PERSONAL HEALTH RECORDS MOBILE APP

LEONG XIAN JUN

UNIVERSITI TUNKU ABDUL RAHMAN

PERSONAL HEALTH RECORDS MOBILE APP

LEONG XIAN JUN

A project report submitted in partial fulfilment of the Requirements for the award of Bachelor of Science (HONOURS) Software Engineering

Lee Kong China Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

SEPTEMBER 2020

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature:

Name: LEONG XIAN JUN

Student ID: 1700517

Date: 10 September 2020

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**Personal Health Records Mobile App**" was prepared by **LEONG XIAN JUN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature:

Supervisor: Ms Beh Hooi Ching

Date: 28th September 2020

ACKNOWLEDGEMENTS

I would like to thank all of the contributors to this project for its successful completion. Great gratitude shall be awarded to my supervisor, Ms Beh Hooi Ching and my moderator, Ms Loo Yim Ling for guiding me throughout the entire project. Their advice and experiences are precious to the entire development process of the project.

Special thanks to UTAR for providing me with an excellent place to gain knowledge. It is my honour to be one of the students in UTAR.

ABSTRACT

Due to the increased population, the data size of the health records can become tremendous. Eventually, it can hardly be handled by humans. Therefore, technology shall take over human manpower in handling and tracking health records. With this, this project developed a system that aids the patients in keeping track their health records while helping the medical staff in handling the health records. Beside health record tracking, the product of this project aimed to provide extra services such as appointment scheduling and medication reminder. Phased development methodology was chosen and implemented in this project. Thus, this project consists of four phases, which are the planning phase, analysis and design phase, development and testing phase as well as the closing phase. Background of the problem was studied in the planning phase. During the analysis and design phase, the project requirements were finalized and system architecture and designs were modelled. In the development and testing phase, it was further separated into three sub-phases, which each sub-phase focused on modules with different priority levels. After completion of the development, the system was tested for its usability and 83 scores were given for the satisfaction of the applications. With the completion of this project, the patients could use the mobile application to keep track of their health records while the medical staff can manage the health records via the web application. Despite there are more for improvement, this project is considered a great success as it fulfils all of the defined objectives.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AES	Advanced Encryption Standard
API	Application Programming Interface
BMI	Body Mass Index
DOM	Document Object Model
HP	Health Prescription
HTTP	Hypertext Transfer Protocol
LTR	Lab Test Result
MR	Medication Record
UAT	User Acceptance Test
UI	User Interface
UX/UI	User Experience and User Interface
OTP	One-Time Password
RSA	Rivest-Shamir-Adleman cryptosystem
SDLC	Software Development Life Cycle
SOP	Standard Operation Procedures
WBS	Work Breakdown Structure

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1. Introduction

Tracking of health records may be a hassle to the patients, especially each record has different fields of data. Besides, the connection between patients and medical staff has been an issue in promoting a better healthcare system. The medical staff will find it difficult to get updates on the patient's health unless the patients report or go back to the medical institution.

To tackle this issue, this project is initialized to demand a solution in enhancing the connection between the patients and the medical staff. Therefore, this chapter discusses the problem statement, objectives, project approach, project solution and project scopes.

1.2. Background of the Problem

Medical records are an essential element in healthcare. A good medical record contains information such as the patient's history, health condition, test result, prescription, medication, recovery progress and so on (Bali et al., 2011). Additionally, for any changes in the medical records, amendments must be made with the date instead of just removing the wrong information (The importance of keeping good medical records, 2017). Moreover, it is also a proof for the medical institution or the medical staff after they complete diagnosis and give the patient medication. With the availability of medical records, it can defend them against some claims or complaints from the patients.

Since medical records will be produced in every meeting with the medical staff, eventually, the medication institution will be accumulated with a huge set of data (Shah, 2019). Without a proper medical records management system, it will be difficult for the medical staff to retrieve the relevant records of a particular patient. Similarly, the patient may visit countless medical institutions to either conduct a body check-up or get a diagnosis on the illnesses. This will have a similar issue as all the data or health records are scattered in different medical institutions.

According to Makary and Daniel (2016), medical error contributes to the third most death cases in the US. For instance, poor information flow will cause the medical staff from another department to give the wrong prescription as some vital information is not followed the patients when they transferred to another facility. Furthermore, this is also considered as a communication problem, which fails to communicate or deliver the correct information between facilities (The 8 Most Common Root Causes of Medical Errors, 2018).

Moreover, human errors may occur especially when there are no SOP or rules in doing a thing. Even SOP is provided, human errors may still occur when the medical staff are not following it. Additionally, poor documentation skill is also one of the human errors as it can greatly affect the medical staff in giving the current diagnosis as the terms used in documentation is highly professional and unique (The 8 Most Common Root Causes of Medical Errors, 2018).

Besides, the patient could lead to medical errors too when they did not cooperate during the diagnosis of their illnesses. For instance, they are not providing their actual identification, which causes the medical staff to fail to identify the background and illness history of the patient and eventually provided an incorrect diagnosis to the patients. On top of that, there are more medical errors such as insufficient and inconsistent knowledge of the new medical staff when they are newly joined in a medical institution, an insufficient workforce, technical failures and poor policies in providing medical services. (The 8 Most Common Root Causes of Medical Errors, 2018).

This is further supported by the article written by Hammer (2016), as the statistics stated there are more than seven thousands cases of death due to the medication errors. However, according to Gorski (2019), the number of death cases may not as big as the others mentioned in their study. Yet, the medical error can be reduced greatly if a proper system is available to analyse and interpret the health condition of the patients.

On top of that, according to Adair (2019), it has listed some of the future trends that an electronic medical records system should adopt. The patients are expected to access their health records from different medical institutions within an application.

Furthermore, according to the survey done by Patel and Johnson (2019), 50% among all the tablet or smartphone users had used a health or wellness application. Among them, 75% of the individuals will use it to track their progress on health-related goals.

In terms of the business world, digitalization is the current trend in operating a business. Similarly, in the medical industry, it is necessary to digitalize the current workflow in the medical institution as it may lead to a better quality of service and high satisfaction level from the patients (Adair, 2019).

1.3. Problem Statements

With several pieces of research done on this topic, the following are listed as the main issues of this project. With the completion of the development, the following will be resolved correctly with an appropriate solution.

1.3.1. Health Records Tracking Issue

Different patients will have a different preference for visiting a hospital or clinic for illness treatment. Some may constantly visit the same hospital while others might visit different hospital due to external factors, such as outstation, travelling and emergency. Therefore, it is common for patients to have their health records scattered around multiple medical institutions.

For the patients to trace their records, they are required to go to particular medical institutions to request for records retrieval. These processes are time-consuming for patients especially when they had visited multiple medical institutions. If they have all of the records, it is still a problem for them to locate a suitable location to store these records.

1.3.2. Limited Health Data Interoperability

Before the existence of any digital method in handling the medical records, all of the diagnosis content was recorded in the form of paper. For each diagnosis, it will usually have one copy respectively. Thus, it is difficult for different medical institutions to access records from other medical institutions (Hersh, 1995).

This issue has been improved with the implementation of a medical portal or e-medical platform in recent years. However, it is ineffective as some records are still scattered around the system of each medical institution. Hence, the patients are reluctant to use these systems due to the hassle of accessing different systems (Heath, 2016).

Additionally, separated systems indicate that the records are not centralized, which may lead to ignorance of some medical data. Such ignorance is critical as the medical staff may provide a different medication as well as treatment to the patient (Jones, 2017).

1.4. Objectives

- To develop a mobile health records application that helps patients to track all of their formal health records while monitoring the users who access to these records
- To develop an interoperable health records system in both web and mobile platforms that allows data sharing by August 2020

1.5. Proposed approach/ solution

1.5.1. Solution

Based on the problems identified, it shows that there is a necessity in developing the application in both web and mobile platform. The web application will mainly serve the medical staff while for the mobile application; it will be used by the patient. Additionally, some of the features from similar systems present in this new system.

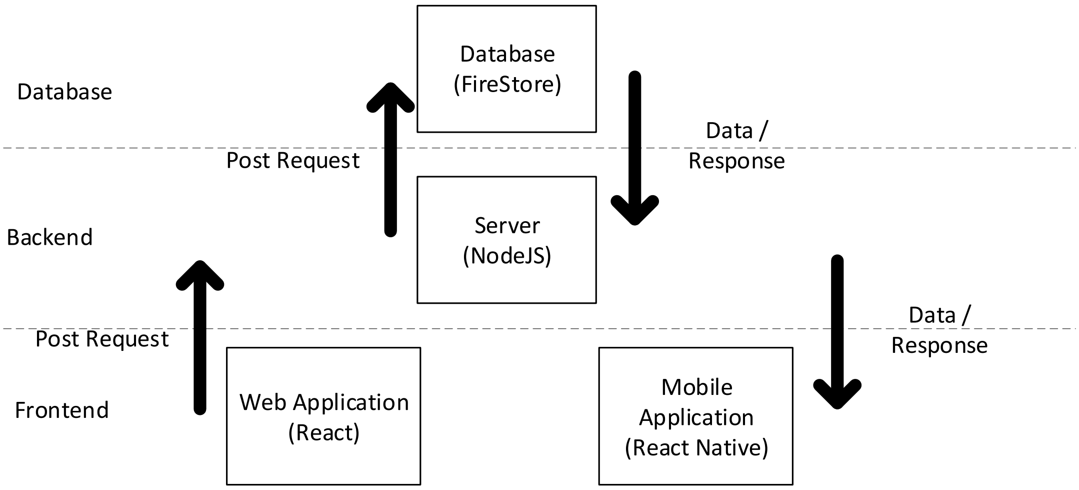


Figure 1.1 System Architecture

As shown in Figure 1.1, the three-tier architecture design is used in this implementation. Such planning is to prevent direct access by the users to the database. On top of that, a strict checking process can be implemented in the server, so it can greatly reduce the possibility of having malicious attacks on the database.

In general, this system will have most of the common features that can be found in other existing systems. On top of that, this system provides a platform for the medical staff to enter the health record of the patient via the web application. Besides, the system will record the access request when the doctor retrieves the health records of the patient as well as the analysis. This is to make sure if data leakage occurs, it is traceable and can be used to identify the potential users for the data leakage.

Similarly, the mobile application displays the graphical visualization of the health data to the patient. In terms of interoperability, the patients can achieve it by sharing the health records entered by them. For instance, they constantly enter their blood sugar level. This can be a piece of great evidence for the doctor to come out with a better interpretation of the patient’s health. Moreover, the application is designed to record the health records access done by the medical staff.

1.5.2. Approach

1.5.2.1. Research Approach

During the execution of this project, the research approach used is quantitative research. According to Creswell and Creswell (2017), quantitative research is a method used to prove the theories by having assumptions first and investigating the correlation of variables. From the collected data, variables can be analysed and eventually come out with a generalized conclusion that can be reproduced by conducting similar research. Therefore, in this project, surveys were used to conduct quantitative research.

Via the questionnaires, the system requirements were double confirmed with the analysis of the responses received. The survey aimed to receive the responses on the expected behaviours or functionalities of a health records system. The questions were created after completing the research on health record-related issues. As the outcome of the survey, the project received a list of user-agreed requirements on the desired features of the health record system.

1.5.2.2. Development Approach

The development approach is also known as the development methodology. Each methodology has its cycle of development, which is known as SDLC. It has generalized all the existing methodology into four different types, which are waterfall, iterative, incremental and agile (Project and Development Approaches, 2014).

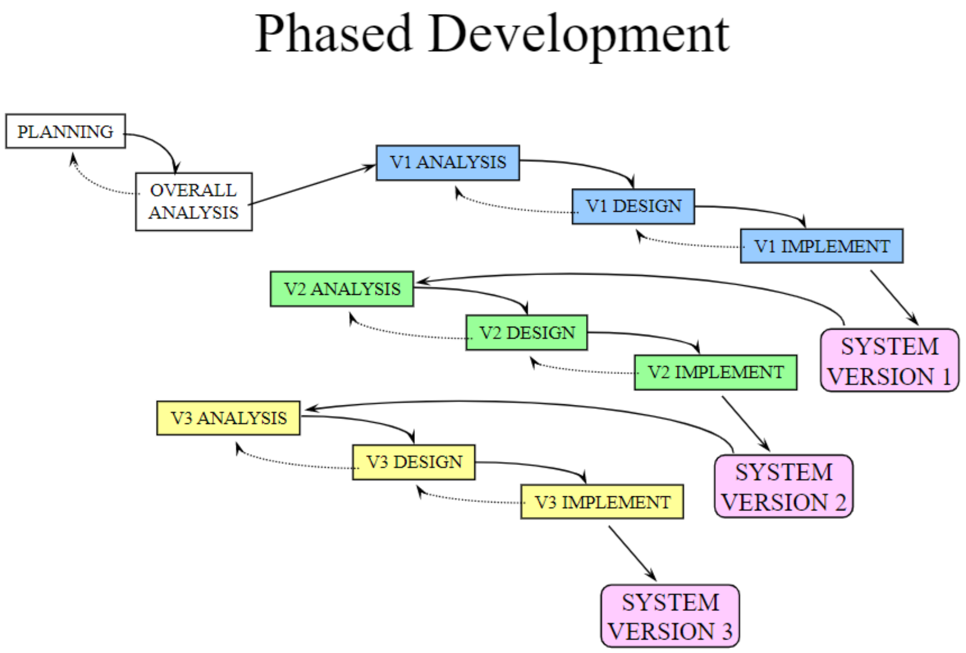


Figure 1.2 Stage Visualization of Phased Development Methodology (Scott, 2016)

For this project, the development approach used in this project is the Phased development methodology, which is an iterative and incremental methodology. According to Opralova (2005) and Schindler (2010), phased development methodology will decompose the whole project into several smaller portions and each portion will be solved one by one based on its priority level.

Moreover, since the most important module is implemented at the earliest phase, the client can be involved in the testing earlier. Thus, this allows the earlier identification of errors within these core modules. Therefore, the possibility of having bugs in the core modules of the system will be greatly reduced when the implementation of the system is completed and the system is ready for delivery (Iterative Development - Phased Iterative Development Model, 2016).

Additionally, with the involvement of clients in the earlier phase, the project will be able to deliver a system that has high user acceptance, as the clients may provide their responses on each phase. Therefore, changes can be made in the consecutive phases in resolving their responses (Opralova, 2005).

1.6. Scope

In this section, it will include the functional scopes, non-functional scope and the target users as well as their respective platform. Additionally, within the scope, it will contain the restriction if applicable.

1.6.1. Target User

The target users of this system are the patients and the staff from medical institutions. The patients will keep track of their health records in the mobile application. For the medical staff, they will update the health records created in the web application when the diagnosis is conducted on the patient.

1.6.2. Features Covered

In this section, it will be separated into four modules that must be implemented in the system.

1.6.2.1. Health Records Tracking

This web application will provide a dashboard for the medical staff to create and update the health record. The system must store the following type of health records.

1. Medical Prescription
2. Medication Record
3. Lab Test Result

The mobile application must allow the patients to display their history of health records that are created by medical staff. Besides, the mobile application must display the analysis of the health records to the patients. Also, the patients can update their health conditions, such as blood sugar level, blood pressure level and BMI, in the mobile application.

1.6.2.2. Appointment Scheduling

This mobile application must allow users to arrange an appointment with the doctor by showing the available time slot that is set by the doctor. The web application must notify the doctor on every appointment made by sending a notification email. If an appointment is close, the doctor must be notified in the web application and the patient must be notified in the mobile application.

If the patient or the doctor fails to attend the appointment, both applications, which in web and mobile platform, must allow them to either reschedule the appointment or cancel it.

1.6.2.3. Health Records Access by Patients

The medical staff can search the patient to retrieve the respective health records and the analysis of the health records in the web application. Upon access, the system must request authentication from both parties, which are the records owner and the medical staff. After both parties authorize this action, the system must retrieve all the health records related to the patient to display to the medical staff. Besides, the system must record the accessed patient health record, so it can track the person who retrieves the record.

For emergencies, the patients need to grant permission to other users of the system to authorize the data access request. During an emergency, which the patient cannot authorize the request, the medical staff need to select emergency request option and search for the patient name. Then, the system must request authorization from the users granted with permission.

If no users have granted the permissions, the medical staff can directly get the information. However, this situation will be minimized by having the system requesting the users to grant permission to other users every time they launch the application.

1.6.2.4. Medication Reminder

The mobile application must remind the patient based on the medication records created by the medical staff. Additionally, when the medication is low in amount, the mobile application must notify the patient to refill the medication.

At the same time, the doctor must be notified in the web application when the patient finishes the medication. If extra medication is needed, the doctor must be able to amend on the medication record in the web application.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Health is very crucial to one, as it can affect one's daily routine. Without a healthy body, one can perform their tasks effectively and efficiently. Even worse, one might halt the company operation. However, health is not something to take care of when and only when one is sick. Instead, it is a long-term process for one to keep themselves healthy. Via an electronic health records system, one can easily keep track of their health condition and seek doctors' help if necessary.

On top of that, the health record system had present for many years, but people still have many concerns in using it. Besides, due to the advancement of information technology, more products were manufactured and offered with different platforms. Other than that, further study was conducted on areas that may impact the success of this project. In short, this literature review aims to:

1. Identify the potential concerns in the health records system implementation
2. Understand the cross-platform application development
3. Determine the most suitable methodology
4. Compare the existing systems for the common features
5. Investigate the tools for the development

2.2. Concerns in the Implementation of a Health Record System

Health records system is a system that enables the patients to keep track of their health records. However, the majority of the patients reluctant to use electronic health records system, as they feel unsafe in using it. To them, it is easier for this online system to leak their data compared to the traditional paper-form recording system.

On top of that, the target users are not limited to patients, but also the medical staff. By involving the medical staff, each health record will be verified by the medical staff like the doctor, and the patients will receive the correct data compared to those entered by themselves into the system. Therefore, this literature review is conducted to identify the potential factors that patients and the medical staff reluctant to use the electronic health records system.

After reviewing three relevant articles, few concerns were pointed out by medical staff as well as experts in the medical fields. Majority of them are worried about the security of the health records system in data sharing. Data is very crucial for the patients, as malicious users can use these data for trading. Also, incomplete data can be fatal to the patient, as it may lead to misdiagnosis. Additionally, it can lead to ineffective treatment and conduction of duplication health condition test (Warren et al., 2019). Hence, this literature review had concluded a few issues that are reflected by the medical staff and the patients.

2.2.1. Data Completeness

Data completeness is crucial for the medical staff in analysing and interpreting the health condition of a patient. Without the aid of previous health records of the patient, the outcome of the diagnosis can only be depended on the doctor knowledge and experience, which is very risky to the patient. Therefore, two different areas are identified that can contribute to data incompletes, which are missing fields in records and scattered data among the medical institutions (McCrorie et al., 2019; Warren et al., 2019).

2.2.1.1. Incomplete Data Fields

According to the respondents in the research of McCrorie and et al (2019), they mentioned that they are not a good typist. Commonly, they will only use one or two fingers in their typing. Therefore, with the implementation of a health records system in their institution, they afraid that it is difficult for them to records everything mentioned by the patients when they are communicating with the patients via a call.

Also, some of the respondents mentioned that clinical opinion is very crucial information, as it recorded the analysis and interpretation done by the doctor. This piece of information can be very useful for the following visits of the patients. Without clinical opinion, the patients who check their health records might feel insecure to the medical institution as it hides important information, which is the clinical opinion (Warren et al., 2019).

2.2.1.2. Scattered Data in different Medical Institutions

In the current medical field, each medical institution may use different health records system in managing their countless amount of records. However, this causes the data to be scattered in different storages, which leads to the incompleteness of data.

On top of that, Warren and et al (2019) noticed there is a need to improve the data sharing of the existing health records system. Based on Warren and et (2019) research outcome, it shown that there are one million patients visited more than two hospitals in this 1-year study period.

2.2.2. Reduced Work Performance of the Medical Staff

For any medical institutions, they will have an SOP that aids the medical staff in taking their necessary actions. SOP is needed to ensure standardization among the medical staff and make sure that they undergo the correct action in treating the patients.

With the implementation of this health records system, this indicates the changes in the SOP. The changes are not in favour of the medical staff, as they had to get used to the existing SOP. Due to the unfamiliarity to the new SOP as well as their unwillingness to adapt the changes, they reflected that the implementation will reduce their work performance (McCrorie et al., 2019).

Also, they questioned that the system may work in a way, which does not suit their current workflow. Then, this will make them be astounded, as they had no idea to continue their tasks using the system. In the end, they might end up using back the previous method, which is recording the health records in a piece of paper (McCrorie et al., 2019).

Other than that, Feng and et al (2020) suggested a new protocol to enhance the existing authentication protocol for Health Records System. According to their research, this new protocol can outplay the old protocol, as it can process the authentication checking faster with lower communication cost.

Overall, the outcome of these researches is limited to a particular region, such as England (Warren et al., 2019). Therefore, further research conducted shall receive data from different regions, so the outcome is more general and trustable. Moreover, data sharing is an unavoidable element in a health record system. Therefore, to tackle the issue of data incompleteness and the reduced working performance, the health records system should consist of features that ease the medical staff to use.

For instance, for each type of health records, a standardized form will be displayed in the system and the staff are required to enter all of the fields before proceeding to the subsequent sessions. For reduced working performance, the workflow in the system shall tally with the actual workflow of the medical staff. Although the workflow cannot fit all of the workflows from different medical institutions, it should be general enough so that the workflows from different medical institutions can be covered.

By having data sharing within the health record system, it can provide patients with a location to centralize all their health records. Meanwhile, the medical staff can access the system and retrieve all the health records of a particular patient before they conduct a diagnosis on the patient. With these data, it can reduce the possibility to have misdiagnosis or medical errors.

For the research of a new authentication protocol, although it can provide a better authentication due to the involvement of both parties, it requires further enhancement on the computation process. This is because it is very heavy especially when the computation is done by mobile phones (Feng et al., 2020). Additionally, it does not consider the occurrence of an emergency. If this is implemented directly, the medical staff will not able to retrieve the patient health information for analysis, as this protocol requires the authentication from both parties, which included the patient.

Therefore, to have better data protection against leakage, the health records system will request authentication from both parties. At the same time, to tackle an emergency, the system will allow the patient to grant permission to other users of the system. Then, they can authorize the data access request when the patient is under an emergency. Regardless of the type of request, the system will record all the health records access.

In short, to have a health records system that has data completeness as well as data sharing, the mentioned issues shall be handled in the system. By tackling these issues, the new system can ease the patients and the medical staff when they are using the system. Additionally, without these issues, the medical staff will no longer reject to use the system. Instead, it helps them by improving the performance of the whole processing as well as providing a better quality of medical services.

2.3. Cross-Platform Application Development

This project requires the development of application on a different platform. To ensure the success of this project, it is important to identify the success factors and select the most suitable framework for cross-platform application development.

With the early identification of the success factors, it can act as reminders in the development. Besides, the selection of a suitable framework can reduce a lot of redundant works. Usually, the framework is ready with a lot of default set up. With this, the developers can continue with their enhancement on their project without wasting time on the setup of the project. For a solo project, the development skill of that person is very crucial, as the success of the project is solely depending on his skill.

According to Okonkwo and Huisman (2019), several root causes can lead to success in application development. Besides, the clients' opinion and expectation should be taken into considerations when developing an application. Therefore, Okonkwo and Huisman (2019) had surveyed to identify the success factors that can affect the development of a mobile application. According to the result, the top three factors that can contribute to the success of the mobile application development are individual development skills, functionality and mobile devices specifications.

As mentioned, a suitable framework can be key to successful application development. To ensure the correct framework, Biørn-Hansen and et al (2019) also pointed out several issues that should be taken into consideration when selecting a particular framework for the development. More than 50% of the respondents from the survey conducted by Biørn-Hansen and et al (2019) agreed three issues that they frequently faced during their development, which are the performance of the framework, user experience and maturity of the framework.

Both pieces of research involved people with different position and background. This diversity can reduce the possibility of having a bias in the result. In the research of Okonkwo and Huisman (2019), it can provide a relevant result as all the participants are from IT background. For Biørn-Hansen and et al (2019), a questionnaire was posted on forums or groups from several platforms.

However, both results cannot represent all of the cases. Okonkwo and Huismanm (2019) need to get responses from developers that work as a team rather than just individual developers. However, the outcome can be applied in this project as this is a solo project rather than a team project. For Biørn-Hansen and et al (2019), they need to increase the sample size of their questionnaires as well as coming out with questions that are in-depth enough.

On top of that, both pieces of research provide insight on affecting factors of the development and the selection criteria in selecting the most suitable framework for cross-platform application development respectively. By early identification of these factors and criteria, the chance of success for this project will be greater, as less error will be faced and a minimal amount of time and effort will be wasted.

In short, the implementation of personal health records system shall have proper planning, which should take serious consideration in both the success factors and the selection of the framework used in development. A suitable framework can ease as well as shorten the time needed for the development of an application. Furthermore, to ensure the success of this project, the developer of this project shall undergo intensive training or self-learning on the languages used, so the development skills are capable enough to deliver the personal health records system.

2.3.1. Native vs Web vs Hybrid Mobile Application Development

Mobile application development can be categorized into 3 different genres, which are native application, web application and hybrid application. Each has its advantages and disadvantages.

With the native application, it is more vigorous as it can directly access the native features provided by mobile devices. Usually, it does not require a connection to be used by users. For the web application, it is not developed based on a specific device. Therefore, it can be easily accessed with any mobile devices since it only required a browser to use it. Unlike a native application, users do not require downloading the web application in their mobile devices. However, this requires users to have a connection before accessing it.

Lastly, a hybrid application is the combination of both the previous categories. It is written in the web application approach but is compiled latter into a native application and distributed into application stores such as Google Store and App Store. In the context of this project, it will develop a hybrid mobile application as the application is written in JavaScript and latter be compiled into a native application (Q. Huynh, Ghimire and Truong, 2017).

In the study done by Oliveira and et al. (2016), web applications have less energy consumption and better computing power compared to native applications. However, it also mentioned this result may not apply to all of the application, as it still depends on the situation of the development. For instance, according to their observation, web application, which written in JavaScript, needs less energy and provides better performance in applications that require intensive computational power and consist of many simple mathematical operations.

2.4. Software Development Methodology

A software development methodology is a tool that separated the whole software lifecycle into several phases (Kumar and Bhatia, 2014). By adopting a methodology, the software developers get to deliver software that is of good quality within the time constraint. According to Kumar and Bhatia (2014), they concluded that different methodologies have a different lifecycle.

Although each of them consists of the basic activities, such as planning, analysis, design and implementation, they have different characteristics (Shaydulin and Sybrandt, 2017). Hence, not all of the methodologies are suitable for a project. Instead, a project will require a specific methodology that can fit the characteristics of the project. Therefore, this literature review is conducted to identify the strengths and weaknesses of each methodology as well as selecting the most suitable methodology for this solo project.

In this literature review, ten models are identified and compared (Shaydulin and Sybrandt, 2017; Kumar and Bhatia, 2014). Overall, the methodologies involved in the comparison are Waterfall model, V-shaped model, Iterative model, Agile Unified Process, Spiral model, Rapid Application Development, Scrum, Feature-Driven Development, Test-Driven Development and Joint Application Design. Additionally, each methodology has its strengths and weaknesses, which determine the suitability of the methodology to a particular type of project.

Waterfall model is the oldest methodology, which consists of a series of activities that will be executed sequentially. Shaydulin and Sybrandt (2017) stated that the Waterfall model is very good at delivering a product that can meet all of the client requirements. However, it would be too time-consuming in collecting the client requirements. Additionally, if some requirements are not defined correctly, phase fallback will be needed and this will consume a lot of resources, which make this methodology inflexible to any changes in the requirements.

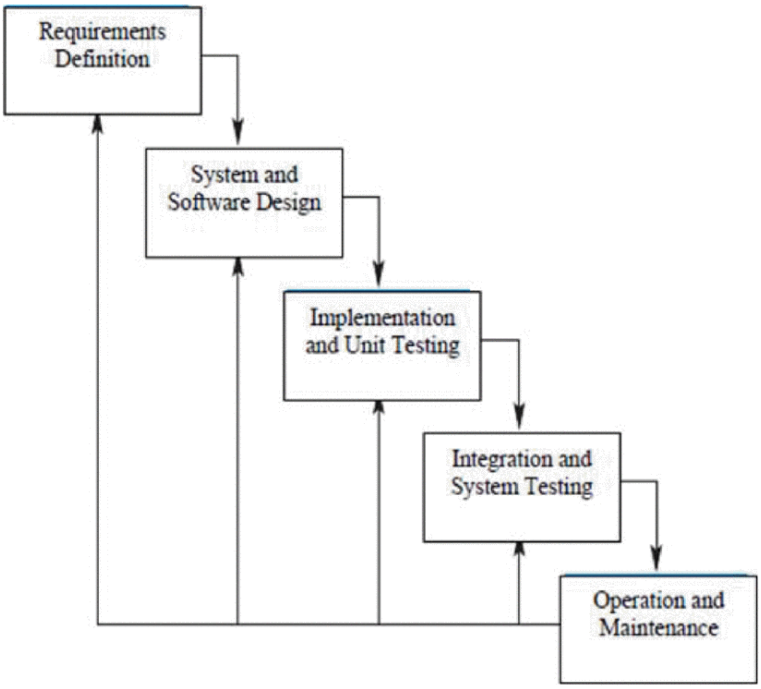

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Figure 2.1 Waterfall Model (Kumar and Bhatia, 2014)

On top of that, V-Shaped Model is another model, which is identical but better than the Waterfall model, as it involves testing at the early stage (Kumar and Bhatia, 2014). With this, more potential defects and bugs can be eliminated early. However, the V-shaped model has a similar issue as the Waterfall model, which is limited flexibility to the changes in the requirements. In short, both the Waterfall and the V-shaped model are only suitable for the small project, which the requirements can be defined clearly and easily. If the project involves many changes in the requirements, it is not suitable to adopt the Waterfall or the V-shaped model.

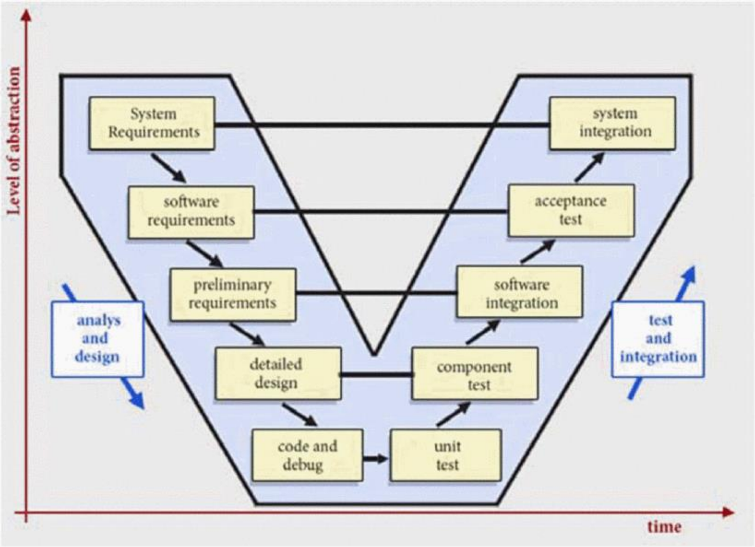


Figure 2.2 V-Shaped Model (Kumar and Bhatia, 2014)

Furthermore, the following methodology is the iterative model. The iterative model allows the implementation of the project to be done in parallel (SDLC - Iterative Model - Tutorialspoint, 2020). With this, the system can be developed part by part. At the same time, errors and defects discovered in the previous iteration can be tackle in the following iterations. Additionally, this methodology can reduce the risk by implementing the high-risk part of the system at an early stage.

However, since not all the requirements are defined clearly at the beginning, changes in the requirements would be frequent, although this methodology is flexible to changes in the requirements.

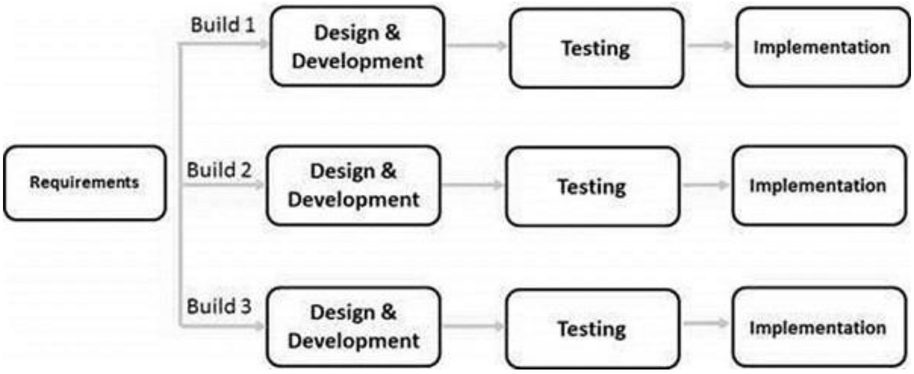


Figure 2.3 Iterative Model (SDLC - Iterative Model - Tutorialspoint, 2020)

Next, the Unified Process will separate the whole development lifecycle into process and workflow. The Unified Process will be iterated in delivering a small part of the systems, which will be integrated into a complete system (Shaydulin and Sybrandt, 2017; Kumar and Bhatia, 2014, pp.5–6). It simplifies the whole implementation of the complete system by producing partial sub-system part by part. Additionally, due to the iterative characteristics, requirement changes can be handled at the consecutive iteration. However, it is usually used together with other methodologies.

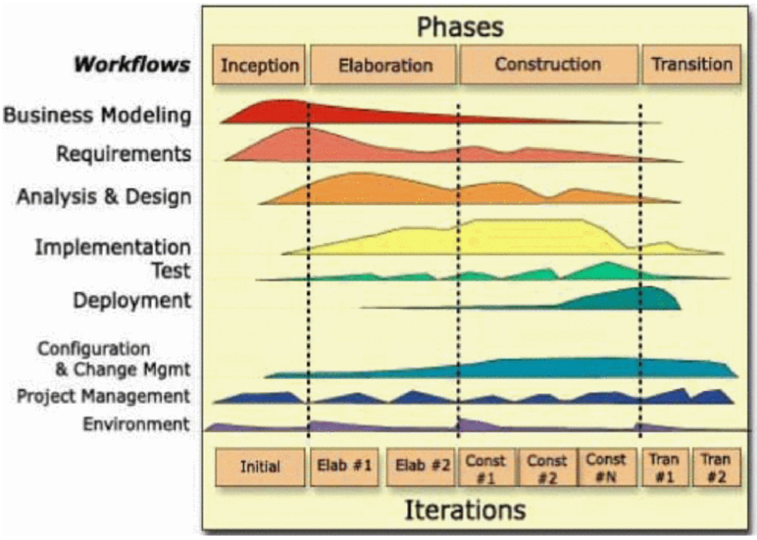


Figure 2.4 Unified Process (Kumar and Bhatia, 2014)

Moreover, another iterative methodology is the Spiral model. This methodology is great in risk management, as the issues discovered in each phase can be tackled in the following phases (Kumar and Bhatia, 2014). However, it requires someone who is expertise risk management to do the risk analysis, which makes this difficult to be used in a small project. Additionally, staff with a higher position will review the outcome after the end of each phase. With this, it can ensure the deliverables can meet the customer requirements.

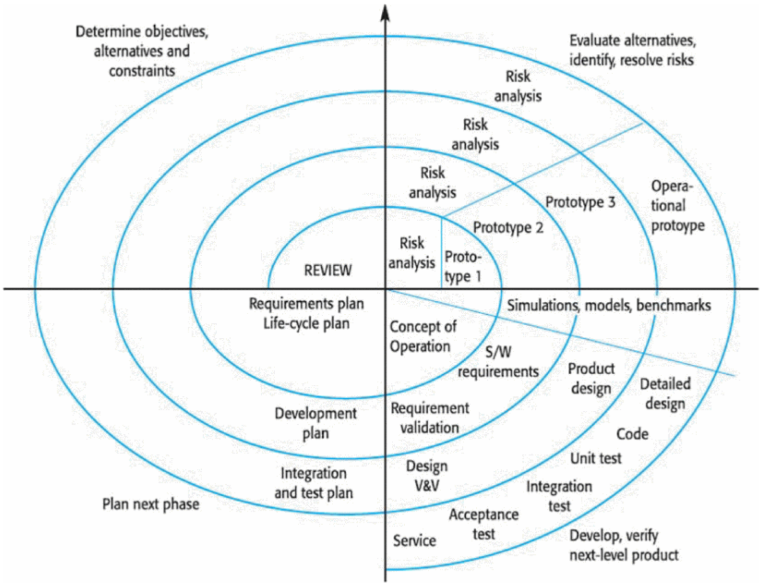


Figure 2.5 Spiral Model (Kumar and Bhatia, 2014)

From the previous methodologies mentioned, none of them will involve users in the development. User involvement will only present after the system is completed. Therefore, rapid application development focuses on the early involvement of users (Shaydulin and Sybrandt, 2017). With this, feedback can be received earlier and respective changes on the requirements can be done earlier. However, this methodology requires developers that are skilful and experienced (Kumar and Bhatia, 2014), as prototypes are needed for the user to test or verify.

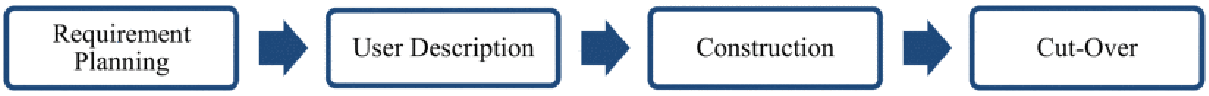


Figure 2.6 Rapid Application Development (Kumar and Bhatia, 2014)

On top of that, scrum is another methodology that requires high maturity level among the developers (Kumar and Bhatia, 2014). The system will be separated by the project manager into multiple tiny tasks. Then, the developers need to be responsible and proactive in completing the tasks. After a short and fixed period, the system will update until the complete system is delivered (Shaydulin and Sybrandt, 2017). Usually, this methodology is adopted in a large project, which handled by a small team.

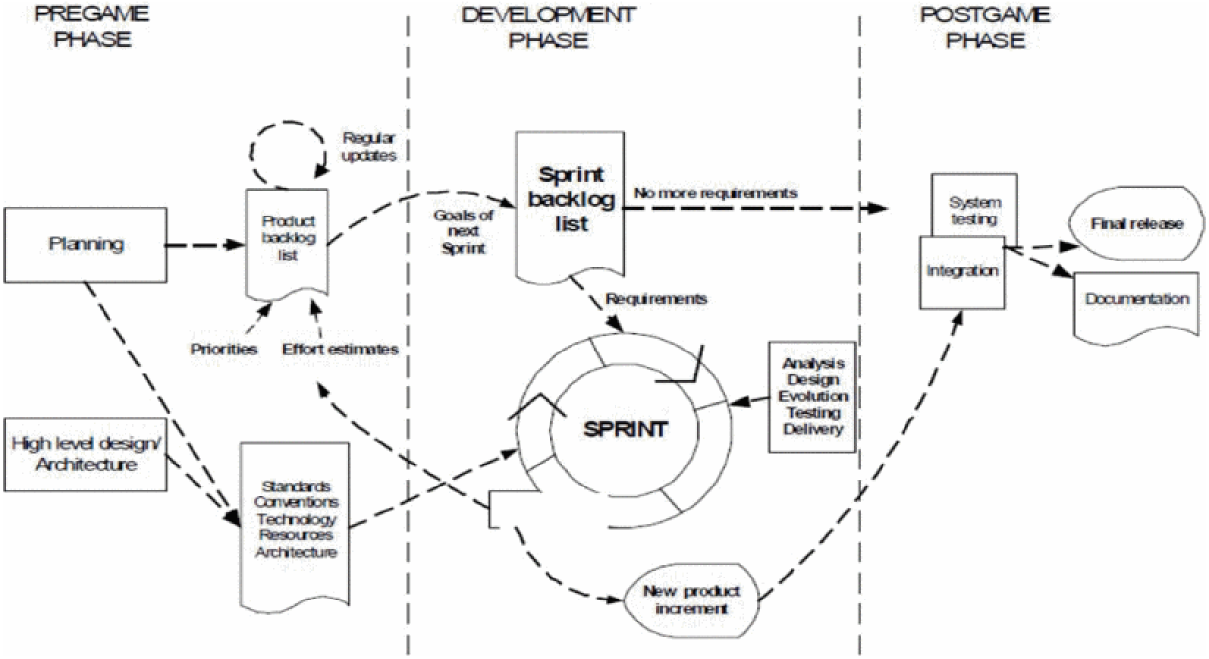


Figure 2.7 Scrum (Kumar and Bhatia, 2014)

Besides, another agile methodology would be feature-driven development. This methodology allows fast development as the system is completed via feature-based iterations. Feature-driven development involves five activities (Kumar and Bhatia, 2014), which are shown in the figure below. However, this methodology is not good at handling the requirement changes. If any changes on the requirements, deep refactoring will be needed.

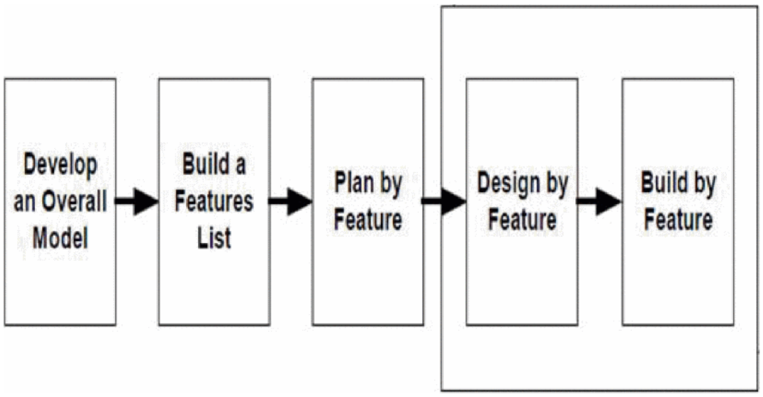


Figure 2.8 Feature-Driven Development (Kumar and Bhatia, 2014)

Furthermore, the next methodology will be test-driven development. According to Kumar and Bhatia (2014), using this methodology will require the developers to prepare test cases before any implementation of the system begins. Then, the implementation is done via several iterations to pass these test cases. After the basic implementation is ready, further refactoring will be conducted so the performance of the implementation can be maximized.

At the same time, with the availability of the test cases, the changes or refactoring will not break the system easily. Instead, the developers can identify the possible defects that lead to such errors after changes are made. However, this methodology will require extra effort in preparing the test cases, which lead to declined productivity (Shaydulin and Sybrandt, 2017).

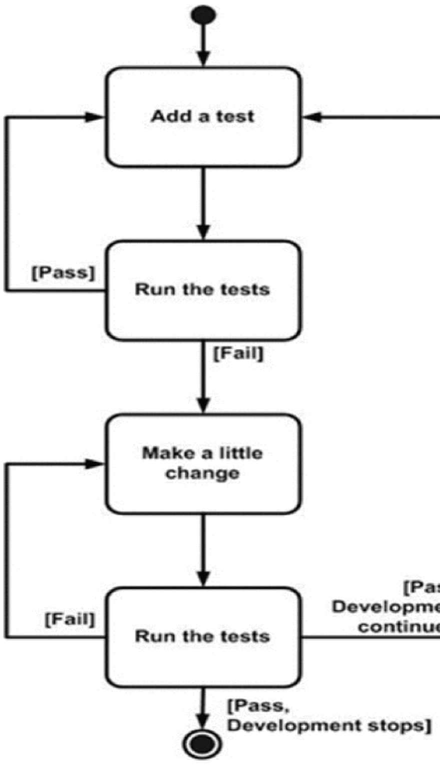


Figure 2.9 Test-Driven Development (Kumar and Bhatia, 2014)

Last but not least, although joint application design is not a complete methodology, it emphasizes user involvement throughout the cycle of the development. According to Shaydulin and Sybrandt (2017, p.7), joint application design is more like a technique used in client requirements collection. By having this methodology, the meeting conducted will be more format and effective in collecting requirements from the clients. Additionally, it can maintain the connection between the clients and the developers throughout the life span of the project.

2.4.1. Finding

Undeniably, each methodology has its strengths and weaknesses. Therefore, to select the most suitable methodology for this project, the methodology has to fit the nature of the project.

First and foremost, the health record system is a solo project. It only has one person who needs to handle all of the SDLC activities. Any methodology that only suitable for a team shall be excluded.

Next, this project involves multiple platforms, which some features only available in either one platform. Therefore, iteration development is very crucial in this project. By completing the system part by part, it can only reduce the possibility to have any rework.

Furthermore, the end-users of the health records system are the patient and the medical staff. As this project has a short development period, it is difficult to involve end-users in the development process, as it might delay the delivery of the system. Therefore, the availability of user involvement shall be avoided in this project.

By comparing the remaining methodologies, the test-driven development will require extra resource in the test case preparation. Besides, if feature-driven development is adapted, refactoring of the implementation is difficult to be conducted when the requirement changes.

Based on the first nature of the project, the Spiral model and Scrum should be avoided as both methodologies are difficult to be adopted in a solo project. Next, the Waterfall model and V-shaped model are both sequential methodologies. Without iteration, it will be difficult to complete the system. Moreover, rapid application development is not suitable for the project context, as it involves users in the development process. Due to the limited time and staff in handling the end-users, having users in the development process will lead to frequent changes in the requirement. Hence, this will make the project to take forever for completion.

Based on the comparison shown in Appendix A, it can conclude that the methodology used in this project is the Phased Development Methodology, which has the characteristics of the iterative model. This is because it can fit the nature of the project, which are solo project, multiplatform development and time constraint.

2.5. Similar Systems

Now, a health records system is never a new thing. Instead, many hospitals or medical institutions had their electronic health records system, which aids them in managing all of the health records. Additionally, this system can help the medical staff by providing them with a complete interpretation of the patient health condition, so the medical staff and give an accurate treatment and diagnosis. Since there are existing systems, this literature review is conducted to investigate the common feature provided by these systems and list out the necessary features that should be implemented in the project.

In total, 11 health records systems were discovered and compared. Among 11 health records systems, 6 systems were developed and used in foreign countries, such as the USA and England. The remaining systems are all from Malaysia.

2.5.1. Systems developed by developers from other countries

In the system comparison, the systems developed by foreigners are MTBC Apps (2016), Capzule (2010), Medical Records (Vladimir, n.d.), My Medical (2012), Genex - Health Records (n.d.) and mHealth by Aliakbarpoor, Comai and Pozzi (2017). To identify the common features that are provided by these systems, Table B-1 is formed and put as Appendix B.

Overall, there are a few systems available in the medical industry. Each of them emphasizes different features. However, these systems are providing common services such as medications reminder, and health rates measurement. Other features include appointment scheduling, health condition visualization, and immunization tracking. Additionally, some of these systems do not allow the patients to enter their health records, instead, they just limited to those records issued by the official medical institutions.

Other than that, some of the systems will receive the patient’s health records, but it does not involve any medical staff. These records are all entered by the patients, which may lead to improper data entry due to the lack of knowledge in the medical field.

2.5.2. Systems developed by developers from Malaysia

In the system comparison, the systems developed by Malaysia developers are IntelSys (2010), Doctor2U (n.d.), Teleme (2017), GetDoc (2017) and DoctorOnCall (2020). By comparing the mentioned system, Table B-2 in Appendix B: System Comparison is created to show the features these systems are providing.

After doing some research on the existing applications/systems in Malaysia, there is less application or system that allows user to store their health record. Most of the systems serve as a platform that provides services such as doctor consultation, medication order, and appointment scheduling with the doctor. These services are very useful and contributed a lot in promoting e-medical. However, all these services will only be used after the patients have an illness.

Despite the majority of the systems are not health records systems, some of the features are recommended to be provided in this project. Based on the analysis, appointment scheduling and online doctor consultation are the most common features. However, since online doctor consultation is not within the scope of this project, it shall be removed from the implementation list and online appointment scheduling should be included.

It is also very vital for one to be aware of their health condition. The patients need to update their health condition frequently, so they can monitor the trend of their health condition. With the aid of technology, analysis of these health conditions can aid the patients for earlier detection of illnesses or disease so they can take action early. On top of that, for the current protocol, the doctors only do diagnosis based on the patient current health condition as well as the history of health records the doctors have. Therefore, it is better if the daily health condition is recorded.

Based on the comparison, the current systems have limited feature in analysing the patient’s health condition. On top of that, the records are isolated from each other, which will be better to share with other parties in some emergencies. This is because, with a complete set of health condition history, the doctor can carry out a more complete and accurate diagnosis of the patient.

In short, although each health records system has a different focus and provides different features, some of the features are common and shall be included in this project. The common features can be concluded as follows:

1. Appointment Scheduling
2. Medication Reminder
3. Health Rate Measurement

Additionally, some features are rarely provided by the existing system. Therefore, in this project, the system shall include these features so the system can provide comprehensive medical support. These features include:

1. Data Visualization
2. Health Records Management with Data Sharing

With data visualization, the system will graphically present the data. Thus, the users can understand their current situations better, as it may be too difficult for the patients to read and understand the raw data. Additionally, medical records usually consist of terms that are professional and unique which are difficult to be understood by common people. In addition to the health records management system, this system allows the sharing of data. Thus, the data can be viewed by medical staff from different medical institutions. Ultimately, this reduces the chance of giving an inaccurate diagnosis to the patients.

2.6. Tools

The tool is very important for any development of software. Without a tool, software engineers need to create an application from scratch. As the saying goes, goods tools are prerequisite to the successful execution of a job. Therefore, this literature review will focus on determining the most suitable tool, so the chance of success is higher.

As the population of mobile users increases, the demands of the mobile application will be increased and the expected quality should be higher as time passes. In the beginning, native development would be the best option as it can provide an excellent experience to the users. However, in the current market, more phone providers had created their operating system. It will take forever for one to develop an application that can fit into all the operating system unless the company has enough workforce to handle each platform. Therefore, native mobile development is no longer efficient as the time and cost for enhancement and maintenance are too high (Brito et al., 2018). This issue will be enlarged if the company is in small or medium-sized.

To tackle this issue, a hybrid mobile development should be adopted and among all the options, React Native should be the best solution in terms of profitability and quality (Brito et al., 2018). For example, a React Native development team can implement React-based web solutions, it is only necessary to change the mobile interface components for the web, maintaining much of the business logic of a product.

In the end, React Native become the best option, as the team can focus only on a single type of language while developing front-end solutions for different platforms (Brito et al., 2018). Therefore, it can ensure quality and speed in the development of mobile applications. Additionally, Biørn-Hansen and et al (2019) found that the respondents from their survey selected React Native as the framework that they are most interested due to its novelty.

Other than that, in terms of familiarity and usage, Biørn-Hansen and et al (2019) indicates that PhoneGap (also known as Apache Cordova) would be the most suitable framework. PhoneGap was released since 2009. In 2019, the stable release of PhoneGap is in version 9.0.0. The documentation is more complete compared to React Native and this indicates getting helps from others would be easier.

Brito and et al (2018) should include the comparison with frameworks that use other programming languages, such as C# and Python, rather than just JavaScript. However, their result is trustable as the data were based on the developed applications that use JavaScript framework. For Biørn-Hansen and et al (2019), they need to increase their sample size, so the result can cover more cases rather than just the small range of people. With the mentioned improvement, the result of the research can only be better and more trustable.

Undeniably, both pieces of research supported that React Native is a good option for hybrid application development. Although in this project, IOS is not covered. It would be good to use React Native, as duplication of the implementation can be avoided if IOS application is requested by the users. Additionally, due to the nature of this project, React Native would be the best option for a small team, as suggested by Brito and et al (2018).

In a nutshell, the framework or tool that will be used in this solo project would be React Native. React Native is the most suitable option, as it can provide expandability and fit the nature of this project, which are solo work and limited time.

2.7. Conclusion

In short, this literature review successfully studied 5 of the areas mentioned and concluded with different results. To implement a health records system, the system should be able to ensure data completeness and at the same time, it will not affect the existing workflow. Instead, it shall boost the performance when the medical staff use this system in their working routine.

Next, cross-platform application development for the solo project requires a good development skill. Therefore, the developer of a solo project must have a strong understanding of the tools, frameworks and programming languages used in the development. Furthermore, based on the comparison in Appendix A, the most suitable methodology of this solo project is the Phased Development Methodology.

Moreover, via the comparison between the similar systems, some common features were identified and shall present in this new system. Additionally, due to the lack of some features in the existing systems, it is recommended to include those features in this new system as well. Last but not least, via the researches done on the tools, this literature review can conclude that React Native would be the best fit tool in the development of this new health records system.

CHAPTER 3

METHODOLOGY

3.1. Introduction

This chapter explains the methodology adopted as well as the project scheduling for the entire development. The methodology used is Phased development methodology and the implementation of this methodology will be discussed in the following subchapter. Additionally, the output documents in this chapter are the work breakdown structure as well as the Gantt chart, which consists of a progress timeline.

3.2. Phased Development Methodology

Since Phased development methodology is an iterative model, it is used as the development methodology of health records system. Phased development methodology will separate the whole development of the health records system into several phases. Each phase will have different focuses on the development of the system. The focus is determined by the severity level of the modules in the system.

By separating into several phases, the most important part of the system will be completed first, which allows the system to operate at the early stage of the development. On top of that, it can ease the time allocation. For instance, a high-priority task will require more time for completion. Moreover, due to the availability of time constraint, adapting Phased development methodology can ensure the final system can perform those critical actions. As the high-priority tasks are completed at the early stage. Additionally, Figure 3.1 shows the development cycle of Phased development methodology for health records system.

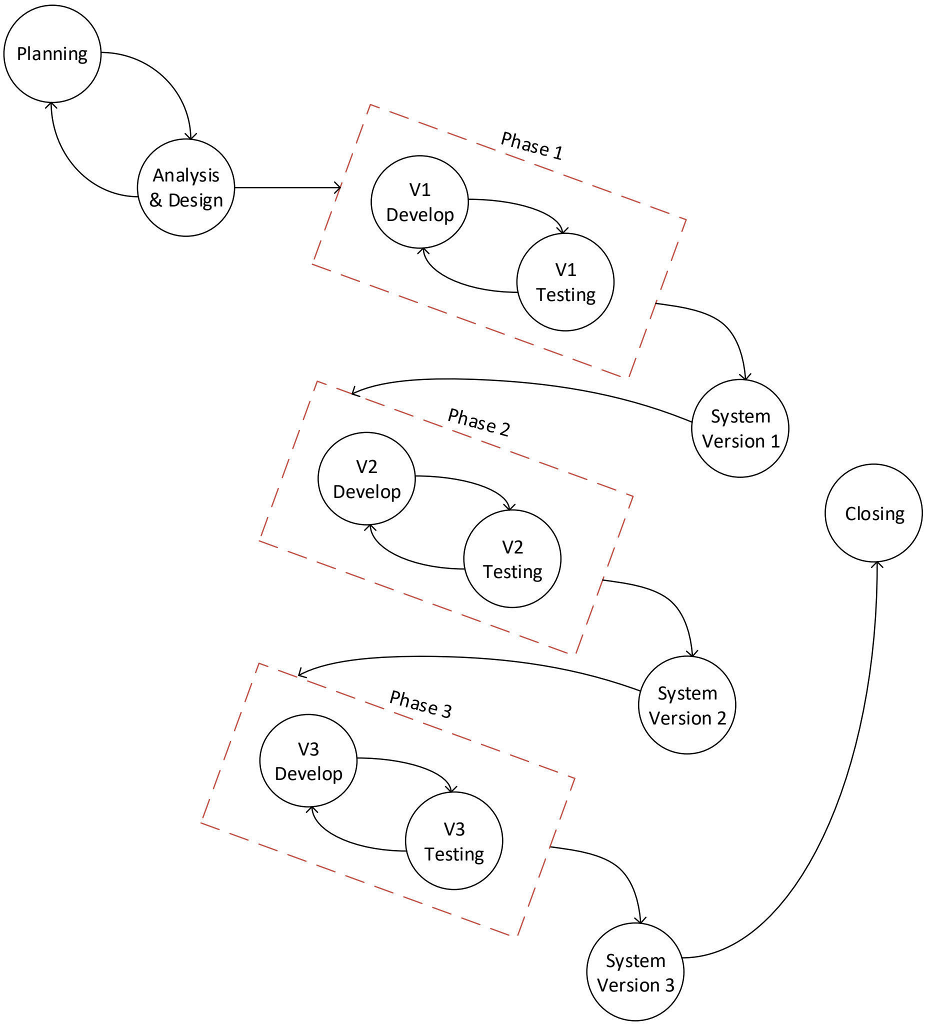


Figure 3.1 Phased Development Methodology

3.2.1. Planning

The planning phase of this project determined the problem encountered by the patient when they tried to access their health records and provide them with a solution that enables flawless sharing of data from multiple medical institutions. By understanding the problem in-depth, more potential issues were identified after an in-depth analysis and interpretation were conducted. The issues discovered are as follows:

1. Health Records Tracking Issue
2. Limited Health Data Interoperability

Then, based on the issues discovered, the objectives of this project were determined. The identified objectives determined the direction of the project development and made sure the latter development on the track. Eventually, this increased the success of this project. The objectives are as follows:

1. To develop a mobile health records application that helps patients to track all of their formal health records while monitoring the users who access these records
2. To develop an interoperable health records system in both web and mobile platforms that allows data sharing by August 2020

According to the identified problem statements and the objectives, the planning phase proceeds with the identification of the project development methodology, the proposed solution as well as the project scopes. Methodology made the development cycle of this project more structured and organized, which became guidance throughout the development cycle of this project. Other than that, the proposed solution demonstrated the architecture of the implementation, which provide the first image of the final deliverables. Moreover, project scopes defined the necessary features in the deliverables for the fulfilment of the project objectives.

3.2.1.1. Requirement Gathering and Elicitation

After the proposal received a green light, further studies on the requirement were conducted to gather requirement for the elicitation. The studies conducted include the distribution of a questionnaire to patients as well as the review on existing systems for common features as well as potential functionalities. Based on the result of the studies, a list of system features was finalized.

3.2.1.1.1. Questionnaire

The questionnaire technique is selected to gather the requirements. These are because the questionnaire contains a list of standard questions that must be asked to each participant. It can ensure a faster requirement gathering process while it can spread among a larger group of people regardless of their status. Other than that, other researches can easily compare the result of this questionnaire since this questionnaire is quantified. Additionally, each answer in the question can help the researcher to decide the actions to be taken, which follows the majority.

The questionnaire managed to get responses from 51 respondents. The questionnaire was conducted to investigate their behaviors and practices in tracking their health condition. With this, the questionnaire can determine the priority of particular features. In total, 10 questions were asked, which can be categorized into 3 groups. The questionnaire was shared on multiple platforms, such as Facebook and WhatsApp. The questionnaire was available for 2 weeks before the data collection ended for this questionnaire.

3.2.1.1.2. Review on existing Systems

In total, 11 systems were reviewed and compared, as shown in Table B-1 and Table B-2 under Appendix B. These systems were developed by both Malaysian companies and foreign companies. Majority of the local systems were focused on the online consultation. Via the comparison of these similar systems, a list of common features was concluded. With this, it can deliver a more thorough system, which provides more service in aiding medical health. The common features are as follows:

1. Appointment Scheduling
2. Medication Reminder
3. Health Rate Measurement

On top of that, some features are recommended to be included in this new health records system. These features are rare especially in those existing systems used by Malaysia hospital or clinic. The recommended features are as follows:

1. Data Visualization

2. Health Records Management with Data Sharing

3.2.1.2. Project Scheduling

With defined project scopes as well as the output of questionnaire and system comparison, a detailed work breakdown was conducted. The work breakdown demonstrated all of the tasks and subtasks that required to be carried. With this, it will provide WBS that can ensure fewer missing tasks.

Moreover, with WBS defined, the project schedule was then conducted to provide a Gantt Chart for the entire development cycle. The Gantt Chart consists of all of the necessary tasks along with the timeline. In the Gantt Chart, each task had an expected delivery time, expected start time as well as end time. By having WBS and Gantt Chart, it provided a guideline for the development, which ensure the project can be delivered within the constraint of cost, time and scope.

3.2.2. Analysis and Design

According to the project scope, an analysis was conducted and respective diagrams were designed to support the implementation of the system. The diagrams produced include use case diagram, data model diagram and data flow diagram. Additionally, use case descriptions were prepared to give detailed information for each use case.

Other than that, prototypes were prepared to decide the design of the system. Hence, improvement can be applied during the implementation of the system. During this phase, both low high-fidelity level and high-fidelity level prototypes were prepared. The low-fidelity level prototypes are wireframe for both mobile and web application. The wireframe shows the process of users conducting activity as well as the display of the content and features within a page. Next, the high-fidelity level prototypes are the design prototype for both mobile and web application. These prototypes contain detail design of the interfaces as well as the user experiences.

3.2.3. Development and Testing

During this phase, the implementation of the system started and separated into three phases. The output of the first phase will be further enhanced in the following phases. The separation of tasks is based on the severity level of each task. The high-priority level tasks will be implemented at the first stage while the tasks with the lowest priority will be implemented last. In each phase, it focusses on different features and each feature contains some testing. The testing conducted is to ensure the further enhancement of the implementation will not break the existing implementation.

3.2.3.1. Phase 1

In the implementation phase 1, this project implemented the features that have the highest priority level. Before that, since it is a new project, it required system setup, which involved the connection between the applications and server.

Then, since the health records are bind to the patient, each patient required to have an account in the system. With an account, the patients can access their health records in different devices. Besides, each medical staff requires an account in the web application. It is important as not everyone can access to the health records, which are the privacy of the patients.

Next, the core of the system is to keep track of the patients' health records. This feature has the highest priority among all of the features. Without this, the users cannot track their health records in the mobile application.

In short, implementation of phase 1 includes all of the systems, which are the web application, mobile application and the backend server. Additionally, it took the longest time for completion.

3.2.3.2. Phase 2

In phase 2, it focused on features that have a medium-priority level. Getting a number at the hospital and waiting there is an issue to the patient. To tackle this issue, appointment scheduling implemented in the system. With this, it can ease the patients in meeting their doctors. On top of that, the environment of the hospital or clinic can be improved as fewer patients are on the spot since the patients can only wait for their consultation turn at their home or elsewhere. In short, phase 2 involved more development on the mobile application.

Moreover, to tackle the data leakage, the system is required to track every access to the patients' health records. With this, when data leakage occurs, the system can identify malicious users by checking the access request records in the database.

3.2.3.3. Phase 3

During the implementation of phase 3, it is the last phase of implementation of this project. Therefore, it only focused on features with the least level of priority. According to the questionnaire done, only a few of them required a reminder for the medication refill. The patients allow to set alarm or create an event in their calendar to remind them to either take or refill their medication.

Furthermore, the patients may change their detail, such as the email or address. Therefore, phase 3 implemented an account management system, which allows the patients to update their information. This feature is not limited to the patient only, but the medical staff as well. Hence, it is implemented on both the web application and the mobile application. In short, phase 3 mainly involves the development of a mobile application.

3.2.4. Closing

Before delivering the system, user acceptance testing was conducted to verify the functionalities of the system. Then, the documentation of the system was completed, which include the proposal of the project, the implementation design of the system as well as the actual system. A detail description of the system was attached to explain the functionalities within the system.

Besides, the presentation slide was prepared to explain the development process of the entire project. This slide included the content from the planning stage of this project until the project closure. At the same time, a poster was created to provide a brief insight into the entire project.

3.3. Project Tools

3.3.1. Architecture

This project implements a three-tier client-server architecture, which consists of three different layers. Separation of the systems allows other systems to continue functioning if one of them is down or broken. On top of that, it can enhance security by restricting direct access to the database.

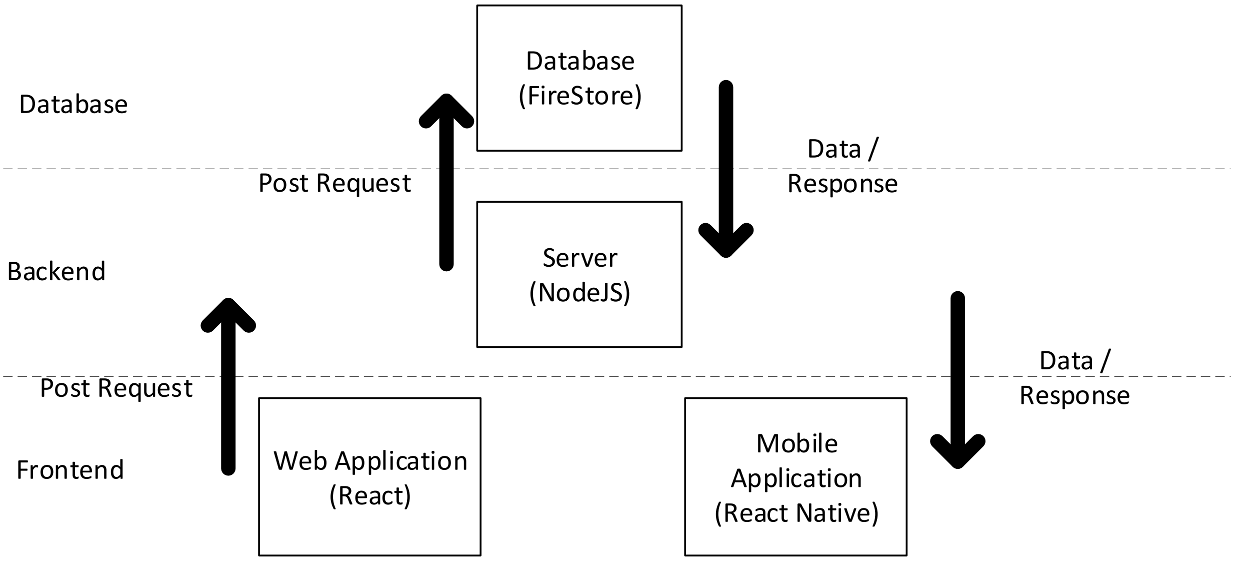


Figure 3.2 System Architecture

Frontend layer displays the information to the system users. It only requested data from the server. At the backend layer, the server fetches all of the relevant data for processing. It then sent the processed data to the requested frontend systems based on the request received. At the last layer, the database will store all of the user's data, which can be retrieved by the server only.

3.3.2. Fire Store

FireStore is a Google service that provides a NoSQL database. NoSQL database allows flexibility in storing the data, as the record is not restricted to a particular format. Instead, it allows complex objects that are nested. On top of that, FireStore consists of a complex querying mechanism, which chains multiple filtering and sorting (Cloud Firestore | Firebase, 2020). Additionally, by default, the data are indexed. Hence, the performance of each query is better, as it is affected by the size of the result set rather than the data set. On the other words, querying from a large data set will not reduce the performance. Instead, the query that returns a large result will affect the performance.

3.3.3. React Native

In this project, React Native is used as the development framework for the mobile application. React Native uses JavaScript programming language. It allows the creation of a native application in both Android and iOS operating systems with a single implementation. Additionally, React Native has fast refresh or hot reloading (Rajput, 2018; React Native, 2020), which allows the changes to be viewed directly without extra time for native builds.

Moreover, Rajput (2018) mentioned that React Native as a JavaScript framework allows plugins from third parties. This allows fast pace development, as some components were developed and shared by other developers. Although this project focuses on Android mobile application, adapting React Native for the mobile side development ensured extendibility of this project, when an iOS application is requested. This also ensures consistency between Android application and iOS application.

3.3.4. React

React uses JavaScript programming language for its development. In this project, it will be used in website development. React can create virtual DOM. Traditionally, any changes in the webpages will require an update on all of the components in the page. With React, it only updates those components, which changes applied. Thus, React will require fewer component updates, which save a lot of resources (John, 2017) since only the affected components are re-rendered rather than the whole page (React, 2020).

On top of that, due to the availability of virtual DOM, John (2017) mentioned that React allows reuse of component in React project. With this, the project is easier for development as well as maintenance.

3.3.5. Node.js

Node.js application is written in JavaScript (About | Node.js, 2020) and it is used in the backend server of this project. Similar to React and React Native, it allows the usage of third-party modules or packages. Additionally, it reduces the time to catch up a new programming language, as React, React Native and Node.js all use JavaScript programming language. Moreover, other benefits of using Node.js include simplicity in accessing JavaScript's knowledge and community support from a huge amount of active developers as well as those big companies like Microsoft (Oleg, 2019).

3.3.6. TypeScript

TypeScript is similar to JavaScript in terms of syntax and semantics. However, it provides a stricter rule in checking the errors (TypeScript - JavaScript that scales., 2020). Thus, it greatly reduces the chance to have syntax error as well as runtime errors. By having TypeScript, the JavaScript code is clearer and simple since the developers are forced to define the type inference.

3.3.7. Axure RP

As a prototyping tool, Axure RP enables the creation of interactive UIs without the needs of coding. In Axure RP, it creates wireframes, which shows the workflow of particular actions (Axure RP 9, 2020). With Axure RP, the design of the prototype can be tested, which aid the developers to decide a better design during the implementation (Axure RP 9, 2020). Additionally, Axure RP was used to create the low-fidelity prototype for this project.

3.4. Work Plan

3.4.1. WBS

1.0 Planning

- 1.1 Study Background of the Problem
- 1.2 Define Problem Statements
- 1.3 Define Objectives
- 1.4 Propose Project Solution
  - 1.4.1 Design Architecture
- 1.5 Propose Project Approaches
  - 1.5.1 Research Approach
  - 1.5.2 Development Approach
- 1.6 Define Scope
  - 1.6.1 Identify Target Users
  - 1.6.2 Identify Covered Features
  - 1.6.3 Identify Uncovered Features
- 1.7 Requirement Gathering and Elicitation
  - 1.7.1 Distribute Questionnaire
    - 1.7.1.1 Generate Questions
    - 1.7.1.2 Distribute Questionnaire
    - 1.7.1.3 Analyse and Interpret the Findings
  - 1.7.2 Review Similar Systems
    - 1.7.2.1 Review on Systems developed by developers from other countries
      - 1.7.2.1.1 Review on MTBC Apps
      - 1.7.2.1.2 Review on Capzule
      - 1.7.2.1.3 Review on Medical Records
      - 1.7.2.1.4 Review My Medical
      - 1.7.2.1.5 Review on Genex
      - 1.7.2.1.6 Review on mHealth
      - 1.7.2.1.7 Compare all of the systems developed by developers from other countries
    - 1.7.2.2 Review on Systems developed by Malaysian
      - 1.7.2.2.1 Review on IntelSys

1.7.2.2.2 Review on Doctor2U

1.7.2.2.3 Review on Teleme

1.7.2.2.4 Review on GerDoc

1.7.2.2.5 Review on DoctorOnCall

1.7.2.2.6 Compare all of the systems developed by Malaysian

1.7.2.3 Identify the common and recommended features

1.8 Literature Review

1.8.1 Identify the potential concerns in the health records system implementation

1.8.2 Understand the cross-platform application development

1.8.3 Determine the most suitable methodology

1.8.4 Investigate the tools for the development

1.9 Project Scheduling

1.9.1 Generate WBS

1.9.1.1 Identify the main features

1.9.1.2 Breakdown features

1.9.2 Generate Gantt Chart

1.9.2.1 Determine Task Dependency

1.9.2.2 Estimate Effort

1.9.2.3 Estimate Start Time and End Time

1.9.2.4 Create a Gantt Chart

2.0 Analysis & Design

2.1 Design Use Case Diagram

2.2 Prepare Use Case Description

2.3 Design Data Model Diagram

2.4 Design Data Flow Diagram

2.5 Low-Level Prototyping

2.5.1 Prepare wireframe for mobile application

2.5.2 Prepare wireframe for web application

2.6 High-Level Prototyping

2.6.1 Prepare design prototype for mobile application

2.6.2 Prepare design prototype for web application

3.0 Phase 1

3.1 System Setup

3.1.1 Implementation

3.1.1.1 Create a repository for backend server

3.1.1.2 Create a repository for mobile application

3.1.1.3 Create a repository for web application

3.1.1.4 Setup FireStore database

3.1.1.5 Configure the connection between backend server and mobile application

3.1.1.6 Configure the connection between backend server and web application

3.1.1.7 Configure the connection between backend server and FireStore database

3.1.2 Testing

3.1.2.1 Test the connection between backend server and FireStore database

3.1.2.2 Test the connection between backend server and mobile application

3.1.2.3 Test the connection between backend server and web application

3.2 Account Creation

3.2.1 Implementation

3.2.1.1 Create web UI to create a medical staff account

3.2.1.2 Create mobile UI to create a patient account

3.2.1.3 Implement an algorithm to create an account and store in the database

3.2.1.4 Create web UI for account login

3.2.1.5 Create mobile UI for account login

3.2.2 Testing

3.2.2.1 Test the result of the account creation so it contains data fields that match the user input

3.3 Health Records Tracking

3.3.1 Implementation

3.3.1.1 Create web UI for medical prescription insertion and update

3.3.1.2 Create web UI for medication record insertion and update

3.3.1.3 Create web UI for lab test result insertion and update

3.3.1.4 Implement an algorithm for insertion and update of health records

3.3.1.5 Implement an algorithm to retrieve patient's health records that are grouped by types

3.3.1.6 Create mobile UI to display patients' history of health records that are created by medical staffs

3.3.1.7 Implement an algorithm that analyses the health records of a patient

3.3.1.8 Create mobile UI to display the analysis of the health records

3.3.1.9 Create mobile UI to update the patient's health conditions

3.3.1.10 Implement an algorithm for the update of patient's health conditions

3.3.2 Testing

3.3.2.1 Test the retrieval of inserted data contains the same size, save data fields and grouped in the correct types

3.3.2.2 Test the result of the analysis whether it has the expected outcome

4.0 Phase 2

4.1 Appointment Scheduling

4.1.1 Implementation

4.1.1.1 Create web UI for doctors to set the appointment timeslots

4.1.1.2 Implement an algorithm to store the appointment timeslots

4.1.1.3 Create mobile UI for appointment arrangement

4.1.1.3.1 Create UI to display and select the available doctor

4.1.1.3.2 Create UI to display and select a timeslot of the selected doctor

4.1.1.4 Implement an algorithm to store the appointment made by the patient

4.1.1.5 Implement an algorithm to retrieve all of the appointments made, which related to the medical staff

4.1.1.6 Implement an algorithm to retrieve all of the appointments made, which related to the patient

4.1.1.7 Create web UI to display all of the appointments made by patients

4.1.1.8 Create mobile UI to display all of the appointments made

4.1.1.9 Implement an algorithm to notify the doctor and patients on the nearing appointments

4.1.1.10 Create web toast message to notify the doctor for the nearing appointments

4.1.1.11 Create mobile push notification to notify patients on the nearing appointments

4.1.1.12 Create web UI for appointment cancellation

4.1.1.13 Create mobile UI for appointment rescheduling and cancellation

4.1.1.14 Implement an algorithm to update the appointment

4.1.2 Testing

4.1.2.1 Test the appointment timeslots after the doctors set their timeslots

4.1.2.2 Test the status and content of the appointment after creation, rescheduling and cancellation of an appointment

4.2 Health Records Access by Patients

4.2.1 Implementation

4.2.1.1 Implement a filtering algorithm to list all of the patients that have any data field, which matches the keyword entered

4.2.1.2 Create web UI to filter and display the list of relevant patients

4.2.1.3 Create web modal to get the medical staff's confirmation on the records retrieval

4.2.1.4 Create mobile modal to get the patient's authorization for the records access

4.2.1.5 Implement the algorithm to check the confirmation and authorization received from both medical staff and patient

4.2.1.6 Implement the algorithm to store the access request in the database

4.2.1.7 Improve the existing algorithm that retrieves the patient’s health records so it can return the analysis as well

4.2.1.8 Create web UI to retrieve the health records of a particular patient and the analysis of the records

4.2.2 Implementation to tackle Emergency

4.2.2.1 Create mobile UI to grant permission to other users of the system to authorize the data access request

4.2.2.2 Implement an algorithm to add the permission-granted users in the database

4.2.2.3 Improve web UI so the medical staff can select “emergency” option when requesting for accessing the health records

4.2.3 Testing

4.2.3.1 Test the result of the filtering algorithm so it matches the expected output

4.2.3.2 Test the authorization to access the health records

4.2.3.3 Test the presence of an access request in the database

4.2.3.4 Test the presence of permission-granted users

5.0 Phase 3

5.1 Medication Reminder

5.1.1 Implementation

5.1.1.1 Improve mobile UI that display the medication records to allow the patient to add either an event in the calendar or an alarm

5.1.1.2 Create mobile push notification to remind the patient on the medication intake

5.1.1.3 Create mobile push notification to remind the patient to refill the medication

5.1.1.4 Create web toast message to notify the doctor on the finished medication of the particular patient

5.1.1.5 Create web UI to amend on the medication record

5.1.1.6 Implement an algorithm to update the medication record amendments

5.1.2 Testing

5.1.2.1 Test the presence of amendments after update of the medication record

5.2 Account Management

5.2.1 Implementation

5.2.1.1 Create web UI to update the detail of medical staff account

5.2.1.2 Create mobile UI to update the detail of the patient account

5.2.1.3 Implement an algorithm to update the account detail

5.2.2 Testing

5.2.2.1 Test the updated detail of the account has a similar value as the user input

6.0 Closing

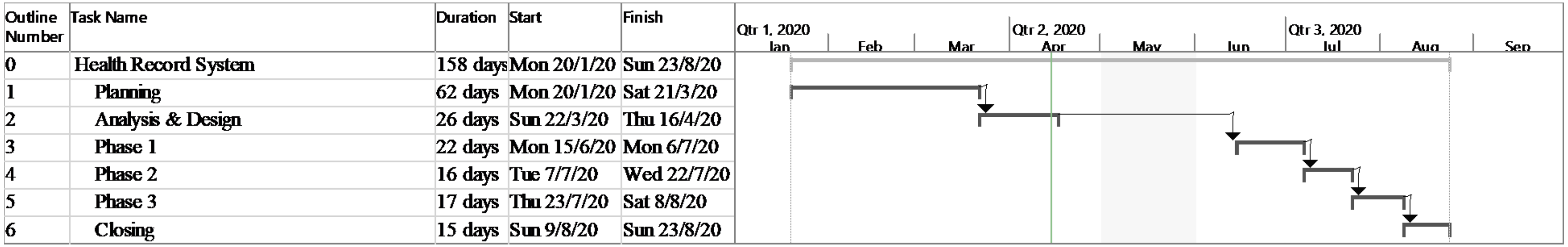
6.1 Conduct User Acceptance Test

6.2 Finalized the documentation of the system

6.3 Prepare the presentation slide

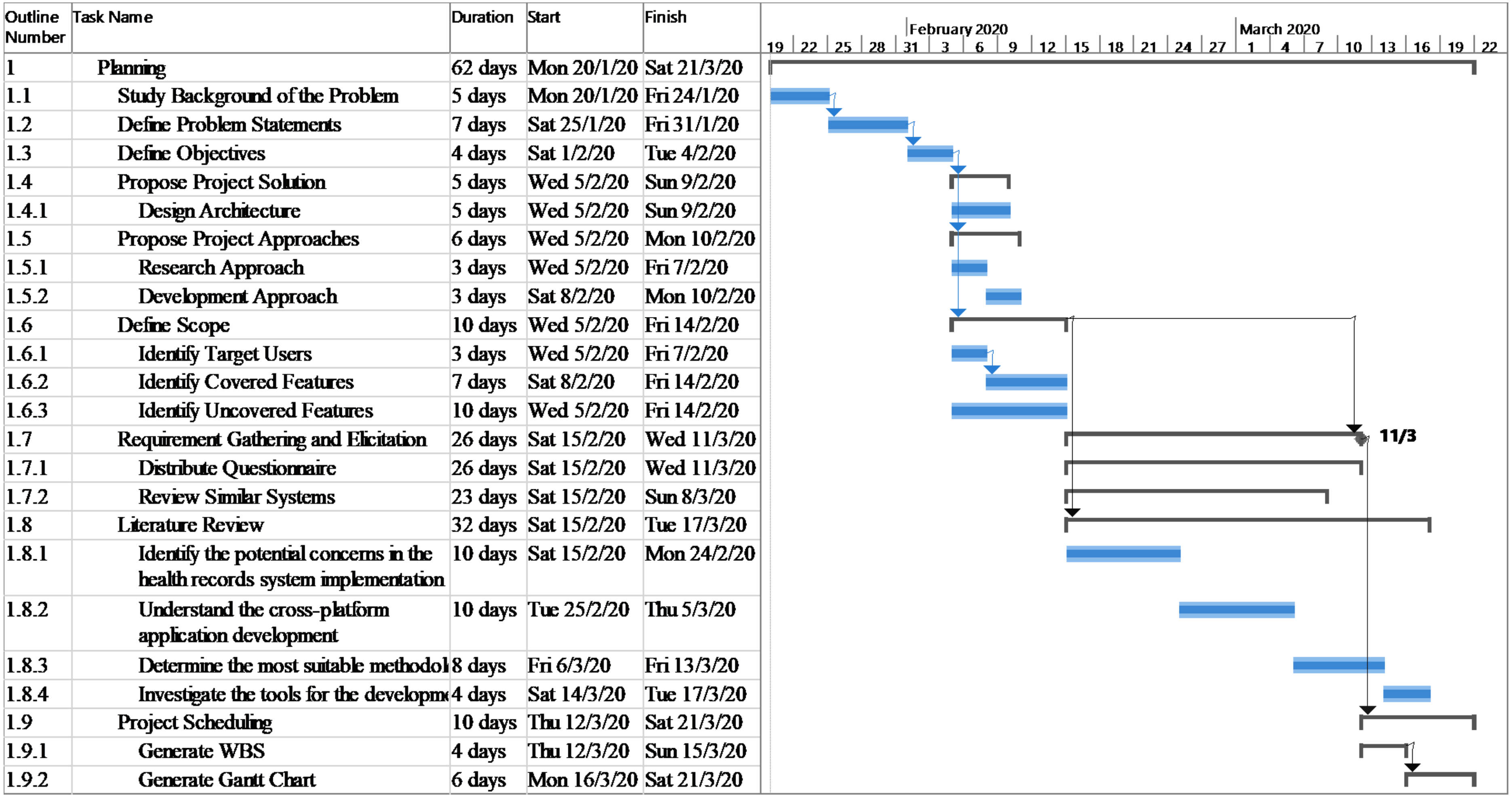
6.4 Prepare a poster for the system

3.4.2. Gantt Chart



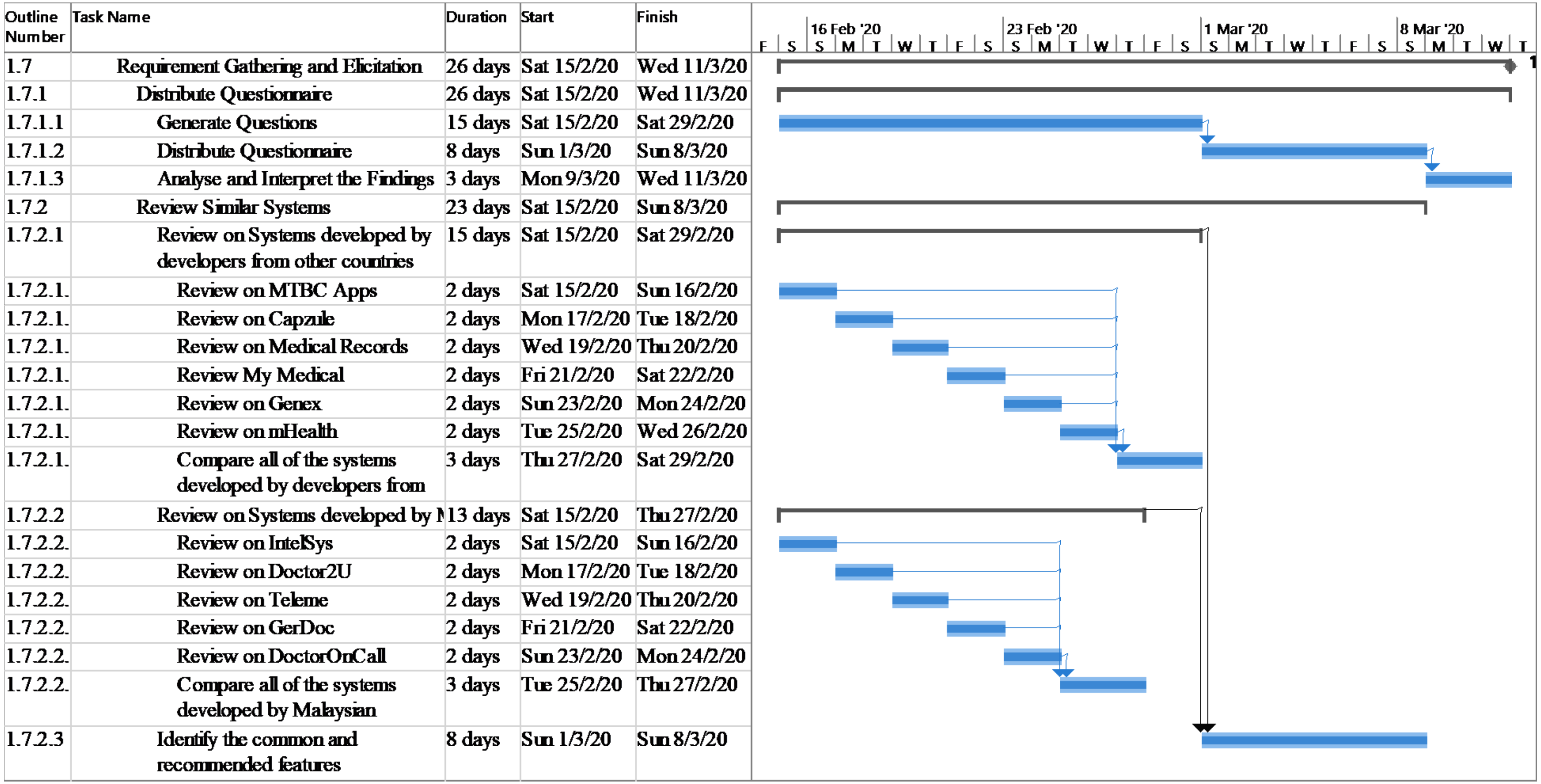
Outline Number	Task Name	Duration	Start	Finish
0	Health Record System	158 days	Mon 20/1/20	Sun 23/8/20
1	Planning	62 days	Mon 20/1/20	Sat 21/3/20
2	Analysis & Design	26 days	Sun 22/3/20	Thu 16/4/20
3	Phase 1	22 days	Mon 15/6/20	Mon 6/7/20
4	Phase 2	16 days	Tue 7/7/20	Wed 22/7/20
5	Phase 3	17 days	Thu 23/7/20	Sat 8/8/20
6	Closing	15 days	Sun 9/8/20	Sun 23/8/20

Figure 3.3 Schedule Overview of Health Record System



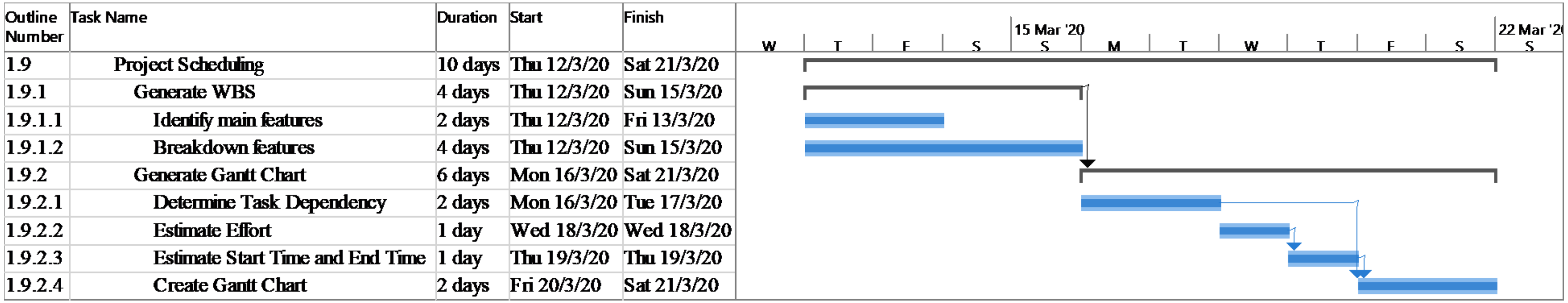
Outline Number	Task Name	Duration	Start	Finish
1	Planning	62 days	Mon 20/1/20	Sat 21/3/20
1.1	Study Background of the Problem	5 days	Mon 20/1/20	Fri 24/1/20
1.2	Define Problem Statements	7 days	Sat 25/1/20	Fri 31/1/20
1.3	Define Objectives	4 days	Sat 1/2/20	Tue 4/2/20
1.4	Propose Project Solution	5 days	Wed 5/2/20	Sun 9/2/20
1.4.1	Design Architecture	5 days	Wed 5/2/20	Sun 9/2/20
1.5	Propose Project Approaches	6 days	Wed 5/2/20	Mon 10/2/20
1.5.1	Research Approach	3 days	Wed 5/2/20	Fri 7/2/20
1.5.2	Development Approach	3 days	Sat 8/2/20	Mon 10/2/20
1.6	Define Scope	10 days	Wed 5/2/20	Fri 14/2/20
1.6.1	Identify Target Users	3 days	Wed 5/2/20	Fri 7/2/20
1.6.2	Identify Covered Features	7 days	Sat 8/2/20	Fri 14/2/20
1.6.3	Identify Uncovered Features	10 days	Wed 5/2/20	Fri 14/2/20
1.7	Requirement Gathering and Elicitation	26 days	Sat 15/2/20	Wed 11/3/20
1.7.1	Distribute Questionnaire	26 days	Sat 15/2/20	Wed 11/3/20
1.7.2	Review Similar Systems	23 days	Sat 15/2/20	Sun 8/3/20
1.8	Literature Review	32 days	Sat 15/2/20	Tue 17/3/20
1.8.1	Identify the potential concerns in the health records system implementation	10 days	Sat 15/2/20	Mon 24/2/20
1.8.2	Understand the cross-platform application development	10 days	Tue 25/2/20	Thu 5/3/20
1.8.3	Determine the most suitable methodol	8 days	Fri 6/3/20	Fri 13/3/20
1.8.4	Investigate the tools for the developm	4 days	Sat 14/3/20	Tue 17/3/20
1.9	Project Scheduling	10 days	Thu 12/3/20	Sat 21/3/20
1.9.1	Generate WBS	4 days	Thu 12/3/20	Sun 15/3/20
1.9.2	Generate Gantt Chart	6 days	Mon 16/3/20	Sat 21/3/20

Figure 3.4 Schedule Overview of Planning Phase



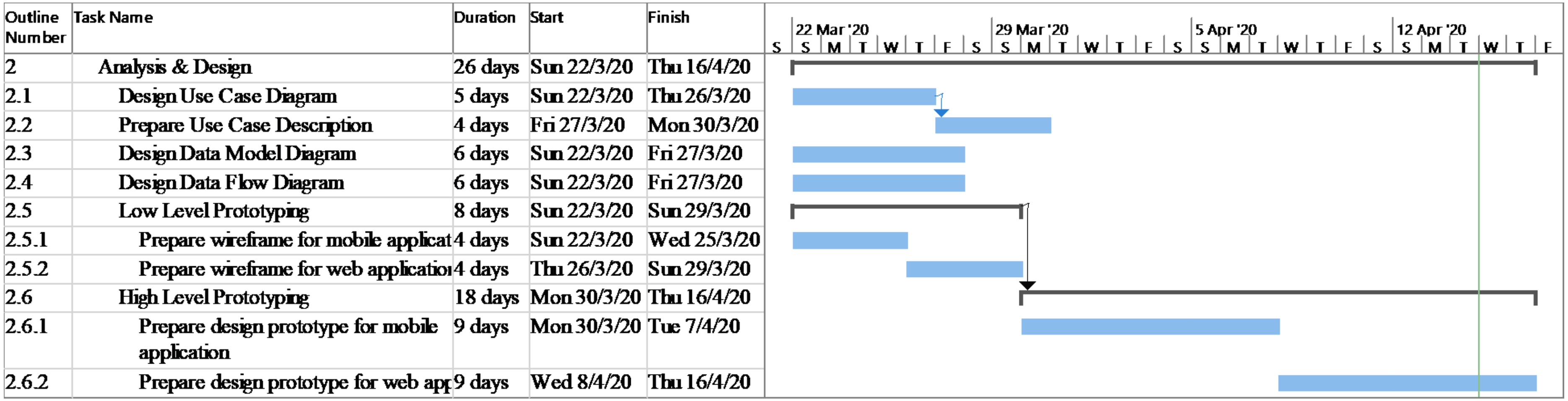
Outline Number	Task Name	Duration	Start	Finish
1.7	Requirement Gathering and Elicitation	26 days	Sat 15/2/20	Wed 11/3/20
1.7.1	Distribute Questionnaire	26 days	Sat 15/2/20	Wed 11/3/20
1.7.1.1	Generate Questions	15 days	Sat 15/2/20	Sat 29/2/20
1.7.1.2	Distribute Questionnaire	8 days	Sun 1/3/20	Sun 8/3/20
1.7.1.3	Analyse and Interpret the Findings	3 days	Mon 9/3/20	Wed 11/3/20
1.7.2	Review Similar Systems	23 days	Sat 15/2/20	Sun 8/3/20
1.7.2.1	Review on Systems developed by developers from other countries	15 days	Sat 15/2/20	Sat 29/2/20
1.7.2.1.	Review on MTBC Apps	2 days	Sat 15/2/20	Sun 16/2/20
1.7.2.1.	Review on Capzule	2 days	Mon 17/2/20	Tue 18/2/20
1.7.2.1.	Review on Medical Records	2 days	Wed 19/2/20	Thu 20/2/20
1.7.2.1.	Review My Medical	2 days	Fri 21/2/20	Sat 22/2/20
1.7.2.1.	Review on Genex	2 days	Sun 23/2/20	Mon 24/2/20
1.7.2.1.	Review on mHealth	2 days	Tue 25/2/20	Wed 26/2/20
1.7.2.1.	Compare all of the systems developed by developers from	3 days	Thu 27/2/20	Sat 29/2/20
1.7.2.2	Review on Systems developed by M	13 days	Sat 15/2/20	Thu 27/2/20
1.7.2.2.	Review on IntelSys	2 days	Sat 15/2/20	Sun 16/2/20
1.7.2.2.	Review on Doctor2U	2 days	Mon 17/2/20	Tue 18/2/20
1.7.2.2.	Review on Teleme	2 days	Wed 19/2/20	Thu 20/2/20
1.7.2.2.	Review on GerDoc	2 days	Fri 21/2/20	Sat 22/2/20
1.7.2.2.	Review on DoctorOnCall	2 days	Sun 23/2/20	Mon 24/2/20
1.7.2.2.	Compare all of the systems developed by Malaysian	3 days	Tue 25/2/20	Thu 27/2/20
1.7.2.3	Identify the common and recommended features	8 days	Sun 1/3/20	Sun 8/3/20

Figure 3.5 Schedule Overview of Requirement Gathering and Elicitation



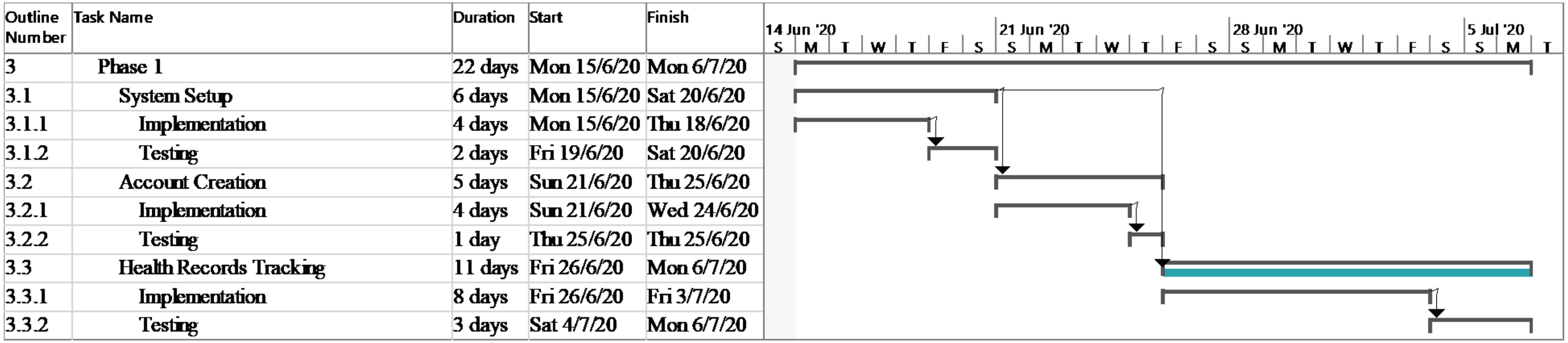
Outline Number	Task Name	Duration	Start	Finish
1.9	Project Scheduling	10 days	Thu 12/3/20	Sat 21/3/20
1.9.1	Generate WBS	4 days	Thu 12/3/20	Sun 15/3/20
1.9.1.1	Identify main features	2 days	Thu 12/3/20	Fri 13/3/20
1.9.1.2	Breakdown features	4 days	Thu 12/3/20	Sun 15/3/20
1.9.2	Generate Gantt Chart	6 days	Mon 16/3/20	Sat 21/3/20
1.9.2.1	Determine Task Dependency	2 days	Mon 16/3/20	Tue 17/3/20
1.9.2.2	Estimate Effort	1 day	Wed 18/3/20	Wed 18/3/20
1.9.2.3	Estimate Start Time and End Time	1 day	Thu 19/3/20	Thu 19/3/20
1.9.2.4	Create Gantt Chart	2 days	Fri 20/3/20	Sat 21/3/20

Figure 3.6 Schedule Overview of Project Scheduling



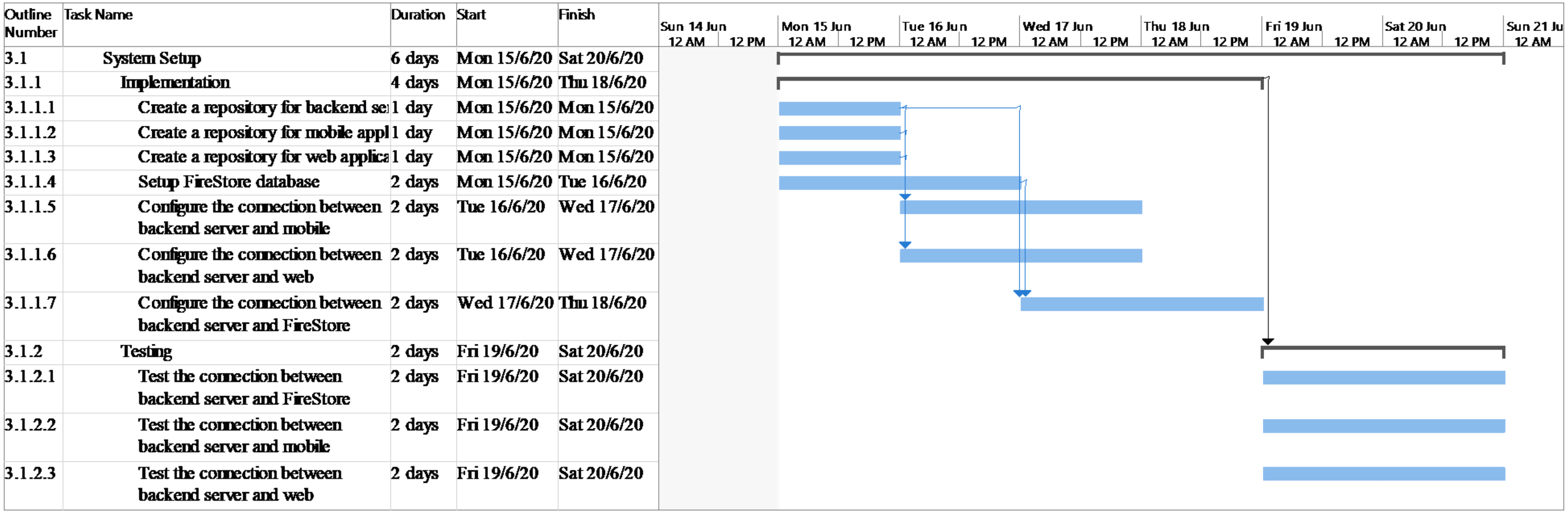
Outline Number	Task Name	Duration	Start	Finish
2	Analysis & Design	26 days	Sun 22/3/20	Thu 16/4/20
2.1	Design Use Case Diagram	5 days	Sun 22/3/20	Thu 26/3/20
2.2	Prepare Use Case Description	4 days	Fri 27/3/20	Mon 30/3/20
2.3	Design Data Model Diagram	6 days	Sun 22/3/20	Fri 27/3/20
2.4	Design Data Flow Diagram	6 days	Sun 22/3/20	Fri 27/3/20
2.5	Low Level Prototyping	8 days	Sun 22/3/20	Sun 29/3/20
2.5.1	Prepare wireframe for mobile applicat	4 days	Sun 22/3/20	Wed 25/3/20
2.5.2	Prepare wireframe for web applicatio	4 days	Thu 26/3/20	Sun 29/3/20
2.6	High Level Prototyping	18 days	Mon 30/3/20	Thu 16/4/20
2.6.1	Prepare design prototype for mobile application	9 days	Mon 30/3/20	Tue 7/4/20
2.6.2	Prepare design prototype for web app	9 days	Wed 8/4/20	Thu 16/4/20

Figure 3.7 Schedule Overview of Analysis and Design Phase



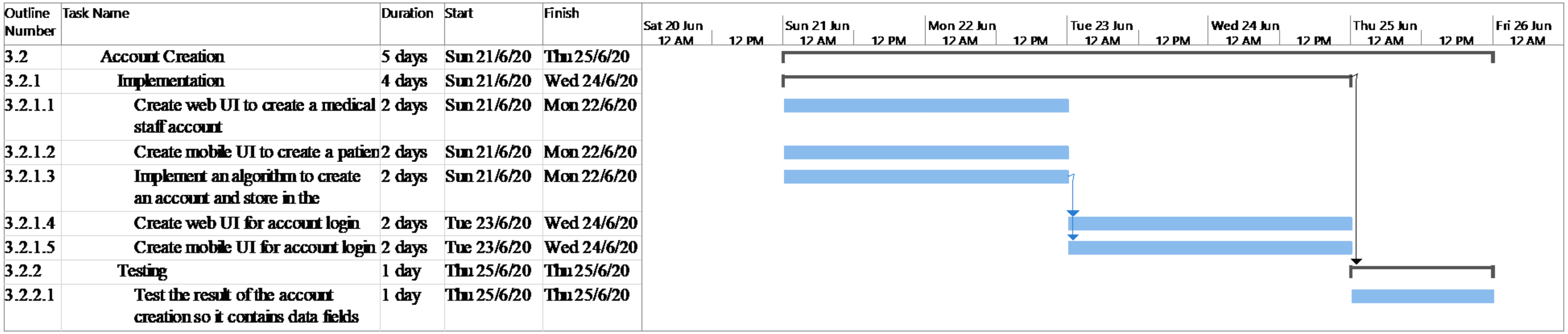
Outline Number	Task Name	Duration	Start	Finish
3	Phase 1	22 days	Mon 15/6/20	Mon 6/7/20
3.1	System Setup	6 days	Mon 15/6/20	Sat 20/6/20
3.1.1	Implementation	4 days	Mon 15/6/20	Thu 18/6/20
3.1.2	Testing	2 days	Fri 19/6/20	Sat 20/6/20
3.2	Account Creation	5 days	Sun 21/6/20	Thu 25/6/20
3.2.1	Implementation	4 days	Sun 21/6/20	Wed 24/6/20
3.2.2	Testing	1 day	Thu 25/6/20	Thu 25/6/20
3.3	Health Records Tracking	11 days	Fri 26/6/20	Mon 6/7/20
3.3.1	Implementation	8 days	Fri 26/6/20	Fri 3/7/20
3.3.2	Testing	3 days	Sat 4/7/20	Mon 6/7/20

Figure 3.8 Schedule Overview of Implementation Phase 1



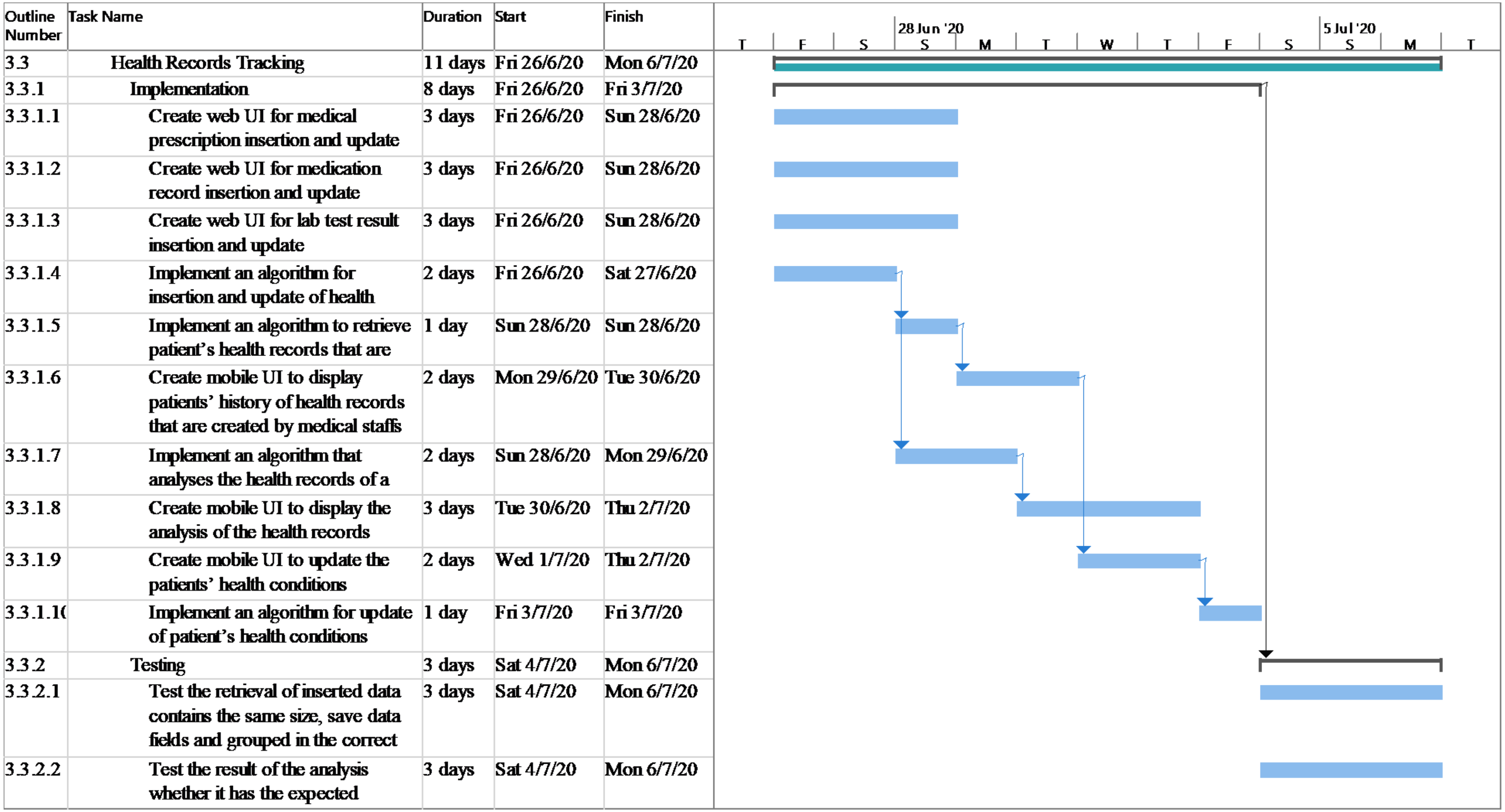
Outline Number	Task Name	Duration	Start	Finish
3.1	System Setup	6 days	Mon 15/6/20	Sat 20/6/20
3.1.1	Implementation	4 days	Mon 15/6/20	Thu 18/6/20
3.1.1.1	Create a repository for backend se	1 day	Mon 15/6/20	Mon 15/6/20
3.1.1.2	Create a repository for mobile appl	1 day	Mon 15/6/20	Mon 15/6/20
3.1.1.3	Create a repository for web applica	1 day	Mon 15/6/20	Mon 15/6/20
3.1.1.4	Setup FireStore database	2 days	Mon 15/6/20	Tue 16/6/20
3.1.1.5	Configure the connection between backend server and mobile	2 days	Tue 16/6/20	Wed 17/6/20
3.1.1.6	Configure the connection between backend server and web	2 days	Tue 16/6/20	Wed 17/6/20
3.1.1.7	Configure the connection between backend server and FireStore	2 days	Wed 17/6/20	Thu 18/6/20
3.1.2	Testing	2 days	Fri 19/6/20	Sat 20/6/20
3.1.2.1	Test the connection between backend server and FireStore	2 days	Fri 19/6/20	Sat 20/6/20
3.1.2.2	Test the connection between backend server and mobile	2 days	Fri 19/6/20	Sat 20/6/20
3.1.2.3	Test the connection between backend server and web	2 days	Fri 19/6/20	Sat 20/6/20

Figure 3.9 Schedule Overview of System Setup Activities



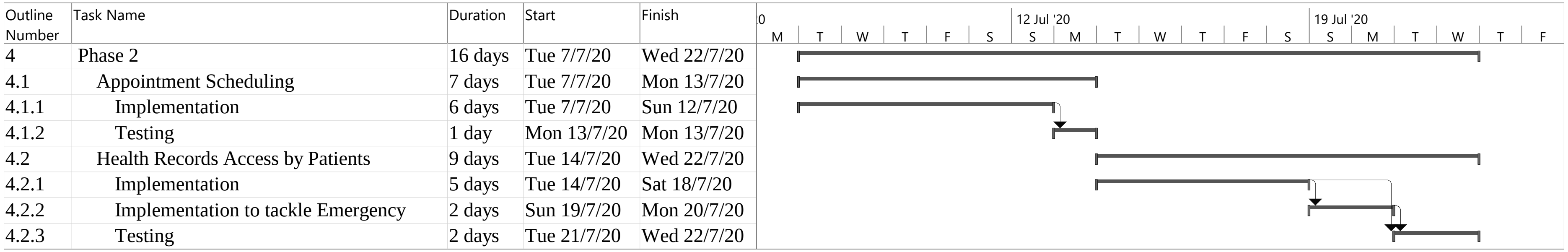
Outline Number	Task Name	Duration	Start	Finish
3.2	Account Creation	5 days	Sun 21/6/20	Thu 25/6/20
3.2.1	Implementation	4 days	Sun 21/6/20	Wed 24/6/20
3.2.1.1	Create web UI to create a medical staff account	2 days	Sun 21/6/20	Mon 22/6/20
3.2.1.2	Create mobile UI to create a patien	2 days	Sun 21/6/20	Mon 22/6/20
3.2.1.3	Implement an algorithm to create an account and store in the	2 days	Sun 21/6/20	Mon 22/6/20
3.2.1.4	Create web UI for account login	2 days	Tue 23/6/20	Wed 24/6/20
3.2.1.5	Create mobile UI for account login	2 days	Tue 23/6/20	Wed 24/6/20
3.2.2	Testing	1 day	Thu 25/6/20	Thu 25/6/20
3.2.2.1	Test the result of the account creation so it contains data fields	1 day	Thu 25/6/20	Thu 25/6/20

Figure 3.10 Schedule Overview of Account Creation Feature



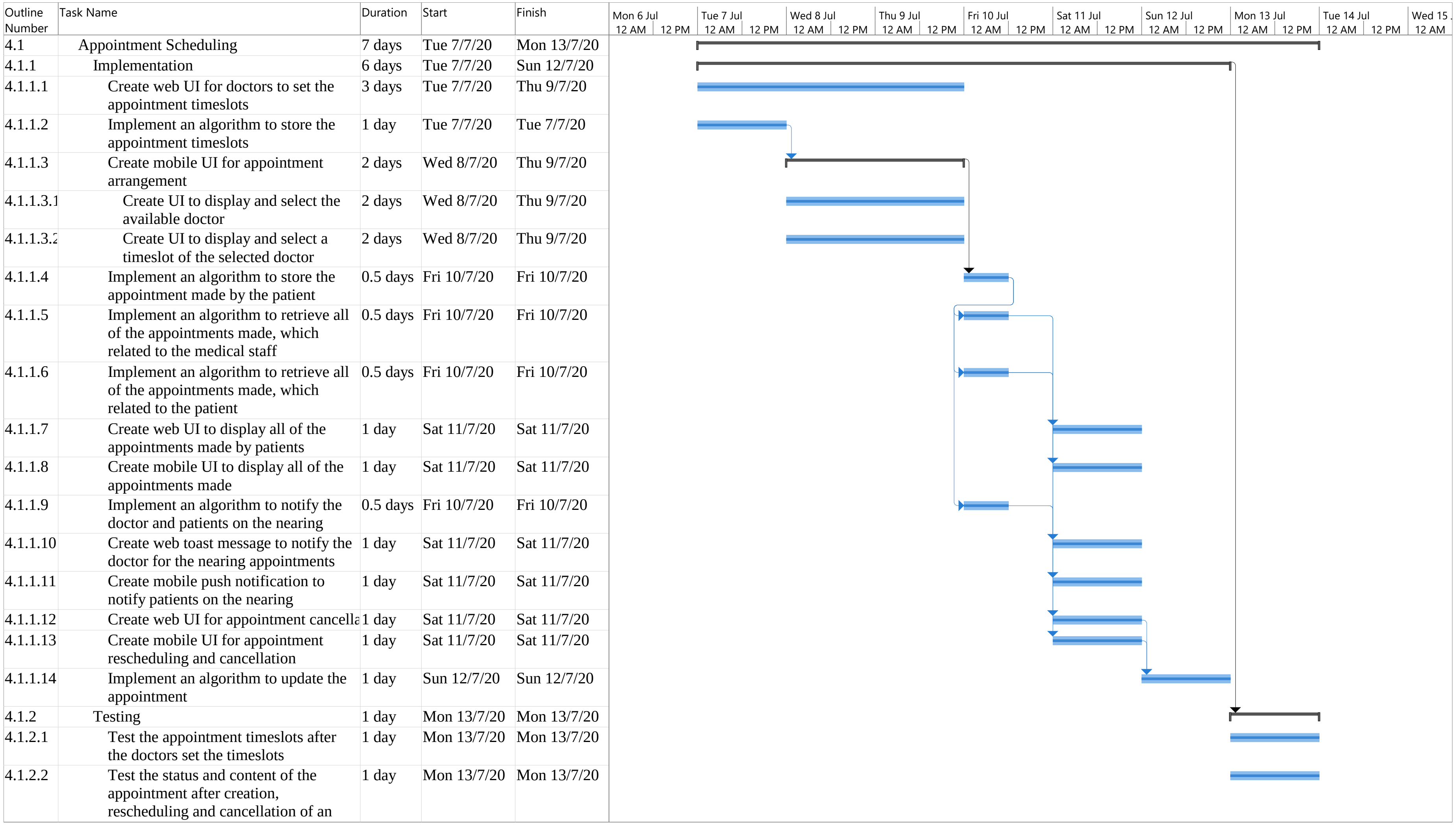
Outline Number	Task Name	Duration	Start	Finish
3.3	Health Records Tracking	11 days	Fri 26/6/20	Mon 6/7/20
3.3.1	Implementation	8 days	Fri 26/6/20	Fri 3/7/20
3.3.1.1	Create web UI for medical prescription insertion and update	3 days	Fri 26/6/20	Sun 28/6/20
3.3.1.2	Create web UI for medication record insertion and update	3 days	Fri 26/6/20	Sun 28/6/20
3.3.1.3	Create web UI for lab test result insertion and update	3 days	Fri 26/6/20	Sun 28/6/20
3.3.1.4	Implement an algorithm for insertion and update of health	2 days	Fri 26/6/20	Sat 27/6/20
3.3.1.5	Implement an algorithm to retrieve patient's health records that are	1 day	Sun 28/6/20	Sun 28/6/20
3.3.1.6	Create mobile UI to display patients' history of health records that are created by medical staffs	2 days	Mon 29/6/20	Tue 30/6/20
3.3.1.7	Implement an algorithm that analyses the health records of a	2 days	Sun 28/6/20	Mon 29/6/20
3.3.1.8	Create mobile UI to display the analysis of the health records	3 days	Tue 30/6/20	Thu 2/7/20
3.3.1.9	Create mobile UI to update the patients' health conditions	2 days	Wed 1/7/20	Thu 2/7/20
3.3.1.1(	Implement an algorithm for update of patient's health conditions	1 day	Fri 3/7/20	Fri 3/7/20
3.3.2	Testing	3 days	Sat 4/7/20	Mon 6/7/20
3.3.2.1	Test the retrieval of inserted data contains the same size, save data fields and grouped in the correct	3 days	Sat 4/7/20	Mon 6/7/20
3.3.2.2	Test the result of the analysis whether it has the expected	3 days	Sat 4/7/20	Mon 6/7/20

Figure 3.11 Schedule Overview of Health Records Tracking Feature



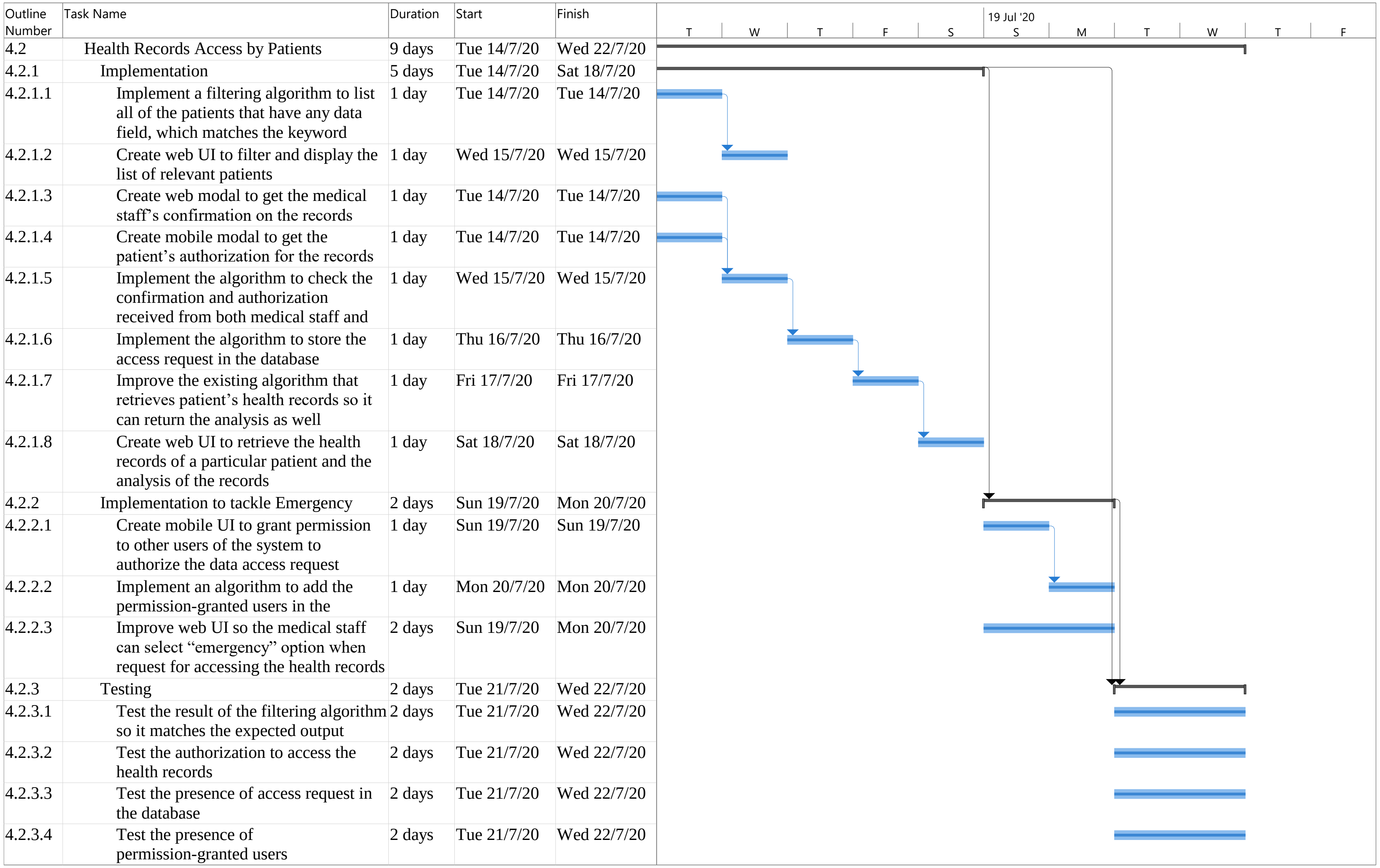
Outline Number	Task Name	Duration	Start	Finish
4	Phase 2	16 days	Tue 7/7/20	Wed 22/7/20
4.1	Appointment Scheduling	7 days	Tue 7/7/20	Mon 13/7/20
4.1.1	Implementation	6 days	Tue 7/7/20	Sun 12/7/20
4.1.2	Testing	1 day	Mon 13/7/20	Mon 13/7/20
4.2	Health Records Access by Patients	9 days	Tue 14/7/20	Wed 22/7/20
4.2.1	Implementation	5 days	Tue 14/7/20	Sat 18/7/20
4.2.2	Implementation to tackle Emergency	2 days	Sun 19/7/20	Mon 20/7/20
4.2.3	Testing	2 days	Tue 21/7/20	Wed 22/7/20

Figure 3.12 Schedule Overview of Implementation Phase 2



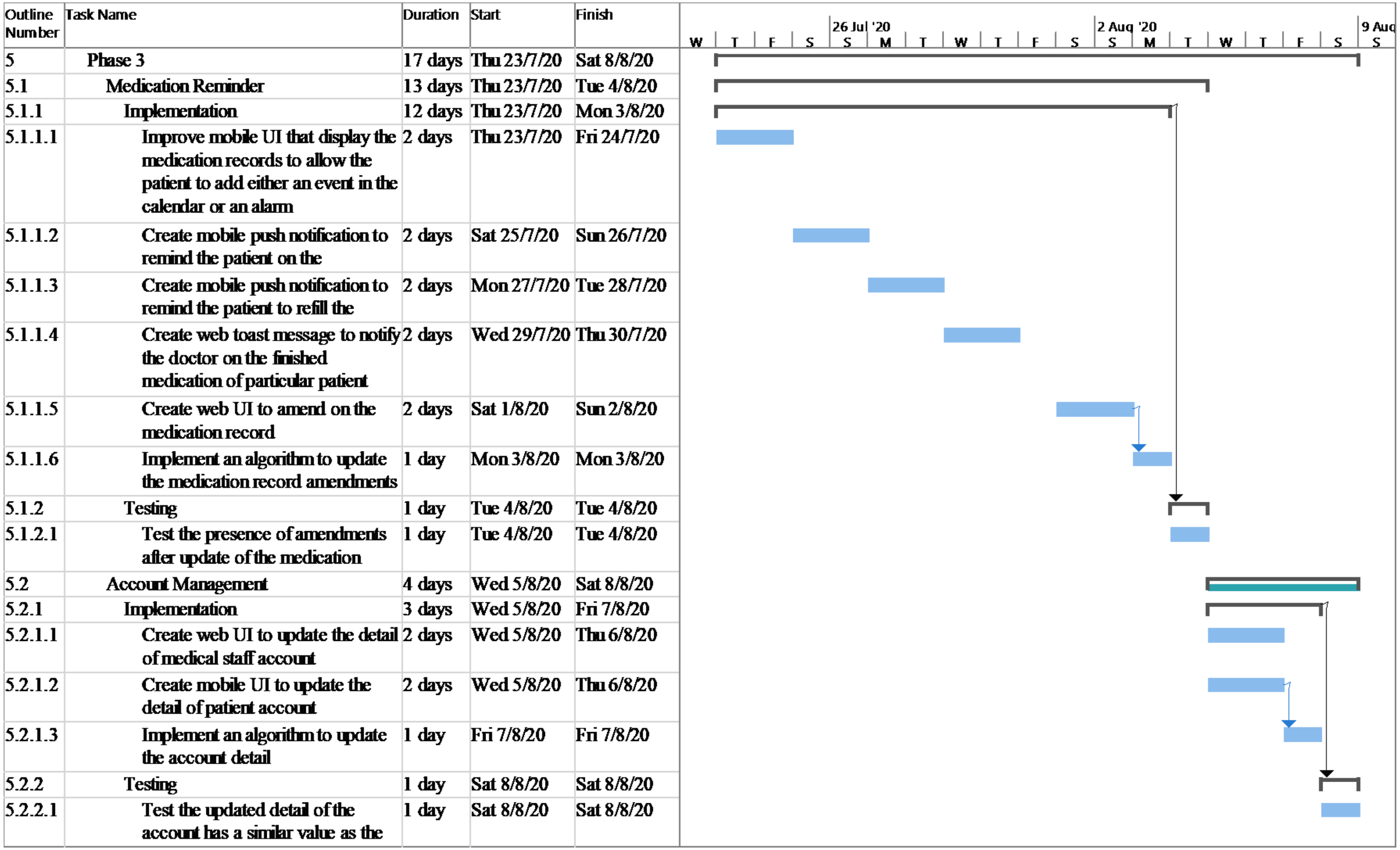
Outline Number	Task Name	Duration	Start	Finish
4.1	Appointment Scheduling	7 days	Tue 7/7/20	Mon 13/7/20
4.1.1	Implementation	6 days	Tue 7/7/20	Sun 12/7/20
4.1.1.1	Create web UI for doctors to set the appointment timeslots	3 days	Tue 7/7/20	Thu 9/7/20
4.1.1.2	Implement an algorithm to store the appointment timeslots	1 day	Tue 7/7/20	Tue 7/7/20
4.1.1.3	Create mobile UI for appointment arrangement	2 days	Wed 8/7/20	Thu 9/7/20
4.1.1.3.1	Create UI to display and select the available doctor	2 days	Wed 8/7/20	Thu 9/7/20
4.1.1.3.2	Create UI to display and select a timeslot of the selected doctor	2 days	Wed 8/7/20	Thu 9/7/20
4.1.1.4	Implement an algorithm to store the appointment made by the patient	0.5 days	Fri 10/7/20	Fri 10/7/20
4.1.1.5	Implement an algorithm to retrieve all of the appointments made, which related to the medical staff	0.5 days	Fri 10/7/20	Fri 10/7/20
4.1.1.6	Implement an algorithm to retrieve all of the appointments made, which related to the patient	0.5 days	Fri 10/7/20	Fri 10/7/20
4.1.1.7	Create web UI to display all of the appointments made by patients	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.8	Create mobile UI to display all of the appointments made	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.9	Implement an algorithm to notify the doctor and patients on the nearing	0.5 days	Fri 10/7/20	Fri 10/7/20
4.1.1.10	Create web toast message to notify the doctor for the nearing appointments	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.11	Create mobile push notification to notify patients on the nearing	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.12	Create web UI for appointment cancella	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.13	Create mobile UI for appointment rescheduling and cancellation	1 day	Sat 11/7/20	Sat 11/7/20
4.1.1.14	Implement an algorithm to update the appointment	1 day	Sun 12/7/20	Sun 12/7/20
4.1.2	Testing	1 day	Mon 13/7/20	Mon 13/7/20
4.1.2.1	Test the appointment timeslots after the doctors set the timeslots	1 day	Mon 13/7/20	Mon 13/7/20
4.1.2.2	Test the status and content of the appointment after creation, rescheduling and cancellation of an	1 day	Mon 13/7/20	Mon 13/7/20

Figure 3.13 Schedule Overview of Appointment Scheduling Feature



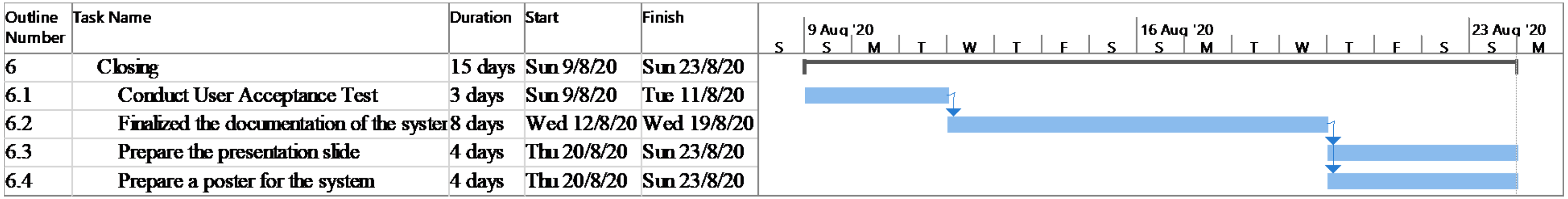
Outline Number	Task Name	Duration	Start	Finish
4.2	Health Records Access by Patients	9 days	Tue 14/7/20	Wed 22/7/20
4.2.1	Implementation	5 days	Tue 14/7/20	Sat 18/7/20
4.2.1.1	Implement a filtering algorithm to list all of the patients that have any data field, which matches the keyword	1 day	Tue 14/7/20	Tue 14/7/20
4.2.1.2	Create web UI to filter and display the list of relevant patients	1 day	Wed 15/7/20	Wed 15/7/20
4.2.1.3	Create web modal to get the medical staff's confirmation on the records	1 day	Tue 14/7/20	Tue 14/7/20
4.2.1.4	Create mobile modal to get the patient's authorization for the records	1 day	Tue 14/7/20	Tue 14/7/20
4.2.1.5	Implement the algorithm to check the confirmation and authorization received from both medical staff and	1 day	Wed 15/7/20	Wed 15/7/20
4.2.1.6	Implement the algorithm to store the access request in the database	1 day	Thu 16/7/20	Thu 16/7/20
4.2.1.7	Improve the existing algorithm that retrieves patient's health records so it can return the analysis as well	1 day	Fri 17/7/20	Fri 17/7/20
4.2.1.8	Create web UI to retrieve the health records of a particular patient and the analysis of the records	1 day	Sat 18/7/20	Sat 18/7/20
4.2.2	Implementation to tackle Emergency	2 days	Sun 19/7/20	Mon 20/7/20
4.2.2.1	Create mobile UI to grant permission to other users of the system to authorize the data access request	1 day	Sun 19/7/20	Sun 19/7/20
4.2.2.2	Implement an algorithm to add the permission-granted users in the	1 day	Mon 20/7/20	Mon 20/7/20
4.2.2.3	Improve web UI so the medical staff can select "emergency" option when request for accessing the health records	2 days	Sun 19/7/20	Mon 20/7/20
4.2.3	Testing	2 days	Tue 21/7/20	Wed 22/7/20
4.2.3.1	Test the result of the filtering algorithm so it matches the expected output	2 days	Tue 21/7/20	Wed 22/7/20
4.2.3.2	Test the authorization to access the health records	2 days	Tue 21/7/20	Wed 22/7/20
4.2.3.3	Test the presence of access request in the database	2 days	Tue 21/7/20	Wed 22/7/20
4.2.3.4	Test the presence of permission-granted users	2 days	Tue 21/7/20	Wed 22/7/20

Figure 3.14 Schedule Overview of Health Record Access by Patients Feature



Outline Number	Task Name	Duration	Start	Finish
5	Phase 3	17 days	Thu 23/7/20	Sat 8/8/20
5.1	Medication Reminder	13 days	Thu 23/7/20	Tue 4/8/20
5.1.1	Implementation	12 days	Thu 23/7/20	Mon 3/8/20
5.1.1.1	Improve mobile UI that display the medication records to allow the patient to add either an event in the calendar or an alarm	2 days	Thu 23/7/20	Fri 24/7/20
5.1.1.2	Create mobile push notification to remind the patient on the	2 days	Sat 25/7/20	Sun 26/7/20
5.1.1.3	Create mobile push notification to remind the patient to refill the	2 days	Mon 27/7/20	Tue 28/7/20
5.1.1.4	Create web toast message to notify the doctor on the finished medication of particular patient	2 days	Wed 29/7/20	Thu 30/7/20
5.1.1.5	Create web UI to amend on the medication record	2 days	Sat 1/8/20	Sun 2/8/20
5.1.1.6	Implement an algorithm to update the medication record amendments	1 day	Mon 3/8/20	Mon 3/8/20
5.1.2	Testing	1 day	Tue 4/8/20	Tue 4/8/20
5.1.2.1	Test the presence of amendments after update of the medication	1 day	Tue 4/8/20	Tue 4/8/20
5.2	Account Management	4 days	Wed 5/8/20	Sat 8/8/20
5.2.1	Implementation	3 days	Wed 5/8/20	Fri 7/8/20
5.2.1.1	Create web UI to update the detail of medical staff account	2 days	Wed 5/8/20	Thu 6/8/20
5.2.1.2	Create mobile UI to update the detail of patient account	2 days	Wed 5/8/20	Thu 6/8/20
5.2.1.3	Implement an algorithm to update the account detail	1 day	Fri 7/8/20	Fri 7/8/20
5.2.2	Testing	1 day	Sat 8/8/20	Sat 8/8/20
5.2.2.1	Test the updated detail of the account has a similar value as the	1 day	Sat 8/8/20	Sat 8/8/20

Figure 3.15 Schedule Overview of Implementation Phase 3



Outline Number	Task Name	Duration	Start	Finish
6	Closing	15 days	Sun 9/8/20	Sun 23/8/20
6.1	Conduct User Acceptance Test	3 days	Sun 9/8/20	Tue 11/8/20
6.2	Finalized the documentation of the syster	8 days	Wed 12/8/20	Wed 19/8/20
6.3	Prepare the presentation slide	4 days	Thu 20/8/20	Sun 23/8/20
6.4	Prepare a poster for the system	4 days	Thu 20/8/20	Sun 23/8/20

Figure 3.16 Schedule Overview of Project Closing

3.5. Summary

This project develops a health records tracking system. In total, four main stages involved, which are planning phase, analysis phase, development and testing phase as well as the closing phase. The development and testing phase involve three iterations. The entire project includes 122 subtasks, which is aimed to complete within 172 days excluding 45 days of holiday. The estimated completion date is Sunday, 23 August 2020.

CHAPTER 4

PROJECT SPECIFICATION

4.1. Introduction

This chapter examines the requirements gathered and determines the specification of the project. With the initial specification, use case diagrams were prepared to visualize functionalities of the system.

4.2. Requirements Specification

The requirements specification section discusses the functional requirements and the non-functional requirement of the system. The functional requirements of the system are separated into two groups, which represent the mobile application and web application respectively. The non-functional requirements of this system include availability requirements, development requirements, operational requirements, performance requirements, security requirements and usability requirements. On top of that, the user of the mobile application is the patient while the user of the web application is the medical staff.

4.2.1. Functional Requirement

The functional requirements of this project are separated into two parts, which are functional requirements of the backend server and the functional requirements of the frontend applications. The frontend applications include the applications in the mobile platform and web platform. In the functional requirements of the frontend application, the term "system" is used to indicate both of the frontend applications. On the other hand, the functional requirements must be implemented in both of the frontend applications.

4.2.1.1. Backend Server

1. The backend server must be able to record the health records created or updated by the medical staff.
2. The backend server must check the authorization and confirmation from both patient and medical staff respectively before sending the health records and the analysis to the web application.

3. The backend server must record every access request in the database.

4.2.1.2. Frontend Application

4.2.1.2.1. Account Management Module

1. The system must authenticate the patient.
2. The system must allow the user to update the detail of their account.

4.2.1.2.2. Health Records Tracking Module

3. The web application must allow the medical staff to create and update the health records.
4. The system must display the patient's history of health records and the analysis of the health records.
5. The web application must allow the medical staff to select the type of health records access.
6. The mobile application must allow the patient to update their health conditions, such as blood sugar level, blood pressure level and BMI.

4.2.1.2.3. Appointment Scheduling Module

7. The web application must allow the medical staff to set their available timeslot for the appointment.
8. The mobile application must allow the patient to arrange an appointment with a doctor.
9. The system must notify the user for every new or nearing appointment.
10. The system must allow the user to reschedule or cancel an appointment.

4.2.1.2.4. Health Records Access by Patients Module

11. The mobile application must allow the patient to permit the medical staff to access their health records.
12. The mobile application must allow the patient to grant access authorization permission to other users of the system, so they can permit the medical staff to access the health records of the patient during an emergency.

4.2.1.2.5. Medication Reminder Module

13. The system must remind the user for the intake and refill of the medication.
14. The web application must allow the medical staff to update the medication records for extra medication.

4.2.2. Non-Functional Requirement

4.2.2.1. Adaptability Requirements

1. The mobile application shall have a design that is responsive to mobile screens with different resolutions.
2. The web application shall have a design that is responsive to different views, such as the desktop view and mobile view.

4.2.2.2. Availability Requirements

1. The application in both web and mobile platforms shall be accessible regardless of the time and venue of the users with the condition that they can access to the Internet.

4.2.2.3. Development Requirements

1. The system shall be built in Android platform and web platform.
2. Code editor used for development would be Visual Studio Code.
3. The methodology used in development is Phased Development Methodology.
4. The programming language used in development is JavaScript.
5. Each data record is stored in the Firebase database.

4.2.2.4. Performance Requirements

1. The system must handle 99% of the exception throw during the runtime of the application and display respective error messages.
2. The system shall process a request and return the result within 3 seconds.
3. The system shall handle multiple concurrent requests without crashing.

4.2.2.5. Security Requirements

1. The system must allow authorized users to access the system only.
2. The system must allow only users who request data from backend if Firebase token is provided.

3. The system must get authorization and confirmation from both patient and medical staff respectively before displaying the records to the medical staff.
4. The system must record every health records accesses by the medical staff.

4.2.2.6. Usability Requirements

1. The UI of the mobile application shall be intuitive and attractive to the patients.
2. The UI of the web application shall be simple for the medical staff to perform their daily tasks.
3. The system shall provide a guidance feature for the new users.
4. The system shall display an error message if the user input is incorrect or invalid.

4.3. Use Cases

This section demonstrates the activities that can be performed by the system users. In total, twelve use cases were identified and described in this section. Each use case description consists of the primary actor of the use case, the relationship with other use cases, the event flow as well as the exceptional flow of the event.

4.3.1. Use Case Diagram

In this project, it consists of two actors, which are the patients and the medical staff. The patient would use the mobile application to keep track of their health records while the medical staff can use the web application in helping them to manage the patients' health records as well as their appointments. Additionally, the use cases of this project include user authentication, health records tracking, data access authorization, appointment scheduling and management, medication reminder as well as the account management.

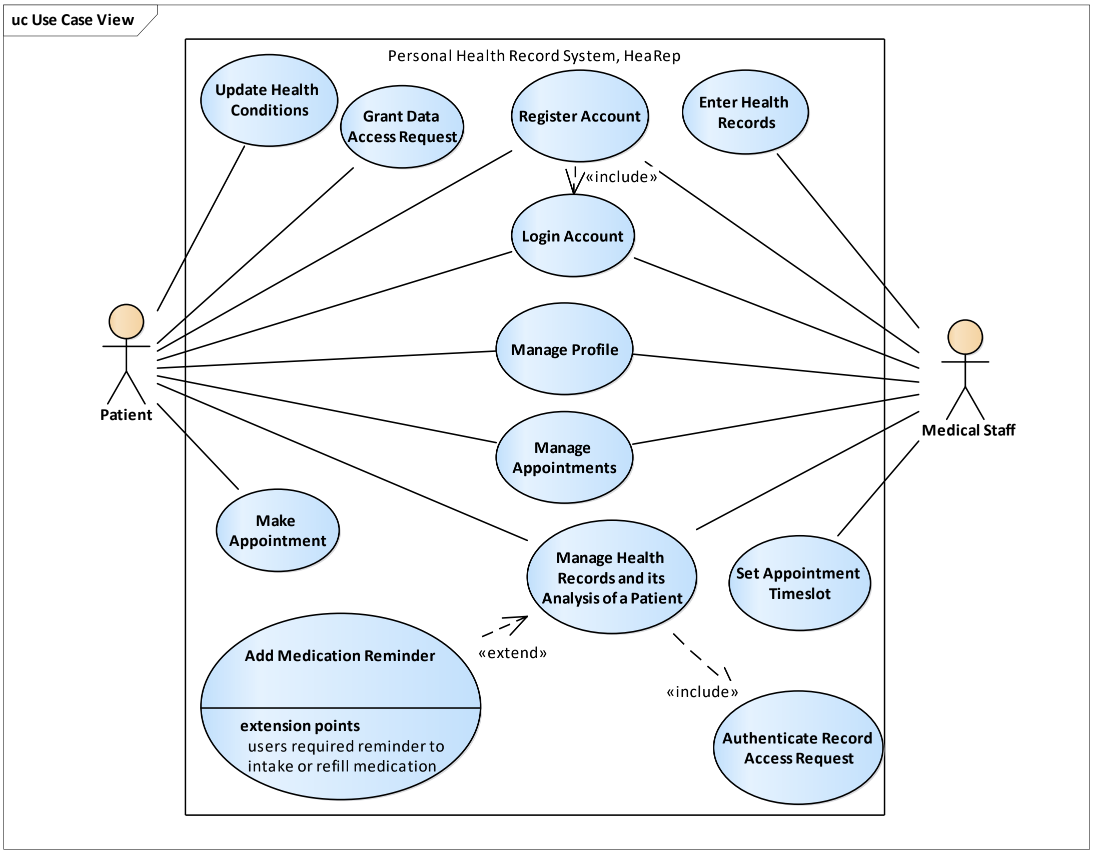


Figure 4.1 Use Case Diagram

4.3.2. Use Case Description

Table 4-1 Use Case of Registering Account

<table>
<tr><td><b>Name</b>: Register Account</td><td><b>ID</b>: 1</td><td><b>Priority</b>: High</td></tr>
<tr><td><b>Actor</b>: Patient, Medical Staff</td><td colspan="2"><b>Type</b>: Detail, Real</td></tr>
<tr><td colspan="3"><b>Stakeholder’s Information</b>:
Patient → person who is new to the mobile application and wants to keep track of the health records via the mobile application
Medical Staff → person who is new to the web application and wants to manage health records via the web application</td></tr>
<tr><td colspan="3"><b>Summary of Use Case</b>:
This use case outlines the steps needed for the user to create an account in either web or mobile application.</td></tr>
<tr><td colspan="3"><b>Triggering Situation</b>:
The patients or medical staff wants to access the system but has no account.</td></tr>
<tr><td colspan="3"><b>Relationship</b>:
• Association: Patient, Medical staff
• Include: Login Account
• Extend: -</td></tr>
<tr><td colspan="3"><b>Normal Event Flow</b>:
<u>Patient</u>
1) The patient clicks on the register button in the mobile application.
2) The patient enters the phone number.
3) The system sends an OTP code to the phone number.
4) The user enters the code for verification. If OTP code expired, perform sub-flow 2.1.
5) The patient enters the full name, gender, age address and email address.
6) Upon success account creation, the mobile application displays a success message and proceed with login (Use Case with ID 2).

<u>Medical Staff</u>
1) The medical staff needs to click on the register button in the web application.
2) The medical staff enter the full name, age, gender, address, email address and the name of the medical institution along with the role in the institution.
3) The system will send a verification email to the medical staff.
4) The web application displays a success message and then proceeds with login (Use Case with ID 2).</td></tr>
<tr><td colspan="3"><b>Sub Event Flow</b>:
<u>Patient</u>
1.1. The system resends a new OTP code to the phone number.</td></tr>
</table>

Optional Flow:

Patient

2.1) If the phone number input has the wrong format, the system displays an error message. Then, the patient required to fill in again.

2.2) If the phone number is used, the system displays an error message. Then, the patient required to use another phone number.

5.1) If the input fields do not follow the input format, the system displays an error message. Then, the user required to fill in again.

5.2) If empty fields present, the system displays an error message. Then, the patient required to fill in the empty fields.

Medical Staff

2.1) If the email address is used, the system displays an error message. Then, the medical staff required to use another email address.

2.2) If empty fields present, the system displays an error message. Then, the medical staff required to fill in the empty fields.

Table 4-2 Use Case of Login Account

Name: Login Account	**ID**: 2	**Priority**: High
Actor: Patient, Medical Staff	**Type**: Detail, Real	
Stakeholder's Information: Patient → person who wants to keep track of the health records via mobile application and has an account Medical Staff → person who wants to manage health records via web application and has an account		
Summary of Use Case: This use case describes the actions needed for the user to log into the system.		
Triggering Situation: The patient or medical staff wants to access the system and has an account.		
Relationship: • Association: Patient, Medical Staff • Include: - • Extend: -		
Normal Event Flow: Patient 1) The patient enters the phone number. 2) The system sends the OTP code to the phone number of the patient. 3) The patient enters the OTP code. If OTP code expired, perform sub-flow 3.1. 4) The system validates the login information. 5) Upon success login, the mobile application displays the home interfaces. Medical Staff 1) The medical staff enters the full name or the email along with the password. If medical staff forgets the password, perform sub-flow 1.1. 2) The system validates the login information. 3) Upon success login, the web application displays the dashboard interface.		
Sub Event Flow: Patient 4.1. The system resends a new OTP code. Medical Staff 1.1. The medical staff click the "forget password" option and the system send a password reset email to the mailbox.		

Optional Flow:

Patient

1.1) If the phone number input has the wrong format, the system displays an error message. Then, the patient required to fill in the number again.

4.1) If the OTP code entered is incorrect, the system displays an error message. Then, the patient required to fill in again.

Medical Staff

2.1) If the login information does not match, the system displays an error message. Then, the medical staff required to reenter.

Table 4-3 Use Case of Entering Health Records

Name: Enter Health Records	**ID**: 3	**Priority**: High
Actor: Medical Staff	**Type**: Detail, Real	
Stakeholder’s Information: Medical Staff → person who meets the patient and records the patient's diagnosis		
Summary of Use Case: This use case outlines the steps needed for the medical staff to create a health record.		
Triggering Situation: 1) During the consultation with the patient, the medical staff records the patient's diagnosis as well as his/her clinical opinions. 2) The medical staff prepared treatment of medication to the patients. 3) After the lab test result is released, the medical staff required to show the result to the patients.		
Relationship: • Association: Medical Staff • Include: - • Extend: -		
Normal Event Flow: 1) The medical staff select the patient by searching and filtering the patient with full name or phone number. 2) The medical staff clicks on the “Add Health Record” button. 3) The medical staff selects the type of health record. 4) The medical staff enters the information on the health record. 5) Upon success storing of the health record, the web application displays a success message.		
Sub Event Flow: -		
Optional Flow: -		

Table 4-4 Use Case of Managing Health Records and its Analysis of a Patient

Name: Manage Health Records and its Analysis of a Patient	**ID**: 4	**Priority**: High
Actor: Patient, Medical Staff	**Type**: Detail, Real	
Stakeholder's Information: Patient → person who want to view their health records as well as the analysis Medical Staff → person who want to view the patient health records as well as his/her analysis		
Summary of Use Case: This use case lays out the actions required for the user to view the health records and analysis.		
Triggering Situation: 1) The patient or medical staff wishes to view the health records. 2) The patient or medical staff wishes to know the analysis of the health records.		
Relationship: • Association: Patient, Medical Staff • Include: Authorize Data Access Request • Extend: Add Medication Reminder		
Normal Event Flow: Patient 1) The patient clicks on the "Health Records" tab in the mobile application. 2) The mobile application displays all of the health records of the patient as well as the analysis of the health records. 3) If the patient always forgot to take or refill medication, the patient can perform sub-flow 3a.1. Medical Staff 1) The medical staff select the patient by searching and filtering the patient with full name or phone number. 2) The web application displays the information of the patient. 3) The medical staff clicks on the "View Health Records" button. To get the patient's authorization, the system performs sub-flow 3b.1. 4) The web application displays all of the health records of the patient as well as the analysis of the health records. 5) To update the health records, the medical staff performs sub-flow 5.1.		

Sub Event Flow:

Patient

3a.1. The patient adds a medication reminder on the mobile phone.

3a.1.1. The patient selects the medication record that requires a reminder.

3a.1.2. The patient selects the type of reminder.

3a.1.3. Upon success addition of the reminder, the mobile application displays a success message.

Medical Staff

3b.1. The system requests authorization from the patient.

3b.1.1. The mobile application displays a modal to get the patient's authorization.

3b.1.2. The patient authorizes the access request by the medical staff to view his/her health records and analysis.

3b.1.3. The mobile application displays a success message.

5.1. The medical staff update a health record.

5.1.1. The medical staff selects the health record to update.

5.1.2. The medical staff enters the latest information.

5.1.3. The system stores the latest changes in the database.

5.1.4. The web application displays a success message.

Optional Flow:

Medical Staff

4.1) If the patient does not authorize the data access request, the web application displays an error message.

Table 4-5 Use Case of Updating Health Conditions

Name: Update Health Conditions	**ID**: 5	**Priority**: High
Actor: Patient	**Type**: Detail, Real	
Stakeholder’s Information: Patient → person who wants to update the health conditions for health monitoring		
Summary of Use Case: This use case lays out the steps needed for the user to update the health conditions.		
Triggering Situation: The patient wishes to keep track of their health condition so it can help the medical staff to conduct a better diagnosis.		
Relationship: • Association: Patient • Include: - • Extend: -		
Normal Event Flow: 1) The patient clicks on the “Update Health Conditions” button. 2) The patient selects one type of conditions and enters its value. 3) If more conditions to be updated, the patient clicks on the plus icon and repeat step 1 and step 2. 4) The system stores the health conditions and the mobile application displays a success message.		
Sub Event Flow: -		
Optional Flow: -		

Table 4-6 Use Case of Setting Appointment Timeslot

Name: Set Appointment Timeslot	**ID**: 6	**Priority**: Medium
Actor: Medical Staff	**Type**: Detail, Real	
Stakeholder’s Information: Medical Staff → person who allows the patient to book for a consultation		
Summary of Use Case: This use case describes the steps needed to set the available timeslot for an appointment.		
Triggering Situation: The medical staff is available to give consultation to the patient.		
Relationship: • Association: Medical Staff • Include: - • Extend: -		
Normal Event Flow: 1) The medical staff selects the “Appointment” tab in the web application. 2) The medical staff selects the “Set available Timeslot” button. 3) The medical staff selects the effective period, the available days in a week and available time in each day. To change another type of appointment, the medical staff performs sub-flow 3.1. 4) Upon success update, the web application displays a success message.		
Sub Event Flow: 3.1. The medical staff change the type of appointment from appointment by date to appointment by number. 3.1.1. The medical staff toggle the type of appointment. 3.1.2. The medical staff set the available time in a day and the expected consultation time for each patient.		
Optional Flow: -		

Table 4-7 Use Case of Making Appointment

Name: Make Appointment	**ID**: 7	**Priority**: Medium
Actor: Patient	**Type**: Detail, Real	
Stakeholder's Information: Patient → person who seek medical consultation and reluctant to wait for a long time in the medical institution		
Summary of Use Case: This use case describes the actions needed for the patient to make an appointment.		
Triggering Situation: 1) The patient needs to meet a specialist, who works at multiple institutions, for chronic diseases. 2) The patient needs to meet a doctor for an illness diagnosis.		
Relationship: • Association: Patient • Include: - • Extend: -		
Normal Event Flow: 1) The patient selects the "Appointment" tab in the mobile application. 2) The patient clicks on the "Make a New Appointment" button. 3) The patient selects the medical staff by searching and filtering the medical staff with fill name or medical institution. 4) The patient picks one timeslot among all of the available timeslots 5) The mobile application prompts the patient for double confirmation on the appointment. 6) Upon success scheduling an appointment, the mobile application displays a success message. The web application displays a toast message to notify the respective medical staff.		
Sub Event Flow: -		
Optional Flow: 4.1) If the type of appointment of the medical staff is the appointment by number, the patient will get a number. 4.2) If no timeslot is found, the mobile application displays a message to notify the patient on the absence of available timeslot.		

Table 4-8 Use Case of Managing Appointments

<table>
<tr><td><b>Name</b>: Manage Appointments</td><td><b>ID</b>: 8</td><td><b>Priority</b>: Medium</td></tr>
<tr><td><b>Actor</b>: Patient, Medical Staff</td><td colspan="2"><b>Type</b>: Detail, Real</td></tr>
<tr><td colspan="3"><b>Stakeholder’s Information</b>:
Patient → person who make an appointment with the medical staff
Medical Staff → person who has appointments with the patients for consultation</td></tr>
<tr><td colspan="3"><b>Summary of Use Case</b>:
This use case illustrates the actions needed for the patient or medical staff to view the appointment they have.</td></tr>
<tr><td colspan="3"><b>Triggering Situation</b>:
1) The patient wishes to know the appointments made as well as those are nearing.
2) The medical staff wants to know the appointments made by the patients as well as the upcoming one.</td></tr>
<tr><td colspan="3"><b>Relationship</b>:
• Association: Patient, Medical Staff
• Include: -
• Extend: -</td></tr>
<tr><td colspan="3"><b>Normal Event Flow</b>:
<u>Patient</u>
1) The patient clicks on the “Appointment” tab in the mobile application.
2) The mobile application displays all of the appointments made in the order of the meeting time.
3) If the patient is unavailable during the meeting time of an appointment, the patient can either perform sub-flow 3a.1 or sub-flow 3.2.

<u>Medical Staff</u>
1) The medical staff clicks on the “Appointment” tab in the web application.
2) The web application displays all of the appointments in the order of the meeting time.
3) The medical staff needs to update the status of the appointment to confirm the attendance.
4) If the medical staff is unavailable during the meeting time of an appointment, the medical staff can reject the appointment.</td></tr>
</table>

Sub Event Flow:

Patient

3a.1. The patient can reschedule the appointment.

- 3a.1.1. The patient selects the appointment to be rescheduled.
- 3a.1.2. The patient clicks on the “Reschedule” button.
- 3a.1.3. The patient picks another timeslot.
- 3a.1.4. The mobile application prompts the patient for double confirmation on the reschedule by displaying the old and new information of the appointment.

3.2. The patient can cancel the appointment.

- 3.2.1. The patient selects the appointment to be cancelled.
- 3.2.2. The patient clicks on the “Cancel” button.
- 3.2.3. The mobile application prompts the patient for double confirmation on the cancellation of the appointment.

Optional Flow:

Patient

2) If no appointment made, the mobile application displays a null state message.

Medical Staff

2) If no appointment made by the patients, the web application displays a null state message.

Table 4-9 Use Case of Granting Data Access Request

<table>
<tr><td><b>Name</b>: Grant Data Access Authorization Permission</td><td><b>ID</b>: 9</td><td><b>Priority</b>: Medium</td></tr>
<tr><td><b>Actor</b>: Patient</td><td colspan="2"><b>Type</b>: Detail, Real</td></tr>
<tr><td colspan="3"><b>Stakeholder’s Information</b>:
Patient → person who allows other users of the system to authorize the data access request</td></tr>
<tr><td colspan="3"><b>Summary of Use Case</b>:
This use case defines the actions needed for the patient to grant permission to other users for access authorization during an emergency.</td></tr>
<tr><td colspan="3"><b>Triggering Situation</b>:
The patient needs someone in the system to authorize the data access request during an emergency.</td></tr>
<tr><td colspan="3"><b>Relationship</b>:
• Association: Patient
• Include: -
• Extend: -</td></tr>
<tr><td colspan="3"><b>Normal Event Flow</b>:
1) The patient clicks on the “Emergency Authorization” option in the profile interfaces of the mobile application.
2) The mobile application displays the list of users who granted data access authorization permission.
3) The patient clicks on the “Add Emergency Contact” button.
4) The patient enters the phone number of another user.
5) The system prompts the user for double confirmation before granting permission to another user.
6) The system displays a success message.
7) If the patient wishes to remove the existing emergency contact, the patient performs Sub Event Flow 6.1.</td></tr>
<tr><td colspan="3"><b>Sub Event Flow</b>:
6.1. The patient removes the existing emergency contact.
6.1.1. The patient clicks on the “Remove” option.
6.1.2. The patient checks on those contacts to be removed.
6.1.3. The patient clicks on the “Confirm” button.
6.1.4. Upon success removal, the mobile application displays a success message.</td></tr>
<tr><td colspan="3"><b>Optional Flow</b>:
5.1) If no user with the entered phone number found, the mobile application displays an error message.</td></tr>
</table>

Table 4-10 Use Case of Managing Profile

<table>
<tr><td><b>Name</b>: Manage Profile</td><td><b>ID</b>: 10</td><td><b>Priority</b>: Low</td></tr>
<tr><td><b>Actor</b>: Patient, Medical Staff</td><td colspan="2"><b>Type</b>: Detail, Real</td></tr>
<tr><td colspan="3"><b>Stakeholder’s Information</b>:
Patient → person who wants to view the profile detail
Medical Staff → person who wants to view the profile detail</td></tr>
<tr><td colspan="3"><b>Summary of Use Case</b>:
This use case outlines the steps needed for the patient or medical staff to view the profile detail.</td></tr>
<tr><td colspan="3"><b>Triggering Situation</b>: The patient or medical staff wishes to view and update the profile’s detail.</td></tr>
<tr><td colspan="3"><b>Relationship</b>:
• Association: Patient, Medical Staff
• Include: -
• Extend: -</td></tr>
<tr><td colspan="3"><b>Normal Event Flow</b>:
<u>Patient</u>
1) The patient clicks on the “Profile” tab in the mobile application.
2) The mobile application displays the profile of the patient account.
3) If the patient changes his/her personal information, the patient performs sub-flow 3a.1.

<u>Medical Staff</u>
1) The medical staff clicks on the name of the medical staff located at the right top corner in the web application.
2) The web application displays the profile of the medical staff account.
3) If the patient changes his/her personal information, the patient performs sub-flow 3b.1.</td></tr>
</table>

Sub Event Flow: Patient 3a.1. The patient updates his/her detail in the patient account. 3a.1.1. The patient clicks on the “Edit” icon located at the right top in the profile interface of the mobile application. 3a.1.2. The patient enters the new information of his/her account. 3a.1.3. The patient clicks on the green “Tick” icon located at the right top in the profile edit interface to confirm the changes. Medical Staff 3b.1. The medical staff updates his/her detail in the medical staff account. 3b.1.1. The medical staff clicks on the “Edit” icon in the web application. 3b.1.2. The web application displays a modal to prompt the medical staff for the latest profile detail. 3b.1.3. The medical staff clicks on the “Confirm” button to confirm the changes.
Optional Flow: -

4.4. Questionnaire Fact-Finding

According to the questionnaire done, 51 responses were received. Within 51 respondents, despite the majority of them received tertiary education, they seldom performed a body check as shown in Figure C-4.

Majority of them does not have the habit of tracking their health condition (Figure C-5), although they agreed that tracking health condition can help and improve the doctor diagnosis on the patient (Figure C-6). In Figure C-8, the majority mentioned that they are rarely sick and a portion of them felt troublesome in keeping these records. For the minority, they usually require an application in helping them to keep track of their health records (Figure C-7).

Other than that, the majority of the respondents supported to have an online scheduling system (Figure C-9). Then, they can queue for their turn for consultation without the need to go to the hospital or clinic earlier and wait there. This is because they did not know the exact time for their consultation time.

Moreover, based on Figure C-11, since most of the respondents are teenagers, they did not have the issue of forgetting about their medication refill. However, they mentioned that alarm and having an event in their calendar would be good approaches in reminding them to refill their medication (Figure C-12).

Furthermore, based on Figure C-14 majority of the respondents admitted that graphical visualization of their health records can aid them in understanding their health condition better. This is because the majority of them can barely understand the content without the guidance of the medical staff as shown in Figure C-13. Since different data may require different data presentation, according to their selections in Figure C-15, they would prefer to have the data to be presented in a table, followed by a line graph and a pie chart. On top of that, they also prefer to keep track of their BMI index, blood sugar level and the recovery period of their disease or illness (Figure C-16).

4.5. Summary

In short, this chapter specifies the functional requirement for both frontend applications and backend server. A use case diagram was prepared to pinpoint the activities a user can perform in the applications in both mobile and web platforms.

CHAPTER 5

DESIGN

5.1. Introduction

This chapter lists the modelling diagrams of the entire implementation. Besides, the architecture design of the system was demonstrated to show the communication between instances, such as backend server and frontend applications. Besides, the wireframes and initial system designs were constructed to visualize the designs of the system.

5.2. System Architecture Design

In this project, three-tiers architecture was implemented. It consists of three layers, which are frontend layer, backend layer and database layer. Frontend layer is the presentation layer of the system, as it is responsible for the visualization of the system. For the backend layer, it mainly focuses on the processing of business logic and data transactions. Without the backend layer, the presentation layer has no way to retrieve data from the database. At last, the database layer will store and retrieve the data for the processing in the backend layer and display in the frontend layer.

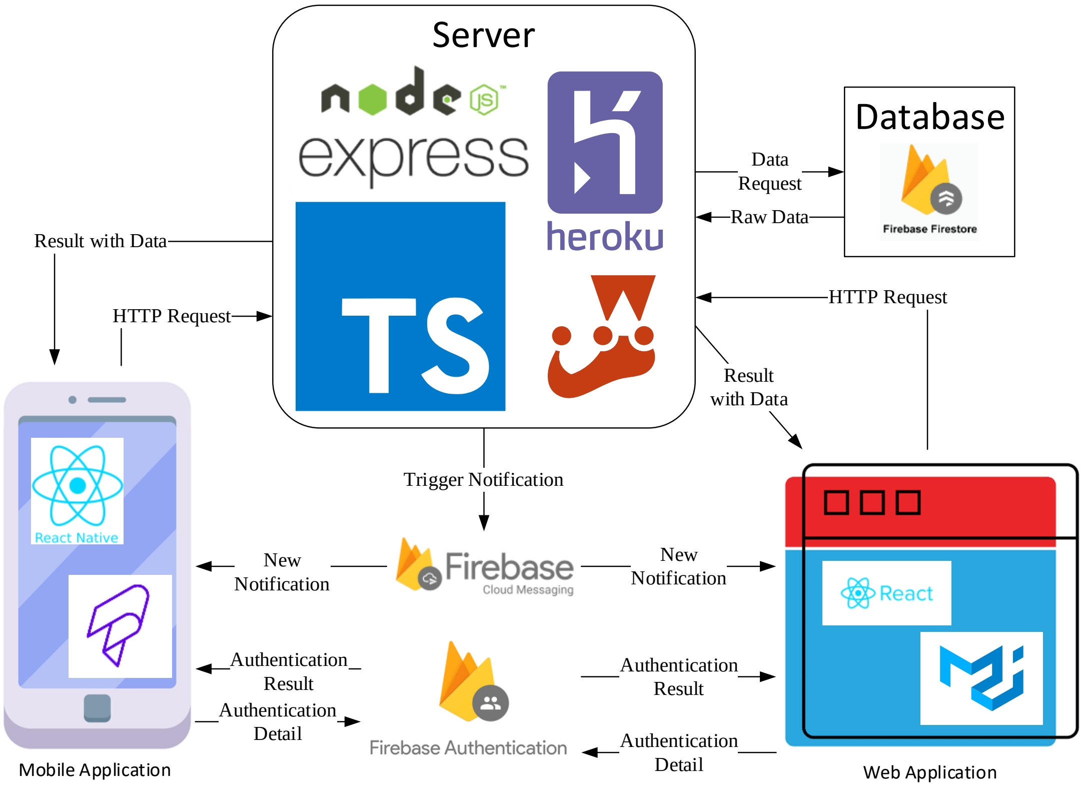


Figure 5.1 Architecture Design of the System

In the backend layer, the system uses NodeJS, Express's, Jest, Typescript and Firebase. ExpressJS allows the implementation of API as it can handle the request sent from the frontend. In short, ExpressJS allows the creation of Restful Application. For database connection, Firebase is integrated with ExpressJS in the backend. Additionally, the backend server uses the messaging service provided by Firebase to send notifications to the users' devices. Other than that, the backend layer also uses Jest for the testing of the business logic. It allows the implementation of the test cases in the backend. On top of that, Typescript is used in tightening the rule during the coding process so much compile errors can be eliminated before the application is compiled for a build. Lastly, Heroku is used for server hosting, so the server can be accessed in an actual environment.

Similar to the backend layer, the presentation layer uses Typescript as well to enhance the implementation. In the presentation layer, both the frontend applications use Firebase Authentication and Firebase Messaging. Firebase Authentication allows the applications to connect to Firebase for their access authentication. It handles the registration of a new account as well. For Firebase Messaging, it allows the devices that are installed with the applications to receive notification from either the Firebase Console or backend server. In the implementation of this project, most notifications are triggered by the backend server to the frontend applications.

In terms of the design, the mobile frontend application uses React Native. In the implementation, React Native Paper library is used as it provides a set of components which provide good functionalities as well as good UI presentation. Meanwhile, the web frontend application uses React. The web frontend application is developed with the Material-UI library. With the usage of third-party libraries, it ensures a fast and easy application development experience.

5.3. Modelling Diagram

The diagrams included are data model diagram and data flow diagram. The data model diagram describes the actual structure of the data stored in the Firebase database. For the data flow diagram, it describes the flow of information between the actors, database as well as the activities.

5.3.1. Data Modelling

This section describes the conceptual and physical model of the data structure implemented in the system. It provides a clearer image of the data structure as well as the relationship between the collections. Additionally, it reduces the chances of having confusion during the implementation of the data structure and enhanced the performance of the development.

5.3.1.1. Conceptual Model Diagram

The conceptual model diagram shows the entity-relationship present within the project. It illustrates an overview of the system's data structure.

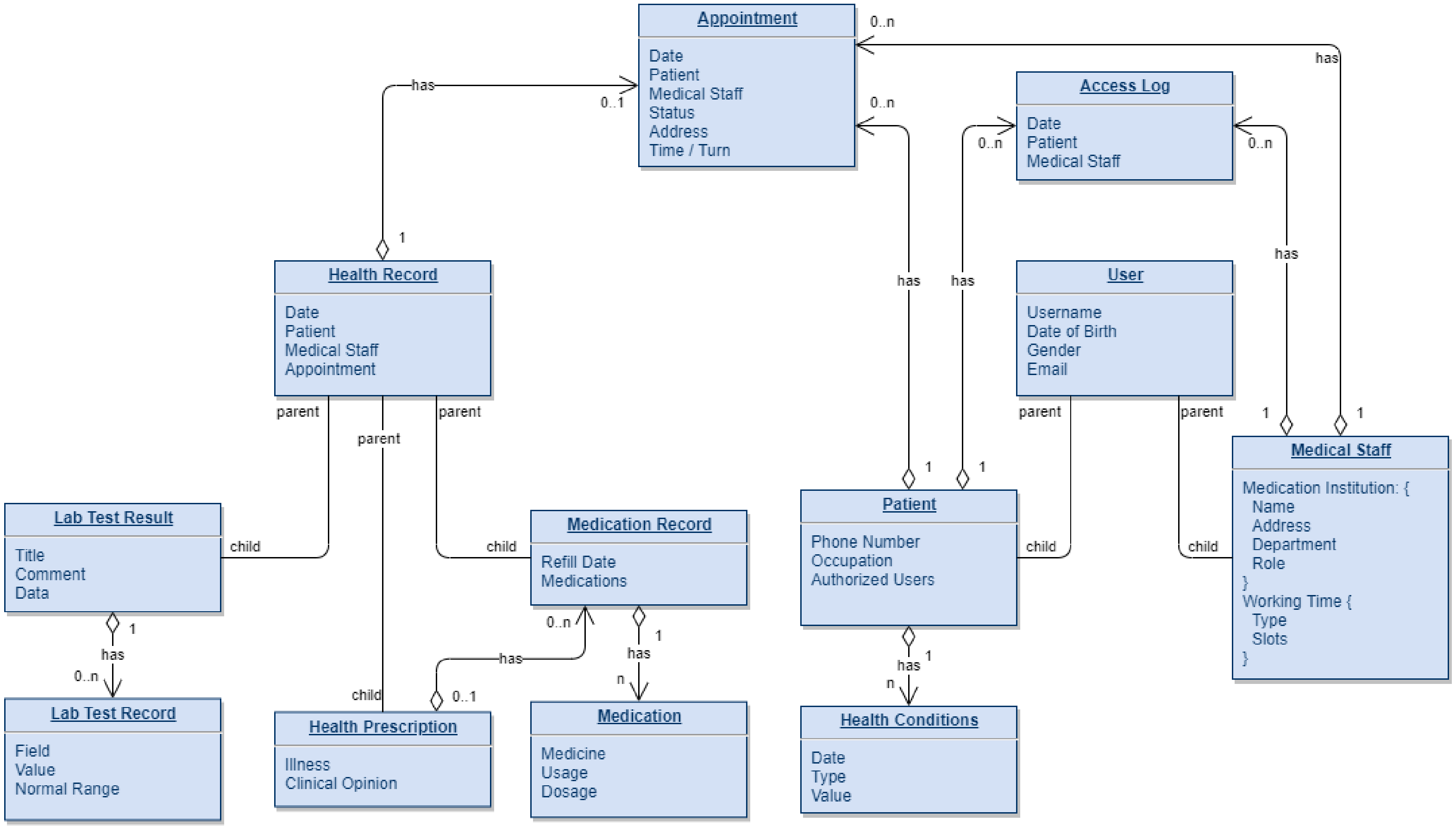
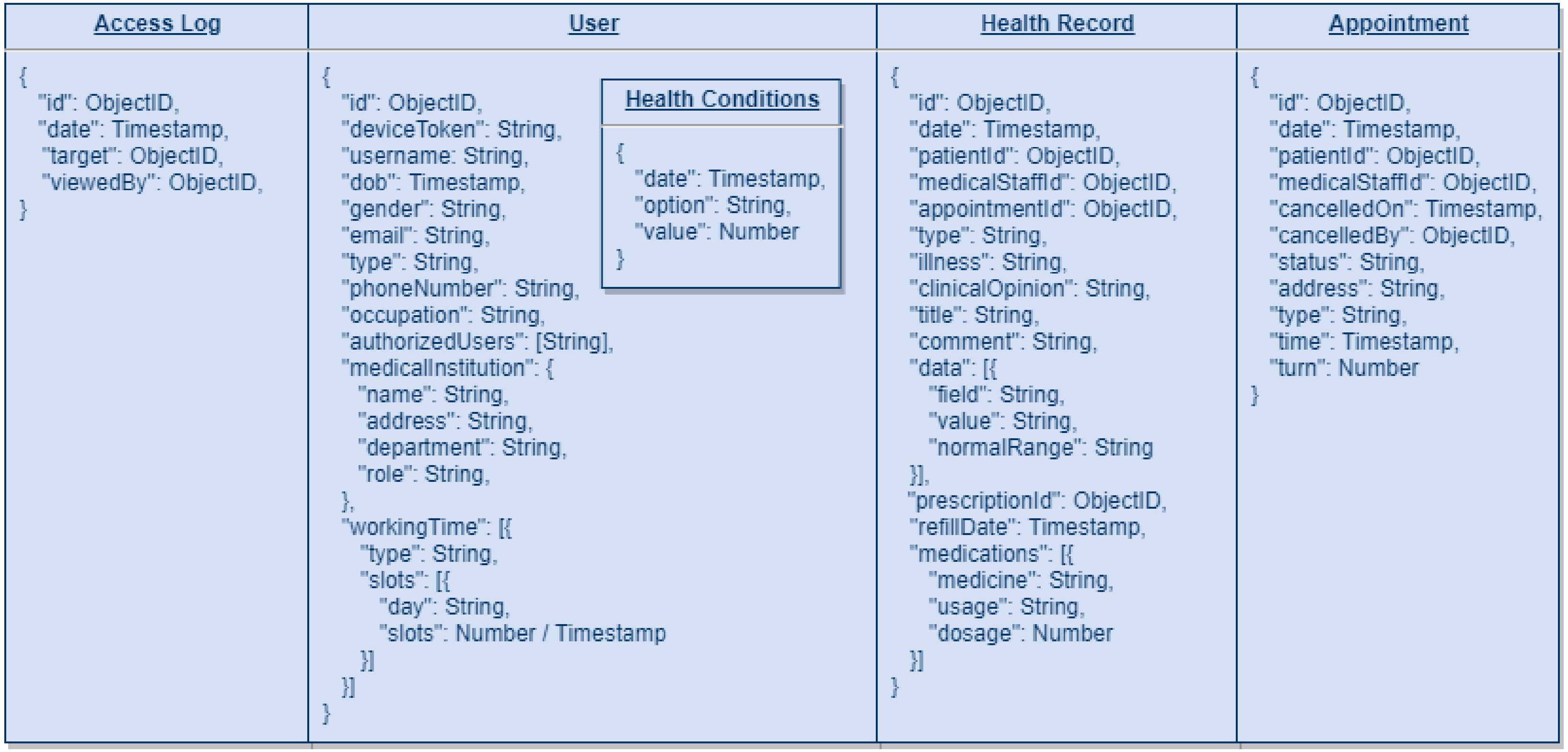


Figure 5.2 Conceptual Model Design of the Data Structure

5.3.1.2. Data Model Diagram

Data model diagram shows the actual data structure implemented in this project. Each of the documents has multiple fields that have different data types. In total, four collections are created in the project, which are the access log, user, health record and appointment collection. Additionally, each user document consists of health conditions sub-collection.



Access Log

```
{
  "id": ObjectID,
  "date": Timestamp,
  "target": ObjectID,
  "viewedBy": ObjectID,
}
```

User

```
{
  "id": ObjectID,
  "deviceToken": String,
  "username: String,
  "dob": Timestamp,
  "gender": String,
  "email": String,
  "type": String,
  "phoneNumber": String,
  "occupation": String,
  "authorizedUsers": [String],
  "medicalInstitution": {
    "name": String,
    "address": String,
    "department": String,
    "role": String,
  },
  "workingTime": [{
    "type": String,
    "slots": [{
      "day": String,
      "slots": Number / Timestamp
    }]
  }]
}
```

Health Conditions

```
{
  "date": Timestamp,
  "option": String,
  "value": Number
}
```

Health Record

```
{
  "id": ObjectID,
  "date": Timestamp,
  "patientId": ObjectID,
  "medicalStaffId": ObjectID,
  "appointmentId": ObjectID,
  "type": String,
  "illness": String,
  "clinicalOpinion": String,
  "title": String,
  "comment": String,
  "data": [{
    "field": String,
    "value": String,
    "normalRange": String
  }],
  "prescriptionId": ObjectID,
  "refillDate": Timestamp,
  "medications": [{
    "medicine": String,
    "usage": String,
    "dosage": Number
  }]
}
```

Appointment

```
{
  "id": ObjectID,
  "date": Timestamp,
  "patientId": ObjectID,
  "medicalStaffId": ObjectID,
  "cancelledOn": Timestamp,
  "cancelledBy": ObjectID,
  "status": String,
  "address": String,
  "type": String,
  "time": Timestamp,
  "turn": Number
}
```

Figure 5.3 Physical Model of the Data Structure

5.3.2. Data Flow Diagram

This section illustrates the flow of information between modules in the system and the actors. In total, 3 layers of diagrams were constructed to showcase the entire information flow. The layers included are context layer, level 0 and level 1. As deeper the layer is, the more detailed flow of information was revealed between the modules and data stores.

5.3.2.1. Context Diagram

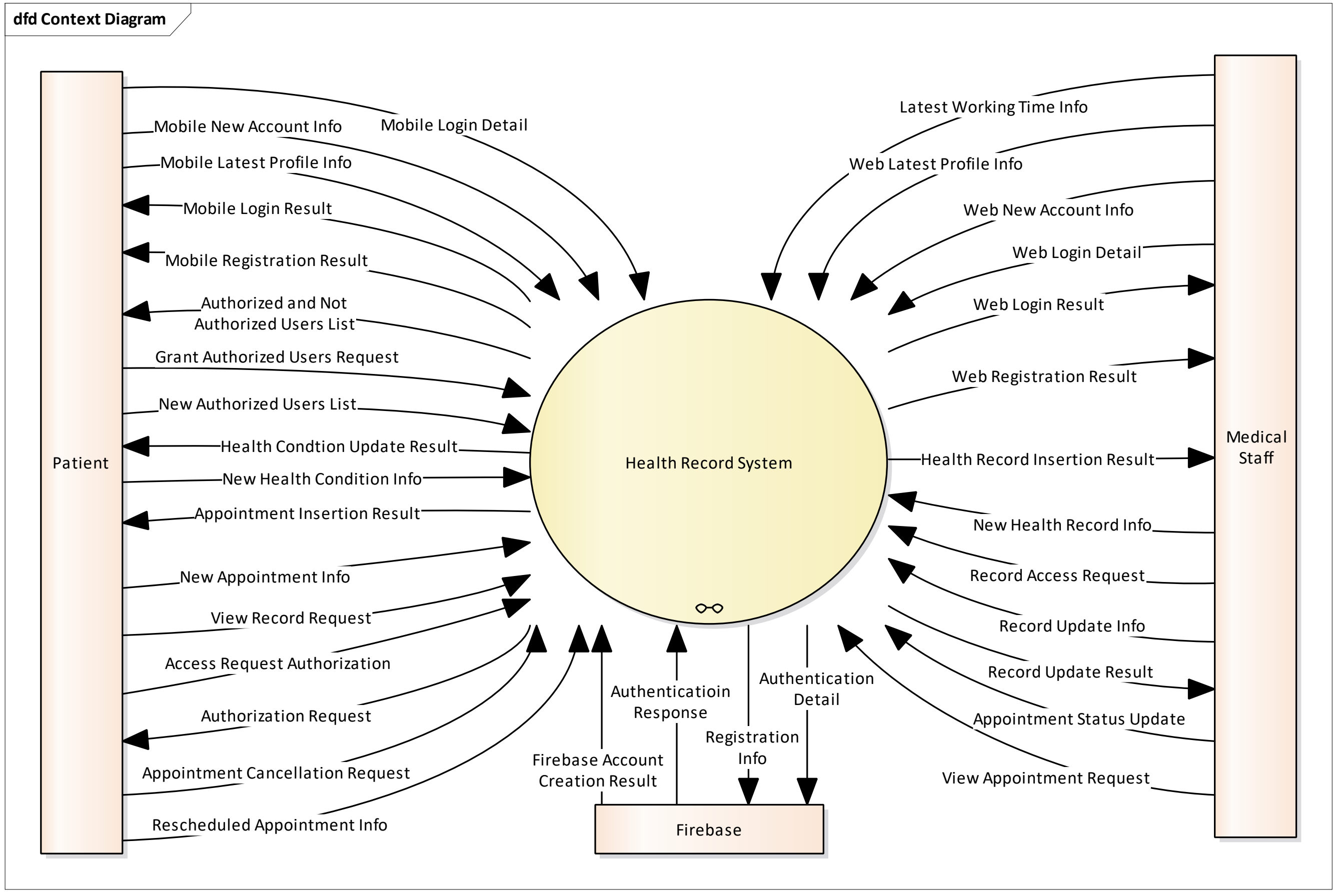


Figure 5.4 Context Diagram of Health Record System

5.3.2.2. Level 0 Diagram

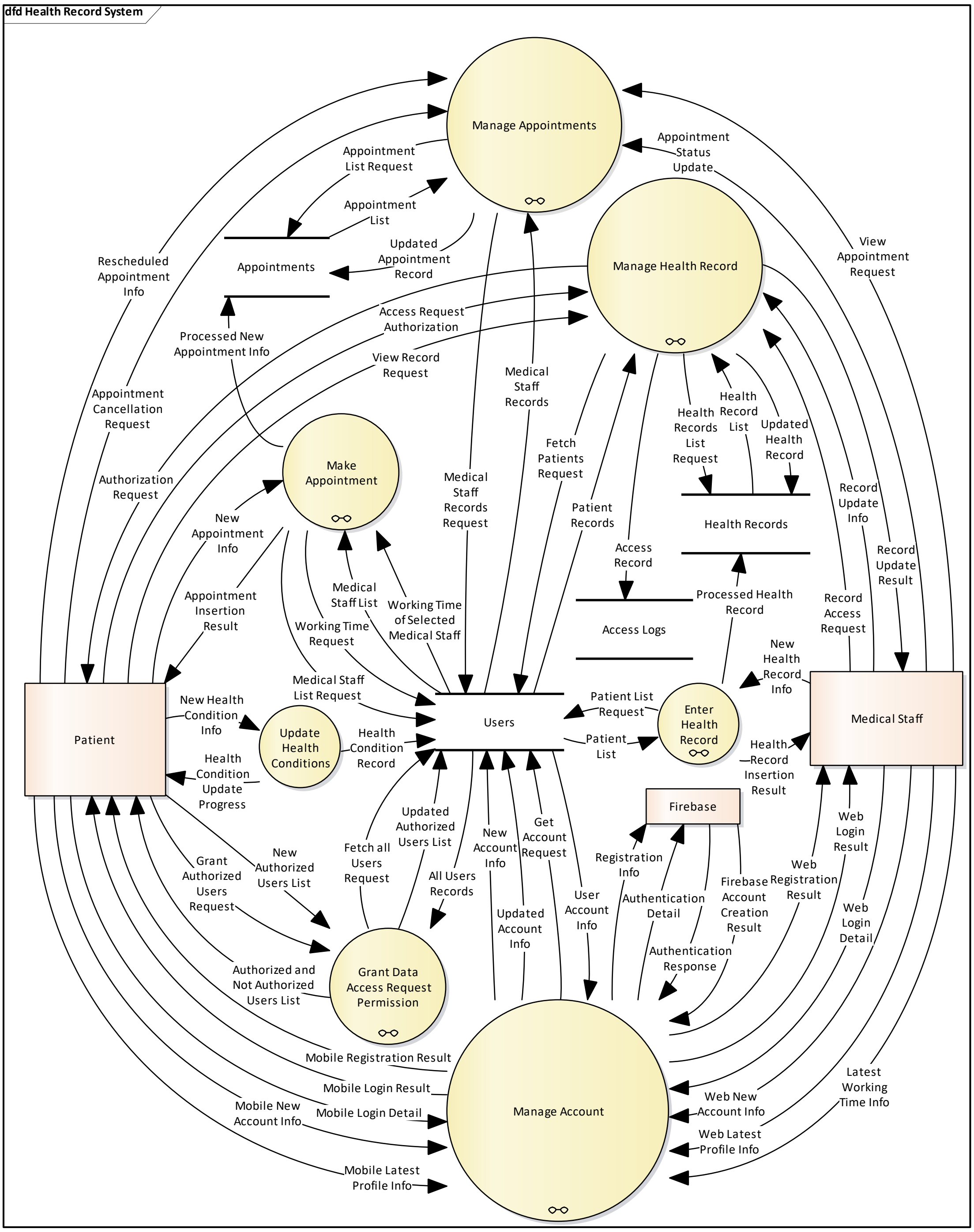


Figure 5.5 Level 0 Data Flow Diagram

5.3.2.3. Level 1 Diagram

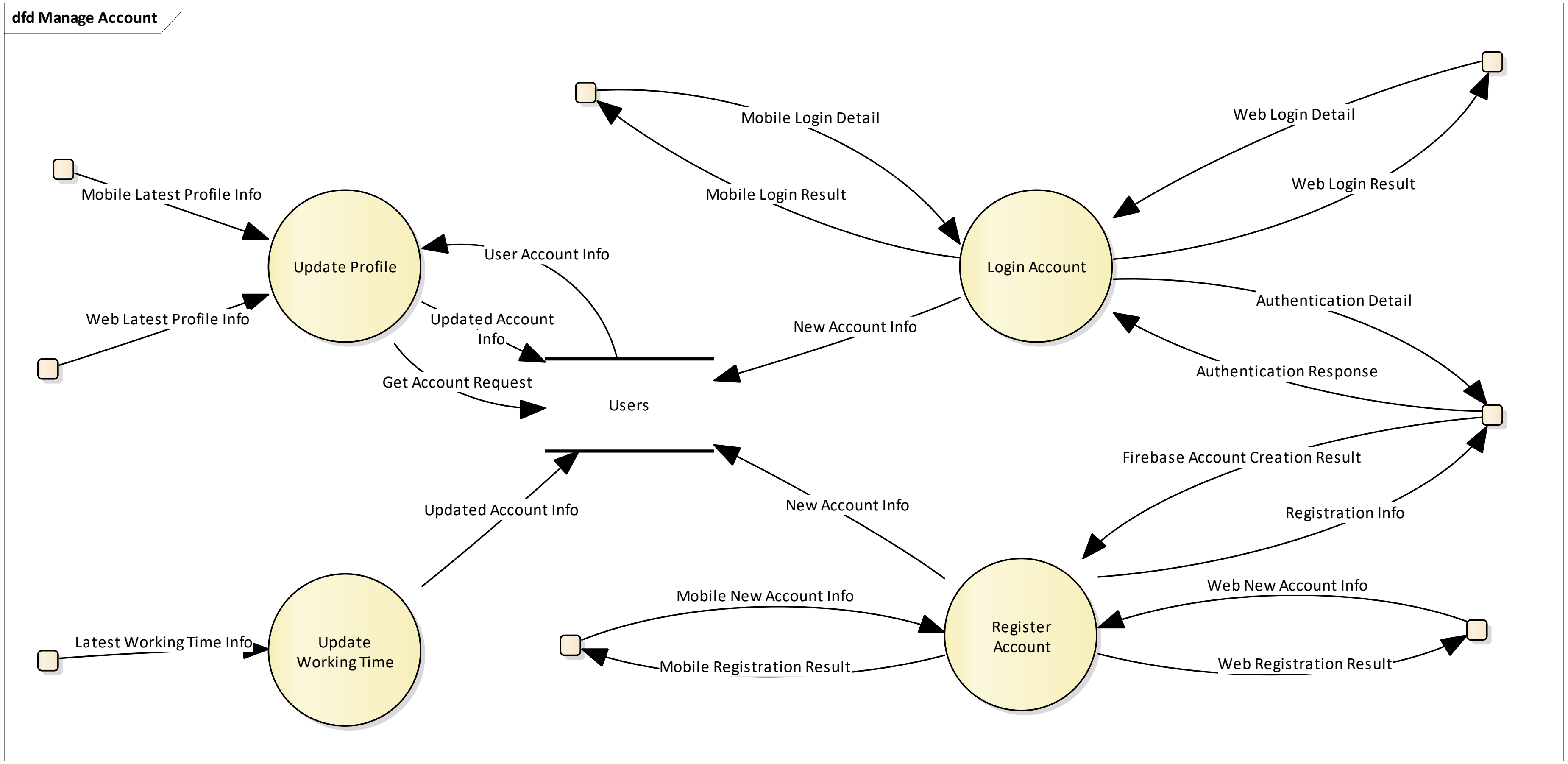


Figure 5.6 Level 1 Diagram for “Manage Account” Process

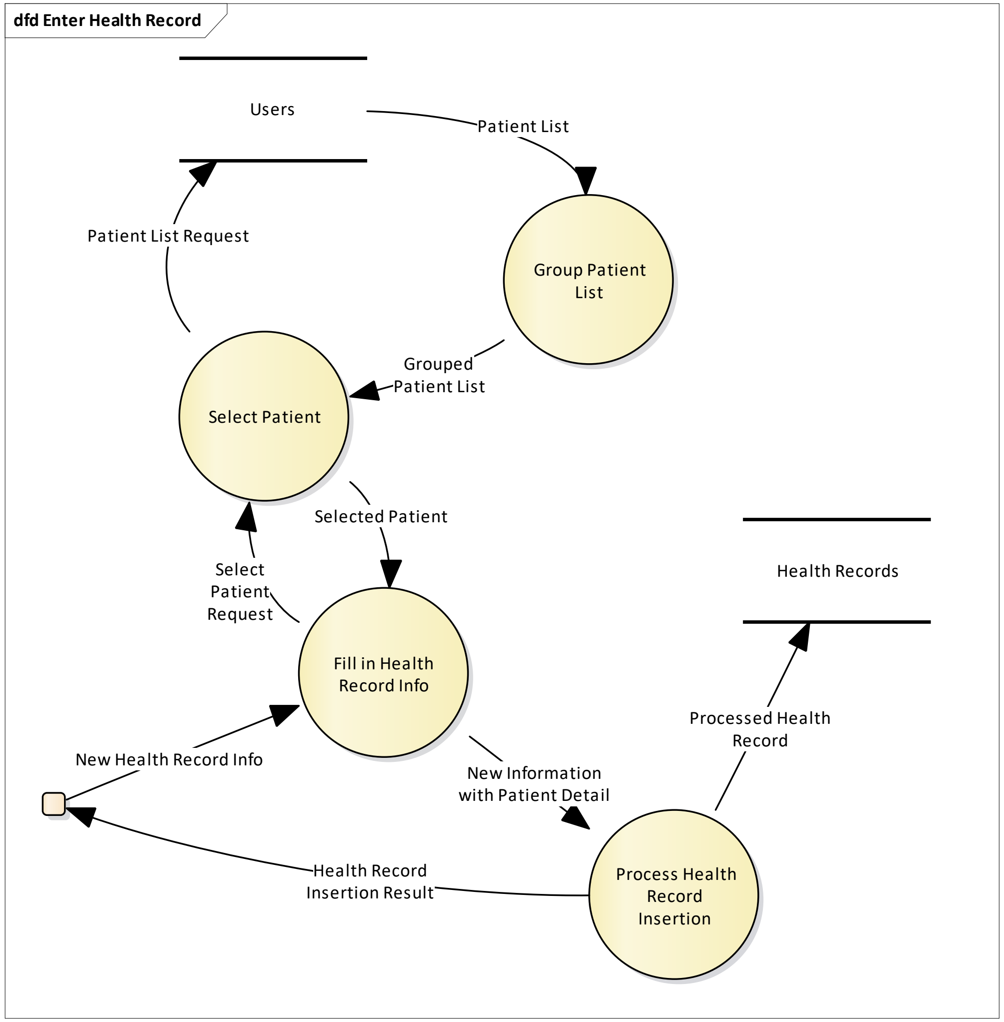


Figure 5.7 Level 1 Diagram for “Enter Health Record” Process

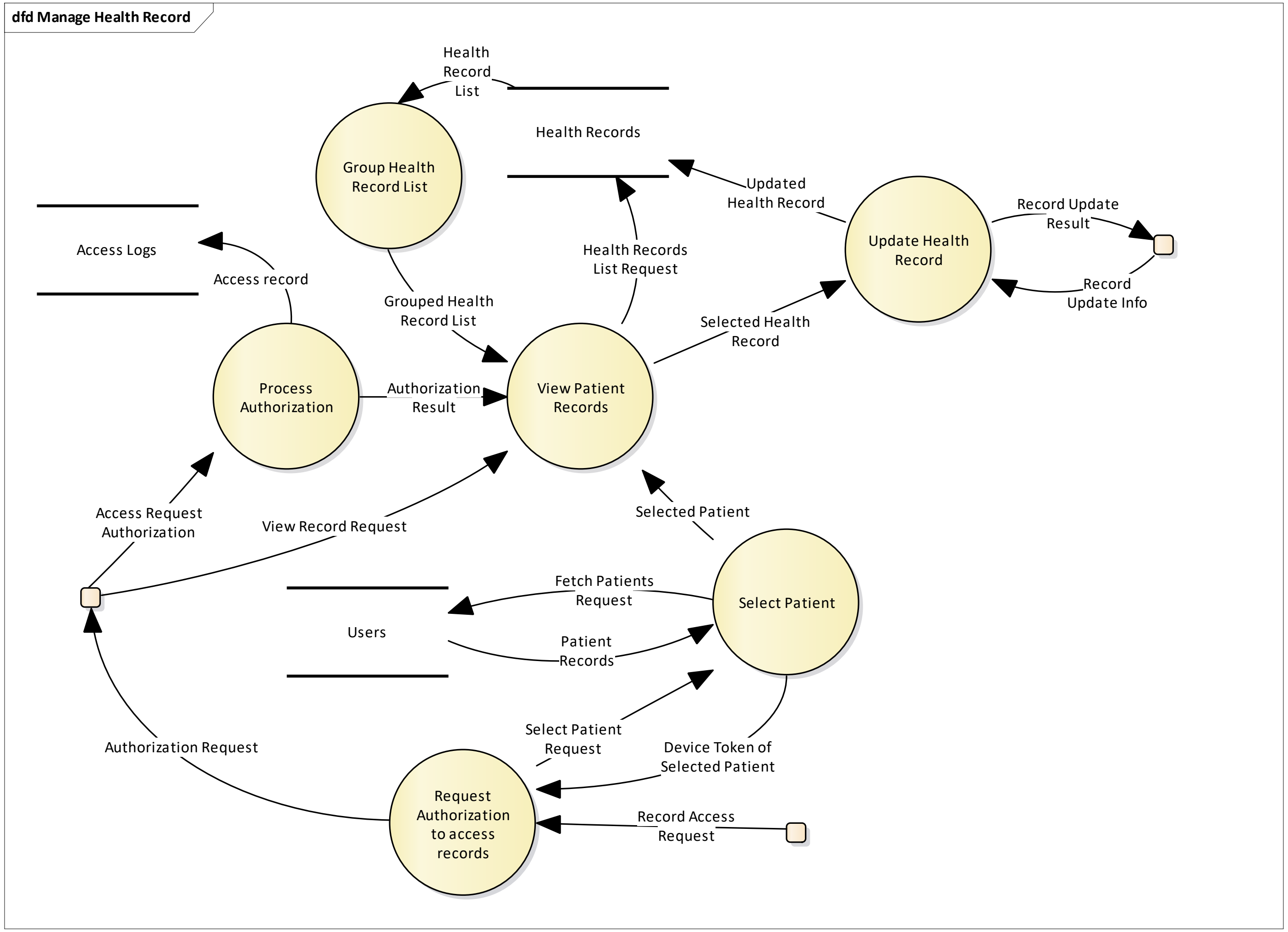


Figure 5.8 Level 1 Diagram of "Manage Health Record" Process

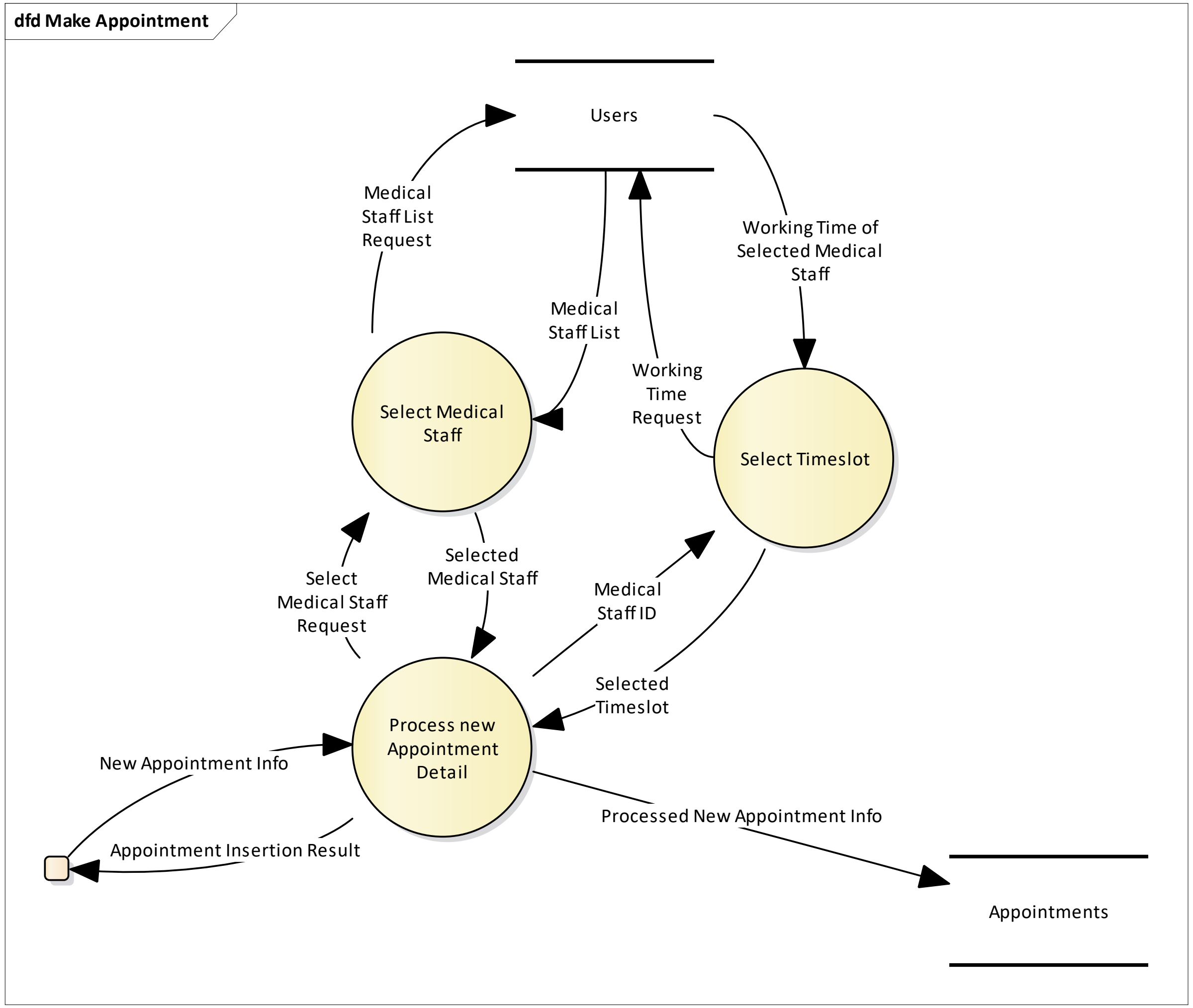


Figure 5.9 Level 1 Diagram for “Make Appointment” Process

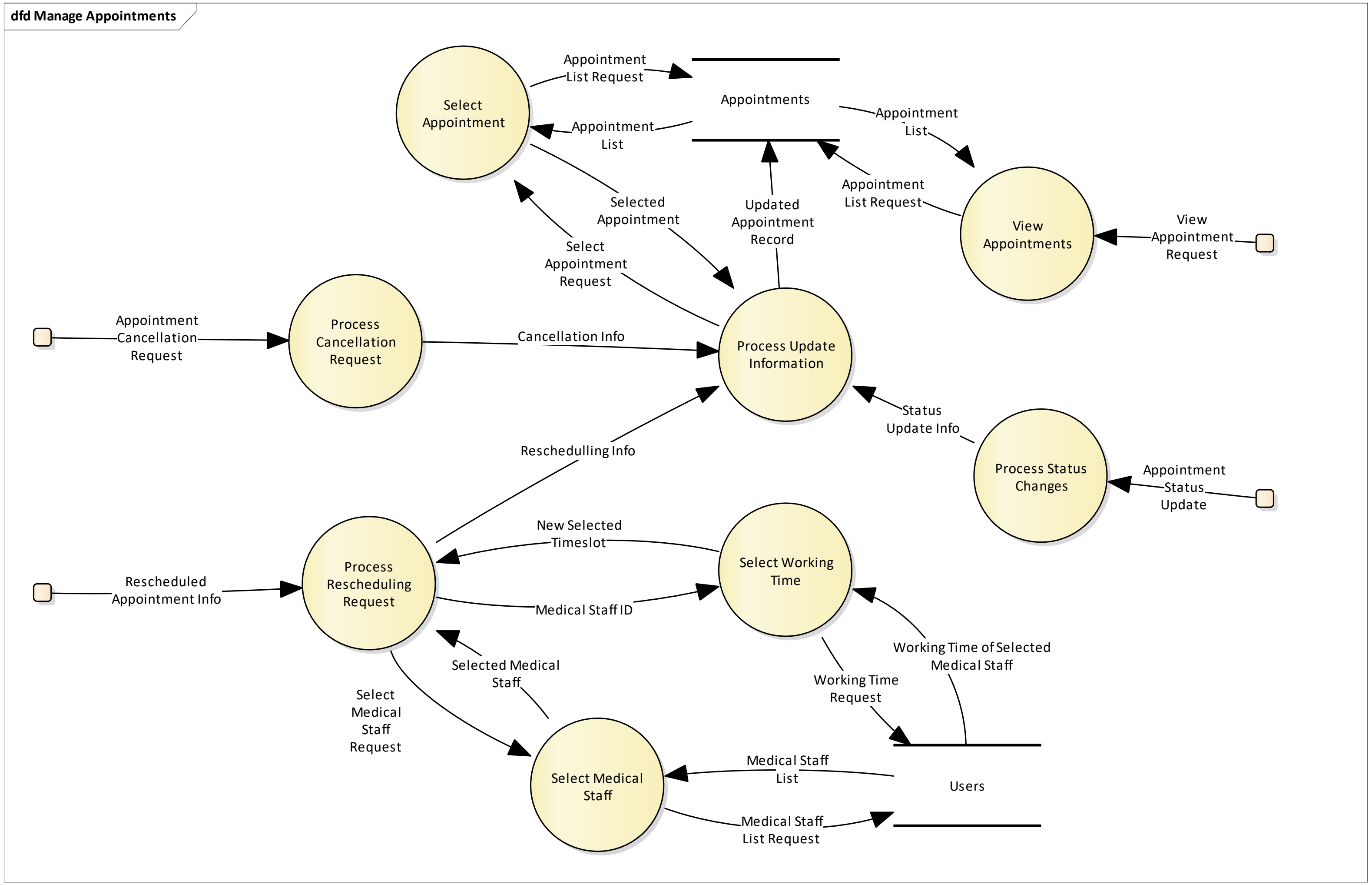


Figure 5.10 Level 1 Diagram of “Manage Appointment” Process

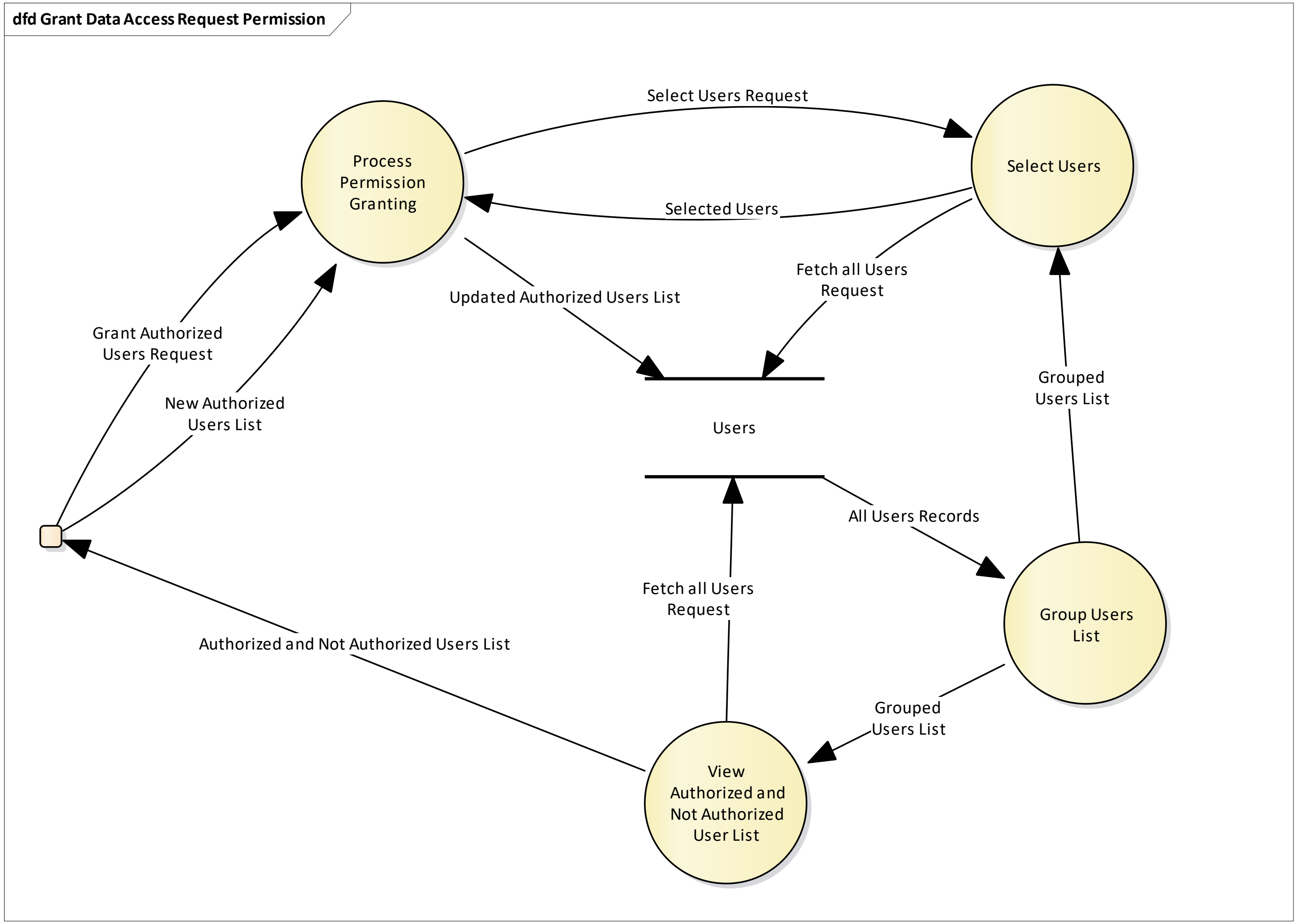

dfd Grant Data Access Request Permission
Select Users Request
Process Permission Granting
Select Users
Selected Users
Fetch all Users Request
Updated Authorized Users List
Grant Authorized Users Request
New Authorized Users List
Users
Grouped Users List
All Users Records
Fetch all Users Request
Group Users List
Authorized and Not Authorized Users List
Grouped Users List
View Authorized and Not Authorized User List

Figure 5.11 Level 1 Diagram of “Grant Data Access Request Permission” Process

5.4. Low-Fidelity Prototype

The following are the skeleton of the application for both mobile and web application. It provided a brief concept on the UX/UI of the final product.

5.4.1. Mobile Application Design

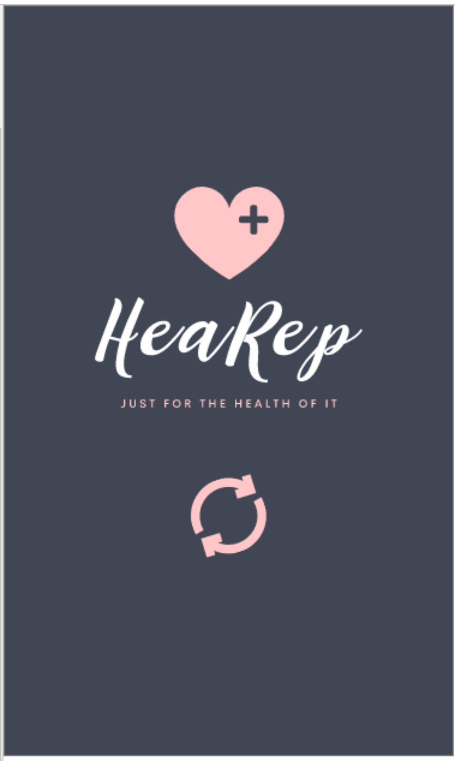


Figure 5.12 Loading Screen of Mobile Application

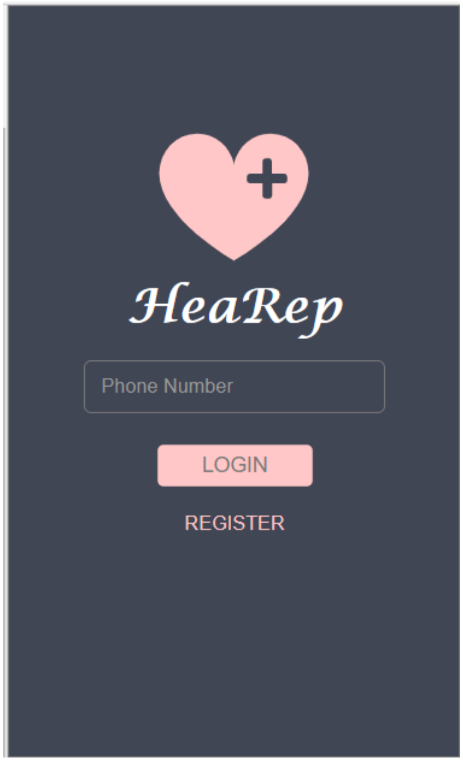


Figure 5.13 Login Screen

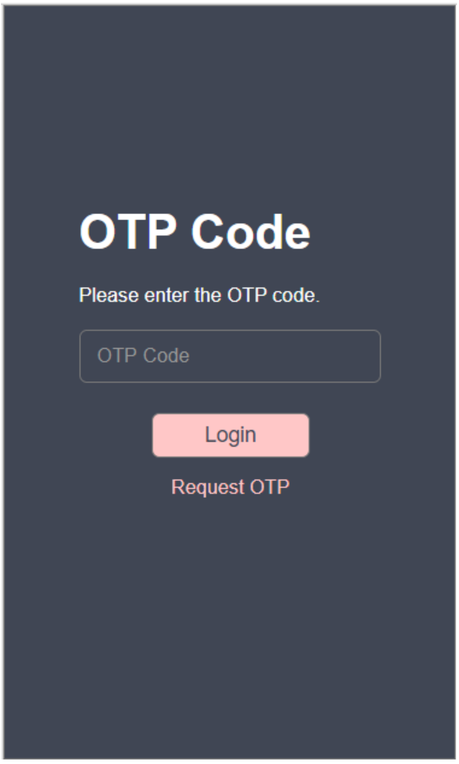
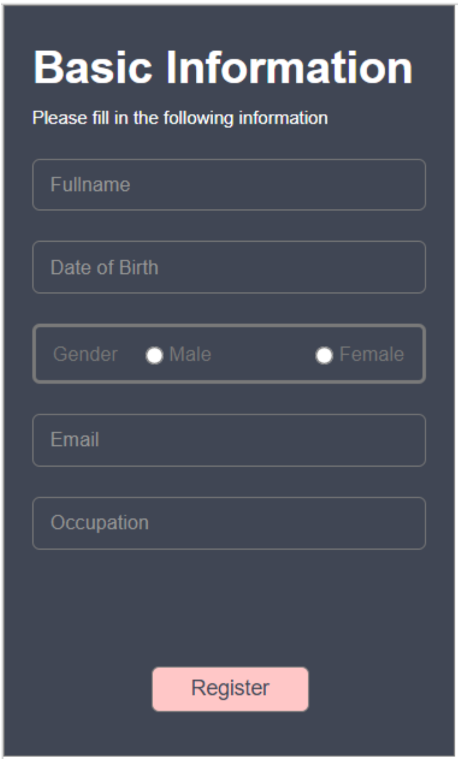


Figure 5.14 OTP Entry Screen



Basic Information

Please fill in the following information

Fullname

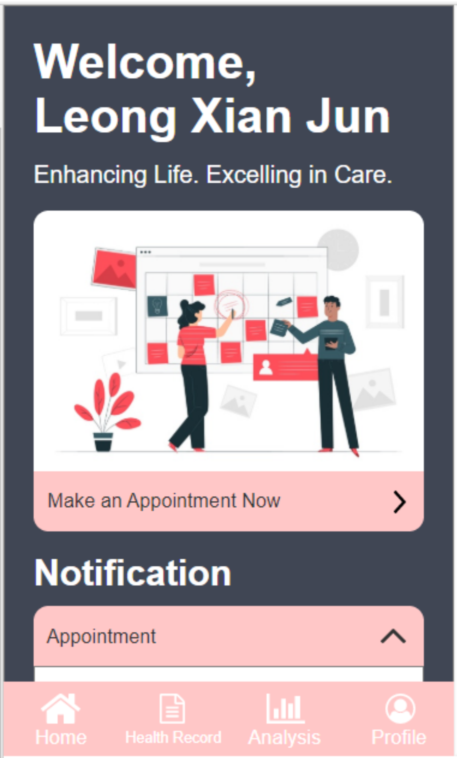
Date of Birth

Gender ● Male ● Female

Email

Occupation

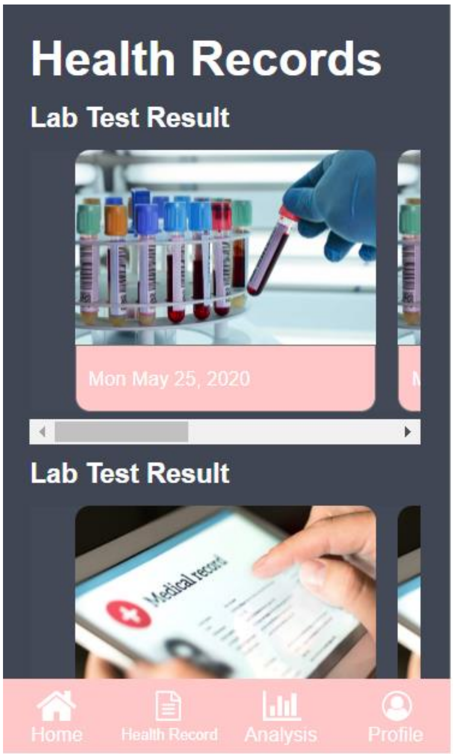
Figure 5.15 Account Detail Screen for new Account



Welcome, Leong Xian Jun

Enhancing Life. Excelling in Care.

Figure 5.16 Welcome Screen of Mobile Application



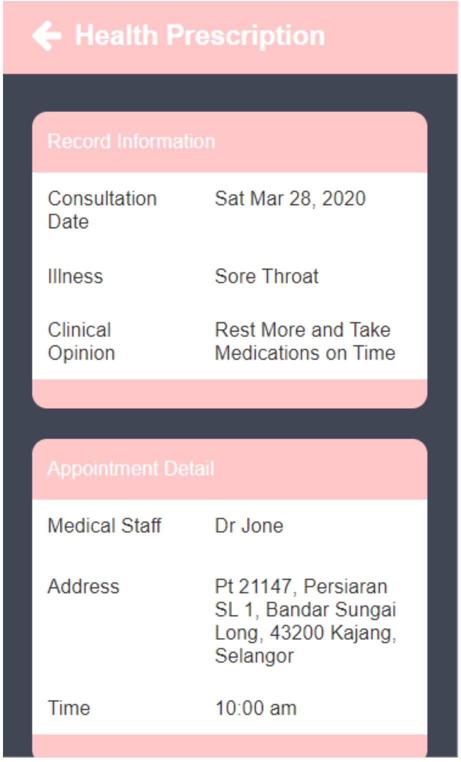
Health Records

Lab Test Result

Mon May 25, 2020

Lab Test Result

Figure 5.17 Health Record Listing Screen of Mobile Application



Health Prescription

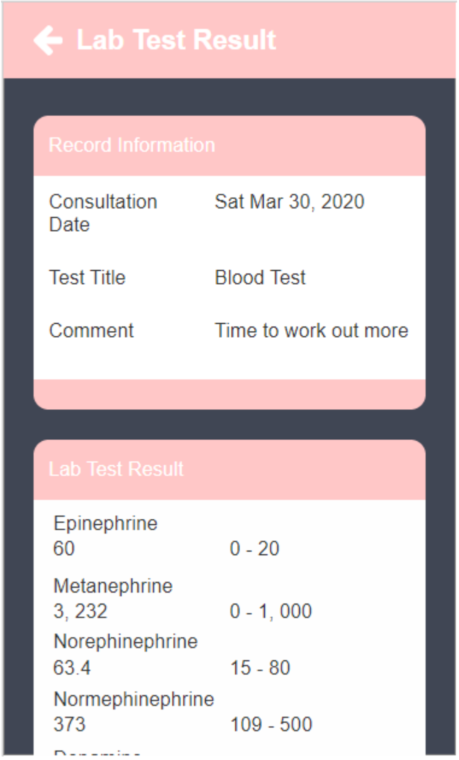
Record Information

Consultation Date	Sat Mar 28, 2020
Illness	Sore Throat
Clinical Opinion	Rest More and Take Medications on Time

Appointment Detail

Medical Staff	Dr Jone
Address	Pt 21147, Persiaran SL 1, Bandar Sungai Long, 43200 Kajang, Selangor
Time	10:00 am

Figure 5.18 Health Prescription Detail Screen



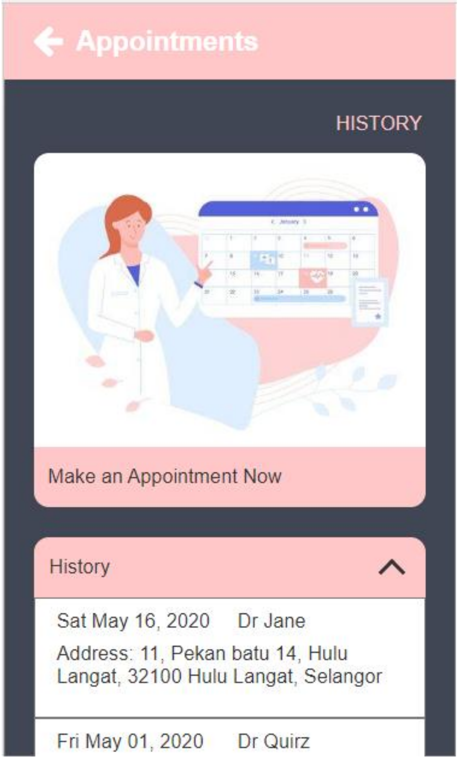
Record Information

Consultation Date	Sat Mar 30, 2020
Test Title	Blood Test
Comment	Time to work out more

Lab Test Result

Epinephrine 60	0 - 20
Metanephrine 3, 232	0 - 1, 000
Norephinephrine 63.4	15 - 80
Normephinephrine 373	109 - 500

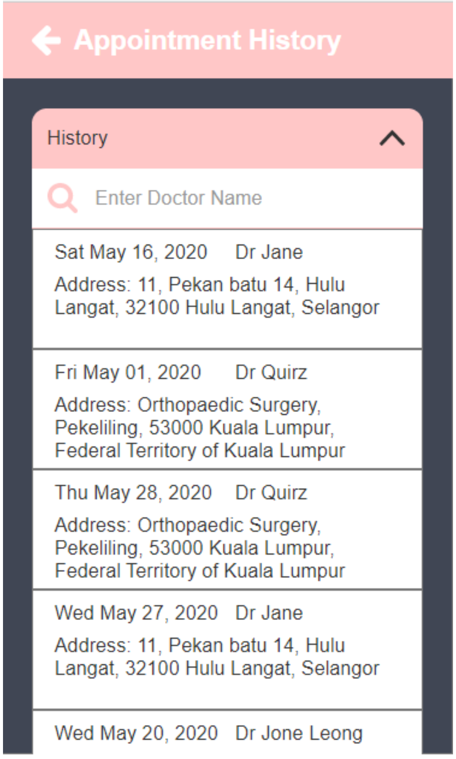
Figure 5.19 Lab Test Result Detail Screen



Sat May 16, 2020 Dr Jane
Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor

Fri May 01, 2020 Dr Quirz

Figure 5.20 Appointment Main Page



Sat May 16, 2020 Dr Jane
Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor

Fri May 01, 2020 Dr Quirz
Address: Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory of Kuala Lumpur

Thu May 28, 2020 Dr Quirz
Address: Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory of Kuala Lumpur

Wed May 27, 2020 Dr Jane
Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor

Wed May 20, 2020 Dr Jone Leong

Figure 5.21 Listing of Appointment History

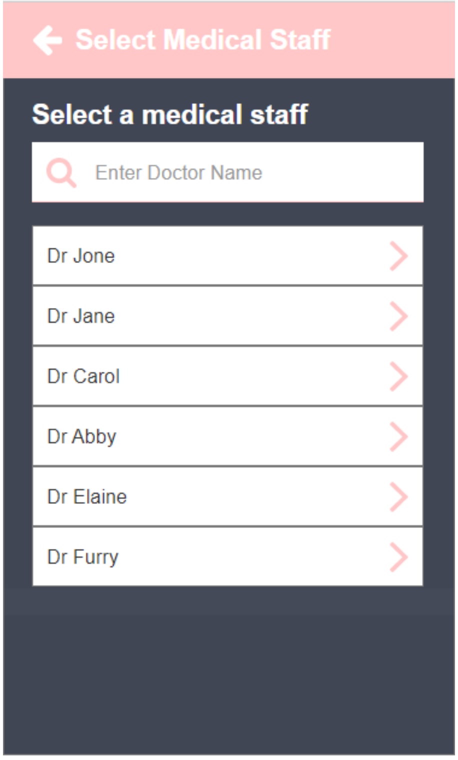


Figure 5.22 Screen to select a medical staff

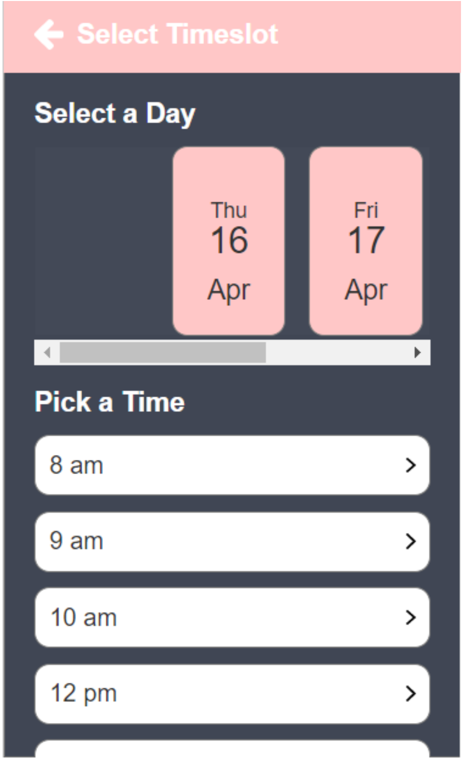
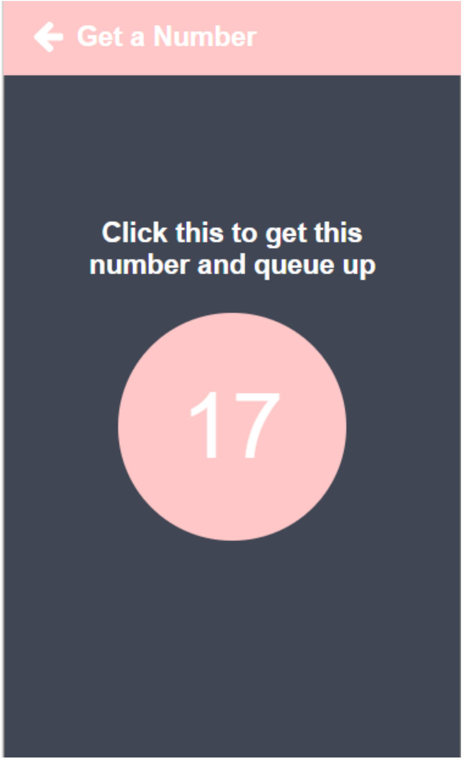
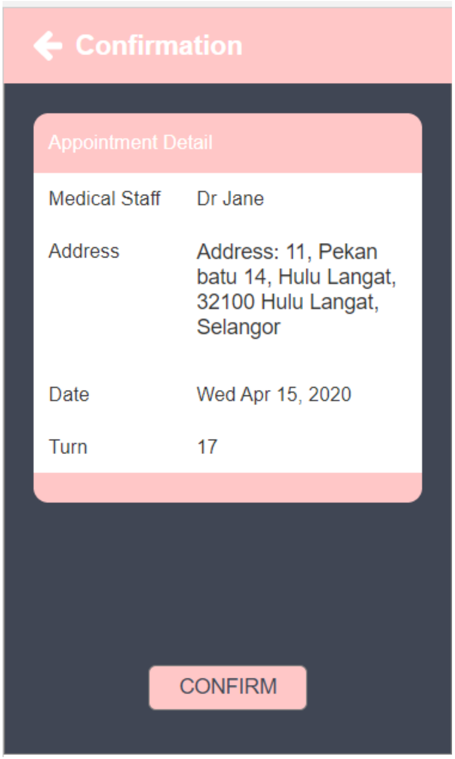


Figure 5.23 Screen to select a timeslot



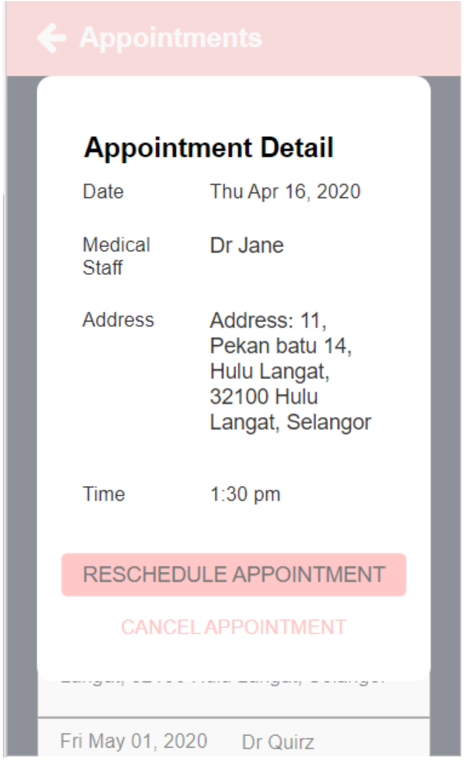
Click this to get this number and queue up

Figure 5.24 Screen to get a number for queueing



Medical Staff	Dr Jane
Address	Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor
Date	Wed Apr 15, 2020
Turn	17

Figure 5.25 Appointment Confirmation Screen



Appointment Detail

Date	Thu Apr 16, 2020
Medical Staff	Dr Jane
Address	Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor
Time	1:30 pm

Fri May 01, 2020 Dr Quirz

Figure 5.26 Appointment Detail Dialog

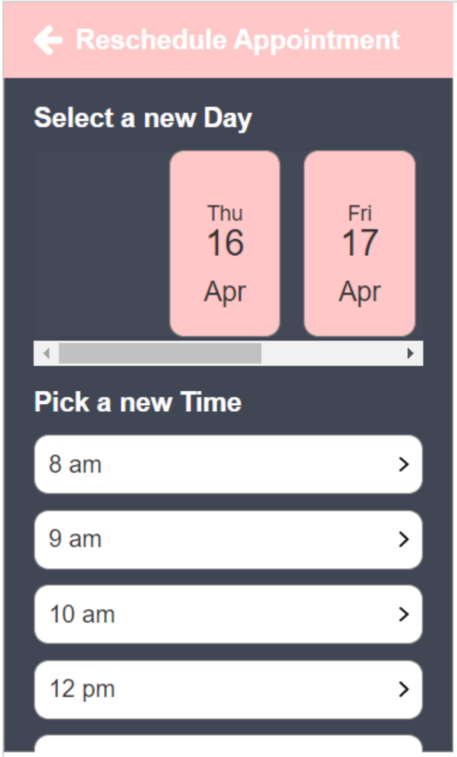
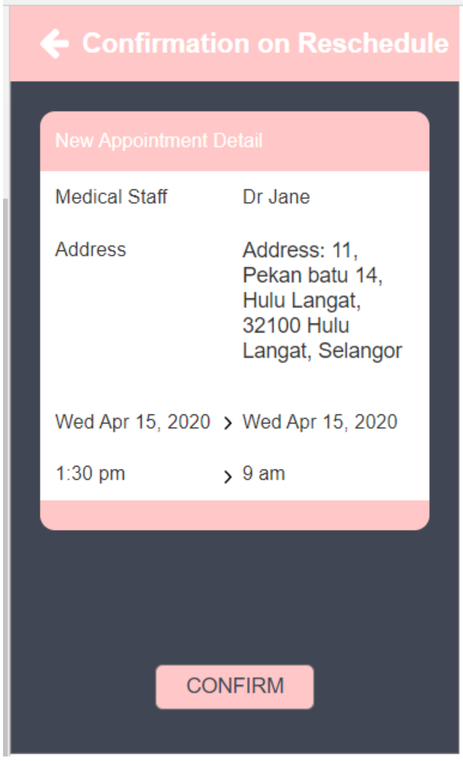


Figure 5.27 Screen to reselect timeslot when rescheduling Appointment



Medical Staff	Dr Jane
Address	Address: 11, Pekan batu 14, Hulu Langat, 32100 Hulu Langat, Selangor
Wed Apr 15, 2020 ›	Wed Apr 15, 2020
1:30 pm ›	9 am

Figure 5.28 Detail of Appointment Rescheduling Screen

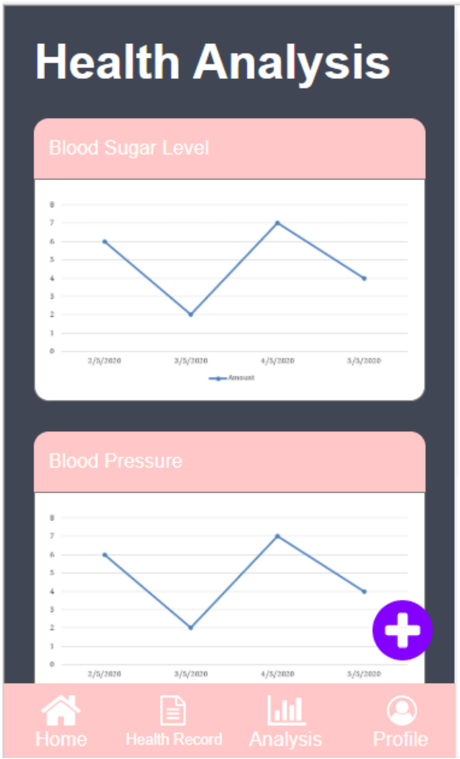
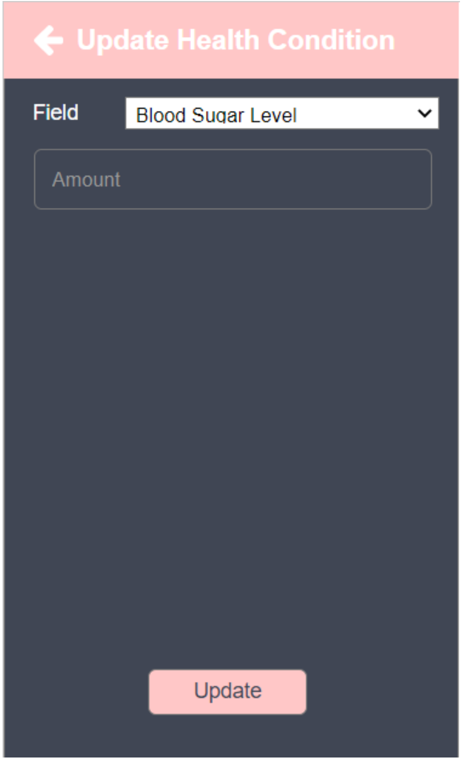
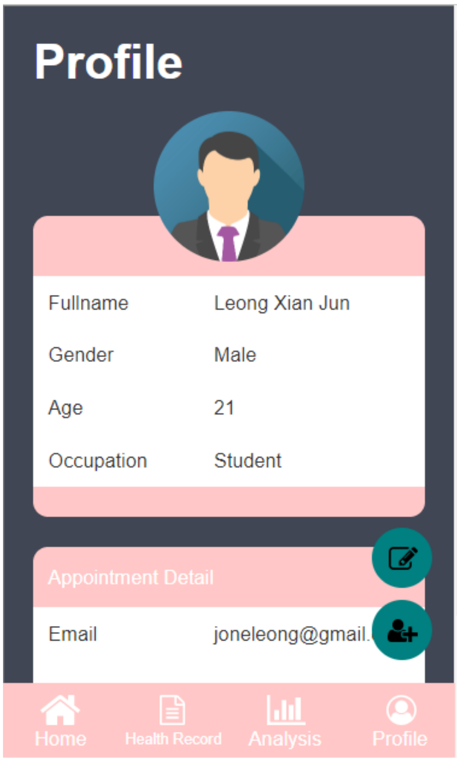


Figure 5.29 Health Analysis Screen



Field: Blood Sugar Level

Figure 5.30 Screen to update health condition



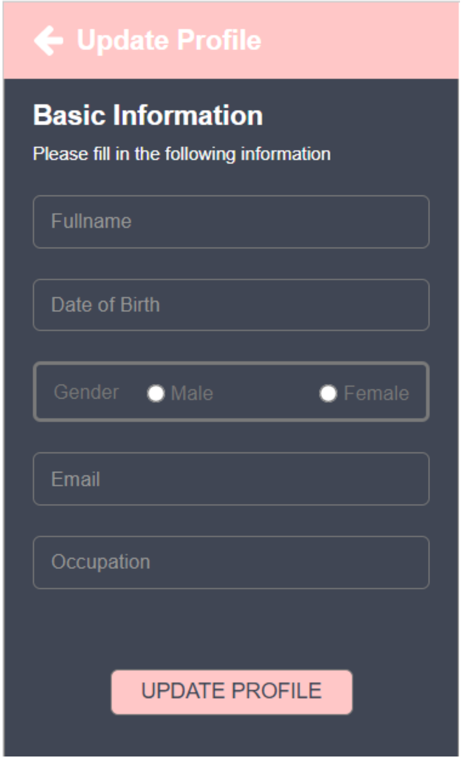
Profile

Fullname	Leong Xian Jun
Gender	Male
Age	21
Occupation	Student

Appointment Detail

Email	joneleong@gmail.

Figure 5.31 Profile Screen



Update Profile

Basic Information

Please fill in the following information

Fullname

Date of Birth

Gender ○ Male ○ Female

Email

Occupation

Figure 5.32 Screen to update the profile detail

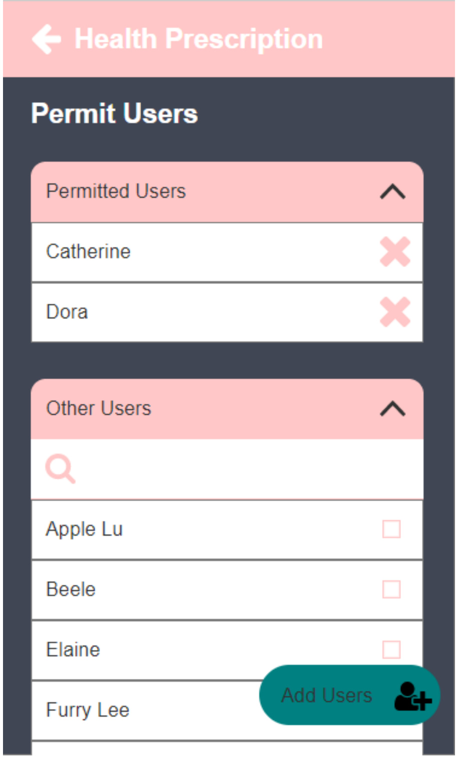


Figure 5.33 Screen to view and update authorized users to permit the access authorization

5.4.2. Web Application Design

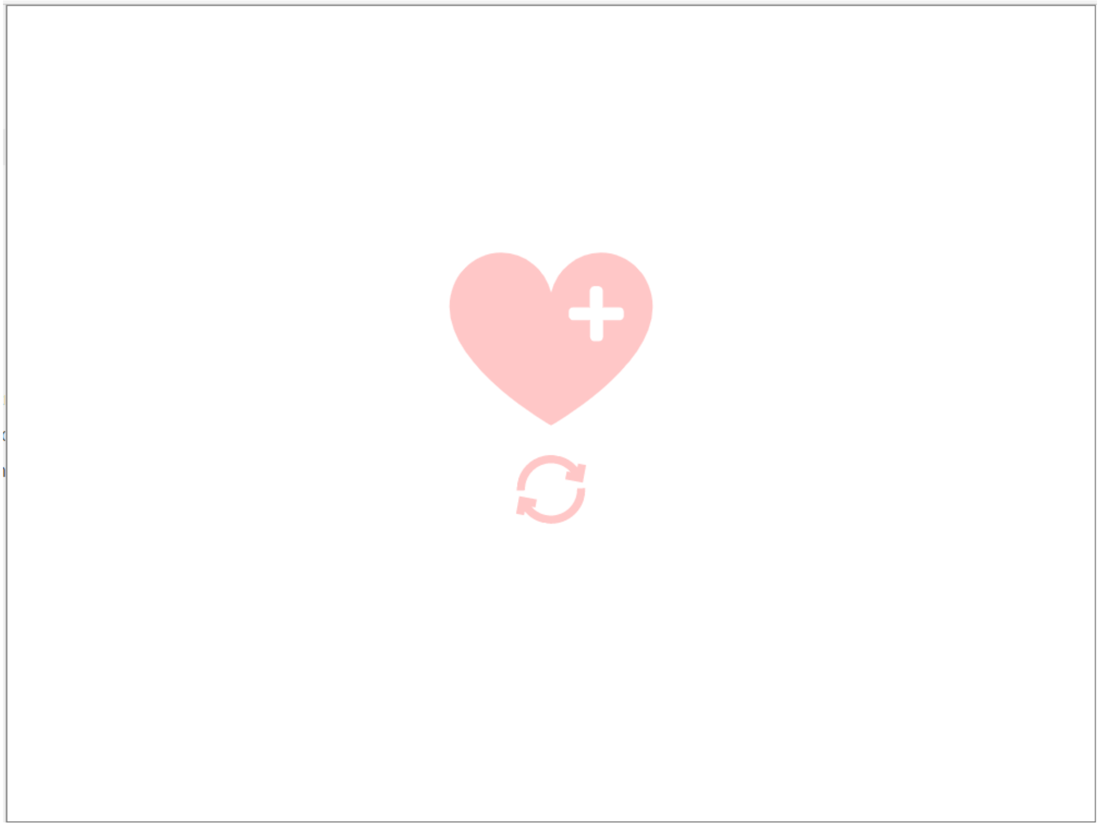


Figure 5.34 Loading Page

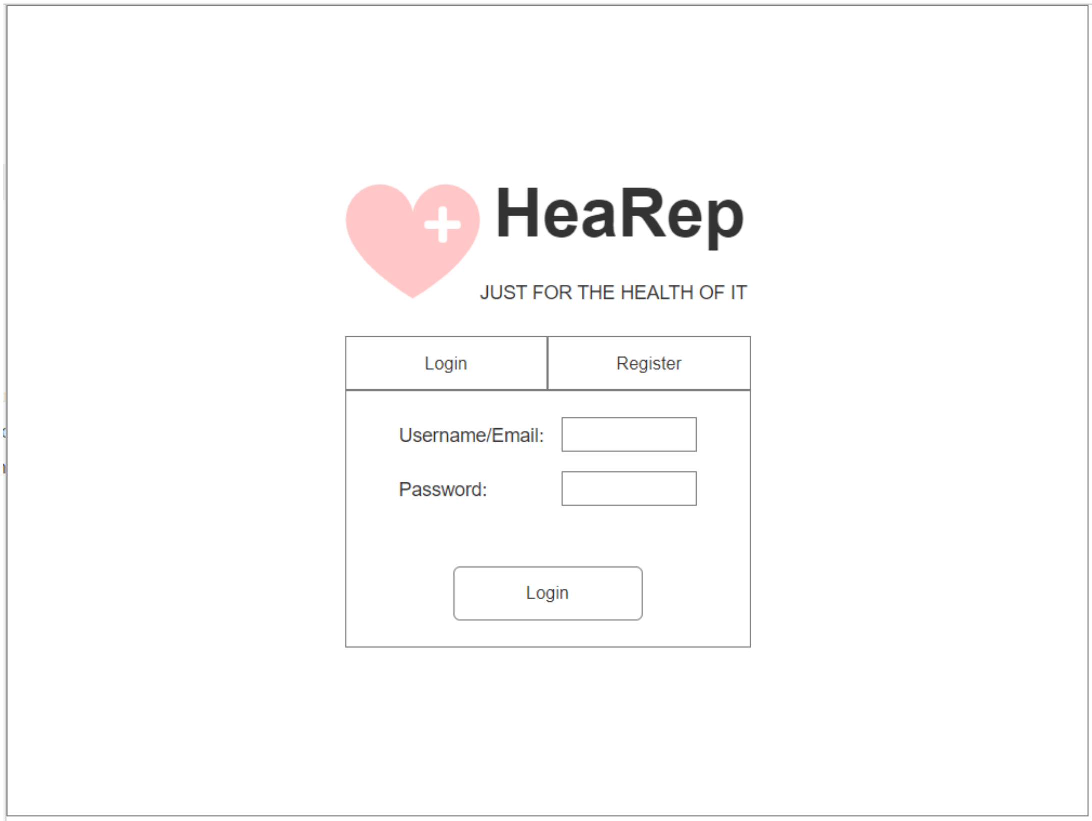


Figure 5.35 Login Conceptual Page

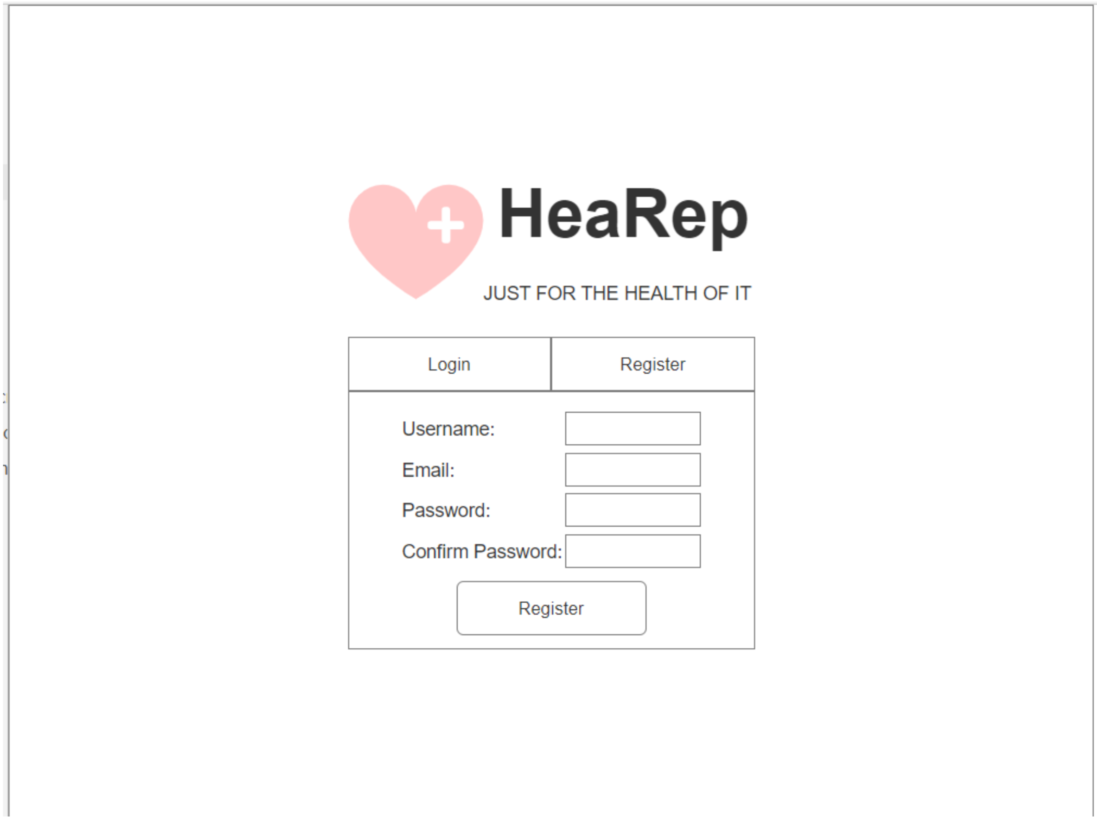
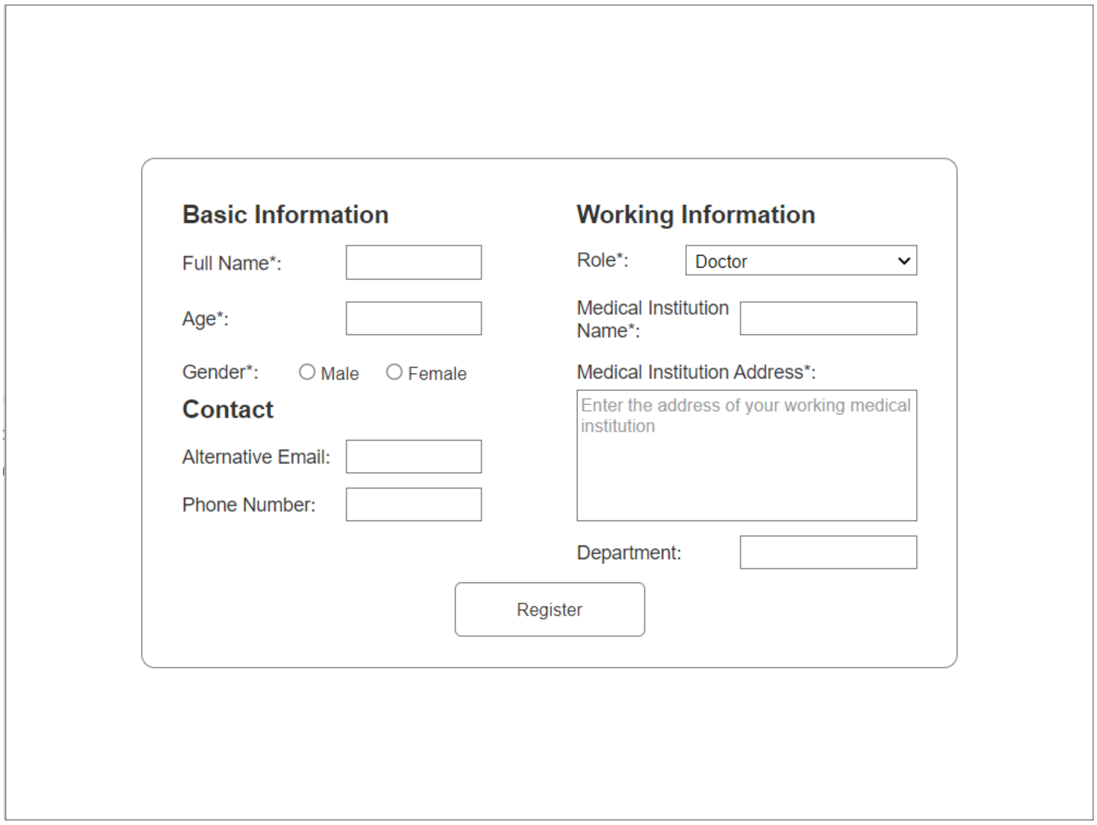


Figure 5.36 Register Conceptual Page



Basic Information

Full Name*:

Age*:

Gender*: ○ Male ○ Female

Contact

Alternative Email:

Phone Number:

Working Information

Role*: Doctor

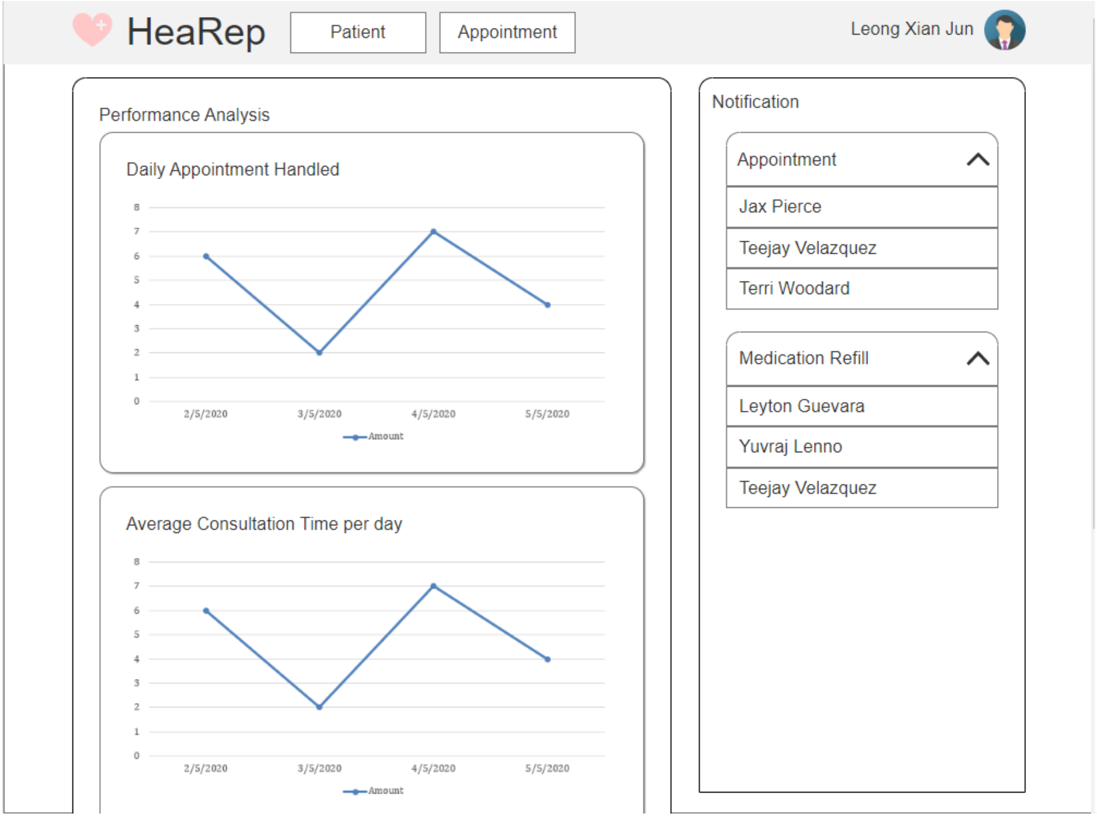
Medical Institution Name*:

Medical Institution Address*:

Enter the address of your working medical institution

Department:

Figure 5.37 Conceptual Design of Basic Information Page



Performance Analysis

Daily Appointment Handled

Average Consultation Time per day

Notification

Appointment

- Jax Pierce
- Teejay Velazquez
- Terri Woodard

Medication Refill

- Leyton Guevara
- Yuvraj Lenno
- Teejay Velazquez

Figure 5.38 Design of Dashboard Page

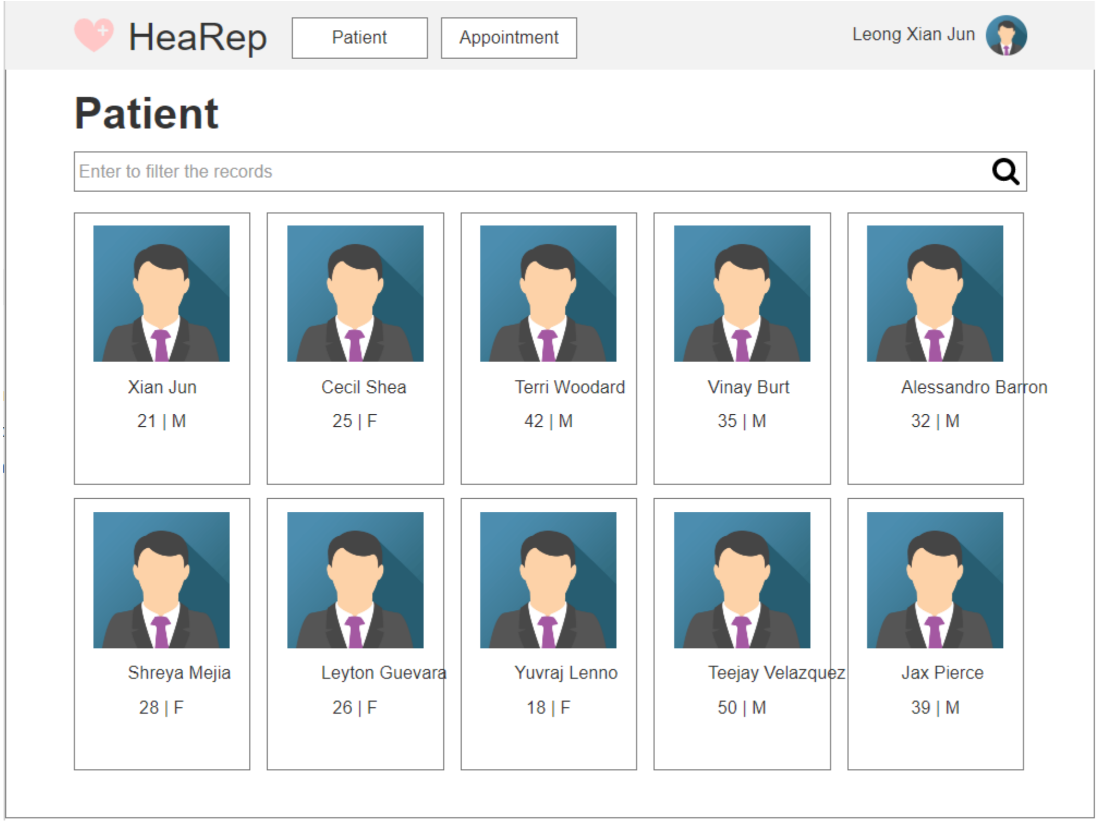
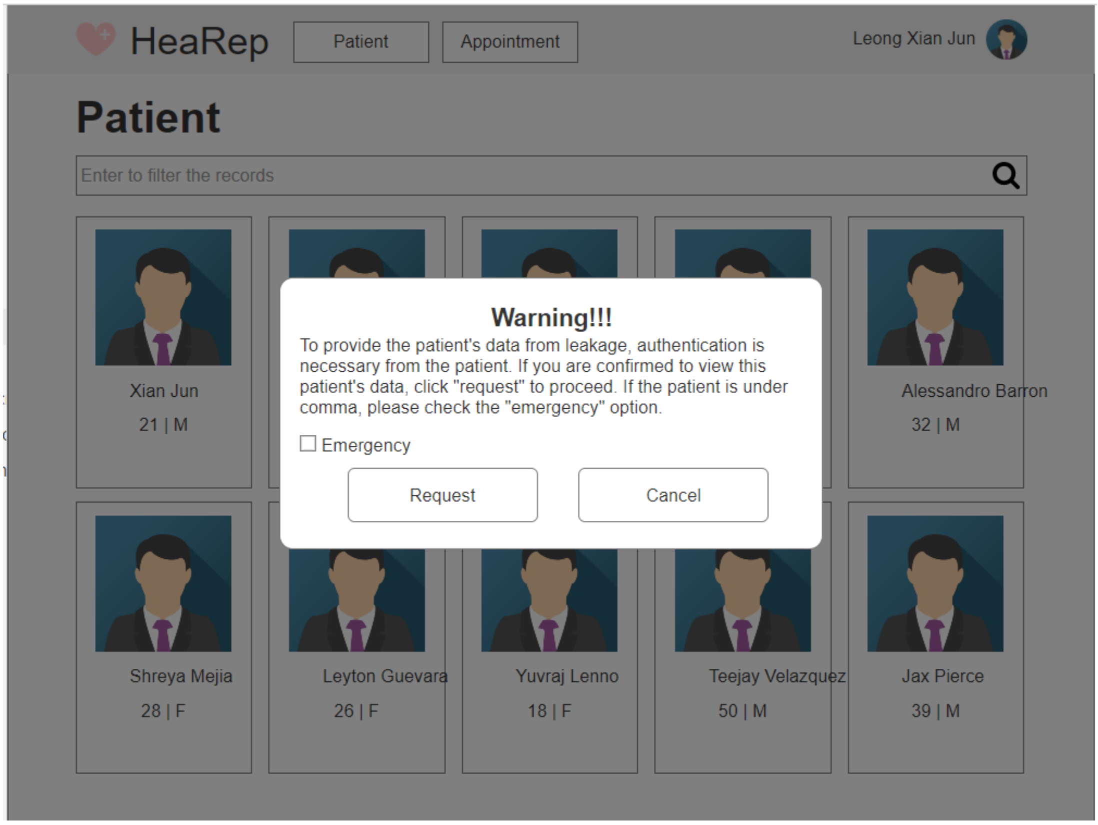


Figure 5.39 Page that lists all patients

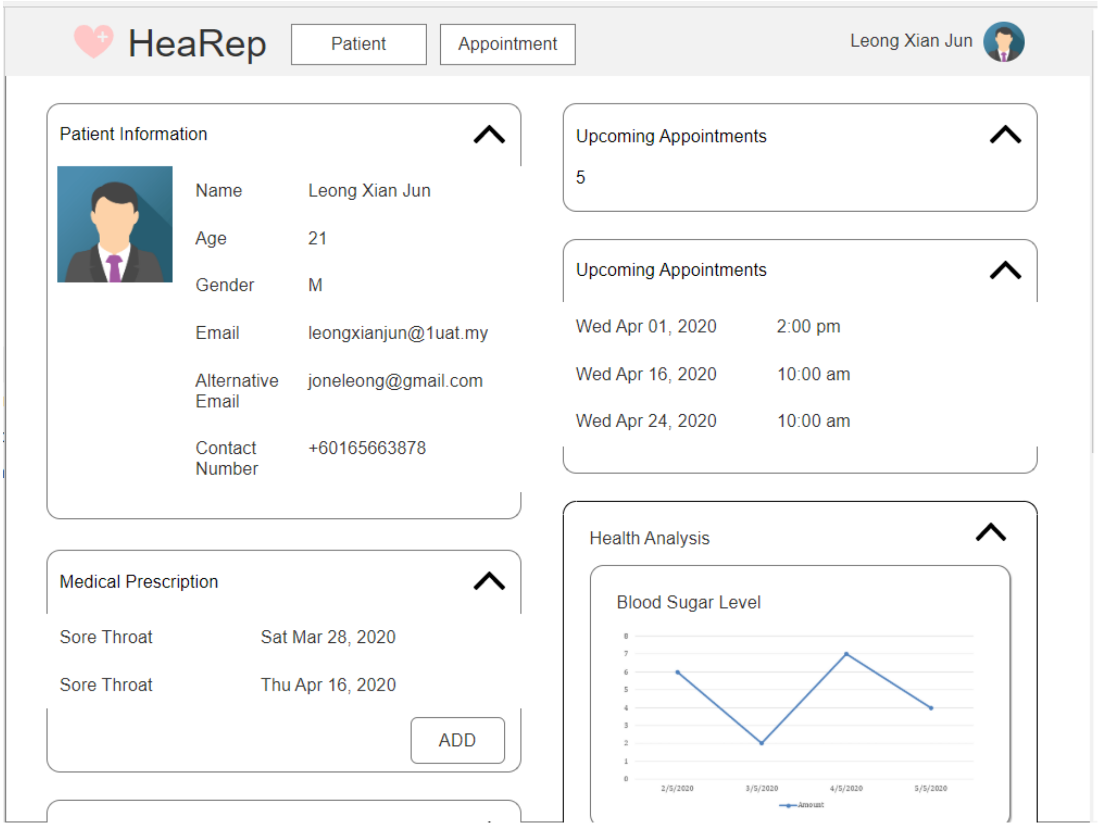


Warning!!!

To provide the patient's data from leakage, authentication is necessary from the patient. If you are confirmed to view this patient's data, click "request" to proceed. If the patient is under comma, please check the "emergency" option.

□ Emergency

Figure 5.40 Design of Access Request Dialog



Patient Information

Name	Leong Xian Jun
Age	21
Gender	M
Email	leongxianjun@1uat.my
Alternative Email	joneleong@gmail.com
Contact Number	+60165663878

Upcoming Appointments

5

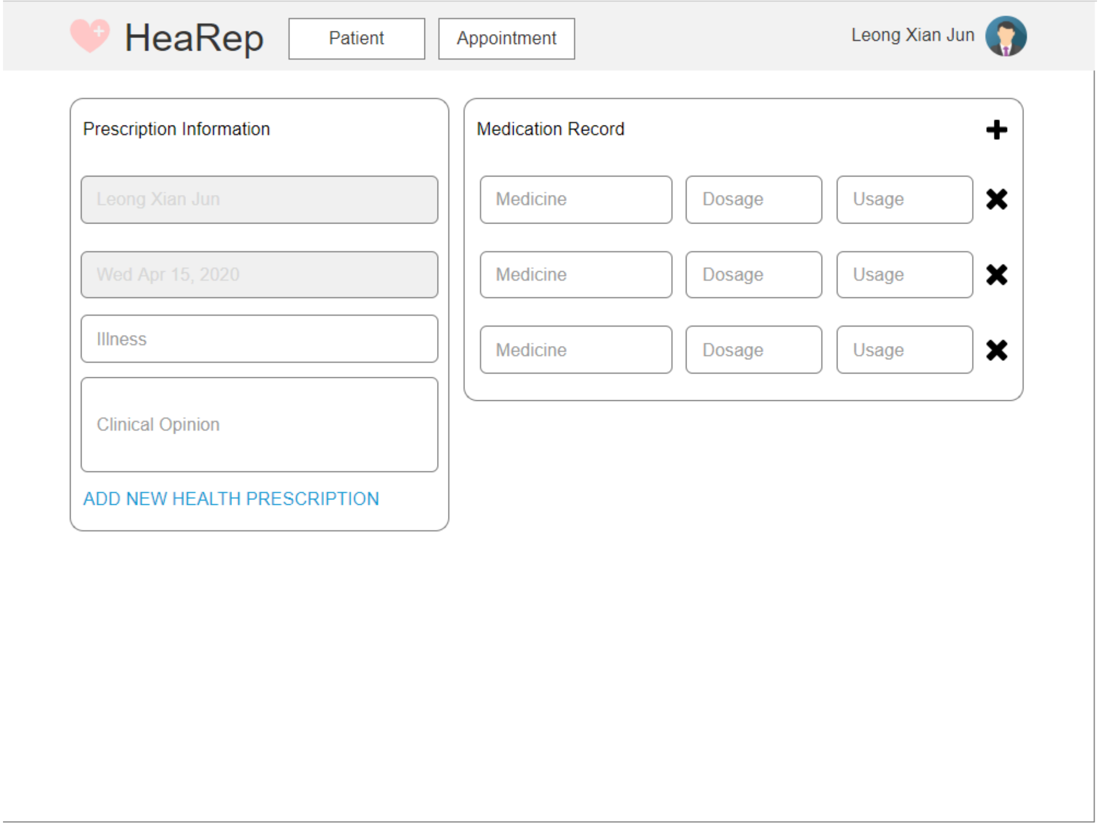
Upcoming Appointments

Wed Apr 01, 2020	2:00 pm
Wed Apr 16, 2020	10:00 am
Wed Apr 24, 2020	10:00 am

Medical Prescription

Sore Throat	Sat Mar 28, 2020
Sore Throat	Thu Apr 16, 2020

Figure 5.41 Design of Patient Detail



Prescription Information

Leong Xian Jun

Wed Apr 15, 2020

Illness

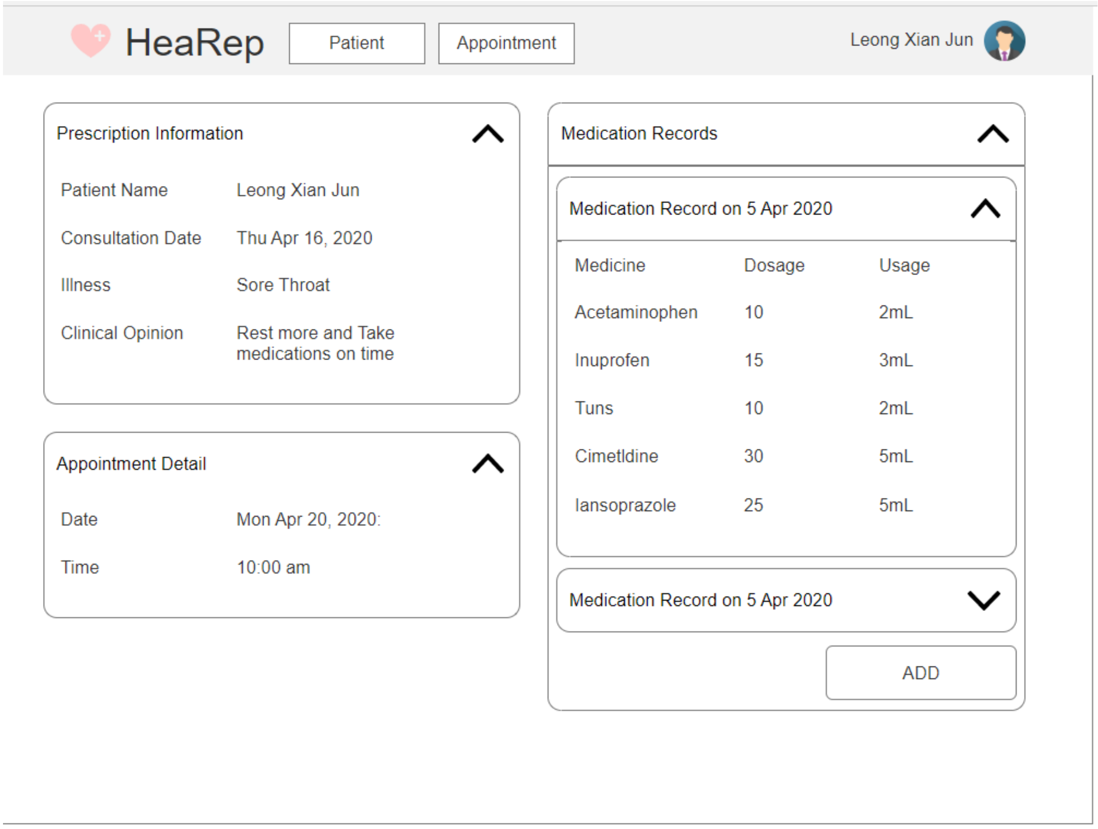
Clinical Opinion

ADD NEW HEALTH PRESCRIPTION

Medication Record

Medicine	Dosage	Usage
Medicine	Dosage	Usage
Medicine	Dosage	Usage

Figure 5.42 Design Page to Add new Health Prescription



Prescription Information

Patient Name	Leong Xian Jun
Consultation Date	Thu Apr 16, 2020
Illness	Sore Throat
Clinical Opinion	Rest more and Take medications on time

Appointment Detail

Date	Mon Apr 20, 2020:
Time	10:00 am

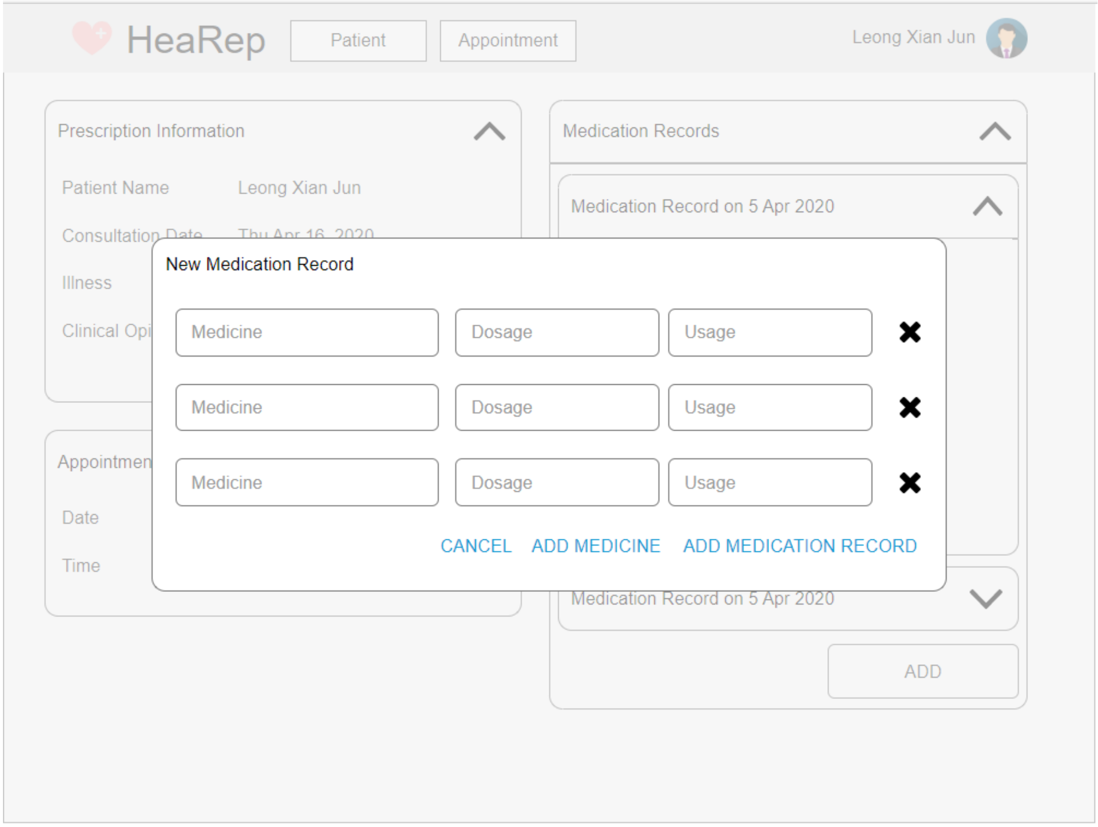
Medication Records

Medication Record on 5 Apr 2020

Medicine	Dosage	Usage
Acetaminophen	10	2mL
Inuprofen	15	3mL
Tuns	10	2mL
Cimetldine	30	5mL
lansoprazole	25	5mL

Medication Record on 5 Apr 2020

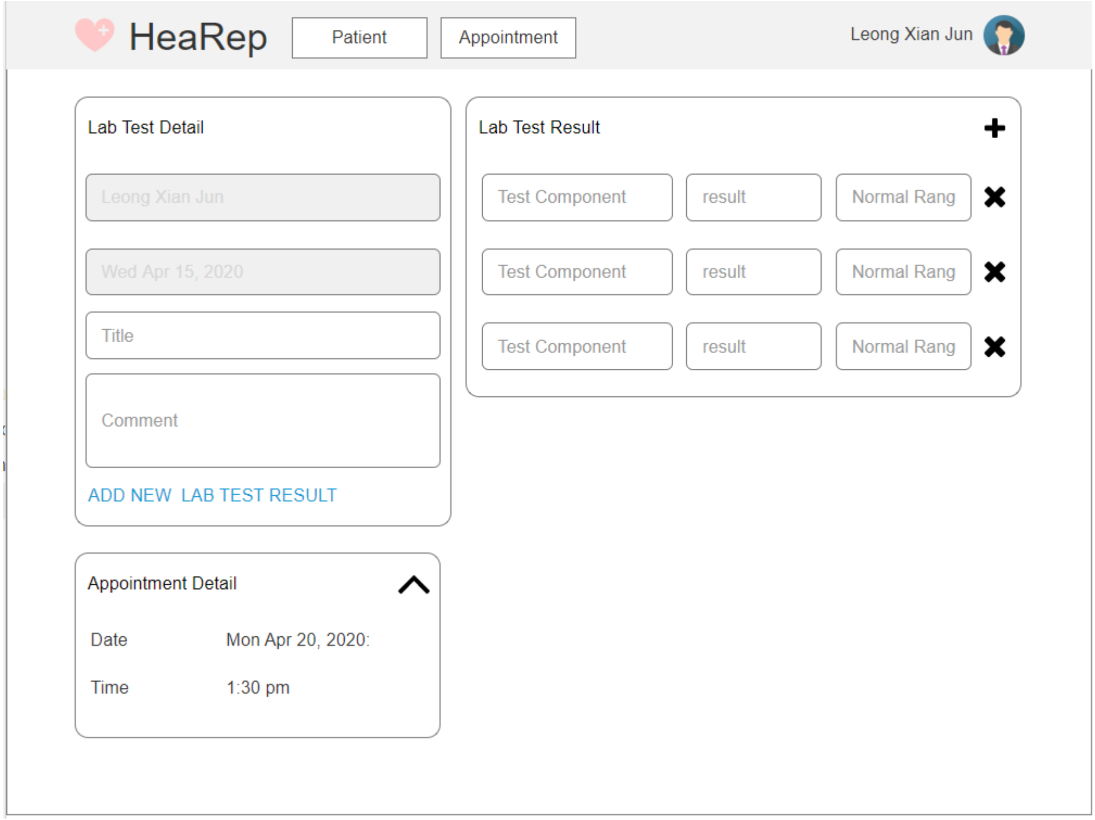
Figure 5.43 Detail Page of Health Prescription



New Medication Record

Medicine	Dosage	Usage	✖
Medicine	Dosage	Usage	✖
Medicine	Dosage	Usage	✖

Figure 5.44 Page to add new Medication Record



Lab Test Detail

Leong Xian Jun

Wed Apr 15, 2020

Title

Comment

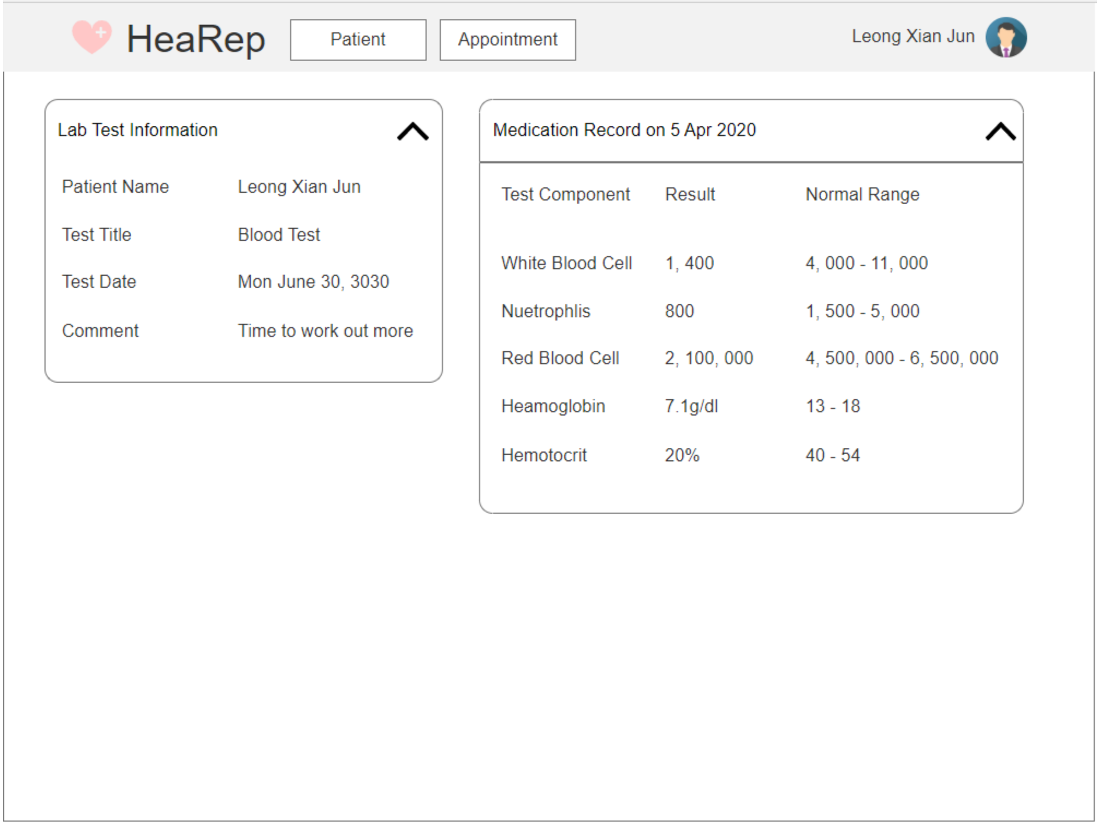
Lab Test Result

Test Component	result	Normal Rang
Test Component	result	Normal Rang
Test Component	result	Normal Rang

Appointment Detail

Date	Mon Apr 20, 2020:
Time	1:30 pm

Figure 5.45 Page to add new Lab Test Result



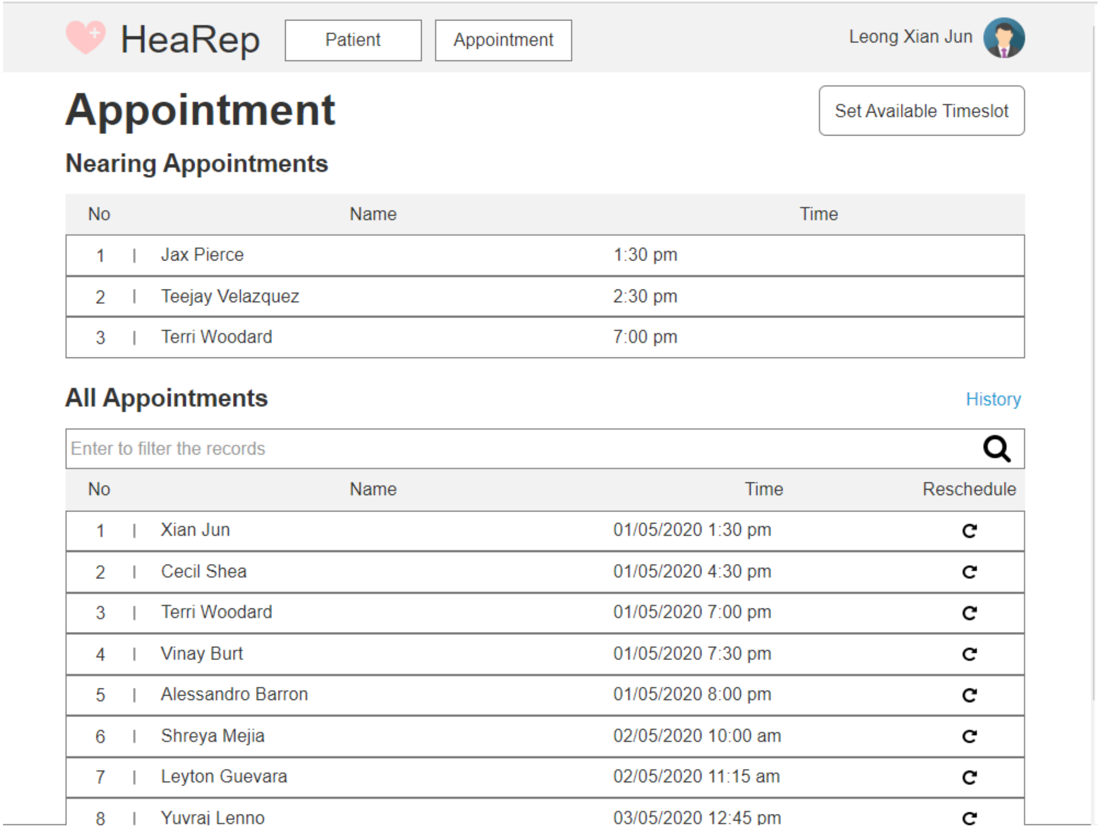
Lab Test Information

Patient Name	Leong Xian Jun
Test Title	Blood Test
Test Date	Mon June 30, 3030
Comment	Time to work out more

Medication Record on 5 Apr 2020

Test Component	Result	Normal Range
White Blood Cell	1, 400	4, 000 - 11, 000
Nuetrophlis	800	1, 500 - 5, 000
Red Blood Cell	2, 100, 000	4, 500, 000 - 6, 500, 000
Heamoglobin	7.1g/dl	13 - 18
Hemotocrit	20%	40 - 54

Figure 5.46 Detail Page of Lab Test Result



Appointment

Nearing Appointments

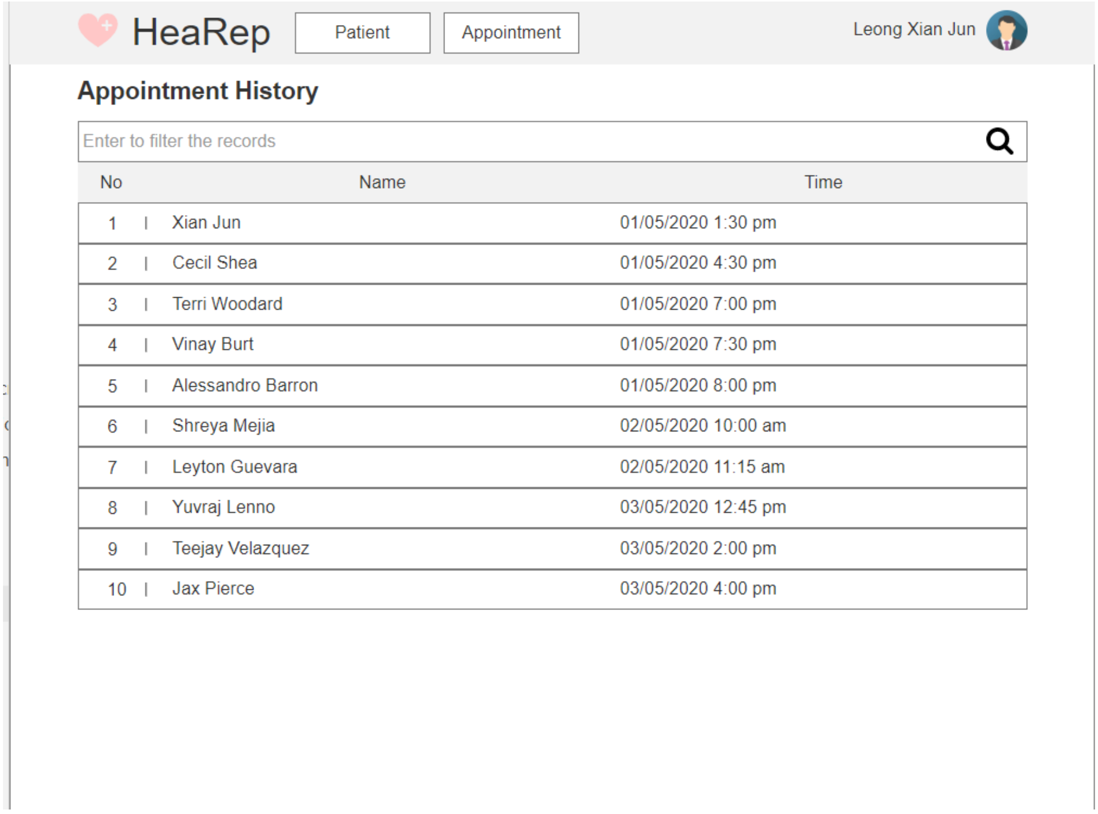
No	Name	Time
1	Jax Pierce	1:30 pm
2	Teejay Velazquez	2:30 pm
3	Terri Woodard	7:00 pm

All Appointments

Enter to filter the records

No	Name	Time	Reschedule
1	Xian Jun	01/05/2020 1:30 pm	↻
2	Cecil Shea	01/05/2020 4:30 pm	↻
3	Terri Woodard	01/05/2020 7:00 pm	↻
4	Vinay Burt	01/05/2020 7:30 pm	↻
5	Alessandro Barron	01/05/2020 8:00 pm	↻
6	Shreya Mejia	02/05/2020 10:00 am	↻
7	Leyton Guevara	02/05/2020 11:15 am	↻
8	Yuvraj Lenno	03/05/2020 12:45 pm	↻

Figure 5.47 Page of Appointment List

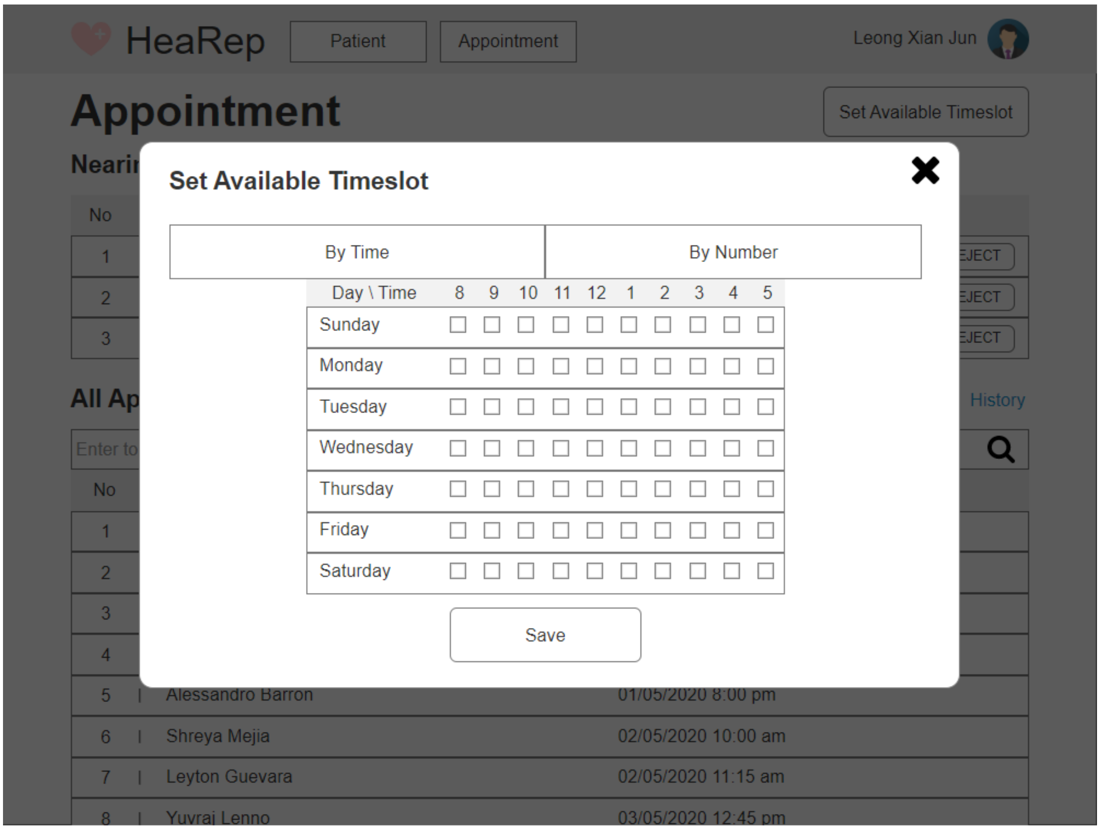


Appointment History

Enter to filter the records

No	Name	Time
1	Xian Jun	01/05/2020 1:30 pm
2	Cecil Shea	01/05/2020 4:30 pm
3	Terri Woodard	01/05/2020 7:00 pm
4	Vinay Burt	01/05/2020 7:30 pm
5	Alessandro Barron	01/05/2020 8:00 pm
6	Shreya Mejia	02/05/2020 10:00 am
7	Leyton Guevara	02/05/2020 11:15 am
8	Yuvraj Lenno	03/05/2020 12:45 pm
9	Teejay Velazquez	03/05/2020 2:00 pm
10	Jax Pierce	03/05/2020 4:00 pm

Figure 5.48 Page of Appointment History List

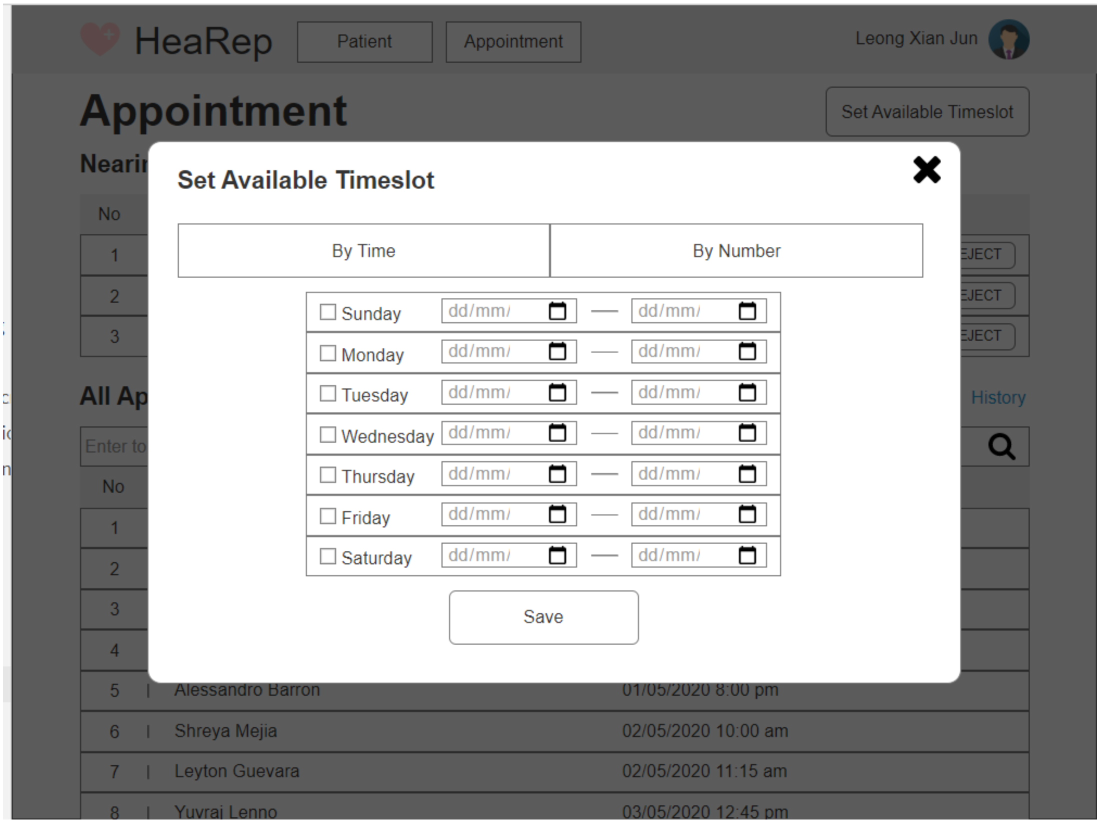


Set Available Timeslot

Day \ Time	8	9	10	11	12	1	2	3	4	5
Sunday	□	□	□	□	□	□	□	□	□	□
Monday	□	□	□	□	□	□	□	□	□	□
Tuesday	□	□	□	□	□	□	□	□	□	□
Wednesday	□	□	□	□	□	□	□	□	□	□
Thursday	□	□	□	□	□	□	□	□	□	□
Friday	□	□	□	□	□	□	□	□	□	□
Saturday	□	□	□	□	□	□	□	□	□	□

No		
5	Alessandro Barron	01/05/2020 8:00 pm
6	Shreya Mejia	02/05/2020 10:00 am
7	Leyton Guevara	02/05/2020 11:15 am
8	Yuvraj Lenno	03/05/2020 12:45 pm

Figure 5.49 Page to set available by time timeslots

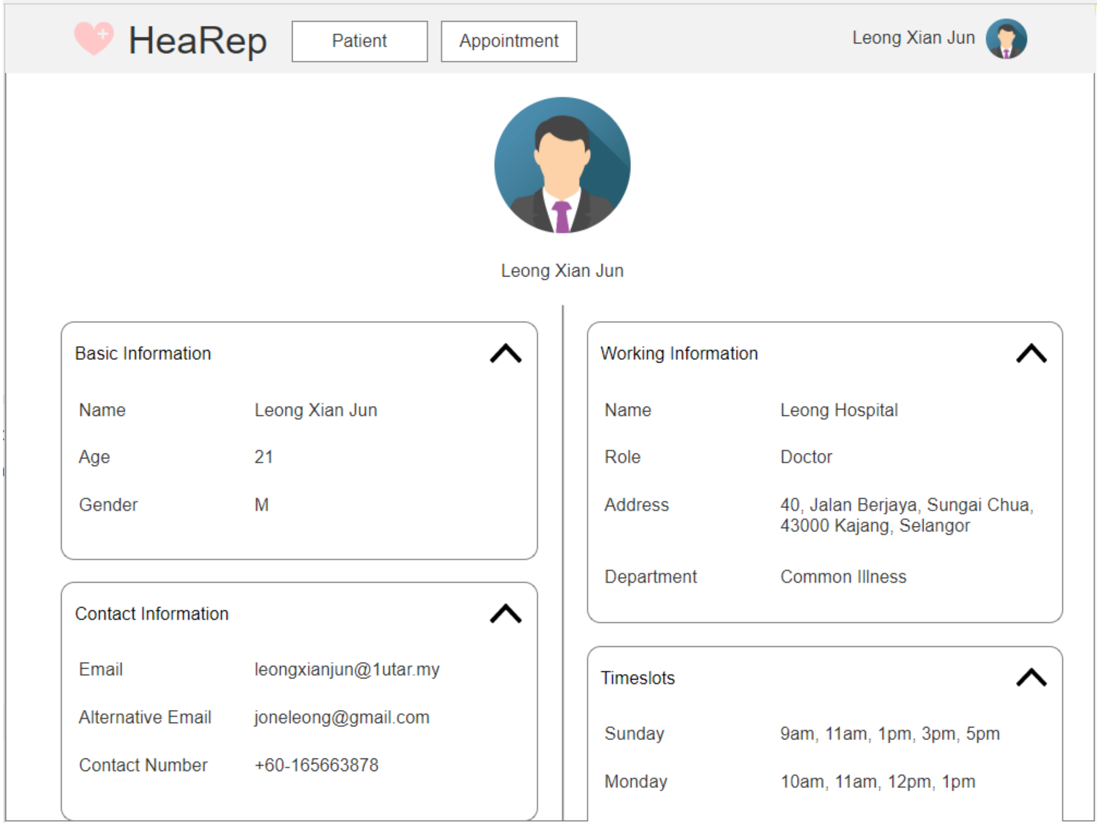


Set Available Timeslot

□ Sunday	dd/mm/	—	dd/mm/
□ Monday	dd/mm/	—	dd/mm/
□ Tuesday	dd/mm/	—	dd/mm/
□ Wednesday	dd/mm/	—	dd/mm/
□ Thursday	dd/mm/	—	dd/mm/
□ Friday	dd/mm/	—	dd/mm/
□ Saturday	dd/mm/	—	dd/mm/

No		
5	Alessandro Barron	01/05/2020 8:00 pm
6	Shreya Mejia	02/05/2020 10:00 am
7	Leyton Guevara	02/05/2020 11:15 am
8	Yuvraj Lenno	03/05/2020 12:45 pm

Figure 5.50 Page to set available by number timeslots



Leong Xian Jun

Basic Information

Name	Leong Xian Jun
Age	21
Gender	M

Contact Information

Email	leongxianjun@1utar.my
Alternative Email	joneleong@gmail.com
Contact Number	+60-165663878

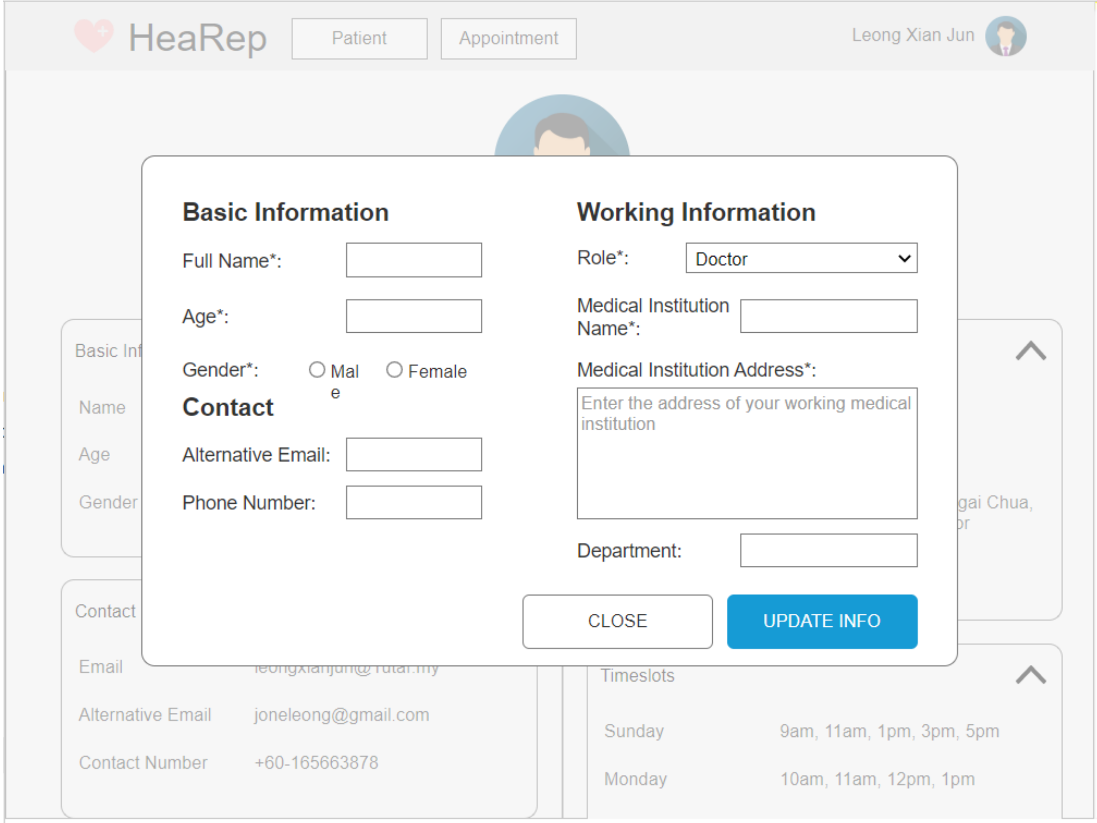
Working Information

Name	Leong Hospital
Role	Doctor
Address	40, Jalan Berjaya, Sungai Chua, 43000 Kajang, Selangor
Department	Common Illness

Timeslots

Sunday	9am, 11am, 1pm, 3pm, 5pm
Monday	10am, 11am, 12pm, 1pm

Figure 5.51 Design Page of Profile Detail



Basic Information

Full Name*:

Age*:

Gender*: ○ Mal e ○ Female

Contact

Alternative Email:

Phone Number:

Working Information

Role*: Doctor

Medical Institution Name*:

Medical Institution Address*:

Enter the address of your working medical institution

Department:

Figure 5.52 Design Page to update Profile

5.5. High-Fidelity Prototype

The following are the initial layout and design of the application for both mobile and web application. The further enhancement will be conducted to improve the design of the UI so that the users can have a better experience in using the system.

5.5.1. Initial System Design for Mobile Application

In this section, a page navigation flow of the mobile application is shown. This page navigation flow indicates the criteria or element for the application users to navigate between the pages. Additionally, the actual initial UI design is shown in section 5.5.1.2.

5.5.1.1. Screen Navigation flow of Mobile Application

The screen navigation flow shows the necessary criteria for the user to navigate between the screen in the mobile application.

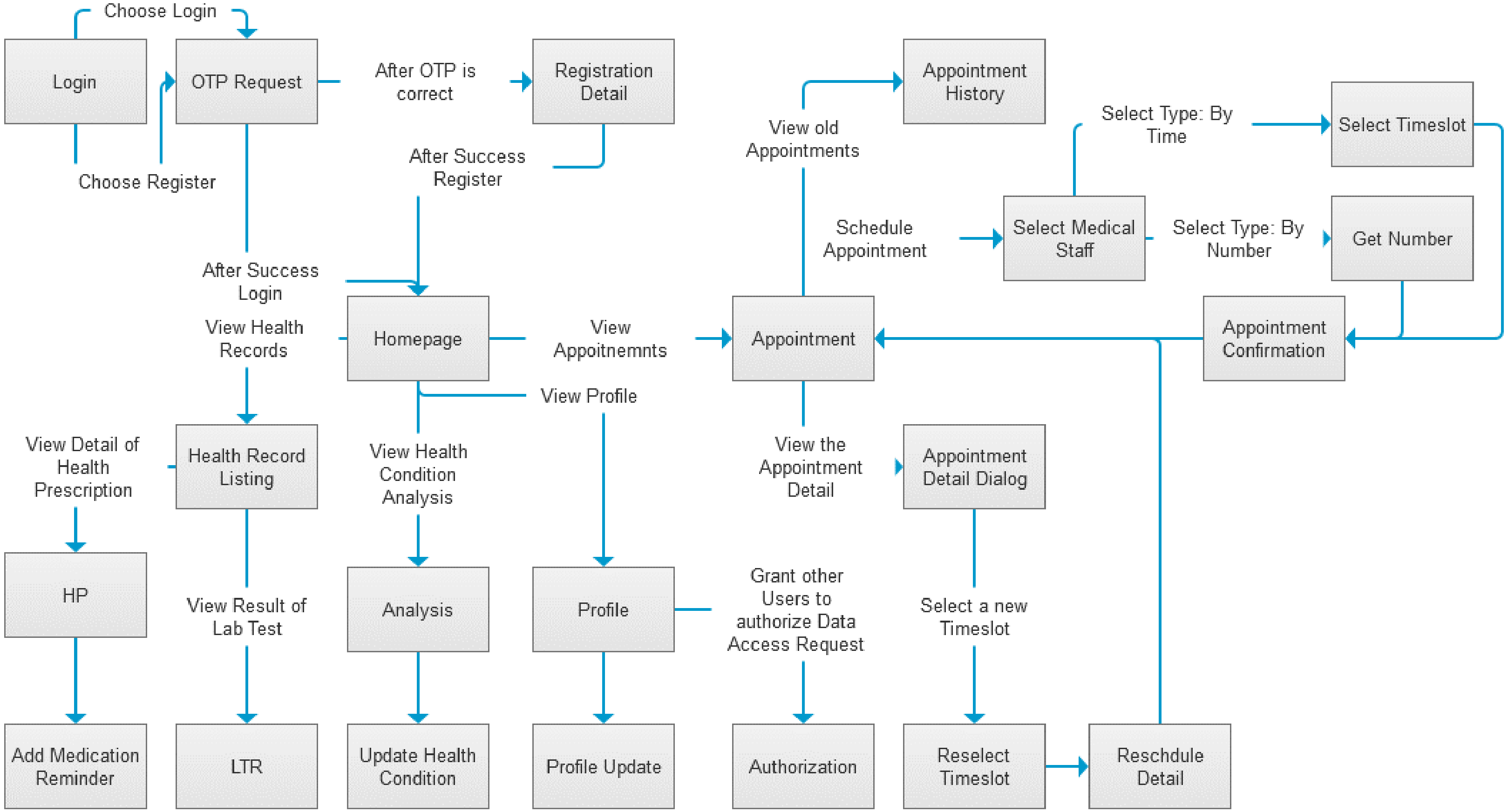
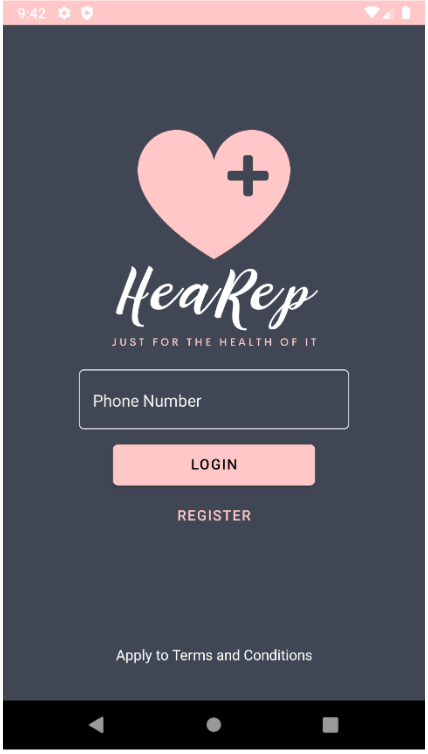


Figure 5.53 Page Navigation flow in Mobile Application

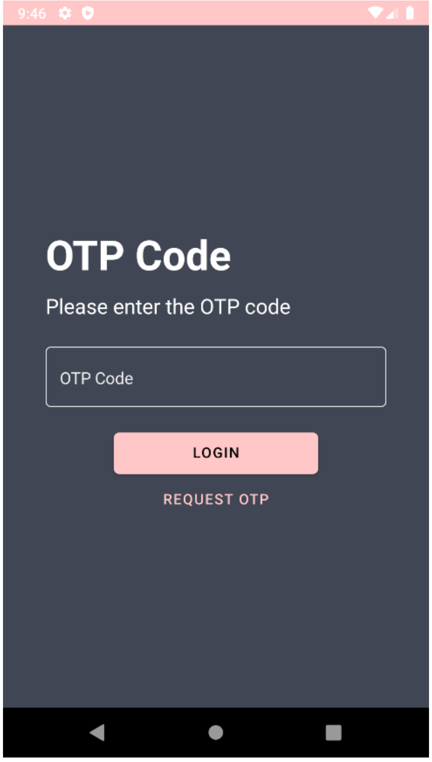
5.5.1.2. Mobile Application UI Design

5.5.1.2.1. Account Creation



Apply to Terms and Conditions

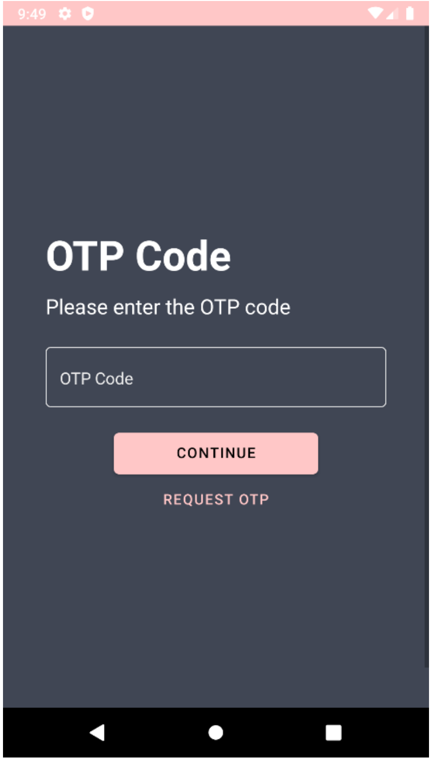
Figure 5.54 Login Page of the Mobile Application (Login)



OTP Code

Please enter the OTP code

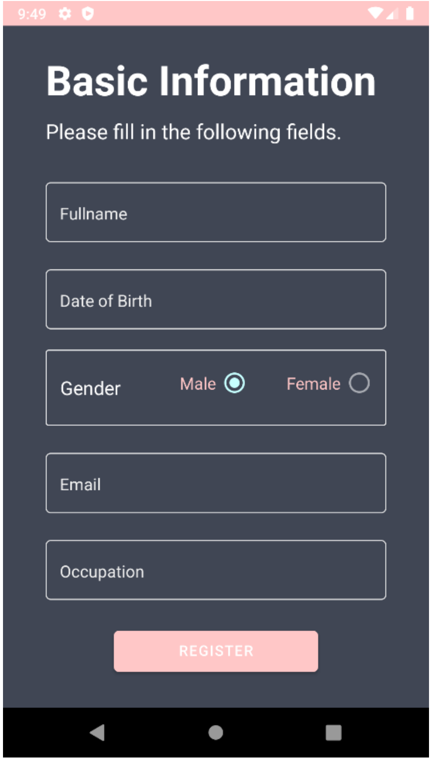
Figure 5.55 OTP Request Page (OTP Request)



OTP Code

Please enter the OTP code

Figure 5.56 OTP Request Page (OTP Request)

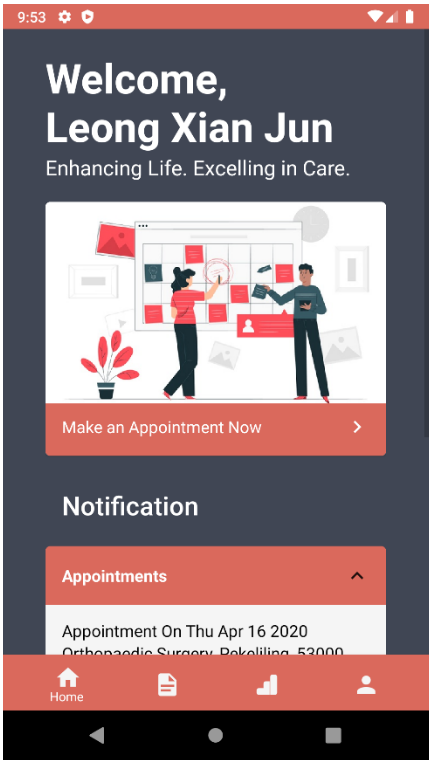


Basic Information

Please fill in the following fields.

Gender Male ● Female ○

Figure 5.57 Registration Page (Registration Detail)



Welcome,
Leong Xian Jun

Enhancing Life. Excelling in Care.

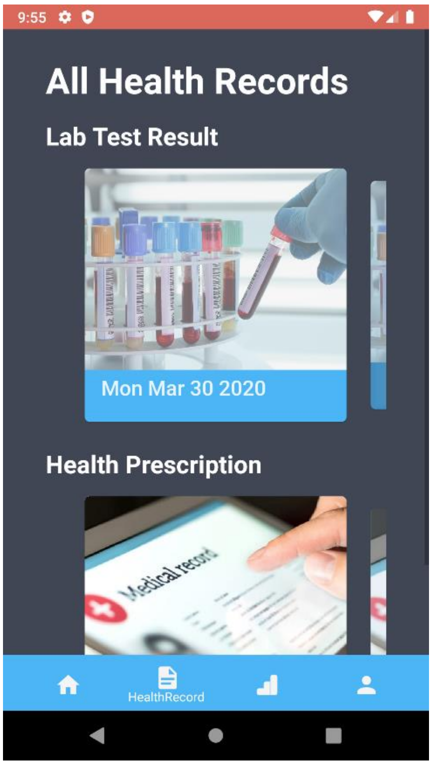
Notification

Appointments

Appointment On Thu Apr 16 2020
Orthopaedic Surgery, Pekeliling, 53000

Figure 5.58 Homepage after login or register (Homepage)

5.5.1.2.2. Health Records Tracking



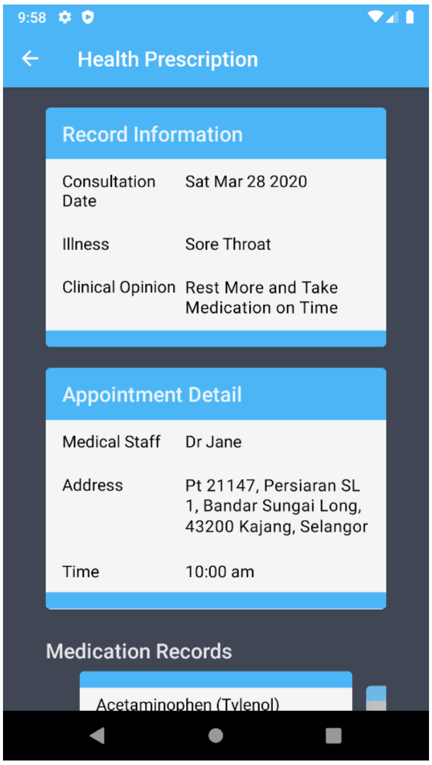
All Health Records

Lab Test Result

Mon Mar 30 2020

Health Prescription

Figure 5.59 Page that lists all of the Health Records (Health Record Listing)



9:58

Health Prescription

Record Information

Consultation Date	Sat Mar 28 2020
Illness	Sore Throat
Clinical Opinion	Rest More and Take Medication on Time

Appointment Detail

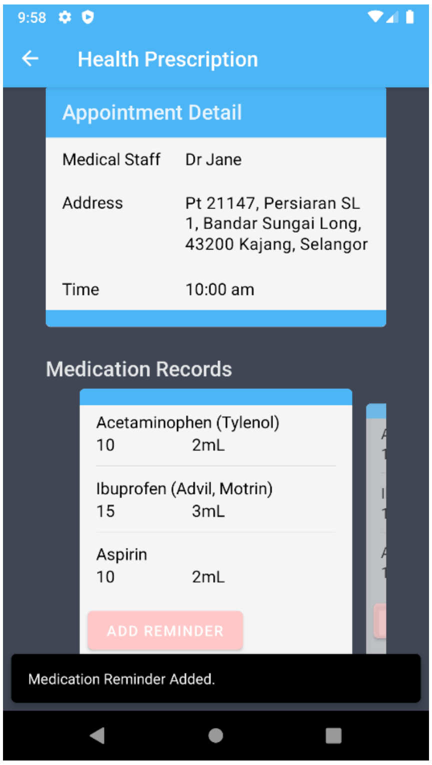
Medical Staff	Dr Jane
Address	Pt 21147, Persiaran SL 1, Bandar Sungai Long, 43200 Kajang, Selangor
Time	10:00 am

Medication Records

Acetaminophen (Tylenol)

Figure 5.60 Health Prescription Detail (HP)

5.5.1.2.2.1. Medication Reminder



9:58

Health Prescription

Appointment Detail

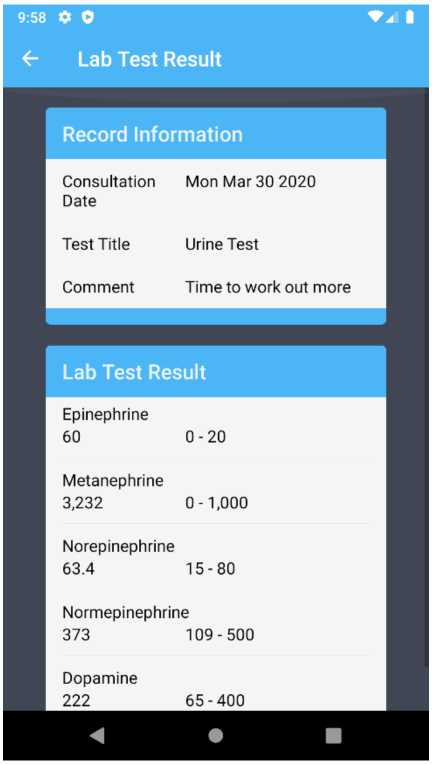
Medical Staff	Dr Jane
Address	Pt 21147, Persiaran SL 1, Bandar Sungai Long, 43200 Kajang, Selangor
Time	10:00 am

Medication Records

Acetaminophen (Tylenol)	
10	2mL
Ibuprofen (Advil, Motrin)	
15	3mL
Aspirin	
10	2mL

Medication Reminder Added.

Figure 5.61 Snack bar after medication reminder added (Add Medication Reminder)



Record Information

Consultation Date	Mon Mar 30 2020
Test Title	Urine Test
Comment	Time to work out more

Lab Test Result

Epinephrine
60 0 - 20

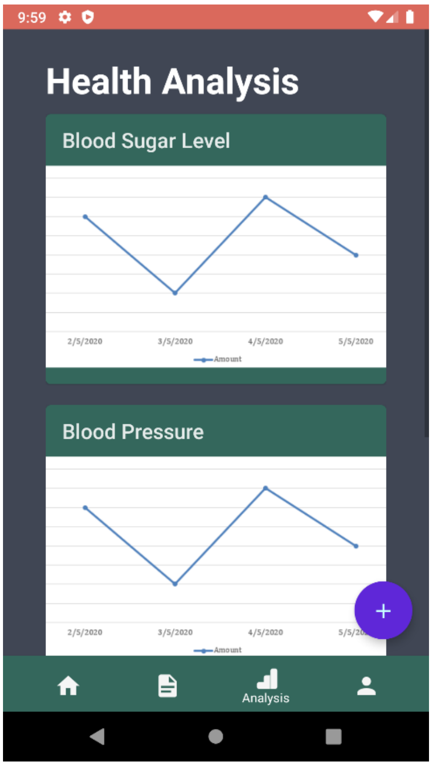
Metanephrine
3,232 0 - 1,000

Norepinephrine
63.4 15 - 80

Normepinephrine
373 109 - 500

Dopamine
222 65 - 400

Figure 5.62 Lab Test Result (LTR)

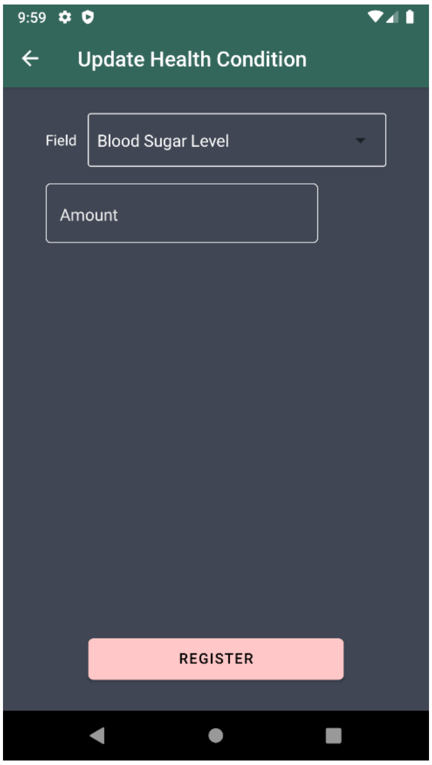


Health Analysis

Blood Sugar Level

Blood Pressure

Figure 5.63 Analysis of Health Condition (Analysis)

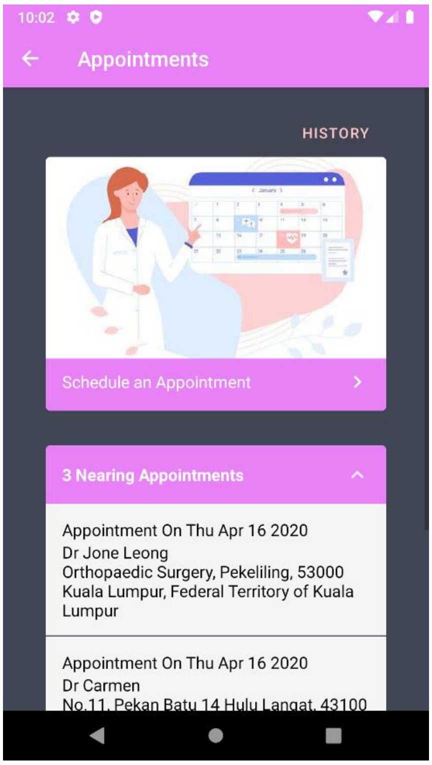


Field: Blood Sugar Level

Amount

Figure 5.64 Page to Update Health Condition (Update Health Condition)

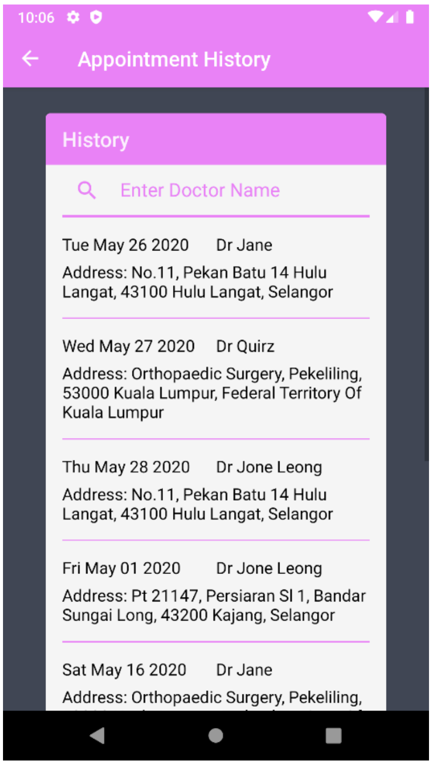
5.5.1.2.3. Appointment Scheduling



Appointment On Thu Apr 16 2020
Dr Jone Leong
Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory of Kuala Lumpur

Appointment On Thu Apr 16 2020
Dr Carmen
No.11, Pekan Batu 14 Hulu Langat, 43100

Figure 5.65 Appointment List (Appointment)



Tue May 26 2020 Dr Jane
Address: No.11, Pekan Batu 14 Hulu Langat, 43100 Hulu Langat, Selangor

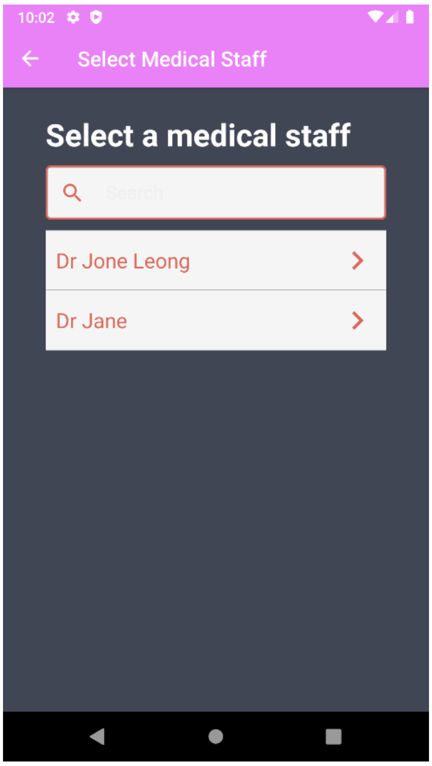
Wed May 27 2020 Dr Quirz
Address: Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory Of Kuala Lumpur

Thu May 28 2020 Dr Jone Leong
Address: No.11, Pekan Batu 14 Hulu Langat, 43100 Hulu Langat, Selangor

Fri May 01 2020 Dr Jone Leong
Address: Pt 21147, Persiaran Sl 1, Bandar Sungai Long, 43200 Kajang, Selangor

Sat May 16 2020 Dr Jane
Address: Orthopaedic Surgery, Pekeliling,

Figure 5.66 Appointment History (Appointment History)



Dr Jone Leong

Dr Jane

Figure 5.67 Page to select a medical staff (Select Medical Staff)

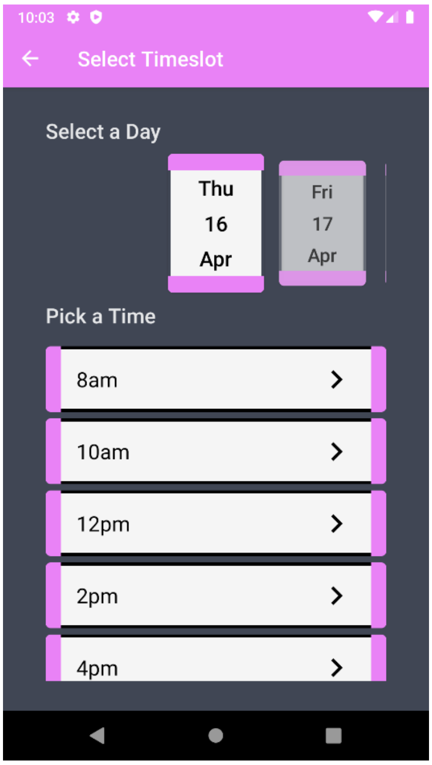
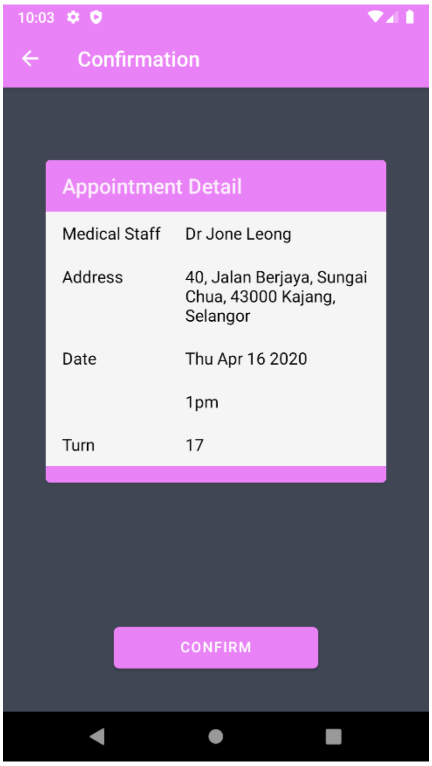


Figure 5.68 Page to select a timeslot (Select Timeslot)



Appointment Detail

Medical Staff	Dr Jone Leong
Address	40, Jalan Berjaya, Sungai Chua, 43000 Kajang, Selangor
Date	Thu Apr 16 2020
	1pm
Turn	17

Figure 5.69 Confirmation Page (Appointment Confirmation)

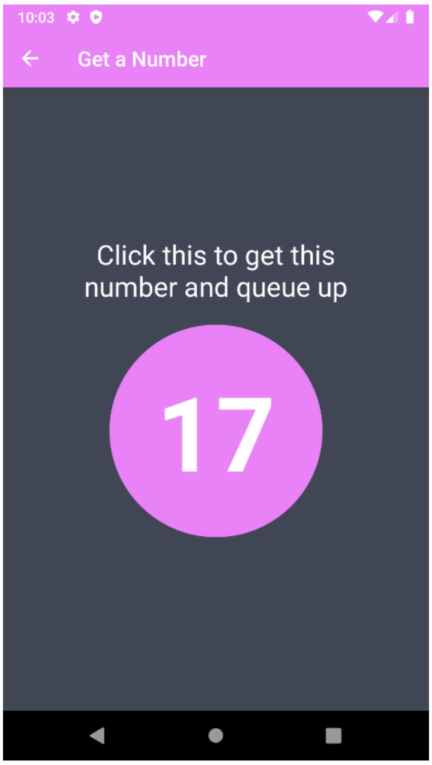
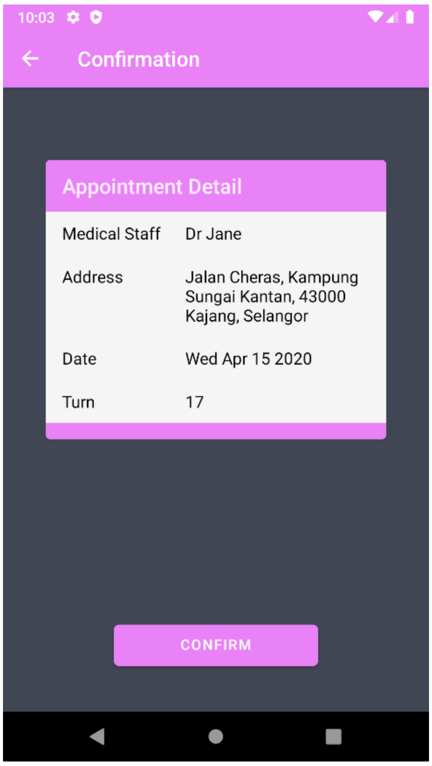
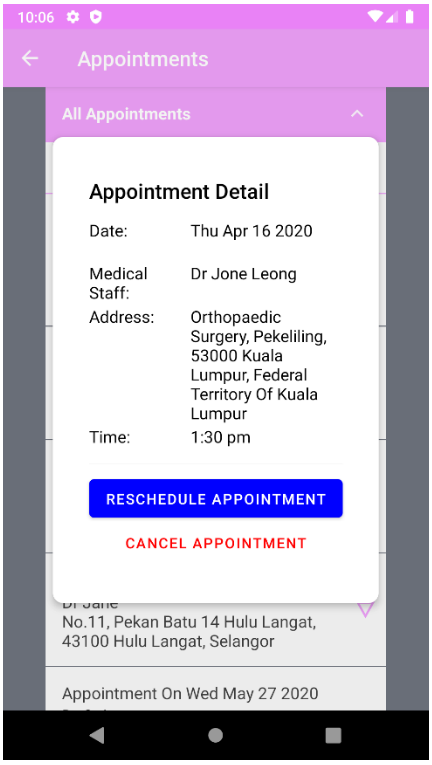


Figure 5.70 Page to get a number (Get Number)



Medical Staff	Dr Jane
Address	Jalan Cheras, Kampung Sungai Kantan, 43000 Kajang, Selangor
Date	Wed Apr 15 2020
Turn	17

Figure 5.71 Appointment Confirmation Page (Appointment Confirmation)



10:06

Appointments

All Appointments

Appointment Detail

Date: Thu Apr 16 2020

Medical Staff: Dr Jone Leong

Address: Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory Of Kuala Lumpur

Time: 1:30 pm

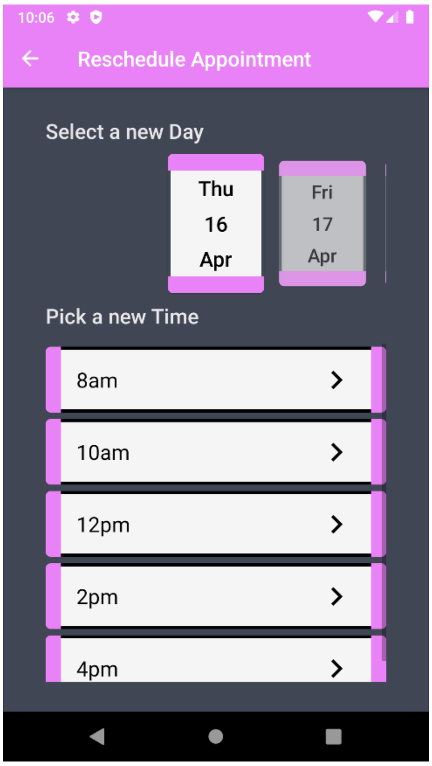
RESCHEDULE APPOINTMENT

CANCEL APPOINTMENT

Dr Jane
No.11, Pekan Batu 14 Hulu Langat, 43100 Hulu Langat, Selangor

Appointment On Wed May 27 2020

Figure 5.72 Appointment Detail (by time) (Appointment Detail Dialog)



10:06

Reschedule Appointment

Select a new Day

Thu 16 Apr

Fri 17 Apr

Pick a new Time

8am

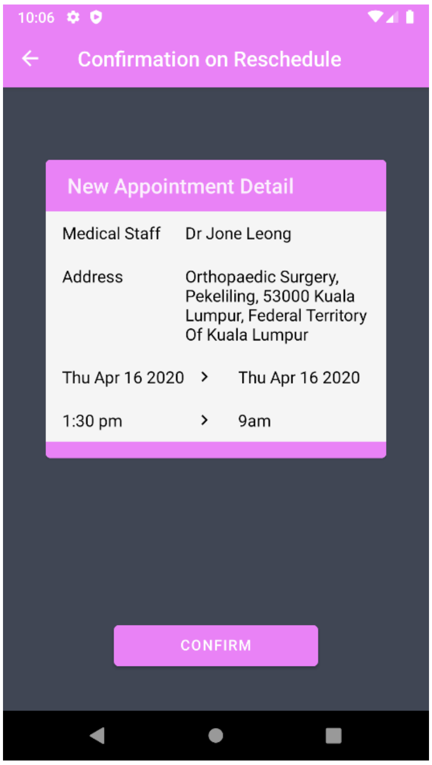
10am

12pm

2pm

4pm

Figure 5.73 Reselect a new Timeslot (Reselect Timeslot)



Confirmation on Reschedule

New Appointment Detail

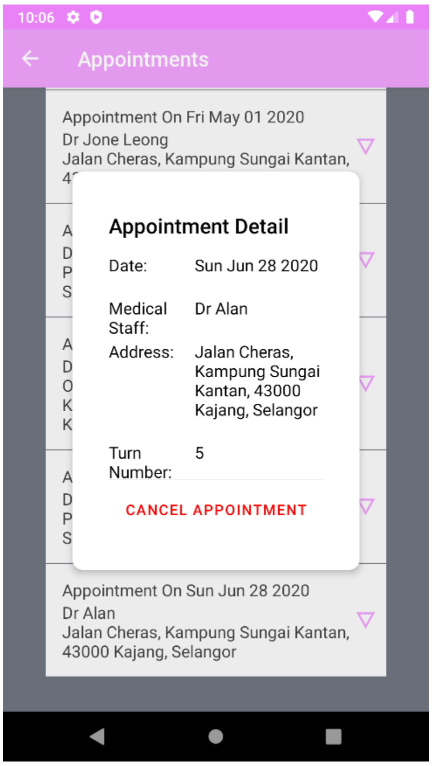
Medical Staff Dr Jone Leong

Address Orthopaedic Surgery, Pekeliling, 53000 Kuala Lumpur, Federal Territory Of Kuala Lumpur

Thu Apr 16 2020 > Thu Apr 16 2020

1:30 pm > 9am

Figure 5.74 Confirmation on the reschedule appointment (Reschedule Detail)



Appointments

Appointment On Fri May 01 2020
Dr Jone Leong
Jalan Cheras, Kampung Sungai Kantan,

Appointment Detail

Date: Sun Jun 28 2020

Medical Staff: Dr Alan

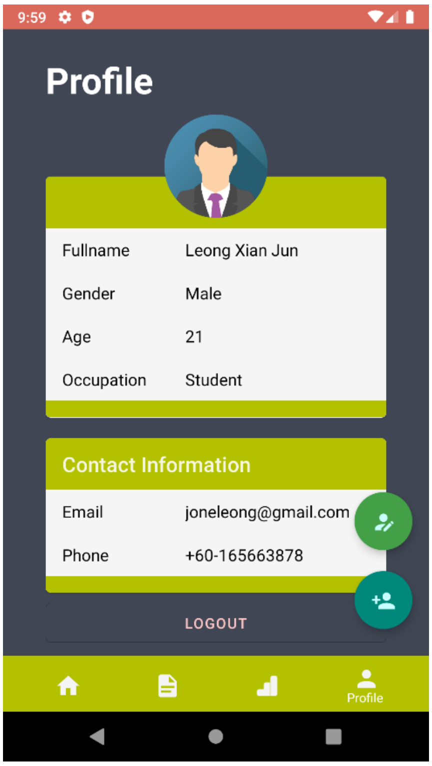
Address: Jalan Cheras, Kampung Sungai Kantan, 43000 Kajang, Selangor

Turn Number: 5

Appointment On Sun Jun 28 2020
Dr Alan
Jalan Cheras, Kampung Sungai Kantan, 43000 Kajang, Selangor

Figure 5.75 Appointment Detail (by number) (Appointment Detail Dialog)

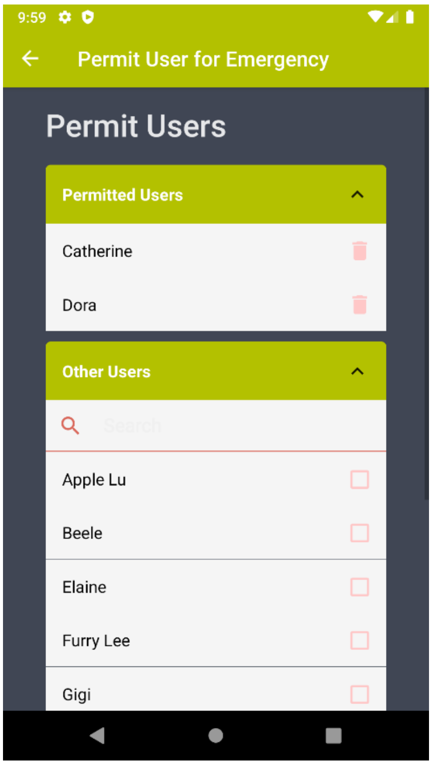
5.5.1.2.4. Health Records Access by Patients



Fullname Leong Xian Jun
Gender Male
Age 21
Occupation Student

Email joneleong@gmail.com
Phone +60-165663878

Figure 5.76 Profile Detail (Profile)

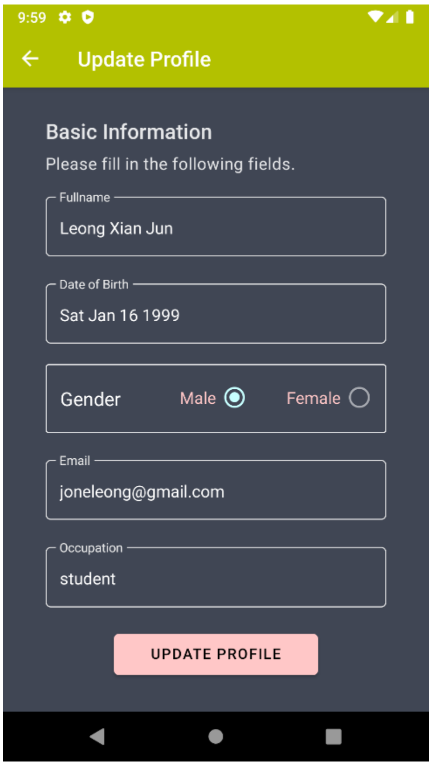


Permitted Users
- Catherine
- Dora

Other Users
- Apple Lu □
- Beele □
- Elaine □
- Furry Lee □
- Gigi □

Figure 5.77 Page to grant other users to authorize access request (Authorization)

5.5.1.2.5. Account Management



9:59

Update Profile

Basic Information

Please fill in the following fields.

Fullname: Leong Xian Jun

Date of Birth: Sat Jan 16 1999

Gender: Male ● Female ○

Email: joneleong@gmail.com

Occupation: student

Figure 5.78 Page to Update Profile (Profile Update)

5.5.2. Initial System Design for Web Application

In this section, it begins with the page navigation flow of the web application. The page navigation flow will not show the initial design and shall refer to as section 5.5.2.2. The page navigation flow also indicates the criteria to navigate between the pages in the web application.

5.5.2.1. Page Navigation flow of Web Application

The page navigation flow shows the necessary criteria for the user to navigate between the pages in the web application.

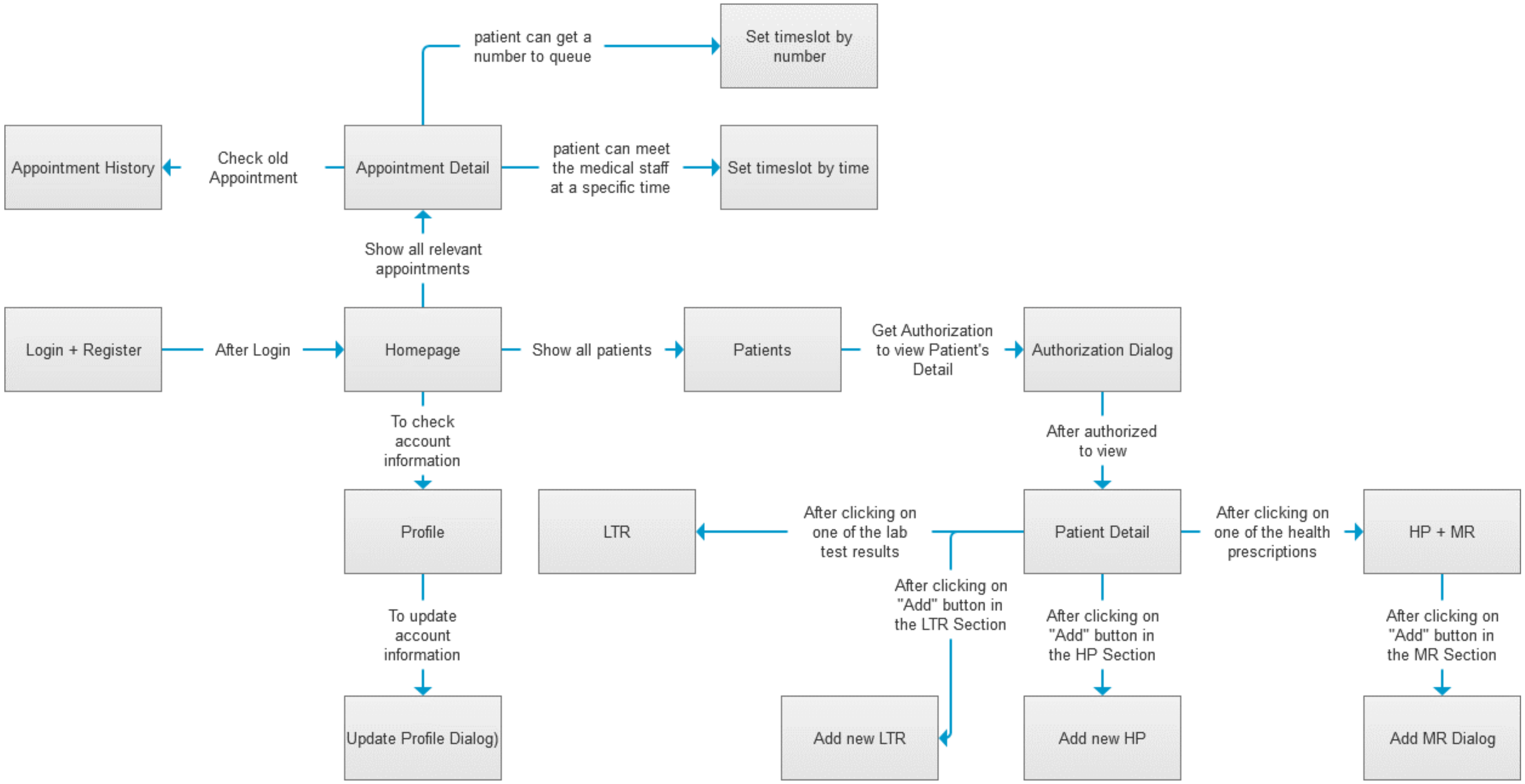

patient can get a number to queue
Set timeslot by number
Appointment History
Check old Appointment
Appointment Detail
patient can meet the medical staff at a specific time
Set timeslot by time
Show all relevant appointments
Login + Register
After Login
Homepage
Show all patients
Patients
Get Authorization to view Patient's Detail
Authorization Dialog
To check account information
After authorized to view
Profile
LTR
After clicking on one of the lab test results
Patient Detail
After clicking on one of the health prescriptions
HP + MR
To update account information
After clicking on "Add" button in the LTR Section
After clicking on "Add" button in the HP Section
After clicking on "Add" button in the MR Section
Update Profile Dialog)
Add new LTR
Add new HP
Add MR Dialog

Figure 5.79 Page Navigation flow of the Web Application

5.5.2.2. Web Application UI Design

5.5.2.2.1. Account Creation

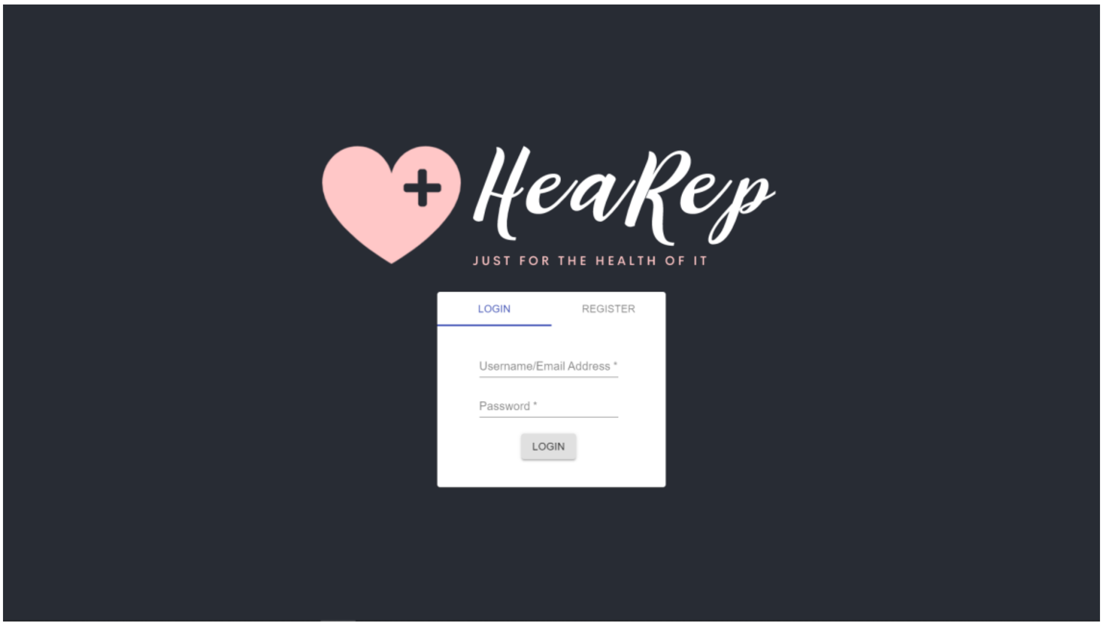


Figure 5.80 Login Tab of the Web Application (Login + Register)

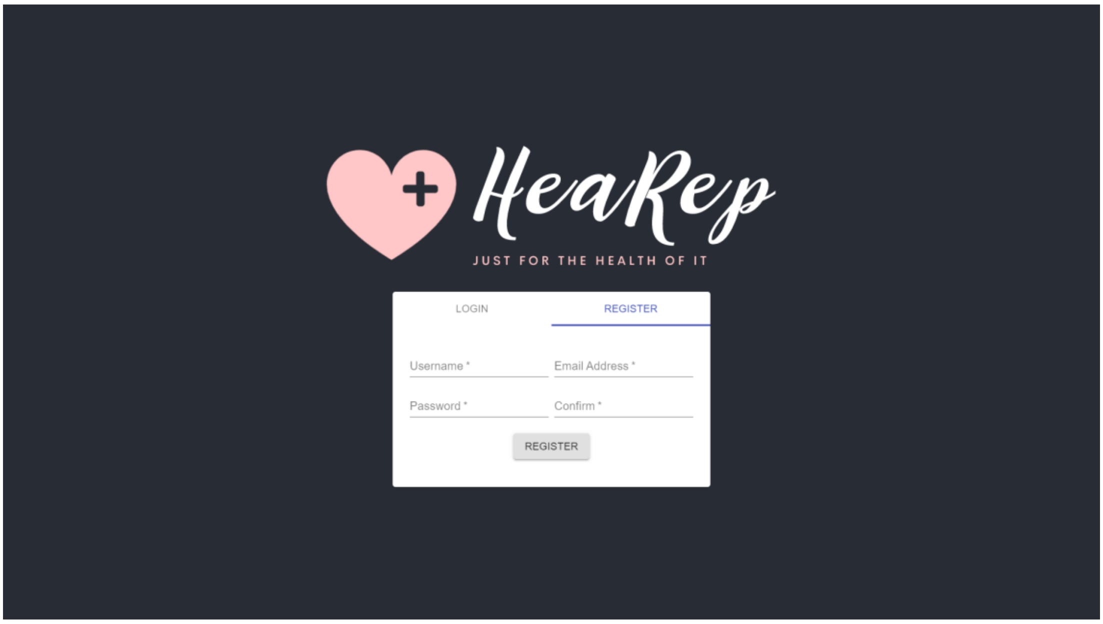
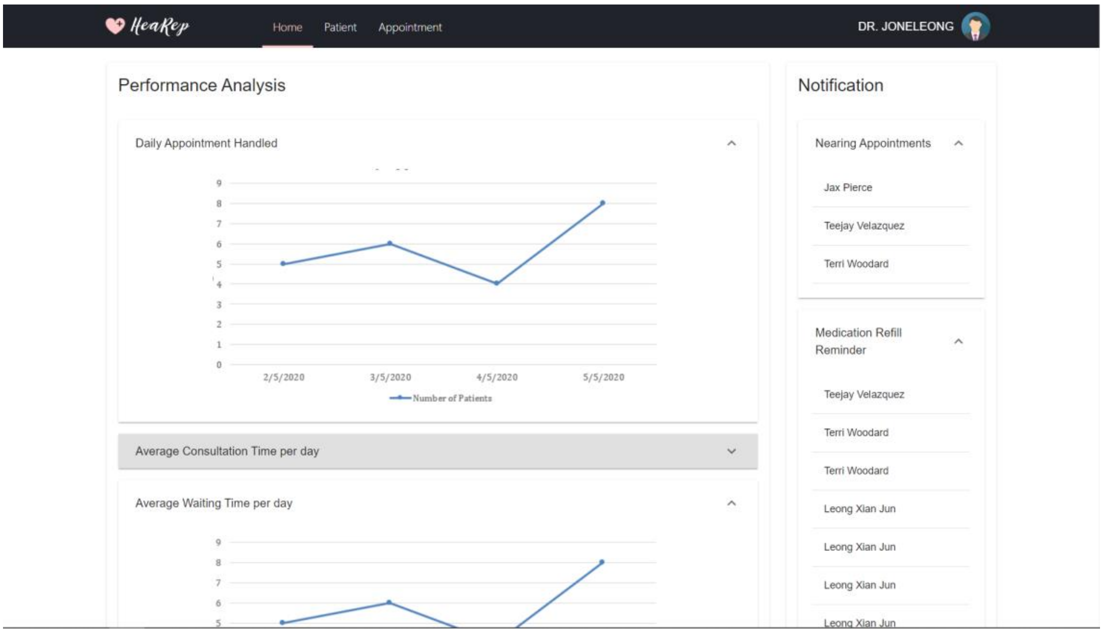


Figure 5.81 Registration Tab of the Web Application (Login + Register)



Performance Analysis

Daily Appointment Handled

Average Consultation Time per day

Average Waiting Time per day

Notification

Nearing Appointments

Jax Pierce

Teejay Velazquez

Terri Woodard

Medication Refill Reminder

Teejay Velazquez

Terri Woodard

Terri Woodard

Leong Xian Jun

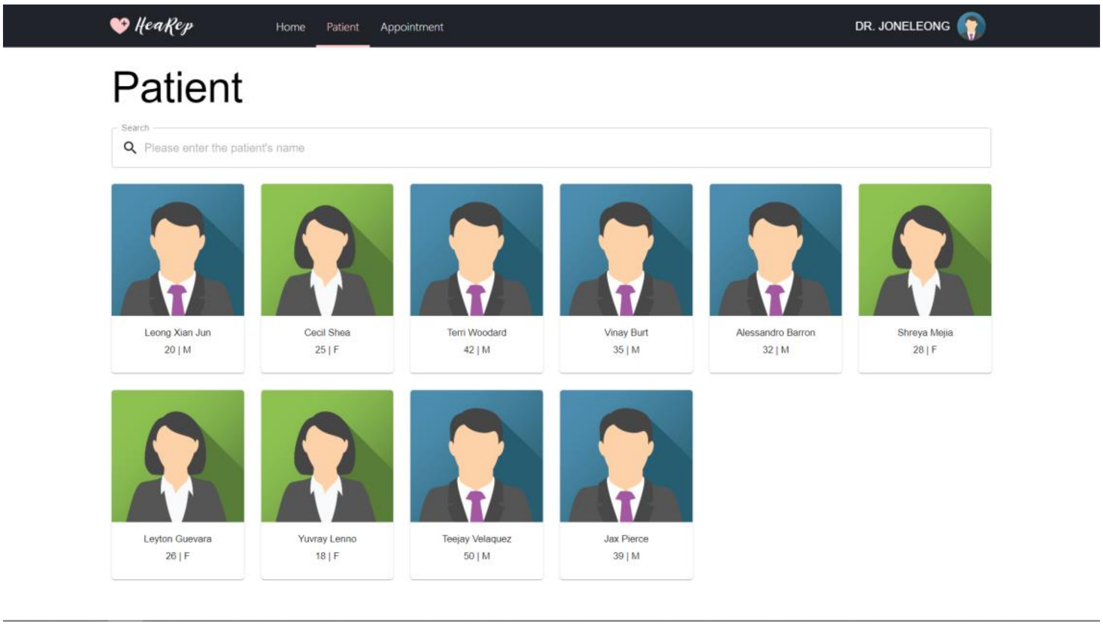
Leong Xian Jun

Leong Xian Jun

Leong Xian Jun

Figure 5.82 Homepage with Performance Analysis (Homepage)

5.5.2.2.2. Health Records Tracking



Patient

Search

Please enter the patient's name

Leong Xian Jun
20 | M

Cecil Shea
25 | F

Terri Woodard
42 | M

Vinay Burt
35 | M

Alessandro Barron
32 | M

Shreya Mejia
28 | F

Leyton Guevara
26 | F

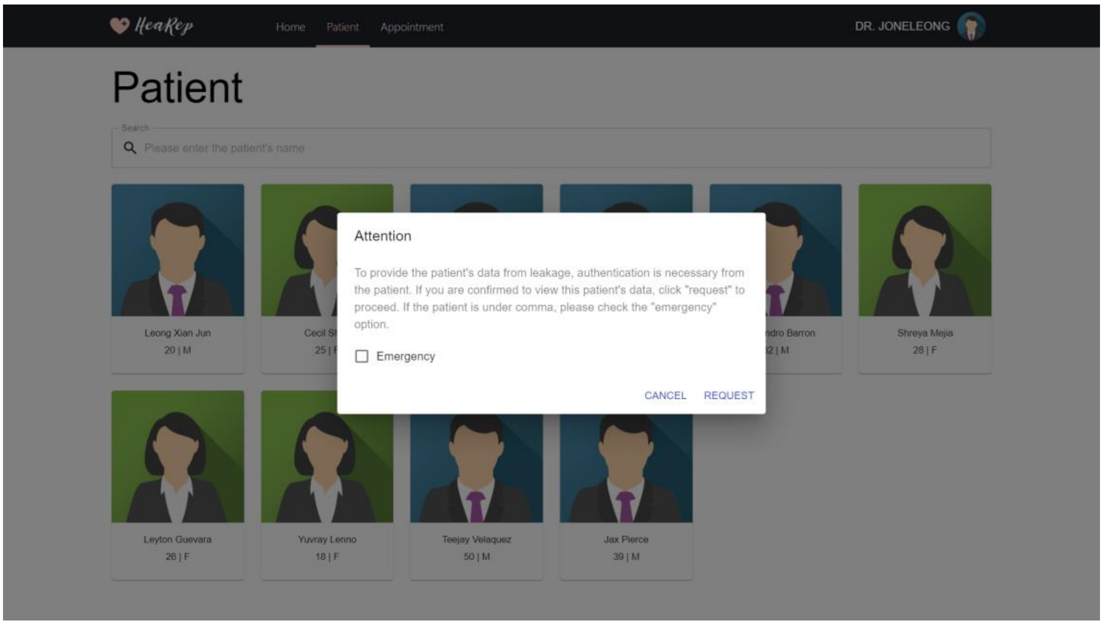
Yuvray Lenno
18 | F

Teejay Velaquez
50 | M

Jax Pierce
39 | M

Figure 5.83 Patient Page that shows all of the patients (Patient)

5.5.2.2.2.1. Health Records Access by Patients



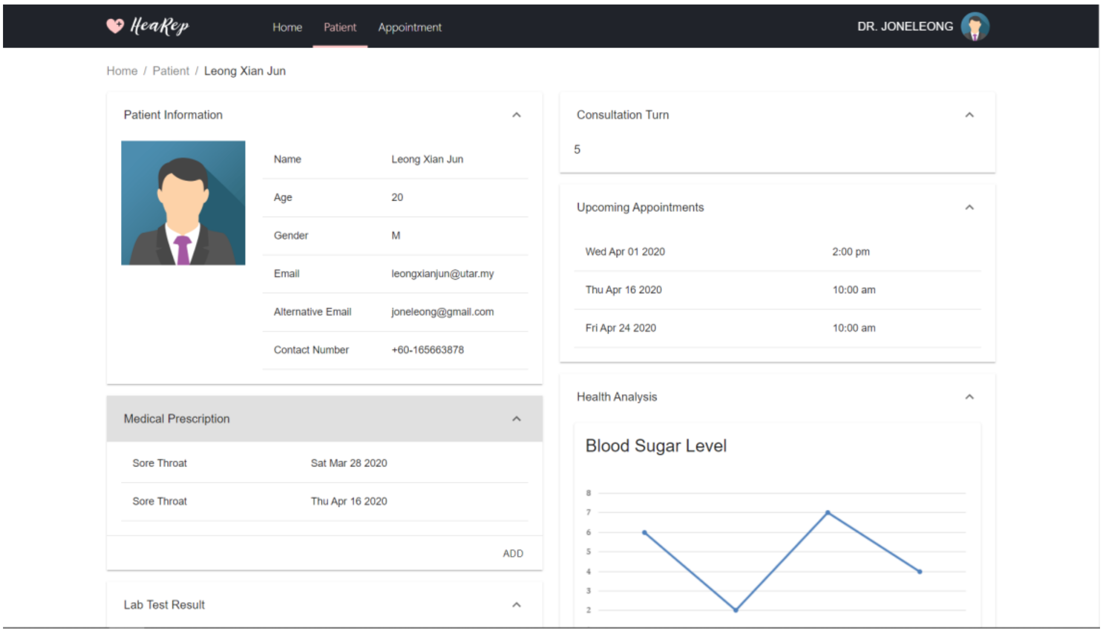
Attention

To provide the patient's data from leakage, authentication is necessary from the patient. If you are confirmed to view this patient's data, click "request" to proceed. If the patient is under comma, please check the "emergency" option.

□ Emergency

CANCEL REQUEST

Figure 5.84 Access Request Dialog to view Patient’s Detail (Authorization Dialog)



Patient Information

Name	Leong Xian Jun
Age	20
Gender	M
Email	leongxianjun@utar.my
Alternative Email	joneleong@gmail.com
Contact Number	+60-165663878

Medical Prescription

Sore Throat	Sat Mar 28 2020
Sore Throat	Thu Apr 16 2020

ADD

Lab Test Result

Consultation Turn

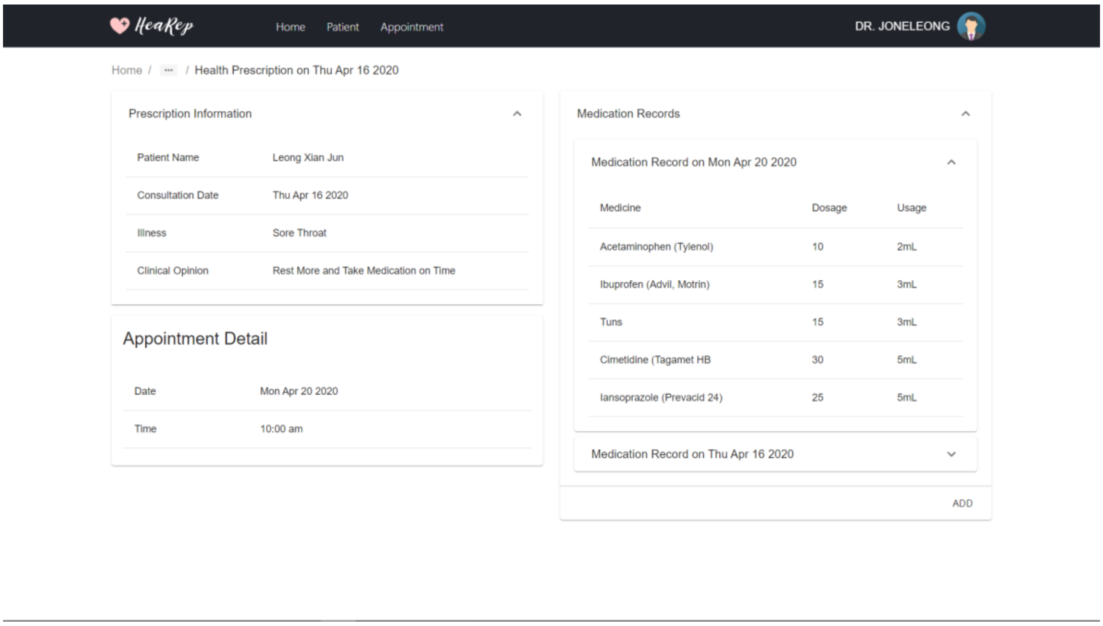
5

Upcoming Appointments

Wed Apr 01 2020	2:00 pm
Thu Apr 16 2020	10:00 am
Fri Apr 24 2020	10:00 am

Health Analysis

Figure 5.85 Detail of the Patient with Health Records and Analysis (Patient Detail)



Home / ··· / Health Prescription on Thu Apr 16 2020

Prescription Information

Patient Name	Leong Xian Jun
Consultation Date	Thu Apr 16 2020
Illness	Sore Throat
Clinical Opinion	Rest More and Take Medication on Time

Appointment Detail

Date	Mon Apr 20 2020
Time	10:00 am

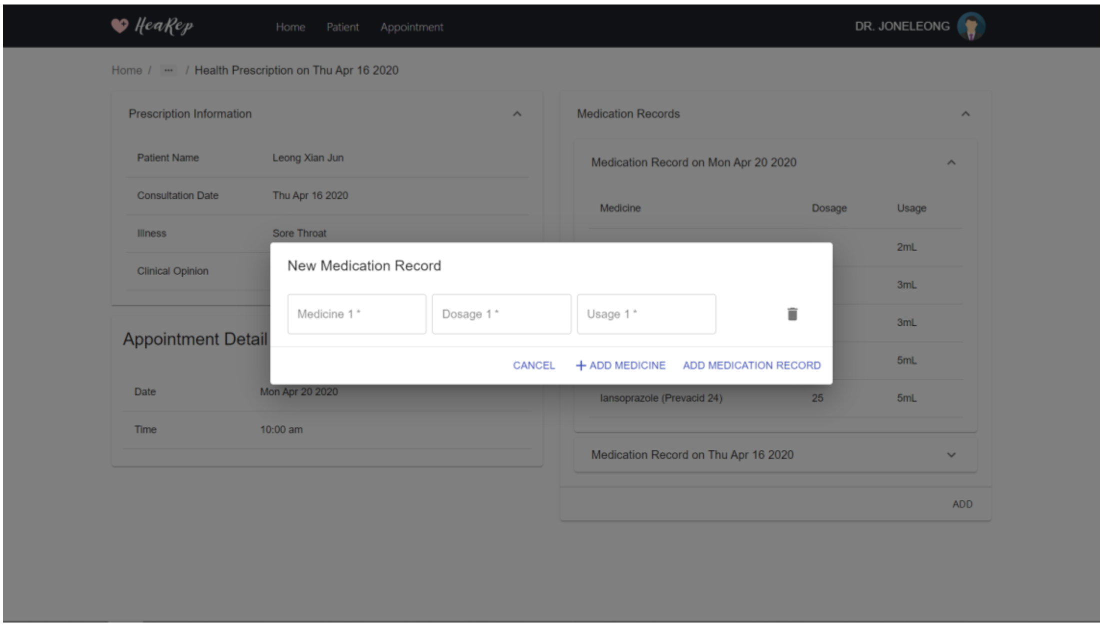
Medication Records

Medication Record on Mon Apr 20 2020

Medicine	Dosage	Usage
Acetaminophen (Tylenol)	10	2mL
Ibuprofen (Advil, Motrin)	15	3mL
Tuns	15	3mL
Cimetidine (Tagamet HB	30	5mL
lansoprazole (Prevacid 24)	25	5mL

Medication Record on Thu Apr 16 2020

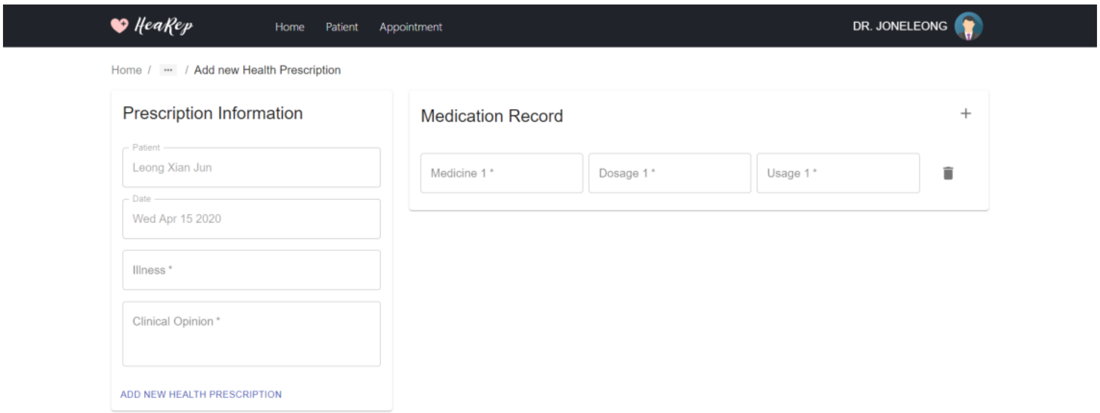
Figure 5.86 Detail of the Health Prescription with Medication Record (HP + MR)



New Medication Record

Medicine 1 * | Dosage 1 * | Usage 1 *

Figure 5.87 Dialog to add new Medication Record (Add MR Dialog)


HeaRep
Home
Patient
Appointment
DR. JONELEONG
Home / ··· / Add new Health Prescription
+
ADD NEW HEALTH PRESCRIPTION

Prescription Information

Patient: Leong Xian Jun

Date: Wed Apr 15 2020

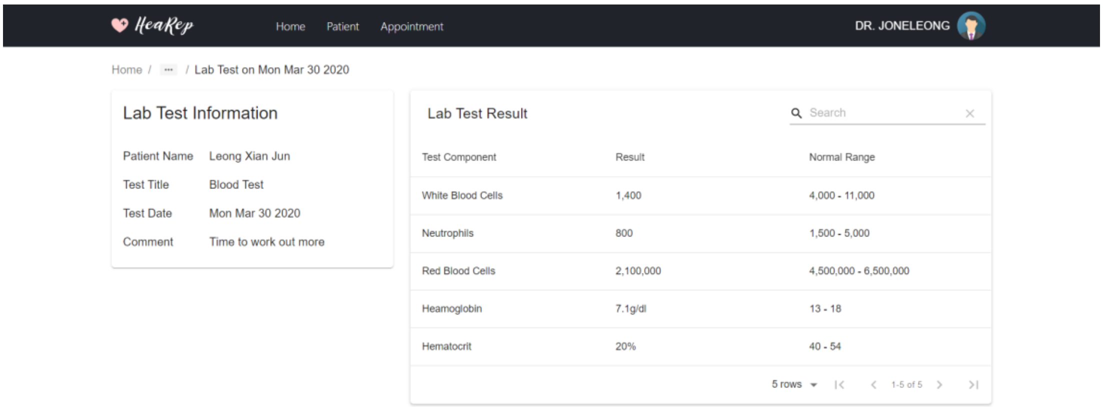
Illness *

Clinical Opinion *

Medication Record

Medicine 1 * | Dosage 1 * | Usage 1 *

Figure 5.88 Page to add new Health Prescription (Add new HP)



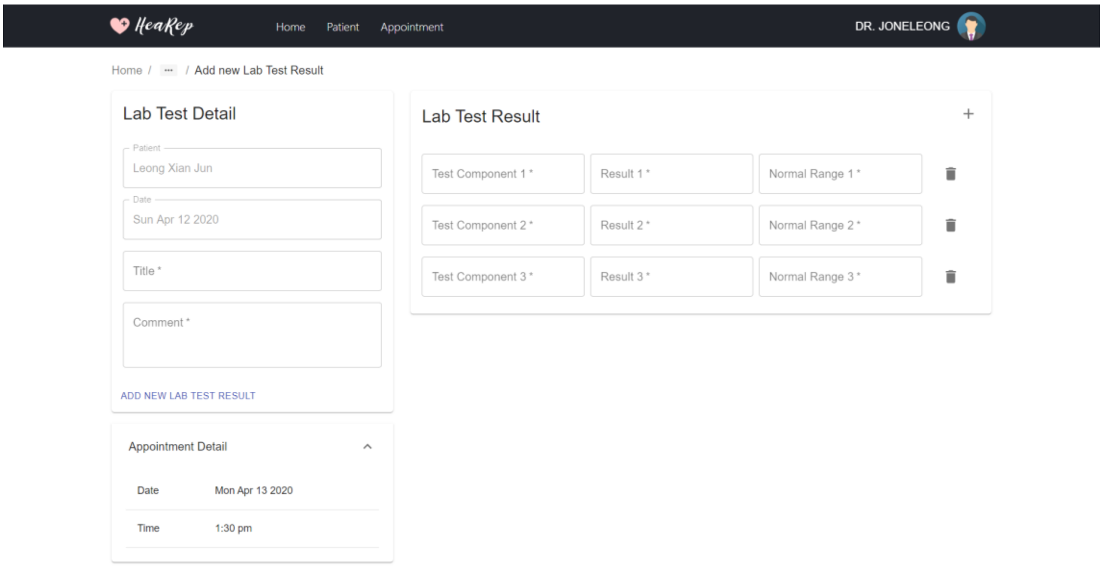
Lab Test Information

Patient Name	Leong Xian Jun
Test Title	Blood Test
Test Date	Mon Mar 30 2020
Comment	Time to work out more

Lab Test Result

Test Component	Result	Normal Range
White Blood Cells	1,400	4,000 - 11,000
Neutrophils	800	1,500 - 5,000
Red Blood Cells	2,100,000	4,500,000 - 6,500,000
Heamoglobin	7.1g/dl	13 - 18
Hematocrit	20%	40 - 54

Figure 5.89 Detail of the Lab Test Result (LTR)



Lab Test Detail

Patient: Leong Xian Jun

Date: Sun Apr 12 2020

Title *

Comment *

Lab Test Result

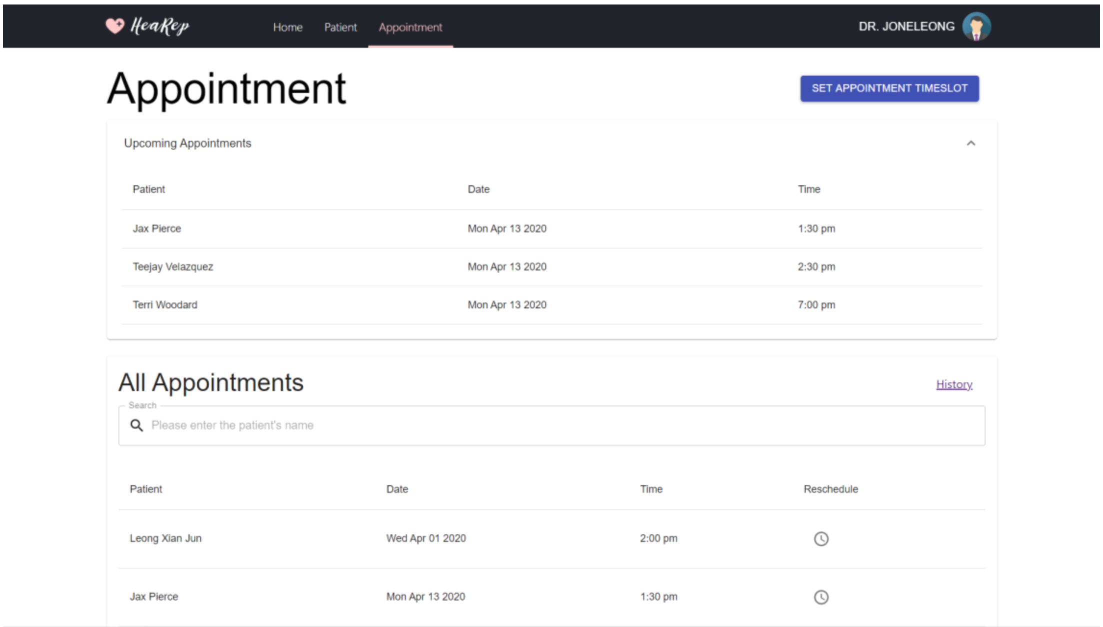
Test Component 1 *	Result 1 *	Normal Range 1 *
Test Component 2 *	Result 2 *	Normal Range 2 *
Test Component 3 *	Result 3 *	Normal Range 3 *

Appointment Detail

Date	Mon Apr 13 2020
Time	1:30 pm

Figure 5.90 Page to add new Lab Test Result (Add new LTR)

5.5.2.2.3. Appointment Scheduling



Appointment

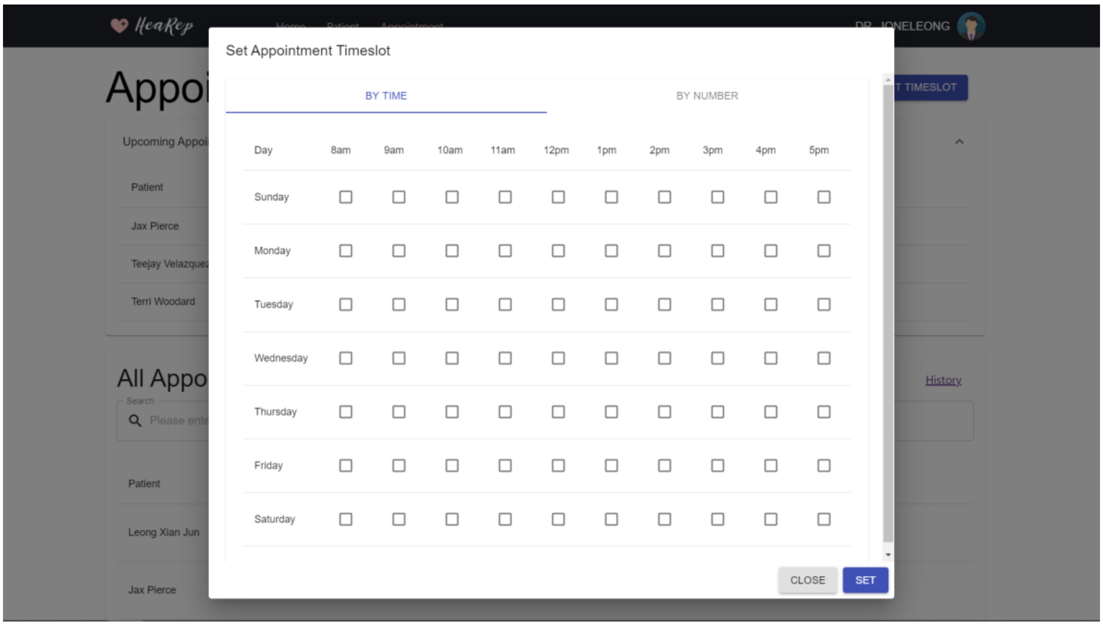
Upcoming Appointments

Patient	Date	Time
Jax Pierce	Mon Apr 13 2020	1:30 pm
Teejay Velazquez	Mon Apr 13 2020	2:30 pm
Terri Woodard	Mon Apr 13 2020	7:00 pm

All Appointments

Patient	Date	Time	Reschedule
Leong Xian Jun	Wed Apr 01 2020	2:00 pm	🕓
Jax Pierce	Mon Apr 13 2020	1:30 pm	🕓

Figure 5.91 Appointment Page with upcoming Appointments (Appointment Detail)

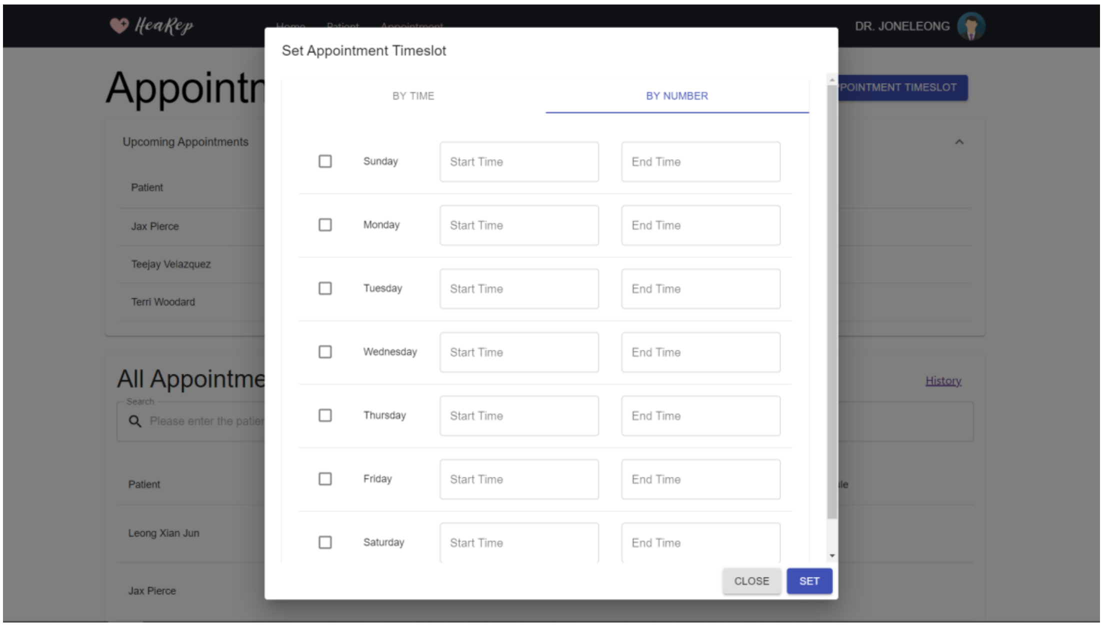


Set Appointment Timeslot

BY TIME | BY NUMBER

Day	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm
Sunday	□	□	□	□	□	□	□	□	□	□
Monday	□	□	□	□	□	□	□	□	□	□
Tuesday	□	□	□	□	□	□	□	□	□	□
Wednesday	□	□	□	□	□	□	□	□	□	□
Thursday	□	□	□	□	□	□	□	□	□	□
Friday	□	□	□	□	□	□	□	□	□	□
Saturday	□	□	□	□	□	□	□	□	□	□

Figure 5.92 Dialog that set timeslot by time (Set timeslot by time)

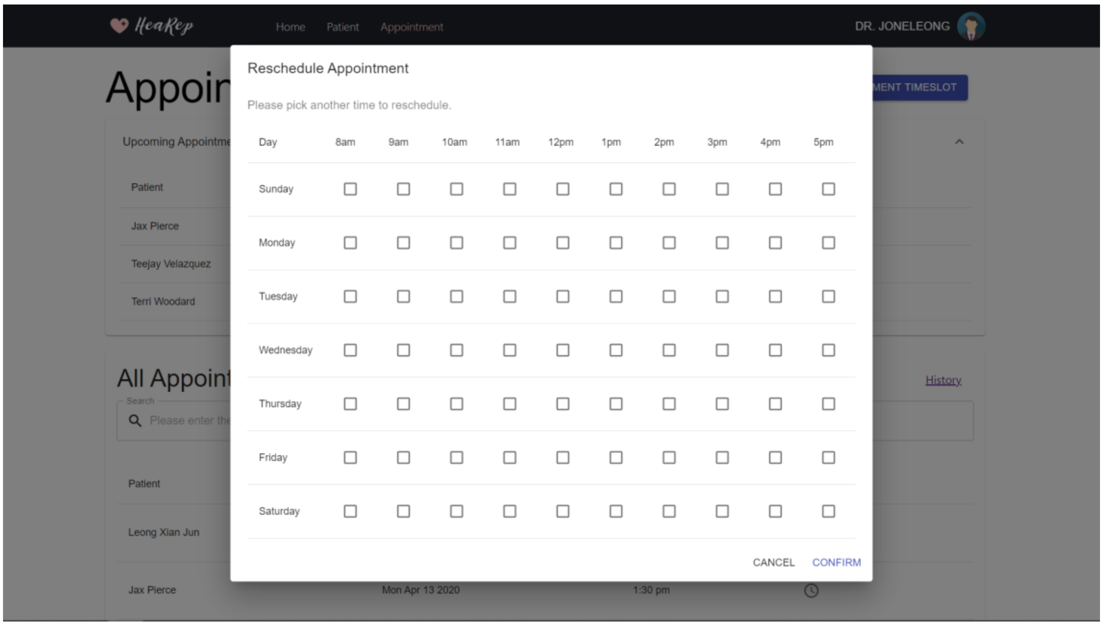


Set Appointment Timeslot

BY TIME | BY NUMBER

□	Sunday	Start Time	End Time
□	Monday	Start Time	End Time
□	Tuesday	Start Time	End Time
□	Wednesday	Start Time	End Time
□	Thursday	Start Time	End Time
□	Friday	Start Time	End Time
□	Saturday	Start Time	End Time

Figure 5.93 Dialog that set timeslot by number (Set timeslot by number)



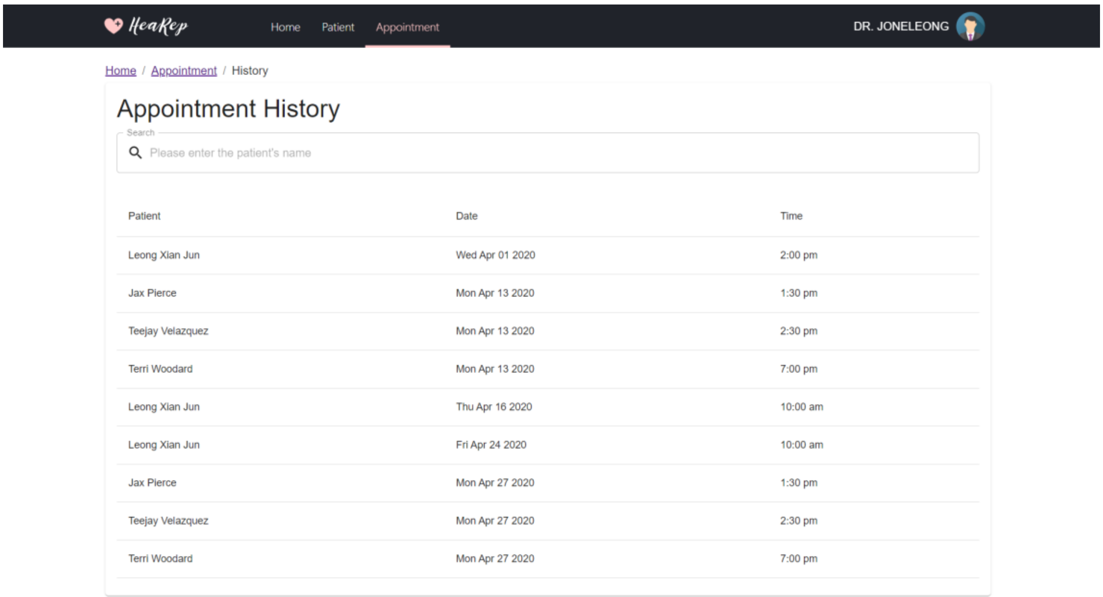
Reschedule Appointment

Please pick another time to reschedule.

Day	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm
Sunday	□	□	□	□	□	□	□	□	□	□
Monday	□	□	□	□	□	□	□	□	□	□
Tuesday	□	□	□	□	□	□	□	□	□	□
Wednesday	□	□	□	□	□	□	□	□	□	□
Thursday	□	□	□	□	□	□	□	□	□	□
Friday	□	□	□	□	□	□	□	□	□	□
Saturday	□	□	□	□	□	□	□	□	□	□

CANCEL CONFIRM

Figure 5.94 Dialog to reschedule an Appointment (Reschedule Dialog)



Home / Appointment / History

Appointment History

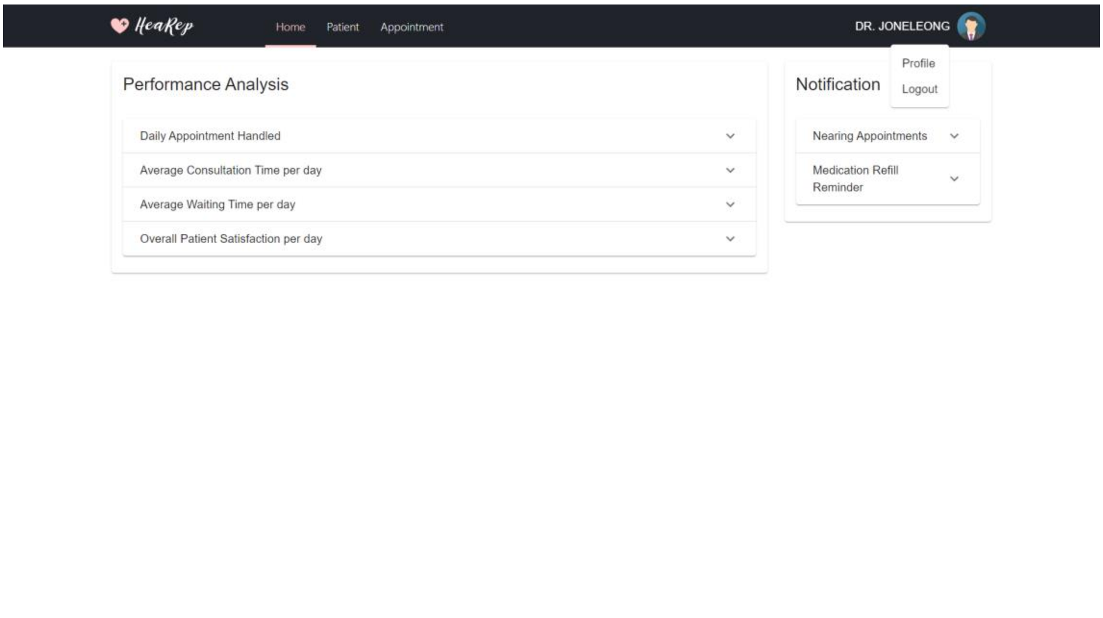
Search

Please enter the patient's name

Patient	Date	Time
Leong Xian Jun	Wed Apr 01 2020	2:00 pm
Jax Pierce	Mon Apr 13 2020	1:30 pm
Teejay Velazquez	Mon Apr 13 2020	2:30 pm
Terri Woodard	Mon Apr 13 2020	7:00 pm
Leong Xian Jun	Thu Apr 16 2020	10:00 am
Leong Xian Jun	Fri Apr 24 2020	10:00 am
Jax Pierce	Mon Apr 27 2020	1:30 pm
Teejay Velazquez	Mon Apr 27 2020	2:30 pm
Terri Woodard	Mon Apr 27 2020	7:00 pm

Figure 5.95 Page that shows the History of Appointment

5.5.2.2.4. Account Management



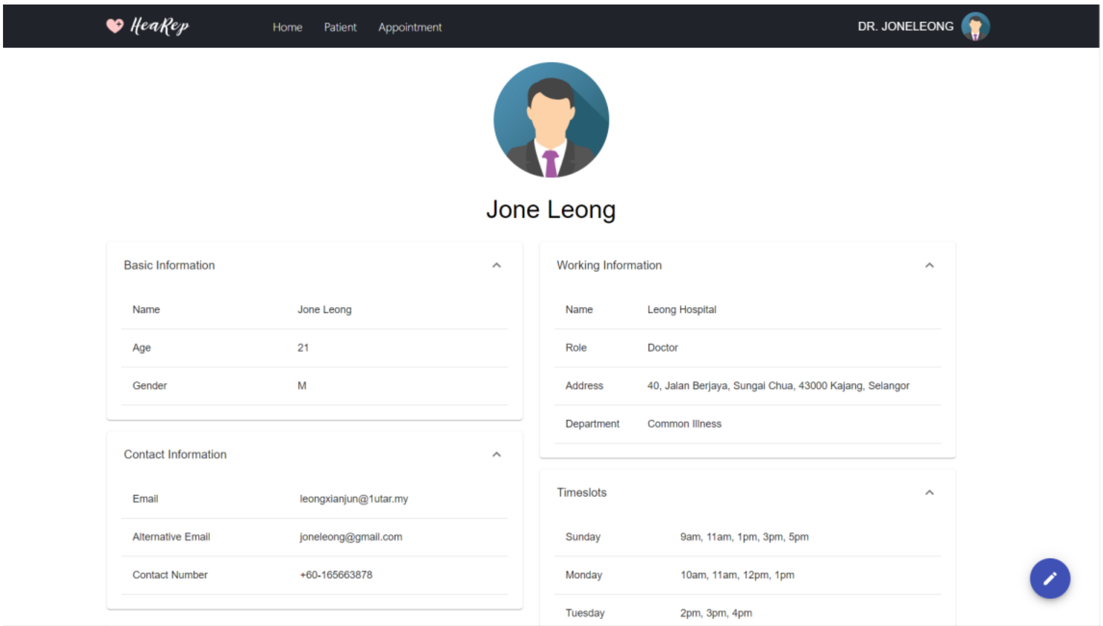
Performance Analysis

- Daily Appointment Handled
- Average Consultation Time per day
- Average Waiting Time per day
- Overall Patient Satisfaction per day

Notification

- Nearing Appointments
- Medication Refill Reminder

Figure 5.96 Menu List to access Profile or logout



Jone Leong

Basic Information

Name	Jone Leong
Age	21
Gender	M

Contact Information

Email	leongxianjun@1utar.my
Alternative Email	joneleong@gmail.com
Contact Number	+60-165663878

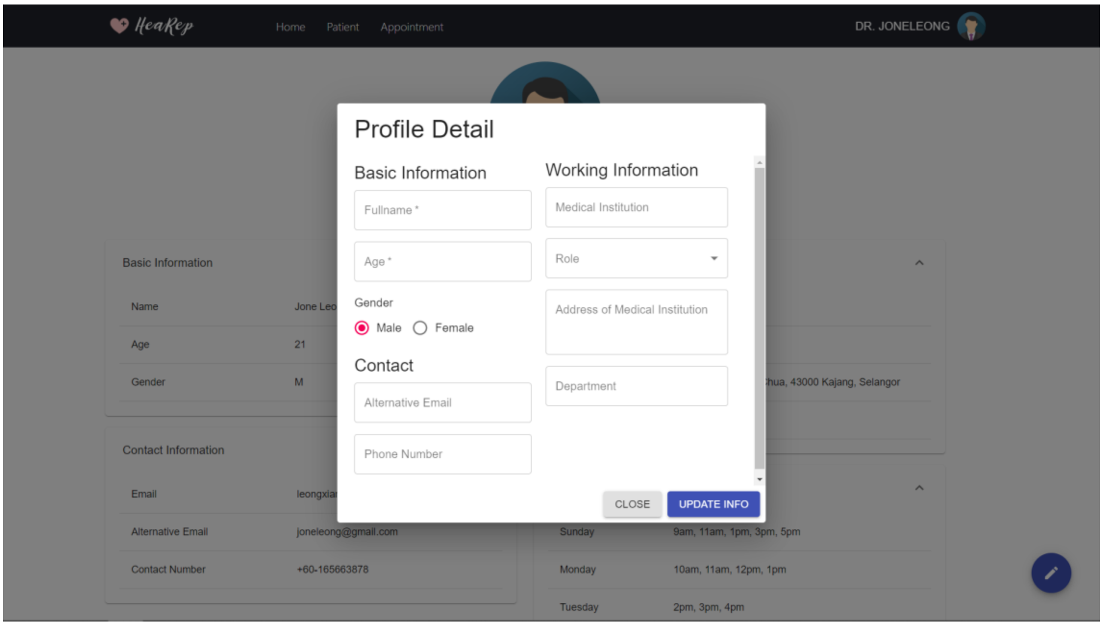
Working Information

Name	Leong Hospital
Role	Doctor
Address	40, Jalan Berjaya, Sungai Chua, 43000 Kajang, Selangor
Department	Common Illness

Timeslots

Sunday	9am, 11am, 1pm, 3pm, 5pm
Monday	10am, 11am, 12pm, 1pm
Tuesday	2pm, 3pm, 4pm

Figure 5.97 Profile Page (Profile)



Gender
● Male ○ Female

Basic Information

Name	Jone Leo
Age	21
Gender	M

hua, 43000 Kajang, Selangor

Contact Information

Email	leongxia
Alternative Email	joneleong@gmail.com
Contact Number	+60-165663878

Sunday	9am, 11am, 1pm, 3pm, 5pm
Monday	10am, 11am, 12pm, 1pm
Tuesday	2pm, 3pm, 4pm

Figure 5.98 Dialog to update the Profile Detail (Update Profile Dialog)

5.6. Summary

In short, this chapter illustrated the overview of the system architecture with a more detailed information flow between applications and server. Besides, modelling diagrams of the system were prepared to showcase the information structure in the system as well as the information flow between modules. Moreover, it also contained a low-fidelity prototype that provided a brief layout on the actual application UI design. With the brief layout, a high-fidelity prototype was prepared to illustrate the initial system design for both applications.

CHAPTER 6

DEVELOPMENT AND TESTING

6.1. Introduction

This section illustrates the execution process of the development as well as the testing conducted. This section would be separated into two parts, which introduces the development and testing respectively. In-depth knowledge of the execution would be obtained from this section.

6.2. Development Execution

During the development of the system, some improvements in the UI design were made. Such changes were reflected in this section. Additionally, the user manual of the system was provided in this section, so guidance or help can be provided to the users of the system to reduce the chance to have confusion.

On top of that, all of the APIs was listed below as proper documentation of their functionalities as well as their parameters. It provided a clearer image of the available API in the system. Last but not least, during development, some good practices were implemented to ease the entire development process.

6.2.1. Improved User Interface

The following are the improvement done on the UI design of both mobile and web applications. Such improvements were made to provide a better user experience.

6.2.1.1. Mobile Application

6.2.1.1.1. Replacement of Fake Graph Images into Real Graph

In the high-fidelity prototype, images were used to present the graph. In the actual implementation, these images were replaced with an actual graph.

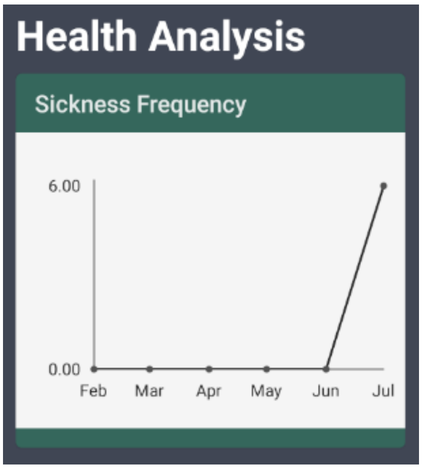


Figure 6.1 Actual Graph Implementation

6.2.1.1.2. Notification

The notification was implemented to inform the user on the latest information sent from the backend server. Moreover, the mobile application also provided In-App notification.

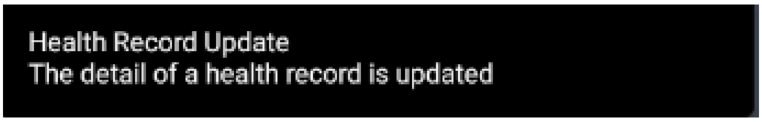
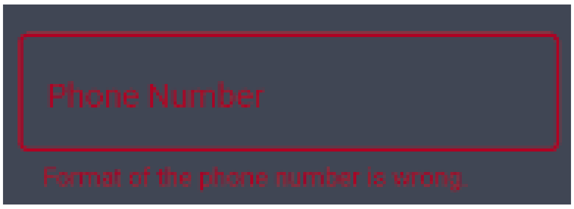

Health Record Update
The detail of a health record is updated

Figure 6.2 In-App Notification Overview

6.2.1.1.3. Error Handling

The mobile application involved pages that required validation. Hence, these pages were updated so they can display the error messages correctly.


Phone Number

Format of the phone number is wrong

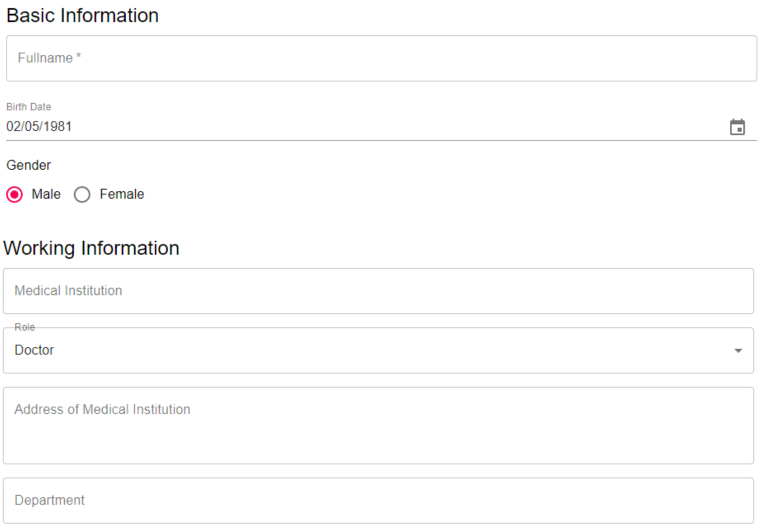
Figure 6.3 Sample of Error Handling in Mobile Application

6.2.1.2. Web Application

Overall, all of the pages are responsive. This is to ensure the user can access the web application using any device with different screen resolutions.

6.2.1.2.1. New Account Information Entry

After the user created an account, the user is required to enter the basic information as well as his/her working information. This will set up the account detail of the user in the web application.



Basic Information

Fullname *

Birth Date
02/05/1981

Gender
● Male ○ Female

Working Information

Medical Institution

Role
Doctor

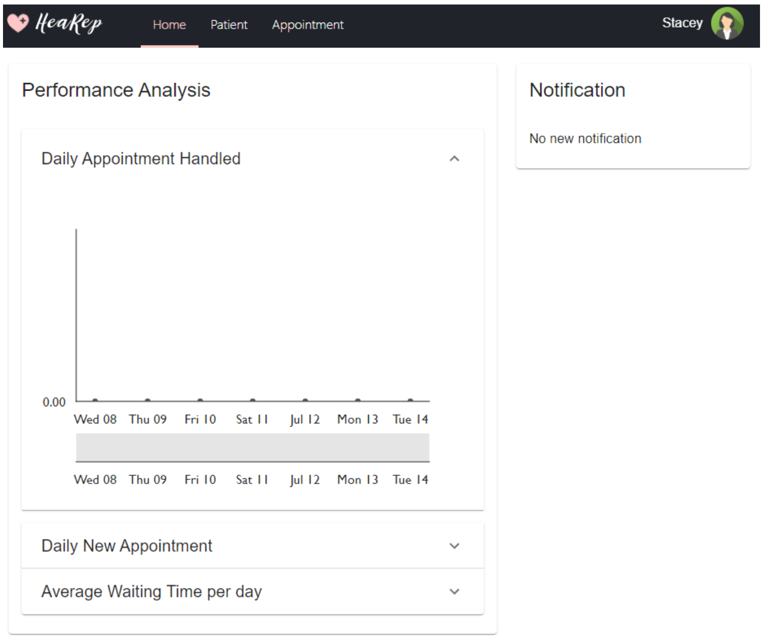
Address of Medical Institution

Department

Figure 6.4 Information Entry Page Overview

6.2.1.2.2. Replacement of Fake Graph Images into Real Graph

In the high-fidelity prototype, images were used to present the graph. Similar to the mobile application, these images were replaced with an actual graph.

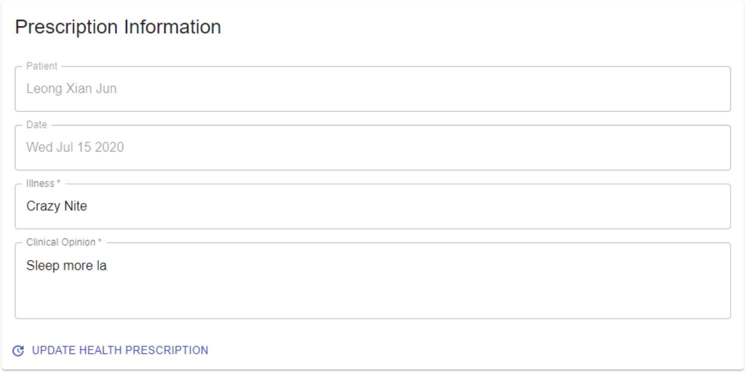


No new notification

Figure 6.5 Actual Graph Presentation in Web Application

6.2.1.2.3. Modification of Health Record

The user is allowed to modify the existing health records by clicking on the “Edit” icon in the detail page.



Prescription Information

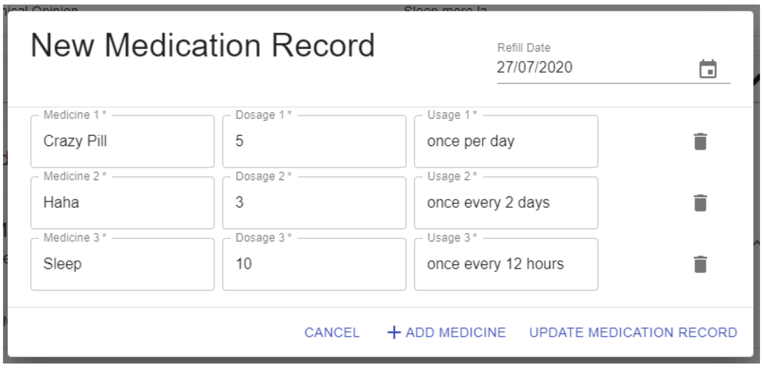
Patient: Leong Xian Jun

Date: Wed Jul 15 2020

Illness *: Crazy Nite

Clinical Opinion *: Sleep more la

Figure 6.6 Page Overview to edit Health Prescription

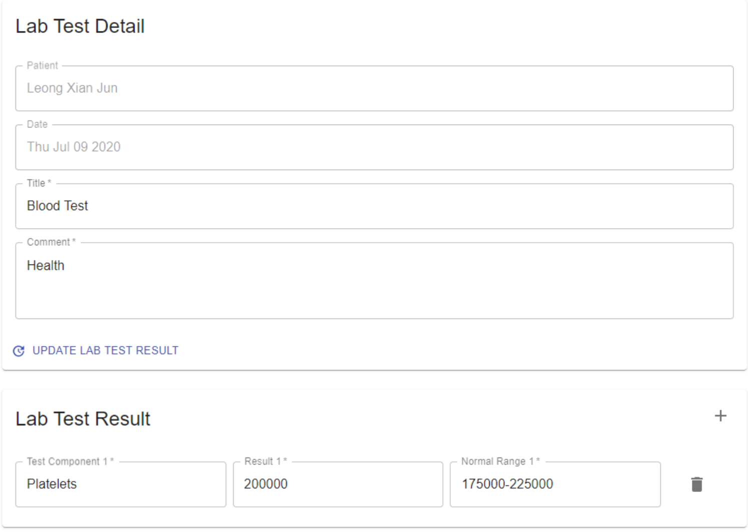


New Medication Record

Refill Date
27/07/2020

Medicine 1 * Crazy Pill	Dosage 1 * 5	Usage 1 * once per day
Medicine 2 * Haha	Dosage 2 * 3	Usage 2 * once every 2 days
Medicine 3 * Sleep	Dosage 3 * 10	Usage 3 * once every 12 hours

Figure 6.7 Dialog Overview to edit Medication Record



Lab Test Detail

Patient
Leong Xian Jun

Date
Thu Jul 09 2020

Title *
Blood Test

Comment *
Health

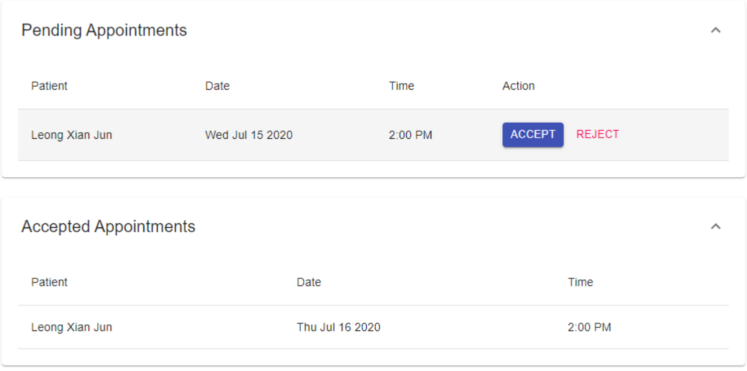
Lab Test Result

Test Component 1 *	Result 1 *	Normal Range 1 *
Platelets	200000	175000-225000

Figure 6.8 Page Overview to edit Lab Test Result

6.2.1.2.4. Grouping of Appointment based on the Status

The appointments are grouped according to their status. Additionally, the medical staff is required to accept or reject a pending appointment.



Pending Appointments

Patient	Date	Time	Action
Leong Xian Jun	Wed Jul 15 2020	2:00 PM	ACCEPT REJECT

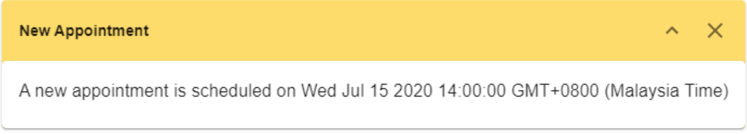
Accepted Appointments

Patient	Date	Time
Leong Xian Jun	Thu Jul 16 2020	2:00 PM

Figure 6.9 Overview of the Appointment Grouping

6.2.1.2.5. Notification Snack bar

Snackbar is a component used in the web application in alerting the user on the new notification of the application. It is located at the bottom left of the page.



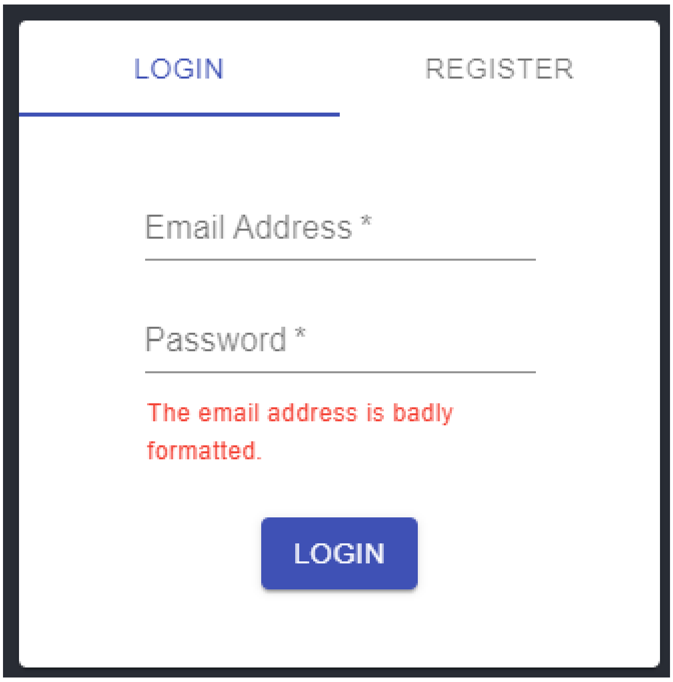
New Appointment

A new appointment is scheduled on Wed Jul 15 2020 14:00:00 GMT+0800 (Malaysia Time)

Figure 6.10 Snackbar Overview

6.2.1.2.6. Error Handling

The web application consists of many pages that require data insertion. Therefore, validation is required. Due to that, these pages were updated to the error messages can be shown correctly.



The email address is badly formatted.

Figure 6.11 Sample of Error Handling in Web Application

6.2.2. User Manual

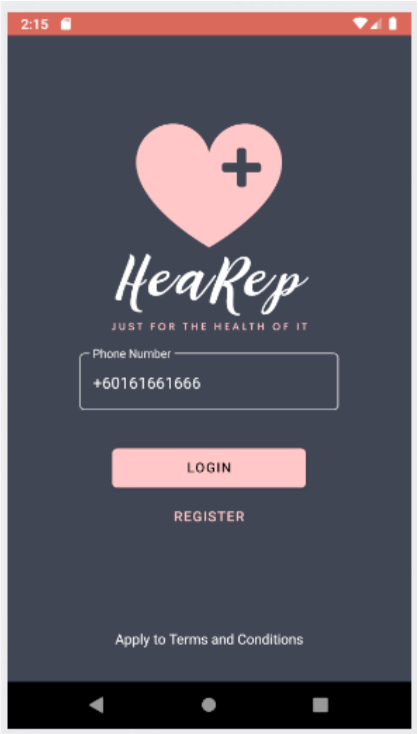
The user manual guides the users of both mobile and web application in using the application. It contains a series of actions to achieve a specific objective.

6.2.2.1. Mobile Application

6.2.2.1.1. Account Registration and Sign In

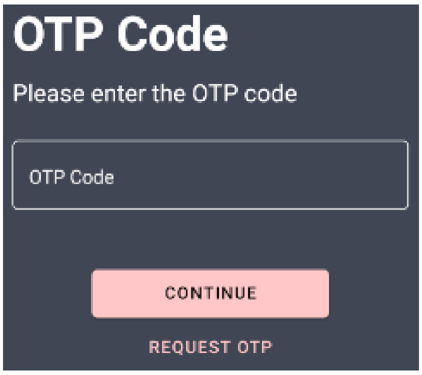
To register an account, the user is required to enter the phone number. Then, the system will check if any user used the phone number. If it is not used by any other user, the user can proceed to enter the OTP code. After entering the OTP code correctly, the user required to enter the basic information of the account profile. With the completion of the profile insertion, the registration of the account is considered completed.

Similarly, to registration, sign-in requires the user to enter their phone number and OTP code. Upon success authentication, the mobile application would direct the user to the homepage.



Phone Number: +60161661666

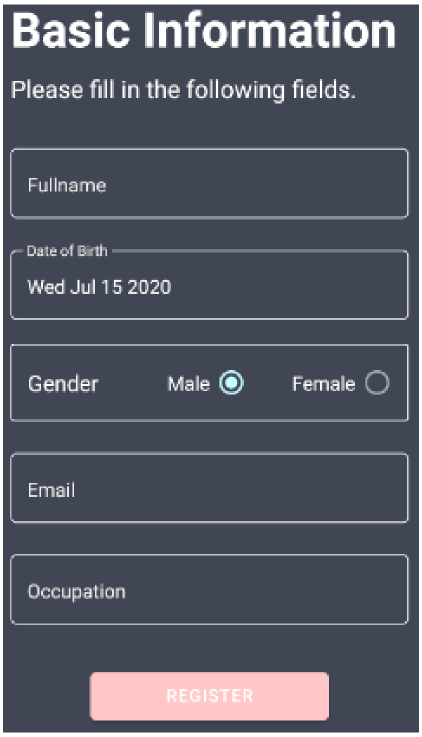
Figure 6.12 Authentication Screen Overview



OTP Code

Please enter the OTP code

Figure 6.13 OTP Entering Screen Overview



Basic Information

Please fill in the following fields.

Fullname

Date of Birth: Wed Jul 15 2020

Gender: Male ● Female ○

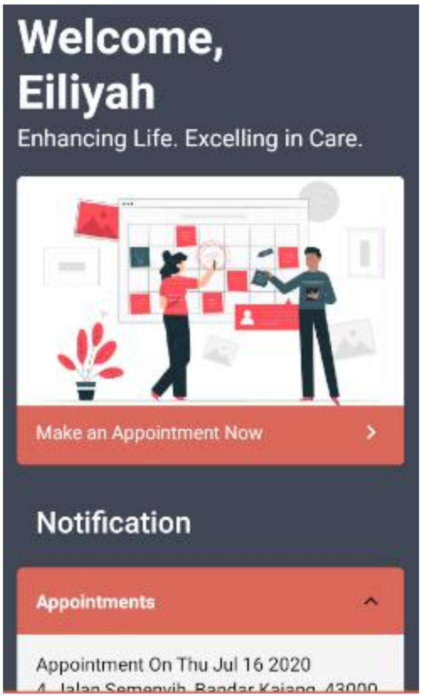
Email

Occupation

Figure 6.14 Page Overview to enter Account Basic Information

6.2.2.1.2. Homepage

After the successful authentication, the mobile application would redirect the user to the homepage of the mobile application. In the homepage, the mobile application shown the nearing appointments of the users.



Welcome, Eiliyah

Enhancing Life. Excelling in Care.

Notification

Appointment On Thu Jul 16 2020

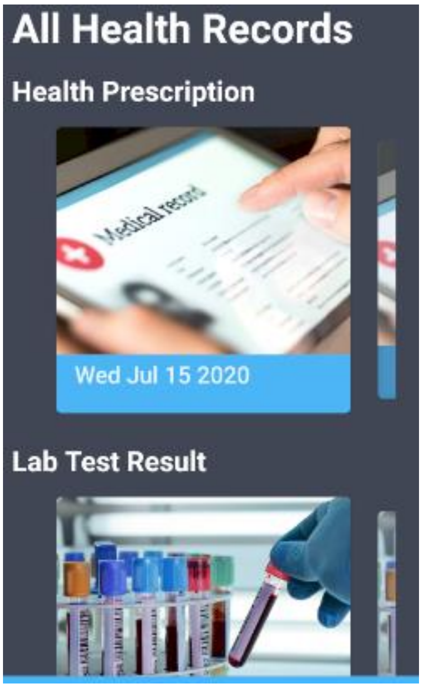
4, Jalan Semenyih, Bandar Kajang, 43000

Figure 6.15 Overview of Homepage

6.2.2.1.3. Health Record Tracking

To check the health records, the user can click on the “Health Record” tab in the bottom navigation bar. The mobile application then redirects the user to the page that lists all of their health records.

To view the detail of the health prescription, the user can simply click on the card component and the mobile application will redirect the user to the detail page. Moreover, to add a reminder for a medication refill, the user can click on the “Add Reminder for Refill” button.



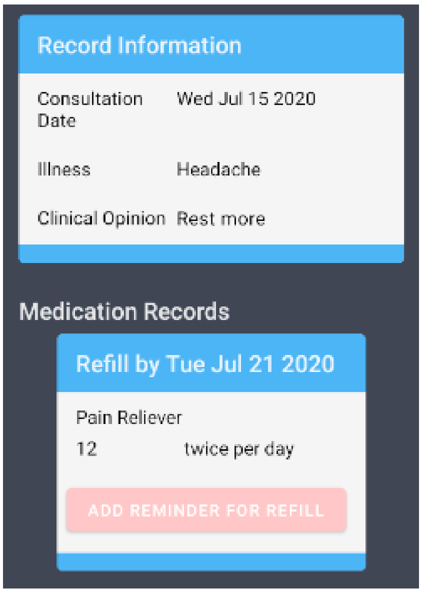
All Health Records

Health Prescription

Wed Jul 15 2020

Lab Test Result

Figure 6.16 Overview of the Health Record List



Record Information

Consultation Date	Wed Jul 15 2020
Illness	Headache
Clinical Opinion	Rest more

Medication Records

Refill by Tue Jul 21 2020

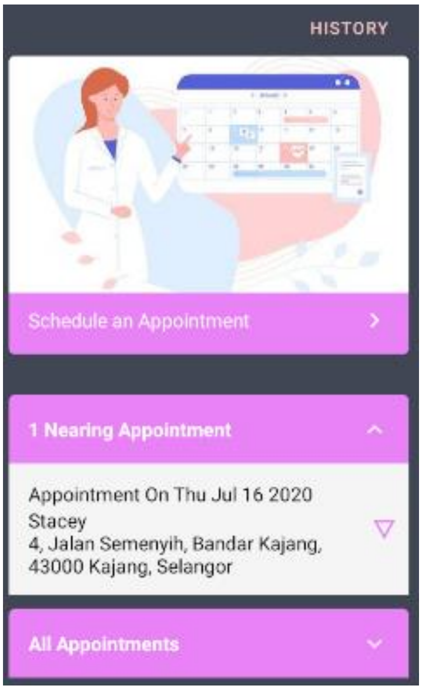
Pain Reliever

12 twice per day

Figure 6.17 Overview of Health Prescription Detail

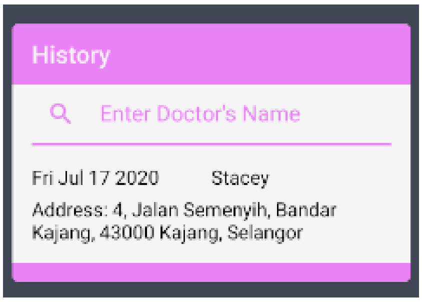
6.2.2.1.4. Appointment Scheduling and Management

The user can schedule an appointment with medical staff via the mobile application. It can be accessed by clicking the card on the homepage. After clicking the card, the mobile application will redirect the user and the list of his/her appointments would be shown. Additionally, the user can view their completed or cancelled appointment on the history page of the appointments.



Appointment On Thu Jul 16 2020
Stacey
4, Jalan Semenyih, Bandar Kajang,
43000 Kajang, Selangor

Figure 6.18 Appointment Page Overview



Fri Jul 17 2020 Stacey
Address: 4, Jalan Semenyih, Bandar Kajang, 43000 Kajang, Selangor

Figure 6.19 Appointment History Overview

To schedule an appointment, the user is required to click on “Schedule an Appointment” card. Then, the mobile application will display a list of the medical staff. The user can filter the medical staff by entering a keyword in the searching bar provided. After selecting a medical staff, the user will either be required to select a timeslot or pick a number for queuing up. After that, the mobile application will prompt the user to finalize his/her selection before submitting the request to the backend server.

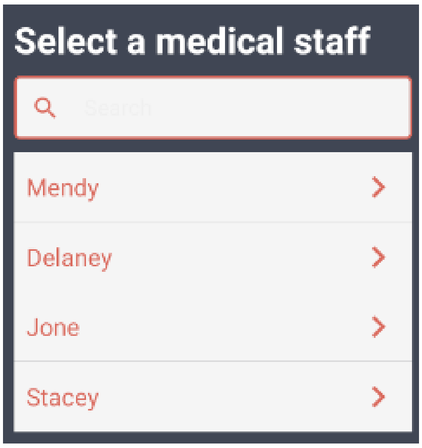


Figure 6.20 Screen Overview to select a medical staff

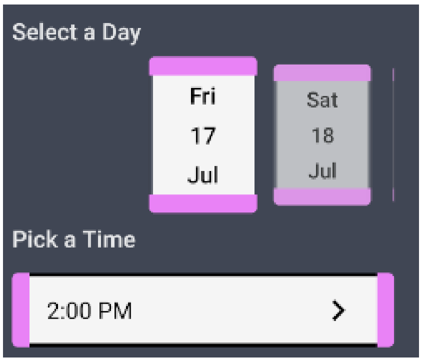
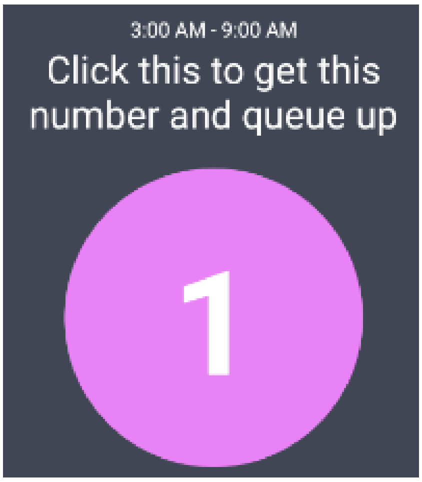


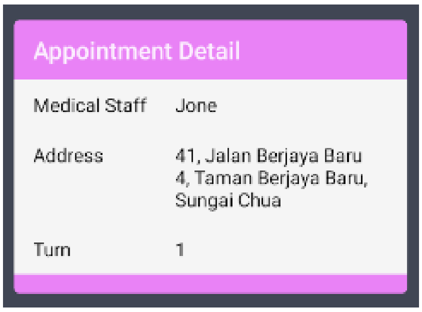
Figure 6.21 Screen Overview to select a timeslot



3:00 AM - 9:00 AM

Click this to get this number and queue up

Figure 6.22 Screen Overview to get a number to queue up



Appointment Detail

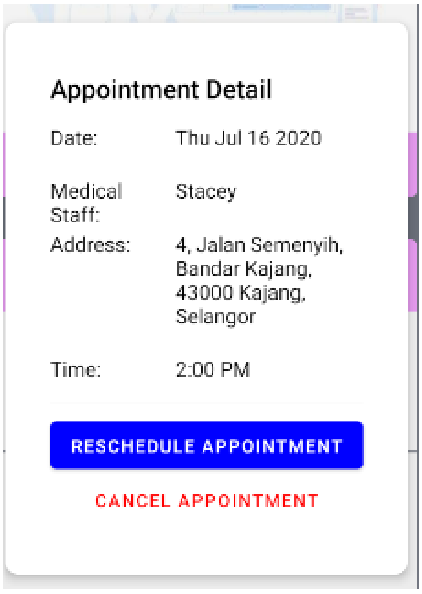
Medical Staff Jone

Address 41, Jalan Berjaya Baru 4, Taman Berjaya Baru, Sungai Chua

Turn 1

Figure 6.23 Screen Overview to confirm the appointment detail

Other than that, the user can reschedule or cancel the appointment made. Such features can be accessed by clicking on the appointment listed in Figure 6.19. Then, a dialogue containing the appointment detail will pop out.



Appointment Detail

Date: Thu Jul 16 2020

Medical Staff: Stacey

Address: 4, Jalan Semenyih, Bandar Kajang, 43000 Kajang, Selangor

Time: 2:00 PM

Figure 6.24 Appointment Detail Dialog Overview

6.2.2.1.5. Health Analysis

The user can check his/her health analysis by clicking on the “Analysis” tab in the bottom navigation bar. The mobile application will then redirect the user and display each analysis in a graph.

To update the health condition, the user can click on the “Add” floating button. Then, the user is required to select the type of health condition and enter the value for it.

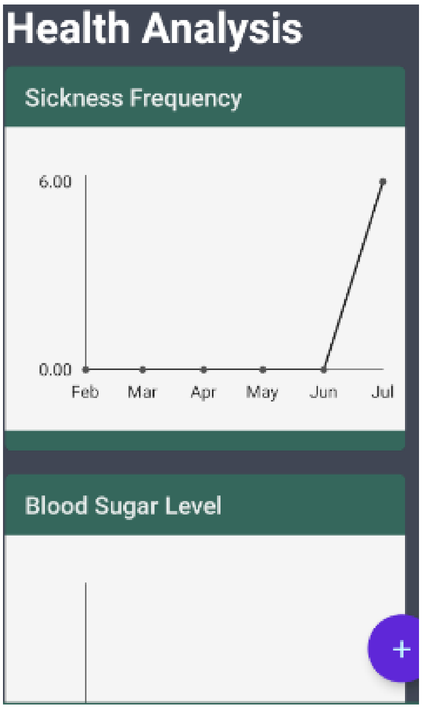
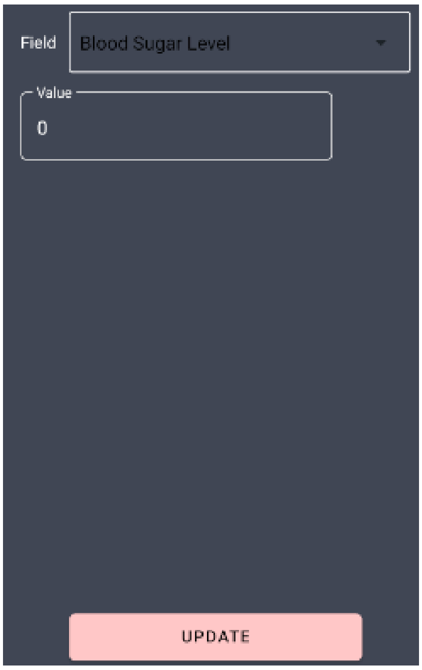


Figure 6.25 Health Analysis Overview



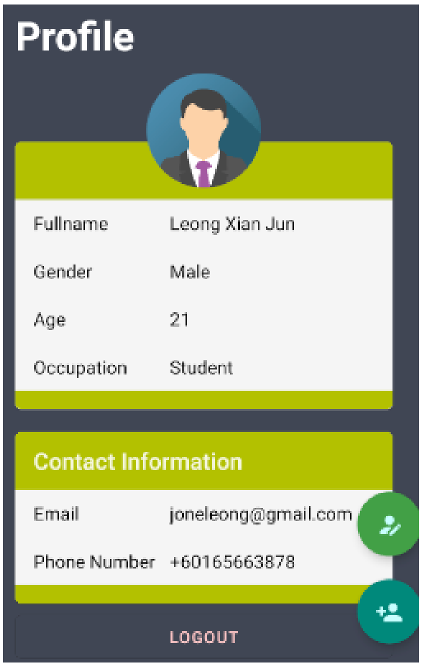
Field Blood Sugar Level

Value 0

Figure 6.26 Page Overview to update Health Condition

6.2.2.1.6. Account Management

To view the profile detail, the user can access it by clicking on the “Profile” tab in the bottom navigation bar. If the user wishes to update the profile information, the user required to press on the “Edit Profile” floating button.



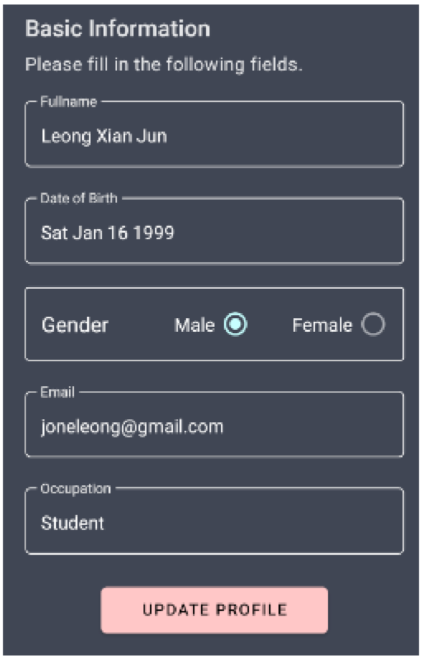
Profile

Fullname	Leong Xian Jun
Gender	Male
Age	21
Occupation	Student

Contact Information

Email	joneleong@gmail.com
Phone Number	+60165663878

Figure 6.27 Profile Page Overview



Basic Information

Please fill in the following fields.

Fullname: Leong Xian Jun

Date of Birth: Sat Jan 16 1999

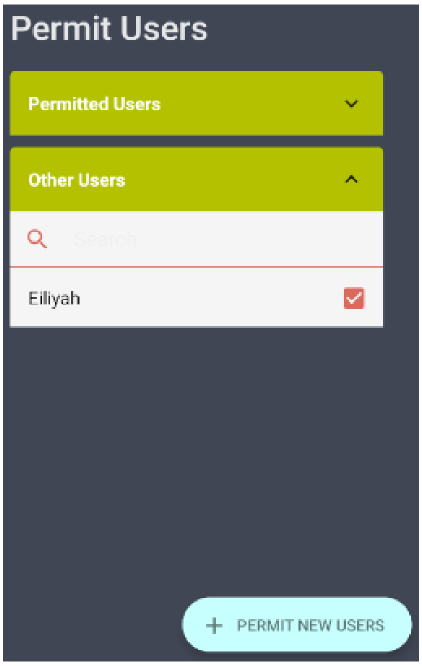
Gender: Male ● Female ○

Email: joneleong@gmail.com

Occupation: Student

Figure 6.28 Page Overview to Update Profile Information

Besides, the user can update his/her authorized user list by clicking on another floating button. This will redirect them to another screen, which lists all of the authorized and non-authorized users. Then, the user can check the user to be authorized and click on the floating button to update the list.



Permit Users

Eilliyah ☑

Figure 6.29 Screen Overview to permit more user for Access Authorization

However, if the user wishes to remove some users from the list, the user required to click on the "Edit" icon located at the top navigation bar. Then, the user is needed to check the user to be removed and click the floating button to submit the removal request.

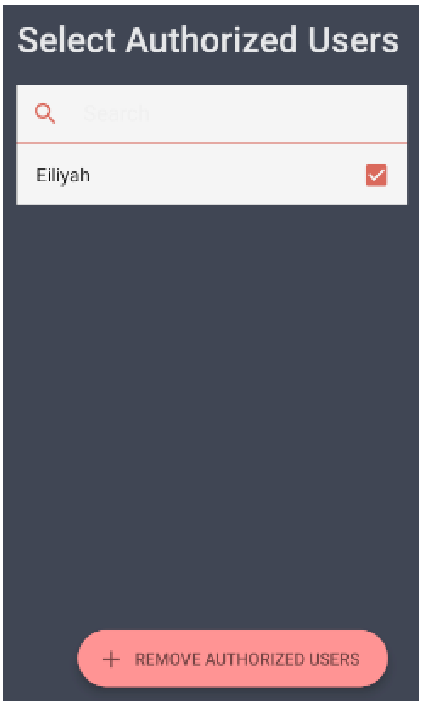


Figure 6.30 Screen Overview to remove Authorized Users from the list

6.2.2.2. Web Application

6.2.2.2.1. Account Registration and Sign In

As a medical staff, he or she is required to create an account via the web application. Upon the successful creation of an account, the medical staff is then required to enter their basic information as well as their working information. Until then, the registration process is completed.

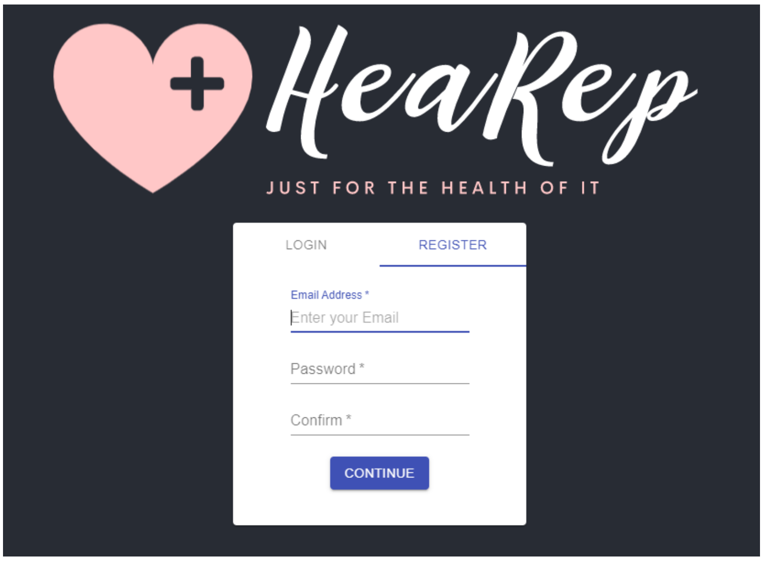
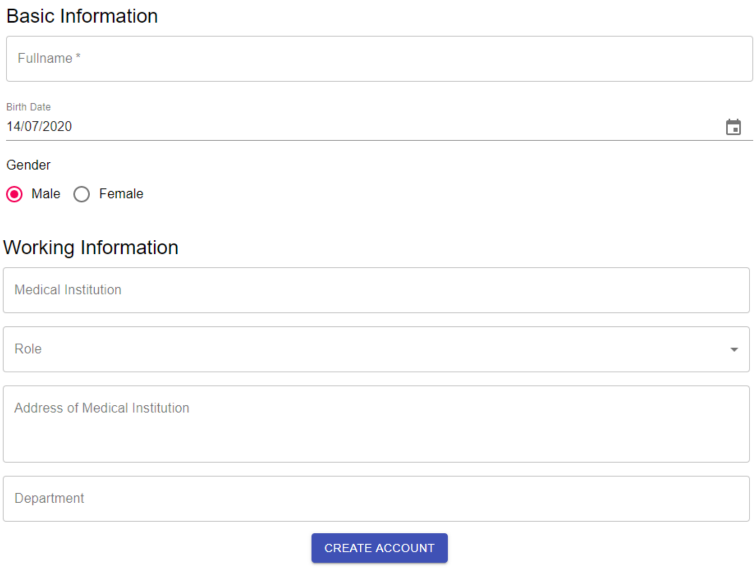


Figure 6.31 Account Registration Overview



Basic Information

Fullname *

Birth Date
14/07/2020

Gender
● Male ○ Female

Working Information

Medical Institution

Role

Address of Medical Institution

Department

Figure 6.32 Account Information Overview

For the consecutive access to the website, the medical staff is required to log in using their created account.

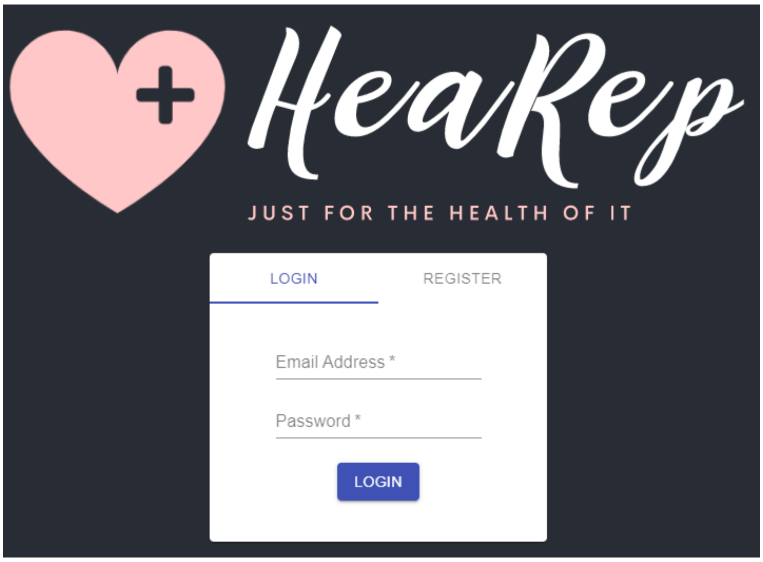
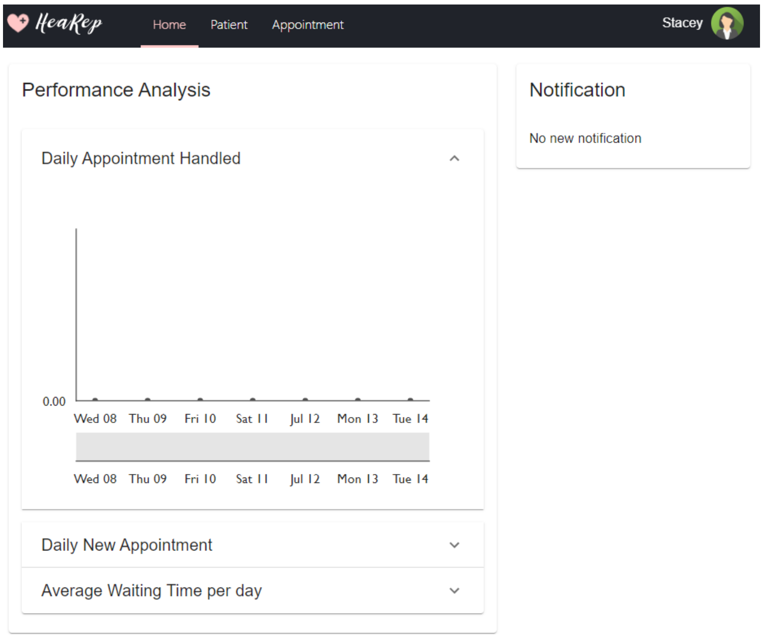


Figure 6.33 Account Login Overview

6.2.2.2.2. Dashboard

After the successful access to the web application, the system will direct the user to the dashboard. In the dashboard, it displays the performance analysis of the user. User can view the graph by expanding each of the analysis tabs.

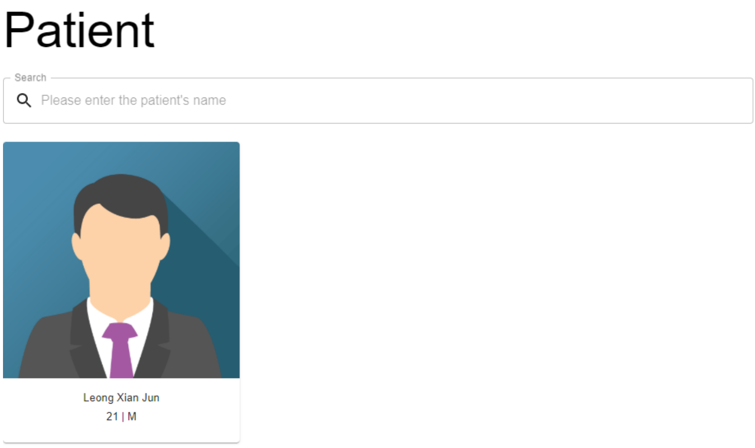


No new notification

Figure 6.34 Dashboard Overview

6.2.2.2.3. Patient and Health Record

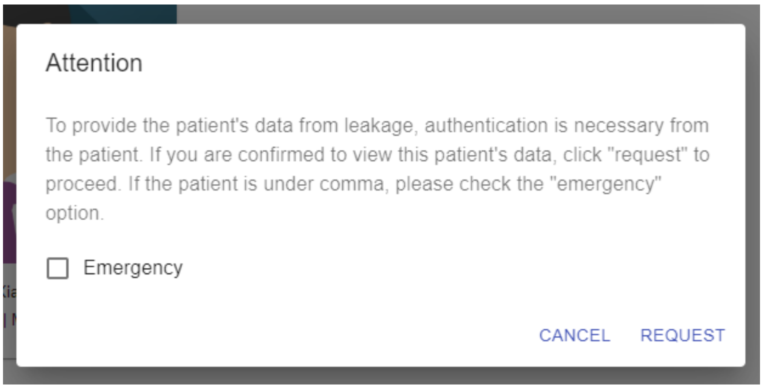
By clicking on the patient tab, the list of patients will be shown. The user can now request access from the user to view the patient's detail. Additionally, the user can filter the patient by entering part of the patient's name in the search bar provided. After being authorized by the patient to view their information, the system will redirect the user to the detail page.



Patient

Leong Xian Jun
21 | M

Figure 6.35 Patient List Overview

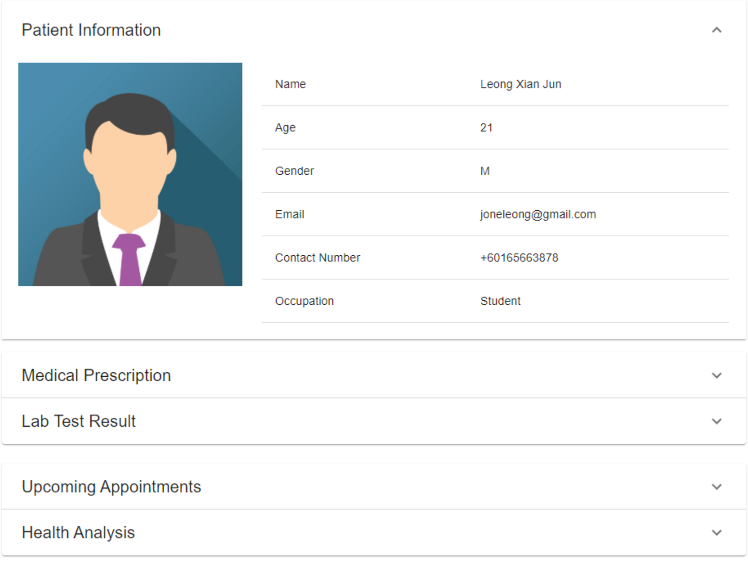


Attention

To provide the patient's data from leakage, authentication is necessary from the patient. If you are confirmed to view this patient's data, click "request" to proceed. If the patient is under comma, please check the "emergency" option.

□ Emergency

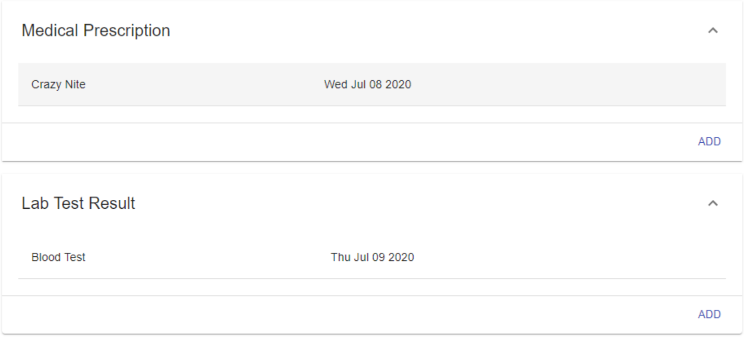
Figure 6.36 Authorization Request Dialog Overview



Name	Leong Xian Jun
Age	21
Gender	M
Email	joneleong@gmail.com
Contact Number	+60165663878
Occupation	Student

Figure 6.37 Patient Detail Page Overview

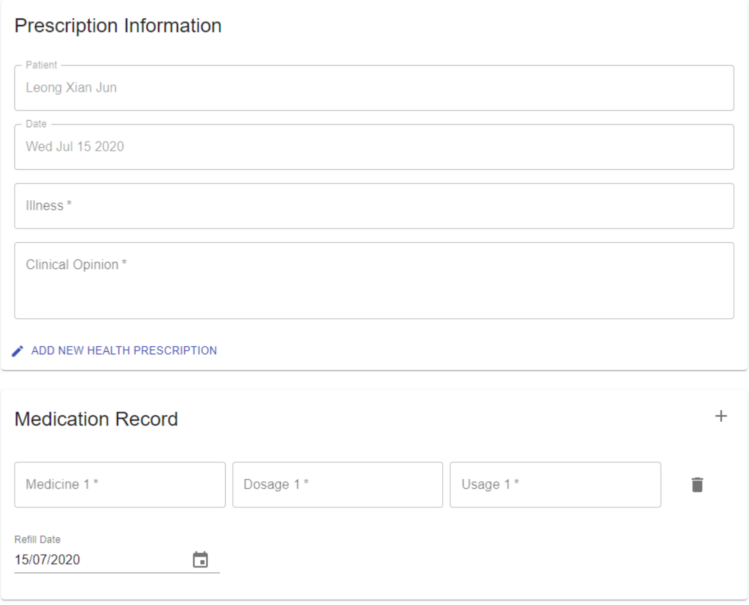
Then, the user can view the patient's health analysis by expanding the health analysis tab. To insert a health record, the user can click on the "Add" button. By clicking the button, the system will redirect the user to either add health prescription or lab test result. Regardless of the selection, the user can fill in several fields and once it is completed, the user can click on the "Insert" button.



Crazy Nite	Wed Jul 08 2020

Blood Test	Thu Jul 09 2020

Figure 6.38 Health Record List Overview



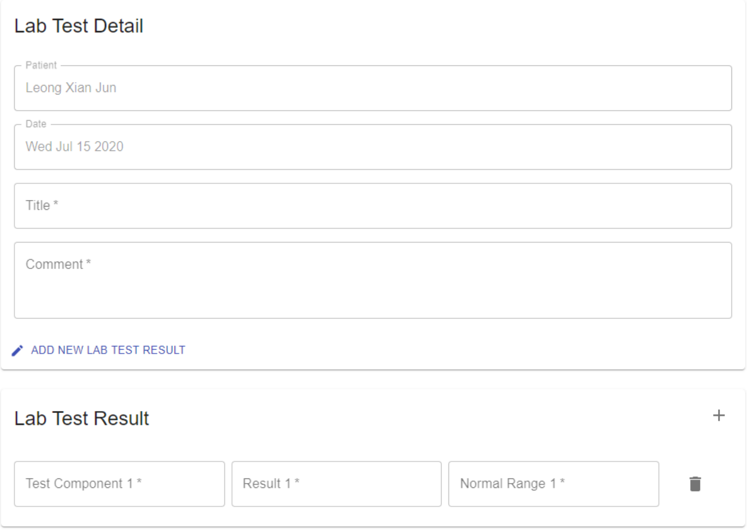
Prescription Information

Patient: Leong Xian Jun

Date: Wed Jul 15 2020

Refill Date: 15/07/2020

Figure 6.39 Page to Add Prescription



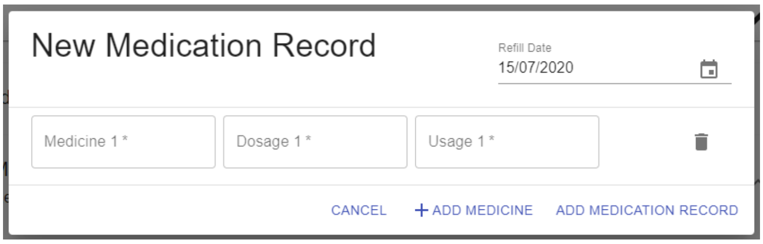
Lab Test Detail

Patient: Leong Xian Jun

Date: Wed Jul 15 2020

Figure 6.40 Page to Add Lab Test Result

To view detail of the health record, the user can simply click on the record from the health record list shown in Figure 6.38. Then, the user gets to view the detail of each record and medication can be done by the user in the detail page. Additionally, the user can insert a medication record for the medical prescription.



New Medication Record

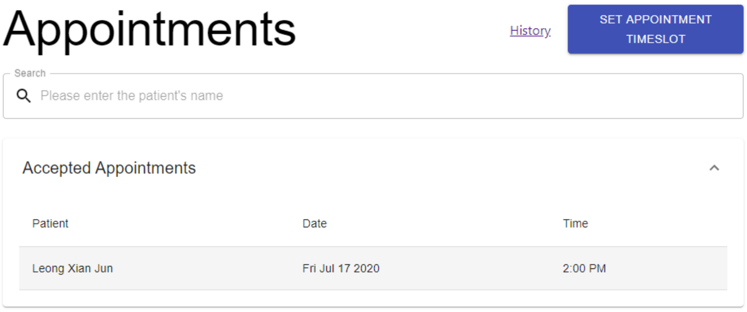
Refill Date: 15/07/2020

Medicine 1 *	Dosage 1 *	Usage 1 *

Figure 6.41 Dialog to add new Medication Record

6.2.2.2.4. Appointment Management

The user can view all of his/her appointment in the appointment tab. The user can choose to accept or reject an appointment. With the accepted appointment, the user can then insert the health record after meeting with the patient.



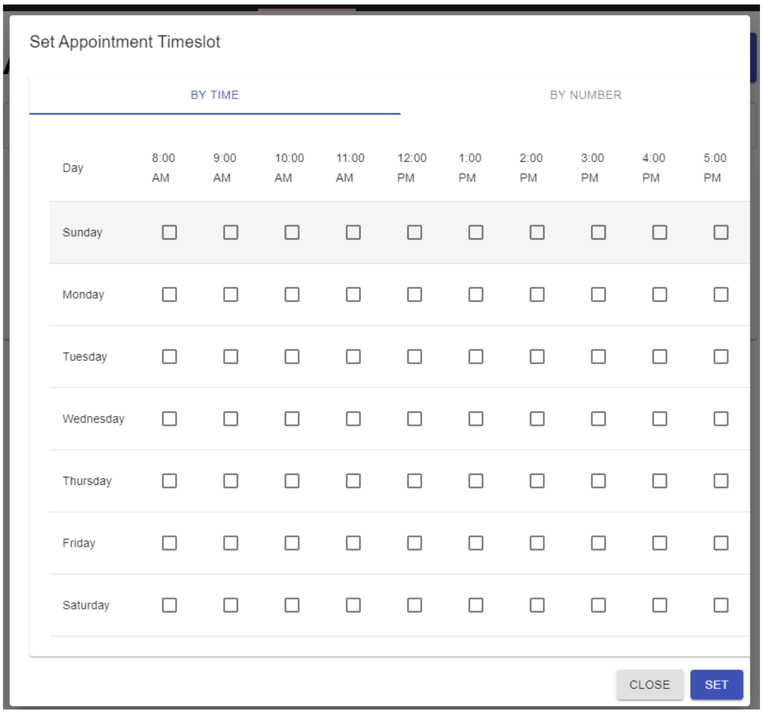
Appointments

Search: Please enter the patient's name

Accepted Appointments

Patient	Date	Time
Leong Xian Jun	Fri Jul 17 2020	2:00 PM

Figure 6.42 Appointment Page Overview

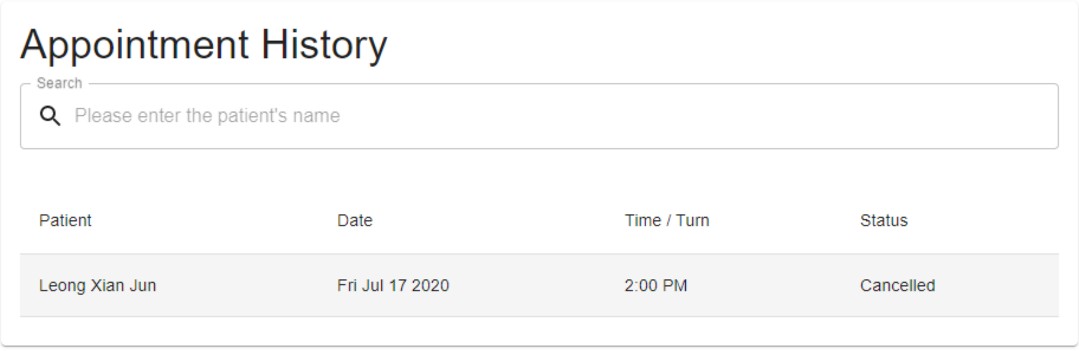


Set Appointment Timeslot

Day	8:00 AM	9:00 AM	10:00 AM	11:00 AM	12:00 PM	1:00 PM	2:00 PM	3:00 PM	4:00 PM	5:00 PM
Sunday	□	□	□	□	□	□	□	□	□	□
Monday	□	□	□	□	□	□	□	□	□	□
Tuesday	□	□	□	□	□	□	□	□	□	□
Wednesday	□	□	□	□	□	□	□	□	□	□
Thursday	□	□	□	□	□	□	□	□	□	□
Friday	□	□	□	□	□	□	□	□	□	□
Saturday	□	□	□	□	□	□	□	□	□	□

Figure 6.43 Dialog to set available timeslots

Other than that, the user can view his/her appointment history by clicking on the "History" link. It displays all of the previous or cancelled appointment.



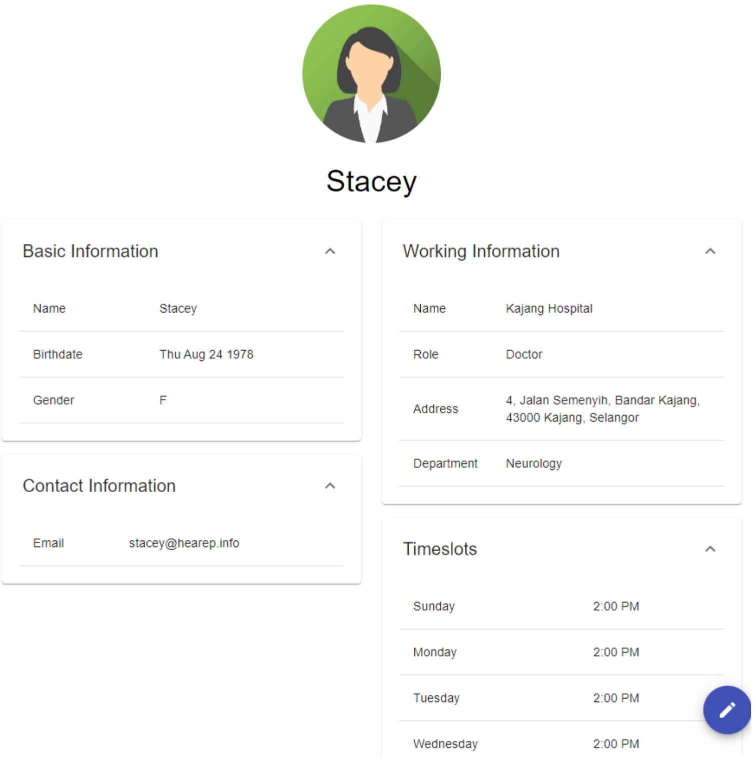
Appointment History

Patient	Date	Time / Turn	Status
Leong Xian Jun	Fri Jul 17 2020	2:00 PM	Cancelled

Figure 6.44 Overview of Appointment History

6.2.2.2.5. Account Management

The users can view and edit their profile. To edit the profile information, the user can click on the edit floating button located at the bottom right of the page. Then, the user can update his/her information and press on the "Update" button to confirm the changes.



Stacey

Basic Information

Name	Stacey
Birthdate	Thu Aug 24 1978
Gender	F

Contact Information

Email	stacey@hearep.info

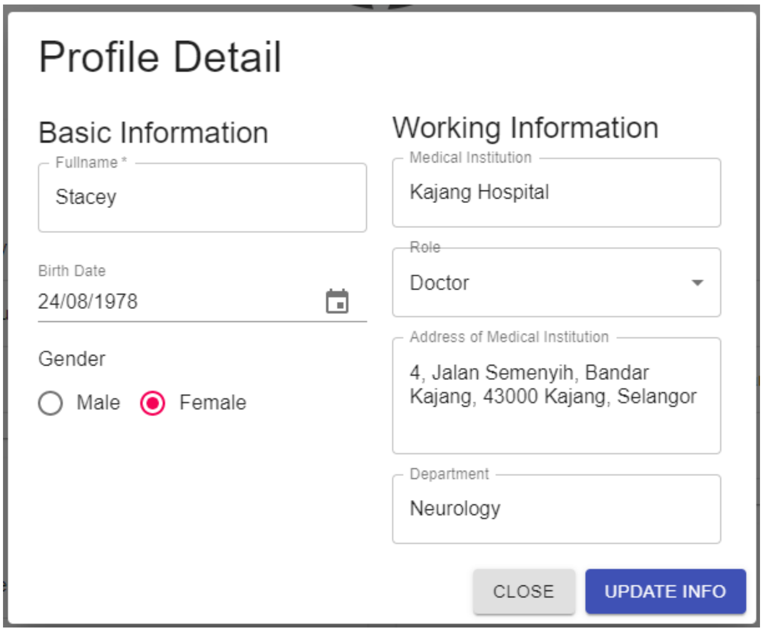
Working Information

Name	Kajang Hospital
Role	Doctor
Address	4, Jalan Semenyih, Bandar Kajang, 43000 Kajang, Selangor
Department	Neurology

Timeslots

Sunday	2:00 PM
Monday	2:00 PM
Tuesday	2:00 PM
Wednesday	2:00 PM

Figure 6.45 Profile Overview



Profile Detail

Basic Information

Fullname * Stacey

Birth Date 24/08/1978

Gender

○ Male ● Female

Working Information

Medical Institution Kajang Hospital

Role Doctor

Address of Medical Institution 4, Jalan Semenyih, Bandar Kajang, 43000 Kajang, Selangor

Department Neurology

Figure 6.46 Dialog to update Profile Information

6.2.3. API List

All of the API is required to have a user token provided by the Firebase Authentication. However, among all of the endpoints, there is one exception, which is endpoint 10: /patient/exist. Moreover, the parameter column showed the content of the input and did not represent the actual attribute naming and data structure in the endpoint. Additionally, not all of the parameter is mandatory for the endpoint.

Table 6-1: API List

No	Endpoint	Description	Parameters
1	/user/get	To fetch the user record	
2	/patient/all	To fetch all of the patient records	
3	/medicalStaff/all	To fetch all of the medical staff records	
4	/user/create	To insert a new user record	username, date of birth, gender, email, type, medical institution, phone number, occupation
5	/user/delete	To remove a user record	
6	/user/update	To update the user record	username, date of birth, gender, email, medical institution, occupation
7	/user/device	To update the device token of the user	device token
8	/user/device/remove	To remove the device token of the user	

No	Endpoint	Description	Parameters
9	/user/authorized/update	To update the patient's list of the authorized user list	List of user ids of the authorized users
10	/user/authorized/remove	To remove the authorized users from the list of the patient	List of user ids of the users to be removed
11	/patient/exist	To check if the user with the phone number exists	phone number
12	/healthrecords/medicalstaff	To fetch all of the health records of a patient by a medical staff	Id of the patient
13	/healthrecords/patient	To fetch all of the patient's health records	
14	/healthrecords/insert	To insert a new health record	Id of the patient, date, type, the id of the appointment, illness, clinical opinion, id of the health prescription, refill date, medications, title, comment, data
15	/healthrecords/delete	To remove a health record	Id of the health record
16	/healthrecords/update	To update a health record	Id of the health record, id of the patient, illness, clinical opinion, refill date, medications, title, comment, data

No	Endpoint	Description	Parameters
17	/healthCondition/option	To fetch the available options of health condition	
18	/healthCondition/update	To update the health condition of the patient	Entry date, option, value
19	/appointment/medicalstaff	To fetch all of the appointments related to the medical staff	
20	/appointment/patient	To fetch all of the patient's appointments	
21	/appointment/get	To fetch an appointment	Id of the appointment
22	/appointment/insert	To create a new appointment record	Id of the medical staff, date, address, type, time or turn
23	/appointment/update	To update the status of an appointment	Id of the appointment, the id of the patient, latest status
24	/appointment/reschedule	To reschedule an appointment	Id of the old appointment, new id of the medical staff, new date, new address, new type, new time or turn
25	/appointment/turn	To get the current number for byNumber appointment	Id of the medical staff, date
26	/appointment/cancel	To cancel an appointment	Id of the medical staff, the id of the appointment

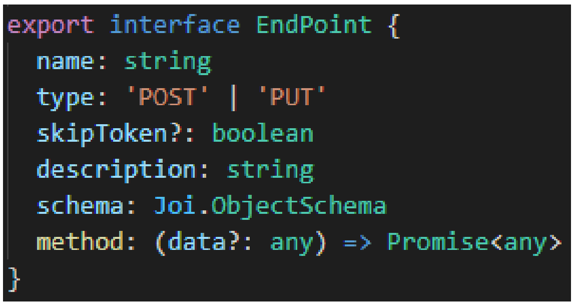
No	Endpoint	Description	Parameters
27	/workingtime/get	to retrieve (1 week) available timeslot of medical staff by a patient	Id of the medical staff, date
28	/workingtime/timeinterval	To get the time interval for working time update	
29	/workingtime/update	To update the working time of a medical staff	Type, slots with day
30	/access/request	To request the authorization from the patient	Id of the patient
31	/access/respond	To respond to the authorization request made by the medical staff	Id of the medical staff, the status of the response
32	/analysis/patient	To fetch the health analysis of a patient	Id of the patient, date
33	/analysis/get	To fetch the performance analysis of the medical staff	date
34	/accessLogs/all	To fetch all of the access logs	

6.2.4. Good Practice

The following practices aided the development so it became smoother. It reduced the chance to have errors including run-time errors. Additionally, good practices or good habits ease the communication between the developers as a standard was made and agreed among the developers.

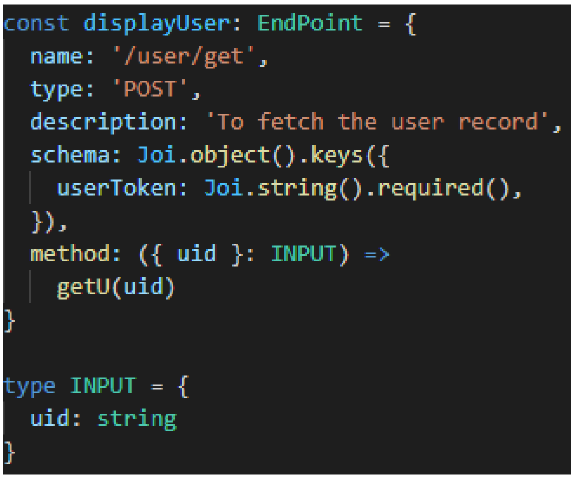
6.2.4.1. Templating

Templating was used in the implementation of the endpoint in the backend server. Each endpoint must consist of several specified elements. By having a template, each endpoint contained the same elements, which prevents the developers from implementing endpoints that lack certain elements.



```
export interface EndPoint {
  name: string
  type: 'POST' | 'PUT'
  skipToken?: boolean
  description: string
  schema: Joi.ObjectSchema
  method: (data?: any) => Promise<any>
}
```

Figure 6.47 Templating Structure for Endpoint



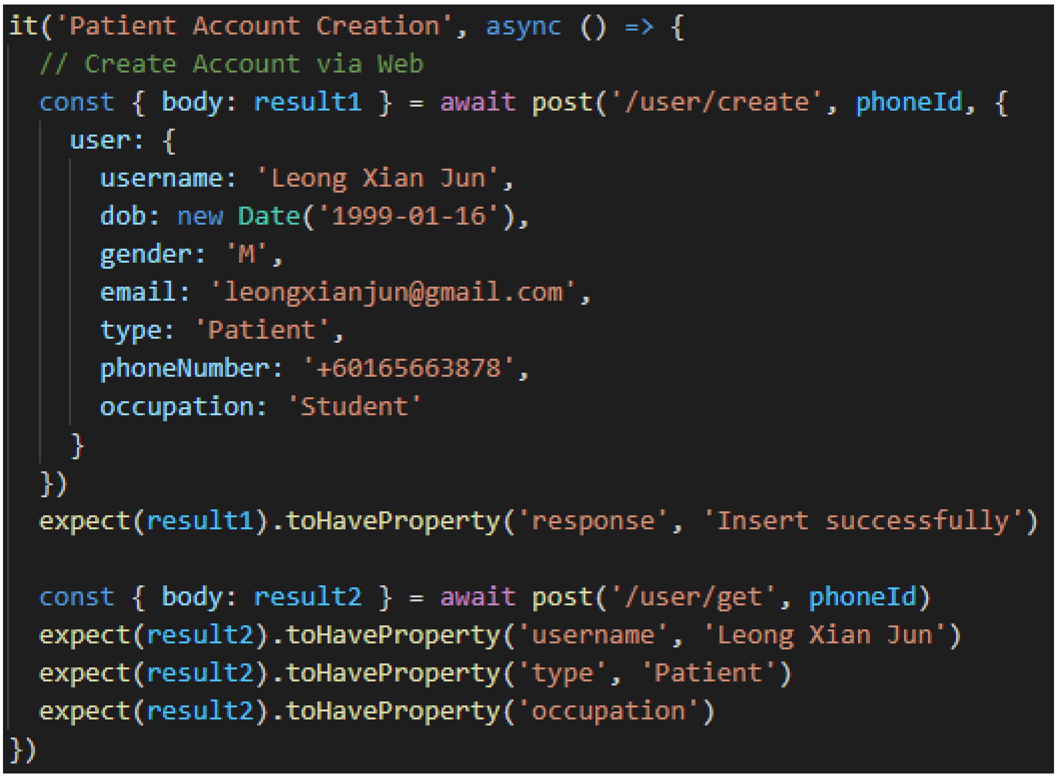
```
const displayUser: EndPoint = {
  name: '/user/get',
  type: 'POST',
  description: 'To fetch the user record',
  schema: Joi.object().keys({
    userToken: Joi.string().required(),
  }),
  method: ({ uid }: INPUT) =>
    getU(uid)
}

type INPUT = {
  uid: string
}
```

Figure 6.48 Code Sample that implemented Templating

In this project, each endpoint shall have five mandatory elements. The developer needs to define its name, type as well as its description. For schema, it was used for data validation. Without a valid set of data, the server would not proceed with the implementation of the endpoint and shall return an error to the client-side. For method, it contained the actual implementation of the endpoint, which also known as the business logic of the process.

6.2.4.2. Writing Automated Testing

Similar to test-driven development, having test cases always aided the developers in producing a better implementation within a shorter time. During the implementation of this project, a test case was written before any endpoint was implemented. With this, the error can be isolated when part of the implementation was delivered wrongly.



```
it('Patient Account Creation', async () => {
  // Create Account via Web
  const { body: result1 } = await post('/user/create', phoneId, {
    user: {
      username: 'Leong Xian Jun',
      dob: new Date('1999-01-16'),
      gender: 'M',
      email: 'leongxianjun@gmail.com',
      type: 'Patient',
      phoneNumber: '+60165663878',
      occupation: 'Student'
    }
  })
  expect(result1).toHaveProperty('response', 'Insert successfully')

  const { body: result2 } = await post('/user/get', phoneId)
  expect(result2).toHaveProperty('username', 'Leong Xian Jun')
  expect(result2).toHaveProperty('type', 'Patient')
  expect(result2).toHaveProperty('occupation')
})
```

Figure 6.49 Sample of Test Case

In each test case, few endpoints were called and each would be expected with certain output. If the actual response is different from the expectation, this indicated that the implementation is incorrect. Thus, changes are required to fix the part that went wrong.

6.3. Testing Execution

In the implementation of this project, three different types of testing were conducted to ensure the system is well developed and able to deliver functionalities to the users as agreed. The testing included are service testing, usability testing and lastly, user acceptance test.

6.3.1. Service Testing

In this project, service testing was conducted to ensure each endpoint is functioning well. This project was tested according to the functionalities of the system. Several endpoints were called within one test case and each matched with an expected response to ensure the entire workflow is functioning. In total, 27 test cases were prepared and executed to test the system.

6.3.1.1. Test Case Listing

Table 6-2: Listing of Service Test Cases

No	Test Case ID	Test Case Title	Status
1	Connection-01	Test Case of Firebase Connection Test	Pass
2	Connection-02	Test Case of FireStore Connection Test	Pass
3	Connection-03	Test Case of Backend Server Connection Test	Pass
4	Connection-04	Test Case of Endpoint Calling Test	Pass
5	User-01	Test Case of Patient Account Creation	Pass
6	User-02	Test Case of Medical Staff Account Creation and Update	Pass
7	User-03	Test Case of Fetching of Non-Existent Account	Pass
8	User-04	Test Case of Fetching Record from Empty User Collection	Pass
9	User-05	Test Case of Fetch all of the Patient Records	Pass
10	User-06	Test Case of Account Removal	Pass
11	User-07	Test Case of Removal of all Accounts	Pass

No	Test Case ID	Test Case Title	Status
12	HealthRecord-01	Test Case of Health Record Insertion, Update and Removal	Pass
13	HealthRecord-02	Test Case of Lab Test Insertion	Pass
14	Appointment-01	Test Case of By-Time Appointment Scheduling Lifecycle	Pass
15	Appointment-02	Test Case of Scheduling Appointment with Medical Staff who has no Working time	Pass
16	Appointment-03	Test Case of By-Time Working Time Update	Pass
17	Appointment-04	Test Case of Appointment Scheduling with not Available Timeslot	Pass
18	Appointment-05	Test Case of Overlapped Appointment Scheduling	Pass
19	Appointment-06	Test Case of By-Number Appointment Scheduling and Cancellation	Pass
20	Appointment-07	Test Case of Invalid Turn Retrieval	Pass
21	Appointment-08	Test Case of Fetching of Current Turn for Appointment Scheduling	Pass
22	Appointment-09	Test Case of Appointment Scheduling outside Operating Hour	Pass
23	Appointment-10	Test Case of Appointment Scheduling on Medical Staff’s Off Day	Pass
24	HealthCondition-01	Test Case of Health Condition Update and Analysis	Pass
25	AuthorizedUser-01	Test Case of Authorized Users Update and Removal	Pass
26	PerformanceAnalysis-01	Test Case of Performance Analysis	Pass
27	AccessLog-01	Test Case of Access Log Checking	Pass

6.3.1.2. Test Case 1: Firebase Connection Test

Table 6-3: Test Case of Firebase Connection Test

Test Case ID	Connection-01	**Status**	Pass
Test Case Title	Firebase Connection Test		
Test Case Summary	It tested the connection established in the backend server to the FireStore.		
Endpoints Involved	-		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Establish a connection to the Firebase with the private keys generated in the Firebase Console.	N/A	The instance ID of the Firebase application is not null.	A connection to Firebase is successfully established.

6.3.1.3. Test Case 2: FireStore Connection Test

Table 6-4: Test Case of FireStore Connection Test

Test Case ID	Connection-02	**Status**	Pass
Test Case Title	FireStore Connection Test		
Test Case Summary	It tested on the connection to FireStore by fetching a set of data from the FireStore		
Endpoints Involved	-		
Post Conditions: Established Firebase Connection			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Called method “tryConnection”, which retrieved data from FireStore database	N/A	The returned item is an array with a size larger than zero.	The backend server managed to retrieve an array of data from the FireStore database.

6.3.1.4. Test Case 3: Backend Server Connection Test

Table 6-5: Test Case of Backend Server Connection Test

Test Case ID	Connection-03	**Status**	Pass
Test Case Title	Backend Server Connection Test		
Test Case Summary	It tested on the connection to the server to ensure the HTTP request is received and responded.		
Endpoints Involved	“/”		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Send the HTTP request with endpoint “/” to the backend server	N/A	An object with property “name” and value of “todoman-backend” is returned.	The backend server managed to handle and respond to the HTTP request sent.

6.3.1.5. Test Case 4: Endpoint Calling Test

Table 6-6: Test Case of Endpoint Calling Test

Test Case ID	Connection-04	**Status**	Pass
Test Case Title	Endpoint Calling Test		
Test Case Summary	It tested on the connection to the server to ensure a particular endpoint is functioning.		
Endpoints Involved	“/try”		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Send HTTP request with endpoint “/try” to the backend server	1) Id of Mobile User	An object with property “result” and value of “success” is returned.	The backend server managed to handle and respond the HTTP request sent as well as validating the input.

6.3.1.6. Test Case 5: Patient Account Creation

Table 6-7: Test Case of Patient Account Creation

Test Case ID	User-01	**Status**	Pass
Test Case Title	Patient Account Creation		
Test Case Summary	It tested on the creation of a patient account.		
Endpoints Involved	“/user/create”, “/user/get”		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) A guest user sends the account information of an account to endpoint “/user/create”	1) Basic User Information	An object with property “response” and value of “Insert successfully” is returned.	The server stored a user record in the database and the data stored has the matched values.
2) The user retrieves its account information from the server via endpoint “/user/get”.	1) The ID of Mobile User	A user object with properties username, type and occupation are returned.	

6.3.1.7. Test Case 6: Medical Staff Account Creation and Update

Table 6-8: Test Case of Medical Staff Account Creation and Update

Test Case ID	User-02	**Status**	Pass
Test Case Title	Medical Staff Account Creation and Update		
Test Case Summary	It tested on the creation of a medical staff account and the profile update of that account. The profile update included the updating of the account information as well as the token of the device that is needed for notification feature.		
Endpoints Involved	"/user/create", "/user/get", "/user/update", "/user/device"		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) A guest user sends the account information of a medical staff account to endpoint "/user/create".	1) Basic User Information 2) Information on the working Medical Institution	An object with property "response" and value of "Insert successfully" is returned.	The server can create medical staff account and update its information correctly. The updated information of the account was stored in the database and its fields matched the values inserted.

2) The user retrieves his/her account information via endpoint “/user/get”.	1) The ID of Web User	A user object with properties username, type and medical institution is returned.	
3) The user updates the account info via endpoint “/user/update”.	1) The ID of Web User 2) New User Information	An object with property “response” and value of “Update successfully” is returned.	
4) User updates the device token field via endpoint “/user/device”.	1) The ID of Web User 2) Token of the device	An object with property “response” and value of “Update successfully” is returned.	
5) The user retrieves the latest account information via endpoint “/user/get”.	1) The ID of Web User	A user object with properties username and token of the device that contained the latest information is returned.	

6.3.1.8. Test Case 7: Fetching of Non-Existent Account

Table 6-9: Test Case of Fetching of Non-Existent Account

Test Case ID	User-03	**Status**	Pass
Test Case Title	Fetching of Non-Existent Account		
Test Case Summary	It tested on the ability of the server to handle request when the user tried to get information of non-existent account.		
Endpoints Involved	"/user/get"		
Post Conditions: A user collection without the record that has the random user ID			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user tried to get random account information via endpoint "/user/get".	1) Random User ID	An object with property "errors" and value of "No such user in the system" is returned.	The system failed to locate the account record from the database and respond to the request with an error.

6.3.1.9. Test Case 8: Fetching Record from Empty User Collection

Table 6-10: Test Case of Fetching Record from Empty User Collection

Test Case ID	User-04	**Status**	Pass
Test Case Title	Fetching Record from Empty User Collection		
Test Case Summary	It tested on the ability of the server to handle request when the user tried to get information from an empty user collection.		
Endpoints Involved	"/user/get"		
Post Conditions: Empty User Collection			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user tried to get account information via endpoint "/user/get".	1) Random User ID	An object with property "errors" and value of "No such user in the system" is returned.	The system failed to locate the account record from the database and respond to the request with an error.

6.3.1.10. Test Case 9: Fetch all of the Patient Records

Table 6-11: Test Case of Fetch all of the Patient Records

Test Case ID	User-05	**Status**	Pass
Test Case Title	Fetch all of the Patient Records		
Test Case Summary	It tested on the ability to retrieve all of the patient records from the user collection.		
Endpoints Involved	"/patient/all"		
Post Conditions: A user collection with at least 1 patient record			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user tried to get all of the patients accounts information via endpoint "/patient/all".	1) The ID of Web User	The returned item is an array with a size larger than one.	The server managed to retrieve data related to the patient account only.

6.3.1.11. Test Case 10: Account Removal

Table 6-12: Test Case of Account Removal

Test Case ID	User-06	**Status**	Pass
Test Case Title	Account Removal		
Test Case Summary	It tested on the ability of the server to soft delete a user account.		
Endpoints Involved	"/user/delete", "/user/get"		
Post Conditions: A user collection with more than 1 records			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user deletes his/her account via endpoint "/user/delete".	1) The ID of the User Account	An object with property "response" and value of "Delete successfully" is returned.	The server deleted the account successfully without removing the account permanently in the database and respond correctly if the account information is requested.
2) The user tried to retrieve information of the deleted account via endpoint "/user/get".	1) The ID of deleted User Account	An object with property "errors" and value of "No such user in the system" is returned.	

6.3.1.12. Test Case 11: Removal of all Accounts

Table 6-13: Test Case of Removal of all Accounts

Test Case ID	User-07	**Status**	Pass
Test Case Title	Removal of all Accounts		
Test Case Summary	It tested on the ability of the server to respond correctly when the user collection has no patient records.		
Endpoints Involved	"/user/delete", "/patient/all"		
Post Conditions: A user collection which has one patient record			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user deletes his/her account via endpoint "/user/delete".	1) The ID of the User Account	An object with property "response" and value of "Delete successfully" is returned.	The server managed to respond correctly when information of all patients is requested while the database has zero patient records.
2) The user tried to retrieve all information of the patient accounts via endpoint "/patient/all".	1) The ID of Web Account	An object with property "errors" and value of "No such user in the system" is returned.	

6.3.1.13. Test Case 12: Health Record Insertion, Update and Removal

Table 6-14: Test Case of Health Record Insertion, Update and Removal

Test Case ID	HealthRecord-01	**Status**	Pass
Test Case Title	Health Record Insertion, Update and Removal		
Test Case Summary	It tested on the creation of health record, the update of the health record and the deletion of the health record from the HealthRecords collection.		
Endpoints Involved	“/healthrecords/insert”, “/healthrecords/medicalstaff”, “/healthrecords/patient”, “healthrecords/update”, “healthrecords/delete”		
Post Conditions: The user collection must contain at least 1 patient user and 1 medical staff user.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Medical staff inserted a health prescription record of the patient via endpoint “/healthrecords/insert”.	1) The ID of the Web User Account 2) Information on the new Health Prescription Record	An object with property “response” and value of “Insert successfully” is returned.	The server managed to insert, update and soft delete the health record from the database correctly. The information inserted into the database matched with the input value provided by the medical staff.

2) Medical staff checked the health record list of the patient via endpoint “/healthrecords/medicalstaff”.	1) The ID of Web User 2) The ID of the Patient	An object with grouped health record list is returned and under health prescription list, it contains one element.	
3) The patient checked his/her health record list via endpoint “/healthrecords/patient”.	1) The ID of Mobile User	An object with grouped health record list is returned and under health prescription list, it contains one element.	
4) The medical staff updated the information of the health prescription via endpoint “/healthrecords/update”.	1) The ID of Web User 2) Latest information on the health prescription	An object with property “response” and value of “Update successfully” is returned.	

5) The medical staff checked the content of the updated health prescription via endpoint “healthrecords/medicalstaff”.	1) The ID of Web User 2) The ID of the Patient	An object with grouped health record list and under health prescription list, it contains one element. The element contains a value that matched with the latest information.	
6) The medical staff inserted a new medication record of the patient via endpoint “/healthrecords/insert”.	1) The ID of Web User 2) Information on the new Medication Record	An object with property “response” and value of “Insert successfully” is returned.	
7) The medical staff checked the latest health record list of the patient via endpoint “/healthrecords/medicalstaff”.	1) The ID of Web User 2) The ID of the Patient	An object with grouped health record list and under health prescription list, it contains one element. This element contains an array of medication records which contains one element.	

8) The medical staff remove the health prescription record of the patient via endpoint "/healthrecords/delete".	1) The ID of Web User 2) The ID of the Health Prescription Record	An object with property "response" and value of "Delete successfully" is returned.	
9) The medical staff checked the health record list of the patient, which is currently empty via endpoint "/healthrecords/medicalstaff".	1) The ID of Web User 2) The ID of the Patient	An object with property "errors" and value of "No more record in the system yet" is returned.	

6.3.1.14. Test Case 13: Lab Test Insertion

Table 6-15: Test Case of Lab Test Insertion

Test Case ID	HealthRecord-02	**Status**	Pass
Test Case Title	Lab Test Insertion		
Test Case Summary	It tested on the ability of the server to insert a health record which is a lab test result.		
Endpoints Involved	"/healthrecords/insert", "/healthrecords/medicalstaff", "/healthrecords/patient"		
Post Conditions: The user collection must contain at least 1 patient user and 1 medical staff user.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The medical staff insert a lab test result of the patient via endpoint "/healthrecords/insert".	1) The ID of Web User 2) Information on the new Lab Test Result	An object with property "response" and value of "Insert successfully" is returned.	The server can insert the lab test result correctly. Retrieval of data correctly grouped the record before sending back to the users.
2) Medical staff checked the health record list of the patient via endpoint "/healthrecords/medicalstaff".	1) The ID of Web User 2) The ID of the Patient	An object with grouped health record list is returned and under "lab test result", it contains one element.	

3) The patient checked his/her health record list via endpoint "/healthrecords/patient".	1) The ID of Mobile User	An object with grouped health record list is returned and under "lab test result", it contains one element.	

6.3.1.15. Test Case 14: By-Time Appointment Scheduling Lifecycle

Table 6-16: Test Case of By-Time Appointment Scheduling Lifecycle

Test Case ID	Appointment-01	**Status**	Pass
Test Case Title	By-Time Appointment Scheduling Lifecycle		
Test Case Summary	It tested on the insertion of the appointment, updating of the appointment status, rescheduling of the appointment, and lastly the closing of an appointment.		
Endpoints Involved	“/appointment/insert”, “appointment/patient”, “appointment/update”, “appointment/reschedule”, “/appointment/medicalstaff”, “/healthrecords/insert”, “/healthrecords/patient”		
Post Conditions: The medical staff had set his/her working time.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Patient scheduled an appointment with medical staff via endpoint “/appointment/insert”.	1) The ID of Mobile User 2) Information on new Appointment	An object with property “response” and value of “Insert successfully” is returned.	

2) Patient checked his/her appointment list via endpoint "/appointment/patient".	1) The ID of Mobile User	An object with grouped appointment list is returned. Under "Pending Appointment", it contains one element.	The server managed to insert, update the status and close an appointment. The inserted information matched the input value as well as the status update made by the medical staff. Additionally, rescheduling of appointment correctly updates the status of the old appointment.
3) The medical staff accept the appointment by updating the status of the appointment via endpoint "/appointment/update".	1) The ID of Web User 2) ID of Appointment 3) Latest Status of the Appointment	An object with property "response" and value of "Update successfully" is returned.	
4) Patient checked his/her appointment list via endpoint "/appointment/patient".	1) The ID of Mobile User	An object with grouped appointment list is returned. Under "Accepted Appointment", it contains one element.	

5) Patient rescheduled the appointment via endpoint “/appointment/reschedule”.	1) The ID of Mobile User 2) The ID of old Appointment 3) Latest Information on the new Appointment	An object with property “response” and value of “Reschedule successfully” is returned.	
6) Medical staff checked his/her appointment list via endpoint “/appointment/medicalstaff”.	1) The ID of Web User	An object with grouped appointment list is returned. Under “Pending Appointment”, it contains one element.	
7) Medical staff accept the appointment again via endpoint “/appointment/update”.	1) The ID of Web User 2) ID of Appointment 3) Latest Status of the Appointment	An object with property “response” and value of “Update successfully” is returned.	

8) After a consultation, medical staff insert a health prescription record into the system via endpoint "/healthrecords/insert".	1) The ID of Web User 2) ID of Appointment 3) Information on Health Prescription Record	An object with property "response" and value of "Insert successfully" is returned.	
9) Patient checked his/her health record list via endpoint "/healthrecords/patient".	1) The ID of Mobile User	An object with grouped health record list is returned and under "Health Prescription", it contains one new record.	

6.3.1.16. Test Case 15: Scheduling Appointment with Medical Staff who has no Working time

Table 6-17: Test Case of Scheduling Appointment with Medical Staff who has no Working time

Test Case ID	Appointment-02	**Status**	Pass
Test Case Title	Scheduling Appointment with Medical Staff who has no Working time		
Test Case Summary	It tested on the ability of the server to handle appointment insertion request if the medical staff did not set his/her working time.		
Endpoints Involved	"/appointment/insert"		
Post Conditions: The medical staff did not set his/her working time.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Patient schedule an appointment with the medical staff via endpoint "/appointment/insert".	1) The ID of Mobile User 2) Information on the new Appointment	An object with property "errors" and value of "This medical staff does not set his/her working time yet" is returned.	The server prevents the insertion of appointment if the medical staff have yet to set their working time.

6.3.1.17. Test Case 16: By-Time Working Time Update

Table 6-18: Test Case of By-Time Working Time Update

Test Case ID	Appointment-03	**Status**	Pass
Test Case Title	By-Time Working Time Update		
Test Case Summary	It tested on the update of the medical staff's working time		
Endpoints Involved	"/workingtime/timeinterval", "/workingtime/update", "/workingtime/get"		
Post Conditions: The medical staff created an account.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Medical staff get the time interval of the system via endpoint "/workingtime/timeinterval".	1) The ID of Web User	An array with ten elements was returned.	The server returned the standard working interval to the users and stored their latest working time correctly in the database. With the latest working time, the server can generate a list of available timeslots for appointment scheduling.
2) Medical staff update the working time (By-Time) via endpoint "/workingtime/update" so they are operating for 6 days per week.	1) The ID of Web User 2) Information on New Working Time	An object with property "response" and value of "Update successfully" is returned.	

3) The patient tried to get the updated working time via endpoint “/workingtime/get”.	1) The ID of Mobile User 2) The ID of Medical Staff 3) Current Date	An array with 6 elements was returned. Each element contains the available slots of the medical staff.	

6.3.1.18. Test Case 17: Appointment Scheduling with not Available Timeslot

Table 6-19: Test Case of Appointment Scheduling with not Available Timeslot

Test Case ID	Appointment-04	**Status**	Pass
Test Case Title	Appointment Scheduling with not Available Timeslot		
Test Case Summary	It tested on the server to ensure appointment with a not available timeslot cannot be inserted successfully.		
Endpoints Involved	“/appointment/insert”		
Post Conditions: The medical staff must set their By-Time working time.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Patient scheduled an appointment with a not available timeslot via endpoint “/appointment/insert”.	1) The ID of Mobile User 2) Information on new Appointment	An object with property “errors” and value of “This medical staff is not available in this timeslot” is returned.	The server handled the wrong request and responded it with a correct error.

6.3.1.19. Test Case 18: Overlapped Appointment Scheduling

Table 6-20: Test Case of Overlapped Appointment Scheduling

Test Case ID	Appointment-05	**Status**	Pass
Test Case Title	Overlapped Appointment Scheduling		
Test Case Summary	It tested on the ability of the server to handle the insertion of a similar appointment.		
Endpoints Involved	"/appointment/insert"		
Post Conditions: The appointment collection contained an appointment record that occupied the same timeslot in this test case.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) Patient schedule an appointment on an occupied timeslot via endpoint "/appointment/insert".	1) The ID of Mobile User 2) Information on the new Appointment	An object with property "errors" and value of "Medical staff has an appointment in this timeslot" is returned.	The server handled the wrong request and responded it with a correct error.

6.3.1.20. Test Case 19: By-Number Appointment Scheduling and Cancellation

Table 6-21: Test Case of By-Number Appointment Scheduling and Cancellation

Test Case ID	Appointment-06	**Status**	Pass
Test Case Title	By-Turn Appointment Scheduling and Cancellation		
Test Case Summary	It tested on the insertion of By-Number Appointment and its cancellation.		
Endpoints Involved	“/workingtime/update”, “/appointment/turn”, “/appointment/insert”, “/appointment/patient”, “/appointment/cancel”, “/appointment/medicalstaff”		
Post Conditions: N/A			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The medical staff updated their working time into By-Number type via endpoint “/workingtime/update”.	1) The ID of Web User 2) Latest Information on By-Number Working Time	An object with property “response” and value of “Update successfully” is returned.	The server handled the insertion of appointment correctly and the status of the appointment matched the actual status. For deletion, the server managed to soft delete the record correctly.

2) The patient fetched the current number of turns that the patient needed to wait before meeting the medical staff via endpoint "/appointment/turn".	1) The ID of Mobile User 2) The ID of the Medical Staff	An object consists of "turn", "startTime" and "endTime" is returned.	
3) The patient inserted an appointment via endpoint "/appointment/insert".	1) The ID of Mobile User 2) Information on the new Appointment	An object with property "response" and value of "Insert successfully" is returned.	
4) The patient checked his/her appointment list via endpoint "/appointment/patient".	1) The ID of Mobile User	An object with grouped appointment lists is returned. Under "Waiting Appointment", it contains one element.	
5) The patient cancelled the newly created appointment via endpoint "/appointment/cancel".	1) The ID of Mobile User 2) The ID of Appointment to be Cancelled	An object with property "response" and value of "Delete successfully" is returned.	

6) The medical staff checked their appointment list via endpoint "/appointment/medicalstaff".	1) The ID of Web User	An object with grouped appointment lists is returned. Under "Cancelled Appointment", it contains one new element.	

6.3.1.21. Test Case 20: Invalid Turn Retrieval

Table 6-22: Test Case of Invalid Turn Retrieval

Test Case ID	Appointment-07	**Status**	Pass
Test Case Title	Invalid Turn Retrieval		
Test Case Summary	It tested on the ability of the server to handle the request when the medical staff did not set their By-Number working time		
Endpoints Involved	"/appointment/turn"		
Post Conditions: The medical staff did not set their By-Number working time.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The patient tried to get the current number of turns of the medical staff for appointment scheduling via endpoint "/appointment/turn".	1) The ID of Mobile User 2) The ID of the Medical Staff	An object with property "errors" and value of "Medical Staff does not offer this service yet" is returned.	The server responded to the request correctly when the medical staff did not set their By-Number working time.

6.3.1.22. Test Case 21: Fetching of Current Turn for Appointment Scheduling

Table 6-23: Test Case of Fetching of Current Turn for Appointment Scheduling

Test Case ID	Appointment-08	**Status**	Pass
Test Case Title	Fetching of Current Turn for Appointment Scheduling		
Test Case Summary	It tested on the ability of the server to send the correct number of turns.		
Endpoints Involved	“/appointment/turn/”		
Post Conditions: The appointments collection contained one By-Number appointment.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The patient tried to get the current number of turns of the medical staff to schedule an appointment via endpoint “/appointment/turn”.	1) The ID of Mobile User 2) The ID of the Medical Staff	An object with property “turns” and value of “1” is returned.	The server managed to calculate the current turn correctly and responded to the users.

6.3.1.23. Test Case 22: Appointment Scheduling outside Operating Hour

Table 6-24: Test Case of Appointment Scheduling outside Operating Hour

Test Case ID	Appointment-09	**Status**	Pass
Test Case Title	Appointment Scheduling outside Operating Hour		
Test Case Summary	It tested on the ability of the server if the current turn is requested and if the appointment is inserted outside of the operating hour of the medical staff.		
Endpoints Involved	"/appointment/turn", "/appointment/insert"		
Post Conditions: The medical staff set the By-Number working time that did not cover the current time.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The patient tried to get the current turn of the medical staff outside of their operating hour via endpoint "/appointment/turn".	1) The ID of Mobile User 2) The ID of Medical Staff	An object with property "errors" and value of "'This medical staff does not operate during this working hour" is returned.	The server handled the invalid request made the users and responded correctly with the expected statement.

2) The patient tried to schedule an appointment with the medical staff outside of their operating hour via endpoint "/appointment/insert".	1) The ID of Mobile User 2) Information on the new Appointment	An object with property "errors" and value of ""This medical staff does not operate during this working hour" is returned.	

6.3.1.24. Test Case 23: Appointment Scheduling on Medical Staff's Off Day

Table 6-25: Test Case of Appointment Scheduling on Medical Staff's Off Day

Test Case ID	Appointment-10	**Status**	Pass
Test Case Title	Appointment Scheduling on Medical Staff's Off Day		
Test Case Summary	It tested on the ability of the server if the current turn is requested and if the appointment is inserted outside of the operating hour of the medical staff.		
Endpoints Involved	"/appointment/turn", "/appointment/insert"		
Post Conditions: The medical staff set the By-Number working time that did not cover the current date.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**

1) The patient tried to get the current turn of the medical staff on their off day via endpoint “/appointment/turn”.	1) The ID of Mobile User 2) The ID of Medical Staff	An object with property “errors” and value of “This medical staff does not operate on this day” is returned.	The server handled the invalid request made the users and responded correctly with the expected statement.
2) The patient tried to schedule an appointment with the medical staff on their off day via endpoint “/appointment/insert”.	1) The ID of Mobile User 2) Information on the new Appointment	An object with property “errors” and value of “The medical staff does not operate on this day” is returned.	

6.3.1.25. Test Case 24: Health Condition Update and Analysis

Table 6-26: Test Case of Health Condition Update and Analysis

Test Case ID	HealthCondition-01	**Status**	Pass
Test Case Title	Health Condition Update and Analysis		
Test Case Summary	It tested on the ability of the server to store the health condition correctly and process the data correctly in generating the health analysis of the patient.		
Endpoints Involved	“/healthCondition/option”, “/healthCondition/update”, “/analysis/patient”		
Post Conditions: The healthRecords collections must contain several records of health prescription.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The patient retrieved the type of health condition via endpoint “/healthCondition/option”.	1) The ID of Mobile User	An array containing a list of health condition type is returned.	The server handled to return the health condition type and store the health condition record in the database correctly. Based on the updated health condition, the server managed to generate the analysis of the patient’s health condition.
2) The patient inserted a new health condition record via endpoint “/healthCondition/update”.	1) The ID of Mobile User 2) Information on new Health Condition Record	An object with property “response” and value of “Insert successfully” is returned.	

3) The patient retrieved the analysis of his/her health condition via endpoint “/analysis/patient”.	1) The ID of Mobile User	An object with processed health condition data is returned. Each element contained either six or seven elements.	

6.3.1.26. Test Case 25: Authorized Users Update and Removal

Table 6-27: Test Case of Authorized Users Update and Removal

Test Case ID	AuthorizedUser-01	**Status**	Pass
Test Case Title	Authorized Users Update and Removal		
Test Case Summary	It tested on the ability of the server to update and remove users from the authorized user list of the patient.		
Endpoints Involved	"/patient/all", "/user/authorized/update", "/user/get", "/user/authorized/remove"		
Post Conditions: The user collection must contain several user records.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The mobile user fetched patient list from the server via endpoint "/patient/all".	1) The ID of Mobile User	An array containing patient data is returned.	

2) The patient updated his/her authorized user list via endpoint "/user/authorized/update".	1) The ID of Mobile User 2) IDs of Authorized Users	An object with property "response" and value of "Update successfully" is returned.	The server managed to update and remove an authorized user from each patient record. The updated authorized user list contains IDs of other users correctly.
3) The patient checked his/her authorized user list via endpoint "/user/get".	1) The ID of Mobile User	A user object with a list of authorized users is returned. The authorized user list contains IDs of all of the newly added authorized users.	
4) The patient updated his/her authorized user list via endpoint "/user/authorized/remove".	1) The ID of Mobile User 2) IDs of Authorized Users	An object with property "response" and value of "Update successfully" is returned.	
5) The patient checked his/her authorized user list via endpoint "/user/get".	1) The ID of Mobile User	A user object with a list of authorized users is returned. The authorized user list did not contain IDs of those newly removed authorized users.	

6.3.1.27. Test Case 26: Performance Analysis

Table 6-28: Test Case of Performance Analysis

Test Case ID	PerformanceAnalsysis-01	**Status**	Pass
Test Case Title	Performance Analysis		
Test Case Summary	It tested on the ability of the server to generate the analysis of the medical staff's performance. Each analysis will contain information on the previous seven days.		
Endpoints Involved	"/analysis/get"		
Post Conditions: The medical staff had several appointments made in the previous seven days.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The medical staff fetched their performance analysis via endpoint "/analysis/get".	1) The ID of Web User	An object containing the amount of newly scheduled appointments, amount of completed appointments and daily average waiting time of each patient is returned.	The server managed to correctly generate the analysis for the previous seven days based on the information inserted or updated.

6.3.1.28. Test Case 27: Access Log Checking

Table 6-29: Test Case of Access Log Checking

Test Case ID	AccessLog-01	**Status**	Pass
Test Case Title	Access Log Checking		
Test Case Summary	It tested on the ability of the server in tracking the person who accessed the health records of the patient.		
Endpoints Involved	“/accessLogs/all”		
Post Conditions: A medical staff accessed the health records of patients several times.			
Test Steps	**Test Data**	**Expected Result**	**Actual Result**
1) The user fetched the list of access logs via endpoint “/accessLogs/all”.	1) The ID of Web User	An array of an object that contains the ID of the medical staff who viewed the health record of the patient and the ID of the patient is returned.	The server managed to track the person who accessed the health record of patients as well as their access time.

6.3.2. Usability Testing

In this project, five users are invited to conduct usability testing. This usability testing was conducted on the patient only. In total, seven scenarios were prepared for the users and they required to conduct the testing according to the scenarios provided, as shown in Table 6-30. At the end of the testing, each user was asked to fill in a satisfaction form.

6.3.2.1. Execution

Every participant conducted the testing under the monitoring of the developer. Should any doubts from the participants, the developers shall provide help to the participants so the testing could be conducted successfully. The entire process of usability testing consists of the following steps.

1. At first, the participants were required to read through all of the seven test scenarios prepared.
2. Then, the participants are required to complete the task as mentioned.
3. The developer was monitoring the execution of the tasks and prepared to answer any questions from the participants.
4. After completed the scenarios, the participants are required to fill in the user satisfaction form.

6.3.2.2. Test Scenario Listing

The following table is the list of test scenario used in the usability test. The sample of user acceptance form is attached in Appendix D and a result sample of the usability test is attached in Appendix E.

Table 6-30: Usability Testing Scenario Listing

No	Test Scenario Title	Test Scenario Description
1	Create an Account	Imagine you are a user who plans to keep track of your health records. Hence, you are required to create an account in the application. With a newly created account, you can only start tracking your health records using the application. Therefore, what would you do to create an account?
2	Schedule an Appointment with the Medical Staff	Before you meet the medical staff, you are required to schedule an appointment with the medical staff. With the agreed date and time, the medical staff may conduct diagnosis on you and health record will be inserted by the medical staff into the system. **<u>Task</u>** 1) You wish to schedule an appointment with the medical staff. 2) You wish to reschedule the appointment due to the incapability to attend the appointment on the stated date and time. Which actions would you take to schedule an appointment? How would you do to reschedule the appointment?

No	Test Scenario Title	Test Scenario Description
3	Keep Track of Health Record	Imagine the medical staff updated your health record after completing the appointment with you, you wish to check the information in the health record. How would you do that?
4	Update your Health Condition	Health condition of the patient can help the medical staff in providing a better diagnosis. Hence, you wish to update your health condition daily. Then, you get to view your health condition in a graph. What would you do to update your health condition? With your updated health condition, where can you view your latest health condition?
5	Update Profile	Imagine you are a user who wishes to update the information of your profile. Before that, you would like to view your profile information in the application to check which information to be updated later. Hence, what would you do to view your profile detail?
6	Update Your Authorized User List	Imagine you are in comma or situation where you cannot permit the access requested by the medical staff. Then, the medical staff have no way to access your health record, as your permission is needed before they can view it. Hence, another person, who has the permission granted by you, to permit the access when you are disabled to permit the access yourself. To do so, what actions do you think are needed to grant permission?

6.3.2.3. Result

After the execution of the usability testing, the result was obtained and tabulated in the following tables. In Table 6-31, it showed the overall satisfaction of all of the participants in the test. Overall, the participants were 83% satisfied with the mobile application despite they had some disfavoured parts in the mobile application. Besides, as shown in Table 6-32, the participants gave a high rating on the usefulness of the mobile application.

Table 6-31: Table of Satisfaction Score

Participant	**Score of Statement**										**Total**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Participant 1	4	1	5	1	5	3	5	1	5	4	85
Participant 2	5	2	5	2	4	1	4	1	4	2	85
Participant 3	5	1	4	1	5	1	4	1	4	2	90
Participant 4	3	2	4	1	4	1	4	2	3	1	77.5
Participant 5	4	2	5	1	3	3	4	2	4	1	77.5
Average Satisfaction Score											83

Table 6-32: Table of Mobile Application Usefulness

	Level of Usefulness of the Application				
	Not Helpful at All **1**	**2**	**3**	**4**	**Very Helpful** **5**
Number of Participants	0	0	1	2	2

Additionally, based on the satisfaction survey form, the favoured and disfavoured parts of all of the participants were generalized into the following table. Some participants mentioned that they favoured certain parts of the system because it provides ease for them. In contrast, some thought that some parts were too troublesome to be used and some may have difficulty to access it.

Table 6-33: Comparison Table between Favoured and Disfavoured Parts in the Mobile Application

Favoured Parts in the System	Disfavoured Parts in the System
• Medicine Reminder • Health Condition Analysis Graph • Appointment Scheduling and Rescheduling	• Update Health Condition • Troublesome of expanding the Notification Tab • Permit Authorized User • Icon with Vague Intention

Despite that, the participants provided some comments and suggestions to improve the mobile application. Some suggestions were critical as they may affect the process flow and the user experience. Thus, these suggestions were implemented after they were reflected to ensure no breaking of the applications would occur. The following list contained the comments and suggestions given by the participants.

1. The application should have an instant update on any changes and after making an appointment.
2. The value of updated health condition should be cleared so re-enter is allowed.
3. User should able to update their health condition with a specific date.
4. Pre-expanded expansion component for easy reading
5. Data required re-opening of the application to be updated.
6. Automatic Calculation of BMI
7. Response for button submission is needed.
8. Standard indication of the health condition should be shown while the users are updating their health condition.
9. More information shall be captured for the appointment scheduled, such as the aim for scheduling the appointment as well as the approximate fees needed for the consultation.
10. Better organization and presentation of the information would be favoured.
11. More health condition options would be extra favoured.

Among all of these listed, item 1, item 2, item 5 and item 7 were resolved partially in this project. For the others, it would be considered as feature improvement in the future development phase.

In short, the usability test provided a positive response from the participants. Most of the participants were satisfied with the UI design as well as the user experiences provided while they were using the mobile application.

6.3.3. User Acceptance Test

The last activity in this project is the execution of the user acceptance test. Five users were invited to perform the user acceptance test. Each user was prepared with a list of test cases and they were required to conduct all of them to make sure the system can provide the agreed functionalities.

During the testing, the developer will not answer any questions asked by the participants unless the participants had no way to continue the testing. Before and after completion of each test scenario, the participants are required to mark down the start time and the end time of the testing respectively.

6.3.3.1. UAT Test Case Listing

The following table shows the list of test cases as well as their status. Each test case consists of several descriptions, which represent the possible activities that can be performed by the user of the application. The result of the user acceptance test attached in Appendix G.

Table 6-34: UAT Test Case Listing

No	Test Case ID	Test Form Index	Tested Module	Test Descriptions
1	UAT-01	F1, F7, F13, F19, F25	Login Account	Able to insert phone number and passcode correctly
				Able to display an error message if the input is invalid
2	UAT-02	F2, F8, F14, F20, F26	Manage Profile	Able to view the profile information
				Able to update the profile information
				Able to log out the account
3	UAT-03	F3, F9, F15, F21, F27	Manage Health Record	Able to view the health record list
				Able to view the detail of the health prescription
				Able to add a reminder for medication refill
				Able to view the detail of the lab test result
4	UAT-04	F4, F10, F16, F22, F28	Manage Health Condition	Able to view the health condition in a graphical approach
				Able to update the health condition
				Able to view the latest health condition graph

No	Test Case ID	Test Form Index	Tested Module	Test Descriptions
5	UAT-05	F5, F11, F17, F23, F29	Manage Appointment	Able to get the list of medical staff
				Able to display an error message if no timeslot is available or it is outside of the medical staff working hour
				Able to pick an available timeslot
				Able to schedule an appointment
				Able to display an error message when insertion of appointment is aborted due to overlapped insertion
				Able to view appointment list
				Able to check the detail of the appointment
				Able to reschedule the appointment
				Able to cancel an appointment
				Able to check the updated appointment list
6	UAT-06	F6, F12, F18, F24, F30	Access Request Authentication	Able to view the authorized user list
				Able to grant data access request to other users
				Able to remove users from the authorized user list
				Able to view the latest list of the authorized users

6.4. Summary

This chapter provides insights on the execution of the development as well as testing of the entire project. Moreover, it provides a manual to guide users to use the mobile and web application respectively. On top of that, some good practices were discussed in this chapter, as it could benefit the development process of a project.

Besides, three types of testing were conducted to ensure the application delivered fulfilled the functional requirement as well as the non-functional requirements. Additionally, it also ensured that the application consists of UI that provide good user experiences.

CHAPTER 7

CONCLUSION

This project is to develop a personal health record application to keep track of the health record of the patient. The expected users of the system are the patients and the medical staff, which include doctor and nurses. The entire development process took seven months for completion. Meanwhile, this chapter demonstrates the achieved objectives, its contribution and the limitation as well as its recommendation for improvements.

7.1. Achievement of Objectives

With the completion of the project, all of the mentioned objectives were achieved with respective features in the system. In total, two objectives were defined and achieved successfully.

The first objective is to develop a mobile health records application that helps patients to track all of their formal health records while monitoring the users who access to these records. With the developed mobile application, the patient can view the health records inserted by the medical staff. Additionally, all of the access requests are stored in the database so the system admin can later track the person who views certain health records.

The second objective is to develop an interoperable health records system in both web and mobile platforms that allows data sharing by August 2020. With both the mobile and web applications, the patients and the medical staff can view the health records in the respective application. The data does not store locally, instead, it is stored at a centralized location, which allows retrieval of these data by either the mobile or web application. On top of that, the development of the project was completed before August 2020.

7.2. Contributions of Project

This project is a development project. During its planning and execution, it adopted the Phased development methodology. The SDLC process was separated into 4 phases, which are planning phase, analysis and design phase, development and testing phase as well as the closing phase. Additionally, the development and testing phase was further separated into three sub-phases.

At the beginning of this project, the potential problem, as well as its background, were studied. A questionnaire was then issued to gather the requirement of this project. With the confirmed requirement, the system architecture, as well as the UI designs, were drafted. The system modelling diagrams were structured as well to illustrate the data structure and the information flow in the system.

Then, the development of the project began with the finalized requirement list, modelling diagrams as well as the UI design. The UI design of the actual product was further refined based on the drafted UI design. Meanwhile, the service testing was conducted to ensure the implementation of the business logic were correctly coded.

Once the product was delivered, usability testing and user acceptance test were conducted to ensure the user satisfied with the mobile application and the mobile application can perform the tasks as agreed in the requirement list. After completion of the testing, the development process moved into the closing phase, which required the completion of the documentation of the project and the demonstration of the project as well as the product.

With the completion of the project, the end product would be able to deliver the following features. The patient would be able to keep track of their health records. Additionally, they can schedule appointments with medical staff via the mobile application. Meanwhile, the medical staff can use the web application to handle their appointments.

Moreover, the system can track the access made by the medical staff when they view the health records of the patients. With this feature, the owner of the health records would be alerted on the records accessing performed by particular medical staff. Lastly, the patient can add a reminder in their calendar to remind them of the medication intake or refill.

Other than that, this project demonstrated the development process of being a full-stack developer. As a full-stack developer, one needed to know the knowledge to program backend and frontend application, database, some design basic concepts, version controlling and ability to work with API. These are not the only attributes to be a full-stack developer but one can discover them in this project.

Moreover, this project demonstrated some good practices that are adopted in development. Defining a practice is similar to defining a standard. With standard or practice, it boosted one’s performance as most of the part in the development are repetitive. Hence, one shall always set their standard or adopt practices while involving in a project.

7.3. Limitations of Project

Undeniably, the project had successfully delivered an end product that fulfilled all of the objectives. However, some limitations have prevented the project from being perfect.

The first limitation is the usage of data encryption in the system. In the current industry, many applications or system are not equipped with this technique when they are sending data between applications or systems. For better security in the system, the system shall encrypt any data before sending it from the backend server to the frontend applications. Then, the hacker or malicious user cannot extract the data by capturing the response given by the backend server after an HTTP request is made. As a medical system, the data is very crucial to the patient, which shall remain private and credential.

The second limitation is the medical staff are required to re-enter data if they were using another system previously. The current system creates extra work for the medical staff, especially those who are working at different medical institutions, as it becomes a redundant work for them. Additionally, it could be one of the reasons if the system has a high drop off rate. Therefore, the system shall have an alternative for the medical staff so it can reduce this redundant work.

Next, the third limitation is the health condition option as well as the health analysis. More options in the health condition analysis can ensure a better diagnosis, which will be performed on the patient. Hence, the patient could receive a more complex diagnosis.

Last but not least, the fourth limitation is the absence of actual medical staff as well as the proper medication institution. Without them, the business logic implemented in the system might not be correct for the medical staff side. The flow of the process may be different from the actual work form. Therefore, it may reduce their working performance due to the unfamiliarity to the new system.

7.4. Recommendations for Future Work

To provide a better service to the public, the system shall undergo continuous development and improvement. For such, the following recommendations were listed to provide a direction for the developers who wish to take over the project. Despite that, the improvement of the system shall not be limited to these suggestions and more directions shall be studied so the hidden improvement can be discovered for more and complete features.

Table 7-1: Table of Recommendations and Improvements

No	Recommendation / Improvement	Description
1	Data encryption for better security while transferring data	The system shall encrypt the response of the HTTP request before sending it back to the client-side. Methods for encryption that can be used include AES and RSA.
2	Enable health records import and export	The system shall enable the medical staff to import or export the health records into or from the system. Thus, it can reduce the need for data re-entry and therefore, improve the working performance of the medical staff.

No	Recommendation / Improvement	Description
3	Better and more complex analysis of the health condition as well as the working performance of the medical staff	The system shall provide more options for the health condition and generate a more complex analysis of it. Then, the medical staff can perform a better diagnosis of the patient as the analysis is more meaningful. For instance, the system could analyse the relationship between the food intake as well as the blood sugar level. With this, the medical staff may provide identify the potential root cause that leads to certain diseases such as diabetes.
4	Involvement of actual medical staff as well as the actual medication institution	The project shall invite and interview the medical staff or the management of the medical institution about their internal process when handling the patient. such sharing can provide a correct direction in delivering a system that is user-friendly to the medical staff in terms of the process flow. Additionally, the correct process flow can reduce the need to perform repeated tasks.
5	Involvement of more edge cases for the service testing	In the current service testing, it contained all of the positive test cases and few edge cases. This is not enough to ensure that the system can be functioning correctly. Especially when API of the system is revealed, all of the input combinations shall be tested to ensure the responses are always expected.

No	Recommendation / Improvement	Description
6	Conduct an In-Depth study on the user experience and user interface design of the applications	Since this is a software engineering project, the user interfaces, as well as the user experience, may not follow the standard. Therefore, an in-depth study shall be conducted to improve it so the user experience can reduce the drop off rate of the system user. Moreover, experience and knowledge from an actual UX/UI designer would be favoured to provide a better application to the user.
7	Caching for better performance in data retrieval	In the current project, every HTTP request will get all of the data from the database. This shall not be an issue if the data size is not large enough. However, as time passes, more data would be stored and time needed to retrieve the same data would be increased tremendously. To reduce the reading time of the same data, the system shall cache the often-fetched data. Thus, the same data can always be sent back to the client-side faster at a lower cost.

REFERENCES

Adair, B., 2019. *Future of Electronic Medical Records | EMR Trends For 2020*. [online] Available at: <https://selecthub.com/medical-software/emr/electronic-medical-records-future-emr-trends/> [Accessed 22 Feb. 2020].

Aliakbarpoor, Y., Comai, S. and Pozzi, G., 2017. Designing a HL7 compatible personal health record for mobile devices. In: *RTSI 2017 - IEEE 3rd International Forum on Research and Technologies for Society and Industry, Conference Proceedings*. Modena: Institute of Electrical and Electronics Engineers Inc.

Anon 2010. *Capzule*. Available at: <https://apps.apple.com/us/app/itoppeople/id386321118> [Accessed 22 Mar. 2020].

Anon 2010. *IntelSys*. Available at: <http://intelsys-solutions.com> [Accessed 22 Mar. 2020].

Anon 2012. *My Medical*. Available at: <https://apps.apple.com/us/app/my-medical/id347860026> [Accessed 22 Mar. 2020].

Anon 2014. *Project and Development Approaches*. [online] University of Waterloo. Available at: <https://uwaterloo.ca/ist-project-management-office/methodologies/project-and-development-approaches> [Accessed 22 Feb. 2020].

Anon 2016. *Iterative Development - Phased Iterative Development Model*. *Andrews Cooper*. Available at: <https://www.andrews-cooper.com/how-we-work/development-strategies/product-development-strategy/phased-iterative-product-development/> [Accessed 27 Feb. 2020].

Anon 2016. *MTBC Apps*. Available at: <https://apps.apple.com/pk/developer/mtbc/id390603874> [Accessed 22 Mar. 2020].

Anon 2017. *GetDoc*. Available at: <https://apps.apple.com/app/getdoc-find-make-appointment/id987814486> [Accessed 22 Mar. 2020].

Anon 2017. *Teleme*. Available at: <https://apps.apple.com/my/app/teleme/id1323899039> [Accessed 22 Mar. 2020].

Anon 2017. *The importance of keeping good medical records*. [online] Junior Doctor. Available at: <https://www.medicalprotection.org/southafrica/junior-doctor/volume-7-issue-1/the-importance-of-keeping-good-medical-records> [Accessed 14 Feb. 2020].

Anon 2018. *The 8 Most Common Root Causes of Medical Errors*. [online] Always Culture. Available at: <https://alwaysculture.com/hcahps/communication-medications/8-most-common-causes-of-medical-errors/> [Accessed 3 Jul. 2020].

Anon 2020. *About | Node.js*. [online] OpenJS Foundation. Available at: <https://nodejs.org/en/about/> [Accessed 28 Mar. 2020].

Anon 2020. *Axure RP 9*. [online] Axure Software Solutions, Inc. Available at: <https://www.axure.com/> [Accessed 28 Mar. 2020].

Anon 2020. *Cloud Firestore | Firebase*. [online] Available at: <https://firebase.google.com/docs/firestore> [Accessed 27 Mar. 2020].

Anon 2020. *DoctorOnCall*. [online] Available at: <https://www.doctoroncall.com.my/> [Accessed 22 Mar. 2020].

Anon 2020. *React*. [online] Facebook Inc. Available at: <https://reactjs.org/> [Accessed 28 Mar. 2020].

Anon 2020. *React Native*. [online] Available at: <https://reactnative.dev/> [Accessed 27 Mar. 2020].

Anon 2020. *SDLC - Iterative Model - Tutorialspoint*. [online] Tutorials Point. Available at: <https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm> [Accessed 22 Mar. 2020].

Anon 2020. *TypeScript - JavaScript that scales*. [online] Microsoft. Available at: <https://www.typescriptlang.org/> [Accessed 28 Mar. 2020].

Anon n.d. *Doctor2U*. Available at: <https://play.google.com/store/apps/details?id=my.doctor2u.client> [Accessed 22 Mar. 2020].

Anon n.d. *Genex*. Available at: <https://play.google.com/store/apps/details?id=com.genexehr.patient> [Accessed 22 Mar. 2020].

Bali, A., Bali, D., Iyer, N. and Iyer, M., 2011. Management of Medical Records: Facts and Figures for Surgeons. *Journal of Maxillofacial and Oral Surgery*, 10(3), pp.199–202.

Biørn-Hansen, A., Grønli, T.M., Ghinea, G. and Alouneh, S., 2019. An Empirical Study of Cross-Platform Mobile Development in Industry. *Wireless Communications and Mobile Computing*, [online] 2019, pp.1–12. Available at: <https://www.hindawi.com/journals/wcmc/2019/5743892/> [Accessed 11 Mar. 2020].

Brito, H., Gomes, A., Santos, A. and Bernardino, J., 2018. JavaScript in mobile applications: React native vs ionic vs NativeScript vs native development. In: *Iberian Conference on Information Systems and Technologies, CISTI*. Caceres: IEEE Computer Society.pp.1–6.

Creswell, J.W. and Creswell, J.D., 2017. *Research Design*. 5th ed. [online] SAGE Publications, Inc. Available at: <https://us.sagepub.com/en-us/nam/research-design/book255675#resources>.

Feng, Q., He, D., Wang, H., Zhou, L. and Choo, K.K.R., 2020. Lightweight Collaborative Authentication with Key Protection for Smart Electronic Health Record System. *IEEE Sensors Journal*, [online] 20(4), pp.2181–2196. Available at: <https://ieeexplore.ieee.org/document/8883229/> [Accessed 6 Mar. 2020].

Gorski, D., 2019. *Are medical errors really the third most common cause of death in the U.S.? (2019 edition) – Science-Based Medicine*. [online] Science-Based Medicine. Available at: <https://sciencebasedmedicine.org/are-medical-errors-really-the-third-most-common-cause-of-death-in-the-u-s-2019-edition/> [Accessed 22 Feb. 2020].

Hammer, R., 2016. *30 Healthcare Statistics That Keep Hospital Executives Up At Night*. [online] ReferralMD. Available at: <https://getreferralmd.com/2016/08/30-healthcare-statistics-keep-hospital-executives-night/> [Accessed 22 Feb. 2020].

Heath, S., 2016. *Top 3 Challenges Limiting Patient Access to Health Data*. [online] Available at: <https://patientengagementhit.com/news/top-3-challenges-limiting-patient-access-to-health-data> [Accessed 22 Feb. 2020].

Hersh, W.R., 1995. The electronic medical record: Promises and problems. *Journal of the American Society for Information Science*, 46(10), pp.772–776.

John, W., 2017. *The Top 5 Benefits of React that Make Life Better*. [online] Available at: <https://www.telerik.com/blogs/5-benefits-of-reactjs-to-brighten-a-cloudy-day> [Accessed 28 Mar. 2020].

Jones, K., 2017. *The problems that occur when health data is not used*. [online] Available at: <https://theconversation.com/the-problems-that-occur-when-health-data-is-not-used-82453> [Accessed 22 Feb. 2020].

Kumar, G. and Bhatia, P.K., 2014. Comparative analysis of software engineering models from traditional to modern methodologies. In: *International Conference on Advanced Computing and Communication Technologies, ACCT*. Institute of Electrical and Electronics Engineers Inc.pp.189–196.

Makary, M.A. and Daniel, M., 2016. Medical error-the third leading cause of death in the US. *BMJ (Online)*, 353.

McCrorie, C., Benn, J., Johnson, O.A. and Scantlebury, A., 2019. Staff expectations for the implementation of an electronic health record system: A qualitative study using normalisation process theory. *BMC Medical Informatics and Decision Making*, [online] 19(1), p.222. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/31727063> [Accessed 6 Mar. 2020].

Okonkwo, C.W. and Huisman, M., 2019. The predicting success factors of mobile applications development: Organizational perspectives. In: *Proceedings of the 12th IADIS International Conference Information Systems 2019, IS 2019*. [online] Utrecht: IADIS Press.pp.11–18. Available at: <https://www.researchgate.net/publication/334581291> [Accessed 9 Mar. 2020].

Oleg, R., 2019. *Node.js is a great runtime environment - and here's why you should use it*. [online] Available at: <https://www.freecodecamp.org/news/what-are-the-advantages-of-node-js/> [Accessed 28 Mar. 2020].

Oliveira, W., Torres, W., Castor, F. and Ximenes, B.H., 2016. Native or Web? A Preliminary Study on the Energy Consumption of Android Development Models. In: *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. [online] Suita, Japan: Institute of Electrical and Electronics Engineers (IEEE).pp.589–593. Available at: <https://ieeexplore-ieee-org.libezp2.utar.edu.my/document/7476678?arnumber=7476678> [Accessed 4 Jul. 2020].

Opralova, A., 2005. *SELECTING A DEVELOPMENT APPROACH*. [online] Available at: <https://www.academia.edu/13239574/SELECTING_A_DEVELOPMENT_APPROACH> [Accessed 26 Feb. 2020].

Patel, V. and Mph, C.J., 2017. Access and Use of Online Medical Records and Technology for Health Needs 1 Trends in Individuals' Access, Viewing and Use of Online Medical Records and Other Technology for Health Needs. *SOURCE: HINTS 4 Cycle*, [online] 4(4). Available at: <https://www.healthit.gov/sites/default/files/page/2019-05/HINTS2018ConsumerDataBrief_0.pdf> [Accessed 22 Feb. 2020].

Q. Huynh, M., Ghimire, P. and Truong, D., 2017. Hybrid App Approach: Could It Mark the End of Native App Domination? *Issues in Informing Science and Information Technology*, 14, pp.049–065.

Rajput, M., 2018. *The Pros and Cons of Using React Native*. [online] Available at: <https://www.mindinventory.com/blog/pros-cons-using-react-native/> [Accessed 27 Mar. 2020].

Schindler, E., 2010. *5 reasons to take a phased approach to software development*. [online] Available at: <http://www.itworld.com/article/2749336/development/5-reasons-to-take-a-phased-approach-to-software-development.html> [Accessed 26 Feb. 2020].

Scott, A.L., 2016. *Phased Development Methodology*. Available at: <https://slideplayer.com/slide/6224634/> [Accessed 27 Feb. 2020].

Shah, R., 2019. *Future Trends of Electronic Medical Records Market with Current Updates 2019-2027 - MarketWatch*. [online] Available at: <https://www.marketwatch.com/press-release/future-trends-of-electronic-medical-records-market-with-current-updats-2019-2027-2019-10-17> [Accessed 22 Feb. 2020].

Shaydulin, R. and Sybrandt, J., 2017. To Agile, or not to Agile: A Comparison of Software Development Methodologies. [online] Available at: <http://arxiv.org/abs/1704.07469> [Accessed 11 Mar. 2020].

Vladimir, Y., n.d. *Medical Records*. Available at: <https://play.google.com/store/apps/details?id=vladimir.yerokhin.medicalrecord&hl=en_US> [Accessed 22 Mar. 2020].

Warren, L.R., Clarke, J., Arora, S. and Darzi, A., 2019. Improving data sharing between acute hospitals in England: An overview of health record system distribution and retrospective observational analysis of inter-hospital transitions of care. *BMJ Open*, [online] 9(12), p.e031637. Available at: <http://bmjopen.bmj.com/lookup/doi/10.1136/bmjopen-2019-031637> [Accessed 6 Mar. 2020].

APPENDICES

Appendix A: Methodology Comparison

Methodology	Strengths	Weaknesses
Waterfall (Shaydulin and Sybrandt, 2017)	• to define a complete set of requirements • inexpensive bug fixing • accurate time estimates	• need a long time for requirement gathering and project planning • inflexibility to requirement changes • manage risk is difficult
V-Shaped Model (Kumar and Bhatia, 2014)	• Early development of test plan • Good for a small project, which the requirements are clearly defined	• Similar to the Waterfall Model • Limited flexibility to requirement changes • The testing phase has no clear steps for issues discovered.
Iterative Model (SDLC - Iterative Model - Tutorialspoint, 2020)	• Development in Parallel • Risk Management • Flexible to changes in requirements	• The design needs to changes as it fails to collect all the client requirements at the beginning
Unified Process (Shaydulin and Sybrandt, 2017)	• emphasize simplicity • flexibility to changes in requirements	
Spiral Model (Kumar and Bhatia, 2014)	• great in risk management • review available at the end of each phase	• does not work well on a small project • require expertise for risk management
Rapid Application Development (Shaydulin and Sybrandt, 2017)	• early user involvement in the development • feedback is incorporated at the early stage	• poor design of system due to the acceptance of short-term functionality and ignorance to the technical debt • the user becomes the obstacle to proceeding the development due to disagreement
Scrum (Shaydulin and Sybrandt, 2017)	• continuous update at a short and regular cadence • better communication between the developers	• quality depends on the project manager, as he/she is the point of contact between the developers and consumers

Methodology	Strengths	Weaknesses
Feature-driven Development (Shaydulin and Sybrandt, 2017)	• featured based iterations allow fast development	• deep refactoring needed if the requirements are changing frequently • do not consider the non-functional qualities that depend on interactions between features
Test-Driven Development (Shaydulin and Sybrandt, 2017)	• Code quality	• productivity declined • distract developers focus on development to conduct code deliverables analysis • heavily depends on testing technology
Joint Application Design (Shaydulin and Sybrandt, 2017)	• contact remained throughout the lifespan of the software process • high user acceptable rate	• not a complete methodology, it just covers the activities in gathering requirements

Appendix B: System Comparison

Table B-1: Comparison of Systems developed in Foreign Countries

No	Features	MTBC Apps (2016)	Capzule (2010)	Medical Records (Vladimir, n.d.)	My Medical (2012)	Genex - Health Records (n.d.)	mHealth (Aliakbarpoor, Comai and Pozzi, 2017)
1	Appointment Scheduling	✓	✗	✗	✗	✓	✗
2	Data Import	✗	✗	✗	✓	✗	✓
3	Data Synchronization	✗	✗	✓[1]	✗	✓	✗
4	Data Visualization	✗	✓	✗	✗	✗	✓
5	Health Rates Measurement	✗	✓	✓	✗	✗	✓
6	Health Records Management	✓[2]	✗	✗	✗	✗	✗
7	Immunization Tracking	✗	✗	✗	✓	✗	✗
8	Medications Reminder	✓	✗	✓	✓	✓	✗
9	Multiple Profiles	✗	✓	✓	✗	✗	✗
10	Payment Handling	✓	✗	✗	✗	✗	✗

[1] The patients can do data synchronization manually only, which may lead to data lost.

[2] The patients can view the records issued by the hospitals that joined MTBC only.

Table B-2: Comparison of Systems developed in Malaysia

No	Features	IntelSys (2010)	Doctor2U (n.d.)	Teleme (2017)	GetDoc (2017)	DoctorOnCall (2020)
1	Ambulance Request	✗	✓	✗	✗	✗
2	Appointment Scheduling	✓	✗	✗	✓	✓
3	Clinic Management System	✓	✗	✗	✗	✗
4	Data Visualization	✗	✗	✗	✗	✗
5	E-Commerce for Medicine	✗	✓	✗	✗	✓
6	Health Records Management	✓[3]	✓[4]	✗	✗	✗
7	Medication Reminder	✗	✗	✓	✗	✗
8	Online Doctor Consultation	✗	✓	✓	✗	✓
9	Records Sharing	✓	✗	✗	✗	✗

[3] The patients have no way to retrieve their health records. It only for medical staff use.

[4] Records limited to those generated by BP.

Appendix C: Result of Questionnaire

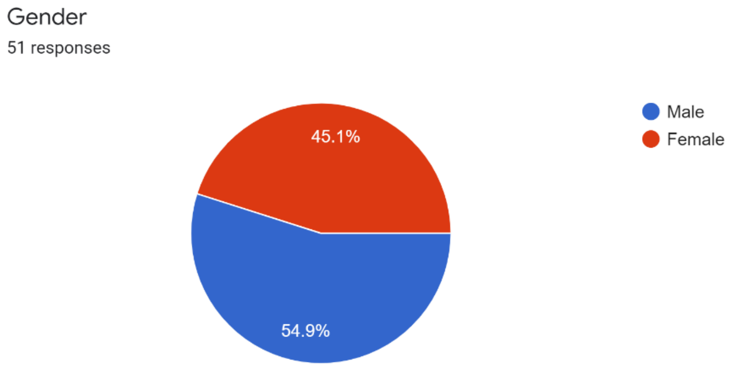


Figure C-1: Gender Distribution

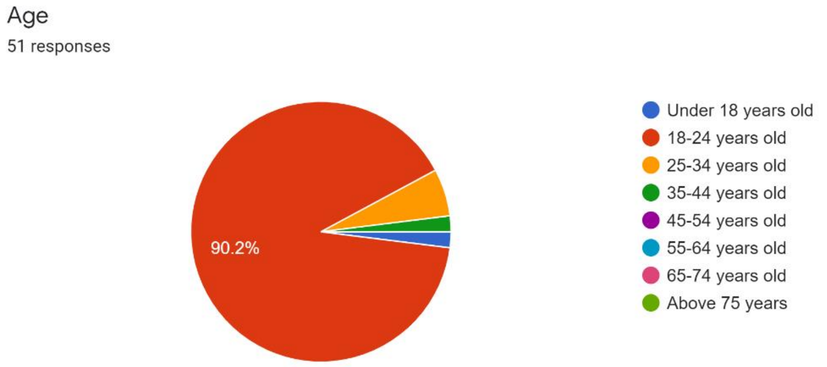


Figure C-2: Age Distribution

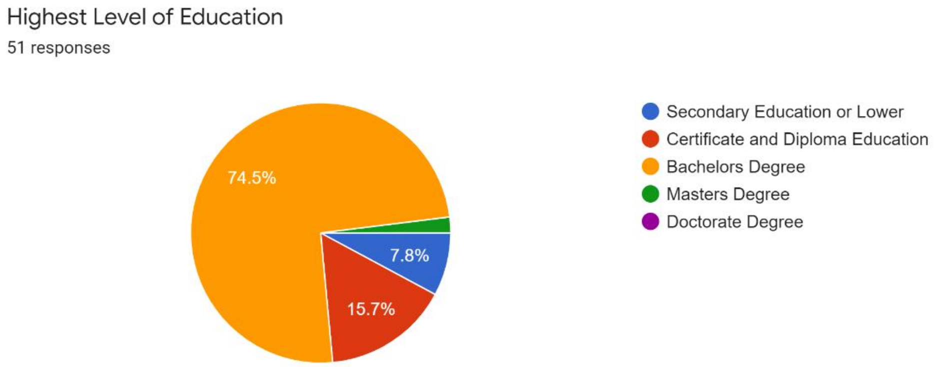
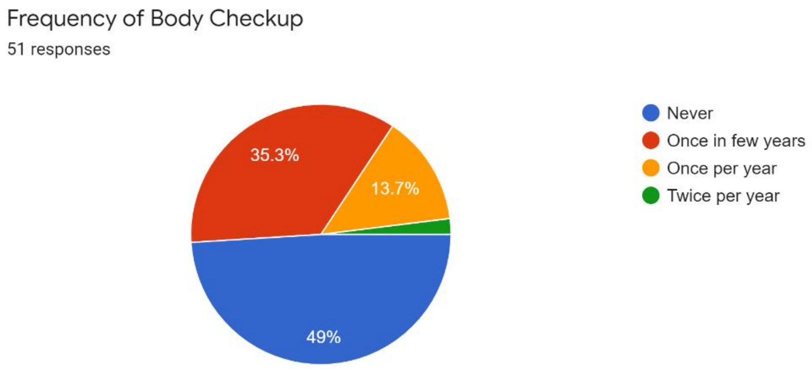


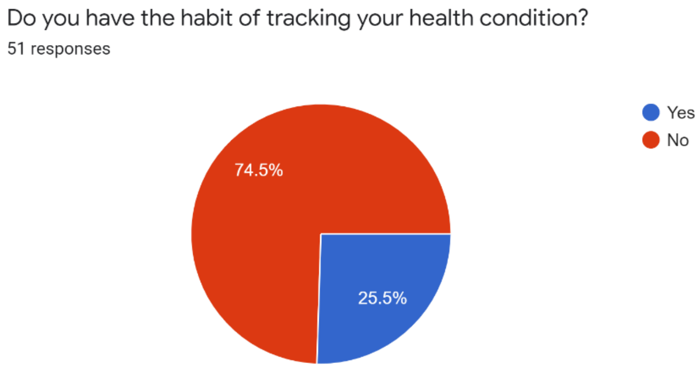
Figure C-3: Distribution of Highest Education Level



Frequency of Body Checkup

51 responses

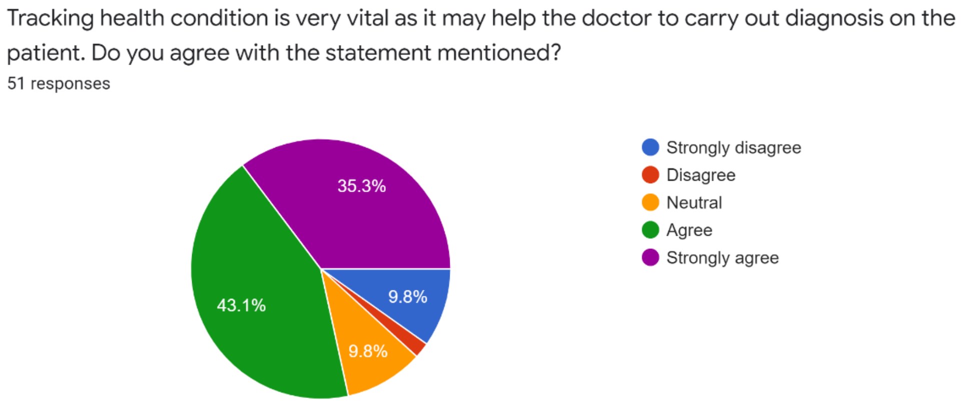
Figure C-4: Frequency of Body Checkup



Do you have the habit of tracking your health condition?

51 responses

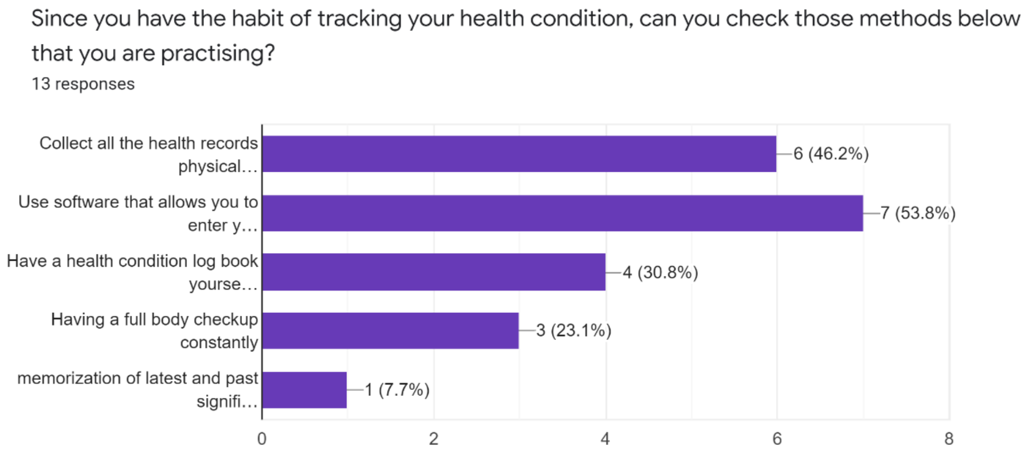
Figure C-5: Distribution of Users who have or have no habit to track their health condition



Tracking health condition is very vital as it may help the doctor to carry out diagnosis on the patient. Do you agree with the statement mentioned?

51 responses

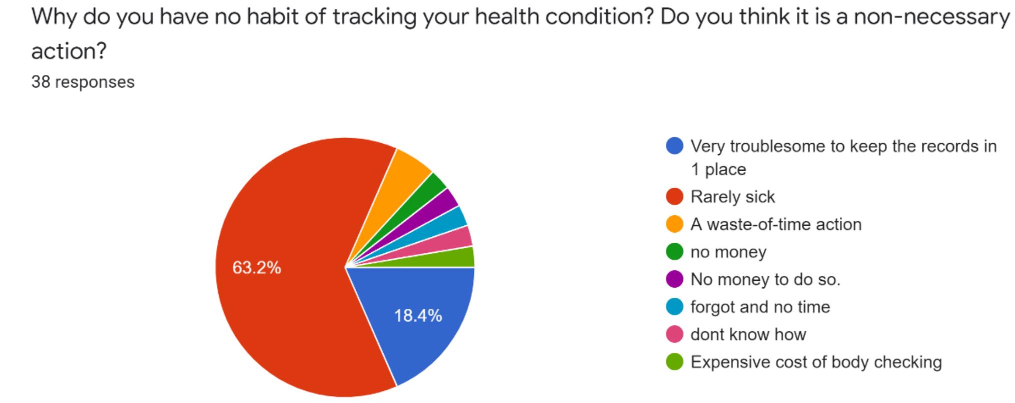
Figure C-6: Distribution of agreement on the importantness of tracking health condition to diagnosis



Since you have the habit of tracking your health condition, can you check those methods below that you are practising?

13 responses

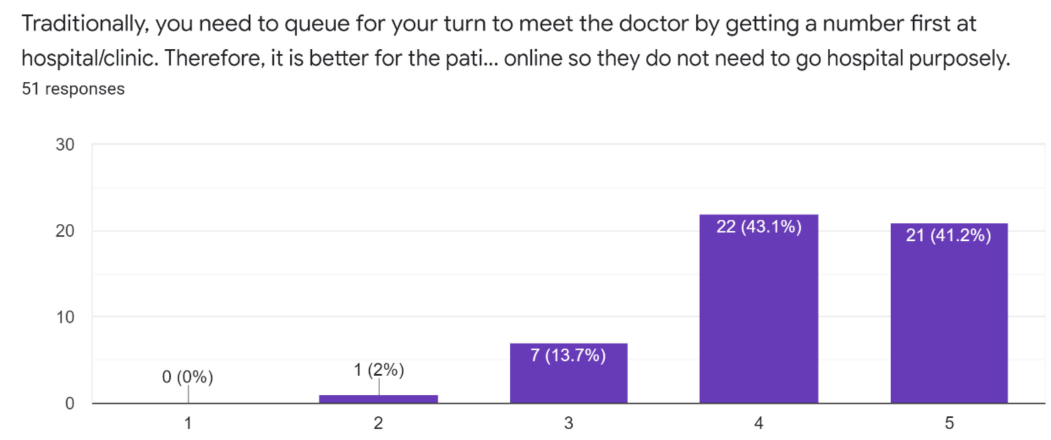
Figure C-7: Frequency of health record tracking habit among the respondents



Why do you have no habit of tracking your health condition? Do you think it is a non-necessary action?

38 responses

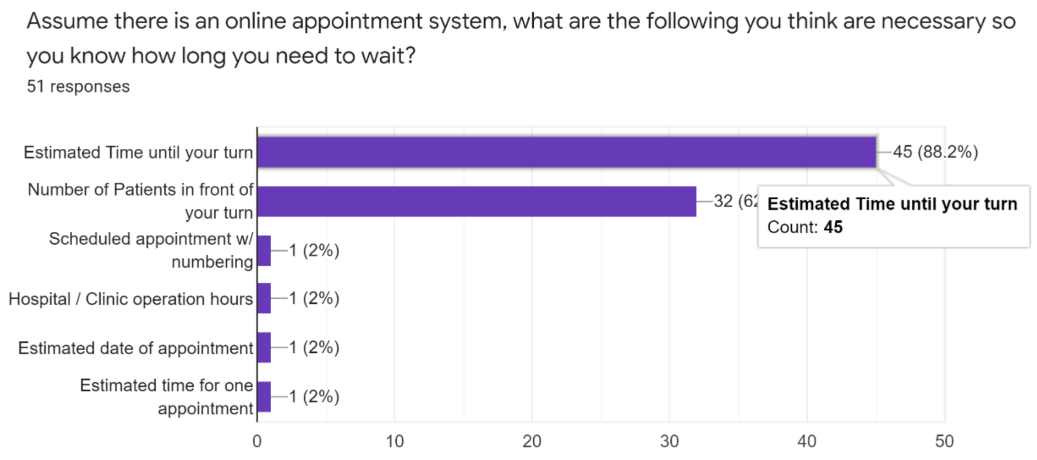
Figure C-8: Distribution of reason for not tracking health condition



Traditionally, you need to queue for your turn to meet the doctor by getting a number first at hospital/clinic. Therefore, it is better for the pati... online so they do not need to go hospital purposely.

51 responses

Figure C-9: Degree of agreement of the necessity to have an online appointment system



Assume there is an online appointment system, what are the following you think are necessary so you know how long you need to wait?

51 responses

Figure C-10: Number of supports on the best approach to show the waiting time

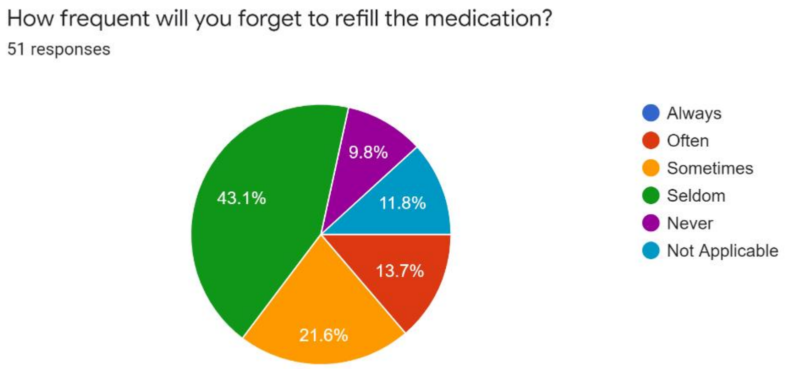


Figure C-11: Level of forgetfulness of the respondents

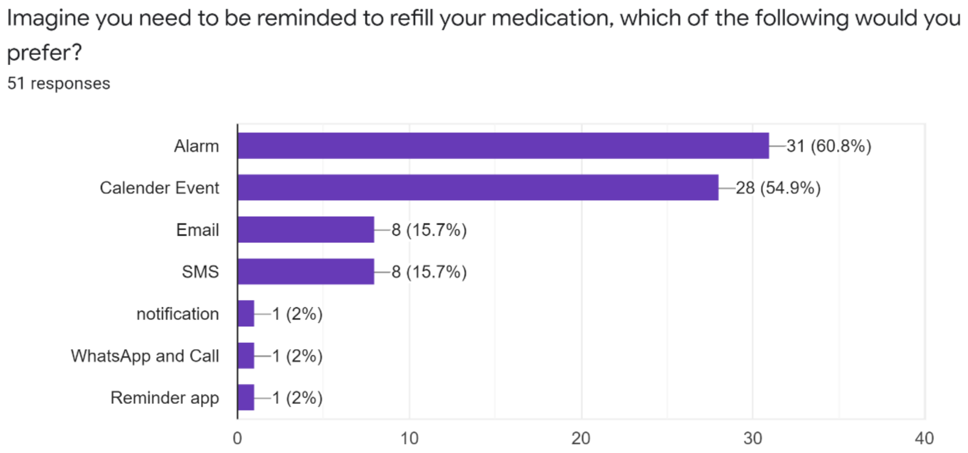


Figure C-12: Number of supports on the best approach to remind respondents on the medication refill

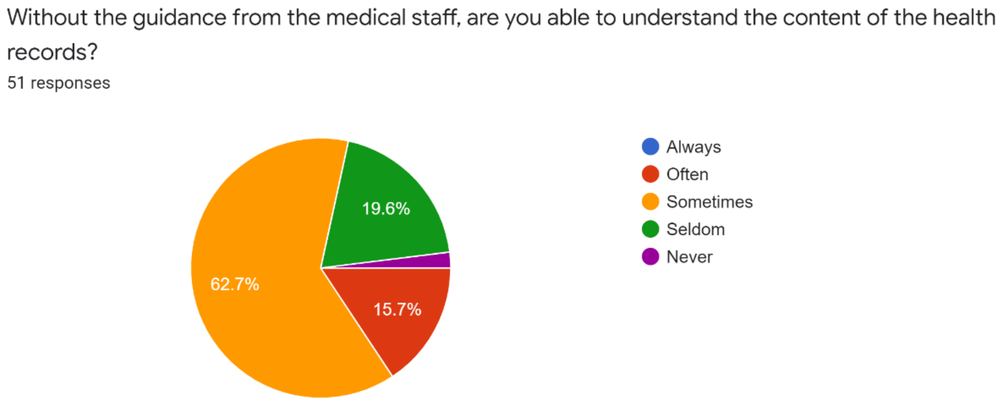


Figure C-13: Distribution of the respondents on the ability to understand the content of health records

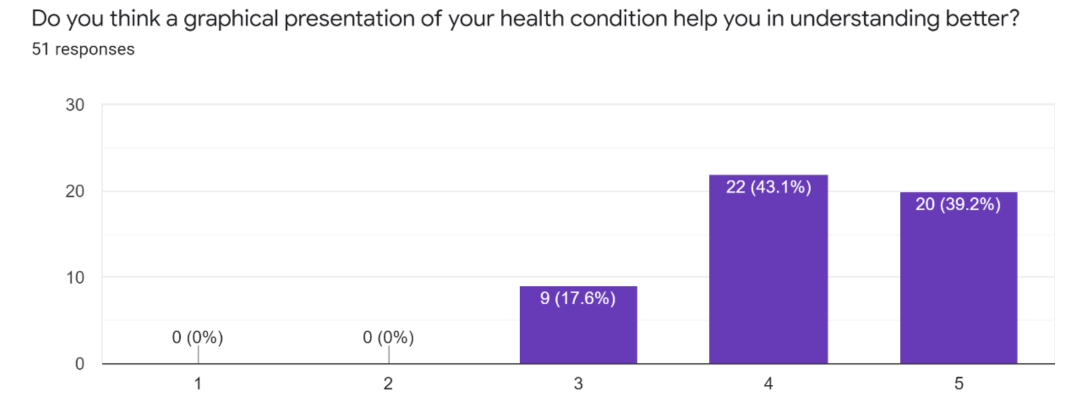


Figure C-14: Number of supports on the necessity to display data in a graphical way

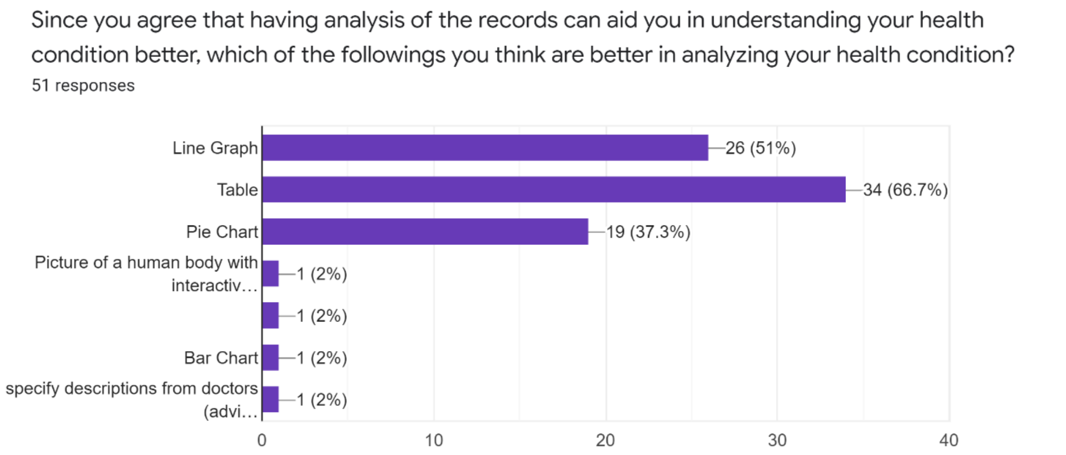


Figure C-15: Number of supports on the best graphical presentation of health condition

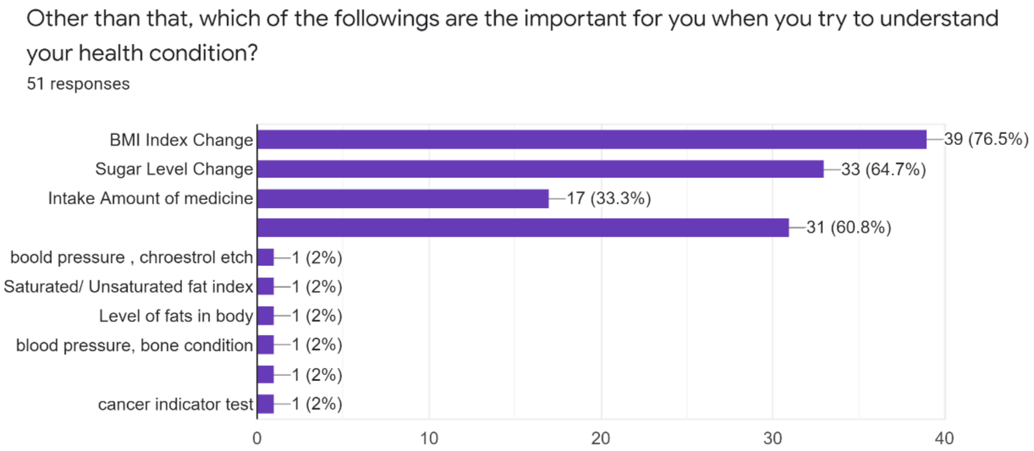


Figure C-16: Number of supports on the most important health condition to be monitored

Appendix D: User Satisfaction Form

Participant #________

Section 1

Please rate the following statements

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
I think that I would like to use this system for tracking my health records.					
I found this is an unnecessarily complex system.					
I thought it is easy for me to use the system.					
Without the support of an expert, I think that I may not able to use the system.					
Without much backtracking or data re-entry, I found this system was easily moved through.					
I thought there was too much inconsistency in this system.					
I would imagine that most people would learn to use this system very quickly.					
I found the system very awkward to use.					
I felt very confident using the system.					

	Strongly Disagree **1**	**2**	**Neutral** **3**	**4**	**Strongly Agree** **5**
Without the learning process, I think I would not be able to get going with this system.					

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	3	4	Very Helpful 5

Q2: Which are the parts in the system you like the most?

___

Q3: Which of the parts in the system you like the least?

___

Q4: What would you comment or question on the system?

___

Appendix E: Result Sample of Usability Test

Participant # 1_______

Section 1

Please rate the following statements

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
I think that I would like to use this system for tracking my health records.				✓	
I found this is an unnecessarily complex system.	✓				
I thought it is easy for me to use the system.					✓
Without the support of an expert, I think that I may not able to use the system.	✓				
Without much backtracking or data re-entry, I found this system was easily moved through.					✓
I thought there was too much inconsistency in this system.			✓		
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very awkward to use.	✓				

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
I felt very confident using the system.					✓
Without the learning process, I think I would not be able to get going with this system.				✓	

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	3	[✓ 4]	Very Helpful 5

Q2: Which are the parts in the system you like the most?

Medicine Calendar Reminder

Q3: Which of the parts in the system you like the least?

Update Health Condition

Q4: What would you comment or question on the system?

The application should have an instant update on any changes and after making an appointment. The value of updated health condition should be cleared so re-enter is allowed. User should able to update their health record by selecting the date.

Participant #2_______

Section 1

Please rate the following statements

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
I think that I would like to use this system for tracking my health records.					✓
I found this is an unnecessarily complex system.		✓			
I thought it is easy for me to use the system.					✓
Without the support of an expert, I think that I may not able to use the system.		✓			
Without much backtracking or data re-entry, I found this system was easily moved through.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very awkward to use.	✓				
I felt very confident using the system.				✓	

	Strongly Disagree **1**	**2**	**Neutral** **3**	**4**	**Strongly Agree** **5**
Without the learning process, I think I would not be able to get going with this system.		✓			

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	3	4	[✓ Very Helpful 5]

Q2: Which are the parts in the system you like the most?

A graph to track the health condition

Q3: Which of the parts in the system you like the least?

Need to manually expand the notification of medications after login

Q4: What would you comment or question on the system?

May be can make the dropdown panel automatically expand when load. Health record will not reload, have to exit the application and log in again. When cancelling the appointment, it will not instantly update the appointment list.

Participant #3________

Section 1

Please rate the following statements

	Strongly Disagree 1	2	Neutral 3	4	Strongly Agree 5
I think that I would like to use this system for tracking my health records.					✓
I found this is an unnecessarily complex system.	✓				
I thought it is easy for me to use the system.				✓	
Without the support of an expert, I think that I may not able to use the system.	✓				
Without much backtracking or data re-entry, I found this system was easily moved through.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very awkward to use.	✓				
I felt very confident using the system.				✓	

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
Without the learning process, I think I would not be able to get going with this system.		✓			

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	3	4	[✓ Very Helpful 5]

Q2: Which are the parts in the system you like the most?

Scheduling feature

Q3: Which of the parts in the system you like the least?

No.

Q4: What would you comment or question on the system?

- Could provide an automatic calculation for BMI
- Some buttons have a delay after pressing it before showing result, perhaps could show some loading icon

Participant #4________

Section 1

Please rate the following statements

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
I think that I would like to use this system for tracking my health records.			✓		
I found this is an unnecessarily complex system.		✓			
I thought it is easy for me to use the system.				✓	
Without the support of an expert, I think that I may not able to use the system.	✓				
Without much backtracking or data re-entry, I found this system was easily moved through.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very awkward to use.		✓			
I felt very confident using the system.			✓		

	Strongly Disagree 1	**2**	**Neutral 3**	**4**	**Strongly Agree 5**
Without the learning process, I think I would not be able to get going with this system.	✓				

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	3	[✓ 4]	Very Helpful 5

Q2: Which are the parts in the system you like the most?

Health Analysis

Q3: Which of the parts in the system you like the least?

Permit Authorized User

Q4: What would you comment or question on the system?

While we are updating our health condition in Health Analysis Function, the system can give a standardized health indication. An example such as Blood Sugar Level Chart, to give the user a glimpse on their current health condition

Participant #5_________

Section 1

Please rate the following statements

	Strongly **Disagree** **1**	**2**	**Neutral** **3**	**4**	**Strongly** **Agree** **5**
I think that I would like to use this system for tracking my health records.				✓	
I found this is an unnecessarily complex system.		✓			
I thought it is easy for me to use the system.					✓
Without the support of an expert, I think that I may not able to use the system.	✓				
Without much backtracking or data re-entry, I found this system was easily moved through.			✓		
I thought there was too much inconsistency in this system.			✓		
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very awkward to use.		✓			
I felt very confident using the system.				✓	

	Strongly Disagree **1**	**2**	**Neutral** **3**	**4**	**Strongly Agree** **5**
Without the learning process, I think I would not be able to get going with this system.	✓				

Section 2

Q1: How helpful is this system to you in tracking your health records?

Not Helpful at All 1	2	[✓ 3]	4	Very Helpful 5

Q2: Which are the parts in the system you like the most?

The function to allow the patient to re-edit the appointment made is very convenient. I like the most about the information regarding the health condition. More information would add extra flavor.

Q3: Which of the parts in the system you like the least?

Some button might need to have labelled as some of it might cause misleading information to the patient. Some of the font sizes are not consistent.

Q4: What would you comment or question on the system?

Capture the aim of making the appointment and the fees used in the consultation would be better. The way of presenting the information can be restructuring and reorganizing.

Appendix F: Sample of User Acceptance Form

Tester No:

Tester's Name:

Date of Testing:

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly			
Able to display an error message if the input is invalid			

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information			
Able to update the profile information			
Able to log out the account			

Test Case 3

Test Case ID	UAT-03	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list			
Able to view the detail of the health prescription			
Able to add a reminder for medication refill			
Able to view the detail of the lab test result			

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Manage Health Condition		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach			
Able to update the health condition			
Able to view the latest health condition graph			

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Manage Appointment		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff			
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour			
Able to pick an available timeslot			
Able to schedule an appointment			
Able to view appointment list			
Able to check the detail of the appointment			
Able to reschedule the appointment			
Able to cancel an appointment			
Able to check the updated appointment list			

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	
Starting Time		**Ending Time**	
Tested Module	Access Request Authentication		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list			
Able to grant data access request to other users			
Able to remove users from the authorized user list			
Able to view the latest list of the authorized users			

Appendix G: Result of User Acceptance Test

Test No	1
Tester's Name:	Chia Yong Fang
Date of Testing:	21/7/2020

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	F1
Starting Time	1:04 pm	**Ending Time**	1:06 pm
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly		Pass	-
Able to display an error message if the input is invalid		Pass	-

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	F2
Starting Time	1.06 pm	**Ending Time**	1.08 pm
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information		Pass	-
Able to update the profile information		Pass	-
Able to log out the account		Pass	-

□Test Case 3

Test Case ID	UAT-03	**Test Form Index**	F3
Starting Time	1.09 pm	**Ending Time**	1.10 pm
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list		Pass	-
Able to view the detail of the health prescription		Pass	-
Able to add a reminder for medication refill		Pass	-
Able to view the detail of the lab test result		Pass	-

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	F4
Starting Time	1.10 pm	**Ending Time**	1.11 pm
Tested Module	Manage Health Condition		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach		Pass	-
Able to update the health condition		Pass	-
Able to view the latest health condition graph		Pass	-

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	F5
Starting Time	1.11 pm	**Ending Time**	1.17 pm
Tested Module	Manage Appointment		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff		Pass	-
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour		Pass	-
Able to pick an available timeslot		Pass	-
Able to schedule an appointment		Pass	-
Able to view appointment list		Pass	-
Able to check the detail of the appointment		Pass	-
Able to reschedule the appointment		Pass	-
Able to cancel an appointment		Pass	-
Able to check the updated appointment list		Pass	-

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	F5
Starting Time	1.20 pm	**Ending Time**	1.22 pm
Tested Module	Access Request Authentication		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list		Pass	-
Able to grant data access request to other users		Pass	-
Able to remove users from the authorized user list		Pass	-
Able to view the latest list of the authorized users		Pass	-

Tester No	2
Tester's Name:	Tan Wei Seng
Date of Testing:	21/7/2020

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	F7
Starting Time	1:04 pm	**Ending Time**	1:08 pm
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly		Pass	-
Able to display an error message if the input is invalid		Pass	-

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	F8
Starting Time	1:08 pm	**Ending Time**	1:09 pm
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information		Pass	-
Able to update the profile information		Pass	-
Able to log out the account		Pass	-

Test Case 3

Test Case ID	UAT-03	**Test Form Index**	F9
Starting Time	1:10 pm	**Ending Time**	1:11 pm
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list		Pass	-
Able to view the detail of the health prescription		Pass	-
Able to add a reminder for medication refill		Pass	-
Able to view the detail of the lab test result		Pass	-

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	F10
Starting Time	1:12 pm	**Ending Time**	1:14 pm
Tested Module	Manage Health Condition		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach		Pass	-
Able to update the health condition		Pass	-
Able to view the latest health condition graph		Pass	-

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	F11
Starting Time	1:15 pm	**Ending Time**	1:24 pm
Tested Module	Manage Appointment		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff		Pass	-
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour		Pass	-
Able to pick an available timeslot		Pass	-
Able to schedule an appointment		Pass	-
Able to view appointment list		Pass	-
Able to check the detail of the appointment		Pass	-
Able to reschedule the appointment		Pass	-
Able to cancel an appointment		Pass	-
Able to check the updated appointment list		Pass	-

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	F12
Starting Time	1:25 pm	**Ending Time**	1:26 pm
Tested Module	Access Request Authentication		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list		Pass	-
Able to grant data access request to other users		Pass	-
Able to remove users from the authorized user list		Pass	-
Able to view the latest list of the authorized users		Pass	-

Tester No	3
Tester’s Name:	Goh Zheng Yi
Date of Testing:	21/ 7 / 2020

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	F13
Starting Time	8:46 PM	**Ending Time**	8:47 PM
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly		Pass	OTP should be hidden
Able to display an error message if the input is invalid		Pass	-

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	F14
Starting Time	8:49 PM	**Ending Time**	8:50 PM
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information		Pass	-
Able to update the profile information		Pass	-
Able to log out the account		Pass	-

Test Case 3

Test Case ID	UAT-03	**Test Form Index**	F15
Starting Time	8:50 PM	**Ending Time**	8:51 PM
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list		Pass	-
Able to view the detail of the health prescription		Pass	-
Able to add a reminder for medication refill		Pass	-
Able to view the detail of the lab test result		Pass	-

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	F16
Starting Time	8:51 PM	**Ending Time**	8:52 PM
Tested Module	Manage Health Condition		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach		Pass	-
Able to update the health condition		Pass	-
Able to view the latest health condition graph		Pass	-

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	F17
Starting Time	8:53 PM	**Ending Time**	8:57 PM
Tested Module	Manage Appointment		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff		Pass	-
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour		Pass	-
Able to pick an available timeslot		Pass	-
Able to schedule an appointment		Pass	-
Able to view appointment list		Pass	-
Able to check the detail of the appointment		Pass	-
Able to reschedule the appointment		Pass	-
Able to cancel an appointment		Pass	-
Able to check the updated appointment list		Pass	-

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	F18
Starting Time	8:57 PM	**Ending Time**	9.00 PM
Tested Module	Access Request Authentication		

Test Descriptions	**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list	Pass	-
Able to grant data access request to other users	Pass	-
Able to remove users from the authorized user list	Pass	-
Able to view the latest list of the authorized users	Pass	-

Tester No 4
Tester's Name: TAN CHANG YONG
Date of Testing: 28/7/2020

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	F19
Starting Time	12:44PM	**Ending Time**	12:46PM
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly		PASS	-
Able to display an error message if the input is invalid		PASS	-

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	F20
Starting Time	12:46PM	**Ending Time**	12:47PM
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information		PASS	-
Able to update the profile information		PASS	-
Able to log out the account		PASS	-

Test Case 3

Test Case ID	UAT-03	**Test Form Index**	F21
Starting Time	12:48PM	**Ending Time**	12:49PM
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list		PASS	-
Able to view the detail of the health prescription		PASS	-
Able to add a reminder for medication refill		PASS	-
Able to view the detail of the lab test result		PASS	-

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	F22
Starting Time	12:50PM	**Ending Time**	12:50PM
Tested Module	Manage Health Condition		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach		PASS	-
Able to update the health condition		PASS	-
Able to view the latest health condition graph		PASS	-

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	F23
Starting Time	12:51PM	**Ending Time**	12:52PM
Tested Module	Manage Appointment		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff		PASS	-
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour		PASS	-
Able to pick an available timeslot		PASS	-
Able to schedule an appointment		PASS	-
Able to view appointment list		PASS	-
Able to check the detail of the appointment		PASS	-
Able to reschedule the appointment		PASS	-
Able to cancel an appointment		PASS	-
Able to check the updated appointment list		PASS	-

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	F24
Starting Time	12:54PM	**Ending Time**	12:55PM
Tested Module	Access Request Authentication		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list		PASS	-
Able to grant data access request to other users		PASS	-
Able to remove users from the authorized user list		PASS	The header blocked the right side of the icon, so it is hard to find the icon
Able to view the latest list of the authorized users		PASS	-

Tester No	5
Tester's Name:	Chong Kee Yew
Date of Testing:	29/07/2020

Test Case 1

Test Case ID	UAT-01	**Test Form Index**	F25
Starting Time	29/07/2020 00:06	**Ending Time**	29/07/2020 00:06
Tested Module	Login Account		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to insert phone number and passcode correctly		Pass	-
Able to display an error message if the input is invalid		Pass	-

Test Case 2

Test Case ID	UAT-02	**Test Form Index**	F26
Starting Time	29/07/2020 00:55	**Ending Time**	29/07/2020 00:57
Tested Module	Manage Profile		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the profile information		Pass	-
Able to update the profile information		Pass	-
Able to log out the account		Pass	-

Test Case 3

Test Case ID	UAT-03	**Test Form Index**	F27
Starting Time	29/07/2020 00:57	**Ending Time**	29/07/2020 00:59
Tested Module	Manage Health Record		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the health record list		Pass	-
Able to view the detail of the health prescription		Pass	-
Able to add a reminder for medication refill		Pass	The button for add reminder is very shallow and it looks like disable on the first sight.
Able to view the detail of the lab test result		Pass	-

Test Case 4

Test Case ID	UAT-04	**Test Form Index**	F28
Starting Time	29/07/2020 00:59	**Ending Time**	29/07/2020 1:01
Tested Module	Manage Health Condition		

Test Descriptions	**Status (Pass / Fail)**	**Comments**
Able to view the health condition in a graphical approach	Pass	-
Able to update the health condition	Pass	-
Able to view the latest health condition graph	Pass	-

Test Case 5

Test Case ID	UAT-05	**Test Form Index**	F29
Starting Time	29/07/2020 1:01	**Ending Time**	29/07/2020 1:04
Tested Module	Manage Appointment		

Test Descriptions	**Status (Pass / Fail)**	**Comments**
Able to get the list of medical staff	Pass	-
Able to display an error message if no timeslot is available or it is outside of the medical staff working hour	Pass	-
Able to pick an available timeslot	Pass	-
Able to schedule an appointment	Pass	-
Able to view appointment list	Pass	-
Able to check the detail of the appointment	Pass	-
Able to reschedule the appointment	Pass	-
Able to cancel an appointment	Pass	-
Able to check the updated appointment list	Pass	-

Test Case 6

Test Case ID	UAT-06	**Test Form Index**	F30
Starting Time	29/07/2020 1:04	**Ending Time**	29/07/2020 1:06
Tested Module	Access Request Authentication		
Test Descriptions		**Status (Pass / Fail)**	**Comments**
Able to view the authorized user list		Pass	-
Able to grant data access request to other users		Pass	-
Able to remove users from the authorized user list		Pass	-
Able to view the latest list of the authorized users		Pass	-