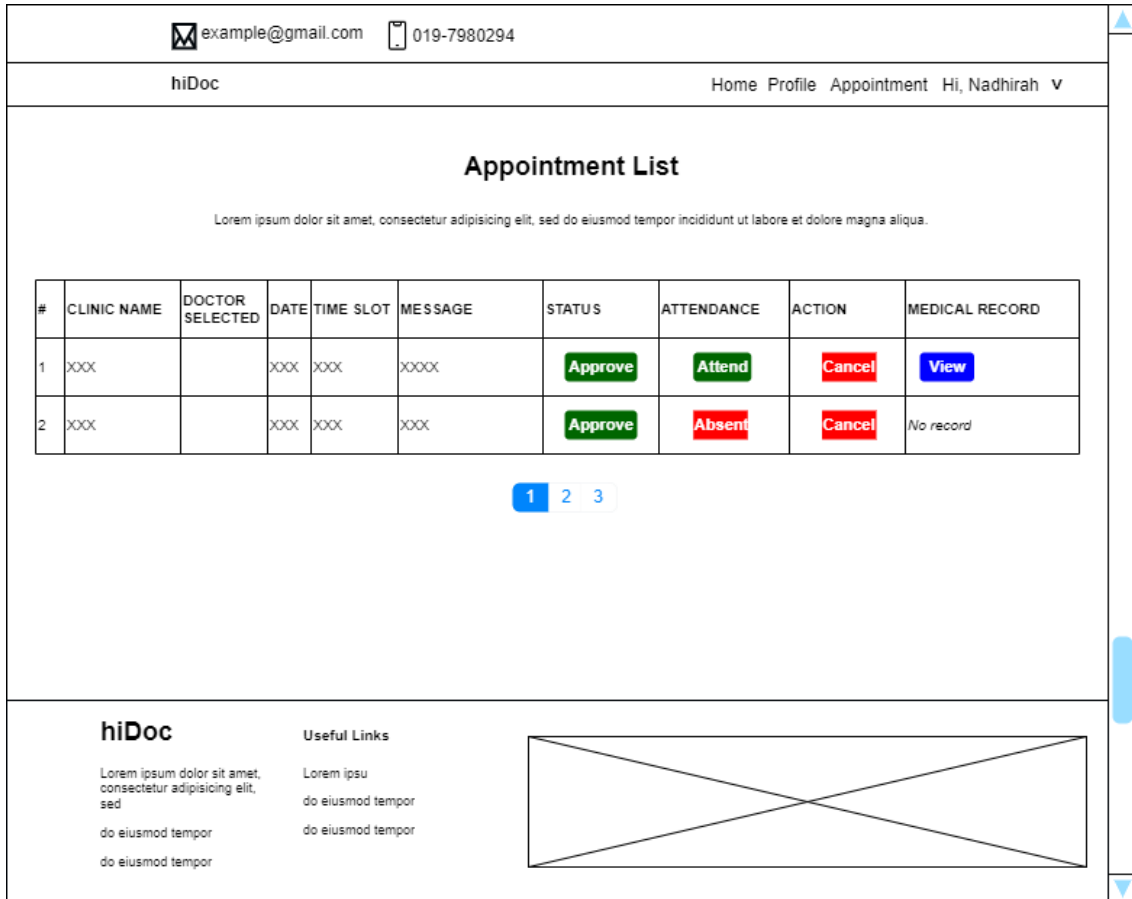


Figure 1. 36 Appointment List Patient Interface

This is the appointment list interface for the patient. This interface displays the patient's appointment list. The patient can see whether their appointment request has been approved or rejected. The patient can also view their medical records by clicking the "view" button in the "MEDICAL RECORD" column. The medical record can be seen at Figure 1.37 below. However, the button will be visible only if the patient attends the appointment. By clicking the "Cancel" button, you can also cancel the appointment.

3.1.5.4 Manage Medical Record Module (View)

MEDICAL RECORD	
Patient Name	xxxx
Clinic	xxxx
Doctor	xxxx
Date	xxxx
Timeslot	xxxx
Patient's Message	xxxxxx
Diagnosis	xxxxxx
Prescribed Medicine	xxxxxx

\_\_\_\_\_  
Authorised Signature

\_\_\_\_\_  
Name

\_\_\_\_\_  
Date

*Generated by,*

**hiDoc**

Figure 1. 37 Medical Record Interface

### 3.2 HARDWARE AND SOFTWARE SPECIFICATION

The following table shows the hardware requirements that will be used to access the Doctor Appointment Booking System (hiDoc).

**Table 1. 6 Hardware Specification**

<b>CPU</b>	<ol style="list-style-type: none"><li>1. AMD Ryzen 3 4300U with Radeon Graphics 2.70 GHz</li><li>2. Intel Pentium 4 Processor or later</li></ol>
------------	--------------------------------------------------------------------------------------------------------------------------------------------------

The following table shows the software requirements that will be used to access the Doctor Appointment Booking System (hiDoc).

**Table 1. 7 Software Specification**

<b>OPERATING SYSTEM</b>	Windows 10
<b>SERVER</b>	XAMPP v3.3.0
<b>PLATFORM</b>	PHP, Laravel, GitHub
<b>EDITOR</b>	Visual Studio Code 1.61
<b>BROWSER</b>	Google Chrome, Microsoft Edge
<b>DATABASE</b>	MySQL

## **APPENDIX C: SDD**

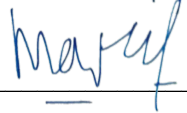
2023

# SOFTWARE DESIGN DESCRIPTION (SDD)

DOCTOR APPOINTMENT BOOKING WEB  
APPLICATION



## DOCUMENT APPROVAL

	Name	Date
<b>Authenticated by:</b>  _____ Muhammad' Arif Bin Mohamad	Muhammad' Arif Bin Mohamad	09/02/2023
<b>Approved by:</b> _____ Client		

Software :

Archiving Place :



## TABLE OF CONTENT

CONTENT	PAGE
<b>DOCUMENT APPROVAL .....</b>	<b>2</b>
<b>TABLE OF CONTENT.....</b>	<b>3</b>
<b>LIST OF FIGURES .....</b>	<b>5</b>
<b>1.1 PROJECT DESCRIPTION.....</b>	<b>7</b>
<b>1.2 SYSTEM IDENTIFICATION .....</b>	<b>8</b>
<b>1.3 ARCHITECTURE / BLUE PRINT .....</b>	<b>9</b>
<b>1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION .....</b>	<b>10</b>
1.1 Manage Account .....	10
1.2 Manage Appointment Booking.....	12
1.3 Manage Medical Record .....	14
1.4 Manage Medicine Information .....	15
1.5 Manage Clinic.....	17
<b>2.1 DETAILED DESCRIPTION .....</b>	<b>19</b>
1.6 Manage Account .....	19
1.6.1 Class 1: clinicProfile Interface .....	19
1.6.2 Class 2: clinicProfileDisplay Interface.....	19
1.6.3 Class 3: drProfile Interface .....	20
1.6.4 Class 4: drProfileDisplay Interface .....	20
1.6.5 Class 5: usermanagement Interface .....	21
1.6.6 Class 6: userProfile .....	21
1.6.7 Class 7: profileController.....	22
1.6.8 Class 8: User, Clinic, Doctor Model .....	26
1.7 Manage Appointment Booking.....	30
1.7.1 Class 9: appointment Interface .....	30
1.7.2 Class 10: appointmentstatus Interface .....	31
1.7.3 Class 11: cancelEmail Interface .....	32
1.7.4 Class 12: makeappointmentdr Interface.....	32
1.7.5 Class 13: updateStatus .....	33

1.7.6 Class 14: viewappointmentdoctor Interface..... 34

1.7.7 Class 15: viewappointmentuser Interface ..... 34

1.7.8 Class 16: AppointmentController..... 35

1.7.9 Class 17: User, Clinic, Doctor, Appointment Model..... 40

1.8 Manage Medical Record ..... 45

1.8.1 Class 18: addRecords Interface ..... 45

1.8.2 Class 19: viewRecords Interface..... 46

1.8.3 Class 20: RecordController..... 47

1.8.4 Class 21: Appointment, Records, Clinic, Doctor Model ..... 48

1.9 Manage Medicine Information ..... 50

1.9.1 Class 22: addMedicine Interface ..... 50

1.9.2 Class 23: editMedicine Interface..... 51

1.9.3 Class 24: Index Interface ..... 51

1.9.4 Class 25: search Interface ..... 52

1.9.5 Class 26: showMedicine Interface ..... 52

1.9.6 Class 27: userView Interface ..... 53

1.9.7 Class 28: medController ..... 53

1.9.8 Class 29: Medicine Model ..... 56

1.10 Manage Clinic..... 58

1.10.1 Class 30: clinicList Interface ..... 58

1.10.2 Class 31: drmanagement Interface ..... 59

1.10.3 Class 32: searchClinic Interface..... 60

1.10.4 Class 33: clinicController ..... 60

1.10.5 Class 34: User, Clinic, Doctor Model ..... 62

**2.2 DATA DICTIONARY..... 64**

1.10.1 users ..... 64

1.1.1.1 clinic ..... 65

1.1.1.2 doctor ..... 66

1.1.1.3 feedback..... 67

1.1.1.4 appointment ..... 67

1.1.1.5 records ..... 68

1.1.1.6 medicine..... 68

**LIST OF FIGURES**

Figure 0.1 Architecture .....	9
Figure 1.0.1 Manage Account Class Diagram.....	10
Figure 1.0.2 Manage Appointment Booking Class Diagram.....	12
Figure 1.0.3 Manage Medical Record Class Diagram.....	14
Figure 1.0.4 Manage Medicine Information Class Diagram.....	15
Figure 1.0.5 Manage Clinic Class Diagram.....	17

**LIST OF TABLES**

Table 0.1 Database Dictionary for ‘users’ .....	64
Table 0.2 Database Dictionary for ‘clinic’ .....	65
Table 0.3 Database Dictionary for ‘doctor’ .....	66
Table 0.4 Database Dictionary for ‘patient’ .....	67
Table 0.5 Database Dictionary for ‘appointment’ .....	67
Table 0.6 Database Dictionary for ‘records’ .....	68
Table 0.7 Database Dictionary for ‘medicine’ .....	68

## CHAPTER 1

### 1.1 PROJECT DESCRIPTION

Doctor Appointment Booking is a web-based project in which patients can book an appointment online from any clinic in the Pekan district. There is a list of doctors with their information. Patients can then select the doctor they want to see by providing the preferred date and time.

The admin, doctor, clinic staff or representative, and patient will be the four users of this project. Manage account, manage appointment booking, manage medical record, manage medicine information, and manage clinic are the five modules in this system.

Users can access and change their profiles in the Manage Account module. On the other hand, the admin can delete any existing doctor accounts. Users can view the clinic and doctor information as well.

The Manage Appointment Booking module allows the patient to schedule an appointment with the clinic or doctor that they want. The doctor, clinic staff and patient can view their appointment history records. Furthermore, the doctor has the authority to update the attendance status while the clinic staff can also update the attendance status as well as the appointment request.

The manage medical record module can be access by the patient, clinic staff and doctor. The clinic staff will add the medical record only if the patient's attend the appointment session. The clinic staff can also view and delete the medical record. The doctor and patient can view the medical record of each appointment session.

In manage medicine information module, the admin can add a new medicine, view existing medicine, and delete any drug, while the other users can view and search about the medicine.

The manage clinic module is the final module. This is where the clinic information will be displayed at. All the users can view the clinic information. In this module also, the clinic

staff can remove the doctor account that works under them while the doctor can change their workplace details. This module is connected to the manage appointment booking module. This module will allow the users to proceed to make an appointment.

## 1.2 SYSTEM IDENTIFICATION

System Title: Doctor Appointment Booking Web Application

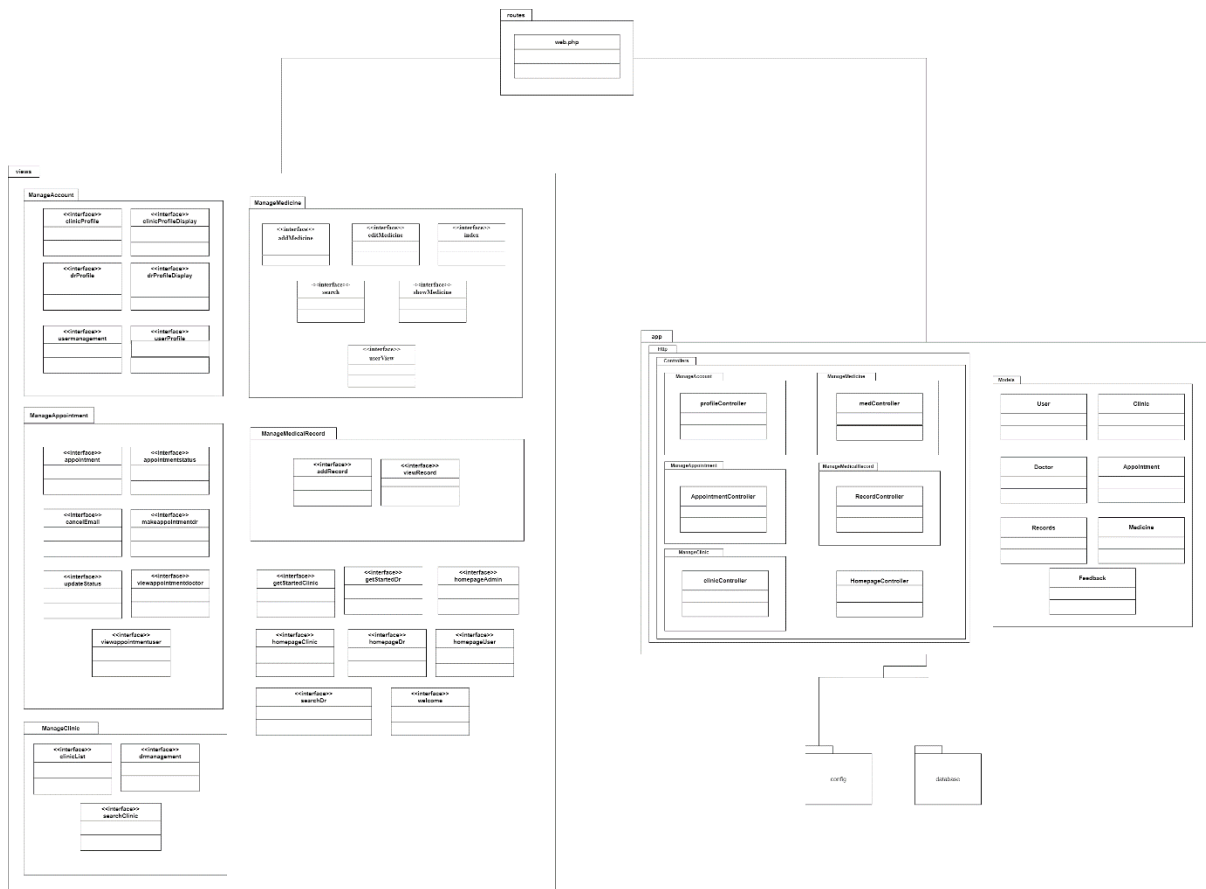
System Abbreviation: hiDoc

Year: 2023

Version: Version 1

System Identification Number: SDD-hiDoc-2023-V1

### 1.3 ARCHITECTURE / BLUE PRINT

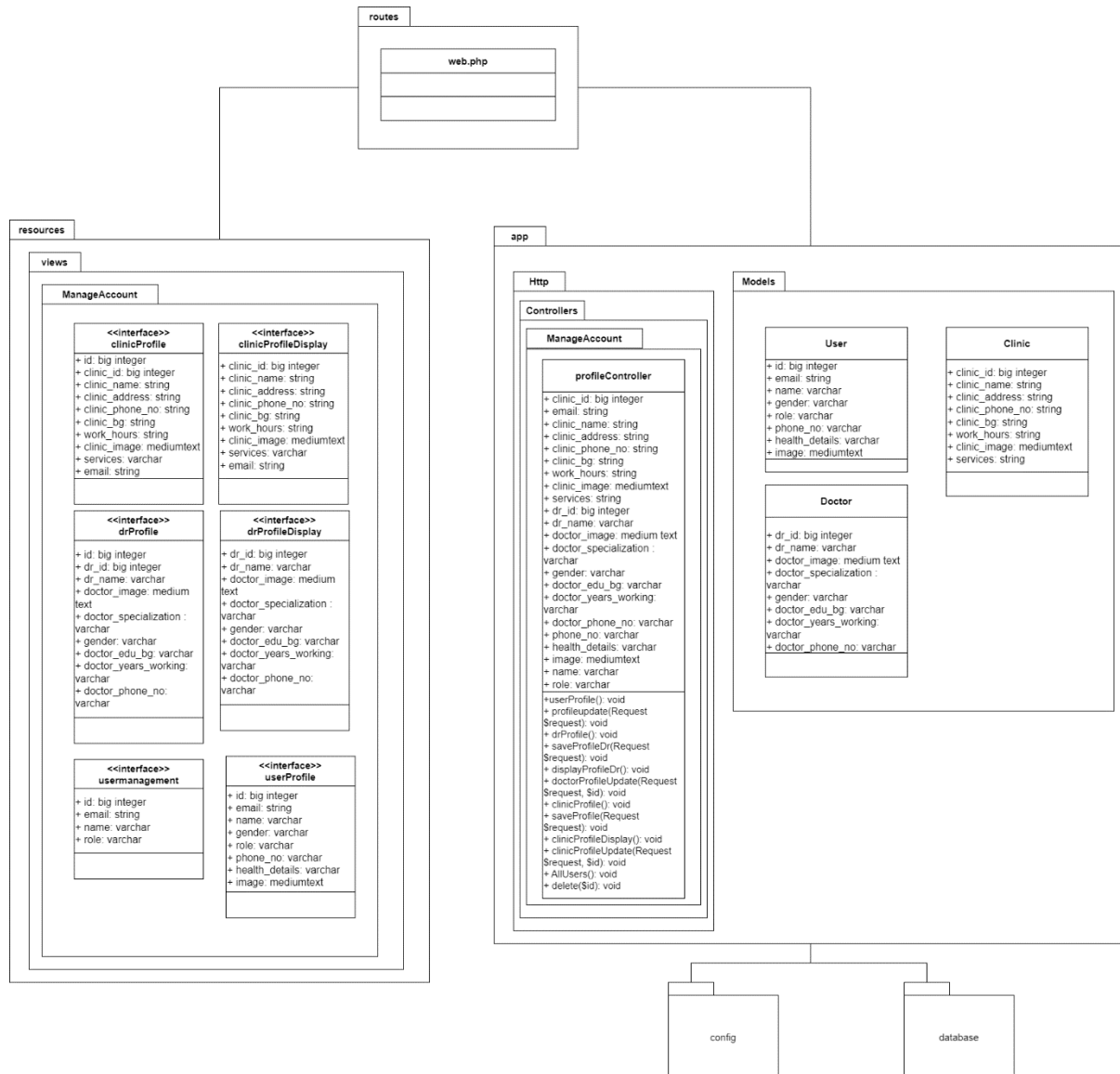


**Figure 0.1 Architecture**

This application makes use of a framework that includes a model, a view, and a controller. Login, manage homepage, manage account, manage medical record, manage clinic, manage appointment booking, and manage medicine information are the seven modules in view. Except for login and manage homepage, the major modules that will be explained in detail in this SDD are all of them. Each of these modules has its own model and controller, as shown in the above architecture diagram.

## 1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

### 1.1 Manage Account



**Figure 1.0.1 Manage Account Class Diagram**

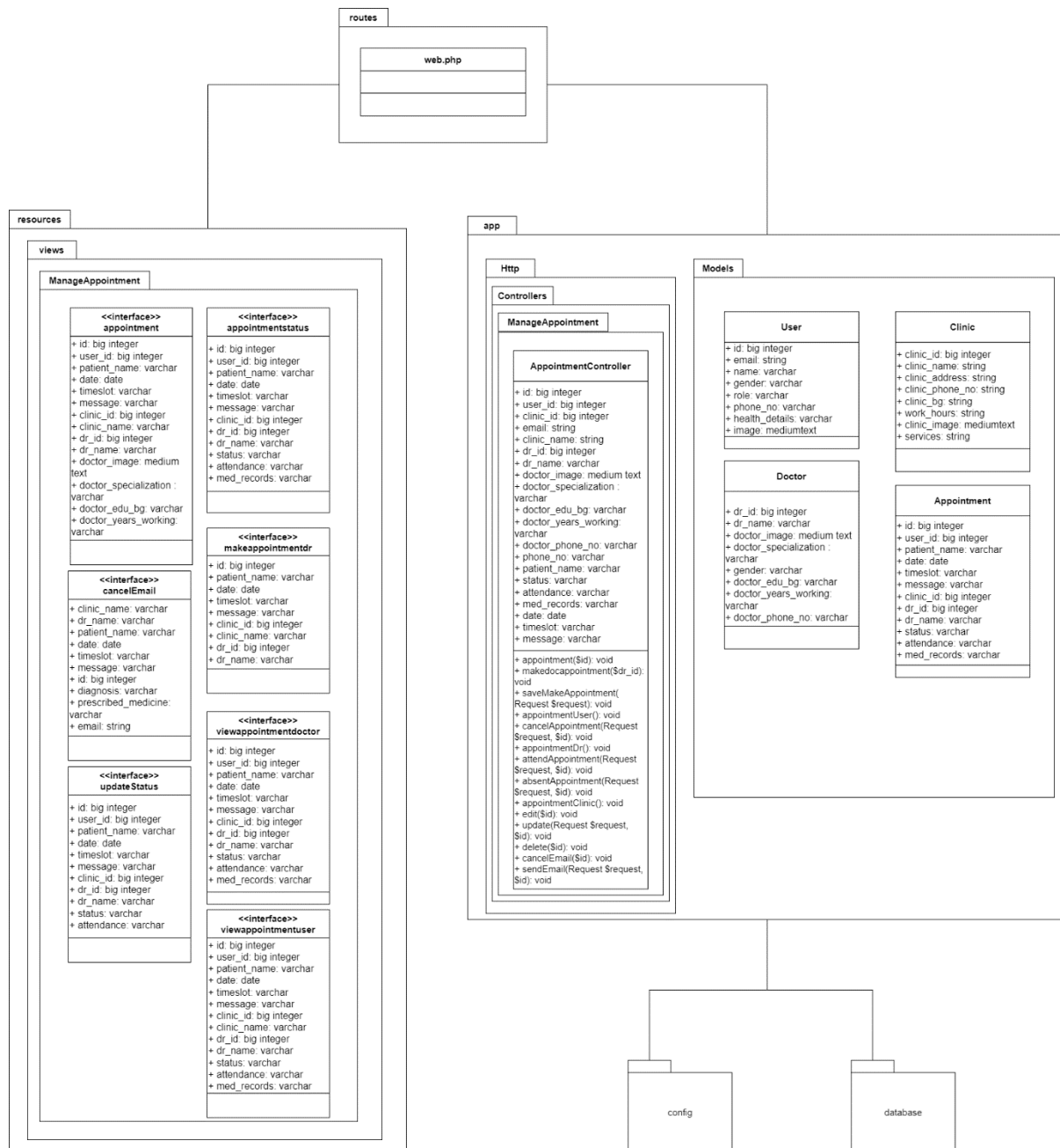
In manage account module, there are many functions that will be perform. All details of these function can be seen in Manage Account Use Case Diagram and Sequence Diagram in SRS. There are a total of 12 functions.

1. **userProfile():** This method is to display the patient's information that are stored inside the database.
2. **profileupdate(Request \$request):** This method is to allow the patient to edit their patient's information that are stored inside the database.



3. drProfile(): This method is to allow the doctor to input their information.
4. saveProfileDr(Request \$request): This method is to store the data that the doctor input into the database.
5. displayProfileDr(): This method is to display the doctor's information that are stored inside the database.
6. doctorProfileUpdate(Request \$request, \$id): This method is to allow the doctor to edit their profile information that are stored inside the database.
7. clinicProfile(): This method is to allow the doctor to input their information.
8. saveProfile(Request \$request): This method is to store the data that the clinic staff input into the database.
9. clinicProfileDisplay(): This method is to display the doctor's information that are stored inside the database.
10. clinicProfileUpdate(Request \$request, \$id): This method is to allow the doctor to edit their profile information that are stored inside the database.
11. AllUsers(): This method is to display the user's information that are stored inside the database.
12. delete(\$id): This method is to allow the admin to delete any user's account.

## 1.2 Manage Appointment Booking



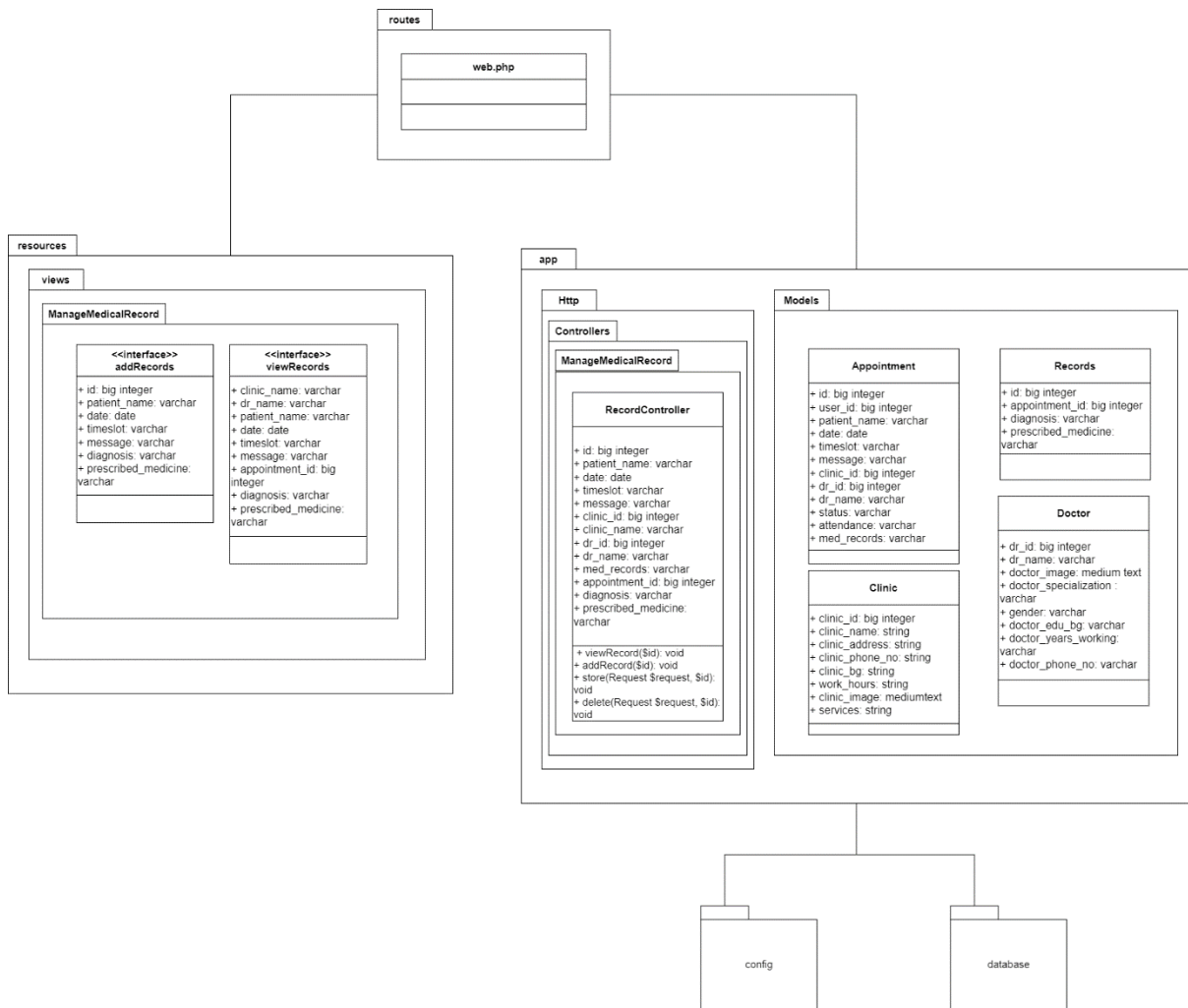
**Figure 1.0.2 Manage Appointment Booking Class Diagram**

In manage appointment booking module, there are many functions that will be perform. All details of these function can be seen in Manage Appointment Booking Use Case Diagram and Sequence Diagram in SRS. There are a total of 14 functions.

1. `appointment($id)`: This method is to display the list of doctors in each clinic.

2. `makedocappointment($dr_id)`: This method is to allow the user to input an appointment request based on the selected doctor.
3. `saveMakeAppointment(Request $request)`: This method is to save the appointment into the database.
4. `appointmentUser()`: This method is to display the appointment list made by the patient.
5. `cancelAppointment(Request $request, $id)`: This method is to allow the patient to cancel the appointment.
6. `appointmentDr()`: This method is to display the appointment for doctor.
7. `attendAppointment(Request $request, $id)`: This method is to allow the doctor to change the attendance status to “Attend”.
8. `absentAppointment(Request $request, $id)`: This method is to allow the doctor to change the attendance status to “Absent”.
9. `appointmentClinic()`: This method is to display the appointment list for the clinic.
10. `edit($id)`: This method is to allow the clinic staff to update the appointment status.
11. `update(Request $request, $id)`: This method is to store the updated appointment status into the database.
12. `delete($id)` : This method is to allow the clinic staff to delete any appointment.
13. `cancelEmail($id)`: This method is for sending an email appointment cancellation to the patient.
14. `sendEmail(Request $request, $id)`: This method is to send the cancellation appointment email to the patient.

### 1.3 Manage Medical Record



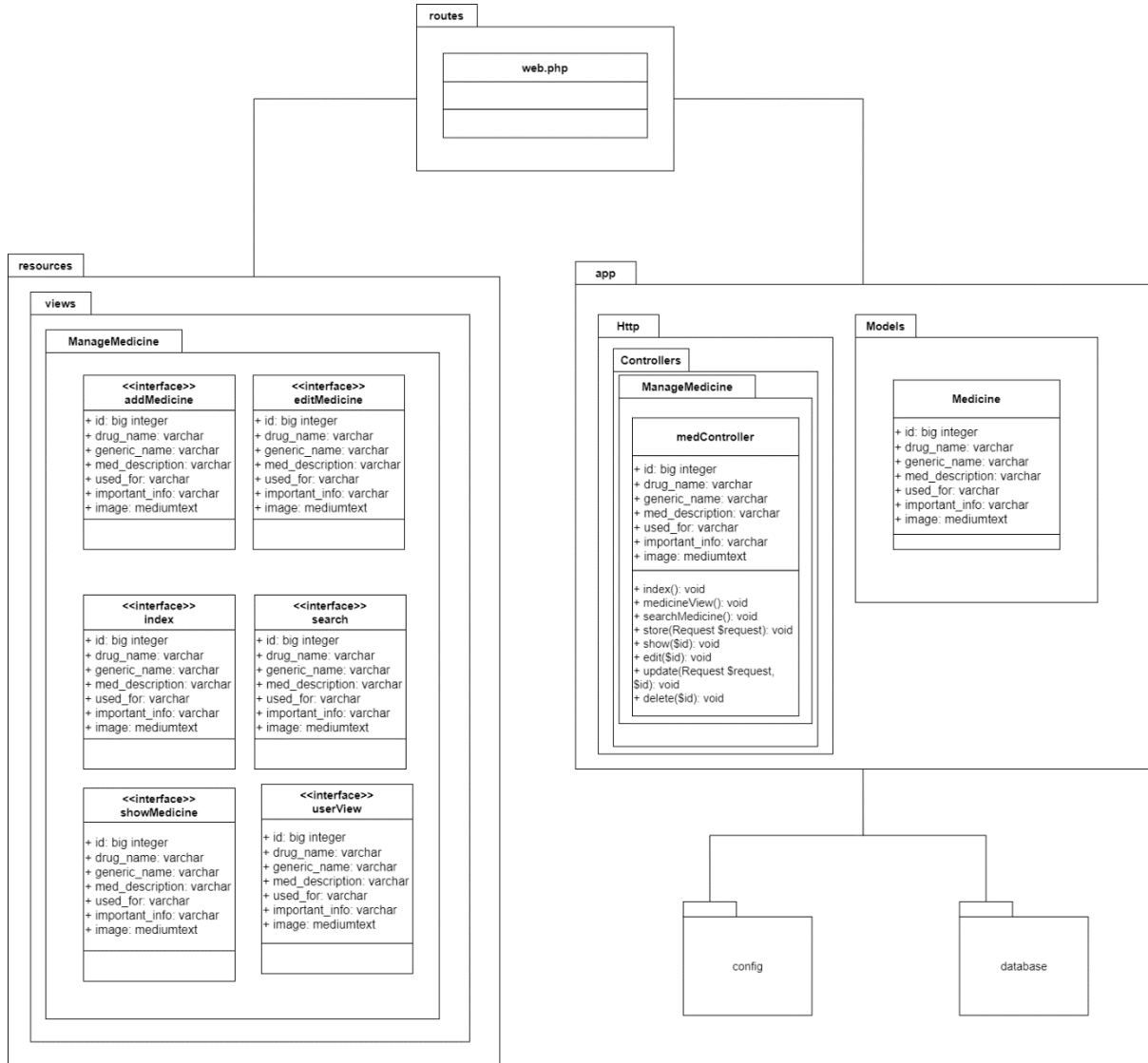
**Figure 1.0.3 Manage Medical Record Class Diagram**

In manage medical record module, there are a few functions that will be perform. All details of these function can be seen in Manage Diabetes Prediction Use Case Diagram and Sequence Diagram in SRS. There are a total of 4 functions.

1. viewRecord(\$id): This method is to display the medical record of each successful appointment session.
2. addRecord(\$id): This method is to allow the clinic staff to input the diagnosed from the appointment session.
3. store(Request \$request, \$id): This method is to store the input from addRecord() into the database.

4. delete(Request \$request, \$id): This method is to allow the clinic staff to delete the medical record.

### 1.4 Manage Medicine Information



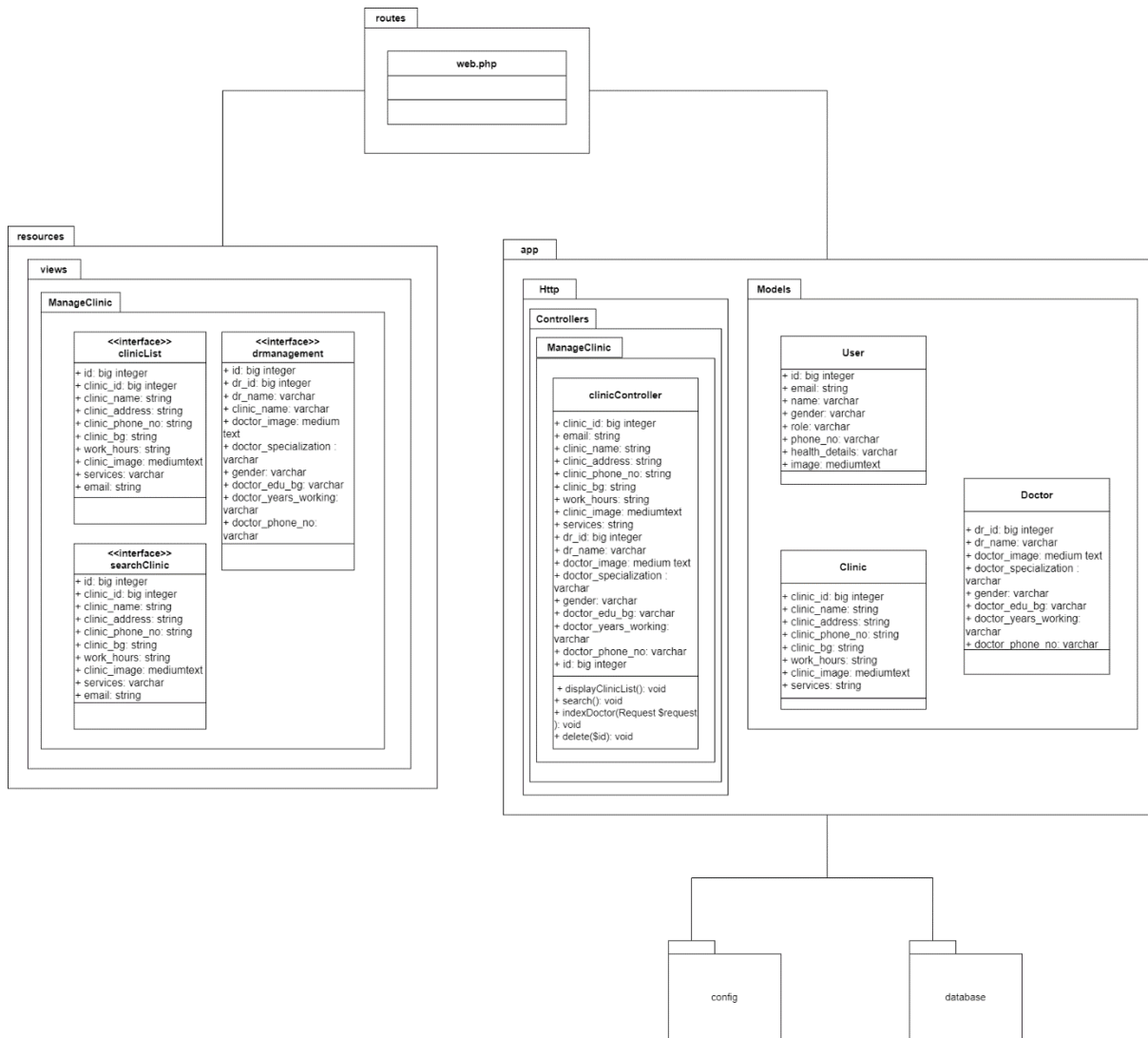
**Figure 1.04 Manage Medicine Information Class Diagram**

In manage medicine information module, there are many functions that will be perform. All details of these function can be seen in Manage Medicine Information Use Case Diagram and Sequence Diagram in SRS. There are a total of 8 functions.

1. index(): This method is to display the medicine information that are stored inside the database.

2. `medicineView()`: This method is to allow the users to view the medicine information in full details.
3. `searchMedicine()`: This method is to allow all the users to search the medicine by name.
4. `store(Request $request)`: This method is to store the medicine data input by the admin into the database.
5. `show($id)`: This method is to allow the admin to view the medicine information in full details.
6. `edit($id)`: This method is to allow the admin to edit the medicine information.
7. `update(Request $request, $id)`: This method is to save the updated data in edit() in the database.
8. `delete($id)` : This method is to allow the admin to delete any medicine information in the database.

## 1.5 Manage Clinic



**Figure 1.0.5 Manage Clinic Class Diagram**

In manage clinic module, there are a few functions that will be perform. All details of these function can be seen in Manage Clinic Use Case Diagram and Sequence Diagram in SRS. There are a total of 4 functions.

1. `displayClinicList()`: This method is to display all the list of clinics that are stored inside the database.
2. `search()` :This method is to allow the users to search for the clinic by name
3. `indexDoctor(Request $request)`: This method is to display the doctor that works under that particular clinic.

4. delete(\$id) This method is to remove the doctor's account that works under the clinic



## CHAPTER 2

## 2.1 DETAILED DESCRIPTION

## 1.6 Manage Account

## 1.6.1 Class 1: clinicProfile Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow clinic staff to input clinic information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar
	work_hours	Varchar
	clinic_image	Medium Text
	services	Varchar
	email	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

## 1.6.2 Class 2: clinicProfileDisplay Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to display and edit the clinic information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar

	work_hours	Varchar
	clinic_image	Medium Text
	services	Varchar
	email	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.6.3 Class 3: drProfile Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow doctor to input clinic information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.6.4 Class 4: drProfileDisplay Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to display and edit the doctor information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	dr_id	Big integer
	dr_name	Varchar

	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.6.5 Class 5: usermanagement Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the admin to view the user's account and delete them.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	dr_id	Big integer
	name	Varchar
	email	Varchar
	role	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.6.6 Class 6: userProfile

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to display and edit the patient information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	name	Varchar
	email	Varchar
	gender	Varchar

	role	Varchar
	phone_no	Varchar
	health_details	Varchar
	image	Medium Text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.6.7 Class 7: profileController

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to manage the functions for the manage account.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	name	Varchar
	email	Varchar
	role	Varchar
	phone_no	Varchar
	health_details	Varchar
	image	Medium Text
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
doctor_edu_bg	Varchar	

	doctor_years_working	Varchar
	doctor_phone_no	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	userProfile()	This method is to display the patient's information that are stored inside the database.
	profileupdate(Request \$request)	This method is to allow the patient to edit their information that are stored inside the database.
	drProfile()	This method is to allow the doctor to input their information.
	saveProfileDr(Request \$request)	This method is to store the data that the doctor input into the database.
	displayProfileDr()	This method is to display the doctor's information that are stored inside the database.
	doctorProfileUpdate(Request \$request, \$id)	This method is to allow the doctor to edit their profile information that are stored inside the database.
	clinicProfile()	This method is to allow the doctor to input their information.
	saveProfile(Request \$request)	This method is to store the data that the doctor input into the database.
	clinicProfileDisplay()	This method is to display the doctor's information that are stored inside the database.
	clinicProfileUpdate(Request \$request, \$id)	This method is to allow the doctor to edit their profile

		information that are stored inside the database.
	AllUsers()	This method is to display the user's information that are stored inside the database.
	delete(\$id)	This method is to allow the admin to delete any user's account.
<b>Algorithm</b>	<p><b>userProfile()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. userProfile() is called</li> <li>3. Go to &lt;&lt;userProfile&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>profileupdate(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. profileUpdate() is called</li> <li>3. Create object "user" to connect to User Model</li> <li>4. REDIRECTED to &lt;&lt;userProfile&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>drProfile()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. drProfile() is called</li> <li>3. Create object "findClinic" to connect to Clinic Model</li> <li>4. Go to saveProfile()</li> <li>5. END</li> </ol> <p><b>saveProfileDr(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. saveProfileDr() is called</li> <li>3. Doctor model is connected</li> <li>4. REDIRECTED to &lt;&lt;drProfileDisplay&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>displayProfileDr()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> </ol>	

2. displayProfileDr() is called
3. Create object “doctor” to connect with Doctor, User and Clinic Model.
4. REDIRECTED to <<drProfileDisplay>> interface
5. END

**doctorProfileUpdate(Request \$request, \$id)**

1. BEGIN
2. doctorProfileUpdate() is called
3. Create object “profile” to connect to Doctor Model
4. REDIRECTED to <<drProfileDisplay>> interface
5. END

**clinicProfile()**

1. BEGIN
2. clinicProfile() is called
3. Go to saveProfile()
4. END

**saveProfile(Request \$request)**

1. BEGIN
2. saveProfile() is called
3. REDIRECTED to <<clinicProfileDisplay>> interface
4. END

**clinicProfileDisplay()**

1. BEGIN
2. clinicProfileDisplay() is called
3. Create object “profile” to connect to User and Clinic Model
4. Go to <<clinicProfileDisplay>> Interface
5. END

**clinicProfileUpdate(Request \$request, \$id)**

1. BEGIN
2. clinicProfileUpdate() is called
3. Create object “profile” to connect to Clinic Model
4. REDIRECTED to <<clinicProfileDisplay>> interface
5. END

	<p><b>AllUsers()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. AllUsers() is called</li> <li>3. Create object “user_table” to connect to User Model</li> <li>4. Go to usermanagement interface</li> <li>5. END</li> </ol> <p><b>delete(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. Delete() is called</li> <li>3. Create object “user_table” to connect to User Model</li> <li>4. REDIRECTED to &lt;&lt;usermanagement&gt;&gt; interface</li> <li>5. END</li> </ol>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.6.8 Class 8: User, Clinic, Doctor Model

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to store, validate, and retrieve the manage account data in the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	name	Varchar
	email	Varchar
	role	Varchar
	phone_no	Varchar
	health_details	Varchar
	image	Medium Text
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
doctor_years_working	Varchar	



	doctor_phone_no	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	userProfile()	This method is to display the patient's information that are stored inside the database.
	profileupdate(Request \$request)	This method is to allow the patient to edit their patient's information that are stored inside the database.
	drProfile()	This method is to allow the doctor to input their information.
	saveProfileDr(Request \$request)	This method is to store the data that the doctor input into the database.
	displayProfileDr()	This method is to display the doctor's information that are stored inside the database.
	doctorProfileUpdate(Request \$request, \$id)	This method is to allow the doctor to edit their profile information that are stored inside the database.
	clinicProfile()	This method is to allow the doctor to input their information.
	saveProfile(Request \$request)	This method is to store the data that the doctor input into the database.

	clinicProfileDisplay()	This method is to display the doctor's information that are stored inside the database.
	clinicProfileUpdate(Request \$request, \$id)	This method is to allow the doctor to edit their profile information that are stored inside the database.
	AllUsers()	This method is to display the user's information that are stored inside the database.
	delete(\$id)	This method is to allow the admin to delete any user's account.
<b>Algorithm</b>	<p><b>userProfile()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. DISPLAY &lt;&lt;userProfile&gt;&gt; interface</li> <li>3. END</li> </ol> <p><b>profileupdate(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. EDIT 'name', 'email', 'gender', 'phone_no', 'health_details', 'image'</li> <li>3. REDIRECTED to &lt;&lt;userProfile&gt;&gt; interface</li> <li>4. DISPLAY success message</li> <li>5. END</li> </ol> <p><b>drProfile()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT 'clinic_id, 'clinic_name'</li> <li>3. DISPLAY &lt;&lt;drProfile&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>saveProfileDr(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. INSERT 'dr_id', 'doctor_image', 'dr_name', 'doctor_phone_no', 'gender', 'doctor_specialization', 'doctor_years_working', 'doctor_edu_bg', 'clinic_id'</li> <li>3. Go to &lt;&lt;drProfileDisplay&gt;&gt; interface</li> </ol>	

4. DISPLAY success message

5. END

**displayProfileDr()**

1. BEGIN

2. GET 'dr\_id' = id

3. SELECT 'doctor.dr\_id', 'doctor.doctor\_image',  
'doctor.dr\_name', 'doctor.doctor\_phone\_no', 'doctor.gender',  
'doctor.doctor\_specialization', 'doctor.doctor\_years\_working',  
'doctor.doctor\_edu\_bg', 'doctor.clinic\_id', 'clinic.clinic\_name'  
WHERE 'doctor.dr\_id' == 'dr\_id'

4. Go to <<drProfileDisplay>> interface

5. END

**doctorProfileUpdate(Request \$request, \$id)**

1. BEGIN

2. FIND 'id'

3. INSERT 'dr\_name', 'doctor\_specialization',  
'doctor\_years\_working', 'doctor\_edu\_bg', 'doctor\_phone\_no',  
'doctor\_edu\_bg', 'gender', 'doctor\_image'

4. REDIRECTED TO <<drProfileDisplay>> INTERFACE

5. DISPLAY success message

6. END

**clinicProfile()**

1. BEGIN

2. DISPLAY <<clinicProfile>> interface

3. END

**saveProfile(Request \$request)**

1. BEGIN

2. INSERT 'clinic\_id', 'clinic\_image', 'clinic\_name', 'clinic  
\_phone\_no', 'clinic\_address', 'clinic\_bg', 'services',  
'work\_hours'

3. Go to <<clinicProfileDisplay>> interface

4. DISPLAY success message

5. END

	<p><b>clinicProfileDisplay()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. GET 'clinic_id' = id</li> <li>3. SELECT 'clinic.clinic_id', 'clinic.clinic_image', 'clinic.clinic_name', 'clinic.clinic_phone_no', 'clinic.services', 'clinic.clinic_bg' WHERE 'clinic.clinic_id' == 'clinic_id'</li> <li>4. Go to &lt;&lt;clinicProfileDisplay&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>clinicProfileUpdate(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. FIND 'id'</li> <li>3. INSERT clinic_name, work_hours, clinic_phone_no, clinic_address, clinic_bg, services, clinic_image</li> <li>4. REDIRECTED TO &lt;&lt;clinicProfileDisplay&gt;&gt; INTERFACE</li> <li>5. DISPLAY success message</li> <li>6. END</li> </ol> <p><b>AllUsers()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT all FROM USER model</li> <li>3. Go to &lt;&lt;usermanagement&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>delete(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. FIND 'id'</li> <li>3. DELETE data in USER model WHERE 'id' == 'id'</li> <li>4. REDIRECTED to &lt;&lt;usermanagement&gt;&gt; interface</li> <li>5. END</li> </ol>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 1.7 Manage Appointment Booking

### 1.7.1 Class 9: appointment Interface

<b>Class Type</b>	Boundary Class
<b>Responsibility</b>	This class is to allow the patient to make an appointment.

Attributes	Attributes Name	Attributes Name
	id	Big integer
	user_id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	clinic_name	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	doctor_edu_bg	Varchar
doctor_years_working	Varchar	
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

### 1.7.2 Class 10: appointmentstatus Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the clinic staff to view their appointment list or history.	
Attributes	Attributes Name	Attributes Name
	id	Big integer
	user_id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	dr_id	Big integer

	dr_name	Varchar
	status	Varchar
	attendance	Varchar
	med_records	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.7.3 Class 11: cancelEmail Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This method is to allow the clinic staff to send a cancellation appointment email to the patient.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	clinic_name	Varchar
	dr_name	Varchar
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	id	Big integer
	diagnosis	Varchar
	prescribed_medicine	Varchar
	email	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.7.4 Class 12: makeappointmentdr Interface

<b>Class Type</b>	Boundary Class
<b>Responsibility</b>	This class is to allow the patient to make an appointment with doctor option selection.

Attributes	Attributes Name	Attributes Name
	id	Big integer
	user_id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	clinic_name	Varchar
	dr_id	Big integer
dr_name	Varchar	
Methods	Method Name	Description
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

### 1.7.5 Class 13: updateStatus

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the clinic staff to update the appointment request and attendance status.	
Attributes	Attributes Name	Attributes Name
	id	Big integer
	user_id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	dr_id	Big integer
	dr_name	Varchar
	status	Varchar
	attendance	Varchar
Methods	Method Name	Description

	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.7.6 Class 14: viewappointmentdoctor Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the doctor to view their appointment list or history.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	user_id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	dr_id	Big integer
	dr_name	Varchar
	status	Varchar
	attendance	Varchar
med_records	Varchar	
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.7.7 Class 15: viewappointmentuser Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the user or patient to view their appointment list or history.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	user_id	Big integer



	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	clinic_id	Big integer
	clinic_name	Varchar
	dr_id	Big integer
	dr_name	Varchar
	status	Varchar
	attendance	Varchar
	med_records	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.7.8 Class 16: AppointmentController

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to manage the functions for the manage appointment booking.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	user_id	Big integer
	clinic_id	Big integer
	email	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
	phone_no	Varchar

	patient_name	Varchar
	status	Varchar
	attendance	Varchar
	med_records	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	appointment(\$id)	This method is to display the list of doctors in each clinic.
	makedocappointment(\$dr_id)	This method is to allow the user to input an appointment request based on the selected doctor.
	saveMakeAppointment(Request \$request)	This method is to save the appointment into the database,
	appointmentUser()	This method is to display the appointment list made by the patient.
	cancelAppointment(Request \$request, \$id)	This method is to allow the patient to cancel the appointment.
	appointmentDr()	This method is to display the appointment for doctor.
	attendAppointment(Request \$request, \$id)	This method is to allow the doctor to change the attendance status to "Attend".
	absentAppointment(Request \$request, \$id)	This method is to allow the doctor to change the attendance status to "Absent".
	appointmentClinic()	This method is to display the appointment list for the clinic.
	edit(\$id)	This method is to allow the clinic staff to update the appointment status of the clinic.

	update(Request \$request, \$id)	This method is to store the updated appointment status into the database.
	delete(\$id)	This method is to allow the clinic staff to delete any appointment.
	cancelEmail(\$id)	This method is for the cancellation email view.
	sendEmail(Request \$request, \$id)	This method is to send the cancellation appointment email to the patient.
<b>Algorithm</b>	<p><b>appointment(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. appointment() is called</li> <li>3. Create object “display” to connect to User, Clinic, Doctor Model</li> <li>4. Create object “list” to connect to Doctor, Clinic Model</li> <li>5. Create object “clinic” to connect to Clinic, Doctor Model</li> <li>6. Go to &lt;&lt;appointment&gt;&gt; interface</li> <li>7. END</li> </ol> <p><b>makedocappointment(\$dr_id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. makedocappointment() is called</li> <li>3. Create object “drlist” to connect to Doctor, Clinic Model</li> <li>4. Go to &lt;&lt;makeappointmentdr&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>saveMakeAppointment(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. saveMakeAppointment() is called</li> <li>3. Connected to Appointment Model</li> <li>4. REDIRECTED to &lt;&lt;viewappointmentuser&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>appointmentUser()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. appointmentUser() is called</li> </ol>	

3. create object “appointment\_table\_user” to connect to Appointment, Clinic Model
4. Go to <<viewappointmentuser>> interface
5. END

**cancelAppointment(Request \$request, \$id)**

1. BEGIN
2. cancelAppointment() is called
3. create object “cancel” to connect to Appointment Model
4. REDIRECTED to <<viewappointmentuser>> interface
5. END

**appointmentDr()**

1. BEGIN
2. appointmentDr() is called
3. Create object “Appointment\_table\_doctor” to connect to Appointment Model
4. Go to <<viewappointmentdr>> interface
5. END

**attendAppointment(Request \$request, \$id)**

1. BEGIN
2. attendAppointment() is called
3. Create object “attend” to connect to Appointment Model
4. REDIRECTED to <<viewappointmentdr >>interface
5. END

**absentAppointment(Request \$request, \$id)**

1. BEGIN
2. absentAppointment() is called
3. Create object “absent” to connect to Appointment Model
4. REDIRECTED to <<viewappointmentdr>> interface
5. END

**appointmentClinic()**

1. BEGIN
2. appointmentClinic() is called

3. create object “appointment\_table\_clinic” to connect to Appointment, User Model
4. Go to <<appointmentstatus>> interface
5. END

**edit(\$id)**

1. BEGIN
2. edit() is called
3. Create object “appointmentedit” to connect to Appointment Model
4. Create object “drlist” to connect to Doctor, Clinic Model
5. Go to <<updatestatus>> interface
6. END

**update(Request \$request, \$id)**

1. BEGIN
2. update() is called
3. Create object “appointmentupdate” to connect to Appointment Model
4. REDIRECTED to <<appointmentstatus>> interface
5. END

**delete(\$id)**

1. BEGIN
2. delete() is called
3. Create object “AppointmentDelete” to connect to Appointment Model
4. REDIRECTED to <<appointmentstatus>> interface
5. END

**cancelEmail(\$id)**

1. BEGIN
2. cancelEmail() is called
3. Create object “Email” to connect to Appointment, User, Clinic Model
4. Go to <<cancelEmail>> interface
5. END

	<p><b>sendEmail(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. sendEmail() is called</li> <li>3. Create object “Email” to connect to Appointment, User, Clinic Model</li> <li>4. REDIRECTED to &lt;&lt;appointmentstatus&gt;&gt; interface</li> <li>5. END</li> </ol>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.7.9 Class 17: User, Clinic, Doctor, Appointment Model

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to store, validate, and retrieve the manage appointment booking data in the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Name</b>
	id	Big integer
	user_id	Big integer
	clinic_id	Big integer
	email	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
	phone_no	Varchar
	patient_name	Varchar
	status	Varchar
	attendance	Varchar
med_records	Varchar	
date	Date	

	timeslot	Varchar
	message	Varchar
<b>Metho ds</b>	<b>Method Name</b>	<b>Description</b>
	appointment(\$id)	This method is to display the list of doctors in each clinic.
	makedocappointment(\$dr_id)	This method is to allow the user to input an appointment request based on the selected doctor.
	saveMakeAppointment(Request \$request)	This method is to save the appointment into the database,
	appointmentUser()	This method is to display the appointment list made by the patient.
	cancelAppointment(Request \$request, \$id)	This method is to allow the patient to cancel the appointment.
	appointmentDr()	This method is to display the appointment for doctor.
	attendAppointment(Request \$request, \$id)	This method is to allow the doctor to change the attendance status to “Attend”.
	absentAppointment(Request \$request, \$id)	This method is to allow the doctor to change the attendance status to “Absent”.
	appointmentClinic()	This method is to display the appointment list for the clinic.
	edit(\$id)	This method is to allow the clinic staff to update the appointment status of the clinic.
	update(Request \$request, \$id)	This method is to store the updated appointment status into the database.
	delete(\$id)	This method is to allow the clinic staff to delete any appointment.

	cancelEmail(\$id)	This method is for the cancellation email view.
	sendEmail(Request \$request, \$id)	This method is to send the cancellation appointment email to the patient.
<b>Algorithm</b>	<p><b>appointment(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT 'doctor.dr_id','doctor.dr_name','doctor.doctor_image','doctor.doctor_specialization','doctor.doctor_years_working','doctor.doctor_edu_bg', 'doctor.doctor_phone_no','doctor.clinic_id','clinic.clinic_name' WHERE 'clinic.clinic_id' == id)</li> <li>3. SELECT all WHERE 'doctor.clinic_id' == id</li> <li>4. Go to &lt;&lt;appointment&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>makedocappointment(\$dr_id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT 'clinic.clinic_id','clinic.clinic_name','doctor.dr_id','doctor.dr_name' WHERE 'doctor.dr_id' == dr_id</li> <li>3. Go to &lt;&lt;makeappointmentdr&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>saveMakeAppointment(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. INSERT 'user_id', 'patient_name', 'clinic_id', 'clinic_name', 'dr_id', 'dr_name', 'date', 'timeslot', 'message'</li> <li>3. REDIRECTED to &lt;&lt;viewappointmentuser&gt;&gt; interface</li> <li>4. DISPLAY success message</li> <li>5. END</li> </ol> <p><b>appointmentUser()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. 'userid' = id</li> </ol>	



3. SELECT  
'appointment.id','appointment.clinic\_id','clinic.clinic\_name','appointment.dr\_id','appointment.date','appointment.timeslot','appointment.message','appointment.status',  
'appointment.attendance','appointment.dr\_name','appointment.med\_records' WHERE 'appointment.user\_id' == 'userid'
4. Go to <<viewappointmentuser>> interface
5. END

#### **cancelAppointment(Request \$request, \$id)**

1. BEGIN
2. FIND 'id'
3. FIND id in Appointment Model
4. UPDATE 'status'=>'Cancel', 'attendance'=>'Cancel'
5. REDIRECTED to <<viewappointmentuser>> interface
6. DISPLAY success message
7. END

#### **appointmentDr()**

1. BEGIN
2. GET 'dr\_id' = id
3. SELECT  
'id','patient\_name','date','timeslot','message','status','attendance','med\_records' WHERE 'dr\_id' == dr\_id ORWHERE 'dr\_name'== name
4. Go to <<viewappointmentdoctor>> interface
5. END

#### **attendAppointment(Request \$request, \$id)**

1. BEGIN
2. FIND 'id'
3. FIND in Appointment Model where 'id' == id
4. UPDATE 'attendance'=>'Attend'
5. REDIRECTED to <<viewappointmentdoctor>> interface
6. DISPLAY success message
7. END

**absentAppointment(Request \$request, \$id)**

1. BEGIN
2. FIND 'id'
3. FIND in Appointment Model where 'id' == id
4. UPDATE 'attendance'=>'Absent'
5. REDIRECTED to <<viewappointmentdoctor>> interface
6. DISPLAY success message
7. END

**appointmentClinic()**

1. BEGIN
2. GET 'clinic\_id' = id
3. SELECT  
'doctor.dr\_id','doctor.dr\_name','doctor.doctor\_image','doctor.doctor\_specialization','doctor.doctor\_years\_working','doctor.doctor\_edu\_bg',  
doctor.doctor\_phone\_no','doctor.clinic\_id','clinic.clinic\_name' WHERE  
'clinic.clinic\_id' == 'clinic\_id'
4. SELECT all FROM Clinic Model where 'clinic\_id' == clinic\_id
5. END

**edit(\$id)**

1. BEGIN
2. SELECT  
'id','date','patient\_name','timeslot','message','status','attendance','dr\_id','dr\_id',  
'dr\_name' WHERE 'appointment.id' == id
3. SELECT 'doctor.dr\_id','doctor.dr\_name'
4. Go to update()
5. END

**update(Request \$request, \$id)**

1. BEGIN
2. FIND 'id'
3. UPDATE all in Appointment Model
4. REDIRECTED to <<updateStatus>> interface
5. DISPLAY success message

6. END

**delete(\$id)**

1. BEGIN

2. FIND 'id'

3. DELETE data in Appointment Model WHERE 'id' == 'id'

4. REDIRECTED to <<updateStatus>> interface

5. DISPLAY success message

6. END

**cancelEmail(\$id)**

1. BEGIN

2. SELECT

'appointment.id','appointment.patient\_name','appointment.date','appointment.timeslot',

'appointment.message','clinic.clinic\_name','users.email','users.name','appointment.dr\_name' WHERE 'appointment.id' == id

3. REDIRECTED to <<updateStatus>> interface

4. END

**sendEmail(Request \$request, \$id)**

1. BEGIN

2. SELECT

'appointment.id','appointment.patient\_name','appointment.date','appointment.timeslot',

'appointment.message','appointment.status','appointment.attendance','clinic.clinic\_name','users.email','users.name','appointment.dr\_name' WHERE

'appointment.id' == id

3. SEND MAIL

4. UPDATE 'status'=>'Cancel','attendance'=>'Cancel' WHERE 'id' == id

5. DISPLAY success message

6. END

## 1.8 Manage Medical Record

### 1.8.1 Class 18: addRecords Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the clinic staff to add the diagnosed of the appointment session between the doctor and the patient.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	diagnosis	Varchar
	prescribed_medicine	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.8.2 Class 19: viewRecords Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the patient, clinic staff and doctor to view the medical record for each appointment session.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	clinic_name	Varchar
	dr_name	Varchar
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	diagnosis	Varchar
	prescribed_medicine	Varchar
	appointment_id	Big integer
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

## 1.8.3 Class 20: RecordController

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to manage the functions for the manage medical record.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	clinic_name	Varchar
	dr_name	Varchar
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	diagnosis	Varchar
	prescribed_medicine	Varchar
	appointment_id	Big integer
	id	Big integer
	clinic_id	Big integer
	dr_id	Big integer
med_records	Varchar	
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	viewRecord(\$id)	This method is to display the medical record of each successful appointment session.
	addRecord(\$id)	This method is to allow the clinic staff to input the diagnosed from the appointment session.
	store(Request \$request, \$id)	This method is to store the input from addRecord() into the database.
	delete(Request \$request, \$id)	This method is to allow the clinic staff to delete the medical record.
<b>Algorithm</b>	<b>viewRecord(\$id)</b> 1. BEGIN	

	<ol style="list-style-type: none"> <li>2. viewRecord() is called</li> <li>3. Create object “record” to connect to Appointment, Records, Clinic Model</li> <li>4. Go to &lt;&lt;viewRecord&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>addRecord(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. addRecord() is called</li> <li>3. Create object “record” to connect to Appointment Model</li> <li>4. Go to &lt;&lt;addRecord&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>store(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. store() is called</li> <li>3. Connected to Records Model</li> <li>4. Create object “exists” to connect to Appointment Model</li> <li>5. REDIRECTED to &lt;&lt;appointmentstatus&gt;&gt; interface</li> <li>6. END</li> </ol> <p><b>delete(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. delete() is called</li> <li>3. Create object “Records” to connect to Records and Appointment Model</li> <li>4. REDIRECTED to &lt;&lt;appointmentstatus&gt;&gt; interface</li> <li>5. END</li> </ol>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**1.8.4 Class 21: Appointment, Records, Clinic, Doctor Model**

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to store, validate, and retrieve the manage medical record data in the database.	
	<b>Attributes Name</b>	<b>Attributes Type</b>

<b>Attributes</b>	clinic_name	Varchar
	dr_name	Varchar
	patient_name	Varchar
	date	Date
	timeslot	Varchar
	message	Varchar
	diagnosis	Varchar
	prescribed_medicine	Varchar
	appointment_id	Big integer
	id	Big integer
	clinic_id	Big integer
	dr_id	Big integer
	med_records	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	viewRecord(\$id)	This method is to display the medical record of each successful appointment session.
	addRecord(\$id)	This method is to allow the clinic staff to input the diagnosed from the appointment session.
	store(Request \$request, \$id)	This method is to store the input from addRecord() into the database.
	delete(Request \$request, \$id)	This method is to allow the clinic staff to delete the medical record.
<b>Algorithm</b>	<b>viewRecord(\$id)</b> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT  'records.appointment_id','appointment.patient_name','appointment.date','appointment.timeslot','records.diagnosis','records.prescribed_medicine','clinic.clinic_name','appointment.dr_name','appointment.message','appointment.id' WHERE 'records.appointment_id' == id</li> <li>3. Go to &lt;&lt;viewRecord&gt;&gt; interface</li> <li>4. END</li> </ol>	

	<p><b>addRecord(\$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>FIND 'id'</li> <li>Go to &lt;&lt;addRecord&gt;&gt; interface</li> <li>END</li> </ol> <p><b>store(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>INSERT 'appointment_id', 'diagnosis', 'prescribed_medicine'</li> <li>UPDATE 'med_records'=&gt;'Exist' in Appointment Model WHERE 'id' == id</li> <li>REDIRECTED to &lt;&lt;updateStatus&gt;&gt; interface</li> <li>Display success message</li> <li>END</li> </ol> <p><b>delete(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>FIND 'id'</li> <li>DELETE data in Appointment Model where 'id' == id</li> <li>END</li> </ol>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 1.9 Manage Medicine Information

### 1.9.1 Class 22: addMedicine Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the admin to add medicine information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
image	Medium text	
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>



	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.9.2 Class 23: editMedicine Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the admin to edit the medicine information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.9.3 Class 24: Index Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to display all the medicine that are stored inside the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>		

	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

#### 1.9.4 Class 25: search Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to search for the medicine by name.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

#### 1.9.5 Class 26: showMedicine Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the admin to view the medicine information in full details.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>

	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.9.6 Class 27: userView Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This method is to allow all the users to view the medicine information that are stored inside the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.9.7 Class 28: medController

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to manage the functions for the manage medicine information.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text

Methods	Method Name	Description
	index()	This method is to display the medicine information that are stored inside the database.
	medicineView()	This method is to allow the users to view the medicine information in full details.
	searchMedicine()	This method is to allow all the users to search the medicine by name.
	store(Request \$request)	This method is to store the medicine data input by the admin into the database.
	show(\$id)	This method is to allow the admin to view the medicine information in full details.
	edit(\$id)	This method is to allow the admin to edit the medicine information.
	update(Request \$request, \$id)	This method is to save the updated data in edit() in the database.
	delete(\$id)	This method is to allow the admin to delete any medicine information in the database.
<b>Algorithm</b>	<p><b>index()</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>index() is called</li> <li>Create object "Medicine" to connect to Medicine Model</li> <li>Go to &lt;&lt;index&gt;&gt; interface</li> <li>END</li> </ol> <p><b>medicineView()</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>medicineView() is called</li> <li>Create object "Medicine" to connect to Medicine Model</li> <li>Go to &lt;&lt;userView&gt;&gt; interface</li> </ol>	

5. END

**searchMedicine()**

1. BEGIN
2. searchMedicine() is called
3. Create object “search\_text” to get search query from users
4. Create object “searchMedicine” to connect to Medicine Model
5. Go to <<search>> interface
6. END

**store(Request \$request)**

1. BEGIN
2. store() is called
3. Create object “Medicine” to connect to Medicine Model
4. REDIRECTED to <<index>> interface
5. END

**show(\$id)**

1. BEGIN
2. show() is called
3. Create object “Medicine” to connect to Medicine Model
4. Go to <<showMedicine>> interface
5. END

**edit(\$id)**

1. BEGIN
2. edit() is called
3. Create object “Medicine” to connect to Medicine Model
4. Go to <<editMedicine>> interface
5. END

**update(Request \$request, \$id)**

1. BEGIN
2. update() is called
3. Create object “Medicine” to connect to Medicine Model
4. REDIRECTED to <<index>> interface
5. END

**delete(\$id)**

	<ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. delete() is called</li> <li>3. Create object “Medicine” to connect to Medicine Model</li> <li>4. REDIRECTED to &lt;&lt;index&gt;&gt; interface</li> <li>5. END</li> </ol>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 1.9.8 Class 29: Medicine Model

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to store, validate, and retrieve the manage medicine information data in the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	drug_name	Varchar
	generic_name	Varchar
	med_description	Varchar
	used_for	Varchar
	important_info	Varchar
	image	Medium text
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	index()	This method is to display the medicine information that are stored inside the database.
	medicineView()	This method is to allow the users to view the medicine information in full details.
	searchMedicine()	This method is to allow all the users to search the medicine by name.
	store(Request \$request)	This method is to store the medicine data input by the admin into the database.

	show(\$id)	This method is to allow the admin to view the medicine information in full details.
	edit(\$id)	This method is to allow the admin to edit the medicine information.
	update(Request \$request, \$id)	This method is to save the updated data in edit() in the database.
	delete(\$id)	This method is to allow the admin to delete any medicine information in the database.
<b>Algorithm</b>	<p><b>index()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT all FROM Medicine Model</li> <li>3. Go to &lt;&lt;index&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>medicineView()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT all FROM Medicine Model</li> <li>3. Go to &lt;&lt;userView&gt;&gt; interface</li> <li>4. END</li> </ol> <p><b>searchMedicine()</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. GET searchquery</li> <li>3. SEARCH Medicine Model WHERE 'drug_name' == searchquery ORWHERE 'generic_name'== searchquery</li> <li>4. Go to &lt;&lt;search&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>store(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. INSERT 'drug_name', 'generic_name', 'med_description', 'used_for', 'important_info', 'image'</li> <li>3. REDIRECTED to &lt;&lt;index&gt;&gt; interface</li> <li>4. DISPLAY success message</li> </ol>	

	<p>5. END</p> <p><b>show(\$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>SELECT all FROM Medicine Model</li> <li>Go to &lt;&lt;showMedicine&gt;&gt; interface</li> <li>END</li> </ol> <p><b>edit(\$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>SELECT all FROM Medicine Model</li> <li>Go to &lt;&lt;editMedicine&gt;&gt; interface</li> <li>END</li> </ol> <p><b>update(Request \$request, \$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>FIND id</li> <li>UPDATE 'drug_name', 'generic_name', 'med_description', 'used_for', 'important_info', 'image'</li> <li>REDIRECTED to &lt;&lt;index&gt;&gt; interface</li> <li>DISPLAY success message</li> <li>END</li> </ol> <p><b>delete(\$id)</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>FIND 'id'</li> <li>DELETE data in Medicine Model WHERE 'id' == id</li> <li>END</li> </ol>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 1.10 Manage Clinic

### 1.10.1 Class 30: clinicList Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to display all the clinic information that are store inside the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>



	id	Big integer
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar
	work_hours	Varchar
	clinic_image	Medium Text
	services	Varchar
	email	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.10.2 Class 31: drmanagement Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow the clinic staff to view the information of doctor's that works under them. The clinic staff can also remove the doctor account in this interface.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	dr_id	Big integer
	dr_name	Varchar
	clinic_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
	doctor_phone_no	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable

<b>Algorithm</b>	Not Applicable
------------------	----------------

### 1.10.3 Class 32: searchClinic Interface

<b>Class Type</b>	Boundary Class	
<b>Responsibility</b>	This class is to allow all the users to search for the clinic by name.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar
	work_hours	Varchar
	clinic_image	Medium Text
	services	Varchar
email	Varchar	
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	Not Applicable	Not Applicable
<b>Algorithm</b>	Not Applicable	

### 1.10.4 Class 33: clinicController

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to manage the functions for the manage clinic.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar
work_hours	Varchar	

	clinic_image	Medium Text
	services	Varchar
	email	Varchar
	dr_id	Big integer
	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
<b>Methods</b>	<b>Method Name</b>	<b>Description</b>
	displayClinicList()	This method is to display all the list of clinics that are stored inside the database.
	search()	This method is to allow the users to search for the clinic by name
	indexDoctor(Request \$request)	This method is to display the doctor that works under that particular clinic.
	delete(\$id)	This method is to remove the doctor's account that works under the clinic
<b>Algorithm</b>	<p><b>displayClinicList()</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>displayClinicList() is called</li> <li>Create object "clinic_table" to connect to User, Clinic Model</li> <li>Go to &lt;&lt;clinicList&gt;&gt; interface</li> <li>END</li> </ol> <p><b>search()</b></p> <ol style="list-style-type: none"> <li>BEGIN</li> <li>search() is called</li> <li>Create object "search_text" to get search query from users</li> <li>Create object "searchClinic" to connect to User, Clinic Model</li> <li>Go to &lt;&lt;searchClinic&gt;&gt; interface</li> <li>END</li> </ol>	

	<p><b>indexDoctor(Request \$request)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. indexDoctor() is called</li> <li>3. Create object “display_dr” to connect to Clinic, User, Doctor Model</li> <li>4. Go to &lt;&lt;drmanagement&gt;&gt; interface</li> <li>5. END</li> </ol> <p><b>delete(\$id)</b></p> <ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. delete() is called</li> <li>3. Create object “Doctor” to connect to Doctor Model</li> <li>4. REDIRECTED to &lt;&lt;drmanagement&gt;&gt; interface</li> <li>5. END</li> </ol>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 1.10.5 Class 34: User, Clinic, Doctor Model

<b>Class Type</b>	Entity Class	
<b>Responsibility</b>	This class is used to store, validate, and retrieve the manage clinic data in the database.	
<b>Attributes</b>	<b>Attributes Name</b>	<b>Attributes Type</b>
	id	Big integer
	clinic_id	Big integer
	clinic_name	Varchar
	clinic_address	Varchar
	clinic_phone_no	Varchar
	clinic_bg	Varchar
	work_hours	Varchar
	clinic_image	Medium Text
	services	Varchar
	email	Varchar
	dr_id	Big integer

	dr_name	Varchar
	doctor_image	Medium Text
	doctor_specialization	Varchar
	gender	Varchar
	doctor_edu_bg	Varchar
	doctor_years_working	Varchar
<b>Meth ods</b>	<b>Method Name</b>	<b>Description</b>
	displayClinicList()	This method is to display all the list of clinics that are stored inside the database.
	search()	This method is to allow the users to search for the clinic by name
	indexDoctor(Request \$request)	This method is to display the doctor that works under that particular clinic.
	delete(\$id)	This method is to remove the doctor's account that works under the clinic
<b>Algor ithm</b>	<b>displayClinicList()</b>	
	<ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. SELECT 'clinic.clinic_image','users.email','clinic.id','clinic.clinic_id','clinic.clinic_name','clinic.clinic_phone_no','clinic.work_hours','clinic.clinic_bg','clinic.clinic_address','clinic.services'</li> <li>3. Go to &lt;&lt;clinicList&gt;&gt; interface</li> <li>4. END</li> </ol>	
	<b>search()</b>	
	<ol style="list-style-type: none"> <li>1. BEGIN</li> <li>2. GET 'searchclinic'</li> <li>3. SELECT 'clinic.clinic_image','users.email','clinic.clinic_id','clinic.clinic_name','clinic.clinic_phone_no','clinic.work_hours','clinic.clinic_bg','clinic.clinic_address','clinic.services' WHERE 'clinic.clinic_name' == 'searchclinic'</li> <li>4. Go to &lt;&lt;searchClinic&gt;&gt; interface</li> <li>5. END</li> </ol>	

**indexDoctor(Request \$request)**

1. BEGIN
2. GET clinic\_id = id
3. SELECT  
'doctor.id','doctor.dr\_id','doctor.dr\_name','doctor.doctor\_image','doctor.doc  
tor\_specialization','doctor.doctor\_years\_working','doctor.doctor\_edu\_bg',  
'doctor.doctor\_phone\_no','doctor.clinic\_id','clinic.clinic\_name','doctor.gender'  
WHERE 'clinic.clinic\_id' == clinic\_id  
Go to <<drmanagement>> interface
4. END

**delete(\$id)**

1. BEGIN
2. FIND 'id'
3. DELETE data in Doctor Model WHERE 'id' == 'id'
4. REDIRECTED to <<drmanagement>> interface
5. DISPLAY success message
6. END

**2.2 DATA DICTIONARY**

There are seven tables for this application which are users, doctor, clinic, appointment, records, feedback, and medicine. Each table's attribute is described in detail in the description section, including the data type, field length, and constraints. Excluding the 'feedback' table, the other tables are involve with the five major modules (manage account, manage appointment booking, manage medical record, manage medicine information, manage clinic).

**1.10.1 users****Table 0.1 Database Dictionary for 'users'**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big integer	Primary Key	Login id for users

name	Varchar	Not null	Login name for users
email	Varchar	Unique key	Email of the users
password	Varchar	Not null	Password of the users
role	Varchar	Null	Role of each user
age	Int	Null	Age of patient
gender	Varchar	Null	Gender of patient
phone_no	Varchar	Null	Phone number of the patient
health_details	Varchar	Null	Patient's health details
image	Medium Text	Null	Patient's image

### 1.1.1.1 clinic

**Table 0.2 Database Dictionary for 'clinic'**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for clinic
clinic_id	Big Integer	Foreign Key	Login id for the clinic
clinic_phone_no	Varchar	Not null	Phone number of the clinic
clinic_address	Varchar	Null	Address of the clinic
clinic_bg	Varchar	Null	Clinic background
work_hours	Varchar	Null	Clinic working or operation hours
clinic_name	Varchar	Null	Name of the clinic

clinic_image	Medium Text	Null	Image of the clinic
services	Varchar	NULL	Services that the clinic can offered

### 1.1.1.2 doctor

**Table 0.3 Database Dictionary for ‘doctor’**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for doctor
dr_id	Big Integer	Foreign Key	Login id for doctor
dr_name	Varchar	Not null	Name of the doctor
doctor_image	Medium Text	Not null	Image of the doctor
doctor_specialization	Varchar	Not null	The department that the doctor’s expertise in
gender	Varchar	Not null	Gender of the doctor
doctor_edu_bg	Varchar	Not null	Doctor education background
doctor_years_working	Varchar	Not null	Years of working of the doctor
doctor_phone_no	Varchar	Not null	Phone number of the doctor



clinic_id	Big Integer	Not null	Workplace of the doctor
-----------	-------------	----------	-------------------------

### 1.1.1.3 feedback

**Table 0.4 Database Dictionary for 'patient'**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for feedback
user_id	Big Integere	Not null	Id of the person giving feedback
message	Varchar	Not null	Feedback message

### 1.1.1.4 appointment

**Table 0.5 Database Dictionary for 'appointment'**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Intger	Primary Key	Id for the appointment
user_id	Big Integere	Foreign Key	Id of the patient booking the appointment
patient_name	Varchar	Not null	Name of the patient
date	Date	Not null	Appointment date
timeslot	Varchar	Not null	Appointment timeslot
message	Varchar	Null	Appointment message from patient

clinic_id	Big Integer	Not null	Id of the booked clinic
dr_id	Big Integer	Null	Id of the booked doctor
dr_name	Varchar	Null	Name of the booked doctor
status	Varchar	Null	Status for appointment (Accepted, Rejected)
attendance	Varchar	Null	Status for appointment attendance (Attend, Absent)
med_records	Varchar	Null	Medical record status (Exist, Null)

#### 1.1.1.5 records

**Table 0.6 Database Dictionary for ‘records’**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
id	Big Integer	Primary Key	Id for medical record
appointment_id	Big Integer	Foreign Key	Id of the appointment
diagnosis	Varchar	Not null	Patient’s diagnosis
prescribed_medicine	Varchar	Not null	Patient’s prescribed medicine

#### 1.1.1.6 medicine

**Table 0.7 Database Dictionary for ‘medicine’**

FIELD NAME	DATA TYPE	CONSTRAINT	DESCRIPTION
------------	-----------	------------	-------------

id	Big Integer	Primary Key	Id for medicine
drug_name	Varchar	Not null	Drug name of the medicine
generic_name	Varchar	Not null	Generic name of the medicine
med_description	Long Text	Not null	Description of the medicine
used_for	Long Text	Not null	Usage of the medicine
important_info	Long Text	Not null	Important info of the medicine
image	Medium Text	Null	Image of the medicine