MALAYSIA BLOOD BANK

ANDROID MOBILE APPLICATION

LIM YI CI

BACHELOR OF COMPUTER SCIENCE

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : LIM YI CI

Date of Birth :

Title : MALAYSIA BLOOD BANK ANDROID MOBILE APPLICATION

Academic Session : SEMESTER II ACADEMIC SESSION 2021/2022

I declare that this thesis is classified as:

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

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

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

Certified by:

_____ (Student's Signature)

New IC/Passport Number
Date: 12 MARCH 2022

_____ (Supervisor's Signature)

DR. NABILAH FILZAH BINTI MOHD RADZUAN
Date: 10/02/2023

NOTE : * IF THE THESIS IS CONFIDENTIAL OR RESTRICTED, PLEASE ATTACH A THESIS DECLARATION LETTER.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

_____
(Supervisor's Signature)

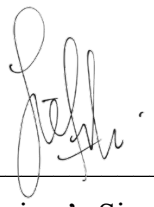
Date: 12 MARCH 2022

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in _____.



(Supervisor's Signature)

Full Name : DR. NABILAH FILZAH BINTI MOHD RADZUAN
Position : Senior Lecturer
Date : 10/02/2023

(Co-supervisor's Signature)

Full Name :

Position :

Date :

STUDENT'S DECLARATION

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

(Student's Signature)

Full Name : LIM YI CI

ID Number : CB19093

Date : 12 MARCH 2022

MALAYSIA BLOOD BANK
ANDROID MOBILE APPLICATION

LIM YI CI

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

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

ACKNOWLEDGEMENTS

I would like to thank my parents for providing a supportive atmosphere during the pandemic time as I developed this thesis.

I also want to express my gratitude to Dr. Nabilah Filzah Binti Mohd Radzuan for agreeing to serve as my supervisor. I now have a greater grasp of the entire development process, the documentation required, and the modifications that must be made to the application thanks to my supervisor's help and advise.

Finally, I'd want to thank everyone who participated in the testing process.

ABSTRACT

A person's life can be saved in large part by blood. Blood comes in four different types: A, B, O, and AB. There are a total of eight blood types, with each group having an option to be RhD positive or RhD negative. The only option for medical facilities and governmental agencies to offer a secure blood supply to individuals in need is through blood donations. It might be difficult to find appropriate donors in a short amount of time.Following the introduction of the MCO in Malaysia, the number of blood bags at the National Blood Centre and other blood banks around the country fell by 40%, from 111,328 bags in May 2018 to 67,135 bags (Mac-May 2020). Accordingly, the purpose of this project is to develop a blood bank application that seeks out appropriate blood donors and encourages the public to actively engage in the blood donation campaign. Malaysia Blood Bank Android Mobile Application is developed which include the blood stock status, upcoming donation activity, donation appointment and emergency donation. A User Acceptance Test (UAT) is carried out to find any application flaws. In conclusion, the goal of this study which is find the compatible blood donors and call on the public to actively participate in the blood donation drive had achieved by successfully developed the application.

ABSTRAK

Nyawa seseorang boleh diselamatkan sebahagian besarnya dengan darah. Darah datang dalam empat jenis: A, B, O, dan AB. Terdapat sejumlah lapan jenis darah, dengan setiap kumpulan mempunyai pilihan untuk menjadi RhD positif atau RhD negatif. Satu-satunya pilihan kemudahan perubatan dan agensi kerajaan untuk menawarkan bekalan darah yang selamat kepada individu yang memerlukan adalah melalui derma darah. Mungkin sukar untuk mencari penderma yang sesuai dalam masa yang singkat. Berikutan pengenalan PKP di Malaysia, bilangan beg darah di Pusat Darah Negara dan bank darah lain di seluruh negara menurun sebanyak 40%, daripada 111,328 beg dalam Mei 2018 kepada 67,135 beg (Mac-Mei 2020). Sehubungan itu, tujuan projek ini adalah untuk membangunkan aplikasi bank darah yang mencari penderma darah yang sesuai dan menggalakkan orang ramai terlibat secara aktif dalam kempen derma darah. Aplikasi Mudah Alih Android Bank Darah Malaysia dibangunkan yang merangkumi status stok darah, aktiviti derma yang akan datang, pelantikan derma dan derma kecemasan. Ujian Penerimaan Pengguna (UAT) dijalankan untuk mencari sebarang kelemahan aplikasi. Kesimpulannya, matlamat kajian ini iaitu mencari penderma darah yang serasi dan menyeru orang ramai untuk mengambil bahagian secara aktif dalam usaha menderma darah telah tercapai dengan berjaya membangunkan aplikasi tersebut.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

UMP	UNIVERSITY MALAYSIA PAHANG
SRS	SOFTWARE REQUIREMENT SPECIFICATION
UAT	USER ACCEPTANCE TEST

CHAPTER 1

INTRODUCTION

1.1 Introduction

Blood plays an important role in saving a person's life. There are four blood types: A, B, O, and AB. Each group can be RhD positive or RhD negative, resulting in a total of eight blood types. Donating blood is the only way for hospitals and legal institutions to provide a safe source of blood to those in need. The donated blood needed to be stored under 2 to 6 °C in order to keep it fresh. (*Storage of Blood and Blood Products | Homerton University Hospital Foundation Trust*, n.d.) Thus, blood banks were built. Blood bank can be meant as a facility that collects blood from donors, types it, separates it into components, stores it, and prepares it for transfusion to patients. A blood bank can be a stand-alone unit or a component of a larger hospital laboratory. (*Medical Definition of Blood Bank*, n.d.) When a patient requires blood transfusion, the blood bank is contacted.

People usually discover blood donation events via newspapers, word of mouth, or mobile phone push notifications. This makes it easy for the masses to miss the news of blood donation activities. Based on the poll 2018, in Malaysia, O is the most common blood type, followed by A and B and the rarest being AB blood type. (*National Blood Centre Calls on Public to Donate Blood*, n.d.) Due to the rarity of the blood type, time is of the essence in any emergency. Time is life for emergency patients, and every minute counts. However, in a small community, it might be difficult to discover appropriate donors in a short amount of time. Additionally, blood banks may or may not have suitable blood type supplies. In today's era, the Internet is the fastest medium for transmitting information. Therefore, these technologies should bring hope to these emergency patients. Hence, a blood bank application is a must for society.

The goal of this study is to come up with a blood bank application which aims to find compatible blood donors and call on the public to actively participate in the blood donation drive.

1.2 Problem Statement

As the pandemic spreads, blood donation statistics have plummeted compared to pre-epidemic levels. This has led to blood shortages all over the world due to movement restrictions and social distancing rules. As proof, the implementation of the movement control order (MCO) in Malaysia, blood supply at the National Blood Centre and other blood banks around the nation dropped by 40%, from 111,328 bags (Mac-May 2018) to 67,135 bags (Mac-May 2020). (Mei Ling et al., 2018)Despite the fact that Blood Donation and Blood Bank Applications have been around for a long time, barely only a small portion of the population downloads and uses it. Since the cases of pandemic are increasing day by day, therefore the frequency of donating blood has been reduced by the blood donor. As the years passed, the necessity of blood by the human is increasing due to the increasing ailments that are infected by the human. Especially for those patients who have unusual blood types, such as RH null blood type, in particular, require a greater amount of blood. Therefore, Blood Bank Application are needed in order to lend a hand to seek a suitable blood type for them. On top of that, although Malaysian blood banks frequently lack distinct blood types, this phenomenon is rarely spread among the population. People are unaware of the locations of hospitals or blood banks that are in need of blood, so in result blood shortages have happened. In order to solve this problem, by utilizing this programme people may get real-time location information on blood type shortages. Aside from that, according to a survey conducted between 2008 and 2014, repeat blood donors are twice as likely as new donors. Hence, the fresh donors are desperately needed, as repeat donors are increasingly unable to donate blood due to a variety of factors. In today's society, nearly every young person owns a smartphone, so a mobile app would be an excellent venue for promoting this activity. Although these issues may appear

minor at first, if they continue to collect, they will eventually become a major issue to the nation. However, all of these issues are readily handled with this blood bank app.

1.3 Objective

1. To determine the existing activities on blood bank records.
2. To develop mobile blood bank application records in order to heighten accessibility of different blood types toward the supply in blood banks and hospitals.
3. To validate the mobile blood bank application towards the increasing number of new blood donors.

1.4 Scope

- User Scope

  i. End User (Anyone who intent to donate)

  ii. Hospital

  iii. Blood Bank

  iv. Donation Organiser

- System Scope

  i. Covers blood bank access, donor discovery, emergency help and donation activities.

- Development Scope

  i. Contains multimedia elements such as graphics and text.

  ii. Using Android Studio as the framework and Firebase as the cloud storage

1.5 Significance of the project

i. Blood Donors

Donors able to identify the location of blood shortages. They have an option of donating blood at a blood drive in their area. They can also view other user's information.

ii. Hospital and Blood Bank

The hospital and blood bank are able to keep society informed about the current blood volume situation. They can post a request for finding a specific blood type donor if there is an emergency case. They are also able to spread blood donation activities to the society rapidly.

iii. Organizer of blood donation

Organizer able to spread blood donation activities to the society rapidly. They can confirm the number of participants' approximate range based on the user's reply.

Chapter 2

Literature Review

2.1 Introduction

Chapter 2 covers the review of existing mobile applications in the healthcare field. Three existing augmented reality application in science education mainly in science were explained in details by focusing in term of its Graphical User Interface (GUI), Mobile Operating System (OS), language provided, connection type, target audience of application, size of application, main function of the application, the advantages and disadvantages of the healthcare application. The comparisons are done to assure that the healthcare application developed in this project is superior to the previous version.

2.2 Review of Existing Systems

Three existing healthcare applications will be reviewed in this section. The three healthcare applications are Bloodkad, Save Life Connect and Friends2Support.org .

2.2.1 Application 1- Bloodkad

Bloodkad is an app that streamlines and simplifies the process of donating blood for Brunei citizens. This application is available in Google Play Store and App Store. The installation of the application is free for Android and iOS users. Figure 2.1 shows the logo of Bloodkad. This application was developed by Nextacloud Technologies Sdn Bhd.

To use this application, users can tap on “Donate Now”, when it is successfully booked an appointment, they will receive a confirmation message. Next, users can also find donors by tapping “Find Donors”. User are required to fill up the request information such as reason, required blood type and location. On the top, any

upcoming donation events are posted. While at the bottom, Bloodkad allow users to view the hospital blood stock. Besides, users can manage blood donation appointments by tapping the “Book Appointment”. Figure 2.2 shows the main page interface of Bloodkad.

Figure 2.1 Logo of Bloodkad

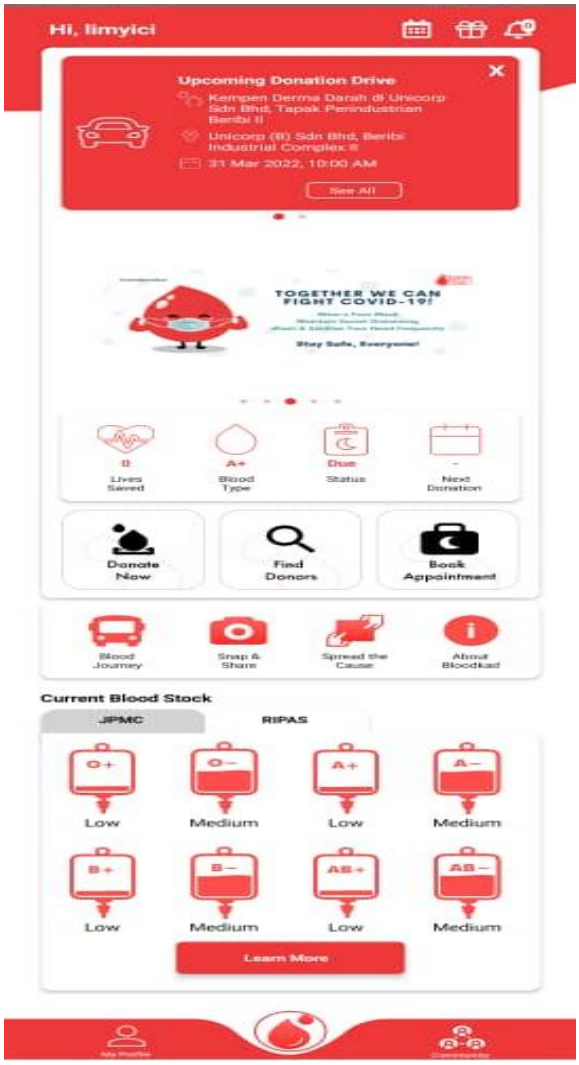


Figure 2.2 Example of Bloodkad Main Page

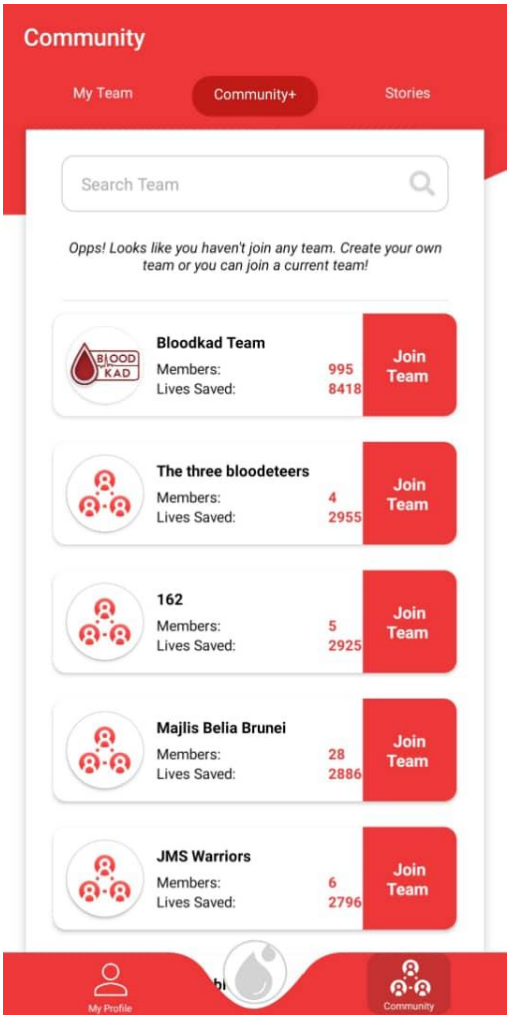


Figure 2.3 Example of Bloodkad Community Page

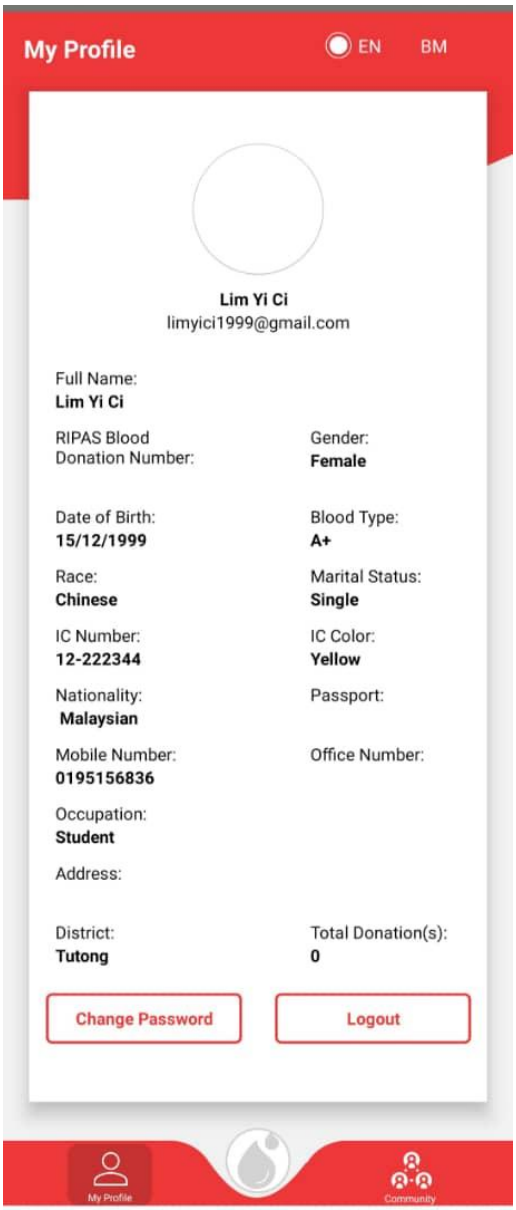


Figure 2.4 Example of Bloodkad Profile Page

2.2.2 Application 2- Save Life Connect

This application solves the problem of blood emergencies by connecting blood donors directly with people in blood need. Currently, Save Life Connect can be found on Google Play Store and App Store. Save Life Connect is a free download to Android and iOS mobile system users. Besides from mobile, it also can be applied by web based. This application was developed by Save Life Foundation. Figure 2.3 shows the logo of Save Life Connect.

Users can view the donation activity by clicking the second tab above which is "Blood Requests". On the contrary, they can make a post for requesting blood. While on the fourth tab "My History", users are able to view all the sent and received blood requests. In the fifth tab, users are allowed to view registered user's blood type and location in the map. They also have the option to filter any blood group on top of the map. Figure 2.4 shows the interface of "Members Nearby" in Save Life Connect.

Figure 2.5 Logo of Save Life Connect

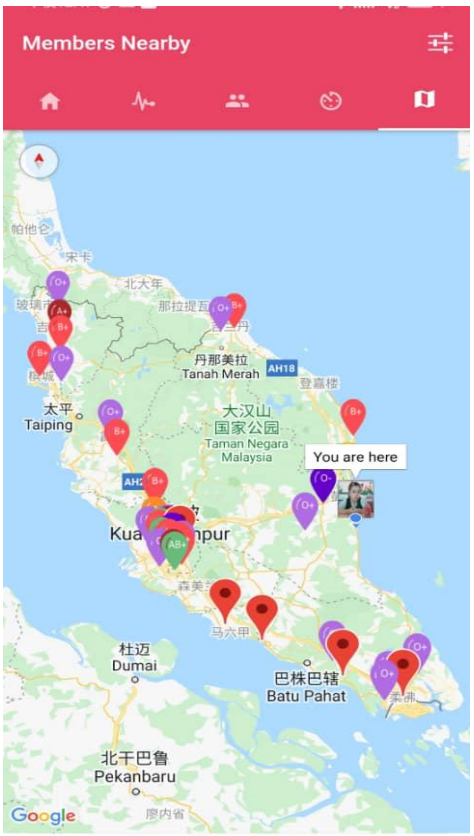


Figure 2.6 Example of nearby blood donors discovery in Save Life Connect

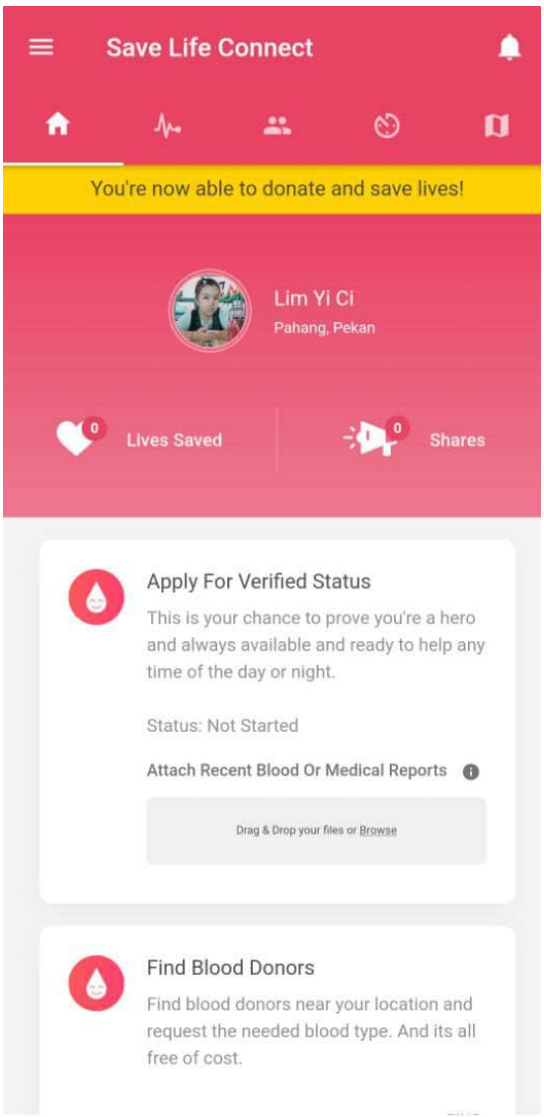


Figure 2.7 Example of Profile Page in Save Life Connect

2.2.3 Application 3- Friends2Support.org

Friends2Support.org is the world's largest voluntary blood donor's organisation with services available in India, Sri Lanka, Bangladesh, Nepal, Malaysia, Oman and Yemen. The main aim is to have a society wherein there is no shortage of blood in the most critical situations. Friends2Support is available in Microsoft Store, Google Play

Store and App Store. Windows, Android and iOS users are free to download. It also provides services by web based. Figure 2.5 is the logo of this application.

This application allows users to register as a donor. Users are needed to fill up their personal details. At the main page, it is a blood donors discovery interface. Users need to choose the donor's requirement from the given selection. To find other functions in the application, users need to click the hamburger menu button on the upper right corner. Figure 2.6 shows the discovery of blood donors in the application.

Figure 2.8 Logo of Friends2Support.org

Figure 2.9 Example of discovery of blood donors in Friends2Support.org

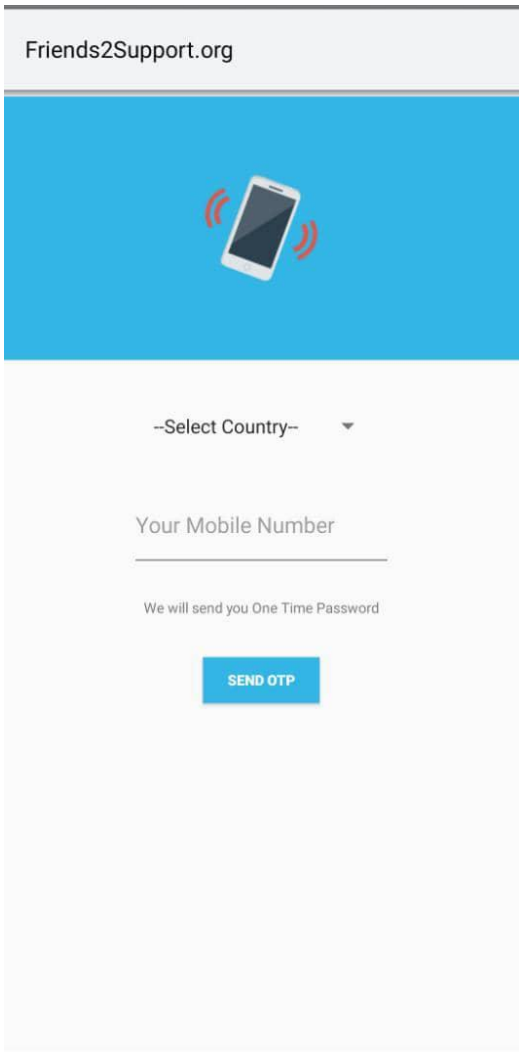


Figure 2.10 Example of Register as Blood Donor in Friends2Support.org

2.3 Comparisons of Three Existing System

Table 2.1 summarises the comparison of three current systems based on the review conducted in Section 2.2. There are 9 elements that are used for comparison which are Graphical User Interface (GUI), Mobile Operating System (OS), language provided, connection type, target audience of application, size of application, main function of the application, the advantages and disadvantages of the healthcare application.

Table 2.1 Comparison of Three Existing Systems

Application Name	**Bloodkad**	**Save Life Connect**	**Friends2Support.org**	**Proposed apps**
Graphical User Interface (GUI)	The interface is clean and simple and well organised.	The interface is beautiful and well organised.	The interface is dull but it is simple to use.	The interface is clean and simple and well organised.
Mobile Operating System (OS)	Android, iOS	Android, iOS	Windows, Android, iOS	Android
Connection Type	Online	Online	Online	Online
Language Provided	English, Malay	English, French, Jawi, Arabic	English	English
Size of Application	32 MB	Varies with device	4.1 MB	Expected to be 30 - 60MB
Year of Development	2018	2018	2012	2022

Target Audience	Brunei Citizens	Everyone	India, Sri Lanka, Bangladesh, Nepal, Malaysia, Oman and Yemen citizens	Malaysia Citizens
Main Function	1. Provide a button for booking blood donation appointment for specific hospital. 2. Provide a button for donor discovery. 3. Provide a button to view donating history. 4. Provide blood stock status of hospitals. 5. Provide a button for users to join any community.	1. Provide a tab for blood request function. 2. Provide a tab for creating a donation team. 3. Provide a tab for blood request history. 4. Provide a tab finding nearby donors.	1. Provide radio buttons for choosing requirements for blood donors. 2. Provide a tab to register as a blood donor.	1. Provide blood stock status of hospitals. 2. Provide finding donors requirements. 3. Provide a button for upcoming donation activities. 4. Provide a button for booking blood donation appointment for specific hospital.

Advantages	1. Provide information explaining every blood stock status. 2. Provide points for rewards and redeem. 3. Provide a function of donation sharing to social media.	1.The map in "Member Nearby" is able to zoom in and out. 2. Provide users' blood donation tracking from donation through delivery. 3. Provide a function of donation sharing to social media.	1. More requirements for blood donors. 2. Provide contact number of matched blood type users.	1. Provide a button for registering a new hospital or organisation. 2. Legal organisations and hospitals authorise it as a representative of blood donation apps. 3. User privacy has low risk of exposure.
Disadvantages	1. There is no button to register a new hospital or organisation.	1. Connection failed errors usually occur. 2. Without legal organisation and hospital authorise it as a representative of blood donation apps.	1. User privacy has a risk of exposure. 2. Poor user interface. 3. Less function other than getting donors' contact and registering as a donor.	1. Only for Android users. 2. Only the English language is available. (these disadvantages will be covered in future work)

2.4 Summary

According to the comparison and analysis performed on the three existing applications, each has its own distinct feature, and some may have a lot of room for improvement. The three current applications' primary purpose is to locate blood donors. According to the research, the suggested application should have certain unique features that set it apart from the competition, such as the ability to identify a local hospital using Google Maps API. In addition, the suggested application will offer some blood type compatibility advice.

Chapter 3

Methodology

3.1 Introduction

In this chapter, the methodology in developing this Blood Bank Application will be discussed as well as the development methodology, interface,frameworks, development tools and instrument that will be used in this application will all be detailed in depth in this chapter. The application's flow will be demonstrated, and the application's modules will be described.

3.2 Methodology

In this project, agile methodology is used to develop the application. Agile techniques attempt to produce the proper product through small cross-functional self-organising teams that supply small pieces of functionality on a regular basis, allowing for frequent user input and course correction as needed. (*What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models*, n.d.) This technique is incremental and iterative, allowing for more flexible implementation of changes. Since the software is separated into sprints, it is very easy to make modifications thanks to the Agile process in mobile app development. Therefore, the development process will not be harmed, and adjustments may be made swiftly.

Agile development methodology include 5 processes which are Define Requirements, Design, Develop, Test and Deploy. To guarantee client satisfaction, the cycle will be repeated as many times as necessary. It will give a quick, continuous, compact, and helpful program for the client to try out. Tiny updates can be implemented in the following cycle by providing a small batch of software to the user. Figure 3.2.1 shows the phases flow of agile development methodology.

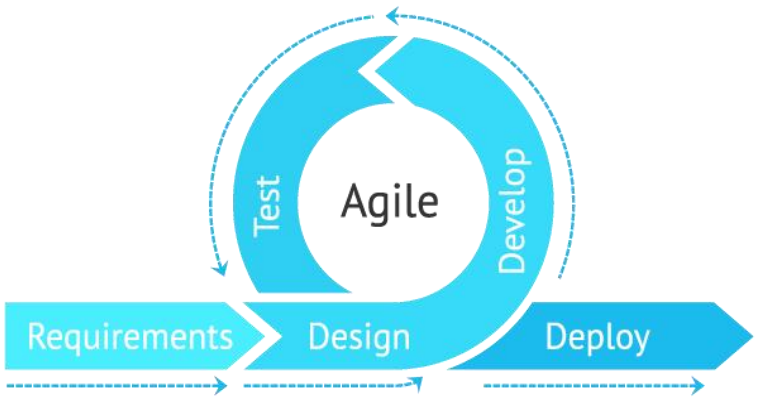


Figure 3.1 Agile Methodology

Advantages	Disadvantages
Lower risk factors as early deliverables are made accessible to the end-users.	Progress is more difficult to assess than it is in Waterfall since it occurs across several cycles.
Adapting to change and responding more quickly	Time, and resources are difficult to anticipate at the initiation of a project.
Process and tools are undervalued in favour of people and encounters. All stakeholders communicate with one another on a regular basis. (*Agile Development – Advantages, Disadvantages and When to Use It? - The Official 360logica Blog*, n.d.)	Lack of official documentation and designing leads to a very heavily reliant on individuals for training and other tasks.
Faster detection and correction of faults and flaws.	There is no clear concept of what the "final result" should look like, hence initiatives have no end.
Time and cost savings	The precise cost will be uncertain.

Table 3.1 Advantages and Disadvantages of Agile Development Methodology

3.2.1 Requirements Phase

The first stage in developing a system is gathering project requirements. In this phase, all the requirements will be listed. For instance, the problems handled by this project, as well as how this project will improve user experience and system performance, are discussed here. The requirements can be divided into 2 categories, functional requirements and non-functional requirements.

Functional requirements describe what a software system must accomplish and how it must work; they may also be described as product features that cater to the demands of users. Non-functional requirement is a specification that outlines the system's operating capabilities and limitations that help it perform better. For example, consider the following: speed, security, and dependability. The requirements will be finalised, approved, and analysed individually.

3.2.2 Design Phase

The high-level design of the system begins to deliver each chosen requirement in the design phase. All associated diagrams and designs, such as the context diagram, data flow diagram, use case diagram, and sequence diagram interface design, will be documented in the Software Requirement Specification (SRS) document. Programming language and system architecture are established in this phase.

3.2.3 Develop Phase

After the design phase, the next phase will be the development phase. The Develop Phase's goal is to turn the system design prototyped during the Design Phase into a functional information system that meets all of the system's defined requirements. The main activity involved in this phase is coding. Software developers build code for the components based on the design document. The most time-consuming part of the software development life cycle is the development phase, yet it is more predictable than the design phase. Finally, this phase produces a functional software product as well as a Source Code Document. (*Software Development Life Cycle: Phases and Models*, n.d.)

3.2.4 Test Phase

The test phase starts after the develop phase. The fundamental purpose of testing is to find all of the flaws and mistakes in the programme before the deploy phase. Aside from finding mistakes, testing is also used to measure the system quality. Testers then go through the Software Testing Life Cycle tasks to ensure that the system is free of errors, flaws, and defects and that its functions perform as planned. The predicted and actual results are compared to ensure that the system is functioning properly. There are 4 types of testing, begin with unit testing, integration testing, system testing and acceptance testing. When the testing results show that there are no mistakes, the project will go on to the deploy phase.

3.2.5 Deploy Phase

Deploy phase is the final phase which goes after the test phase. In this phase, the system is then deployed to a real-world setting where end users may start using it. In the production environment, all data and components are available. (*Software Development Life Cycle: The Phases of SDLC - TestLodge Blog*, n.d.) User feedback is solicited and will be considered in the next iterations.

3.3 Project Requirement

3.3.1 Functional Requirements

System Requirement	
1.	System should allow all users to register by using their email.
2.	System should allow all users to reset the password after clicking the check email button.
3.	System should allow all users to login by using the registered email and password.

4.	System should allow all users to update their profile.
5.	System should allow donors to view others' profiles.
6.	System should allow all users to view their own profile.
7.	System should allow donors to view the information of donation activity.
8.	System should allow donors to view the location of donation activity via Google Maps.
9.	System should allow hospitals and blood banks to create donation activity .
10.	System should allow hospitals and blood banks to update donation activity information.
11.	System should allow hospitals and blood banks to delete donation activity information.
12.	System should allow hospitals and blood banks to view donation activity information.
13.	System should allow donation organisers to create donation activity .
14.	System should allow donation organisers to update donation activity information.
15.	System should allow donation organisers to delete donation activity information.
16.	System should allow donation organisers to view donation activity information.
17.	System should allow hospitals and blood banks to update blood stock status.
18.	System should allow donors to view blood stock status.

19.	System should allow donors to book blood donation appointments.
20.	System should allow donors to delete blood donation appointments.
21.	System should allow donors to view blood donation appointments.
22.	System should allow donors to update blood donation appointments.
23.	System should allow hospitals and blood banks to view blood donation appointments.
24.	System should allow donors to view emergency donation help.
25.	System should allow hospitals and blood banks to create emergency donation help.
26.	System should allow hospitals and blood banks to update emergency donation help.
27.	System should allow hospitals and blood banks to delete emergency donation help.
28.	System should allow hospitals and blood banks to view emergency donation help.
29.	System should allow all users to view blood type compatibility advice.

Table 3.2 Functional Requirement - System Requirement

User Requirement	
1.	Hospitals, blood banks and donation organisers should apply authentication when registering an account.

Table 3.3 Functional Requirement - User Requirement

3.3.2 Non-Functional Requirements

Performance	The system response time should not exceed 3 seconds.
Usability	The application interface should be user-friendly and easy to use.(*Nonfunctional Requirement Examples - Requirements Quest*, n.d.)
Maintainability	The mean preventive maintenance time update for the blood bank application should be less than 3 hours every 2 weeks.
Scalability	The Blood Bank Application shall be scalable to allow for unlimited expansion in the number of donors and legal entities.
Reliability	The Blood Bank Application has a 0.0001 chance of failing to get access to the system and make modifications. (*What Are Non-Functional Requirements? Types and Examples - WINaTALENT | Blog*, n.d.)
Accessibility	The blood type compatibility advice in Blood Bank Application are displayed in a table form instead of text.

Table 3.4 Non-Functional Requirement

3.4 Project Constraints

The following are the project's constraints:

i. Without push notification for donation activities

When legal organisations create donations activities, there is no push notification to all users. This will cause the activities does not spread efficiently because it requires users to manually open the application to view the activities.

3.5 Project Limitation

The Blood Bank App's drawbacks include the fact that it can only be installed on Android devices. On top of that, Blood Bank Application are only applicable to mobile but not web service.

The other limitation of this application is that it only provides an English version. Since English is the official language to most countries, this application is developed with English. Only English Learners are able to use this application.

3.6 Proposed Design

3.6.1 Context Diagram

Figure 3.2 shows the context diagram of Blood Bank Application. The software system's boundaries are defined and clarified by the context diagram. It determines the information flows between the system and the external entities.

There are 6 external entities for Blood Bank Application which are Firebase, blood donor, hospital, blood bank, donation organiser and Google Map API. Firebase acts as a database to store and retrieve data to the system. Google Map API provides the location result to the system according to the address. All users are required to register an account for the first time then login to the system.

Blood donors are able to view the blood stock information, emergency donation information and the donation activity information. They also able to create and delete the appointment information and view and update their profile information.

For hospital and blood bank users, create, update, delete and view can be done for donation activity information, emergency donation information. They are able to update and view their blood stock status and view the appointment information. Besides, they can also update and view their profile information.

For donation organisers, they are only able to view the blood stock status and view and update the profile information. They can delete, update, view and create donation activity information.

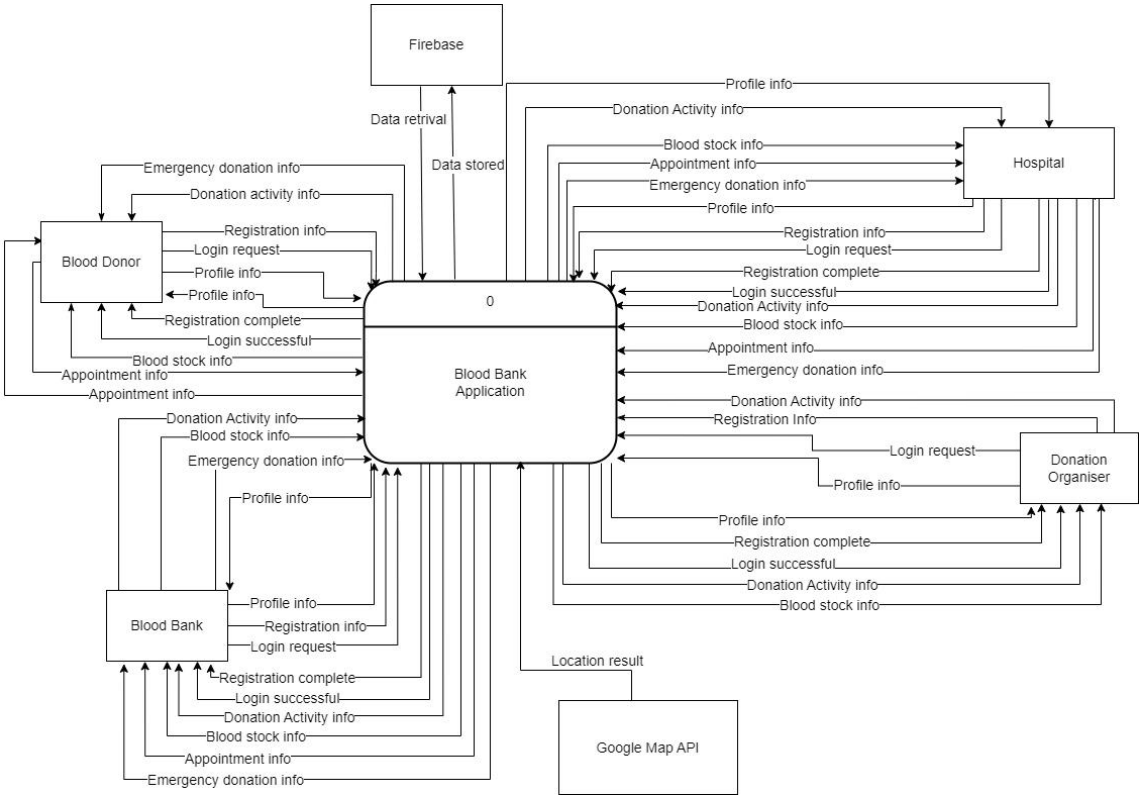


Figure 3.2 Context Diagram

3.6.2 Use Case Diagram

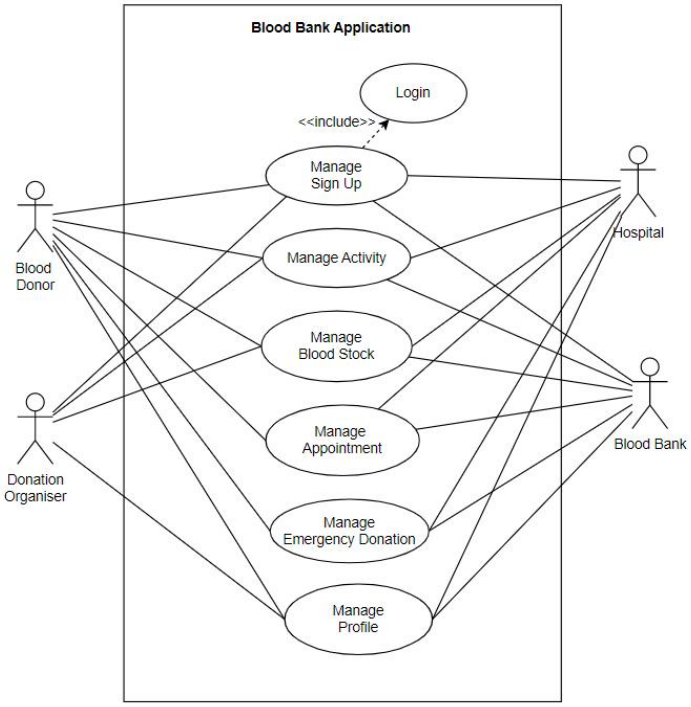


Figure 3.3 Use Case Diagram of Blood Bank Application

Module	**Requirements**	**User**
Manage Sign Up	The system allows blood donors, hospitals, donation organisers and blood banks to login, register an account and reset password by inserting the user's email address.	Blood donor, Hospital, Donation organiser, Blood bank
Manage Activity	The system allows hospital, donation organiser and blood bank to create, update, view and delete a donation activity. Blood donors are only allowed to view the upcoming donation activities.	Blood donor, Hospital, Donation organiser, Blood bank
Manage Blood Stock	The system allows hospital and blood bank to update and view their own blood stock status. Blood donors and donation organisers are only allowed to view the	Blood donor, Hospital, Donation organiser, Blood bank

	blood stock status of each hospital and blood bank by searching the name.	
Manage Appointment	The system allows blood donors to view, book, update and delete the donation appointment. Hospitals and blood banks are only allowed to view the appointment that was made by blood donor.	Blood donor, Hospital, Blood bank
Manage Emergency Donation	The system allows hospitals and blood banks to create, update, delete and view the emergency donation. Blood donors are only allowed to view the emergency donation.	Blood donor, Hospital, Blood bank
Manage Profile	The system allows blood donors, hospital, donation organiser and blood bank to view and update their own profile.	Blood donor, Hospital, Donation organiser, Blood bank

Table 3.5 Use Case Module Description of Blood Bank Application

3.6.3 Activity Diagram

3.6.3.1 Manage Sign Up

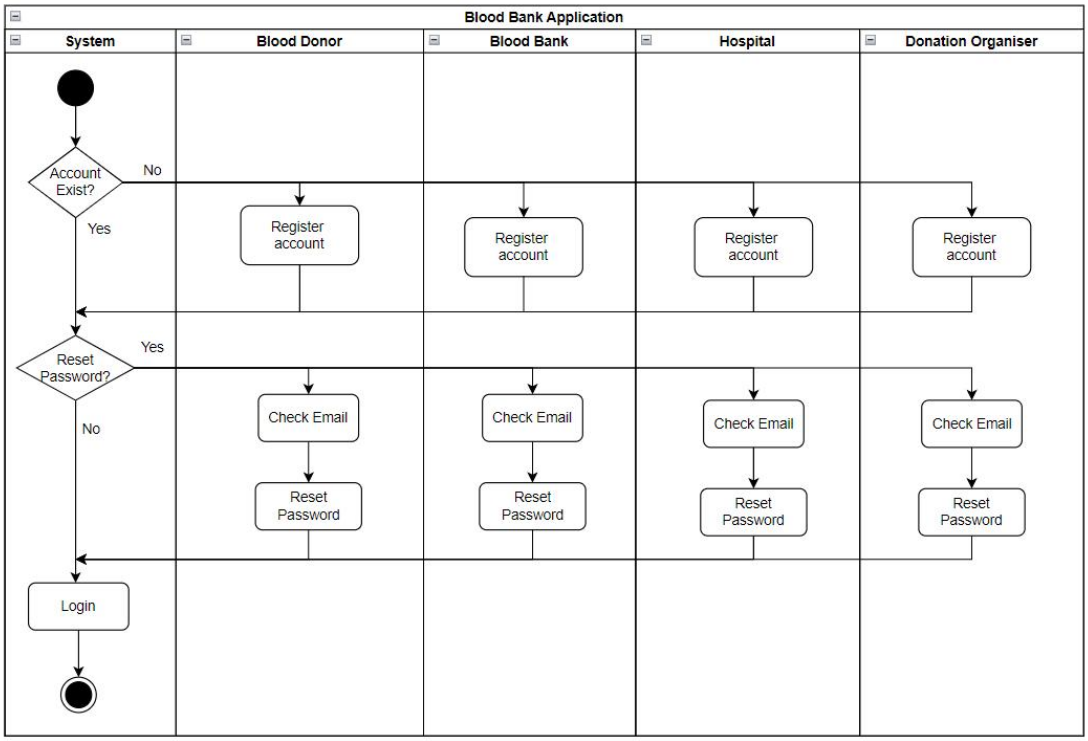


Figure 3.4 Manage Sign Up Activity Diagram

3.6.3.2 Manage Activity

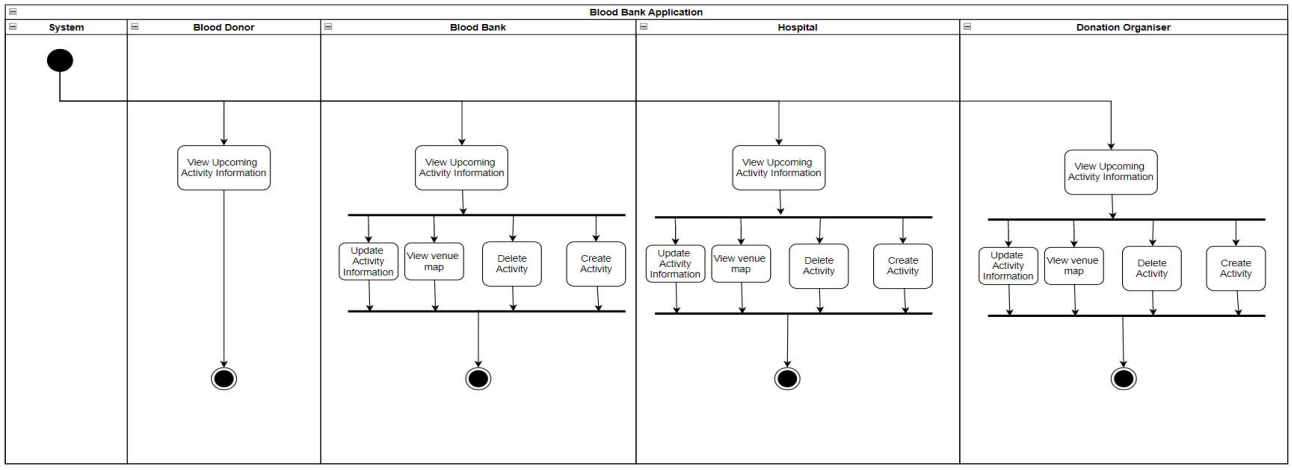


Figure 3.5 Manage Activity Activity Diagram

3.6.3.3 Manage Blood Stock

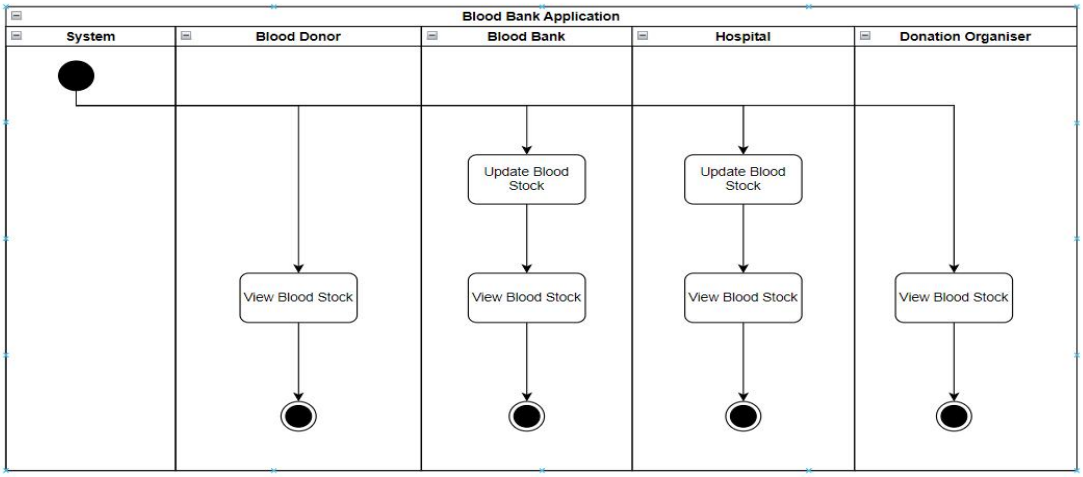


Figure 3.6 Manage Blood Stock Activity Diagram

3.6.3.4 Manage Appointment

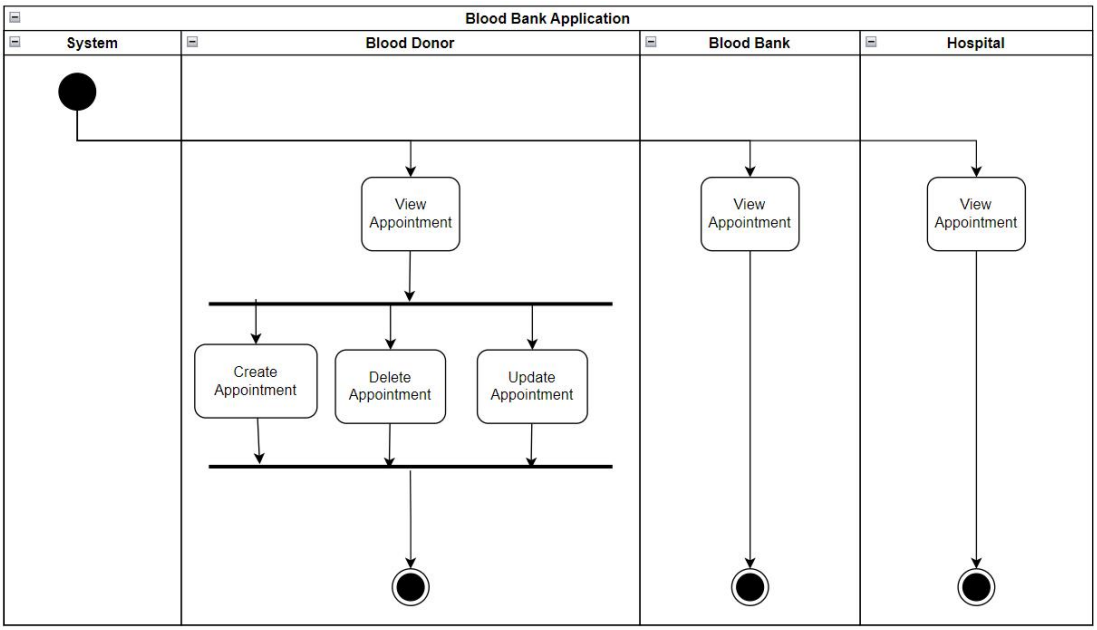


Figure 3.7 Manage Appointment Activity Diagram

3.6.3.5 Manage Emergency Donation

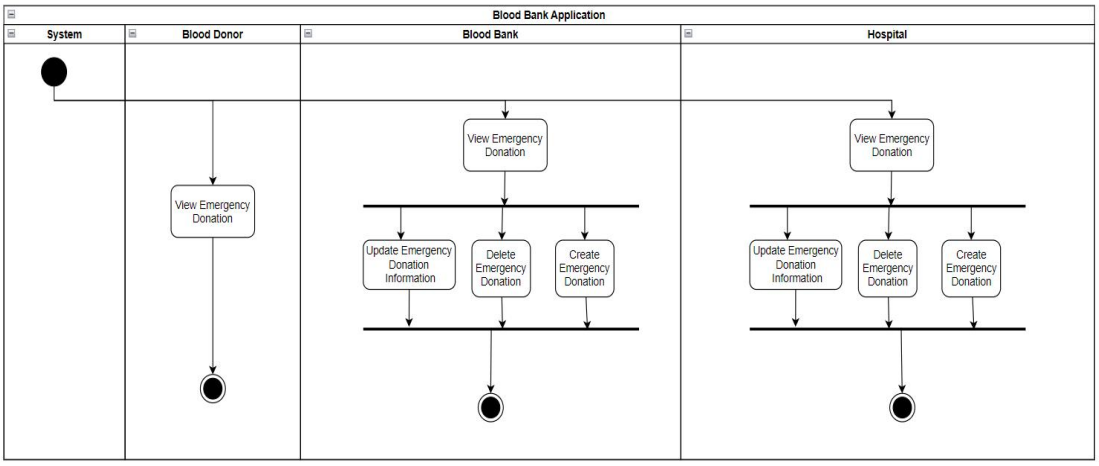


Figure 3.8 Manage Emergency Donation Activity Diagram

3.6.3.6 Manage Profile

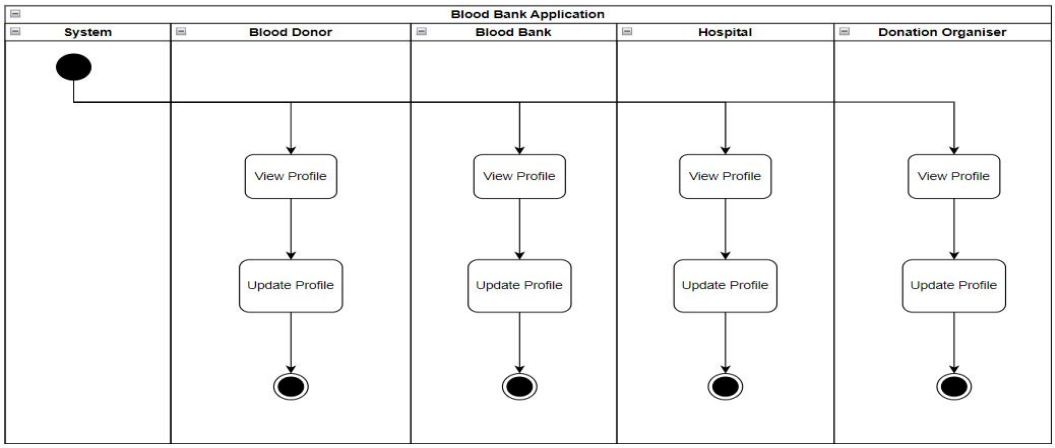
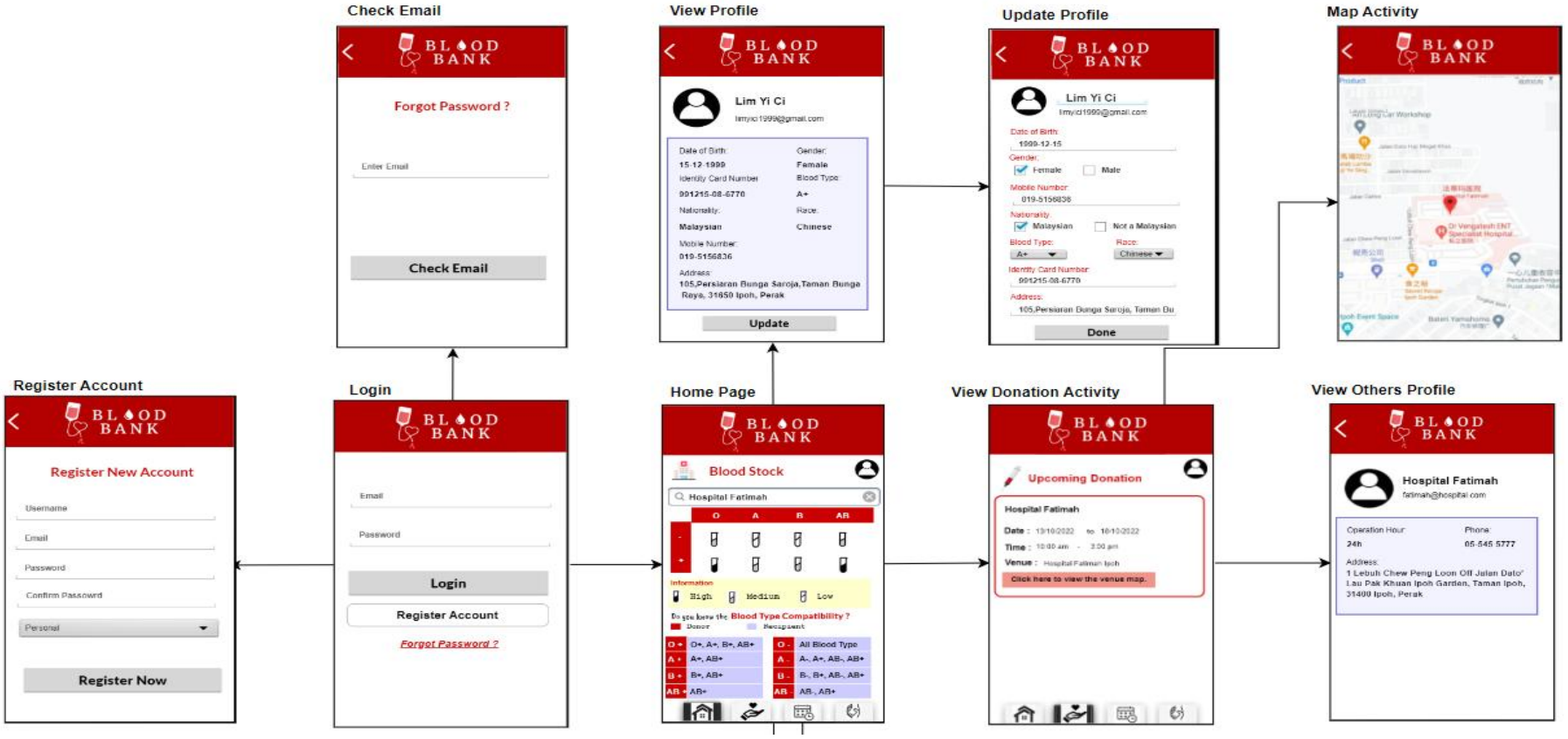


Figure 3.9 Manage Profile Activity Diagram

3.6.4 Storyboard

3.6.4.1 Blood Donor Storyboard



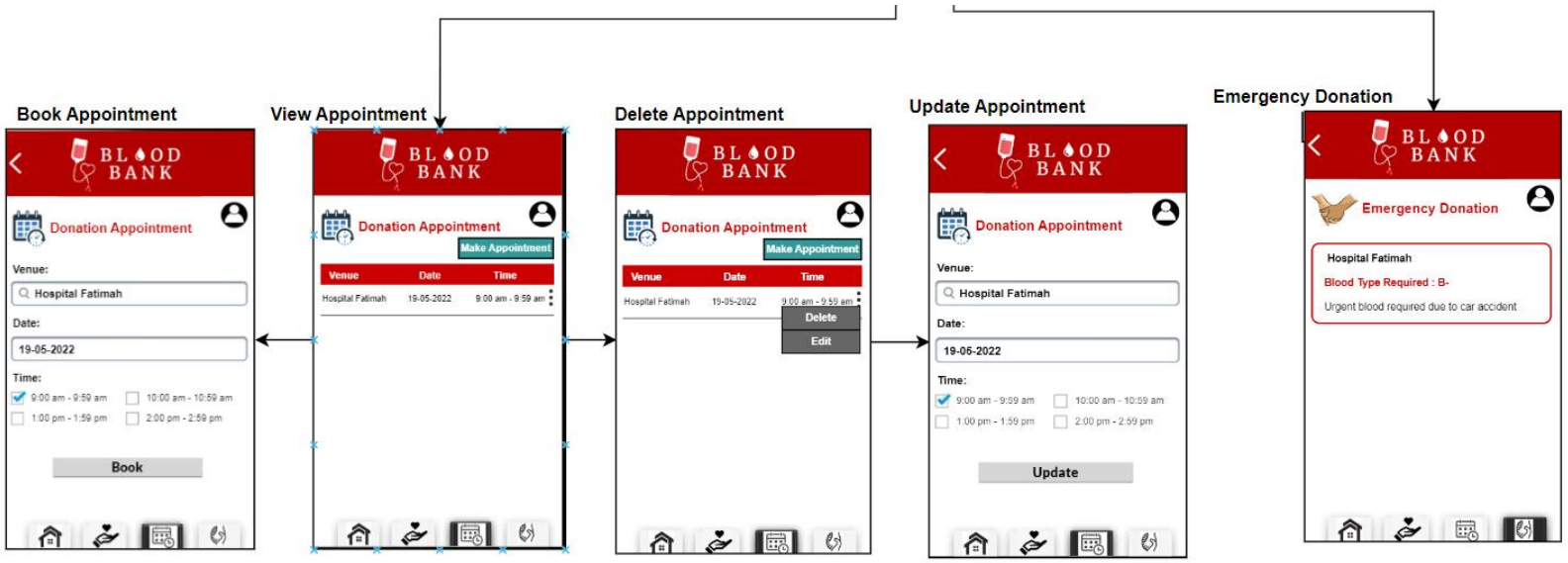
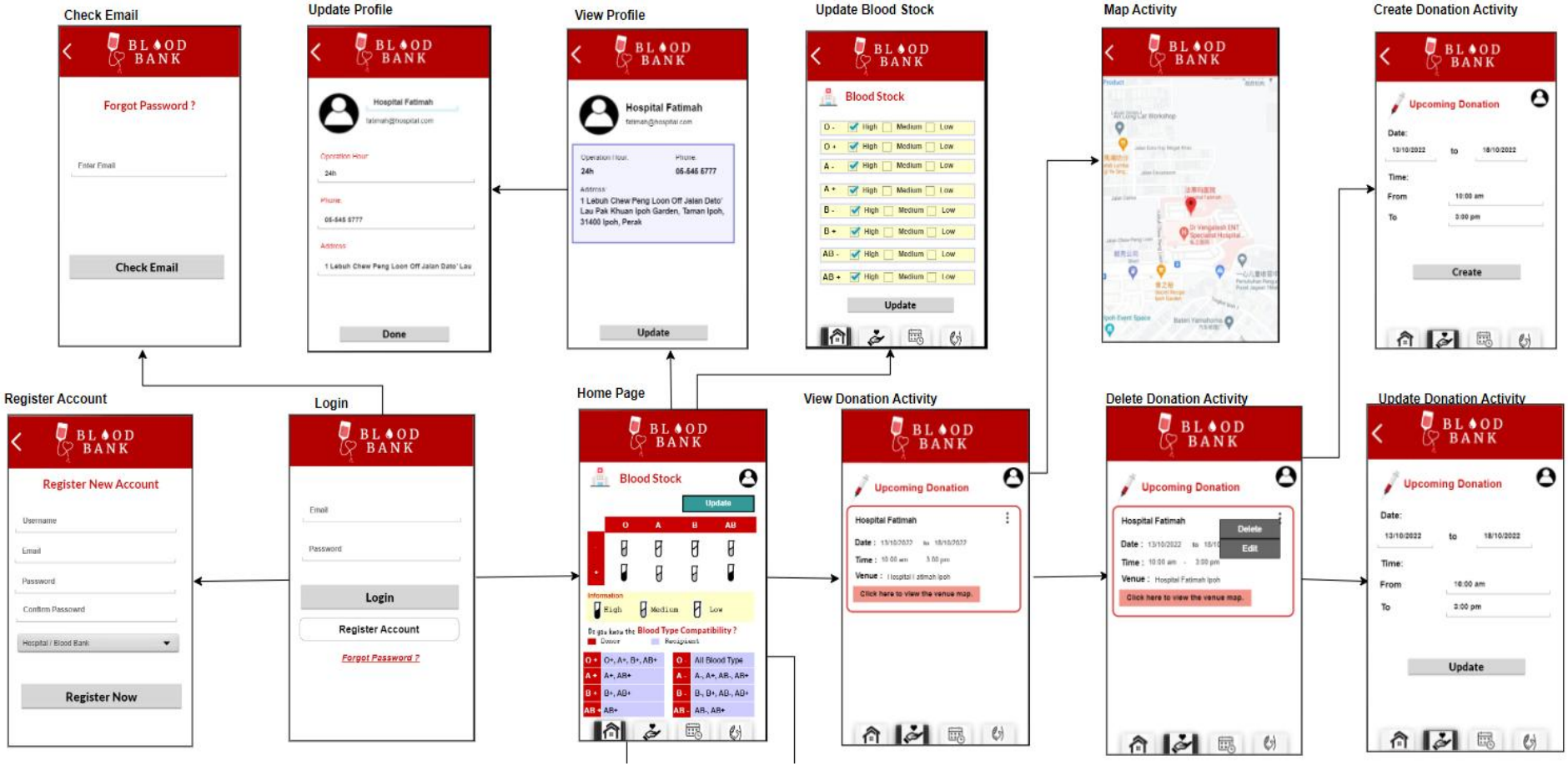


Figure 3.10 Blood Donor Storyboard of Blood Bank Application

3.6.4.2 Hospital and Blood Bank Storyboard



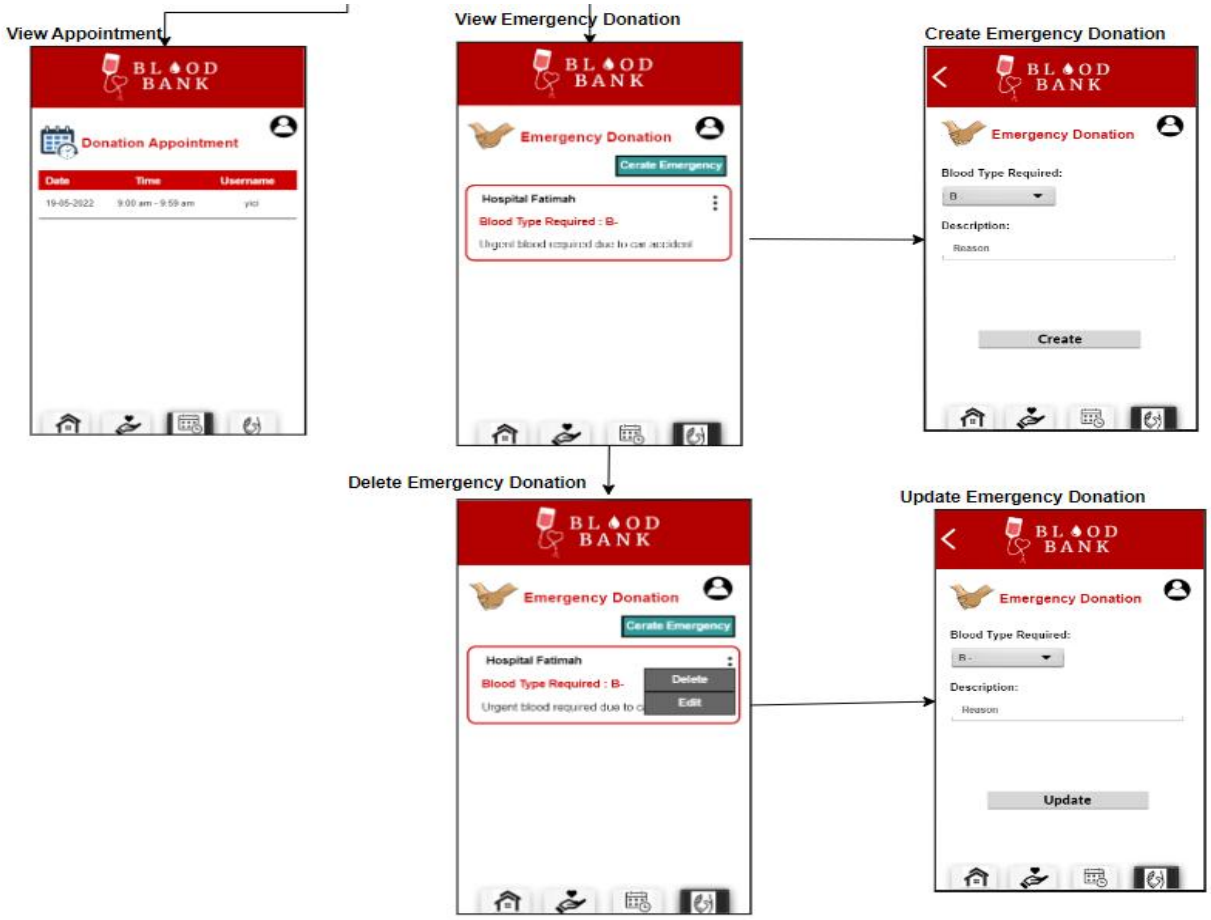


Figure 3.11 Hospital and Blood Bank Storyboard of Blood Bank Application

3.6.4.3 Donation Organiser Storyboard

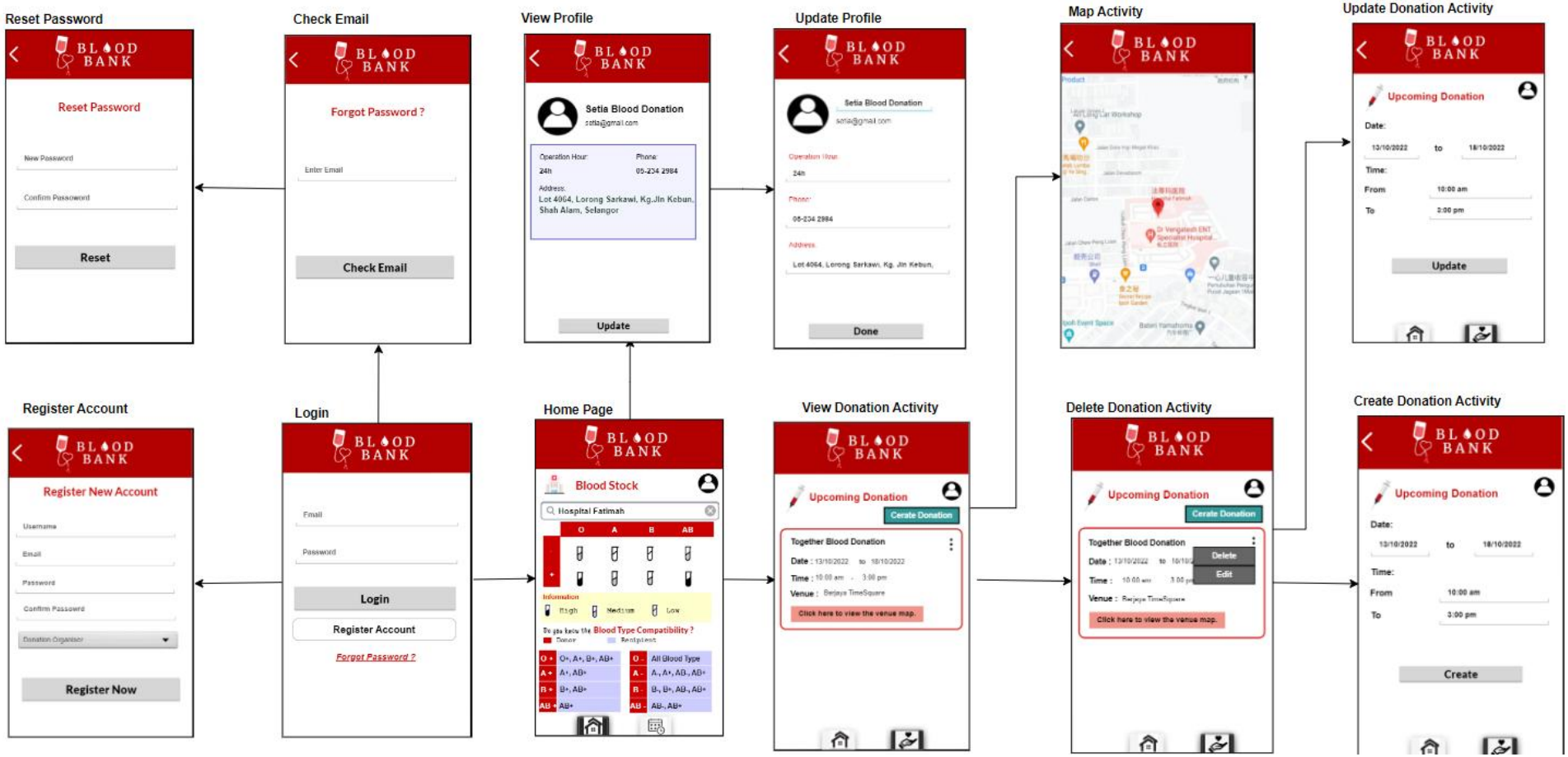


Figure 3.12 Donation Organiser Storyboard of Blood Bank Application

3.7 Data Design

3.7.1 Entity Relationship Diagram (ERD)

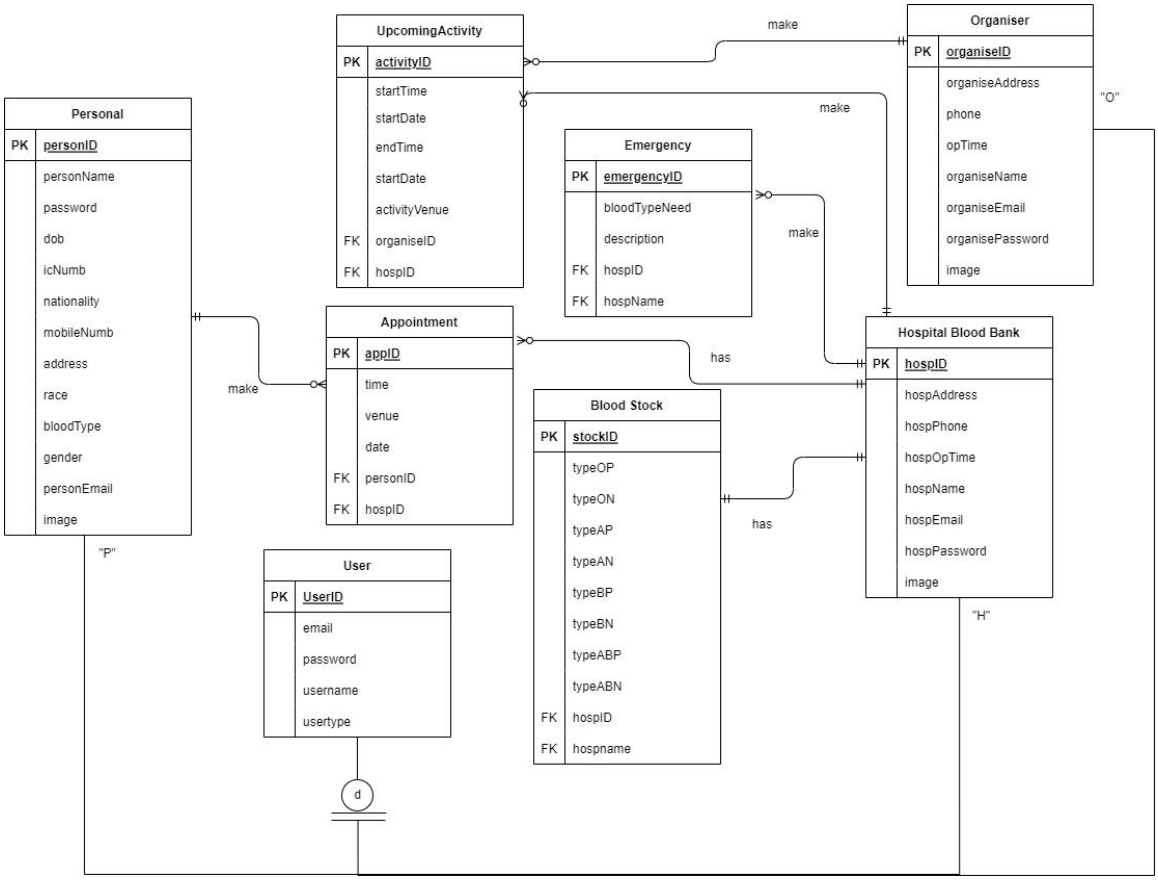


Figure 3.13 ERD Diagram

3.7.2 Data Dictionary

User

Field Name	Description	Data Type	Constraint
UserID	User ID	VARCHAR(100)	PK
email	Email Address	VARCHAR(200)	-
password	Password	VARCHAR(100)	-
username	Username	VARCHAR(50)	-
usertype	User type	VARCHAR(100)	-

Table 3.6 User Table

Personal

Field Name	Description	Data Type	Constraint
personID	Person ID	VARCHAR(100)	PK
personName	Person Name	VARCHAR(100)	-
password	Password	VARCHAR(100)	-
dob	Date of Birth	DATE	-
icNumb	IC Number	INT	-
nationality	Nationality	VARCHAR(100)	-
mobileNumb	Mobile Number	INT	-
address	Address	VARCHAR(200)	-
race	Race	VARCHAR(50)	-
bloodType	Blood Type	VARCHAR(50)	-
gender	Gender	VARCHAR(50)	-
personEmail	Person Email	VARCHAR(100)	-
image	Hospital Blood Bank Profile Image	VARCHAR(200)	-

Table 3.7 Personal Table

Organiser

Field Name	Description	Data Type	Constraint
organiseID	Organisation ID	VARCHAR(100)	PK
organiseAddress	Organisation Address	VARCHAR(200)	-
phone	Phone Number	INT	-
opTime	Operation Time	VARCHAR(50)	-
organiseName	Organisation Name	VARCHAR(100)	-
organiseEmail	Organisation Email	VARCHAR(100)	-
organisePassword	Organisation Password	VARCHAR(100)	-

image	Organisation Profile Image	VARCHAR(200)	-

Table 3.8 Organiser Table

Hospital Blood Bank

Field Name	Description	Data Type	Constraint
hospID	Hospital Blood Bank ID	VARCHAR(100)	PK
hospAddress	Hospital Blood Bank Address	VARCHAR(200)	-
hospPhone	Hospital Blood Bank Phone Number	INT	-
hospOpTime	Hospital Blood Bank Operation Time	VARCHAR(50)	-
hospName	Hospital Blood Bank Name	VARCHAR(100)	-
hospEmail	Hospital Blood Bank Email	VARCHAR(100)	-
hospPassword	Hospital Blood Bank Password	VARCHAR(100)	-
image	Hospital Blood Bank Profile Image	VARCHAR(200)	-

Table 3.9 Hospital Blood Bank Table

UpcomingActivity

Field Name	Description	Data Type	Constraint
activityID	Activity ID	VARCHAR(100)	PK
startTime	Starting Time	VARCHAR(50)	-
endTime	End Time	VARCHAR(50)	-
startDate	Starting Date	VARCHAR(50)	-
endDate	End Date	VARCHAR(50)	-
activityVenue	Activity Venue	VARCHAR(100)	-
organiseID	Organisation ID	VARCHAR(100)	FK
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK

Table 3.10 UpcomingActivity Table

Emergency

Field Name	Description	Data Type	Constraint
emergencyID	Emergency ID	VARCHAR(100)	PK
bloodTypeNeed	Blood Type Need	VARCHAR(10)	-
description	Description	VARCHAR(200)	-
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK
hospName	Hospital Blood Bank Name	VARCHAR(100)	FK

Table 3.11 Emergency Table

Appointment

Field Name	Description	Data Type	Constraint
appID	Appointment ID	VARCHAR(100)	PK
date	Date	DATE	-
time	Time	VARCHAR(50)	-
venue	Venue	VARCHAR(100)	-
personID	Person ID	VARCHAR(100)	FK
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK

Table 3.12 Appointment Table

Blood Stock

Field Name	Description	Data Type	Constraint
stockID	Blood Stock ID	VARCHAR(100)	PK
typeOP	Type O Positive	VARCHAR(50)	-
typeON	Type O Negative	VARCHAR(50)	-
typeAP	Type A Positive	VARCHAR(50)	-
typeAN	Type A Negative	VARCHAR(50)	-
typeBP	Type B Positive	VARCHAR(50)	-
typeBN	Type B Negative	VARCHAR(50)	-

typeABP	Type AB Positive	VARCHAR(50)	-
typeABN	Type AB Negative	VARCHAR(50)	-
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK
hospname	Hospital Blood Bank Name	VARCHAR(100)	FK

Table 3.13 Blood Stock Table

3.8 Proof of Initial Concept

Figure 3.14 shows the register account interface of Blood Bank Application. The users are required to register an account for the first time by inserting the email, password and type of user. Then click the "Register" button for registering. An account will be created.

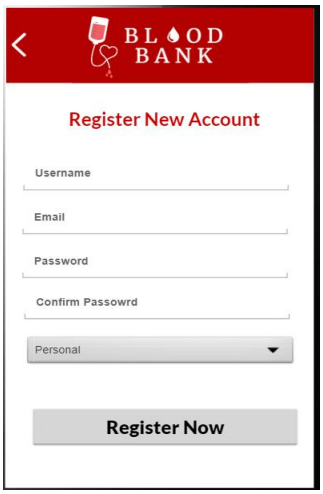


Figure 3.14 Register Account Interface

Figure 3.15 shows the check email interface of Blood Bank Application. Users are needed to insert the registered email and click the "Check Email" button. An email is send to the email address to reset the password.

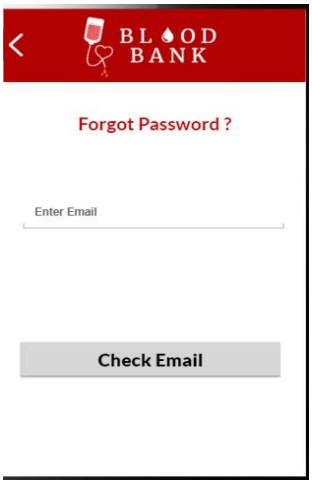


Figure 3.15 Check Email Interface

Figure 3.16 shows the login interface of Blood Bank Application. The users are required to login by inserting the correct email and password, then click the “Login” button. System will redirect to the home page interface when login is successful.

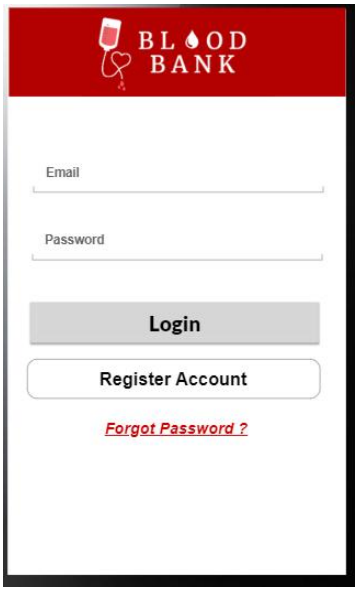


Figure 3.16 Login Interface

Figure 3.17 shows the home page interface for donation organiser and blood donor user of Blood Bank Application. Users are able to search the hospital or blood bank name for viewing its blood stock status. The home page also shows the information about the blood type compatibility. The profile icon represents the button to view profile interface.

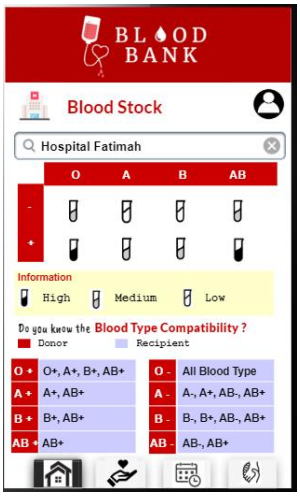


Figure 3.17 Home Page Interface for Donation Organiser and Blood Donor User

Figure 3.18 shows the home page interface for hospital and blood bank organisations of Blood Bank Application. The home page displays the organisation's blood stock. The home page also shows the information about the blood type compatibility. Users are able to update the blood stock status by clicking the "Update" button, the system will redirect to update blood stock interface. The profile icon represents the button to view profile interface.

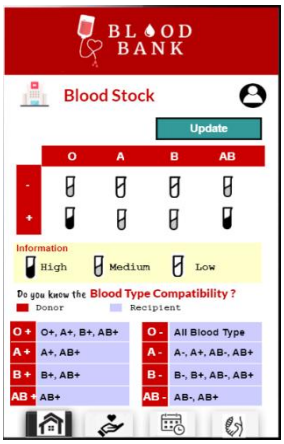


Figure 3.18 Home Page Interface for Hospital and Blood Bank Organisations

Figure 3.19 shows the view profile interface of Blood Bank Application. User able to view their own profile. They are also able to update the profile information by clicking the "Update" button by redirecting to the update profile interface.

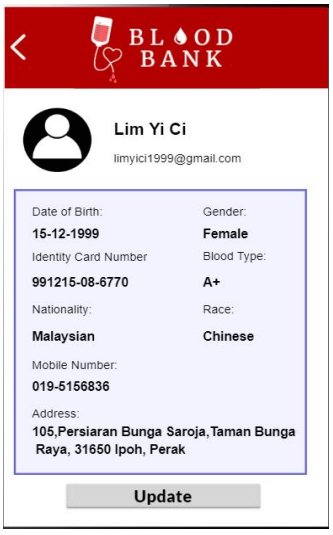


Figure 3.19 View Profile Interface

Figure 3.20 shows the update profile interface of Blood Bank Application. Users are able to update their profile information, then click the “Update” button after finishing inserting.

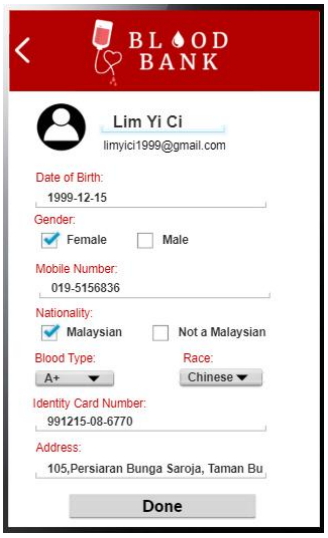


Figure 3.20 Update Profile Interface

Figure 3.21 shows the update blood stock interface of Blood Bank Application. Users are able to choose the blood stock status of each blood type. The status will be updated after clicking the “Update” button. The system will retrieve back to the home page after updating.

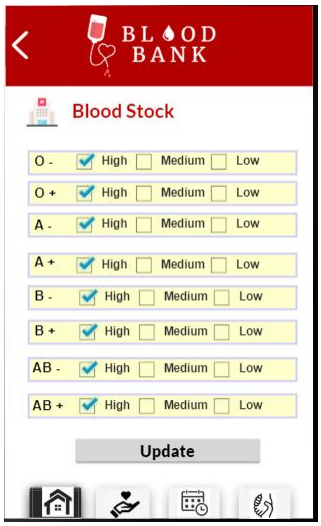


Figure 3.21 Update Blood Stock Interface

Figure 3.22 shows the view donation activity interface for blood donor users of Blood Bank Application. Users are able to view the upcoming donation activities such as the time, date and the venue which is displayed by Google map.

Figure 3.22 View Donation Activity Interface for Blood Donor

Figure 3.23 shows the view donation activity interface for hospital, blood bank and donation organiser of Blood Bank Application. Users are able to view the upcoming donation activities such as the time, date and the venue which is displayed by Google map. Clicking the "Create Donation" button will be retrieved to create donation activity interface.

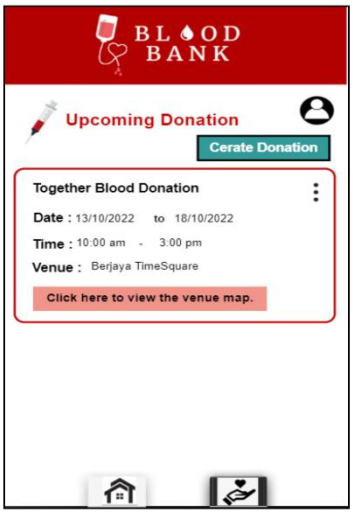


Figure 3.23 View Donation Activity Interface for Hospital, Blood Bank and Donation Organiser

Figure 2.24 shows the map activity interface of Donation Activity. User need to click the search bar in order to display the venue location in the map. The red location shows the venue location.

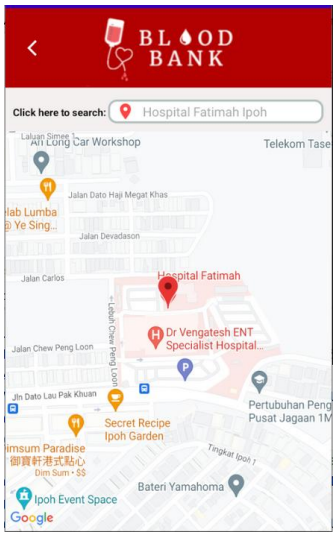


Figure 3.24 Map Activity Interface of Donation Activity

Figure 3.25 shows the create donation activity interface of Blood Bank Application. Users are required to insert the activity date and time. The activity will be created after clicking the "Create" button. Then the system will back to view donation activity interface.

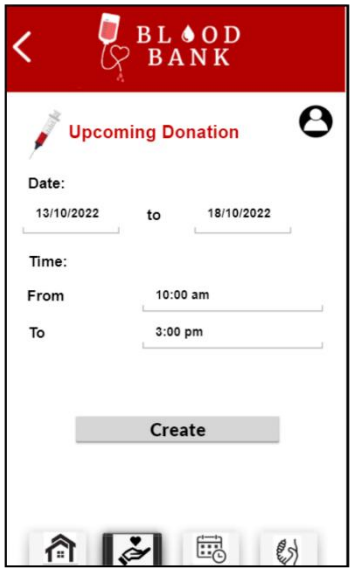


Figure 3.25 Create Donation Activity Interface

Figure 3.26 shows the delete donation activity interface of Blood Bank Application. Users needed to click the kebab menu for showing the delete button and edit button. The activity will be deleted after clicking the “Delete” button. When the “Edit” button is clicked, the system will retrieve to update donation activity interface.

Figure 3.26 Delete Donation Activity Interface

Figure 3.27 shows the update donation activity interface of Blood Bank Application. Users are able to update the donation activity information and click the “Update” button to update the database. The system will back to view donation activity interface.

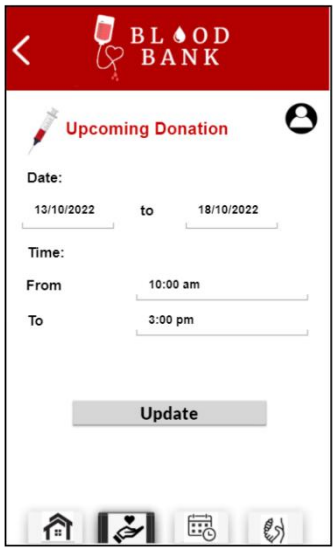


Figure 3.27 Update Donation Activity Interface

Figure 3.28 shows the view appointment interface for blood donor users of Blood Bank Application. Users are able to view the booked appointment information such as the venue, date and time. The "Make Appointment" will retrieve users to create appointment interface.

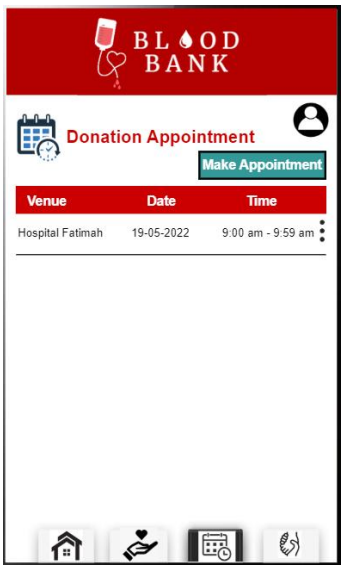


Figure 3.28 View Appointment Interface

Figure 3.29 shows the delete appointment interface for blood donor users of Blood Bank Application. Users are able to delete the appointment by clicking the kebab menu "Delete" button, then the appointment will be deleted from the database.

Figure 3.29 Delete Appointment Interface

Figure 3.30 shows the book appointment interface of Blood Bank Application. Users are able to insert the booking appointment information then click the “Book” button for booking. Then the system will retrieve to view appointment interface.

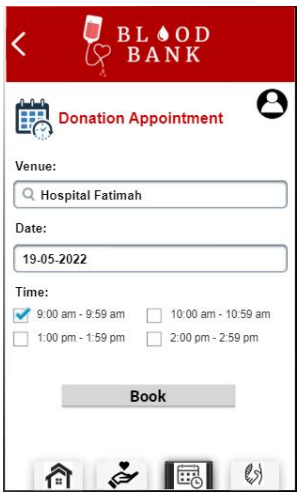


Figure 3.30 Book Appointment Interface

Figure 3.31 shows the update appointment interface of Blood Bank Application. Users update the appointment information and click the “Update” button for updating the information. The system will retrieve to view appointment interface after complete update.

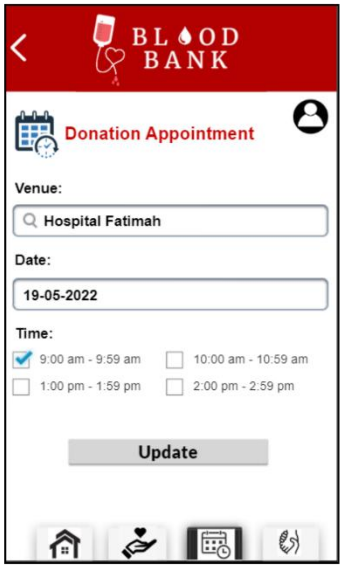


Figure 3.31 Update Appointment Interface

Figure 3.32 shows the view emergency donation interface for blood donor users of Blood Bank Application. The emergency donation will be displayed along with the blood type required and the description.

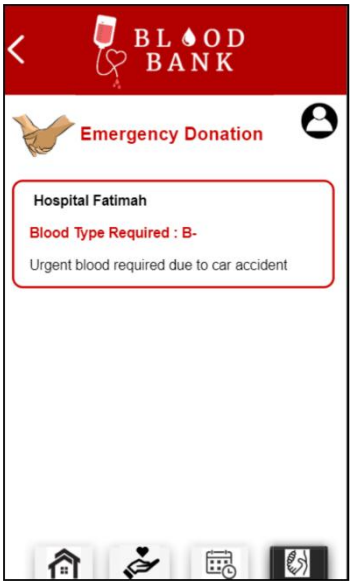


Figure 3.32 View Emergency Donation Interface for Blood Donor

Figure 3.33 shows the view emergency donation interface for hospital and blood bank organisation of Blood Bank Application. Their own emergency donation will be displayed along with the blood type required and the description. The "Create Emergency" button will retrieve to create emergency donation interface after clicking it.

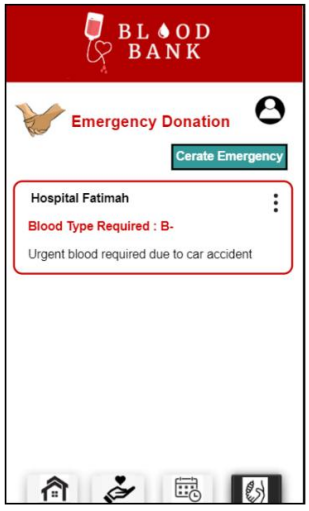


Figure 3.33 View Emergency Donation Interface for Hospital

and Blood Bank Organisation

Figure 3.34 shows the create emergency donation interface for hospital and blood bank organisation of Blood Bank Application. Users are required to fill up the emergency donation information and click the "Create" button in order to create the new emergency donation. The system is back to view emergency donation interface.

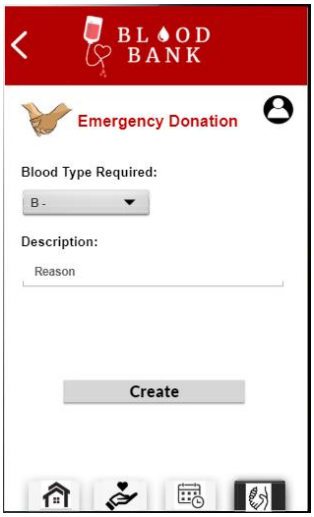


Figure 3.34 Create Emergency Donation Interface for Hospital and Blood Bank Organisation

Figure 3.35 shows the delete emergency donation interface of Blood Bank Application. Users are able to delete the emergency donation by clicking the kebab menu "Delete" button, then the emergency donation will be deleted from the database. For the "Edit" button, the system will retrieve to update emergency donation interface.

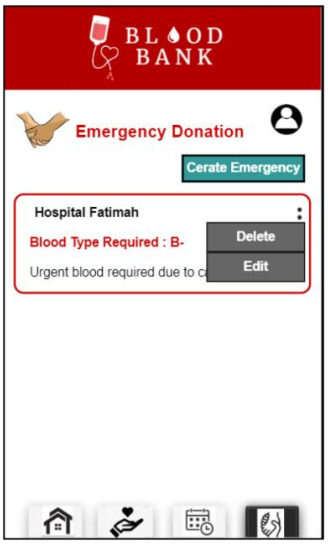


Figure 3.35 Delete Emergency Donation Interface

Figure 3.36 shows the update emergency donation interface of Blood Bank Application. Users are needed to fill up the emergency donation information and click the "Update" button for updating the information. The system will retrieve to view emergency donation interface after complete update.

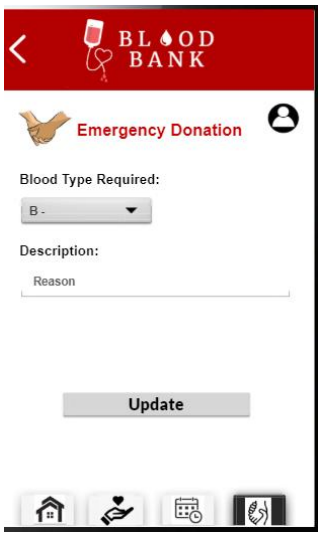


Figure 3.36 Update Emergency Donation Interface

3.9 Testing Plan

In the testing part of the application, User Acceptance Test(UAT) is used. UAT is a software testing process to ensure that it can perform required duties in real-world circumstances while adhering to specifications.

Each test activity is designed and implemented according to the function in each module, such as display, search and update. The test activities are shown in the table below.

No.	Module	Activities	Expected Result	Status		Actual Result	Comments
1.	Sign Up	Register Account	An account will be registered using the inserted email and password.	Yes ☐	No ☐		-
		Reset Password	The password will be reset.	Yes ☐	No ☐		-
		Login	Redirect to the home page using the password and email.	Yes ☐	No ☐		-
		Check Email	System send reset password email to the address.	Yes ☐	No ☐		-
2.	Activity	Display upcoming activities	Inserted upcoming activities will be displayed.	Yes ☐	No ☐		-
		Display Google map	Google maps will be displayed according to the venue given.	Yes ☐	No ☐		-
		Display others profile	Display others profile	Yes ☐	No ☐		-
		Delete activity	The activity chosen to delete will be removed from the database.	Yes ☐	No ☐		-

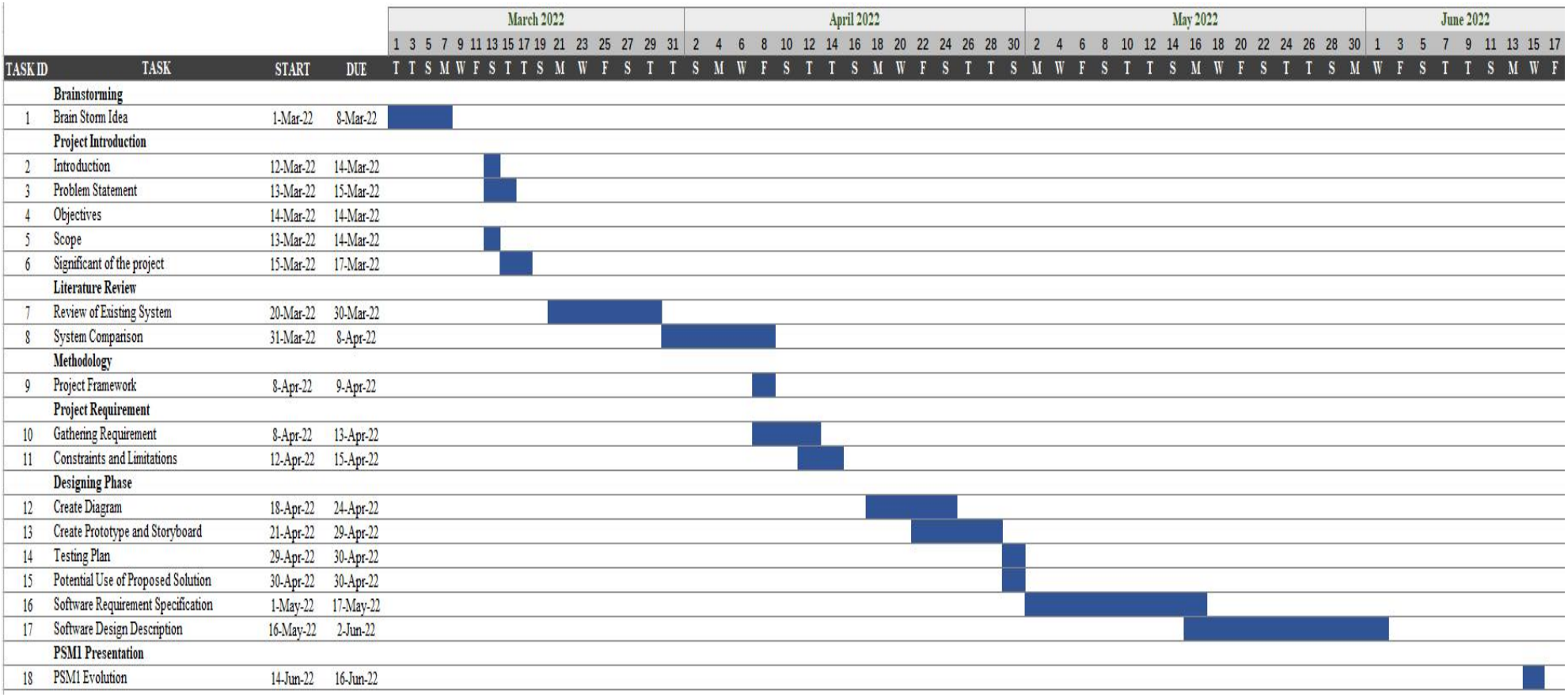
		Create activity	An activity will be created by inserting the time and date.	Yes ☐	No ☐		-
		Update activity	The time or date will be updated from the chosen activity.	Yes ☐	No ☐		-
3.	Blood Stock	Display blood stock status	The home page will display the hospital stock status.	Yes ☐	No ☐		-
		Search hospital	Able to search hospital name to display the hospital blood stock status.	Yes ☐	No ☐		-
		Update blood stock status	Blood stock status will be updated.	Yes ☐	No ☐		-
4.	Appointment	Display appointment	Booked appointments will be displayed on the Appointment main page.	Yes ☐	No ☐		-
		Create appointment	An appointment will be created by inserting the time, date and venue.	Yes ☐	No ☐		-
		Delete appointment	The selected appointment will be deleted from the database.	Yes ☐	No ☐		-

		Update appointment	The updated appointment information will be update to the database.	Yes ☐	No ☐		-
5.	Emergency	Display emergency donation	Emergency donations will be displayed on the Emergency Donation main page.	Yes ☐	No ☐		-
		Create emergency donation	An emergency donation will be created by inserting the description and the blood type needed.	Yes ☐	No ☐		-
		Update emergency donation	The blood type needed or description of the selected emergency donation will be updated and display the new information on the Emergency Donation mainpage.	Yes ☐	No ☐		-
		Delete emergency donation	The selected emergency donation will be deleted from the database.	Yes ☐	No ☐		-

This test has been performed by:
Name : Lim Yi Ci

Signature	: LimYiCi
Date	: 23 May 2022

Table 3.14 UAT Testing Plan



TASK ID	TASK	START	DUE
	Brainstorming		
1	Brain Storm Idea	1-Mar-22	8-Mar-22
	Project Introduction		
2	Introduction	12-Mar-22	14-Mar-22
3	Problem Statement	13-Mar-22	15-Mar-22
4	Objectives	14-Mar-22	14-Mar-22
5	Scope	13-Mar-22	14-Mar-22
6	Significant of the project	15-Mar-22	17-Mar-22
	Literature Review		
7	Review of Existing System	20-Mar-22	30-Mar-22
8	System Comparison	31-Mar-22	8-Apr-22
	Methodology		
9	Project Framework	8-Apr-22	9-Apr-22
	Project Requirement		
10	Gathering Requirement	8-Apr-22	13-Apr-22
11	Constraints and Limitations	12-Apr-22	15-Apr-22
	Designing Phase		
12	Create Diagram	18-Apr-22	24-Apr-22
13	Create Prototype and Storyboard	21-Apr-22	29-Apr-22
14	Testing Plan	29-Apr-22	30-Apr-22
15	Potential Use of Proposed Solution	30-Apr-22	30-Apr-22
16	Software Requirement Specification	1-May-22	17-May-22
17	Software Design Description	16-May-22	2-Jun-22
	PSM1 Presentation		
18	PSM1 Evolution	14-Jun-22	16-Jun-22

Figure 3.37 Gantt Chart PSM1

3.10 Potential Use of Proposed Solution

The blood bank application is developed for Malaysian citizens as a media for obtaining blood donation information as it cooperates with the legal organisations. Blood bank applications are potential to be the essential application as blood donation is advocated to everyone. Through the blood bank application, it increases the numbers of blood donating either for donor or receiver.

Chapter 4

Implementation, Result and Discussion

4.1 Introduction

The implementation and development of the suggested mobile application will be discussed in this chapter. The proposed mobile application will be described in-depth and accurately, and it will serve as the answer to the issue raised in Chapter 1's. Along with discussion of the functionalities, a description will be given.

4.2 Development Environment

4.2.1 Integrated Development Environment (IDE)

To propose this Blood Bank Application, Android Studio IDE is used as the software implementation. As opposed to Kotlin, Java is used. Additionally, Android Studio will work with the cloud service Firebase**.**

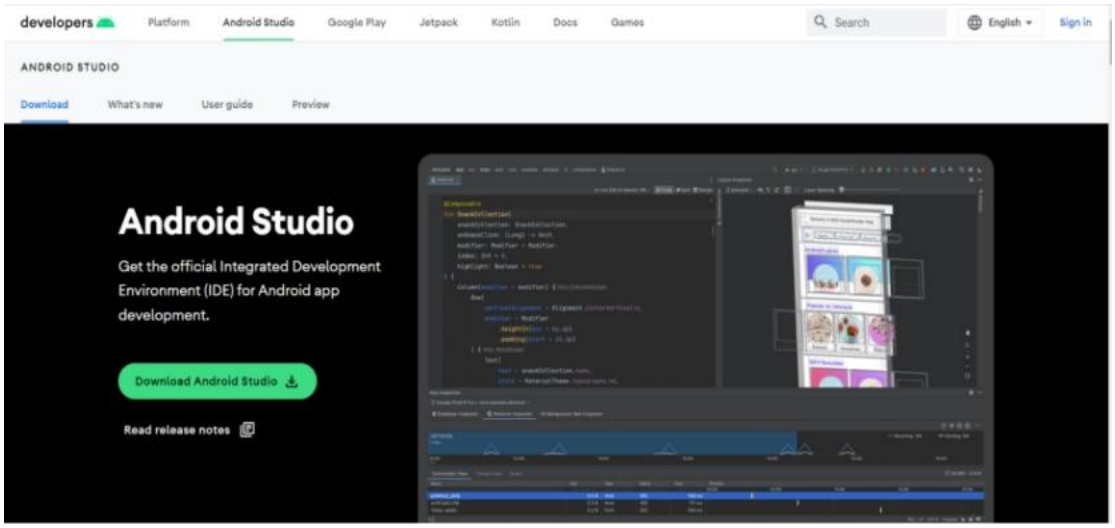


Figure 4.1 Android Studio

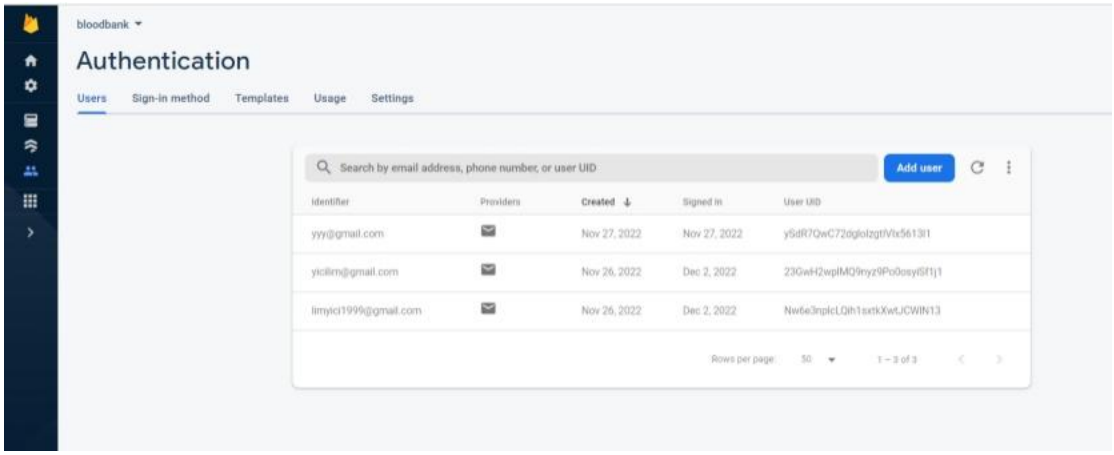


Figure 4.2 User Authentication of Blood Bank Application

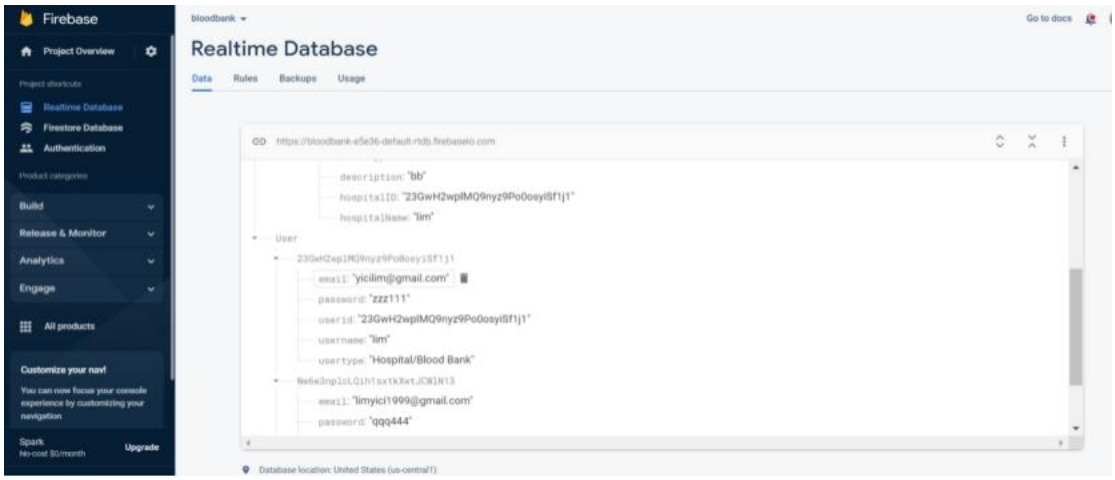


Figure 4.3 Realtime Database of Blood Bank Application

```
mAuth.createUserWithEmailAndPassword(email,password)
        .addOnCompleteListener(task -> {

            if(task.isSuccessful()){
                User user=new User(email,password,username,type,FirebaseAuth.getInstance().getCurrentUser().getUid());
                FirebaseDatabase.getInstance().getReference( path: "User")
                        .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
                        .setValue(user).addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if(task.isSuccessful()){
                            Toast.makeText( context: RegisterAccount.this, text: "Successfully Registered.",Toast.LENGTH_SHORT).show();

                            startActivity(new Intent( packageContext: RegisterAccount.this,login.class));

                        }else{
                            Toast.makeText( context: RegisterAccount.this, text: "Failed to register!" + task.getException(), Toast.LE
                        }
                    }
                });
            }else{
                Toast.makeText( context: RegisterAccount.this, text: "Failed to register!" + task.getException(), Toast.LENGTH_SHORT).
            }
        });
```

Figure 4.4 Sign Up User

```
mAuth.signInWithEmailAndPassword(loginemail,loginpassword).addOnCompleteListener(task->
{
    if(task.isSuccessful())
    {

        Toast.makeText( context: login.this, text: "Correct password and email.", Toast.LENGTH_SHORT).show();
        queryA= mFirebaseDatabaseReference.child("User").child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child("");
        ValueEventListener eventA=new ValueEventListener()
        {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot)
            {
                successLoginA(snapshot);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }

            private void successLoginA(DataSnapshot snapshot)
            {

                String gettype=snapshot.child("usertype").getValue().toString();
                if (snapshot.exists())
                {
```

Figure 4.5 Login User

```
emailbtn.setOnClickListener(v->
{
    String getemail=searchemail.getText().toString().trim();
    if(getemail.isEmpty()){
        searchemail.setError("Please fill in the correct email.");
        searchemail.requestFocus();
        return;
    }else
    {
        auth.sendPasswordResetEmail(getemail)
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if(task.isSuccessful())
                        {
                            Toast.makeText( context: CheckEmail.this, text: "Check your email",Toast.LENGTH_SHORT).show();
                            startActivity(new Intent( packageContext: CheckEmail.this,login.class));
                            finish();
                        }
                        else
                        {
                            Toast.makeText( context: CheckEmail.this, text: "Error: "+task.getException(),Toast.LENGTH_SHORT).show();
                        }
                    }
                });
```

Figure 4.6 Check Email

```
EmergencyInfo=new ArrayList<>();
databaseReference= FirebaseDatabase.getInstance().getReference( path: "Emergency");
QueryEmergency querydonor=new QueryEmergency();

querydonor.get(key).addValueEventListener(new ValueEventListener()
{
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        EmergencyInfo.clear();

        for(DataSnapshot emergencyDataSnap :snapshot.getChildren())
        {
            Emergency emergency=emergencyDataSnap.getValue(Emergency.class);
            emergency.setKey(emergencyDataSnap.getKey());
            EmergencyInfo.add(emergency);

            key=emergencyDataSnap.getKey();
        }

        EmergencyAdapter adapter=new EmergencyAdapter( mContext: HospitalEmergency.this,EmergencyInfo);
        emergencyview.setAdapter(adapter);

    }
```

Figure 4.7 Hospital and Blood Bank Emergency Donation

```
create.setOnClickListener(v->
{
    if(emergency_edit==null)
    {
        String description=editDescription.getText().toString().trim();
        String savename=getname;

        if(description.isEmpty()){
            editDescription.setError("Please fill in the reason.");
            editDescription.requestFocus();
            return;
        }

        emergency=new Emergency(type,description,savename,userid);
        query.add(emergency).addOnSuccessListener(suc ->
        {
            Toast.makeText( context: this,  text: "Successfully save", Toast.LENGTH_LONG).show();
            startActivity(new Intent( packageContext: CreateEmergency.this, HospitalEmergency.class));

        }).addOnFailureListener(er ->
        {
            Toast.makeText( context: this,  text: "" + er.getMessage(), Toast.LENGTH_LONG).show();
            startActivity(new Intent( packageContext: CreateEmergency.this, HospitalEmergency.class));

        });
    }
```

Figure 4.8 Hospital and Blood Bank Create Emergency Donation

```
AppointmentInfo=new ArrayList<>();
databaseReference= FirebaseDatabase.getInstance().getReference( path: "Appointment");
QueryAppointment query=new QueryAppointment();

query.get(key).addValueEventListener(new ValueEventListener()
{
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot)
    {
        AppointmentInfo.clear();

        for(DataSnapshot appointmentDataSnap :snapshot.getChildren())
        {
            Appointment appointment=appointmentDataSnap.getValue(Appointment.class);
            appointment.setKey(appointmentDataSnap.getKey());
            AppointmentInfo.add(appointment);

            key=appointmentDataSnap.getKey();
        }

        DonorAppointmentAdapter adapter=new DonorAppointmentAdapter( mContext: DonorAppointment.this,AppointmentInfo);
        appointmentview.setAdapter(adapter);

    }
```

Figure 4.9 Personal Donation Appointment

```
    appointment=new Appointment(savedate,timeresult,getvenue,userid,hospID);
    query.add(appointment).addOnSuccessListener(suc ->
    {
        Toast.makeText( context: this,  text: "Successfully save", Toast.LENGTH_LONG).show();
        startActivity(new Intent( packageContext: DonorCreateAppointment.this, DonorAppointment.class));

    }).addOnFailureListener(er ->
    {
        Toast.makeText( context: this,  text: "" + er.getMessage(), Toast.LENGTH_LONG).show();
        startActivity(new Intent( packageContext: DonorCreateAppointment.this, DonorAppointment.class));

    });
}
else
{
    HashMap<String,Object> hashMap=new HashMap<>();
    hashMap.put("venue",venue.getText().toString());
    hashMap.put("date",getdate.getText().toString());
    hashMap.put("time",timeresult);
    query.update(appointment_edit.getKey(),hashMap).addOnSuccessListener(suc ->
    {
        Toast.makeText( context: this,  text: "Successfully save", Toast.LENGTH_LONG).show();
        startActivity(new Intent( packageContext: DonorCreateAppointment.this, DonorAppointment.class));

    }).addOnFailureListener(er ->
```

Figure 4.10 Personal Create Donation Appointment

```
}
HashMap hosp=new HashMap();
hosp.put("hospAddress",address.getText().toString());
hosp.put("hospOpTime",hour.getText().toString());
hosp.put("hospName",hospname.getText().toString());
hosp.put("hospPhone",phone.getText().toString());

DatabaseReference databaseReference=FirebaseDatabase.getInstance().getReference( path: "HospitalBloodBank");
databaseReference.child(userid).updateChildren(hosp).addOnCompleteListener(new OnCompleteListener() {
    @Override
    public void onComplete(@NonNull Task task) {
        if(task.isSuccessful())
        {
            Toast.makeText( context: HospitalUpdateProfile.this,  text: "Successfully save", Toast.LENGTH_SHORT).show();
            startActivity(new Intent( packageContext: HospitalUpdateProfile.this, HospitalViewProfile.class));

        }else
        {
            Toast.makeText( context: HospitalUpdateProfile.this,  text: "Update failed.", Toast.LENGTH_SHORT).show();
            startActivity(new Intent( packageContext: HospitalUpdateProfile.this, HospitalViewProfile.class));
        }
```

Figure 4.11 Hospital and Blood Bank Update Profile

```
private void search2(DataSnapshot snapshot){
    if(snapshot.exists())
    {
        for(DataSnapshot da:snapshot.getChildren())
        {
            String getadd=da.child("hospAddress").getValue(String.class);
            String gettime=da.child("hospOpTime").getValue(String.class);
            String getphone=da.child("hospPhone").getValue(String.class);
            String id2=da.child("hospID").getValue(String.class);

            if(id2.equals(userid))
            {
                ophour.setText(gettime);
                address.setText(getadd);
                hospphone.setText(getphone);
            }
```

Figure 4.12 Hospital and Blood Bank View Profile

```
    upcomingActivity=new UpcomingActivity(savestarttime,saveendtime,savestartdate,saveenddate,savevenue,
    query.add(upcomingActivity).addOnSuccessListener(suc ->
    {
        Toast.makeText( context: this,  text: "Successfully save", Toast.LENGTH_LONG).show();
        startActivity(new Intent( packageContext: HospitalCreateActivity.this, HospitalActivity.class));

    }).addOnFailureListener(er ->
    {
        Toast.makeText( context: this,  text: "" + er.getMessage(), Toast.LENGTH_LONG).show();
        startActivity(new Intent( packageContext: HospitalCreateActivity.this, HospitalActivity.class));

    });
}
else
{
    HashMap<String,Object> hashMap=new HashMap<>();
    hashMap.put("startTime",starttime.getText().toString());
    hashMap.put("endTime",endtime.getText().toString());
    hashMap.put("startDate",startdate.getText().toString());
    hashMap.put("endDate",enddate.getText().toString());
    hashMap.put("activityVenue",activityvenue.getText().toString());

    query.update(activity_edit.getKey(),hashMap).addOnSuccessListener(suc ->
    {
        Toast.makeText( context: this,  text: "Successfully save", Toast.LENGTH_LONG).show();
```

Figure 4.13 Hospital and Blood Bank Create Upcoming Donation

```
mMap = googleMap;
Dexter.withContext(getContext())
        .withPermission(Manifest.permission.ACCESS_FINE_LOCATION)
        .withListener(new PermissionListener() {
            @Override
            public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {
                if (ActivityCompat.checkSelfPermission(getContext(), Manifest.permission.ACCESS_FINE_LOCATION)
                        != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(getContext(),
                        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
                    return;
                }
                mMap.setMyLocationEnabled(true);
                fusedLocationProviderClient.getLastLocation().addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        Toast.makeText(getContext(), text: "error"+e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }).addOnSuccessListener(new OnSuccessListener<Location>() {
                    @Override
                    public void onSuccess(Location location) {
                        LatLng latLng=new LatLng(location.getLatitude(),location.getLongitude());
                        mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, zoom: 17));
```

Figure 4.14 Map View of Upcoming Donation Venue

```
if(snapshot.exists())
{
    for(DataSnapshot da:snapshot.getChildren())
    {
        String savedid=da.child("hospID").getValue(String.class);
        if(savedid.equals(userid))
        {
            HashMap hashMap=new HashMap<>();
            hashMap.put("typeOP",typeOP);
            hashMap.put("typeON",typeON);
            hashMap.put("typeAP",typeAP);
            hashMap.put("typeAN",typeAN);
            hashMap.put("typeBP",typeBP);
            hashMap.put("typeBN",typeBN);
            hashMap.put("typeABP",typeABP);
            hashMap.put("typeABN",typeABN);
            databaseReference.child(key).updateChildren(hashMap).addOnCompleteListener(new OnCompleteListener() {
                @Override
                public void onComplete(@NonNull Task task) {
                    Toast.makeText( context: UpdateBloodStock.this, text: "Successfully save", Toast.LENGTH_LONG).show();
                    startActivity(new Intent( packageContext: UpdateBloodStock.this, HospitalHome.class));
                }
            }).addOnFailureListener(er ->
```

Figure 4.15 Update Blood Stock Status

```
private void check()
{
    findname=search.getText().toString().trim();
    databaseReference= FirebaseDatabase.getInstance().getReference( path: "BloodStock");
    ValueEventListener event=new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) { display(snapshot); }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    };
    databaseReference.addValueEventListener(event);


}
```

Figure 4.16 Display Blood Stock

4.3 System Output and Result

A system or application's user interface is a crucial component. User interface is the means by which people and computers interact and communicate. Effective control and interaction with the computer are made possible by a suitable user interface. The proposed mobile application's user interfaces are designed and presented in this section.

4.3.1 Application User Interface

The Blood Bank application enables users to create an account and log in. If a user does not already have an account, they must first create one.

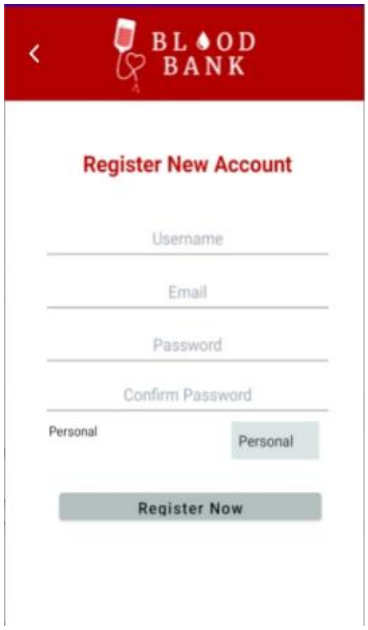


Figure 4.17 Register Account Interface

A valid email address, a username with at least six characters, and a password with at least six characters are required for the user to create an account. User also need to choose a user type before clicking the “Register Now” button. An error message will be display when the user does not fill in the username, incorrect email address format and the confirm password is not match with the password. Account will be created once the “Register Now” button is clicked.

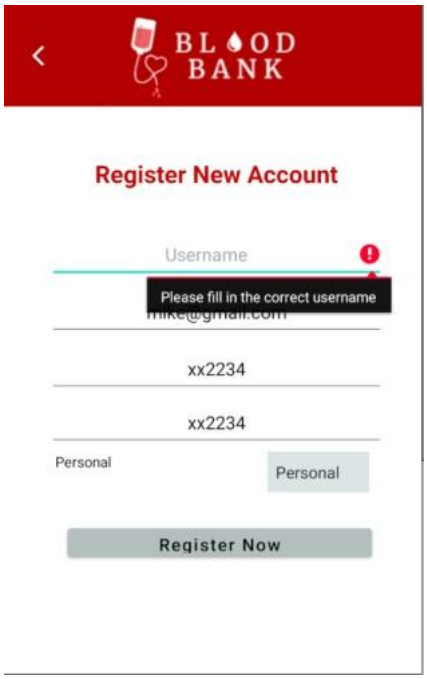


Figure 4.18 Error Message for No Input Username

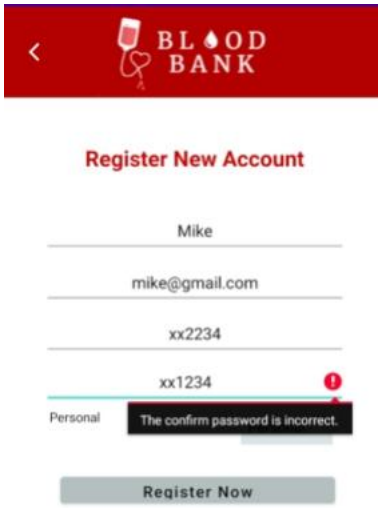


Figure 4.19 Error Message for Wrong Confirm Password

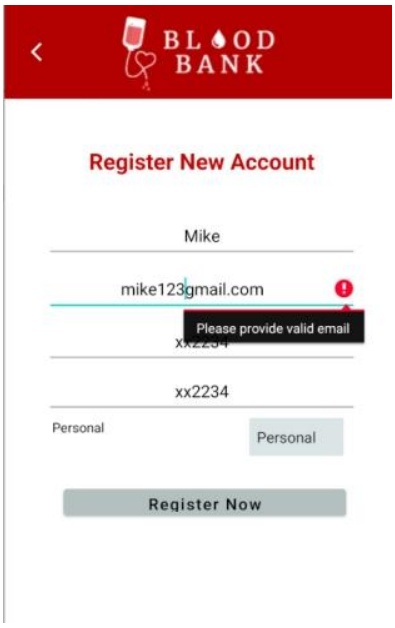


Figure 4.20 Error Message for Incorrect Email Address Format

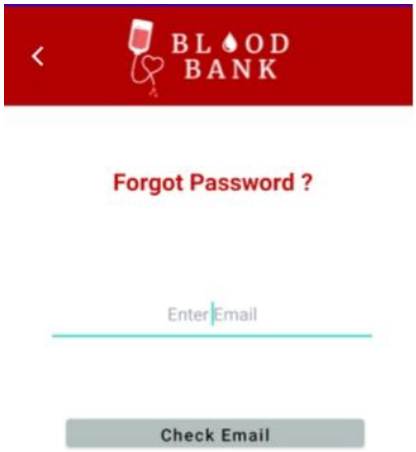


Figure 4.21 Forgot Password Interface

Figure above shows the interface for forgot password. User are required to insert their email and a reset password email is send to the according to the input email address.

Figure 4.22 Login Interface

Figure above shows the login interface. User required to insert their registered email, password, and account type to login. User will be redirected to their home page according to their user type. If the email, password and user type is incorrect, error message will pop up.

Figure 4.23 Error Message for Incorrect Email and Password

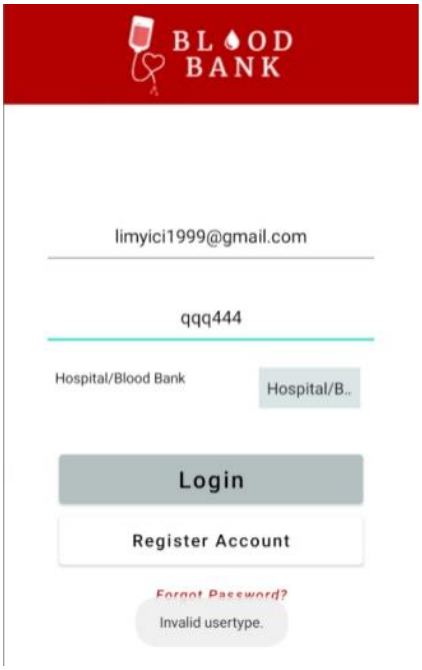


Figure 4.24 Error Message for Incorrect User Type

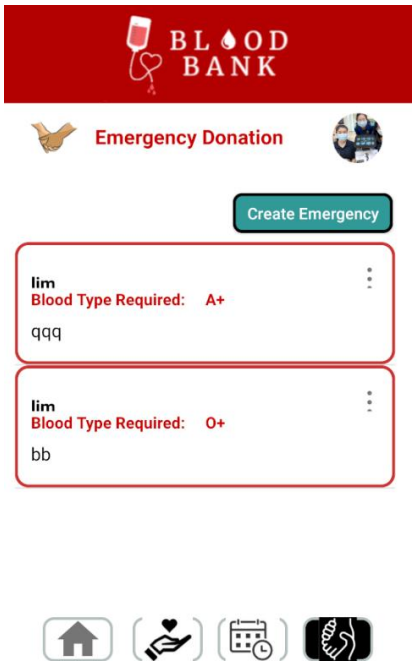


Figure 4.25 Emergency Donation Interface for Hospital and Blood Bank

Figure above shows the emergency donation for hospital and blood bank user type. All their created notice will display in it. They are able to create an emergency notice by clicking on the “Create Emergency” button. Then the create emergency interface will be shown.

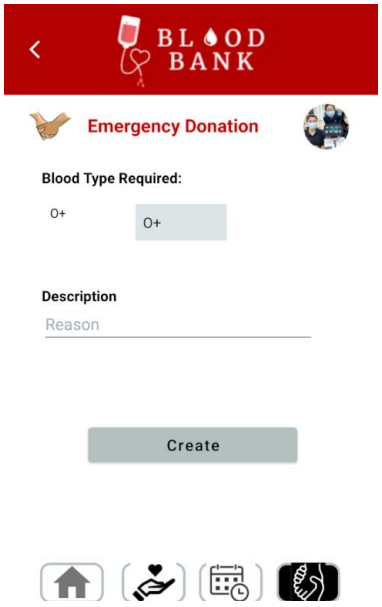


Figure 4.26 Create Emergency Donation Interface

Figure above shows the create emergency donation interface. Hospital and blood bank needed to choose the blood type required and fill in the description. If the description is not fill in, an error message will be display.

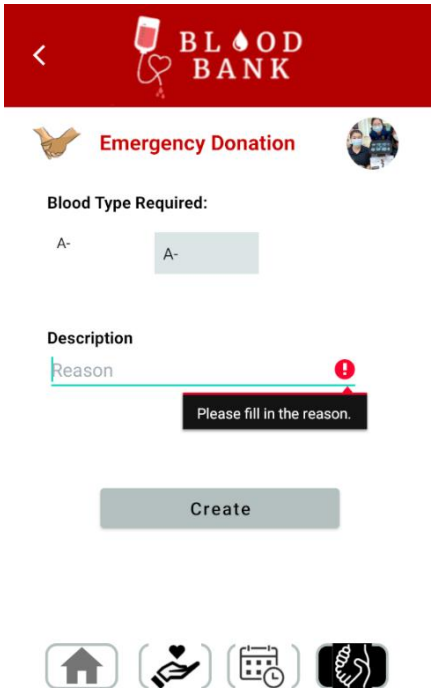


Figure 4.27 Error Message for Empty Description

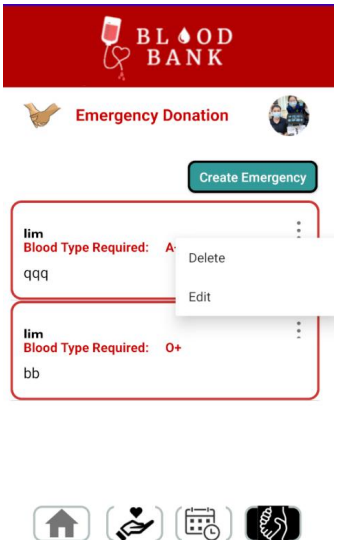


Figure 4.28 Option Menu in Emergency Donation for Hospital and Blood Bank

Figure above shows the option menu when the kebab menu(3 dots button) is clicked. User able to choose for delete the emergency notice or edit it.

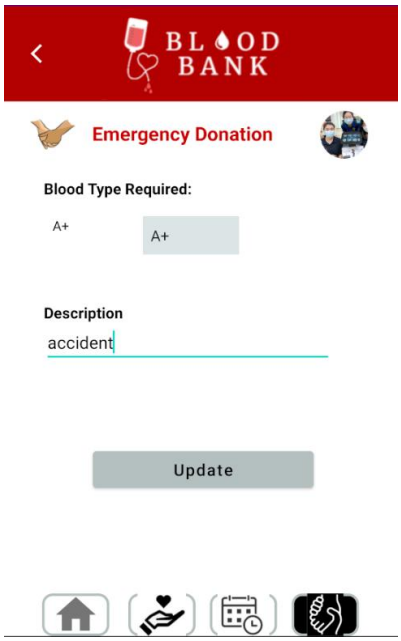


Figure 4.29 Edit Emergency Donation Interface

Figure above shows the interface to edit the emergency donation notice. After clicking the “Update” button, the notice will be updated.

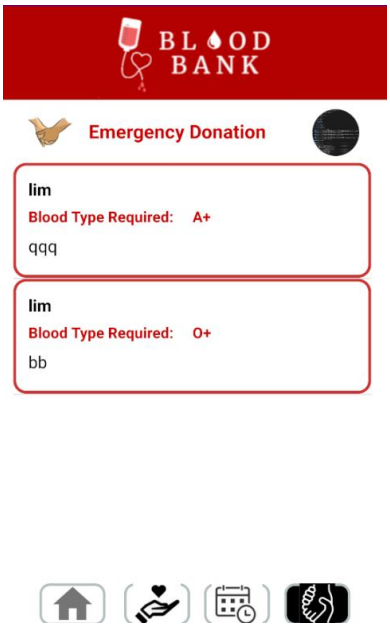


Figure 4.30 Emergency Donation Interface for Personal

Figure above shows the emergency donation interface for personal user. They are able to view all the emergency donation notices from all hospital and blood bank.

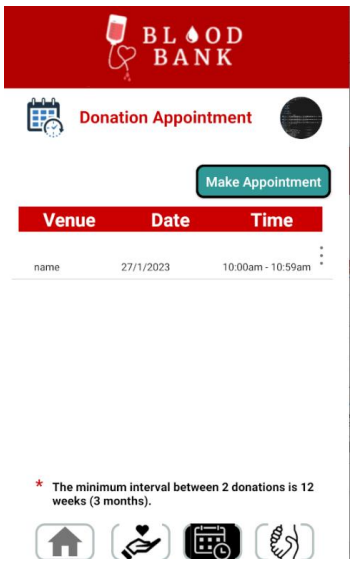


Figure 4.31 Donation Appointment Interface for Personal

Figure above shows the donation appointment interface for personal user. They able to make an appointment for donation by clicking the “Make Appointment” button. Then, user will be redirected to book appointment interface.

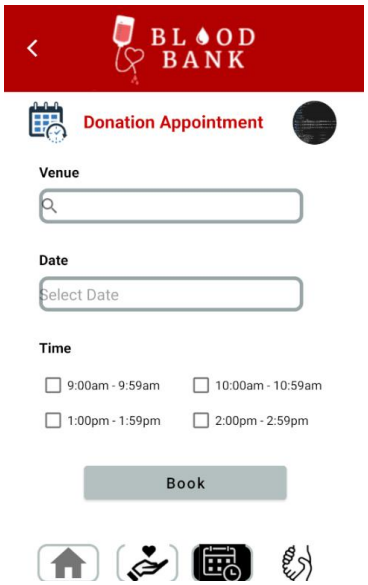


Figure 4.32 Book Appointment Interface

Figure above is the interface for booking a donation appointment. User needed to insert a venue, choose a date and a slot. Then click "Book" button to make appointment. If the venue, date, and time are not fill, error message will be display.

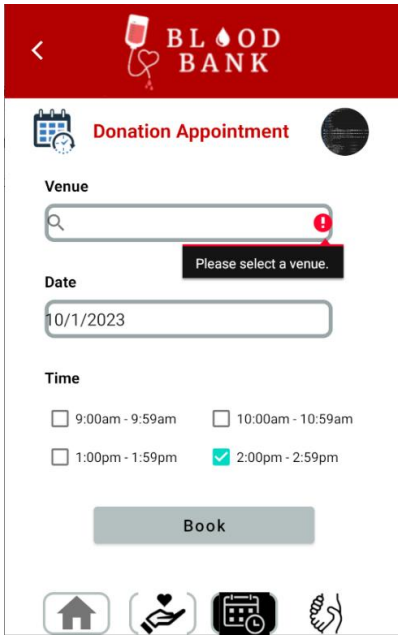


Figure 4.33 Error Message for Empty Venue

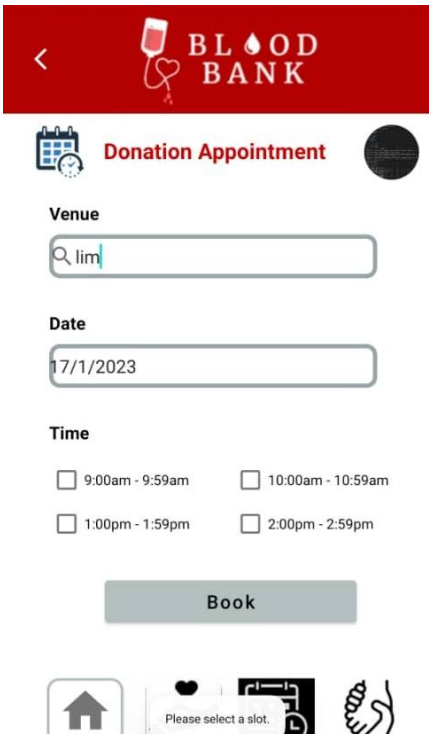


Figure 4.34 Error Message for Empty Slot

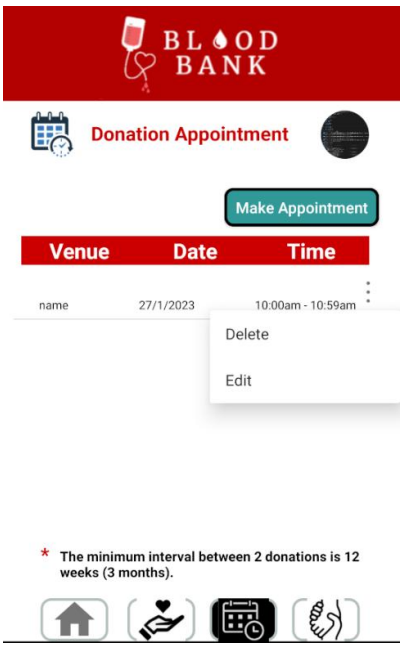


Figure 4.35 Option Menu in Donation Appointment for Personal

Figure above shows the option menu when the kebab menu(3 dots button) is clicked. User able to choose for delete the donation appointment or edit it.

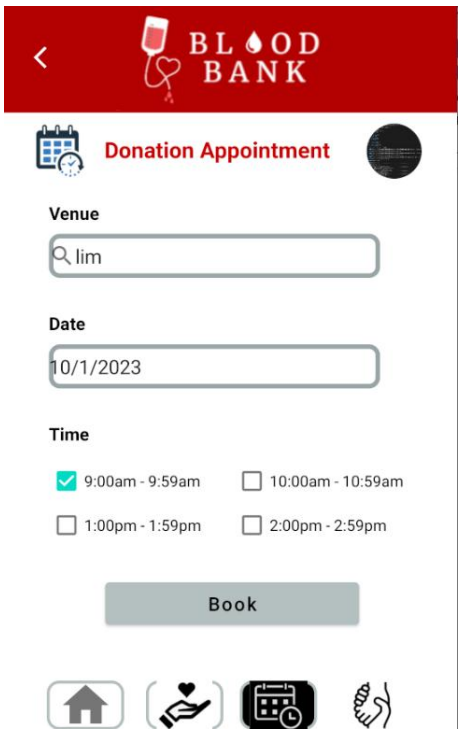


Figure 4.36 Edit Donation Appointment Interface

Figure above shows the interface to edit the donation appointment. After clicking the "Update" button, the donation appointment information will be updated.

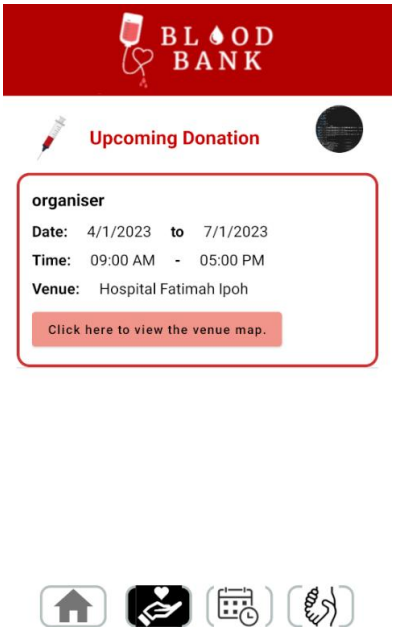


Figure 4.37 Upcoming Donation Interface

Figure above shows the interface of upcoming donation. It displays the starting date, starting time, end time, end date and venue. User clicks the "Click here to view the venue map", it will shows the map of the venue.

Figure 4.38 Venue Map Interface

Figure above shows the interface of venue map. User is needed to click the search bar to operate. The red location marker shows the venue in the map.

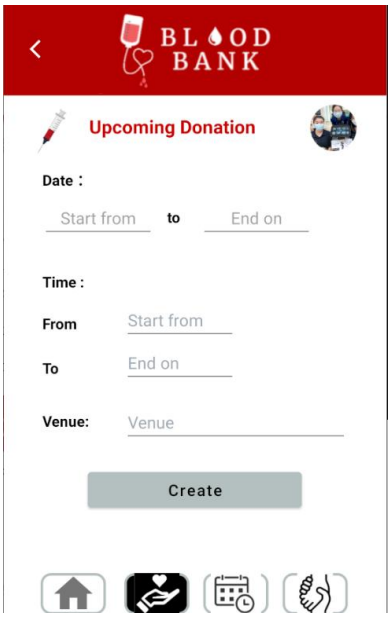


Figure 4.39 Create Upcoming Donation Interface

Figure 4.40 Date Picker for Create Upcoming Donation Interface

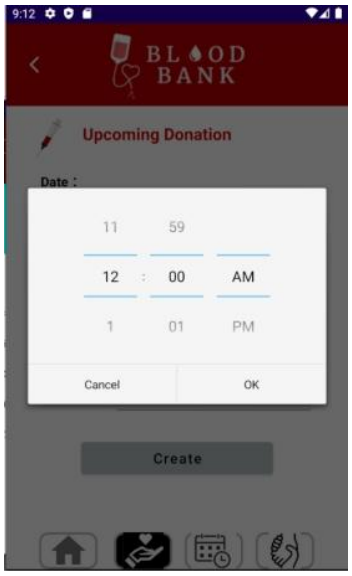


Figure 4.41 Time Picker for Create Upcoming Donation Interface

Figure above shows the interface of create upcoming donation. User needed to insert the starting date and time, close date and time and venue. Date picker will pop up when clicking the date input box. Time picker will pop up when double click the time input box.

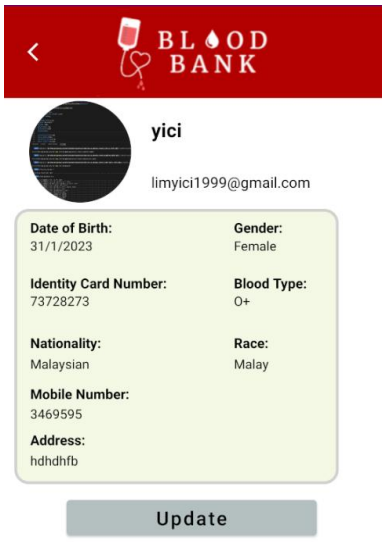


Figure 4.42 View Own Profile Interface

Figure above shows the view own profile interface. User able to view all the inserted own information in this interface. The "Update" button will redirect user to update the information.

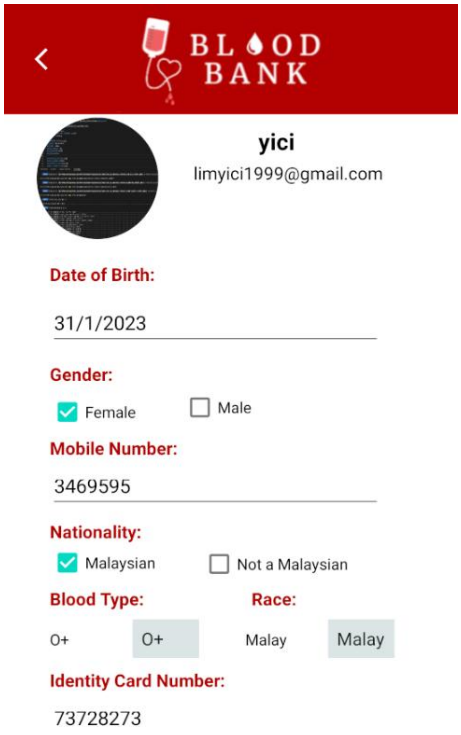


Figure 4.43 Update Profile Interface

Figure above shows the update profile interface. User able to edit their information and update it by clicking the "Done" button.

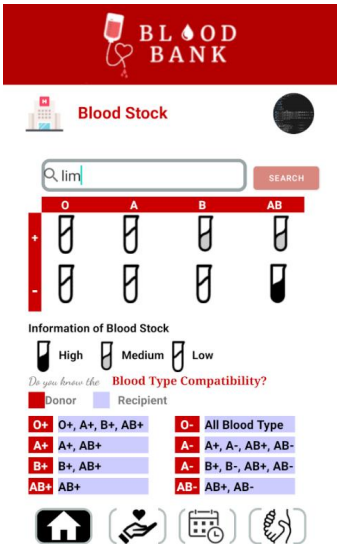


Figure 4.44 Home Interface

Figure above shows the home interface. User able to view the blood stock status by search the hospital or blood bank name in the search bar. Then the blood stock status of it will be display. The blood stock status information is listed in the interface.

Figure 4.45 Update Blood Stock Interface

Figure above shows update blood stock interface. User can choose the blood stock status by clicking the related check box of each blood type. The blood stock status will be updated after clicking the update button.

4.4 Testing and Result Discussion

This is the User Acceptance Test (UAT) of Malaysia Blood Bank Android Mobile Application – Blood Bank Application. Tests are performed on the application's functionality, and the test case, input, expected outcomes, actual outcomes and test status are stated. It will be acknowledged and discussed that for failed test case. The testers will send the feedback using Google Form. The google form result is shown in Appendix C.

The functions, text fields, buttons, data shown, and other aspects of each test case have all been thoroughly tested for each interface based on the interfaces in the Blood Bank application. The test cases are shown in the table below.

Module	Manage Sign Up				
Objective	To allow user sign up an account in the application.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**
TC01-01	Insert with blank input	Blank Input	Display error message	Display error message	Pass
TC01-02	Insert with wrong email or password	Incorrect email or password	Display error message	Display error message	Pass
TC01-03	Login using wrong email address format.	Incorrect email address format	Display error message	Display error message	Pass
TC01-04	Login with correct email and password and click “Login” button for login	Insert correct email and password and click “Login” button	Successfully login	Successfully login	Pass
TC01-05	Click “Register Account” to	Click “Register Account”	Redirect to register account	Redirect to register account	Pass

	register account		interface	interface	
TC01-06	Click "Forgot Password" to reset password	Click "Forgot Password"	Redirect to check email interface	Redirect to check email interface	Pass
TC01-07	Register account	Fill in all information and click "Register Now" button	Successfully register account	Successfully register account	Pass
TC01-08	Insert registered email when register account	Insert registered email and click "Register Now" button	Display error message	Display error message	Pass
TC01-09	Insert email for reset password	Insert email	Get rest password email	Get rest password email	Pass
TC01-10	Select wrong user type when login	Select wrong user type	Display error message	Display error message	Pass

Table 4.1 Sign Up Module Testing

Module	Manage Emergency Donation				
Objective	To allow user to view the emergency donation information. Hospital and Blood Bank able to create, update and delete the information.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**
TC-02-01	Insert with blank input	Blank Input	Display error message	Display error message	Pass
TC-02-02	Display emergency donation information	-	Display emergency donation information	Display emergency donation information	Pass

TC-02-03	Click "Create Emergency" button for create an emergency	Click "Create Emergency" button	Redirect to create emergency donation interface	Redirect to create emergency donation interface	Pass
TC-02-04	Create emergency donation notice	Insert emergency donation information and click "Create" button	Successfully create emergency donation notice	Successfully create emergency donation notice	Pass
TC-02-05	Display "Delete" and "Edit" option menu	Click on the kebab menu	Display "Delete" and "Edit" option menu	Display "Delete" and "Edit" option menu	Pass
TC-02-06	Delete an emergency donation notice	Click "Delete" button	Successfully delete emergency donation notice	Successfully delete emergency donation notice	Pass
TC-02-07	Edit an emergency donation notice	Click "Edit" button and update the information	Successfully update emergency donation notice	Successfully update emergency donation notice	Pass

Table 4.2 Emergency Donation Module Testing

Module	Manage Appointment				
Objective	To allow donor to make appointment and hospital and blood bank view the appointment details.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**

TC-03-01	Insert with blank input	Blank Input	Display error message	Display error message	Pass
TC-03-02	Display appointment information	-	Display appointment information	Display appointment information	Pass
TC-03-03	Go to make appointment interface	Click "Create Emergency" button	Redirect to make appointment interface	Redirect to make appointment interface	Pass
TC-03-04	Make an appointment	Insert appointment information and click "Book" button	Successfully make an appointment	Successfully make an appointment	Pass
TC-03-05	Display "Delete" and "Edit" option menu	Click on the kebab menu	Display "Delete" and "Edit" option menu	Display "Delete" and "Edit" option menu	Pass
TC-03-06	Delete an appointment	Click "Delete" button	Successfully delete appointment	Successfully delete appointment	Pass
TC-03-07	Edit an appointment	Click "Edit" button and update the information	Successfully update appointment	Successfully update appointment	Pass

Table 4.3 Appointment Module Testing

Module	Manage Profile				
Objective	To allow user to view and update their own profile.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**
TC-04-01	Display profile information	-	Display profile information	Display profile information	Pass
TC-04-02	Update profile information	Insert profile information and	Successfully update profile	Successfully update profile	Pass

		click "Done" button	information	information	
TC-04-03	Go to update profile interface	Click "Update" button	Redirect to update profile interface	Redirect to update profile interface	Pass
TC-04-04	Insert with blank input	Blank Input	Display error message	Display error message	Pass

Table 4.4 Profile Module Testing

Module	Manage Activity				
Objective	To allow hospital, donation organiser and blood bank to create, update, view and delete donation activity. Blood donors are only allowed to view the upcoming donation activities.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**
TC-05-01	Insert with blank input	Blank Input	Display error message	Display error message	Pass
TC-05-02	Display upcoming donation information	-	Successfully display upcoming donation information	Successfully display upcoming donation information	Pass
TC-05-03	Display venue location in Google Map	Click "Click here to view the venue map" button	Successfully display location map	Successfully display location map	Pass
TC-05-04	Go to create donation activity interface	Click "Create Donation" button	Redirect to create donation activity interface	Redirect to create donation activity interface	Pass
TC-05-05	Create a donation activity	Insert donation activity and click "Create"	Successfully create a donation activity	Successfully create a donation activity	Pass

		button			
TC-05-06	Display "Delete" and "Edit" option menu	Click on the kebab menu	Display "Delete" and "Edit" option menu	Display "Delete" and "Edit" option menu	Pass
TC-05-07	Delete a donation activity	Click "Delete" button	Successfully delete donation activity	Successfully delete donation activity	Pass
TC-05-08	Edit donation activity	Click "Edit" button and update the information	Successfully update donation activity	Successfully update donation activity	Pass

Table 4.5 Activity Module Testing

Module	Manage Blood Stock				
Objective	To allow hospital and blood bank to update and view their own blood stock status. Blood donors and donation organisers are only allowed to view the blood stock status of each hospital and blood bank by searching the name.				
Test ID	**Test Case**	**Input**	**Expected Result**	**Actual Result**	**Test Status**
TC-06-01	Insert with blank input	Blank Input	Display error message	Display error message	Pass
TC-06-02	Update blood stock status	Click the check box of the blood stock status and click "Update" button	Successfully update blood stock status	Successfully update blood stock status	Pass
TC-06-03	Display blood stock status	Insert hospital or blood bank name in	Successfully display blood	Successfully display blood	Pass

		the search bar and click “Search”	stock status	stock status	

Table 4.6 Blood Stock Module Testing

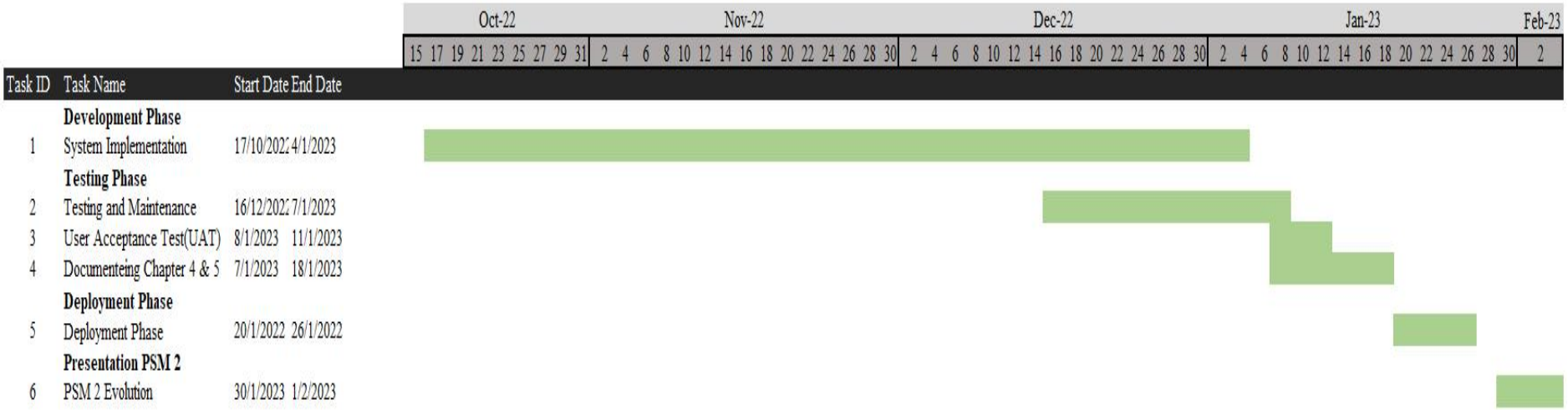


Figure 4.46 Gantt Chart PSM2

Chapter 5

Conclusion

5.1 Introduction

There are a total of five chapters in this thesis. The project's concept, goal, and issue description were covered in Chapter 1 along with the proposed application's scope. The application that exists was covered in Chapter 2. The three current applications and the suggested application are compared. Diagrams were used in Chapter 3 to describe the methodology, application structure, and architecture. The storyboard will be displayed to demonstrate the application's flow. After the application is constructed, the outputs and outcomes were described in Chapter 4. The limits of the application and potential future work to enhance the application will be covered in Chapter 5.

5.2 Objective Revisited

The goal of this study is to come up with a blood bank application which aims to find compatible blood donors and call on the public to actively participate in the blood donation drive. This application's goal has been achieved.

1. To determine the existing activities on blood bank records.
2. To develop mobile blood bank application records in order to heighten accessibility of different blood types toward the supply in blood banks and hospitals.
3. To validate the mobile blood bank application towards the increasing number of new blood donors.

   The first objective is achieved where the blood stock of the hospital and blood bank is displayed. The second objective is also achieved where the hospital and blood bank able to create an emergency donation post and it is able to view by any donor. The last objective is achieved as this Malaysia Blood Bank Android Mobile Application is built at the end of the project.

5.3 Limitations

The Blood Bank App's drawbacks include the fact that it can only be installed on Android devices. On top of that, Blood Bank Application are only applicable to mobile but not web service. The other limitation of this application is that it only provides an English version. Since English is the official language to most countries, this application is developed with English. Only English Learners are able to use this application.

5.4 Future Work

There are various improvements that can be made to the Blood Bank Application in the future.

1. Adding a pop-up notification when blood stock status of any hospital and blood bank is low.
2. Develop Blood Bank Application in multi-language so that not only English Learners able to use it.
3. Adding a live chat between donor and hospital, blood bank and organizer.

References

Agile Development – Advantages, Disadvantages and when to use it? - The Official 360logica Blog. (n.d.). Retrieved June 4, 2022, from https://www.360logica.com/blog/agile-development-advantages-disadvantages-and-when-to-use-it/

Medical Definition of Blood bank. (n.d.). Retrieved June 5, 2022, from https://www.medicinenet.com/blood_bank/definition.htm

Mei Ling, L., Sheng Hui, T., G Tan, A. K., & Sai Ling, G. (2018). *DETERMINANTS OF BLOOD DONATION STATUS IN MALAYSIA: PROFILING THE NON-DONORS, OCCASIONAL DONORS AND REGULAR DONORS*. *36*(1), 43–62. https://doi.org/10.21315/km2018.36.1.3

National blood centre calls on public to donate blood. (n.d.). Retrieved June 4, 2022, from https://www.malaysiakini.com/news/430415

Nonfunctional Requirement Examples - Requirements Quest. (n.d.). Retrieved June 5, 2022, from https://requirementsquest.com/nonfunctional-requirement-examples/#availability

Software Development Life Cycle: Phases and Models. (n.d.). Retrieved June 5, 2022, from https://ncube.com/blog/software-development-life-cycle-guide

Software Development Life Cycle: The phases of SDLC - TestLodge Blog. (n.d.). Retrieved June 5, 2022, from https://blog.testlodge.com/software-development-life-cycle/

Storage of blood and blood products | Homerton University Hospital Foundation Trust. (n.d.). Retrieved June 5, 2022, from https://www.homerton.nhs.uk/storage-of-blood-and-blood-products/

What Are Non-Functional Requirements? Types and Examples - WINaTALENT | Blog. (n.d.). Retrieved June 5, 2022, from https://winatalent.com/blog/2020/05/what-are-non-functional-requirements-types-and-examples/

What Is Agile Methodology? - Overview Of Agile Software Development And Agile Models. (n.d.). Retrieved June 5, 2022, from https://www.digite.com/agile/agile-methodology/

APPENDIX A
SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2022

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

Malaysia Blood Bank Android Mobile Application (MBBA)

DOCUMENT APPROVAL

	Name	Date
Authenticated by:  _____ Developer	LIM YI CI	26/5/2022
Approved by: _____ Supervisor	DR. NABILAH FILZAH BINTI MOHD RADZUAN	

Software : Draw IO, Microsoft Word

Archiving Place : Google Drive

TABLE OF CONTENT

CONTENT **PAGE**

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

The Malaysia Blood Bank Application is a mobile application that helps Malaysian citizens to have a fast and easy way to donate and get the information about blood donation from legal organisations. Furthermore, the blood stock status from each hospital and blood bank is open for the Blood Bank Application user to have a rapid supplement of blood. Malaysia Blood Bank Application consists of 6 modules which are manage sign up, manage activity, manage blood stock, manage appointment, manage emergency and manage profile.

First, manage sign up is required for all types of users to login, register an account and reset password by inserting the user's email address.

In the manage activity module, hospital, donation organiser and blood bank are able to create, update, view and delete a donation activity. Blood donors are only allowed to view the upcoming donation activities.

The next module is manage blood stock. This module allows hospital and donation organiser to update, view and delete their own blood stock status. While blood donors and donation organisers are only allowed to view the blood stock status of each hospital and blood bank by searching the name.

Besides, the manage appointment module allows blood donors to view, book and delete the donation appointment. Hospital and blood banks are only allowed to view and delete the appointment that was made by the blood donor.

On top of that, manage emergency module allows hospitals and blood banks to create, update, delete and view the emergency donation. However, blood donors are only allowed to view the emergency donation.

The final module is manage profile. This module allows blood donors, hospital, donation organiser and blood bank to view and update their own profile.

1.2 SYSTEM IDENTIFICATION

This document uses the following convention:

System Identification Number: SRS-MBBA-V01-22

SRS	Software Requirements Specification
MBBA	Malaysia Blood Bank Application

V01	Version 1
22	Year 2022

Table 1.1 System Identification

1.3 CONTEXT DIAGRAM

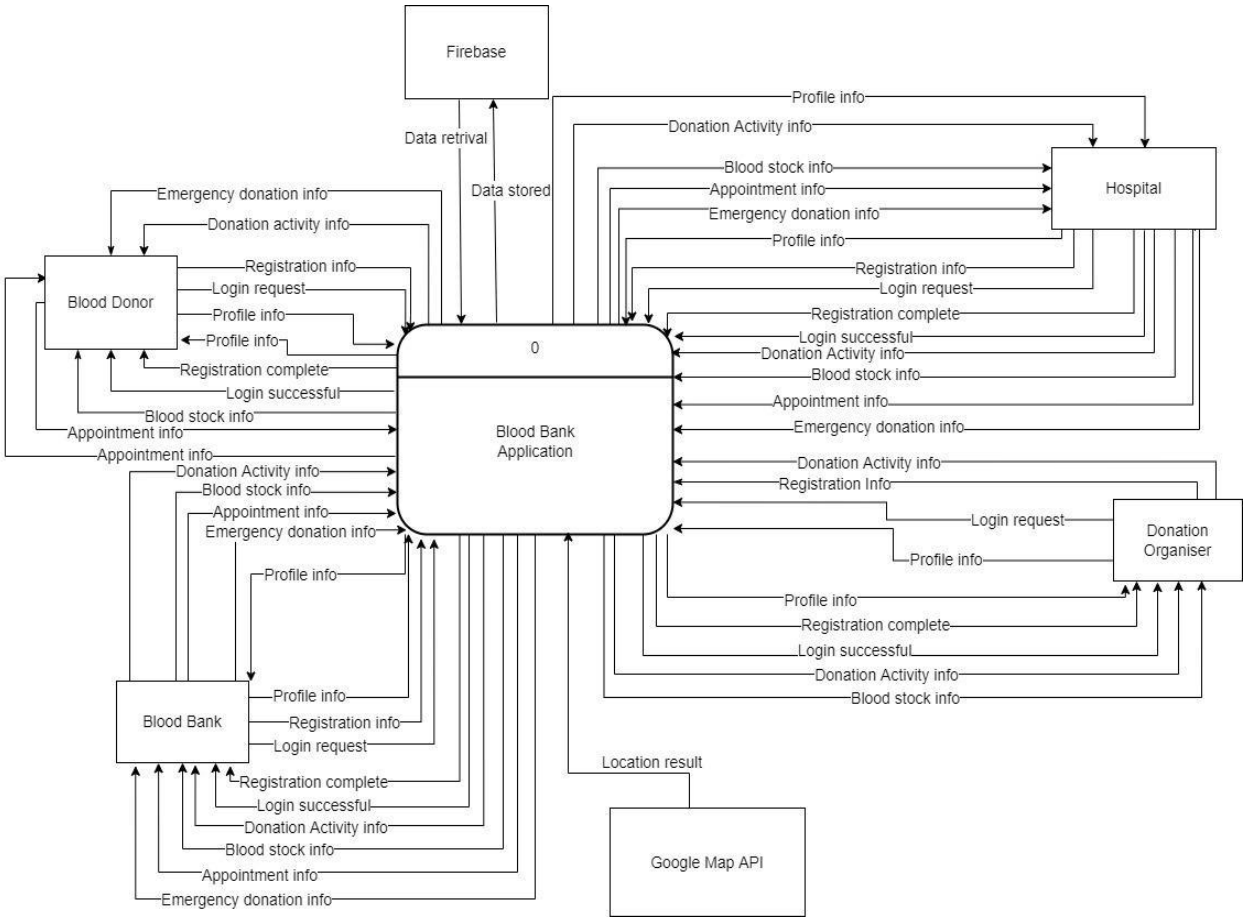


Figure 1.1 Context Diagram of Blood Bank Application

Figure 1.1 shows the context diagram of Blood Bank Application. The software system's boundaries are defined and clarified by the context diagram. It determines the information flows between the system and the external entities.

There are 6 external entities for Blood Bank Application which are Firebase, blood donor, hospital, blood bank, donation organiser and Google Map API. Firebase acts as a database to store and retrieve data to the system. Google Map API provides the location result to the system according to the address. All users are required to register an account for the first time

then login to the system.

Blood donors are able to view the blood stock information, emergency donation information and the donation activity information. They are also able to create, update and delete the appointment information and view and update their profile information.

For hospital and blood bank users, create, update, delete and view can be done for donation activity information, emergency donation information. They are able to update and view their blood stock status and view the appointment information. Besides, they can also update and view their profile information.

For donation organisers, they are only able to view the blood stock status and view and update the profile information. They can delete, update, view and create donation activity information.

1.4 DATA FLOW DIAGRAM

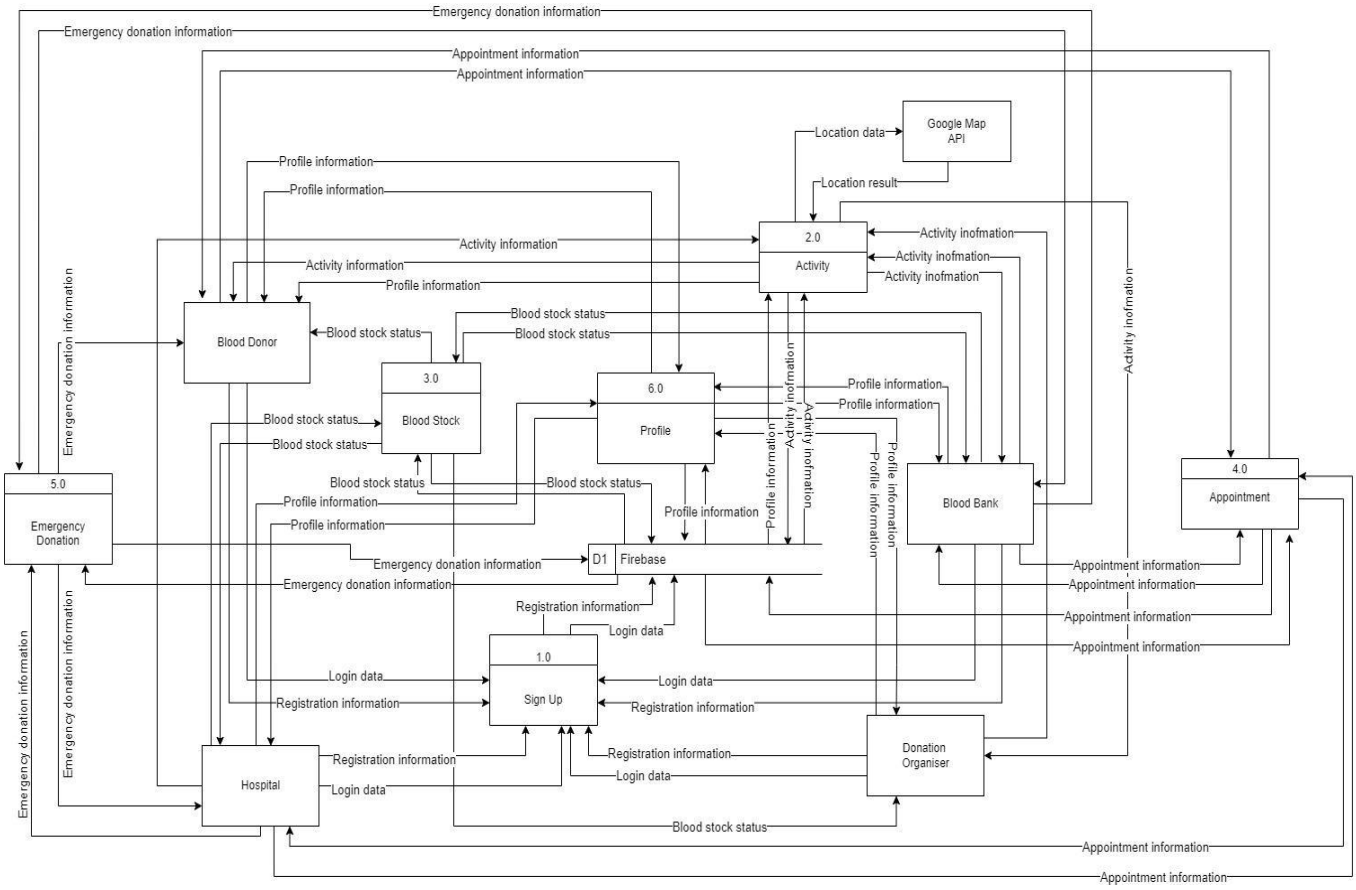


Figure 1.2 Level 1 Data Flow Diagram of Blood Bank Application

Figure 1.2 shows the Level 1 Data Flow Diagram of Blood Bank Application. There are 6 processes in this data flow diagram, namely “Signup”, “Activity”, “Blood Stock”, “Appointment”, “Emergency Donation” and “Profile”. The flow starts when users sign up in the application by entering the registration information. The registration information will be sent to the firebase database. Once the user has registered an account, the user can login to the application. Users are able to choose for the process “Activity”, “Blood Stock”, “Appointment”, “Emergency Donation” and “Profile” according to the user type. The second process is “Activity”. Blood banks, hospitals and donation organisers are able to update, view, delete and create the activity information while blood donors are only able to view activity information and the profile for blood bank and organiser. The third process is “Blood Stock”. Blood banks and hospitals are able to update and view the blood stock status while blood donors and donation organiser are only able to view blood stock status. The fourth process is “Appointment”. Blood banks and hospitals are able to delete and view the appointment information while blood donors are only able to view appointment information. The fifth process is “Emergency Donation”. Blood banks and hospitals are able to update, view, delete and create the emergency donation information while blood donors are only able to view emergency donation information. The sixth process is “Profile”. All users are able to view and update their profile information. The activity information, profile information, emergency donation information, appointment information and blood stock information are stored and retrieved from the firebase.

CHAPTER 2

2.1 USE CASE DIAGRAM AND DESCRIPTION

Use case diagram is a diagram to summarise the details of the system's users and their interactions with the system. Figure 2.1 depicts the use case diagram, relationship of functionality and the requirements of Malaysia Blood Bank Application system.

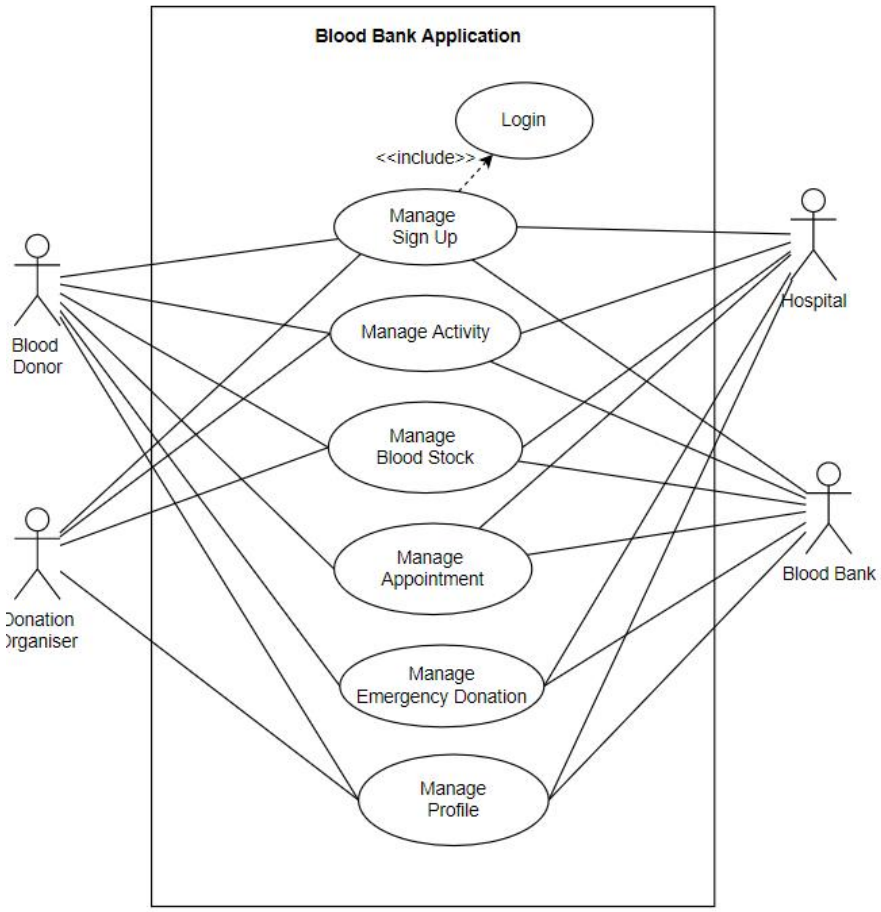


Figure 2.1 Use Case Diagram of MBBA

Module	Requirements	User
Manage Sign Up	The system allows blood donors, hospitals, donation organisers and blood banks to login, register an account and reset password by inserting the user's email address.	Blood donor, Hospital, Donation organiser, Blood bank

Manage Activity	The system allows hospital, donation organiser and blood bank to create, update, view and delete a donation activity. Blood donors are only allowed to view the upcoming donation activities.	Blood donor, Hospital, Donation organiser, Blood bank
Manage Blood Stock	The system allows hospital and donation organiser to update and view their own blood stock status. Blood donors and donation organisers are only allowed to view the blood stock status of each hospital and blood bank by searching the name.	Blood donor, Hospital, Donation organiser, Blood bank
Manage Appointment	The system allows blood donors to view, book, update and delete the donation appointment. Hospitals and blood banks are only allowed to view the appointment that was made by blood donor.	Blood donor, Hospital, Blood bank
Manage Emergency Donation	The system allows hospitals and blood banks to create, update, delete and view the emergency donation. Blood donors are only allowed to view the emergency donation.	Blood donor, Hospital, Blood bank
Manage Profile	The system allows blood donors, hospital, donation organiser and blood bank to view and update their own profile.	Blood donor, Hospital, Donation organiser, Blood bank

Table 2.1 Use Case Description of MBBA

2.1.1 Manage Sign Up

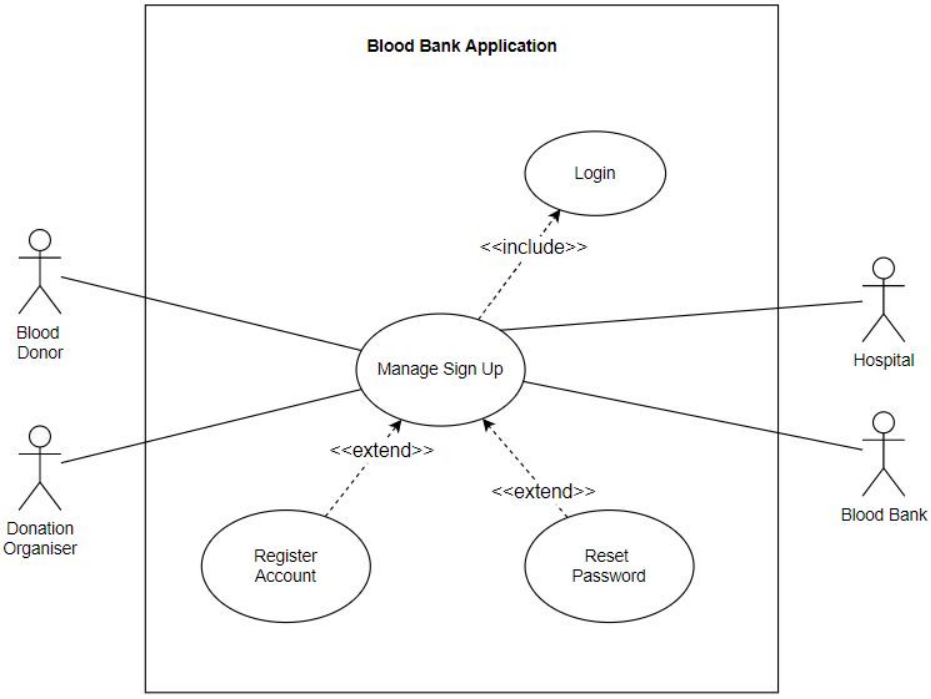


Figure 2.2 Manage Sign Up Use Case Diagram

Use Case ID	SRS-MBBA-UC-01
Brief Description	This use case allows blood donors, hospitals, donation organisers and blood banks to login, register an account and reset password by inserting the user's email address.
Actor	Blood Donor, Hospital, Donation Organiser and Blood Bank
Pre-Conditions	1. User is connected to the Internet.
Basic Flow	1. The use case begins with the user starts the application. 2. User insert email and password. **[A1: Reset password] [A2: Register account]** 3. User clicks on the <<LOGIN>> button. 4. System checks email and password in the database. 5. System redirect to home page. **[E1: Invalid email or password]** 6. The use case ends.

Alternative Flow	**A1: Reset password** 1. User clicks on the <<RESET PASSWORD>> button. 2. System redirect to "Check Email" page. 3. User insert email and click on the <<CHECK EMAIL>> button. 4. System send a reset password email to the email address. 5. User reset the password from the email. 6. The use case continues to step number 2 in basic flow. **A2: Register account** 1. User clicks on the <<REGISTER ACCOUNT>> button. 2. System redirect to "Register Account" page. 3. User fill up the information and click the <<REGISTER NOW>> button. 4. System saves the information to the database. 5. System redirect to the "Login" page. 6. The use case continues to step number 2 in basic flow.
Exceptional Flow	**E1: Invalid email or password** 1. System display "Invalid email or password". 3. The use case continues to step number 2 in basic flow.
Post-Conditions	Users are able to redirect to the home page when login is successful.
Rules	-
Constraints	-

Table 2.2 Manage Sign Up Use Case Description

2.1.2 Manage Activity

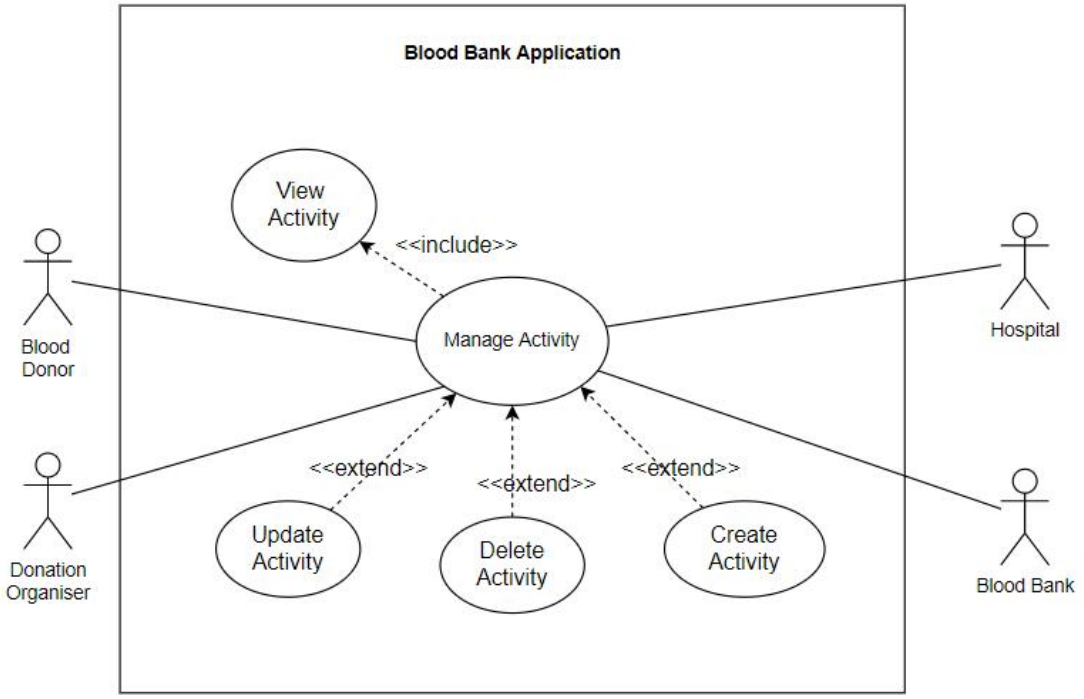


Figure 2.3 Manage Activity Use Case Diagram

Use Case ID	SRS-MBBA-UC-02
Brief Description	This use case allows the hospital, donation organiser and blood bank to create, update, view and delete a donation activity. Blood donors are only allowed to view the upcoming donation activities.
Actor	Blood Donor, Hospital, Donation Organiser and Blood Bank
Pre-Conditions	1. User is connected to the Internet. 2. User is successfully login to the system.
Basic Flow	**Blood Donor** 1. The use case begins with the user going to the upcoming activity page. 2. System retrieves data from the database. 3. System displays upcoming donation information. **[A1: View others profile][A2: View venue map]** 4. The use case ends.

	Hospital, Blood Bank, Donation Organiser 1. The use case begins with the user going to the upcoming activity page. 2. System retrieves data from the database. 3. System displays upcoming donation information. **[A2: View venue map] [A3: Delete activity] [A4: Update activity] [A5: Create activity]** 4. The use case ends.
Alternative Flow	**A1: View others profile** 1. User clicks on others username. 2. System retrieves data from the database. 3. System displays others' profile information. 4. The use case continues to step number 4 in basic flow. **A2: View venue map** 1. User clicks on the view venue map button. 2. System redirect to map activity interface. 3. User click on the search bar to start the searching location. 4. System displays the location on the map. 5. The use case continues to step number 4 in basic flow. **A3: Delete activity** 1. User clicks on the kebab menu. 2. User select <<DELETE>> button. 3. System delete the activity from the database. 4. The use case continues to step number 2 in basic flow. **A4: Update activity** 1. User clicks on the kebab menu. 2. User select <<EDIT>> button.

	3. System redirect to “Update Donation Activity” page. 4. User update activity information and click the <<UPDATE>> button. 5. System saves the updated activity information to the database. 6. The use case continues to step number 2 in basic flow. **A5: Create activity** 1. User clicks on the <<CREATE DONATION>> button. 2. System redirect to “Create Donation Activity” page. 3. User inserts the activity information and clicks <<CREATE>> button. 4. System saves the new activity to the database. 5. The use case continues to step number 2 in basic flow.
Exceptional Flow	-
Post-Conditions	All the upcoming activities information is displayed.
Rules	-
Constraints	-

Table 2.3 Manage Activity Use Case Description

2.1.3 Manage Blood Stock

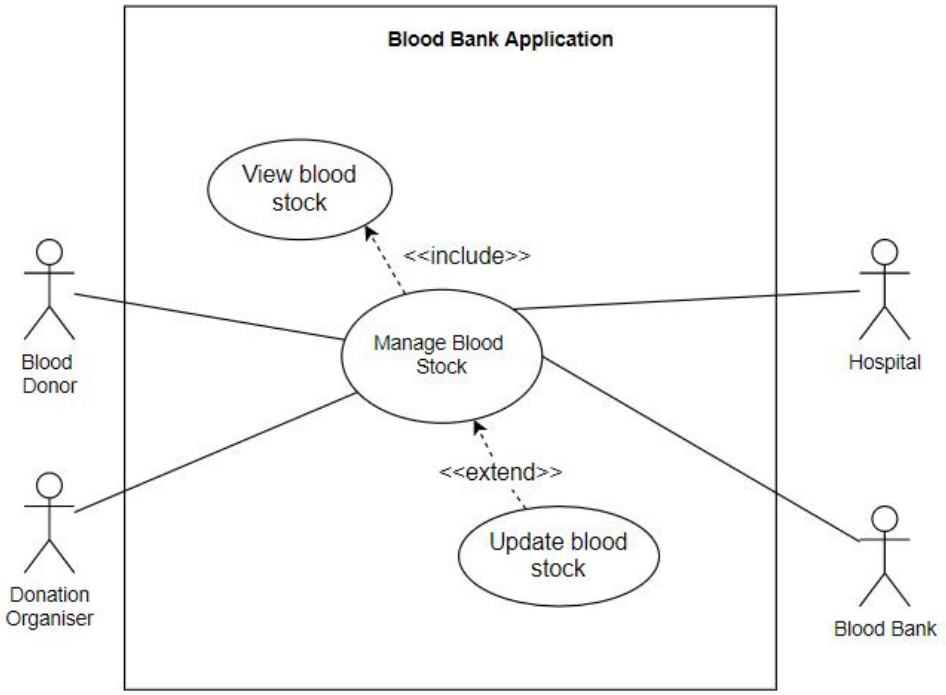


Figure 2.4 Manage Blood Stock Use Case Diagram

Use Case ID	SRS-MBBA-UC-03
Brief Description	This use case allows hospital and donation organiser to update and view their own blood stock status. Blood donors and donation organisers are only allowed to view the blood stock status of each hospital and blood bank by searching the name.
Actor	Blood Donor, Hospital, Donation Organiser and Blood Bank
Pre-Conditions	1. User is connected to the Internet. 2. User is successfully login to the system.
Basic Flow	**Blood Donor, Donation Organiser** 1. The use case begins with the user going to the home page. 2. System retrieves data from the database. 3. System displays the blood stock status. **[A1: Search hospital name]** 4. The use case ends.

	Hospital, Blood Bank 1. The use case begins with the user going to the home page. 2. System retrieves data from the database. 3. System displays the blood stock status. **[A2: Update blood stock]**
Alternative Flow	**A1: Search hospital name** 1. User clicks on the search bar and searches for the hospital name. 2. The use case continues to step number 2 in basic flow. **A2: Update blood stock** 1. User clicks on the <<UPDATE>> button. 2. System redirect to "Update Blood Stock" page. 3. User updates the blood stock status and clicks the <<UPDATE>> button. 4. System saves the updated blood stock status to the database. 5. The use case continues to step number 2 in basic flow.
Exceptional Flow	-
Post-Conditions	The blood stock status is displayed on the home page.
Rules	-
Constraints	-

Table 2.4 Manage Blood Stock Use Case Description

2.1.4 Manage Appointment

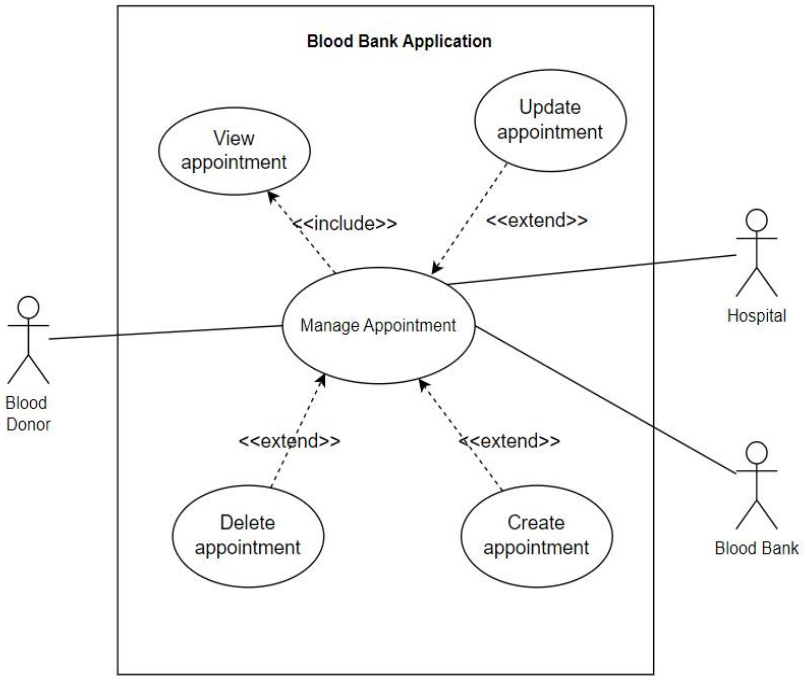


Figure 2.5 Manage Appointment Use Case Diagram

Use Case ID	SRS-MBBA-UC-04
Brief Description	This use case allows blood donors to view, book,update and delete the donation appointment. Hospitals and blood banks are only allowed to view the appointment that was made by blood donor.
Actor	Blood Donor, Hospital and Blood Bank
Pre-Conditions	1. User is connected to the Internet. 2. User is successfully login to the system.
Basic Flow	**Blood Donor** 1. The use case begins with the user going to the View Appointment page. 2. System retrieves data from the database. 3. System displays the booked appointment. **[A1: Make appointment] [A2: Delete appointment] [A3: Update appointment]**

	4. The use case ends. **Hospital, Blood Bank** 1. The use case begins with the user going to the View Appointment page. 2. System retrieves data from the database. 3. System displays the booked appointment. 4. The use case ends.
Alternative Flow	**A1: Make appointment** 1. User clicks on the <<MAKE APPOINTMENT>> button. 2. System redirect to "Book Appointment" page. 3. User inserts the book appointment information and clicks the <<BOOK>> button. 4. System saves the new booked appointment to the database. 5. The use case continues to step number 2 in basic flow. **A2: Delete appointment** 1. User clicks on the kebab menu. 2. User select <<DELETE>> button. 3. System delete the appointment from the database. 4. The use case continues to step number 2 in basic flow. **A3: Update appointment** 1. User clicks on the kebab menu. 2. User select <<EDIT>> button. 3. System redirect to "Update Appointment" page. 4. User update activity information and click the <<UPDATE>> button. 5. System saves the updated activity information to the database. 6. The use case continues to step number 2 in basic flow.

Exceptional Flow	-
Post-Conditions	The booked appointment is displayed on the View Appointment page.
Rules	-
Constraints	-

Table 2.5 Manage Appointment Use Case Description

2.1.5 Manage Emergency Donation

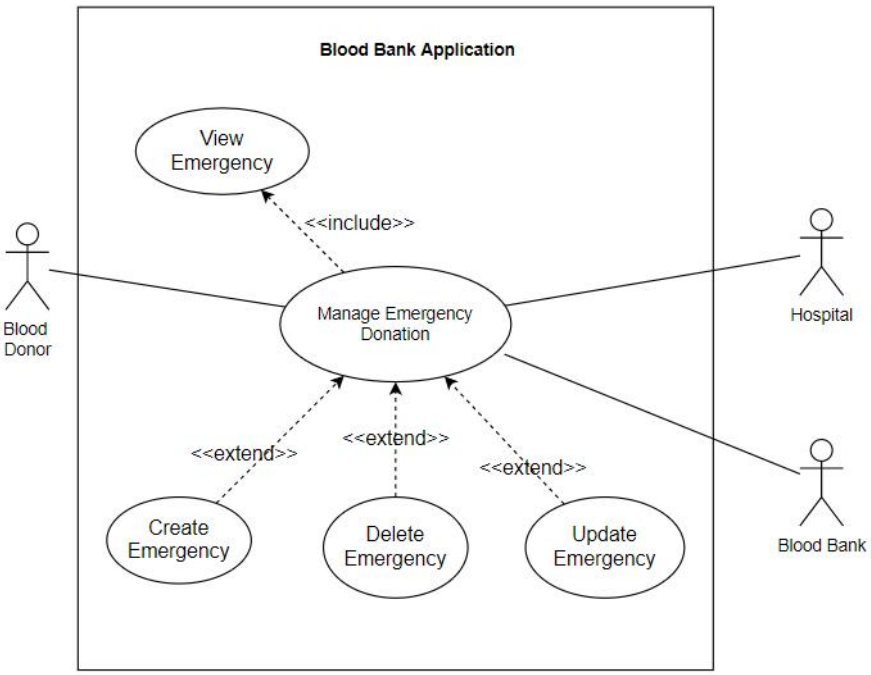


Figure 2.6 Manage Emergency Donation Use Case Diagram

Use Case ID	SRS-MBBA-UC-05
Brief Description	This use case allows hospitals and blood banks to create, update, delete and view the emergency donation. Blood donors are only allowed to view the emergency donation.
Actor	Blood Donor, Hospital and Blood Bank

Pre-Conditions	1. User is connected to the Internet. 2. User is successfully login to the system.
Basic Flow	**Blood Donor** 1. The use case begins with the user going to the View Emergency Donation page. 2. System retrieves data from the database. 3. System displays the emergency donation. 4. The use case ends. **Hospital, Blood Bank** 1. The use case begins with the user going to the View Emergency Donation page. 2. System retrieves data from the database. 3. System displays the emergency donation. **[A1: Delete emergency][A2: Edit emergency][A3: Create emergency]** 4. The use case ends.
Alternative Flow	**A1: Delete emergency** 1. User clicks on the kebab menu. 2. User selects the <<DELETE>> button. 3. System delete the emergency donation from the database. 4. The use case continues to step number 2 in basic flow. **A2: Edit emergency** 1. User clicks on the kebab menu. 2. User selects the <<EDIT>> button. 3. System redirect to “Update Emergency Donation” page. 4. User updates the emergency donation information and clicks the <<UPDATE>> button. 4. System saves the updated emergency donation information. 5. The use case continues to step number 2 in basic flow.

	A3: Create emergency 1. User clicks the <<CREATE EMERGENCY>> button. 2. System redirect to the "Create Emergency Donation" page. 3. User inserts the emergency donation information and clicks the <<CREATE>> button. 4. System saves the new emergency donation to the database. 5. The use case continues to step number 2 in basic flow.
Exceptional Flow	-
Post-Conditions	The emergency donation information is displayed on the View Emergency page.
Rules	-
Constraints	-

Table 2.6 Manage Emergency Donation Use Case Description

2.1.6 Manage Profile

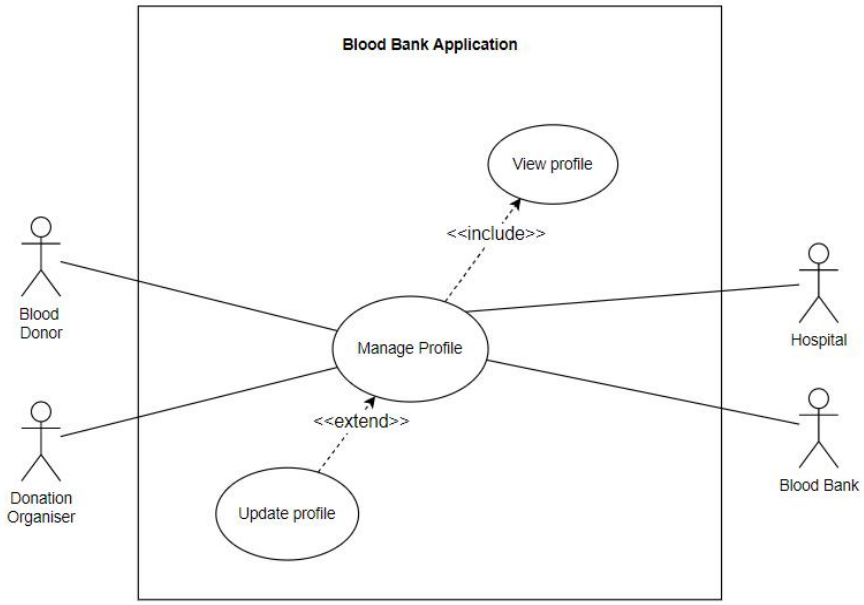


Figure 2.7 Manage Profile Use Case Diagram

Use Case ID	SRS-MBBA-UC-06
Brief Description	This use case allows blood donors, hospital, donation organiser and blood bank to view and update their own profile.
Actor	Blood Donor, Hospital, Donation Organiser and Blood Bank
Pre-Conditions	1. User is connected to the Internet. 2. User is successfully login to the system.
Basic Flow	1. The use case begins with the user clicks the profile button. 2. System retrieves data from the database. 3. System displays the user's profile information. **[A1: Update profile]** 4. The use case ends.
Alternative Flow	**A1: Update profile** 1. User clicks the <<UPDATE>> button. 2. System retrieve to “Update Profile” page. 3. User updates the profile information and clicks the <<DONE>> button. 4. System saves the updated profile information. 5. The use case continues to step number 2 in basic flow.
Exceptional Flow	-
Post-Conditions	The profile information is displayed on the View Profile page.
Rules	-
Constraints	-

Table 2.7 Manage Profile Use Case Description

2.2 SEQUENCE DIAGRAM

2.2.1 Manage Sign Up

Basic Flow

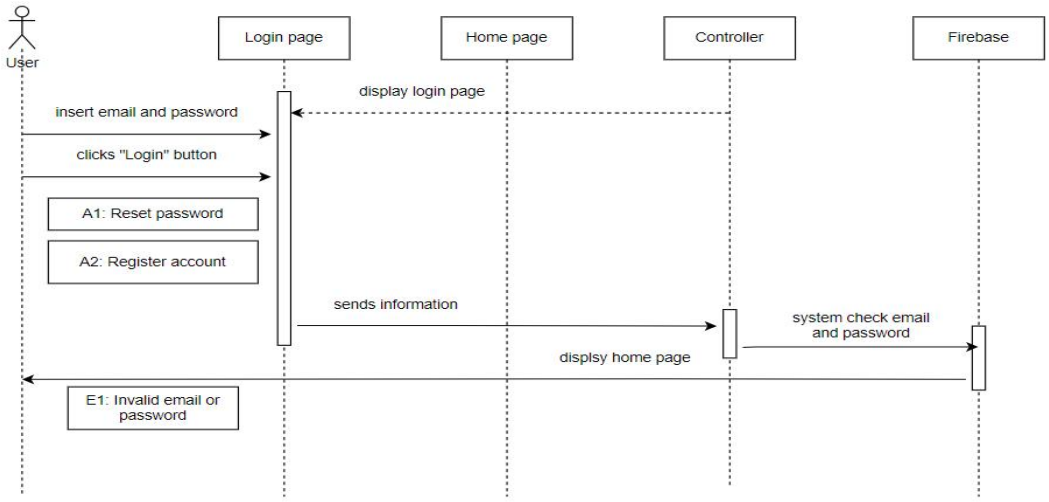


Figure 2.8 Sign Up Basic Flow Sequence Diagram

Based on figure 2.8, the application displays a login page to the user. Then the user inserts email and password and clicks the "Login button". There are 2 alternative flows for users which are reset password and register account. The information is sent to the controller and the system will check email and password. If the email and password is correct, the system displays the home page. An exceptional flow of invalid email or password is shown for the incorrect email or password.

Alternative Flow

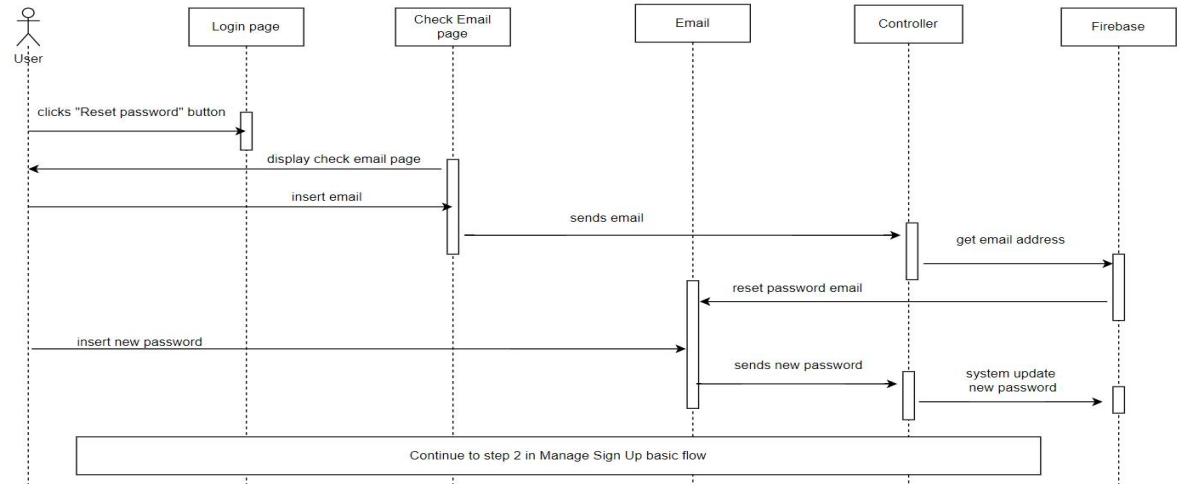


Figure 2.9 Sign Up [Alternative Flow: Reset Password] Sequence Diagram

Figure 2.9 shows the alternative flow of sign up. This happens when a user clicks the “Reset password” button. Then the system displays the check email page and the user needs to insert email. System send a reset password email to the email address. After user had set new password, system update the new password then back to the login page. Users will continue in the step 2 of basic flow.

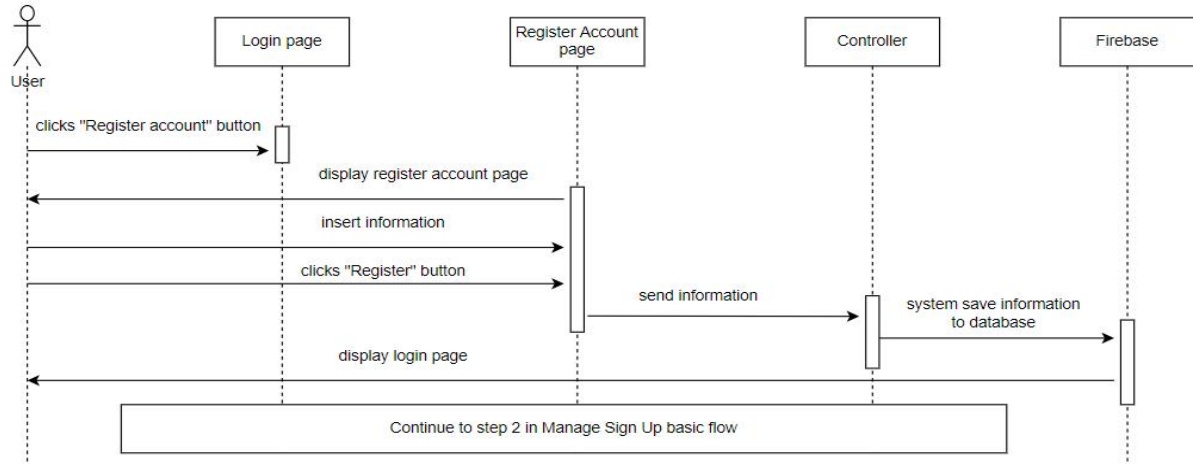


Figure 2.10 Sign Up [Alternative Flow: Register Account] Sequence Diagram

Figure 2.10 shows the alternative flow register account. It starts with the user clicking on “Register account” button. System display register account page and user required to insert information. The information will be saved in the database after clicking the “Register” button. Login page will be redirected. User continues with step 2 in basic flow.

Exceptional Flow

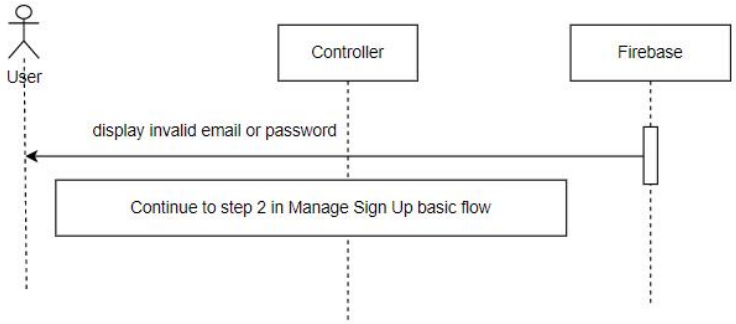


Figure 2.11 Sign Up [Exceptional Flow: Invalid Email or Password] Sequence Diagram

Figure 2.11 shows the exceptional flow invalid email or password. This happens when user inserts the invalid email or password when login. The system will then display invalid email or password. User continues with step 2 in the basic flow.

2.2.2 Manage Activity

Basic Flow

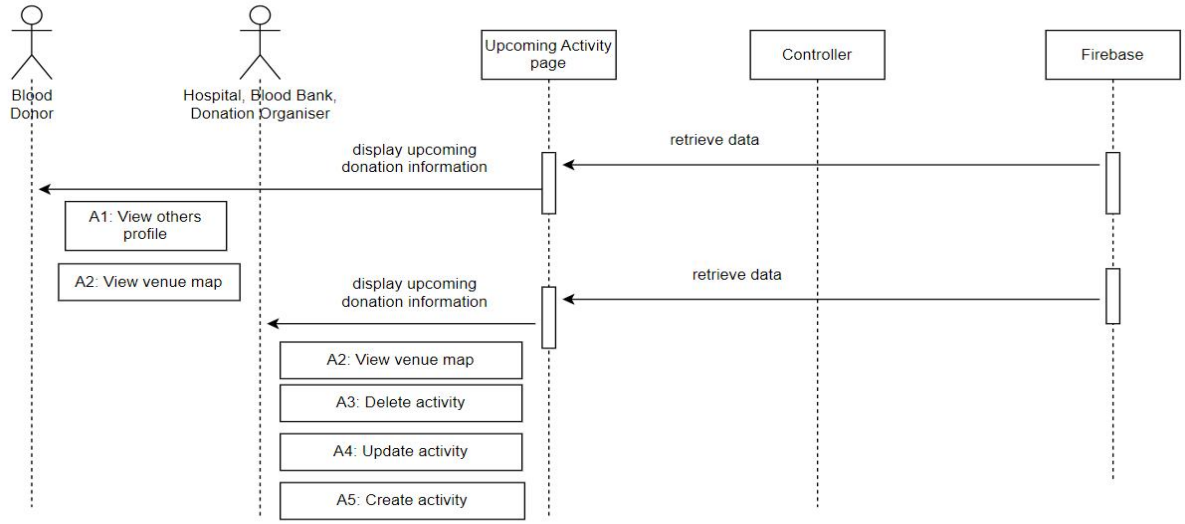


Figure 2.12 Activity Basic Flow Sequence Diagram

Based on figure 2.12, the manage activity starts with the system retrieving data from the database and displaying upcoming donation information to the upcoming activity page. An alternative flow for blood donors is to view others' profiles. For hospital, blood bank and donation organiser users, the alternative flows are delete activity, edit activity and create activity.

Alternative Flow

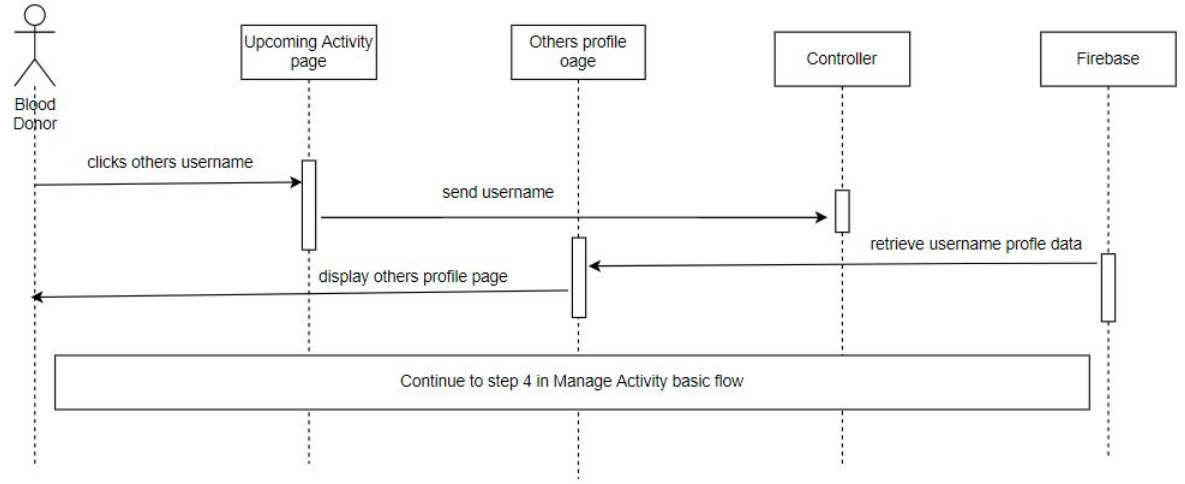


Figure 2.13 Activity [Alternative Flow: View Others Profile] Sequence Diagram

Figure 2.13 shows the view others profile alternative flow. It starts with the user clicking on others username. The system retrieves the username profile data from the database and displays it on the other profile page. The flow will then continue with step 4 in basic flow.

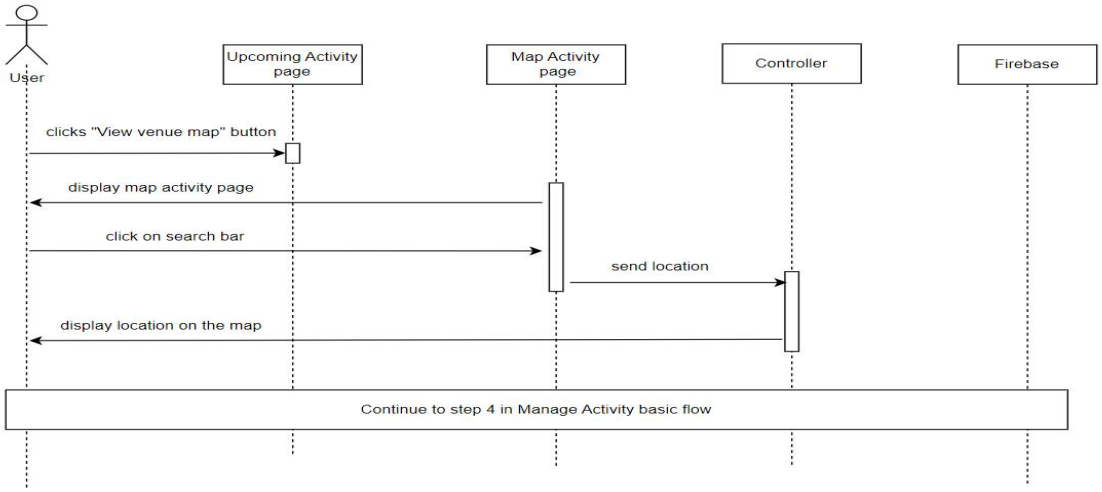


Figure 2.14 Activity [Alternative Flow: View venue map] Sequence Diagram

Figure 2.14 shows the view venue map alternative flow. The flow starts when the user clicks on the “View venue map” button. System then displays the map activity page. Users click on the search bar to start the searching location. The location will be show on the map. User continues the flow with step 4 in basic flow.

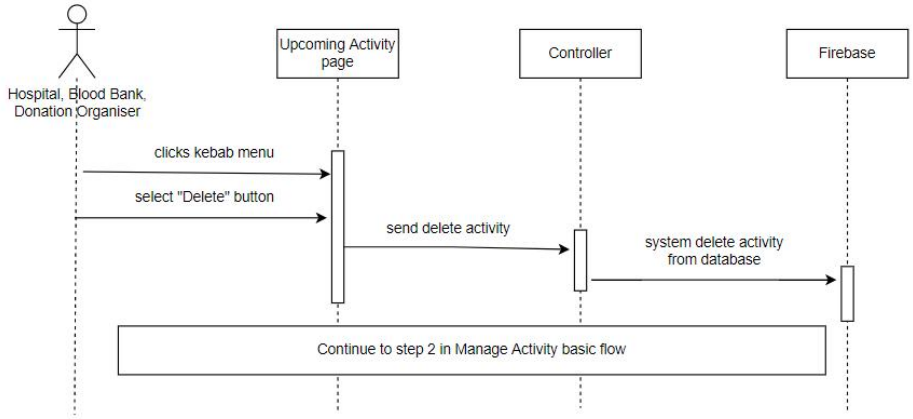


Figure 2.15 Activity [Alternative Flow: Delete Activity] Sequence Diagram

Figure 2.15 shows the delete activity alternative flow. The flow starts when the user clicks on the kebab menu and selects the “Delete” button. System will delete the selected activity from the database. User continues the flow with step 2 in basic flow.

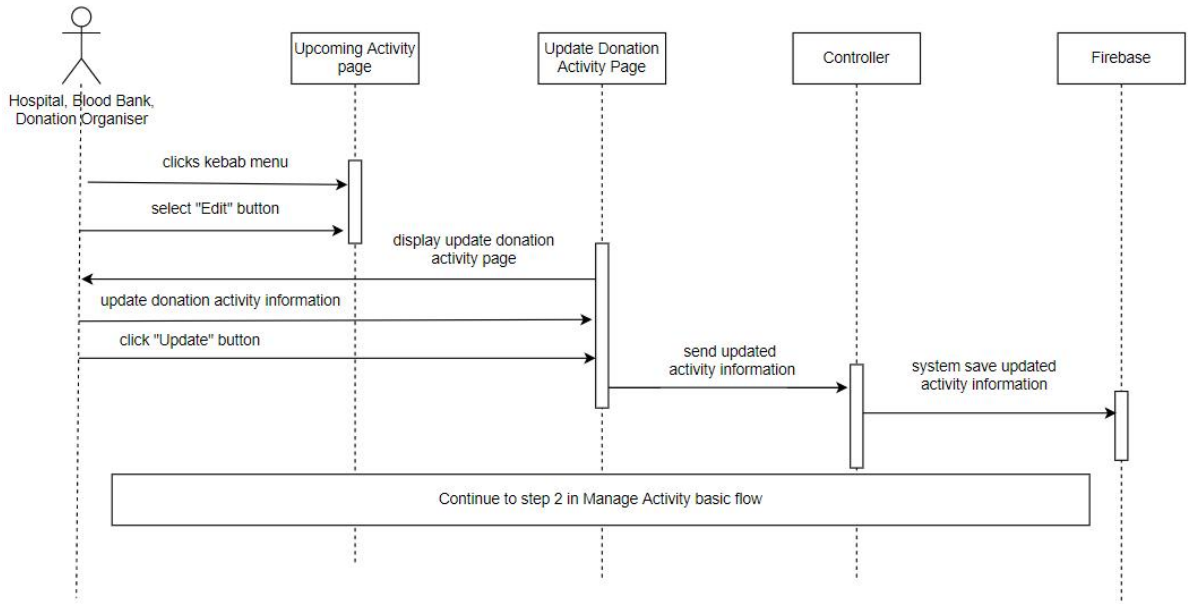


Figure 2.16 Activity [Alternative Flow: Update Activity] Sequence Diagram

Figure 2.16 shows the update activity alternative flow in manage activity module. The flow starts with user clicks on the kebab menu and selecting the "Edit" button. System display update donation activity page for user to update the donation activity information. Then click the "Update" button for the system to save it to the database. User continues the flow with step 2 in basic flow.

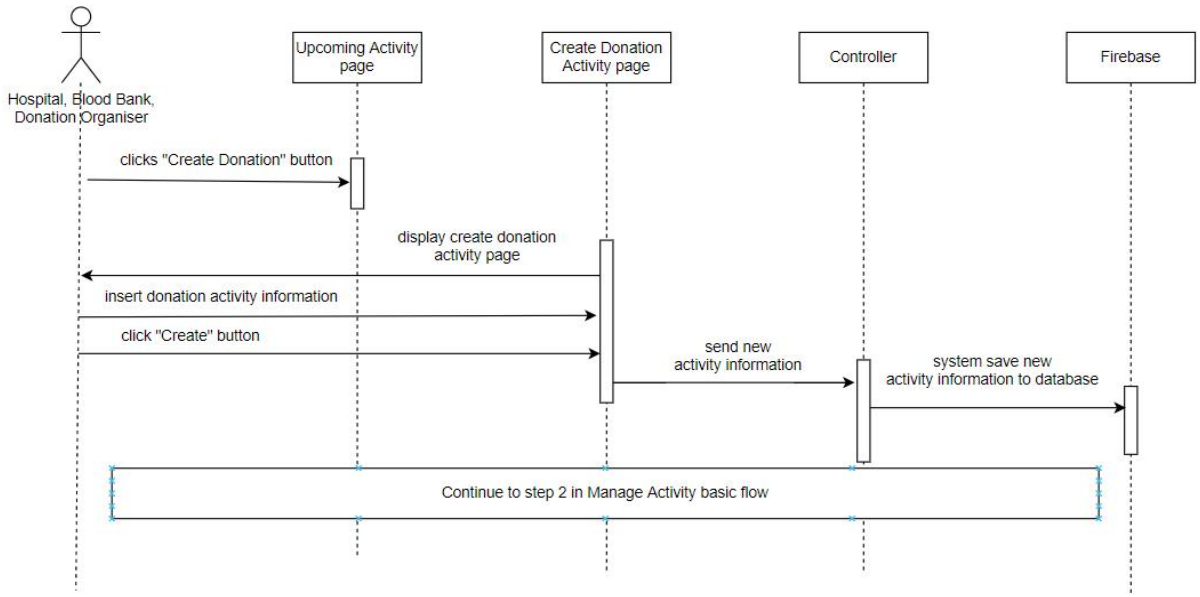


Figure 2.17 Activity [Alternative Flow: Create Activity] Sequence Diagram

Figure 2.17 shows the create activity alternative flow in manage activity module. The flow starts with user clicks on the "Create Donation" button. System display create donation

activity page to insert the donation activity information. Then click the "Create" button for the system to save it to the database. User continues the flow with step 2 in basic flow.

2.2.3 Manage Blood Stock

Basic Flow

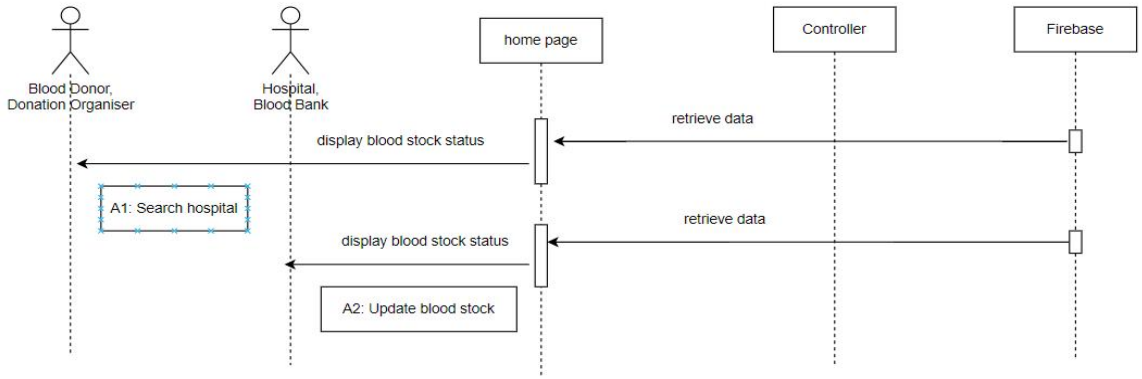


Figure 2.18 Blood Stock Basic Flow Sequence Diagram

Based on the figure 2.18, the system retrieves data from the database and displays the blood stock status on the home page. There is a search hospital alternative flow for blood donors and donation organiser. Update blood stock alternative flow is displayed to hospital and blood banks.

Alternative Flow

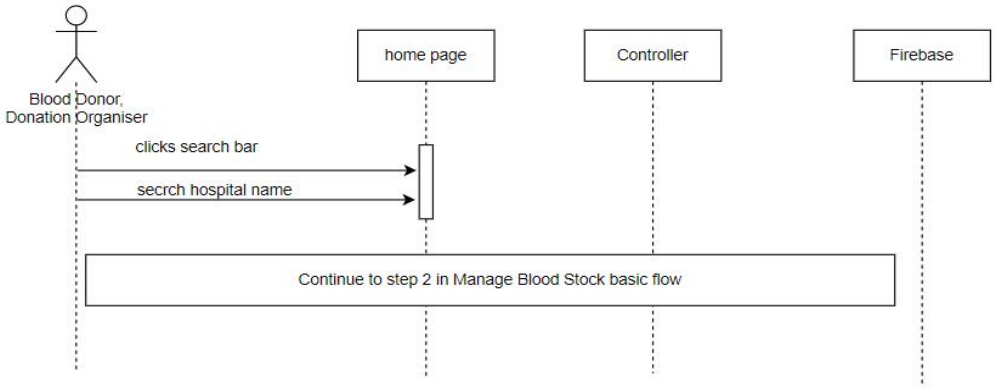


Figure 2.19 Blood Stock [Alternative Flow: Search Hospital] Sequence Diagram

Figure 2.19 shows the search hospital alternative flow where it begins with the user clicking the search bar and searching for the hospital name. The flow will then continue with step 2 in basic flow.

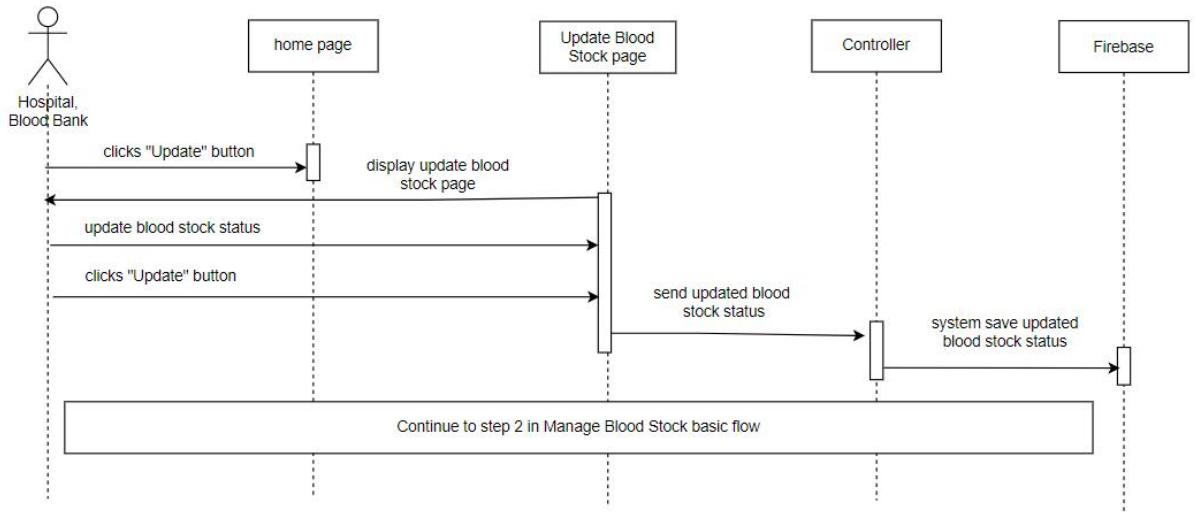


Figure 2.20 Blood Stock [Alternative Flow: Update Blood Stock] Sequence Diagram

Figure 2.20 shows the update blood stock alternative flow in manage blood stock module. The flow starts with user clicks on the "Update" button. System display update blood stock page for user to update the blood stock information. Then click the "Update" button for the system to save it to the database. User continues the flow with step 2 in basic flow.

2.2.4 Manage Appointment

Basic Flow

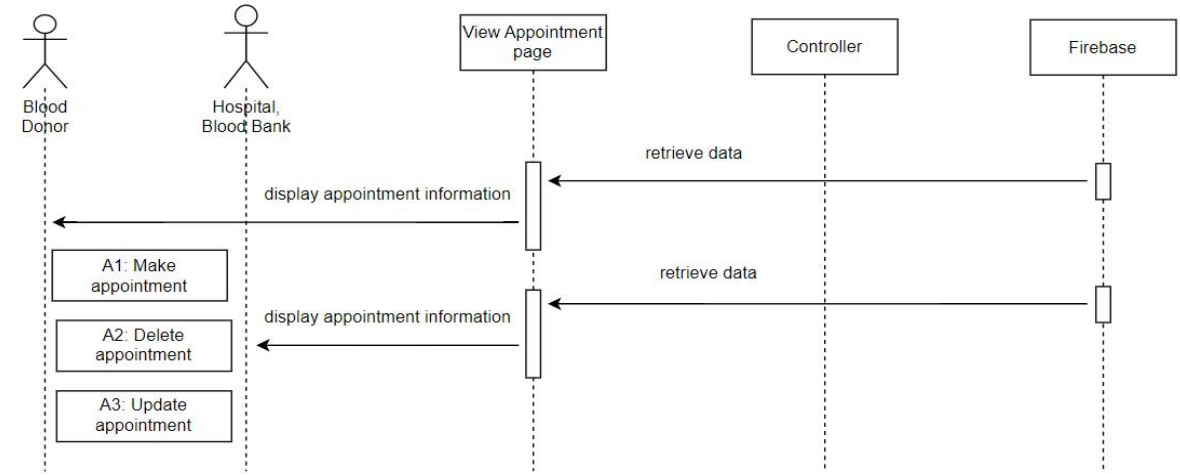


Figure 2.21 Appointment Basic Flow Sequence Diagram

Based on figure 2.21, the manage appointment module starts with the system retrieving data from the database and displaying appointment information to the view appointment page. An alternative flow for blood donors is to make appointment. For hospitals and blood banks, the alternative flow is delete appointment and update appointment.

Alternative Flow

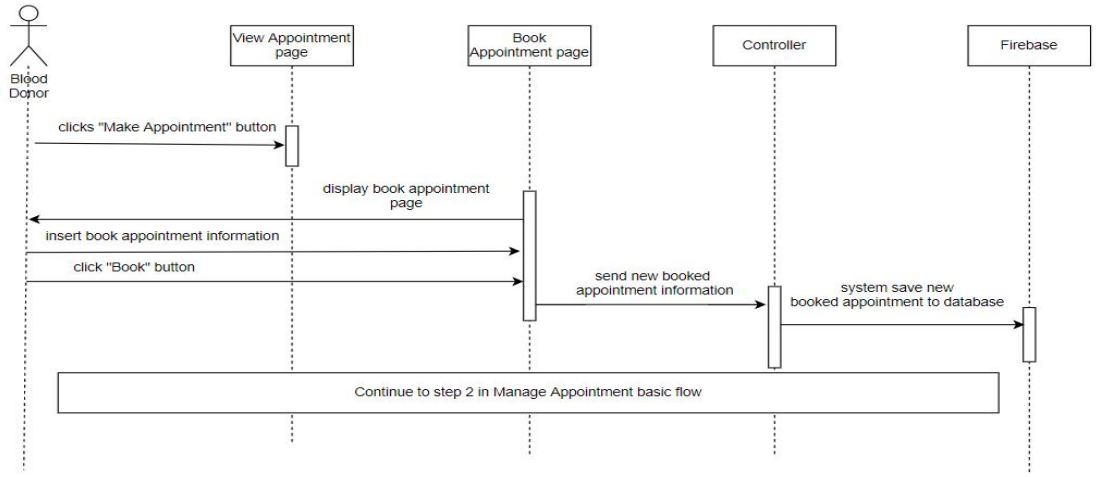


Figure 2.22 Appointment [Alternative Flow: Make Appointment] Sequence Diagram

Figure 2.22 shows the make appointment alternative flow in manage appointment module. The flow begins with user clicks on the “Make Appointment” button. System display book appointment page to insert the book appointment information. Then click the “Book” button for the system to save it to the database. User continues the flow with step 2 in basic flow.

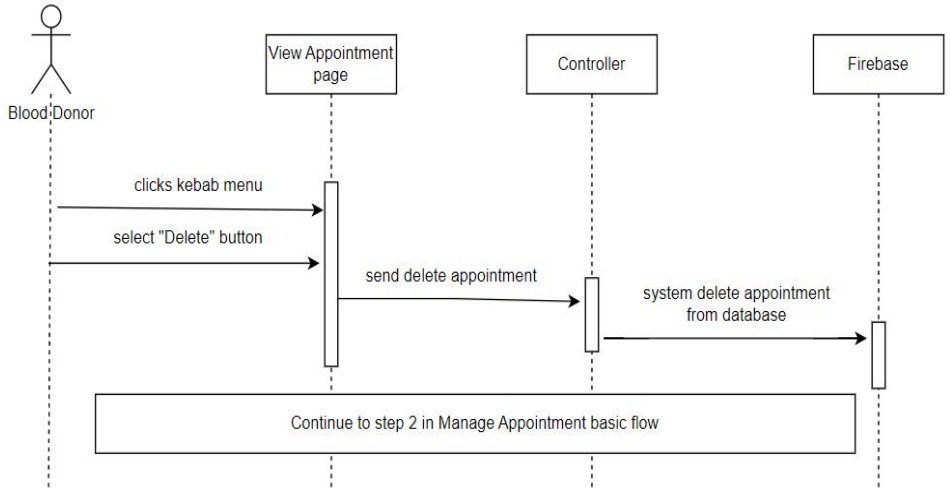


Figure 2.23 Appointment [Alternative Flow: Delete Appointment] Sequence Diagram

Figure 2.23 shows the delete appointment alternative flow. The flow begins when the user clicks on the kebab menu and selects the “Delete” button. System will delete the selected appointment from the database. User continues the flow with step 2 in basic flow.

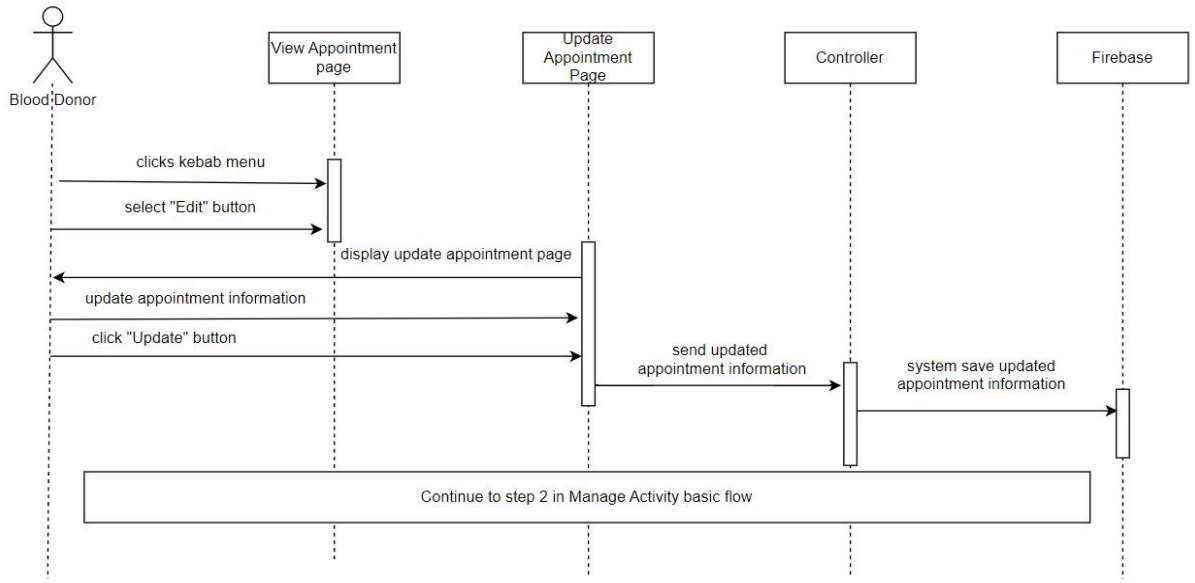


Figure 2.24 Appointment [Alternative Flow: Update Appointment] Sequence Diagram

Figure 2.24 shows the update appointment alternative flow in manage activity module. The flow starts with user clicks on the kebab menu and selecting the "Edit" button. System display update appointment page for user to update the appointment information. Then click the "Update" button for the system to save it to the database. User continues the flow with step 2 in basic flow.

2.2.5 Manage Emergency Donation

Basic Flow

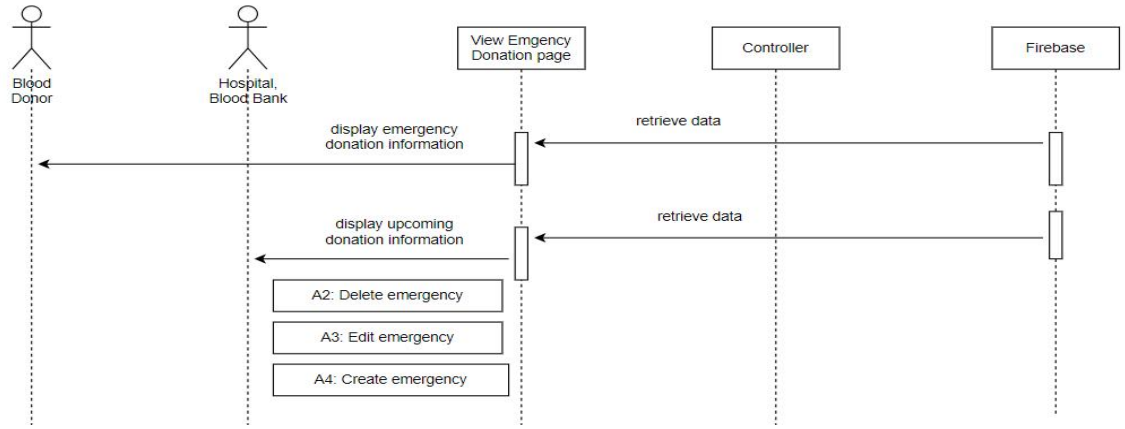


Figure 2.25 Emergency Donation Basic Flow Sequence Diagram

Based on figure 2.25, the manage emergency donation starts with the system retrieving data from the database and displaying emergency donation information to the view emergency donation page. For hospital, blood bank and donation organiser users, there are 3 alternative flows, delete emergency, edit emergency and create emergency.

Alternative Flow

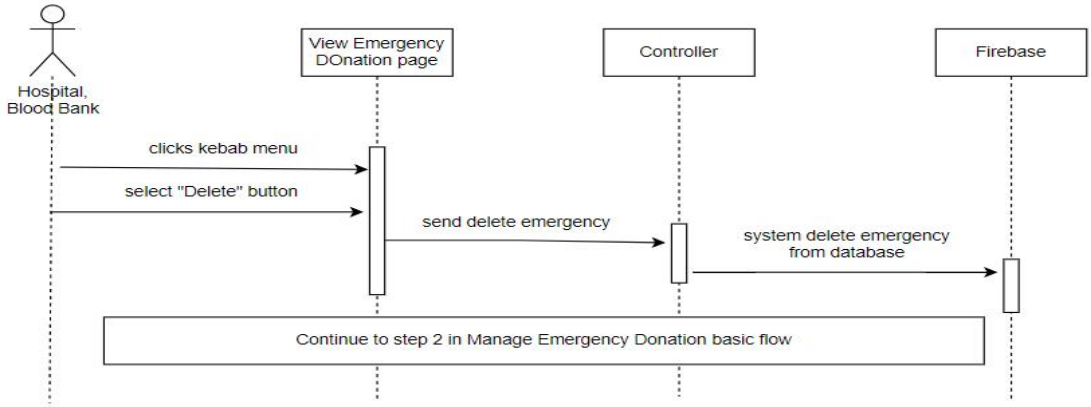


Figure 2.26 Emergency Donation [Alternative Flow: Delete Emergency] Sequence Diagram

Figure 2.26 shows the delete emergency alternative flow. The flow begins when the user clicks on the kebab menu and selects the "Delete" button. System will delete the selected emergency from the database. User continues the flow with step 2 in basic flow.

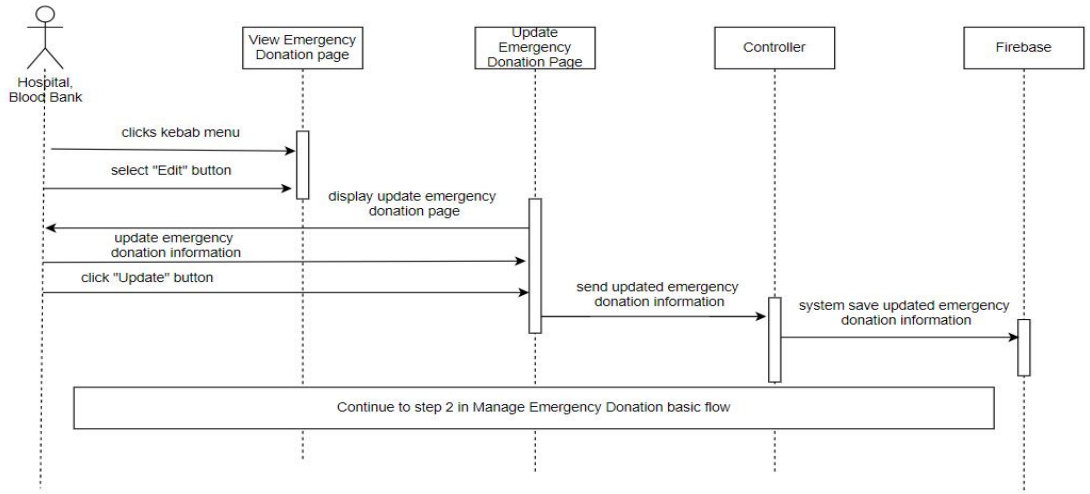


Figure 2.27 Emergency Donation [Alternative Flow: Update Emergency] Sequence Diagram

Figure 2.27 shows the update emergency alternative flow in manage emergency donation module. It begins with user clicks on the kebab menu and selecting the "Edit" button. System display update emergency donation page for user to update the emergency donation information. Then click the "Update" button for the system to save it to the database. User continues the flow with step 2 in basic flow.

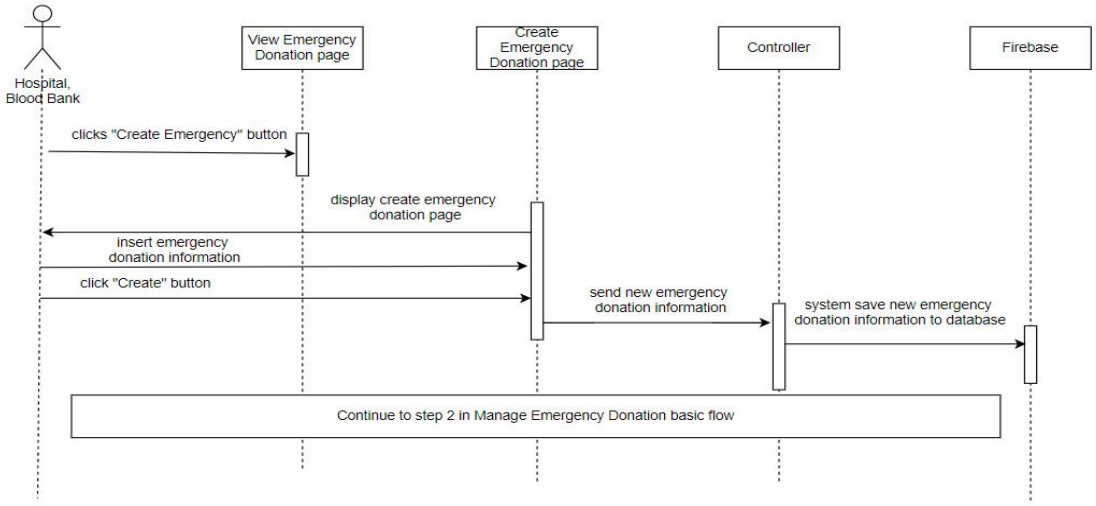


Figure 2.28 Emergency Donation [Alternative Flow: Create Emergency] Sequence Diagram

Figure 2.28 shows the create emergency alternative flow in manage emergency donation module. The flow starts with user clicks on the “Create Emergency” button. System display create emergency donation page to insert the emergency donation information. Then click the “Create” button for the system to save it to the database. User continues the flow with step 2 in basic flow.

2.2.6 Manage Profile

Basic Flow

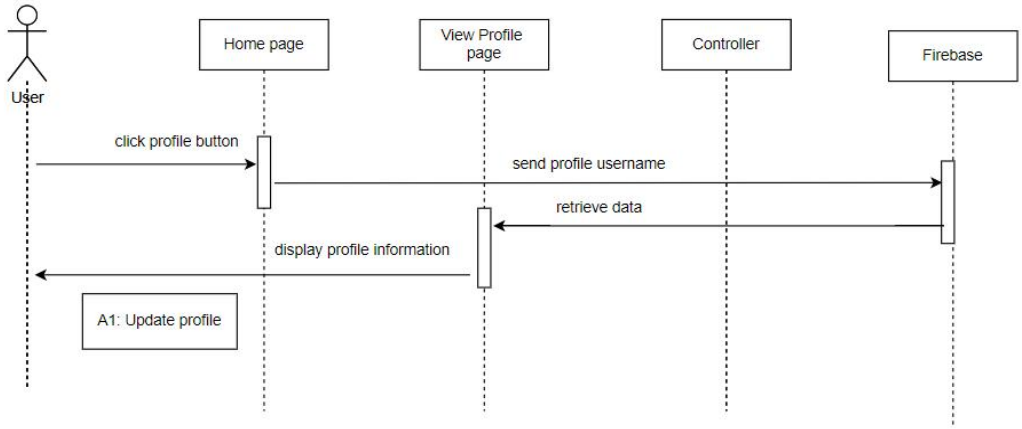


Figure 2.29 Profile Basic Flow Sequence Diagram

Based on figure 2.29, the manage appointment module starts with the user clicks on the profile button. System retrieves data from the database and displays the profile information on the view profile page. There is an update profile alternative flow in this module.

Alternative Flow

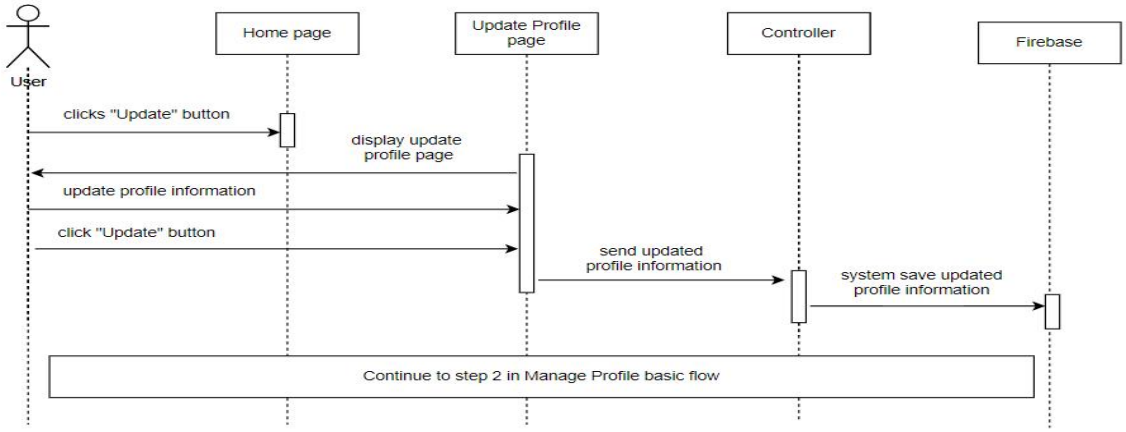


Figure 2.30 Profile [Alternative Flow: Update Profile] Sequence Diagram

Figure 2.30 shows the update profile alternative flow in manage profile module. The flow starts with user clicks on the “Update” button. System display update profile page for user to update the profile information. Then click the “Update” button for the system to save it to the database. User continues the flow with step 2 in basic flow.

CHAPTER 3

3.1 INTERFACE DESIGN

Figure 3.1 shows the register account interface of Blood Bank Application. The users are required to register an account for the first time by inserting the email, password and type of user. Then click the "Register" button for registering. An account will be created.

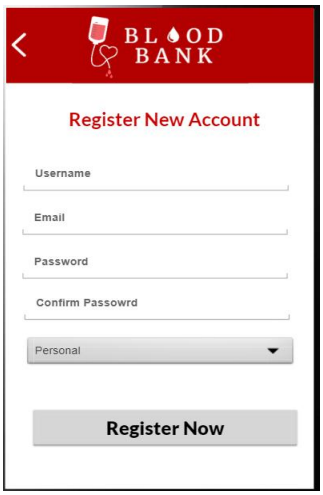


Figure 3.1 Register Account Interface

Figure 3.2 shows the check email interface of Blood Bank Application. Users are needed to insert the registered email and click the "Check Email" button. An email is send to the email address to reset the password.

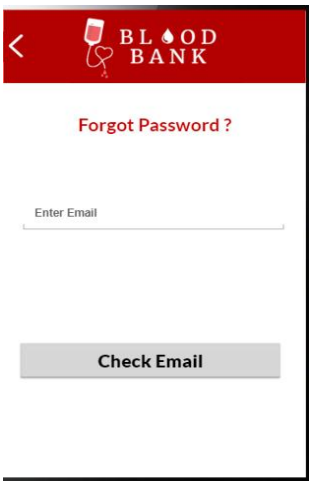


Figure 3.2 Check Email Interface

Figure 3.3 shows the login interface of Blood Bank Application. The users are required to login by inserting the correct email and password, then click the “Login” button. System will redirect to the home page interface when login is successful.

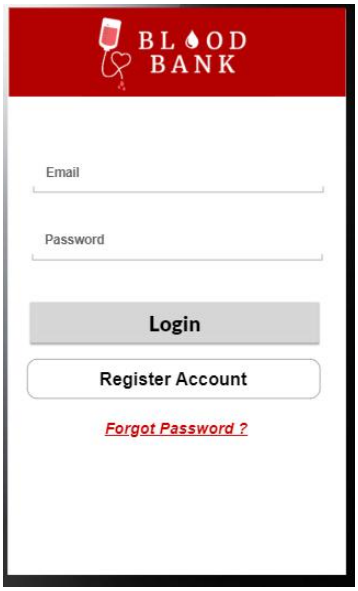


Figure 3.3 Login Interface

Figure 3.4 shows the home page interface for donation organiser and blood donor user of Blood Bank Application. Users are able to search the hospital or blood bank name for viewing its blood stock status. The home page also shows the information about the blood type compatibility. The profile icon represents the button to view profile interface.

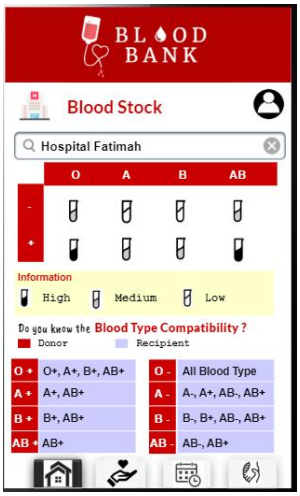


Figure 3.4 Home Page Interface for Donation Organiser and Blood Donor User

Figure 3.5 shows the home page interface for hospital and blood bank organisations of Blood Bank Application. The home page displays the organisation's blood stock. The home page also shows the information about the blood type compatibility. Users are able to update the blood stock status by clicking the "Update" button, the system will redirect to update blood stock interface. The profile icon represents the button to view profile interface.

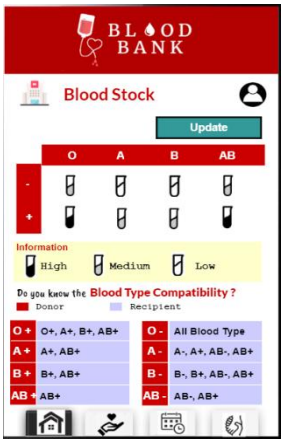


Figure 3.5 Home Page Interface for Hospital and Blood Bank Organisations

Figure 3.6 shows the view profile interface of Blood Bank Application. User able to view their own profile. They are also able to update the profile information by clicking the "Update" button by redirecting to the update profile interface.

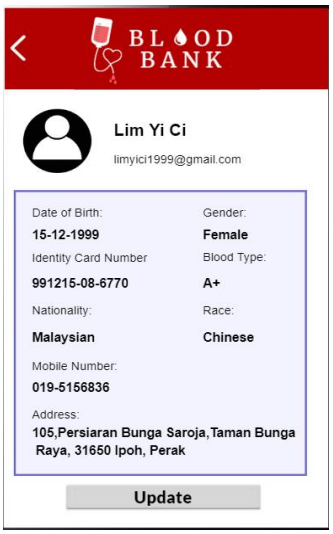


Figure 3.6 View Profile Interface

Figure 3.7 shows the update profile interface of Blood Bank Application. Users are able to update their profile information, then click the "Update" button after finishing inserting.

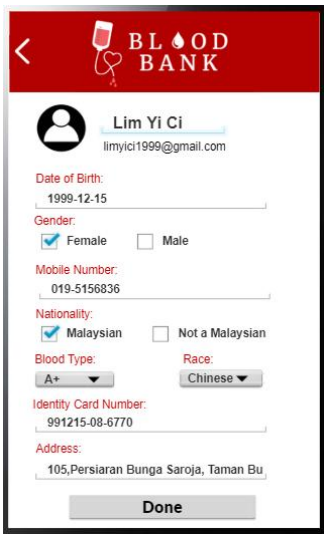


Figure 3.7 Update Profile Interface

Figure 3.8 shows the update blood stock interface of Blood Bank Application. Users are able to choose the blood stock status of each blood type. The status will be updated after clicking the "Update" button. The system will retrieve back to the home page after updating.

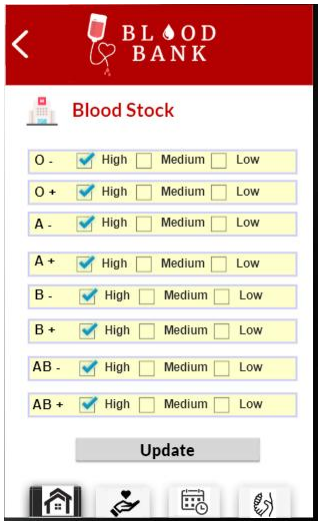


Figure 3.8 Update Blood Stock Interface

Figure 3.9 shows the view donation activity interface for blood donor users of Blood Bank Application. Users are able to view the upcoming donation activities such as the time, date and the venue which is displayed by Google map.

Figure 3.9 View Donation Activity Interface for Blood Donor

Figure 3.10 shows the view donation activity interface for hospital, blood bank and donation organiser of Blood Bank Application. Users are able to view the upcoming donation activities such as the time, date and the venue which is displayed by Google map. Clicking the "Create Donation" button will be retrieved to create donation activity interface.

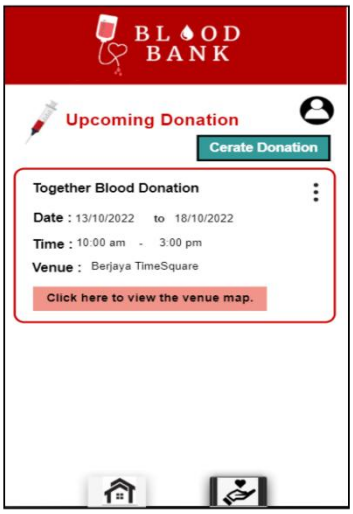


Figure 3.10 View Donation Activity Interface for Hospital, Blood Bank and Donation Organiser

Figure 3.11 shows the map activity interface of Donation Activity. User need to click the search bar in order to display the venue location in the map. The red location shows the venue location.

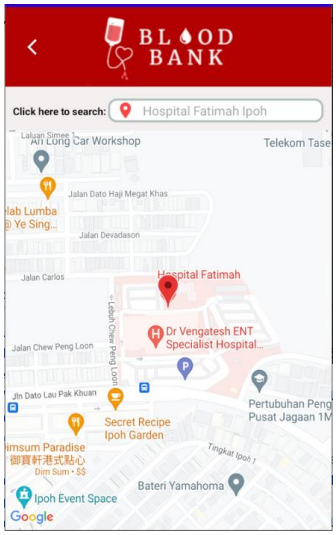


Figure 3.11 Map Activity Interface of Donation Activity

Figure 3.12 shows the create donation activity interface of Blood Bank Application. Users are required to insert the activity date and time. The activity will be created after clicking the “Create” button. Then the system will back to view donation activity interface.

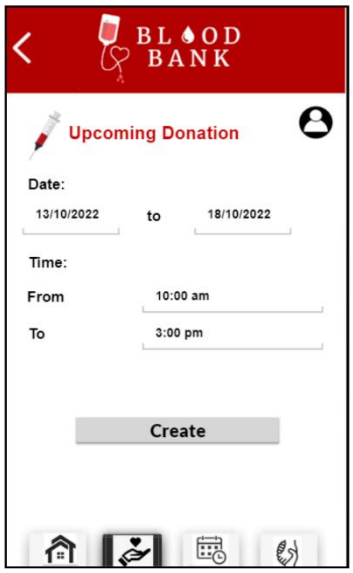


Figure 3.12 Create Donation Activity Interface

Figure 3.13 shows the delete donation activity interface of Blood Bank Application. Users needed to click the kebab menu for showing the delete button and edit button. The activity will be deleted after clicking the “Delete” button. When the “Edit” button is clicked, the system will retrieve to update donation activity interface.

Figure 3.13 Delete Donation Activity Interface

Figure 3.14 shows the update donation activity interface of Blood Bank Application. Users are able to update the donation activity information and click the “Update” button to update the database. The system will back to view donation activity interface.

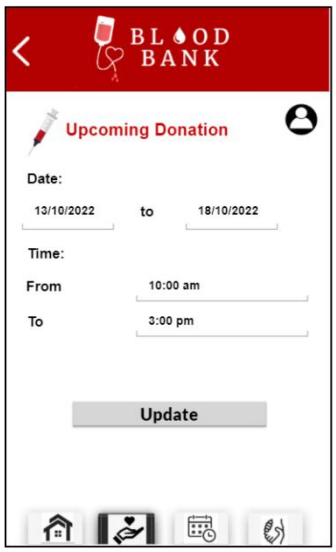


Figure 3.14 Update Donation Activity Interface

Figure 3.15 shows the view appointment interface for blood donor users of Blood Bank Application. Users are able to view the booked appointment information such as the venue, date and time. The "Make Appointment" will retrieve users to create appointment interface.

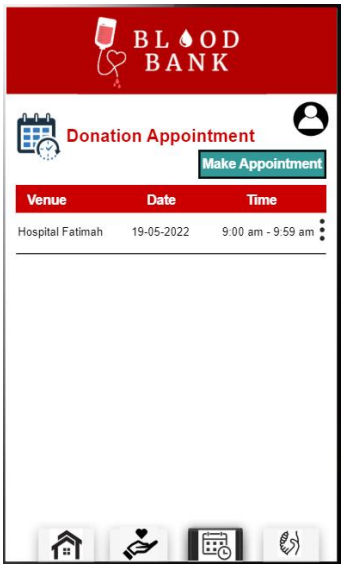


Figure 3.15 View Appointment Interface

Figure 3.16 shows the delete appointment interface for blood donor users of Blood Bank Application. Users are able to delete the appointment by clicking the kebab menu "Delete" button, then the appointment will be deleted from the database.

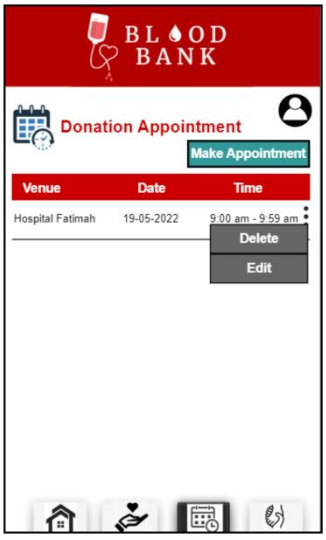


Figure 3.16 Delete Appointment Interface

Figure 3.17 shows the book appointment interface of Blood Bank Application. Users are able to insert the booking appointment information then click the "Book" button for booking. Then the system will retrieve to view appointment interface.

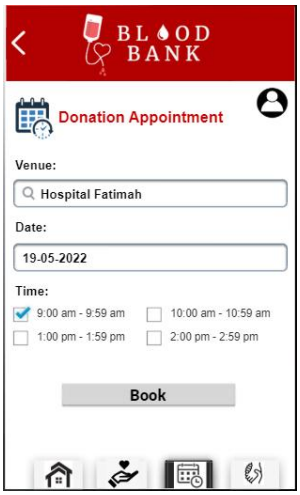


Figure 3.17 Book Appointment Interface

Figure 3.18 shows the update appointment interface of Blood Bank Application. Users update the appointment information and click the "Update" button for updating the information. The system will retrieve to view appointment interface after complete update.

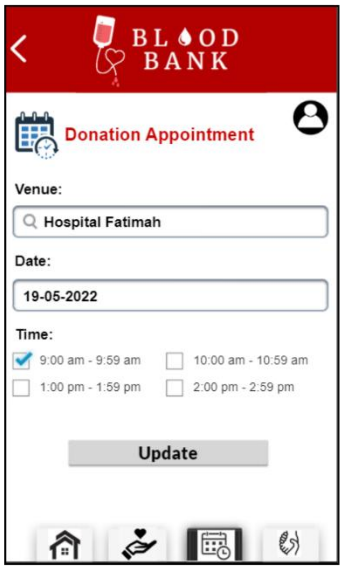


Figure 3.18 Update Appointment Interface

Figure 3.19 shows the view emergency donation interface for blood donor users of Blood Bank Application. The emergency donation will be displayed along with the blood type required and the description.

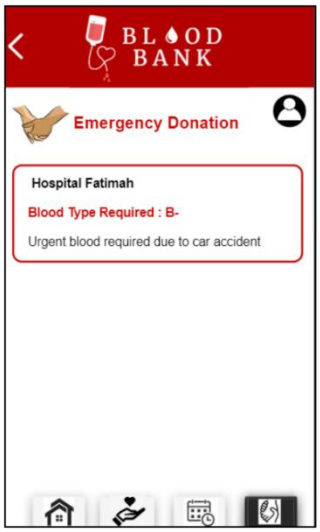


Figure 3.19 View Emergency Donation Interface for Blood Donor

Figure 3.20 shows the view emergency donation interface for hospital and blood bank organisation of Blood Bank Application. Their own emergency donation will be displayed along with the blood type required and the description. The "Create Emergency" button will retrieve to create emergency donation interface after clicking it.

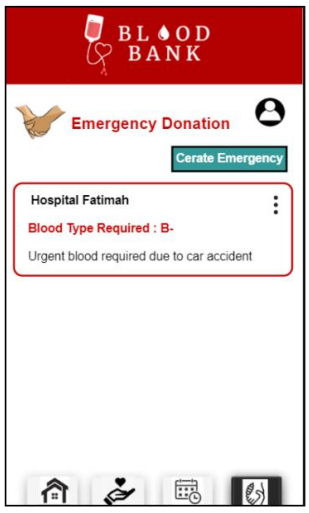


Figure 3.20 View Emergency Donation Interface for Hospital and Blood Bank Organisation

Figure 3.21 shows the create emergency donation interface for hospital and blood bank organisation of Blood Bank Application. Users are required to fill up the emergency donation information and click the “Create” button in order to create the new emergency donation. The system is back to view emergency donation interface.

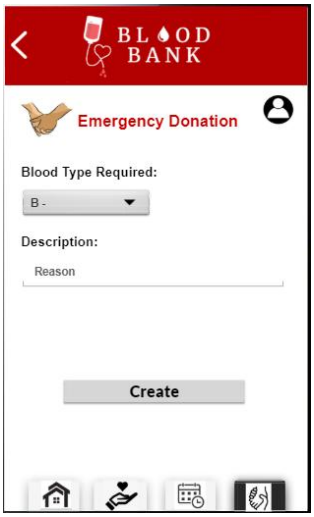


Figure 3.21 Create Emergency Donation Interface for Hospital and Blood Bank Organisation

Figure 3.22 shows the delete emergency donation interface of Blood Bank Application. Users are able to delete the emergency donation by clicking the kebab menu “Delete” button, then the emergency donation will be deleted from the database. For the “Edit” button, the system will retrieve to update emergency donation interface.

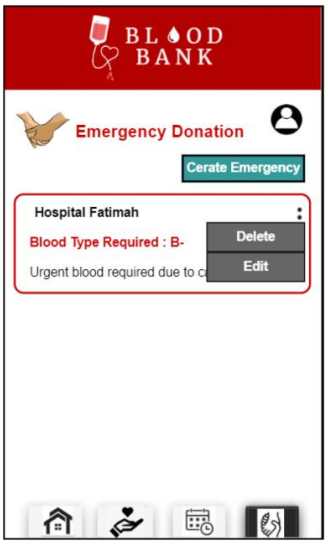


Figure 3.22 Delete Emergency Donation Interface

Figure 3.23 shows the update emergency donation interface of Blood Bank Application. Users are needed to fill up the emergency donation information and click the "Update" button for updating the information. The system will retrieve to view emergency donation interface after complete update.

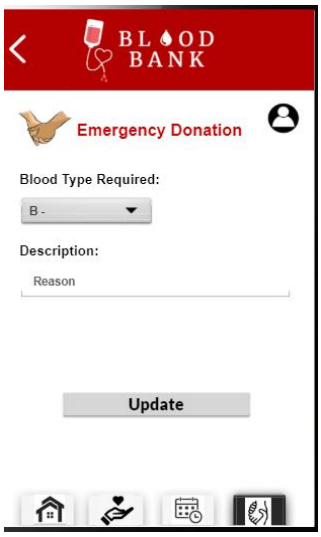


Figure 3.23 Update Emergency Donation Interface

3.2 HARDWARE AND SOFTWARE SPECIFICATION

Hardware	Specification	Function
Android Mobile	Redmi Note 10S	A device used to run the Blood Bank Application.
Laptop	Intel(R) Core(TM) i5-8265U CPU 1.6 GHz with 8GB RAM	To run Android Studio and develop the Blood Bank Application.

Table 3.1 Hardware Specification

Software	Specification	Function
Microsoft Office World 2019	Version 2019	System documentation.
Draw.io	Version 14.6.13	To design and create all the diagrams in the documentations.
Windows 10	Windows 10 Home Version 21H1	Platform to run all the other software required.
Android Studio	Version 4.1	An integrated development environment designed for Android app development using Java language.
Firebase	Version 20.2.3	A mobile and web application development platform that enables real time syncing of data.

Table 3.2 Software Specification

APPENDIX B
SOFTWARE DESIGN DOCUMENT (SDD)

2022

SOFTWARE DESIGN DESCRIPTION (SDD)

Malaysia Blood Bank Android Mobile Application (MBBA)

DOCUMENT APPROVAL

	Name	**Date**
Authenticated by:  _____ Developer	LIM YI CI	26/5/2022
Approved by: _____ Supervisor	DR. NABILAH FILZAH BINTI MOHD RADZUAN	

Software : Draw IO, Microsoft Word

Archiving Place : Google Drive

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1.1 PROJECT DESCRIPTION

The Malaysia Blood Bank Application is a mobile application that helps Malaysian citizens to have a fast and easy way to donate and get the information about blood donation from legal organisations. Furthermore, the blood stock status from each hospital and blood bank is open for the Blood Bank Application user to have a rapid supplement of blood. Malaysia Blood Bank Application consists of 6 modules which are manage sign up, manage activity, manage blood stock, manage appointment, manage emergency and manage profile.

First, manage sign up is required for all types of users to login, register an account and reset password by inserting the user's email address.

In the manage activity module, hospital, donation organiser and blood bank are able to create, update, view and delete a donation activity. Blood donors are only allowed to view the upcoming donation activities.

The next module is manage blood stock. This module allows hospital and donation organiser to update, view and delete their own blood stock status. While blood donors and donation organisers are only allowed to view the blood stock status of each hospital and blood bank by searching the name.

Besides, the manage appointment module allows blood donors to view, book, update and delete the donation appointment. Hospital and blood banks are only allowed to view the appointment that was made by the blood donor.

On top of that, manage emergency module allows hospitals and blood banks to create, update, delete and view the emergency donation. However, blood donors are only allowed to view the emergency donation.

The final module is manage profile. This module allows blood donors, hospital, donation organiser and blood bank to view and update their own profile.

1.2 SYSTEM IDENTIFICATION

This document uses the following convention:
System Identification Number: SDD-MBBA-V01-22

SDD	Software Design Description
MBBA	Malaysia Blood Bank Application
V01	Version 1
22	Year 2022

Table 1.1 System Identification

1.3 ARCHITECTURE / BLUE PRINT

1.3.1 General Architecture

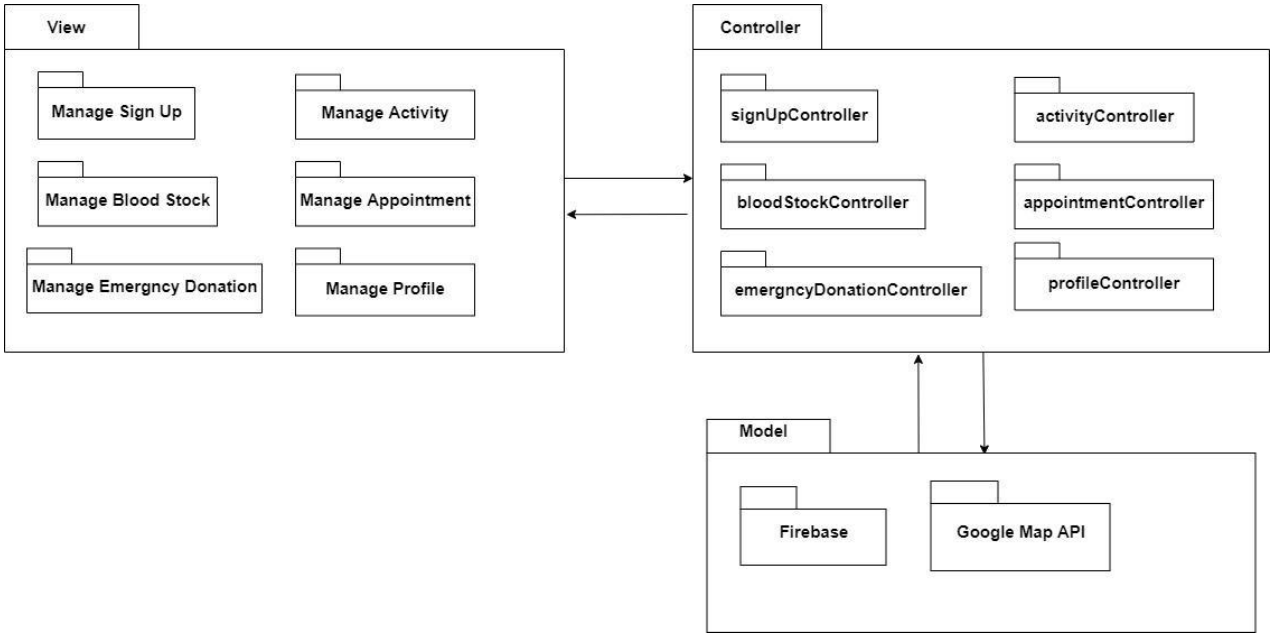


Figure 1.1 MVC Diagram

Figure 1.1 shows the MVC diagram of Blood Bank Application. In the view layer, there are 6 modules which are manage sign up, manage activity, manage blood stock, manage appointment, manage emergency donation and manage profile. Each module has its own controller. In the model layer, Firebase is a database which stores and retrieves data. Google Map displays the location according to the address.

1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

1.4.1 Application Layer

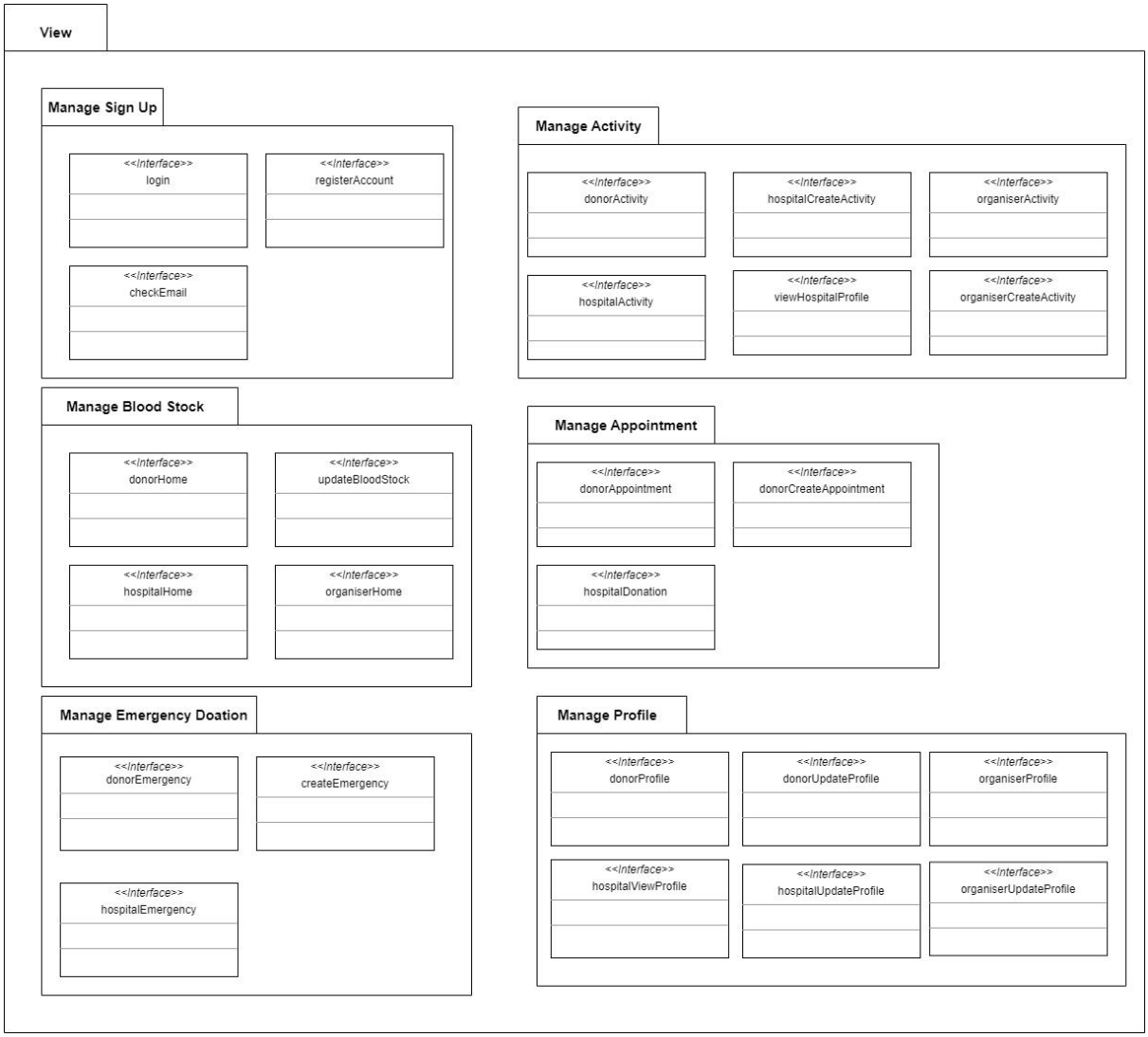


Figure 1.2 Application Layer of MBBA

1.4.1.1 Manage Sign Up

Class Name	Description
login	An interface that allows all users to login by correct email and password.
checkEmail	An interface that sends a reset password to the inserted email address.
registerAccount	An interface that allows first time users to register an account before login to the application.

Table 1.2 Manage Sign Up Interface

1.4.1.2 Manage Activity

Class Name	Description
donorActivity	An interface that allows blood donors to view the upcoming activities information.
organiserActivity	An interface that allows donation organisers to view own created upcoming activities information.
hospitalCreateActivity	An interface that allows blood banks and hospitals to create a donation activity.
hospitalActivity	An interface that allows blood banks and hospitals to view own created upcoming activities information.
viewHospitalProfile	An interface that allows blood donors to view blood banks or hospital profile information.
organiserCreateActivity	An interface that allows donation organisers to create a donation activity.

Table 1.3 Manage Activity Interface

1.4.1.3 Manage Blood Stock

Class Name	Description
donorHome	An interface that allows blood donors to view blood stock status of each hospital and blood bank by searching the hospital name.
organiserHome	An interface that allows donation organisers to view blood stock status of each hospital and blood bank by searching the hospital name.
hospitalHome	An interface that allows blood banks and hospitals to view their own blood stock status.

updateBloodStock	An interface for blood banks and hospitals to update their blood stock status.

Table 1.4 Manage Blood Stock Interface

1.4.1.4 Manage Appointment

Class Name	Description
donorAppointment	An interface for blood donors to view their booking appointment.
hospitalDonation	An interface for hospitals and blood banks to view the booked appointment from blood donors.
donorCreateAppointment	An interface for blood donors to book hospital or blood banks donation appointments.

Table 1.5 Manage Appointment Interface

1.4.1.5 Manage Emergency Donation

Class Name	Description
donorEmergencyDonation	An interface for blood donors to view the emergency donation information.
hospitalEmergency	An interface for hospitals and blood banks to view the emergency donation information.
createEmergency	An interface that allows hospitals and blood banks to create an emergency donation.

Table 1.6 Manage Emergency Donation Interface

1.4.1.6 Manage Profile

Class Name	Description
donorProfile	An interface to view the donor's own profile information.
donorUpdateProfile	An interface to update donor's profile information.
hospitalViewProfile	An interface to hospital and blood bank to view their own profile information.
hospitalUpdateProfile	An interface to the hospital and blood bank to update the profile information.
organiserProfile	An interface to view the organiser's own profile information.

organiserUpdateProfile	An interface to update organiser's profile information.

Table 1.7 Manage Profile Interface

1.4.2 Business Service Layer

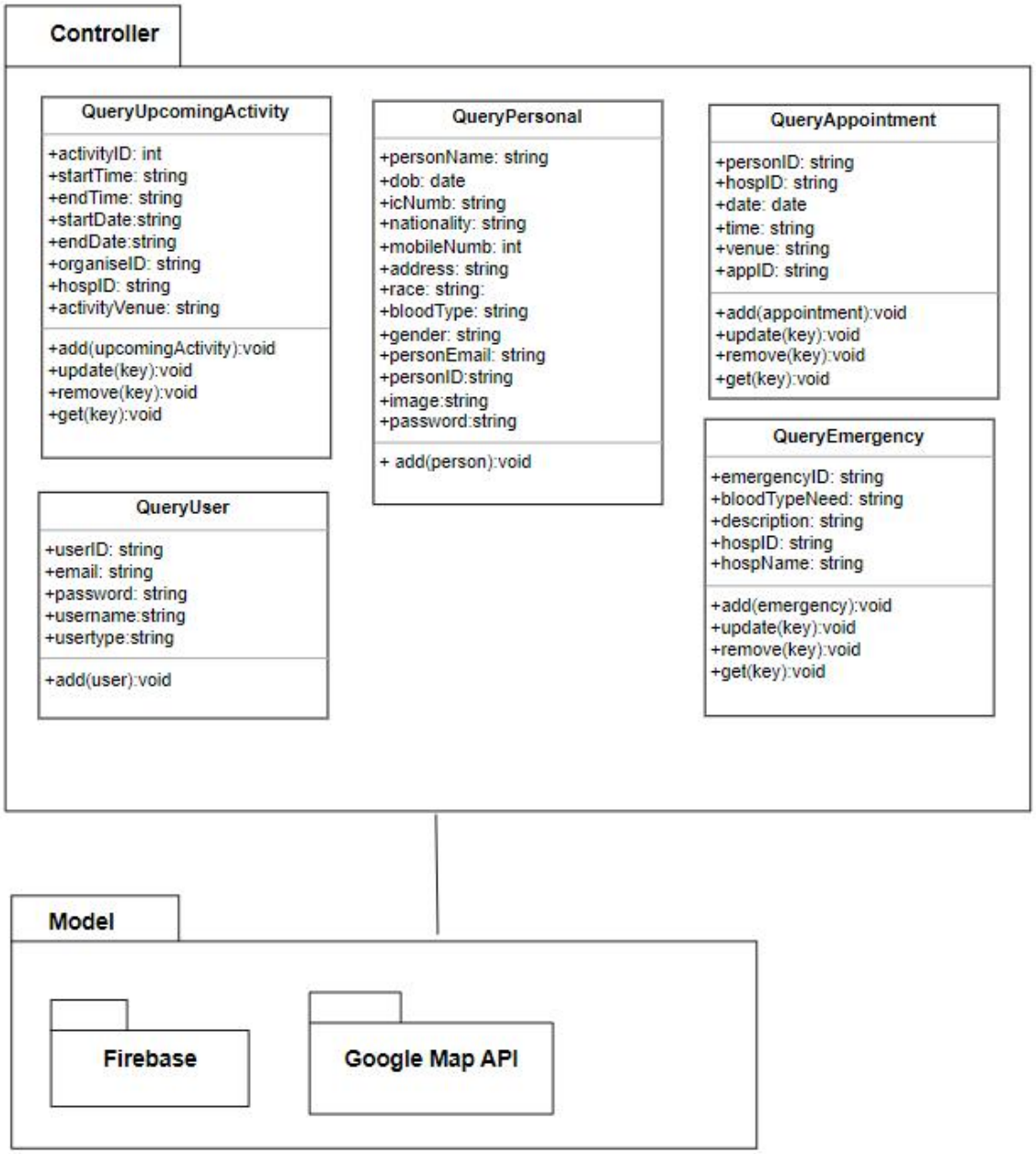


Figure 1.3 Business Layer of MBBA

1.4.2.1 Controller Layer

Class Name	Description
QueryUser	This controller manages the interaction between the database and the manage sign up interfaces.
QueryUpcomingActivity	This controller manages the interaction between the database and the manage activity interfaces.
QueryAppointment	This controller manages the interaction between the database and the manage appointment interfaces.
QueryEmergency	This controller manages the interaction between the database and the manage emergency donation interfaces.
QueryPersonal	This controller manages the interaction between the database and the donor profile interfaces.

Table 1.8 Controller Layer of MBBA

1.4.2.2 Model Layer

Class Name	Description
Firebase	This model is used to store and retrieve the data.
Google Map API	This model is used to view the hospitals and blood banks location.

Table 1.9 Model Layer of MBBA

CHAPTER 2

2.1 DETAILED DESCRIPTION

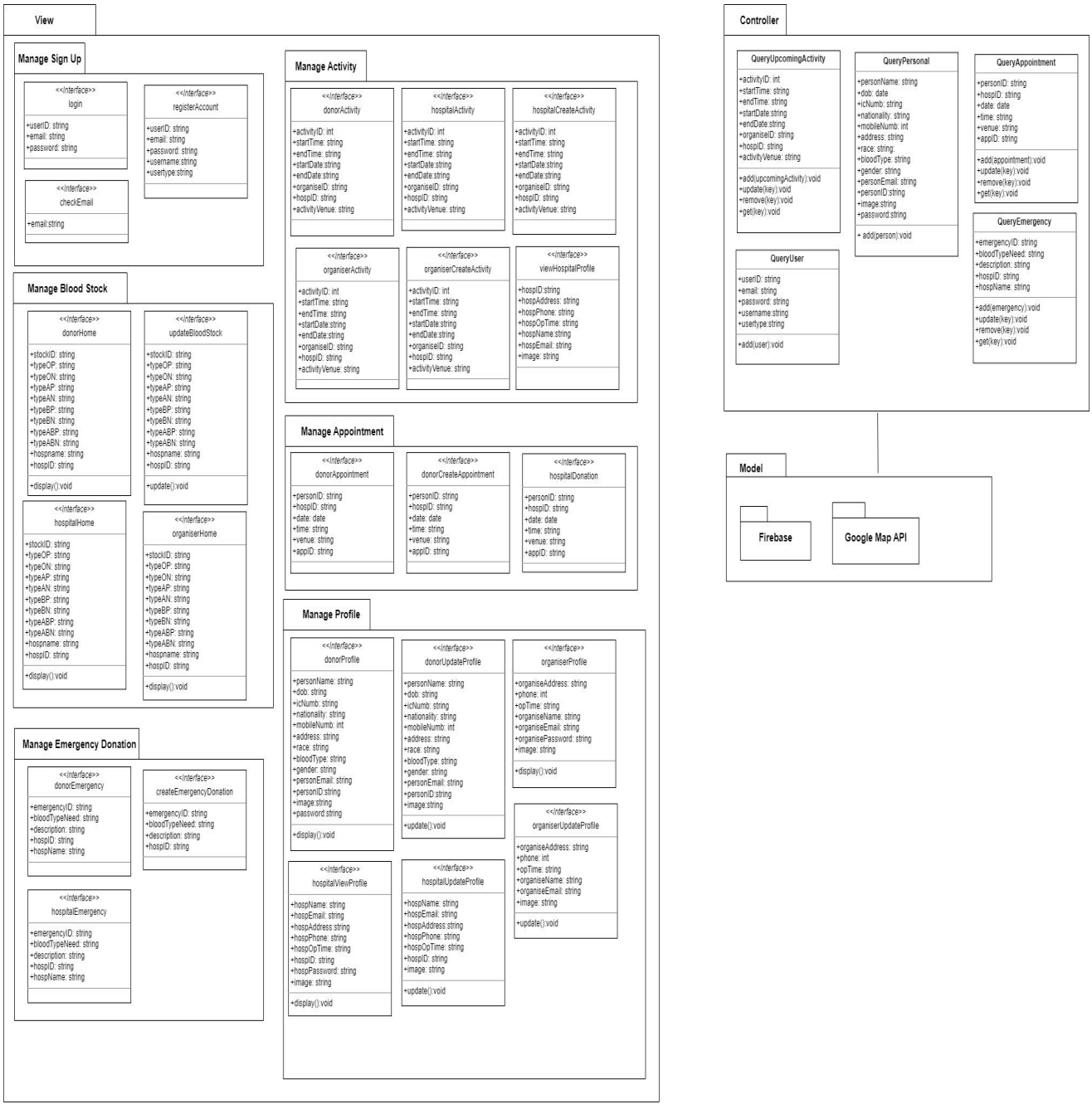


Figure 2.1 Detailed Design

2.1.1 Manage Sign Up

2.1.1.1 Login

Class Type	Boundary Class	
Responsibility	An interface that allows all users to login into the application by correct email and password.	
Attributes	**Attributes Name**	**Attributes Type**
	userID email password	String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.1 Detail Description Login

2.1.1.2 checkEmail

Class Type	Boundary Class	
Responsibility	An interface that sends a reset password to the inserted email address.	
Attributes	**Attributes Name**	**Attributes Type**
	email	String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.2 Detail Description checkEmail

2.1.1.3 registerAccount

Class Type	Boundary Class	
Responsibility	An interface that allows first time users to register an account before login to the application.	
Attributes	**Attributes Name**	**Attributes Type**
	userID email password username usertype	String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.3 Detail Description registerAccount

2.1.1.4 QueryUser

Class Type	Controller Class	
Responsibility	This controller manages the interaction between the database and the manage sign up interfaces.	
Attributes	**Attributes Name**	**Attributes Type**
	userID email password username	String String String String

	usertype	String
Methodes	**Method Name**	**Description**
	add(user)	Add user information to database.
Algorithm	**add()** BEGIN Insert email, password, username and usertype Register account Redirect to login page END	

Table 2.4 Detail Description QueryUserController

2.1.2 Manage Activity

2.1.2.1 donorActivity

Class Type	Boundary Class	
Responsibility	An interface that allows blood donors to view the upcoming activities information.	
Attributes	**Attributes Name**	**Attributes Type**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	String String String String String String String String

Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.5 Detail Description viewDonationActivity

2.1.2.2 hospitalActivity

Class Type	Boundary Class	
Responsibility	An interface that allows blood banks and hospitals to view own created upcoming activities information.	
Attributes	**Attributes Name**	**Attributes Type**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	String String String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.6 Detail Description hospitalActivity

2.1.2.3 hospitalCreateActivity

Class Type	Boundary Class	
Responsibility	An interface that allows blood banks and hospitals to create a donation activity.	
Attributes	**Attributes Name**	**Attributes Type**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	String String String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.7 Detail Description hospitalCreateActivity

2.1.2.4 viewHospitalProfile

Class Type	Boundary Class	
Responsibility	An interface that allows blood donors to view blood banks or hospital profile information.	
Attributes	**Attributes Name**	**Attributes Type**

	hospID hospAddress hospPhone hospOpTime hospName hospEmail image	String String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.8 Detail Description viewHospitalProfile

2.1.2.5 organiserActivity

Class Type	Boundary Class	
Responsibility	An interface that allows blood donors to view the upcoming activities information.	
Attributes	**Attributes Name**	**Attributes Name**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	activityID startTime endTime startDate endDate organiseID hospID activityVenue

Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.9 Detail Description organiserActivity

2.1.2.6 organiserCreateActivity

Class Type	Boundary Class	
Responsibility	An interface that allows blood banks and hospitals to create a donation activity.	
Attributes	**Attributes Name**	**Attributes Name**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	activityID startTime endTime startDate endDate organiseID hospID activityVenue
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.10 Detail Description organiserCreateActivity

2.1.2.7 QueryUpcomingActivity

Class Type	Controller Class	
Responsibility	This controller manages the interaction between the database and the manage activity interfaces.	
Attributes	**Attributes Name**	**Attributes Type**
	activityID startTime endTime startDate endDate organiseID hospID activityVenue	activityID startTime endTime startDate endDate organiseID hospID activityVenue
Methodes	**Method Name**	**Description**
	add(upcomingActivity)	Add new activity information to database.
	update(key)	Update activity information to database.
	remove(key)	Delete an activity.
	get(key)	Get an activity information key from database.
Algorithm	**add(upcomingActivity)** BEGIN Insert activity information Read firebase Insert activity information to database END	

	remove(key) BEGIN Select activity Read firebase Delete activity from database END **update(key)** BEGIN Select activity Insert update activity information Read firebase Update activity information to database END **get(key)** BEGIN Read firebase Select all activity information END

Table 2.11 Detail Description QueryApcomingActivity

2.1.3 Manage Blood Stock

2.1.3.1 donorHome

Class Type	Boundary Class
Responsibility	An interface that allows blood donors to view blood stock status of each hospital and blood bank by searching the hospital name.

Attributes	**Attributes Name**	**Attributes Type**
	stockID	String
	typeOP	String
	typeON	String
	typeAP	String
	typeAN	String
	typeBP	String
	typeBN	String
	typeABP	String
	typeABN	String
	hospID	String
	hospname	String
Methodes	**Method Name**	**Description**
	Display()	Display blood stock from selected hospital or blood bank name
Algorithm	**Display()** BEGIN Insert hospital or blood bank name. Read firebase. Retrieve data from firebase. Display blood stock status. END	

Table 2.12 Detail Description donorHome

2.1.3.2 updateBloodStock

Class Type	Boundary Class

Responsibility	An interface for blood banks and hospitals to update their blood stock status.	
Attributes	**Attributes Name**	**Attributes Type**
	stockID	String
	typeOP	String
	typeON	String
	typeAP	String
	typeAN	String
	typeBP	String
	typeBN	String
	typeABP	String
	typeABN	String
	hospID	String
	hospname	String
Methodes	**Method Name**	**Description**
	Update()	Update blood stock
Algorithm	**Update()** BEGIN Update blood stock status. Read firebase. Update blood stock status to firebase. END	

Table 2.13 Detail Description updateBloodStock

2.1.3.3 hospitalHome

Class Type	Boundary Class

Responsibility	An interface that allows blood banks and hospitals to view their own blood stock status.	
Attributes	**Attributes Name**	**Attributes Type**
	stockID typeOP typeON typeAP typeAN typeBP typeBN typeABP typeABN hospID hospname	String String String String String String String String String String String
Methodes	**Method Name**	**Description**
	Display()	Display blood stock.
Algorithm	**Display()** BEGIN Read firebase. Retrieve data from firebase. Display blood stock status. END	

Table 2.14 Detail Description hospitalHome

2.1.3.4 organiserHome

Class Type	Boundary Class	
Responsibility	An interface that allows donation organisers to view blood stock status of each hospital and blood bank by searching the hospital name.	
Attributes	**Attributes Name**	**Attributes Type**
	stockID typeOP typeON typeAP typeAN typeBP typeBN typeABP typeABN hospID hospname	String String String String String String String String String String String
Methodes	**Method Name**	**Description**
	Display()	Display blood stock from selected hospital or blood bank name
Algorithm	**Display()** BEGIN Insert hospital or blood bank name. Read firebase. Retrieve data from firebase. Display blood stock status.	

	END

Table 2.15 Detail Description organiserHome

2.1.4 Manage Appointment

2.1.4.1 donorAppointment

Class Type	Boundary Class	
Responsibility	An interface for blood donors to view their booking appointment.	
Attributes	**Attributes Name**	**Attributes Type**
	date time venue personID hospID appID	String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.16 Detail Description donorAppointment

2.1.4.2 donorCreateAppointment

Class Type	Boundary Class	
Responsibility	An interface for blood donors to book hospital or blood banks donation appointments.	
Attributes	**Attributes Name**	**Attributes Type**

	date time venue personID hospID appID	String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.17 Detail Description donorCreateAppointment

2.1.4.3 hospitalDonation

Class Type	Boundary Class	
Responsibility	An interface for hospitals and blood banks to view the booked appointment from blood donors.	
Attributes	**Attributes Name**	**Attributes Type**
	date time venue personID hospID appID	String String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable

Algorithm	Not Applicable

Table 2.18 Detail Description hospitalDonationAppointment

2.1.4.4 QueryAppointment

Class Type	Controller Class	
Responsibility	This controller manages the interaction between the database and the manage appointment interfaces.	
Attributes	**Attributes Name**	**Attributes Type**
	date time venue personID hospID appID	String String String String String String
Methodes	**Method Name**	**Description**
	add(appointment)	Add new appointment information to database.
	update(key)	Update appointment information to database.
	remove(key)	Delete an appointment.
	get(key)	Get an appointment information key from database.
Algorithm	**add(appointment)** BEGIN Insert appointment information Read firebase Insert appointment information to database	

	END **remove(key)** BEGIN Select appointment Read firebase Delete appointment from database END **update(key)** BEGIN Select appointment Insert update appointment information Read firebase Update appointment information to database END **get(key)** BEGIN Read firebase Select all appointment information END

Table 2.19 Detail Description appointmentController

2.1.5 Manage Emergency Donation

2.1.5.1 donorDonation

Class Type	Boundary Class

Responsibility	An interface for blood donors to view the emergency donation information.	
Attributes	**Attributes Name**	**Attributes Type**
	emergencyID bloodTypeNeed description hospID hospName	String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.20 Detail Description donorEmergency

2.1.5.2 createEmergency

Class Type	Boundary Class	
Responsibility	An interface that allows hospitals and blood banks to create an emergency donation.	
Attributes	**Attributes Name**	**Attributes Type**
	emergencyID bloodTypeNeed description hospID	String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable

Algorithm	Not Applicable

Table 2.21 Detail Description createEmergency

2.1.5.3 hospitalEmergency

Class Type	Boundary Class	
Responsibility	An interface for hospitals and blood banks to view the emergency donation information.	
Attributes	**Attributes Name**	**Attributes Type**
	emergencyID bloodTypeNeed description hospID hospName	String String String String String
Methodes	**Method Name**	**Description**
	Not Applicable	Not Applicable
Algorithm	Not Applicable	

Table 2.22 Detail Description hospitalEmergency

2.1.5.4 QueryEmergency

Class Type	Controller Class	
Responsibility	This controller manages the interaction between the database and the manage emergency donation interfaces.	
Attributes	**Attributes Name**	**Attributes Type**
	emergencyID	String

	bloodTypeNeed description hospID hospName	String String String String
Methodes	**Method Name**	**Description**
	add(emergency)	Add new emergency information to database.
	update(key)	Update emergency information to database.
	remove(key)	Delete an emergency.
	get(key)	Get an emergency information key from database.
Algorithm	**add(emergency)** BEGIN Insert emergency information Read firebase Insert emergency information to database END **remove(key)** BEGIN Select emergency Read firebase Delete emergency from database END **update(key)** BEGIN	

	Select emergency Insert update emergency information Read firebase Update emergency information to database END **get(key)** BEGIN Read firebase Select all emergency information END

Table 2.23 Detail Description QueryEmergency

2.1.6 Manage Profile

2.1.6.1 donorProfile

Class Type	Boundary Class	
Responsibility	An interface to view the donor's own profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	personName	String
	dob	String
	icNumb	String
	nationality	String
	mobileNumb	INT
	address	String
	race	String
	bloodType	String
	gender	String

	personEmail personID image password	String String String String
Methodes	**Method Name**	**Description**
	Display()	Display donor's profile.
Algorithm	**Display()** BEGIN Read firebase. Retrieve data from firebase. Display donor's profile. END	

Table 2.24 Detail Description donorProfile

2.1.6.2 donorUpdateProfile

Class Type	Boundary Class	
Responsibility	An interface to update donor's profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	personName dob icNumb nationality mobileNumb address race bloodType	String String String String INT String String String

	gender personEmail personID image password	String String String String String
Methodes	**Method Name**	**Description**
	Update()	Update donor's profile
Algorithm	**Update()** BEGIN Update donor's profile information. Read firebase. Update donor's profile information to firebase. END	

Table 2.25 Detail Description donorUpdateProfile

2.1.6.3 hospitalViewProfile

Class Type	Boundary Class	
Responsibility	An interface to hospital and blood bank to view their own profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	hospName hospEmail hospAddress hospPhone hospOpTime hospID hospPassword	String String String String String String String

	image	String
Methodes	**Method Name**	**Description**
	Display()	Display hospital or blood bank's profile.
Algorithm	**Display()** BEGIN Read firebase. Retrieve data from firebase. Display hospital or blood bank's profile. END	

Table 2.26 Detail Description hospitalViewProfile

2.1.6.4 hospitalUpdateProfile

Class Type	Boundary Class	
Responsibility	An interface for hospital and blood bank to update the profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	hospName hospEmail hospAddress hospPhone hospOpTime hospID hospPassword image	String String String String String String String String
Methodes	**Method Name**	**Description**
	Update()	Update hospital or blood bank's profile

Algorithm	**Update()** BEGIN Update hospital or blood bank's profile information. Read firebase. Update hospital or blood bank's profile information to firebase. END

Table 2.27 Detail Description hospitalUpdateProfile

2.1.6.5 organiserProfile

Class Type	Boundary Class	
Responsibility	An interface to view the organiser's own profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	organiseAddress phone opTime organiseName organiseEmail organisePassword image	String String String String String String String
Methodes	**Method Name**	**Description**
	Display()	Display organiser's profile.
Algorithm	**Display()** BEGIN Read firebase. Retrieve data from firebase. Display organiser's profile.	

	END

Table 2.28 Detail Description organiserProfile

2.1.6.6 organiserUpdateProfile

Class Type	Boundary Class	
Responsibility	An interface for hospital and blood bank to update the profile information.	
Attributes	**Attributes Name**	**Attributes Type**
	organiseAddress phone opTime organiseName organiseEmail organisePassword image	String String String String String String String
Methodes	**Method Name**	**Description**
	Update()	Update organiser's profile
Algorithm	**Update()** BEGIN Update organiser's profile information. Read firebase. Update organiser's profile information to firebase. END	

Table 2.29 Detail Description organiserUpdateProfile

2.1.6.7 QueryPersonal

Class Type	Controller Class

Responsibility	This controller manages the interaction between the database and the donor profile interfaces.	
Attributes	**Attributes Name**	**Attributes Type**
	personName dob icNumb nationality mobileNumb address race bloodType gender personEmail personID image password	String String String String INT String String String String String String String String
Methodes	**Method Name**	**Description**
	add(person)	Add a new person information to database.
Algorithm	**Update()** BEGIN Insert person information Read firebase Insert person information to database END	

Table 2.30 Detail Description QueryPersonal

2.2 DATA DICTIONARY

User

Field Name	Description	Data Type	Constraint
UserID	User ID	VARCHAR(100)	PK
email	Email Address	VARCHAR(200)	-
password	Password	VARCHAR(100)	-
username	Username	VARCHAR(50)	-
usertype	User type	VARCHAR(100)	-

Table 2.31 User Table

Personal

Field Name	Description	Data Type	Constraint
personID	Person ID	VARCHAR(100)	PK
personName	Person Name	VARCHAR(100)	-
password	Password	VARCHAR(100)	-
dob	Date of Birth	DATE	-
icNumb	IC Number	INT	-
nationality	Nationality	VARCHAR(100)	-
mobileNumb	Mobile Number	INT	-
address	Address	VARCHAR(200)	-
race	Race	VARCHAR(50)	-
bloodType	Blood Type	VARCHAR(50)	-
gender	Gender	VARCHAR(50)	-
personEmail	Person Email	VARCHAR(100)	-
image	Hospital Blood Bank Profile Image	VARCHAR(200)	-

Table 2.32 Personal Table

Organiser

Field Name	Description	Data Type	Constraint
organiseID	Organisation ID	VARCHAR(100)	PK
organiseAddress	Organisation Address	VARCHAR(200)	-
phone	Phone Number	INT	-
opTime	Operation Time	VARCHAR(50)	-
organiseName	Organisation Name	VARCHAR(100)	-
organiseEmail	Organisation Email	VARCHAR(100)	-
organisePassword	Organisation Password	VARCHAR(100)	-
image	Organisation Profile Image	VARCHAR(200)	-

Table 2.33 Organiser Table

Hospital Blood Bank

Field Name	Description	Data Type	Constraint
hospID	Hospital Blood Bank ID	VARCHAR(100)	PK
hospAddress	Hospital Blood Bank Address	VARCHAR(200)	-
hospPhone	Hospital Blood Bank Phone Number	INT	-
hospOpTime	Hospital Blood Bank Operation Time	VARCHAR(50)	-
hospName	Hospital Blood Bank Name	VARCHAR(100)	-
hospEmail	Hospital Blood Bank Email	VARCHAR(100)	-
hospPassword	Hospital Blood Bank Password	VARCHAR(100)	-
image	Hospital Blood Bank Profile Image	VARCHAR(200)	-

Table 2.34 Hospital Blood Bank Table

UpcomingActivity

Field Name	Description	Data Type	Constraint
activityID	Activity ID	VARCHAR(100)	PK
startTime	Starting Time	VARCHAR(50)	-
endTime	End Time	VARCHAR(50)	-
startDate	Starting Date	VARCHAR(50)	-
endDate	End Date	VARCHAR(50)	-
activityVenue	Activity Venue	VARCHAR(100)	-
organiseID	Organisation ID	VARCHAR(100)	FK
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK

Table 2.35 Activity Table

Emergency

Field Name	Description	Data Type	Constraint
emergencyID	Emergency ID	VARCHAR(100)	PK
bloodTypeNeed	Blood Type Need	VARCHAR(10)	-
description	Description	VARCHAR(200)	-
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK
hospName	Hospital Blood Bank Name	VARCHAR(100)	FK

Table 2.36 Emergency Table

Appointment

Field Name	Description	Data Type	Constraint
appID	Appointment ID	VARCHAR(100)	PK
date	Date	DATE	-
time	Time	VARCHAR(50)	-
venue	Venue	VARCHAR(100)	-
personID	Person ID	VARCHAR(100)	FK

hospID	Hospital Blood Bank ID	VARCHAR(100)	FK

Table 2.37 Appointment Table

Blood Stock

Field Name	Description	Data Type	Constraint
stockID	Blood Stock ID	VARCHAR(100)	PK
typeOP	Type O Positive	VARCHAR(50)	-
typeON	Type O Negative	VARCHAR(50)	-
typeAP	Type A Positive	VARCHAR(50)	-
typeAN	Type A Negative	VARCHAR(50)	-
typeBP	Type B Positive	VARCHAR(50)	-
typeBN	Type B Negative	VARCHAR(50)	-
typeABP	Type AB Positive	VARCHAR(50)	-
typeABN	Type AB Negative	VARCHAR(50)	-
hospID	Hospital Blood Bank ID	VARCHAR(100)	FK
hospname	Hospital Blood Bank Name	VARCHAR(100)	FK

Table 2.38 Blood Stock Table

APPENDIX C
USER ACCEPTANCE TEST(UAT)

2023

USER ACCEPTANCE TEST (UAT)

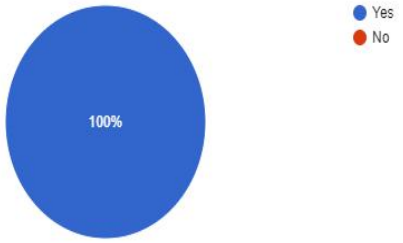
Malaysia Blood Bank Android Mobile Application (MBBA)


Manage Sign Up
1. Cannot login when insert with wrong email or password
Copy
13 responses
Yes
No
100%
2. Cannot login using wrong email address format.
Copy
13 responses
Yes
No
100%
3. Able to login with correct email and password
Copy
13 responses
Yes
No
100%
4. Able to register account
Copy
13 responses
Yes
No
100%

5. Able to reset password through email address

Copy

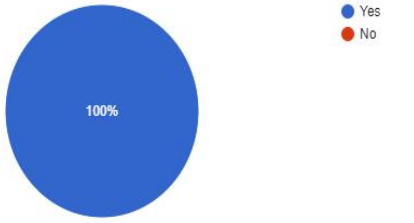
13 responses



6. Cannot login when select wrong user type .

Copy

13 responses

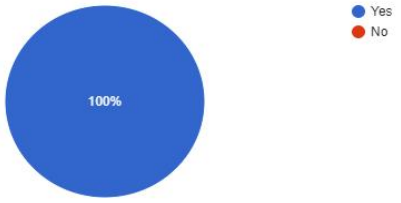


Manage Emergency Donation

7. Able to display all emergency donation notice

Copy

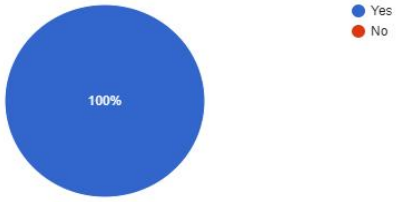
13 responses



8. Able to create,edit and delete an emergency notice

Copy

13 responses

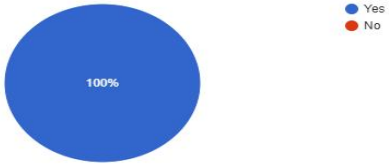


Manage Appointment

9. Able to display appointment information

Copy

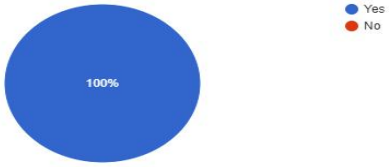
13 responses



10. Able to create,edit and delete appointment information

Copy

13 responses

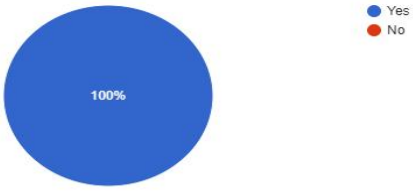


Manage Profile

11. Able to view profile information

Copy

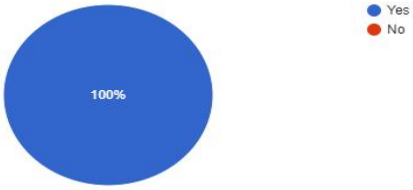
13 responses



12. Able to update profile information.

Copy

13 responses

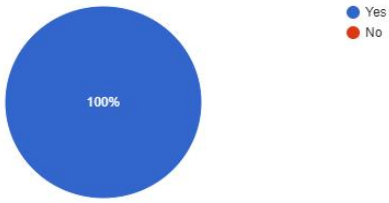


Manage Activity

13. Able to view upcoming donation

Copy

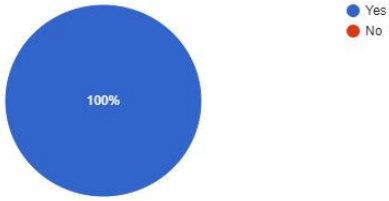
13 responses



14. Able to view venue location in Google Map

Copy

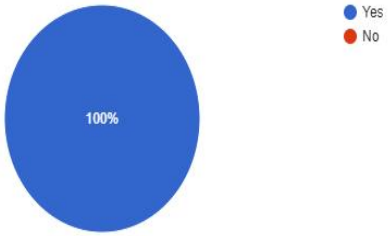
13 responses



15. Able to create,edit and delete upcoming donation.

Copy

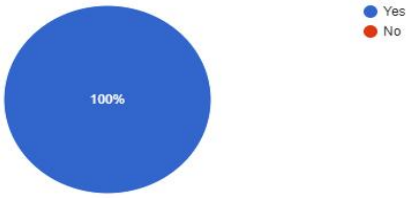
13 responses



Manage Blood Stock

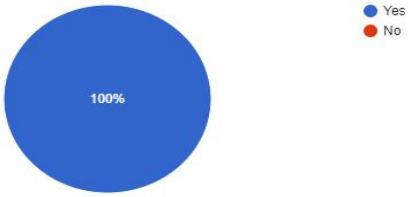
16. Able to update blood stock. Copy

13 responses



17. Able to view blood stock status by searching the hospital or blood bank name. Copy

13 responses



18. All the data insert with blank input will display error message Copy

13 responses