SUNWAY UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTING AND INFORMATION SYSTEMS

SEG2202 SOFTWARE ENGINEERING

ACADEMIC SESSION: MARCH 2021

FINAL ASSESSMENT COURSEWORK **DUE DATE: 12 JULY 2021**

STUDENT NAME: **Raveesh Shibchurn**

STUDENT ID: **19053156**

PROGRAMME: **Bachelor's in Computer Science**

YEAR / SEMESTER: March 2021, Semester 6

INSTRUCTIONS TO CANDIDATES

- This is an **INDIVIDUAL** assignment.
- This assignment will contribute **30%** to the coursework mark and to your final grade.

IMPORTANT NOTICE

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

Coursework submitted after the deadline but within 1 week will be accepted for a maximum mark of 40%. Work handed following the extension of 1 week after the original deadline will be regarded as a non-submission and marked zero.

Academic Honesty Acknowledgement

"I Raveesh Shibchurn (student name) verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

R.shibchurn / 4th July 2021 (Student's Signature / Date)

Table of Contents

+Blood App Report

Introduction

Blood is a sine qua non-substance for one to live. Lifeblood transports necessary nutrients to cells and is the medium for the extraction of metabolic waste from the cellular part. Every day around the globe, people encounter acute accidents or suffer from dire diseases; thus, they are in need of blood transfusion to sustain their metabolism and immune system. This vital body fluid is composed of various fragments – plasma, blood platelets, white blood cells, only to mention these – which can be segmented into their components; thus, saving more lives since some patients sometimes require transfusion of specific elements of blood. Moreover, an individual has a specific blood group which may be either group A, B, O or AB. During a transfusion, the patient's blood type should match with the donors. In certain situations, a patient might not get any donor matching his blood group; thus, the patient may lose his life if he does not get blood promptly.

With the advent of mobile technology, everything can be executed within seconds; therefore, an android blood bank application seems to be the perfect panacea to this issue. The Android application- Blood Bank is developed for easy search of available blood from blood banks or blood donors in nearby areas in cases of emergencies without any delay.

The administrators are the main authority in the system since they are the link between the blood bank and the users. They are the ones that accept any blood requests and verify the stocks in the blood bank's inventory.

Project objective

+Blood application has as main aim to ease the communication between blood seekers, blood donors and blood banks. In order to achieve this goal, we implemented various features, explained in the project scope.
In a world whereby people have a mobile device with them round the clock, this application will prove to be valuable as the seekers will have access to a network of donors within their area; thus, they can contact a donor with a finger touch. This app will allow quick communication between the donor and seeker that is critical in times of dire need of blood. We aim to help the user make a quick decision in selecting a donor since the application saves all the donors' data. We hope that this app proves to be useful in case of emergencies where blood can be dispensed quickly; thus, help in saving thousands of lives.

Project Scope

The application consists of several system requirements. Firstly, the app is built up with an Android Sdk as front end and SQL system as backend. Furthermore, it should be installed on an android device with processor – i3, hard disk – 5 GB and memory – 1GB RAM that have access to the internet. These are all fundamentals to access the features of the app.

The Blood bank app is designed to provide a user-friendly environment. The application is not made to make any source of income. The features, of the app are self-explanatory and quite easy to use. The following describes the features:

- A sign in page allows the user to authenticate himself to have access to the app. Thus, this authentication system allows only authorised individual to have access to the app.
- A sign-up page where new user can direct to register and provide the system with his information– name, address, phone number and email along with their details of blood group, and authentication details.
- A search page where the users can view a list of potential donors, classified by blood groups A, B, AB, and O. The users can use the search filter to display a list of donors having specific blood, for example, A+.
- A profile page where the users can edit their information and change the settings according to their preferences. The users can also switch their profiles from donor to requestor or vice visa by checking or unchecking the donor tag.
- The users can view a specific donor profile where details- blood type, location, amongst others- of the donor are displayed. The users can call the user directly by clicking on the call button to receive more details about the donor.
- The mobile application will allow the blood seekers to view the stock of the blood group in the blood bank pages.
- The request page is made for the user to request a particular or related blood group and they can add a specific donor that they contacted to get his blood pint donation.

Moreover, the app can be used by any person and medical staff who can register promptly for a patient in need of blood using the patient details. The administrator has the responsibility to check the enquiries made by the seekers to obtain blood from the blood bank; thus, the admin will have to conduct the transaction and make sure the blood is delivered. The administrator also needs to update the blood bank system when the blood bank staff sends the updated stock of the blood bank. When a donor updates his information, this application updates promptly the donors' data.

The application can be implemented in any country, with different admins controlling the systems in a specific country. Thus, the admins will ensure that the users will have the donor list and blood bank details from their respective countries. Therefore, allowing the seekers to search for donors within its proximity. However, the limitation of the +Blood is that it works with the internet; thus, if the server is down, the user will no longer have access to the donors' data.

The timeline of this project is about 6 months and after releasing the software, our software company will have to maintain and update the system requirements. The budget allocation from the blood bank authority is limited to 50,000 RM. The amount of money used will depend on numerous factors. Some of the factors are requirement changes, new features implementation request, new non-functional requirements request amongst others. This is a limited budget; thus, we need to plan a very thoughtful strategy to save money.

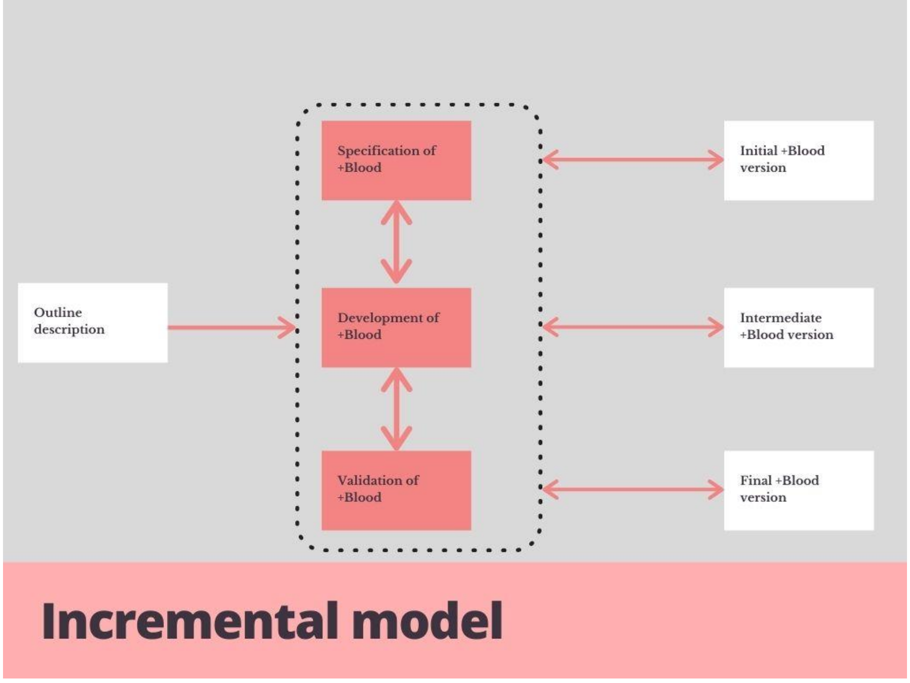
Software Process Model

The most efficient software process model to use is the agile incremental model. Incremental development model consists of various development cycles whereby different version of the software is produced incrementally. In another increment, the developers can add new functions to the initial versions. Thus, applying an incremental model will prove to be most convenient since we need to update the application version based on feedback and testing.

Due to a limited budget, we should consider the most cost-friendly model. Thus, an incremental model is ideal because the cost to add changes in the system is less compared to other models. As mentioned, the system requirements may evolve; thus, in an incremental model, changes can be added during any stage; thus, making it more cost-efficient. Moreover, incremental development is more budget-friendly because when the team has listed the requirement priorities, we can stop the project at any stage where the incremental price of features is lower than the incremental development price.

Moreover, the stakeholders, mainly the user and customer, find it hard to identify and detail all the system requirements. In the incremental model, customers are allowed to give any feedback while the project is in progress. If ever the customers have any uncertainty about the +Blood system requirement, they can ask to change the requirements at any stage of the model, and we can add the changes in the system.

An incremental process model prioritizes customer involvement by delivering the software at an earlier stage where all the main requirements of the system are included. Thus, using this model is beneficial to +Blood development. Since it is through feedback and testing that the developers can enhance the user and system requirements. Due to a limited time for deployment, the incremental model is the best choice for the development of +Blood.
Below is a diagram to show how the incremental model will work.



The development of the +Blood application relies on this model. Specification, development, and validation of the app are simultaneous activities. The steps to develop the system are broken down into increments. Each increment focuses on a specific features' implementation of +Blood. Once, the development of a system feature begins, the requirements are frozen. Thus, if the system requirements evolve, they will be implemented in the next increment.

Outline description is the process of gathering details of the +Blood. In order to collect this data, the team of developers will interview the stakeholders to grasp the system and user requirements. This will be elaborated in the requirement plan section.

The specification part is also part of the requirement plan. During this step, the developers will gather details of the +Blood from the stakeholders and feedback on the system. It is during this process that we collect, define, and confirm system requirement changes. We will discuss more on this in the next section.

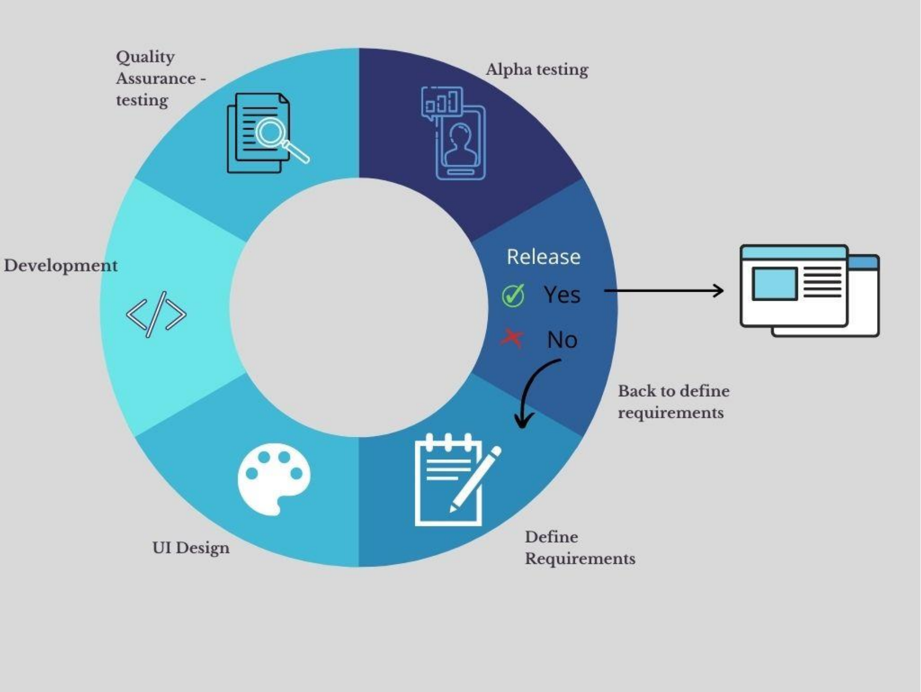
In the development phase, we have both the system design and implementation. These processes, being inter-leaved, proves to be an advantage for the +Blood developing team because the system requirements evolve with time; thus, both the design and implementation needs to be changed.

In the design stage, the team will come up with the architectural design (shown in the UML section), the UI design (prototype), and the database design. The implementation of +Blood is concerned with the programming side. The system is programmed using Android SDK as the front end and SQL system as a backend. The system developers use the UML diagrams and system designs to ensure that the system meets all the requirements.

In the validation stage, we will verify and test the system. First, we will test the various components of the system and eventually evaluate the whole +Blood system. Alpha testing is executed on +Blood. Why Alpha testing? This type of testing is more suitable for +Blood 's software process model since user involvement is prioritized. The users will test alongside the +Blood to find any errors in the system. The users are making practical usage of the +Blood; thus, their point of view will be different compared to the developers. Consequently, the developers will have a grasp on various requirements that they might have omitted. Having the users testing alongside is more cost-efficient since they can halt any expensive unrequired changes.

+Blood system development is abiding by the five principles of the agile method. The agile incremental model is more customer-centric and uses continuous customer feedback to refine successively and deliver a software system. Therefore, allowing the developers to change the requirements at a faster pace and to develop new versions. The +Blood developer team will work alongside the customers to beget a better system since the customer is helping in defining the requirements and testing the application's components. Due to the small timeline for this project, applying an incremental approach to the delivery of the system proves to be efficient. Using the agile approach, the +Blood team will ensure a conducive environment where distributing the work among the developers is based on talent and knowledge. The +Blood app favors a simplistic approach to design.

The agile model has a software lifecycle that is characterized by the software process model that defines the flow of the stages of the application lifecycle. The following diagram illustrates the software lifecycle working alongside the incremental model.



The agile method that we will make use of is Extreme programming. This agile method is more relevant to the development of +Blood. Extreme programming promotes incremental planning, incremental releases, straightforward design, test-first development, and refactoring. XP programmers work alongside the same features, one of them will focus mainly on coding while the other one will focus on the strategic features and will verify for process convulsions in the system.

Requirement Plan

A requirement plan needs to be implemented to gather system and user requirements. The diagram below shows all the stakeholders involved.

Stakeholders	Role
Client Blood Bank authority	Pay for the development of the +Blood application. Define the requirements of the system. Give a description of the system.
Blood Bank manager	Help the blood bank authority in decision making.
User	User of the +Blood application.
Client manager	Interacts with customers routinely to update them on projects. He/she is often the bridge between clients and the company.
Database administrator	Information technician having as responsibility to direct and perform all activities related to maintaining a successful database environment. He/she ensure that the +Blood's databases and their related applications operate functionally and efficiently and securely.
Security Specialist	Handles +Blood IT security, ensuring the system's data stay secure and are protected against cyberattacks.
System administrator	Must manage +Blood application after the release. He/she is the link between the user and the blood bank administrator.
Blood bank administrator	Communicate with the System administrator to confirm blood pint order and send updates on the quantity of blood pints.
System architect	Define the architecture of +Blood application system to achieve user and system requirements. Responsible in segmenting the system into components, defining component interactions and interfaces, and deciding on the technologies and resources to be used in the design.
Software company CEO (Chief Executive Officers)	Senior manager of the software developing company.
Software developer	Develop +Blood application by considering the specified requirements, system architecture and design, and budget received.
Marketing manager	Responsible for managing the promotion and positioning of +Blood application.
Financial manager	Advise the client on suitable business planning and help the client in decision making processes to ensure that the budget chosen for +Blood development is not excessive.
Software tester	Involve in the testing of +Blood's components and the entire system. Verify if the system is abiding to the system requirements and decide whether the application is ready for release.
System accessor	Responsible for evaluating the management, operational, and technical security controls implemented in +Blood to figure out the effectiveness of the controls by taking into account the legal and security standard.

Requirement plan focuses on describing and documenting the system and user requirements. Requirement plan is broken down into 4 stages:

1. Requirement elicitation

This is the process where the software company will work alongside other stakeholders to define the +Blood's requirements, services, performance only to mention these. The first aspect of this stage is requirement discovery in which the company will gather details about the +Blood application and its system and user requirement. The first type of detail gathering that the company will put into practice is a closed interview with the blood bank authority. The following are questions that can be used for this step:

- What aim do you have for users to fulfill in your mobile app?
- What is the primary goal of +Blood application?
- Do you have an existing application and looking for a team to update the current version?
- What is your target audience?
- What is the demographic of your audience?
- Is +Blood accessible for public use or private use?
- What are the features that you want to implement in +Blood? Explain in detail
- Are there any features that you want us to implement in future release?
- Does +Blood have different user roles with own associated features?
- Is there a target launch date for +Blood?
- Is there a budget limit and if so, can we exceed the limit if some implementations are costing more?
- What are the expected compatible devises and operating system versions?
- Can the app be used without internet connection?
- What is the approximate number of users you expect to access +Blood app?

Following this interview, we will write the requirement documentation draft by classifying and organizing them. After completion of the first requirement documentation, we will conduct an open interview session with most if not all stakeholders. This will give the opportunity to the technical team to further grasp the system and user requirements and to note down what requirements it omitted. This allows the technical writers of the system and user requirements and the developers to analyze and prioritize the requirements. Moreover, during this formal meeting, the company will be able to discuss and negotiate on the requirements which will be written in the contract.

Following this new interview session, the team will reclassify and organize the requirements again. Moreover, we will come up with a requirement draft in which the data are categorized according to their priority.

2. Requirement specification

This step requires the team to write down both the system requirement and user requirement. Since, +Blood development is using agile method, customer is still involved in this process. Thus, the requirements can be finalized alongside the customer. The following shows the user requirement:

- Users of +Blood should be able to register and key in their details in the system.
- Registered users should be able to log in the +Blood.
- The users should be able to view the blood group quantities in the blood bank.
- The users should be able to search for a specific blood group donor.
- The users should be able to view a donor profile.
- The users should be able to edit their details.
- The users should be able to contact the donor to get more information.
- The users should be able to request a specific blood group pint.
- The users should be able to add note if they want blood pint from a specific donor.

After defining the user requirements, we can focus and write down the system requirements. The system requirements for the +Blood are written down in the system functionalities section of this report.

3. Requirement validation

This process is essential to verify whether the +Blood's requirements are defined in accordance with the customer needs. Requirement error will cost the company more money; thus, we should ensure that all the requirements are cross-checked and verified by tech team and the customer. The following are aspects that we considered when analyzing the requirements:

- Are any requirements conflicting with each other?
- Is the requirement abiding by the request of the customers?
- Is the requirement defined fully and concisely?
- Can the requirement be implemented?
- Can the requirement be tested?

We implemented a second requirement validation that is prototyping. The prototype is shown in the prototype section of the report.

Since we are using an incremental approach for the development of +Blood, a change management process is implemented. It is a known fact that the system will encounter bugs and will require updates. In order to manage the change requests coming from customers, users, software testers, and developers, we need to have a backup plan to organize and prioritize change request. In this way, the change management process will inform the developers to update the requirements. Using this method ensures that the changes are documented, implemented, verified, and tested. Due to the limit of the budget, this will help to save money since changes can be executed at first attempt. Moreover, the stakeholders of +Blood project will be informed about all the changes being undergone.

System Functionality

Functional requirements:

- +Blood should consist of a register system to allow the user to create a profile; thus, gather relevant information about the user. All the data recorded should be saved in the user database. The basic of +Blood is to create a community of donors and recipients. Consequently, this feature is the starting point to the creation of this active community.

- +Blood should allow the users to log in the system. This implementation is important because it authenticates the users and then direct them to the home page where they can start using the app.

- +Blood should consist of a search bar with a filter. This feature will allow the users to search for donors having a specific blood group. The filter enhances the search by applying a specific criteria like location or blood type A+. Both are essential components of +Blood when a user needs to look up for a match donor within proximity.

- +Blood should consist of a profile page where the users can edit information, change settings, and contact us. The users should be able to update their information that they input during registration since for instance they might have change phone number and the other users will not be able to contact them. The contact us features will allow the users to send the admin messages if they found a bug or had an unpleasant experience with a user online.

- +Blood must allow the users to switch from donors to normal user at any time. This implementation is crucial to keep all donors active. For example, some users may no longer be fit to donate but still want to use the app; thus, they just need to switch to a normal user. This will directly remove the user from the donor list and donor database.

- +Blood should allow users to view the blood bank’s stock. This feature will allow the users to verify whether there is stock of the blood type that they are looking for. If ever they see that the stock is low or none, they can start searching for a donor in the search area.

- +Blood should allow the user to view a specific donor profile. This feature allows the user to know more about the donor and to locate them in case they need blood urgently in case the blood bank stock is empty. On the donor profile, the user will be able to call or message the donor by acquiring his phone number. When clicking on either the message or call button, the phone number of the donor will be displayed on the user's phone in-built phone dial. This implementation will allow the users to contact a donor quickly and ask for relevant information.

- +Blood should allow the users to order a blood pint of any blood type. The users should be able to request a specific donor's blood. This feature is the very essence of the application, that is to order blood and receive it quickly. The users may be in touch with a specific match donor and will like to collect his donation; thus, having an order form with a text area will allow the users to mention their special request.

Non-functional requirements:

- Security
  Ensuring the protection of data in the system is one of the non-functional requirements. The fact that +Blood collects personal details from users shows that the system databases need to have security. +Blood operates most of its features based on the data collected from users and blood bank; thus, it is imperative to ensure that no data are lost or corrupted. We can implement a 2-way verification when users are registering. If ever a user feels threatened by another user who contacted him, the user should be able to contact us and the admin will, after verification, remove the misbehaved user from the system. This implementation promotes a conducive and safe online environment.

- Performance
  The system should be able to response to errors within 0.5 seconds by displaying a pop-up message. Upon successful order of a blood pint, a pop-up message should be seen to inform the user. The blood request should be sent to the admin within 0.8 to 2 seconds. +Blood should handle requests promptly as someone's life might be at risk.

- Availability
  +Blood should be operating round the clock and everyday (24/7). This is because users may request for blood at any time.

- Usability
  A simple UI design should be implemented where the user can navigate easily in the application. The icons and buttons should be self-explanatory. The interface should reduce short term memory load.

- Capacity and scalability
  Rent a cloud service to ensure that we can scale up the capacity of our data storage if needed or scale down if even we do not have lot of users. For example, we can use Microsoft azure cloud services to handle our data. When implementing a cloud service, we will not have to worry about the physical server maintenance.

- Recoverability
  If ever a user has forgot his authentication details, there should be an account recovery system implemented. For instance, send a code to the user's mail that will allow him to change his password in the system after keying the code.

- Regulatory
  The +Blood system should always have active donors in the community. Thus, we can send mails to donors that have been inactive to inform them to switch to seekers if they are not available to donate at the moment. This implementation ensures that our donors list is active and up to date.

- Reliability
  The system should ensure that all information from the donors is reliable. This can be done by cross-checking relevant details of the user using a computer vision technology which can verify the ID cards, passport, and even medical records. This implementation will ensure that all the data of user are relevant, especially his blood group. Moreover, this will enhance the user experience.

Prototype

For the UI, a simplistic design approach was implemented for the +Blood app, where the user will gain access to a diverse set of features that will help them to connect to donors and to request for blood pint. The app's simplistic design includes a standardized theme throughout the pages as well as a navigation bar at the bottom allowing fluid movement across the app. The app's ease of access makes it a powerful tool that anyone within the society can take advantage of to get blood in emergency.

A peach color has been chosen for the main parts of the application based on the psychology of colors. This color has been chosen because psychologically, it is said to motivate, to generate a positive attitude, and in general to create an enthusiasm for life.
The prototype of +Blood application was developed using the 8 Golden Rules of Interface Design.

The prototype can be accessed using the following link:
https://www.figma.com/file/WnoJ03RMoCZDXak0mkVTzJ/Blood-App?node-id=0%3A1

The following consists of pictures and description of the key features of the +Blood application.

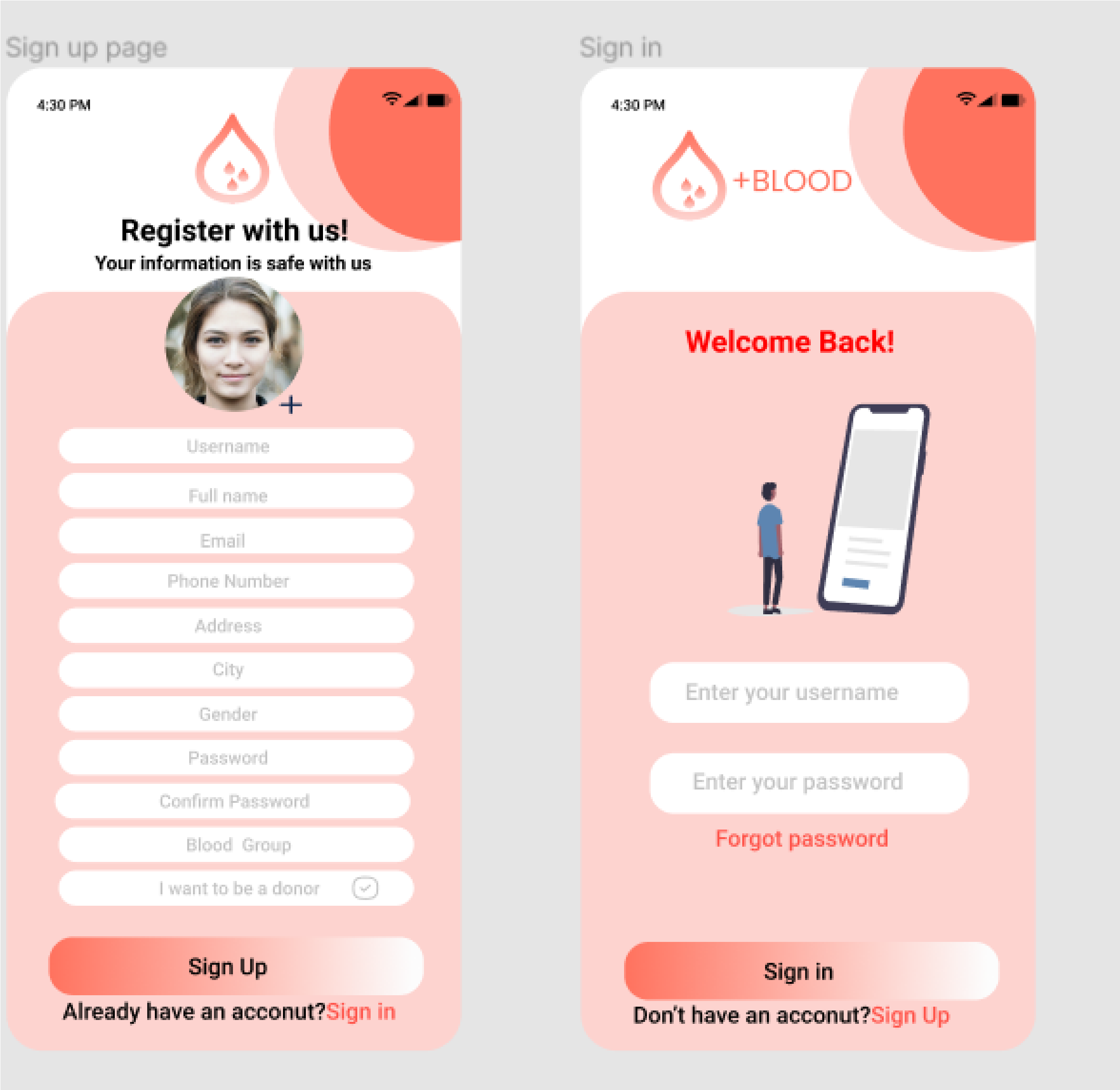
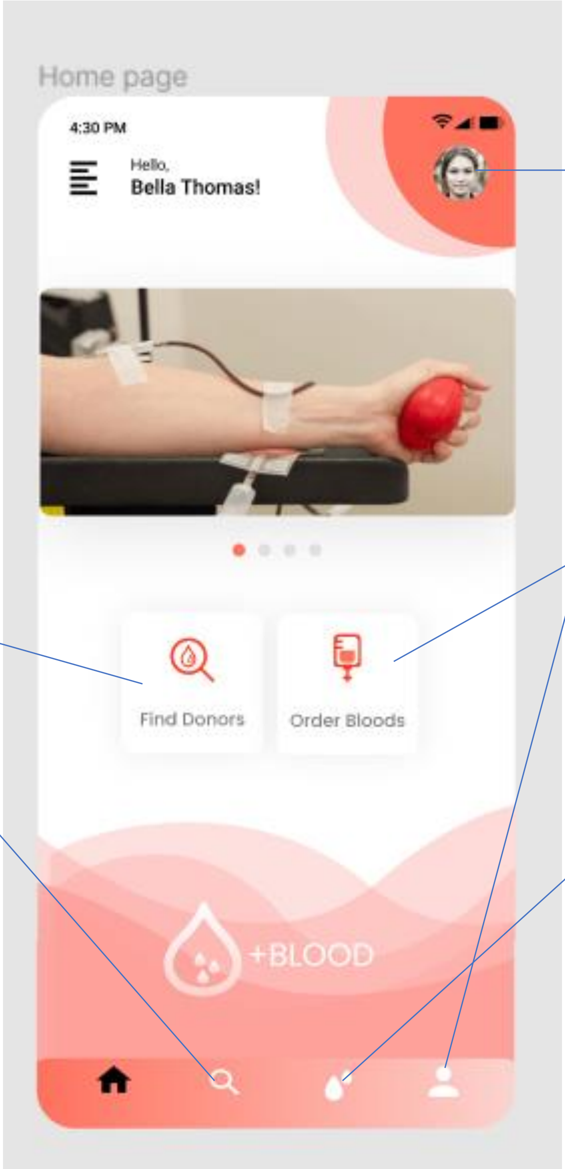


Figure 1

Figure 1 depicts the sign-up page where the users can key in their details to register in the +Blood app. The sign in page is where the users key in their authentication details to access their account and log into the system.



When clicking on the profile picture or the profile icon, the user will be directed to the profile page.

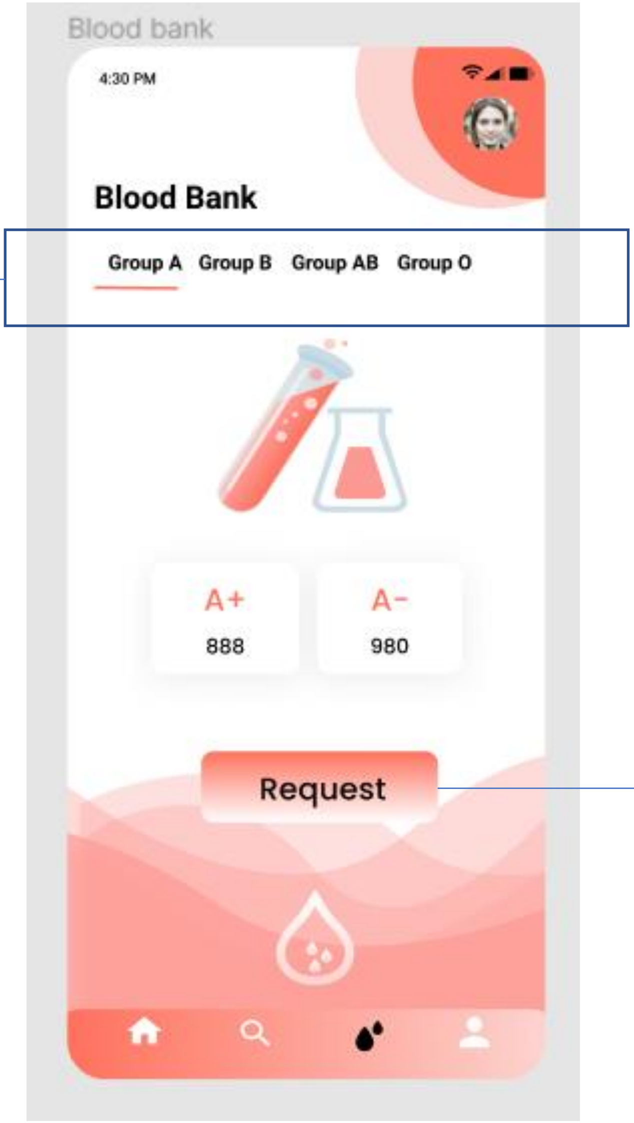
When clicking on either Find Donors or the search icon, the user will be directed to the search page.

When clicking on the Order Bloods button, the user will be directed to the blood order page.

When clicking on the blood icon, the user will be directed to the blood bank stock page.

Figure 2

Figure 2 shows the home page which is the first page displayed when you successfully sign. The navigation menu at the bottom is standard for all the other pages. The profile picture at the right top is also accessible in all the other pages.

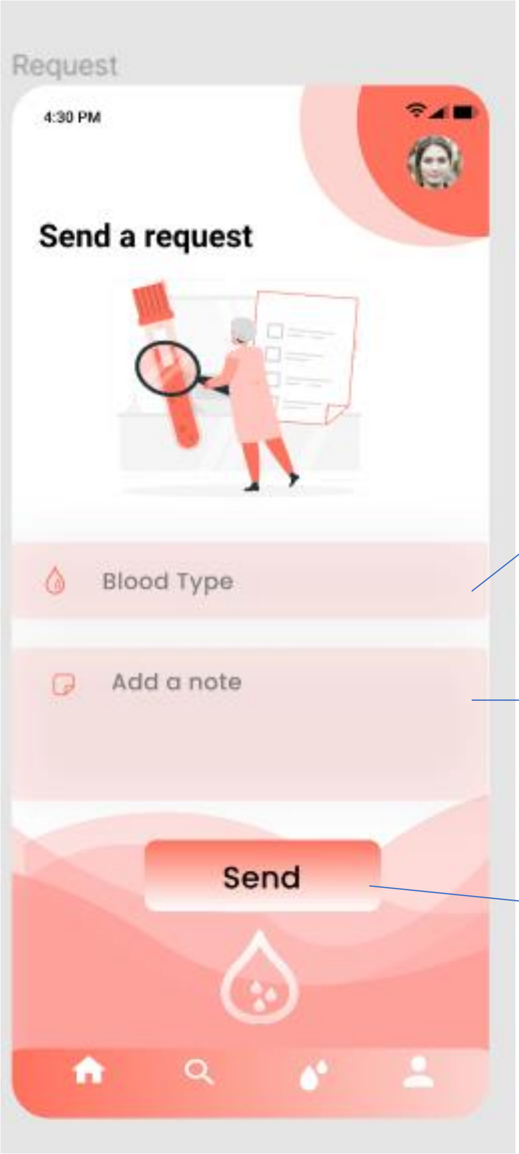


This feature allows you to switch between blood groups to display their respective stock.

By clicking on the request button, the user will be directed to the blood order page.

Figure 3

Figure 3 shows the blood bank stock page which allows the user to view the stock of respective blood group.



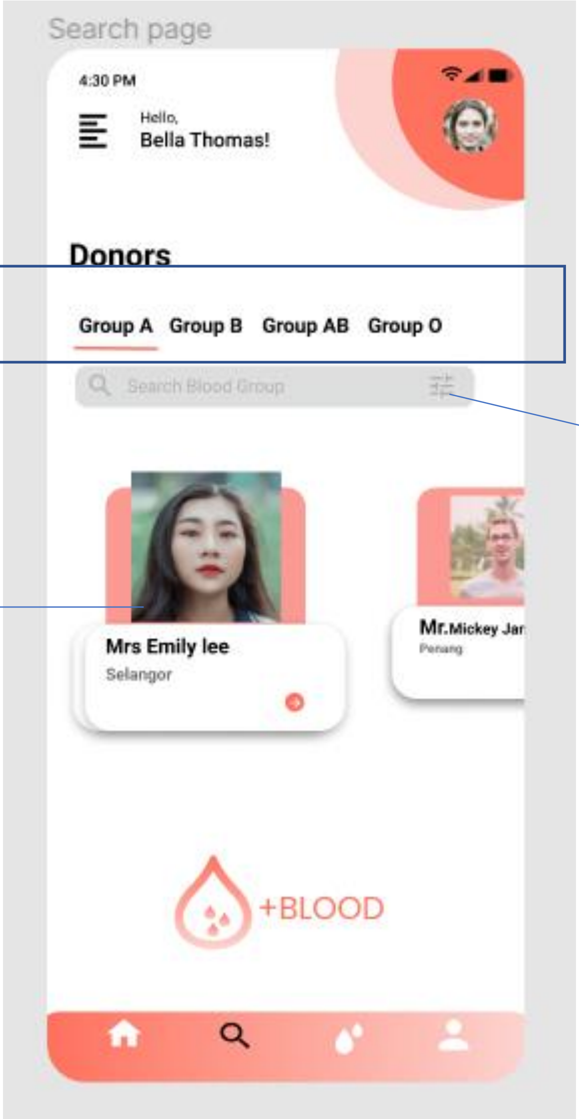
Field for users to input the specific blood type that they need.

Text area for users to add a special request for example a specific donor name to receive his donation.

By clicking on this button, the blood pint order request is sent to the admin.

Figure 4

Figure 4 shows the blood order page where the users can send request promptly to the admin to get a pint of blood.



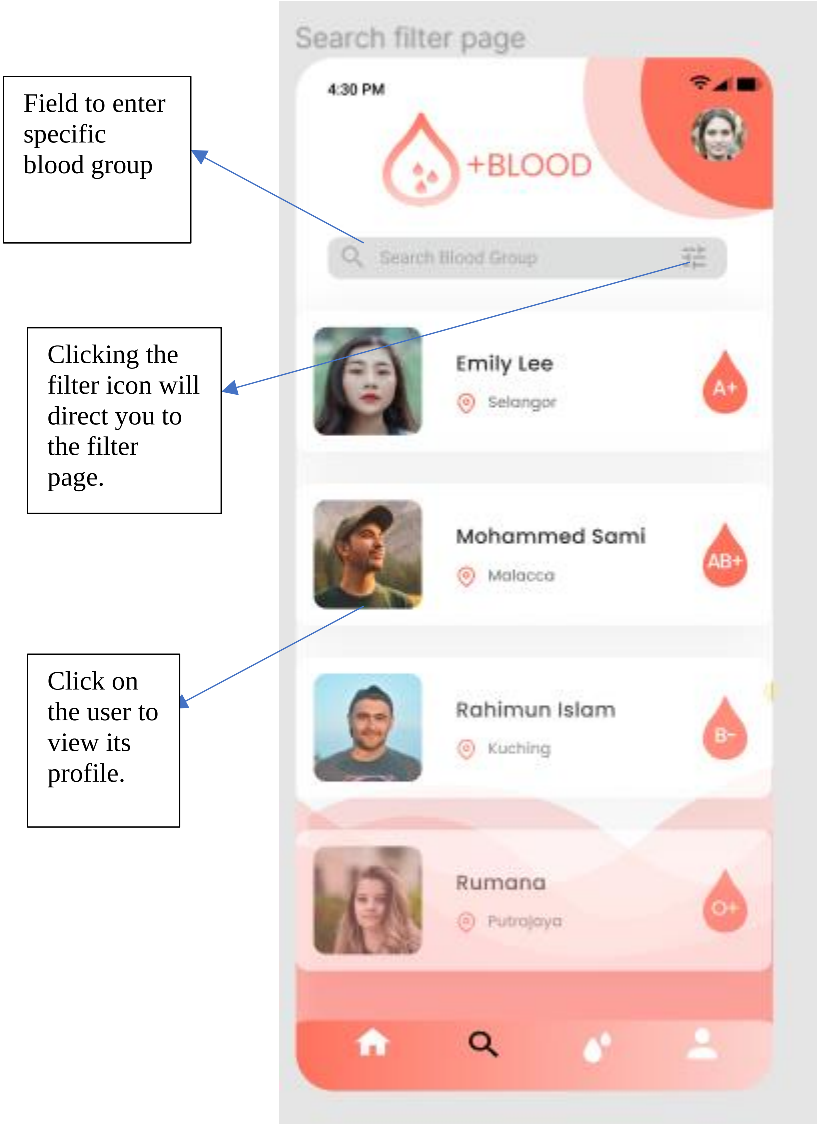
The users can view which donor profiles by blood group by switching between these tabs.

By clicking the search tab, the user will be directed to the search filter page.

By clicking on the profile, the user will be directed to the donor profile.

Figure 5

Figure 5 shows the search page, by default the group A donor profiles are displayed. The users can select which blood group user profiles they want to view by switching between the tabs.

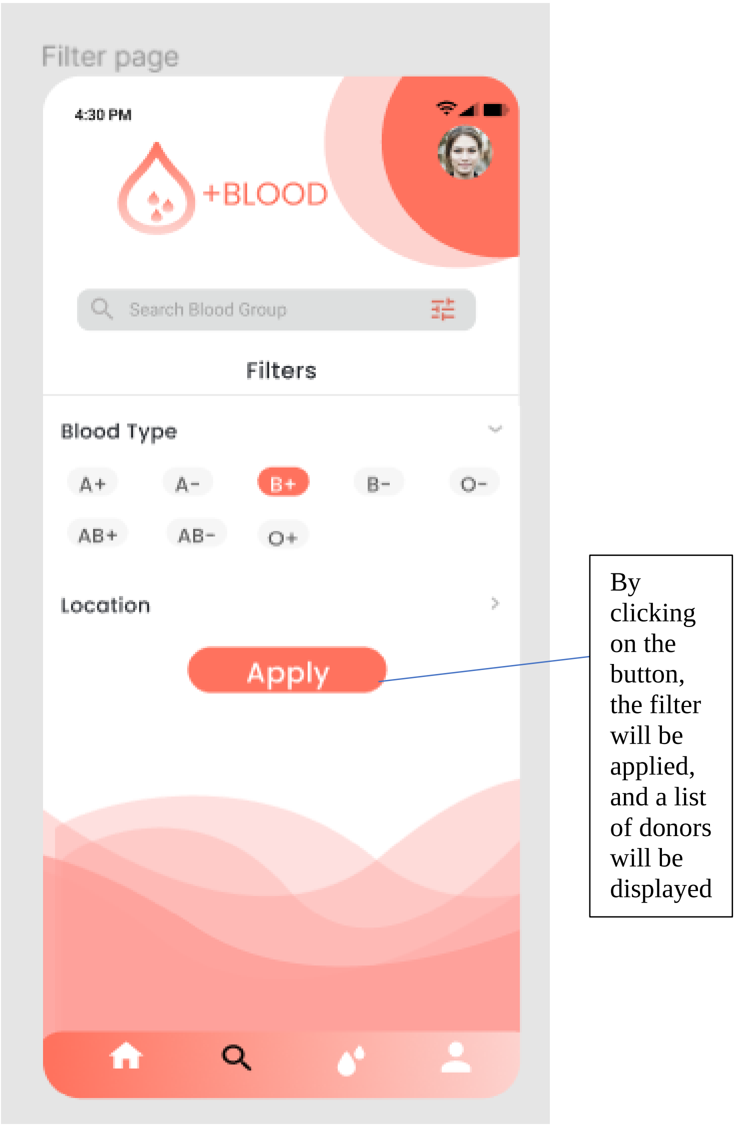


Field to enter specific blood group

Clicking the filter icon will direct you to the filter page.

Click on the user to view its profile.

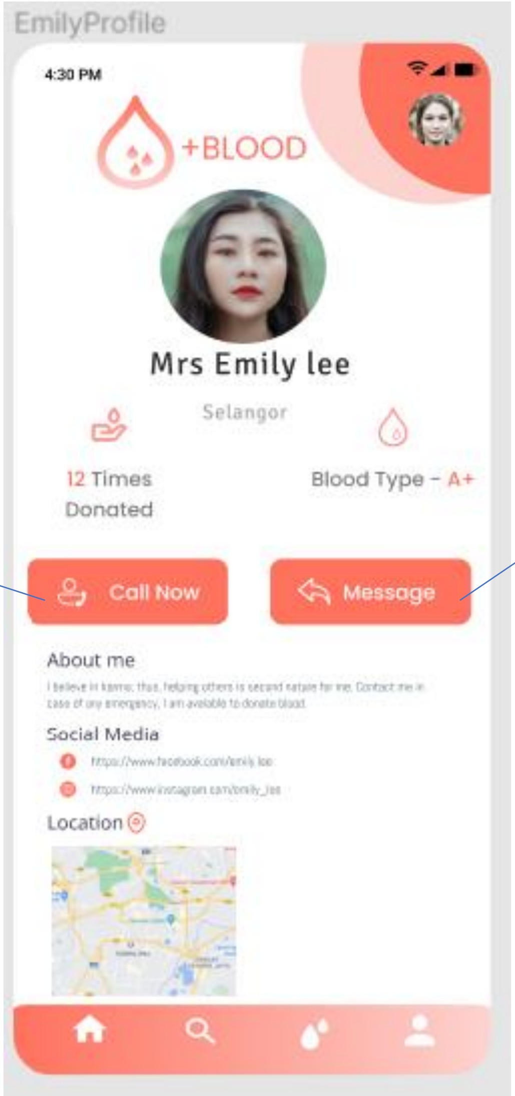
Figure 6



By clicking on the button, the filter will be applied, and a list of donors will be displayed

Figure 7

Figure 6 shows the search filter page while figure 7 shows the filter page. Both pages work together. By default, the search filter page will display all donors irrespective of blood group. When the user applies a specific filter, the list of donors will change according to the selected filters.



About me

I believe in karma; thus, helping others is second nature for me. Contact me in case of any emergency. I am available to donate blood.

Social Media

https://www.facebook.com/emily lee

https://www.instagram.com/emily_lee

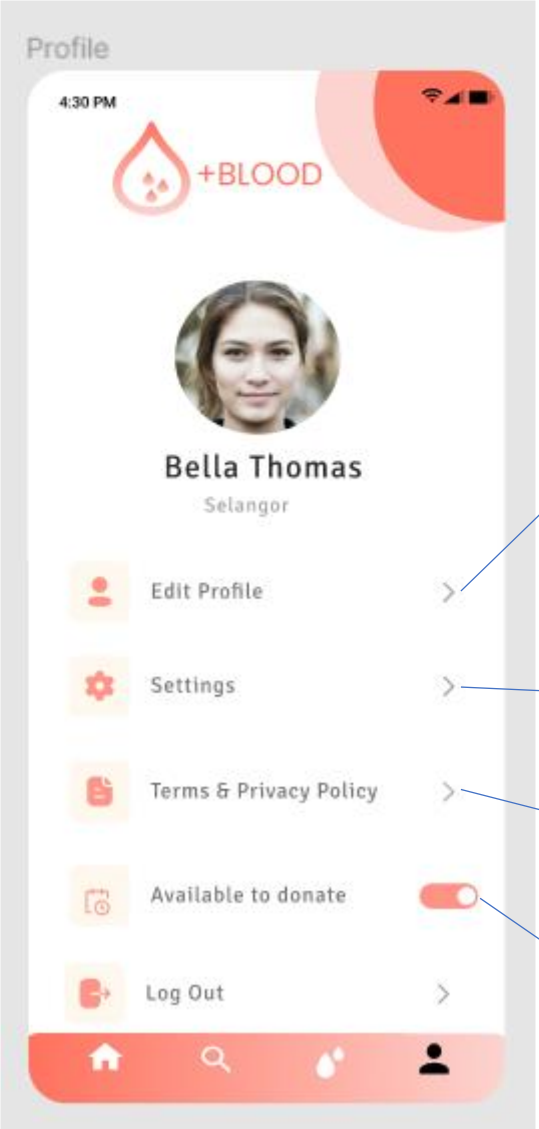
Location

By clicking this, the phone number of the user will be displayed on your phone’s call dial system where the user can proceed to call the donor.

By clicking this, the phone number of the user will be displayed on your phone’s call dial system where the user can proceed to message the donor.

Figure 8

Figure 8 displays the donor profile that the users selected from the donor list. The application does not have an in-built call and messaging system. Thus, by clicking on the buttons, the phone number of the donor will be displayed on the users’ phone call dial system where they can decide to call or message the donor.



Click to edit your details or to change other sensitive information e.g password.

Select to change setting preferences.

On click, the app will display the terms and privacy policy.

User can tag or untag as donor with this feature.

Figure 9

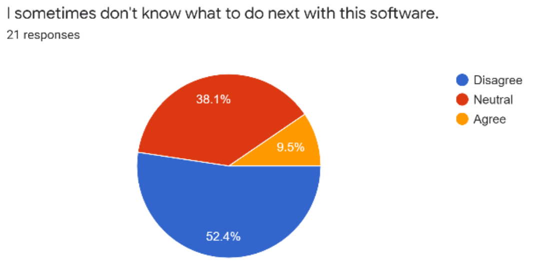
Figure 9 shows the user profile where the user can have access to distinctive features. The users can check or uncheck the donor tag to decide whether they want to be displayed in the donor list; thus, sharing their profile on the +Blood app.

Prototype analysis

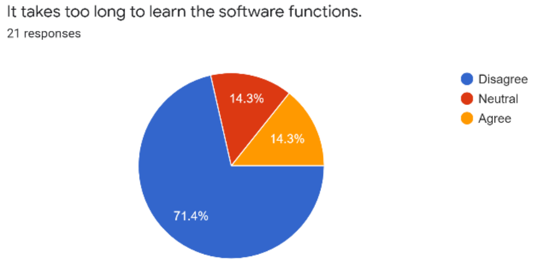
In order to improve +Blood, we have conducted a survey. The survey was implemented using the Software Usability Measurement Inventory which is a proven approach to collect users' feedback and to measure software quality from the end user's point of view. The link to the SUMI survey is: https://forms.gle/zmWG2Ki9zJRmL8xd7.

According to the statistics, 38.1 and 61.9 are the percentages of male and female respondents. The age of respondents was between 18 and 51 with a mean of 21.8 and a mode of 21. The responses mainly come from a younger generation consisting mainly of students. 47.6% of the respondents find the application of +Blood to be very important to them while 38.1% finds it important. Most of the respondents (57.1%) have a decent understanding of the software functions.

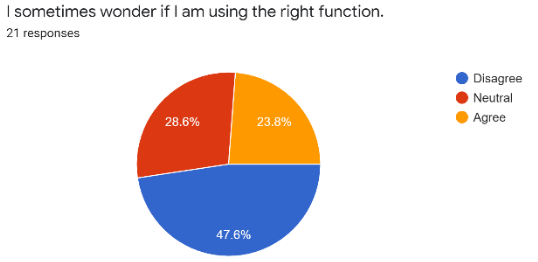
Questions from both categories were shortlisted and a summary of the statistics is as follows:


I sometimes don't know what to do next with this software.

21 responses

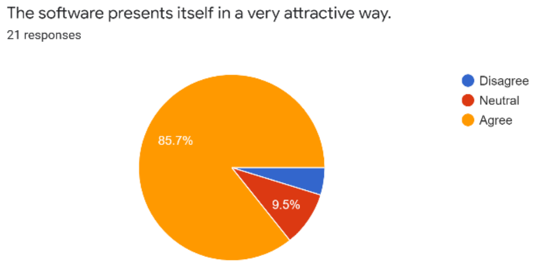

It takes too long to learn the software functions.

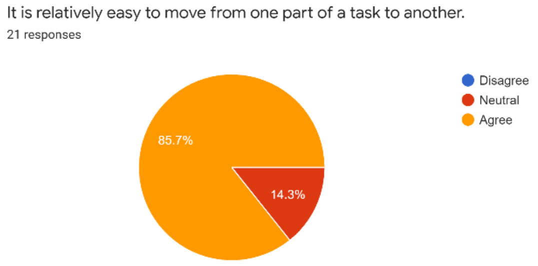
21 responses

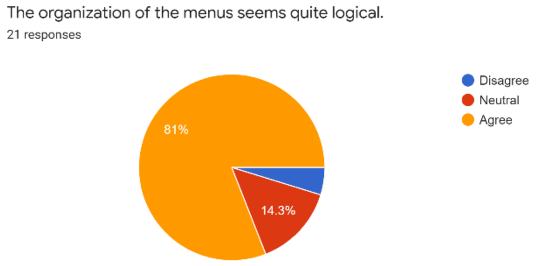

I sometimes wonder if I am using the right function.

21 responses

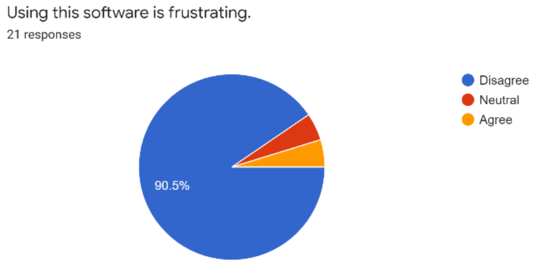
Considering the above stats, we can see that some users may not understand how to user the app. Thus, we may need to implement a guide to explain all the features after registration. We can even make a guide video.







The above diagrams shows that there is an overall satisfaction of the user interface and that +Blood promotes user experience.



From the above diagram, we can notice that most users are enjoying their experience with +Blood.

Analysis of the subjective questions:

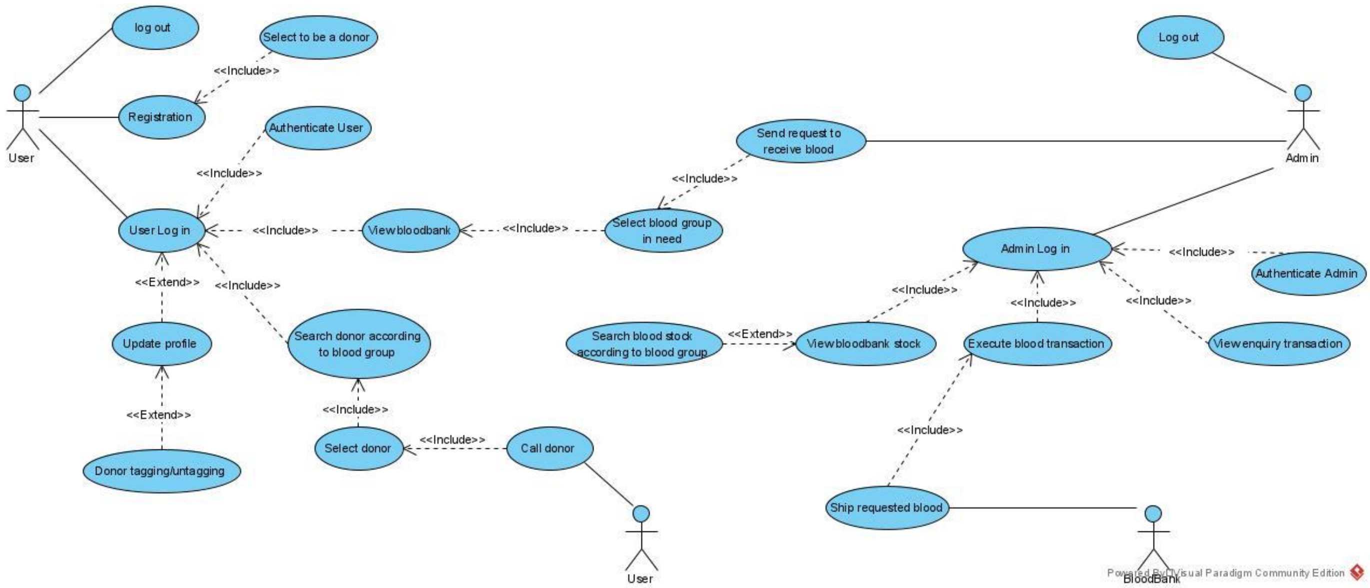
The first question is what do you think is the best aspect of this software, and why?
The responses are incredibly positive, most users find the search function to find a match donor. The users mentioned about the easiness to order a blood pint and to connect with donors. Moreover, they appreciated the UI of the application especially the organisation of the features and the calm and friendly colour scheme. Some users mentioned that the technology being used is familiar and easy to understand.

The second question is what do you think needs most improvement, and why?
For this question, some users mentioned about that we need to focus on keeping the security and confidentiality of donors and suggested to reduce the donors' details displayed. While other users mentioned to add more details about the health history of the donor. The users also pointed out that the application needs to have a built-in chat and call system. Moreover, one user asked to improve on the blood order system. Since, when a user clicks on request button found in the blood group stock page, it is directed to the order page. However, the user needs to key in again the blood type despite clicking the request button from a specific blood type stock page.

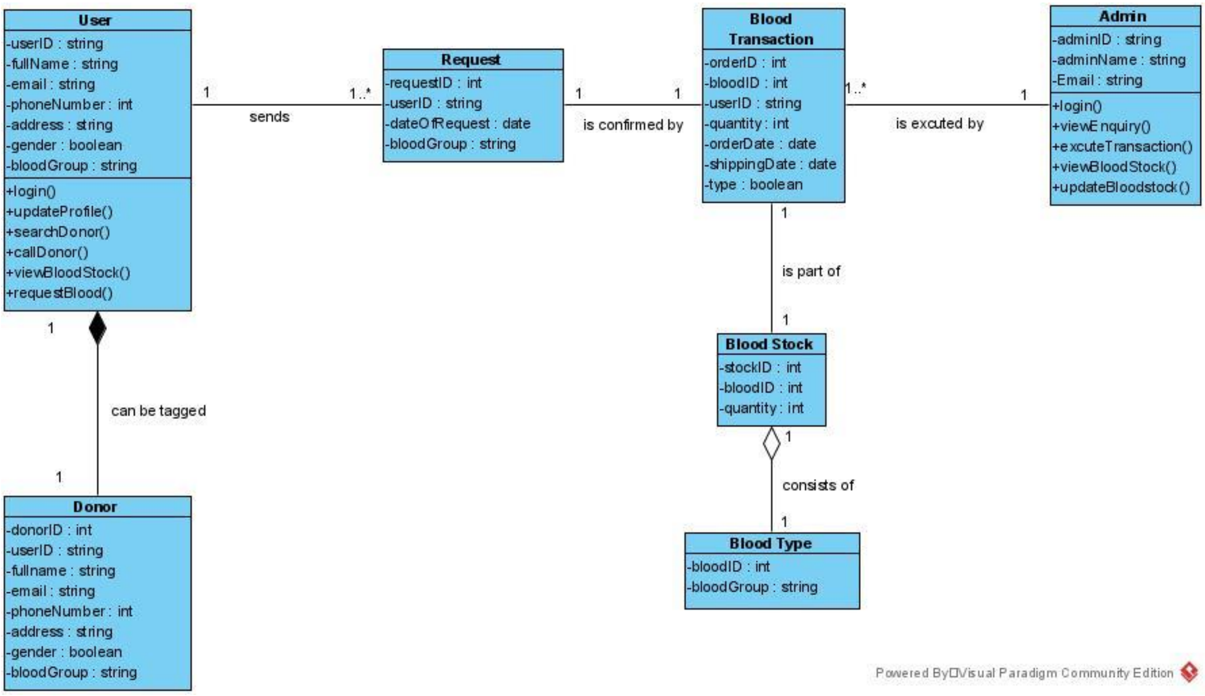
Considering all the responses, we can acknowledge that the acceptance of the +Blood is remarkably high and most of the users found its usage and functions promote user experience and eases the process of getting a blood pint especially during emergencies.

UML Diagrams

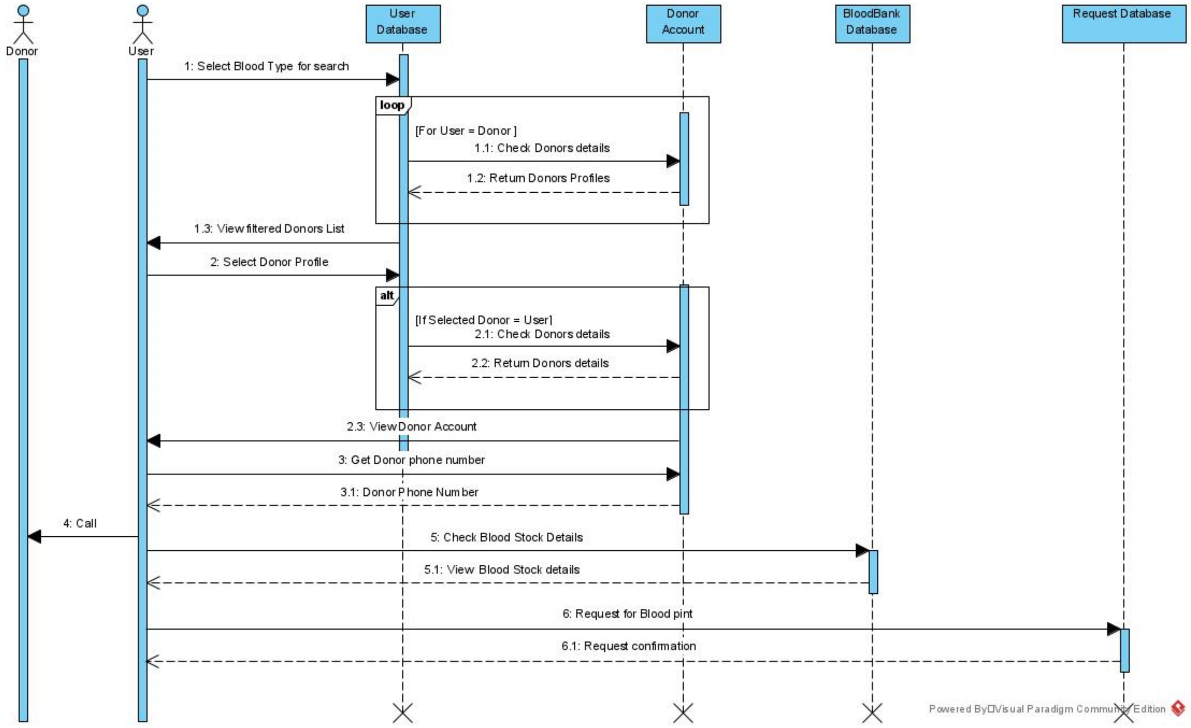
Use Case Diagram

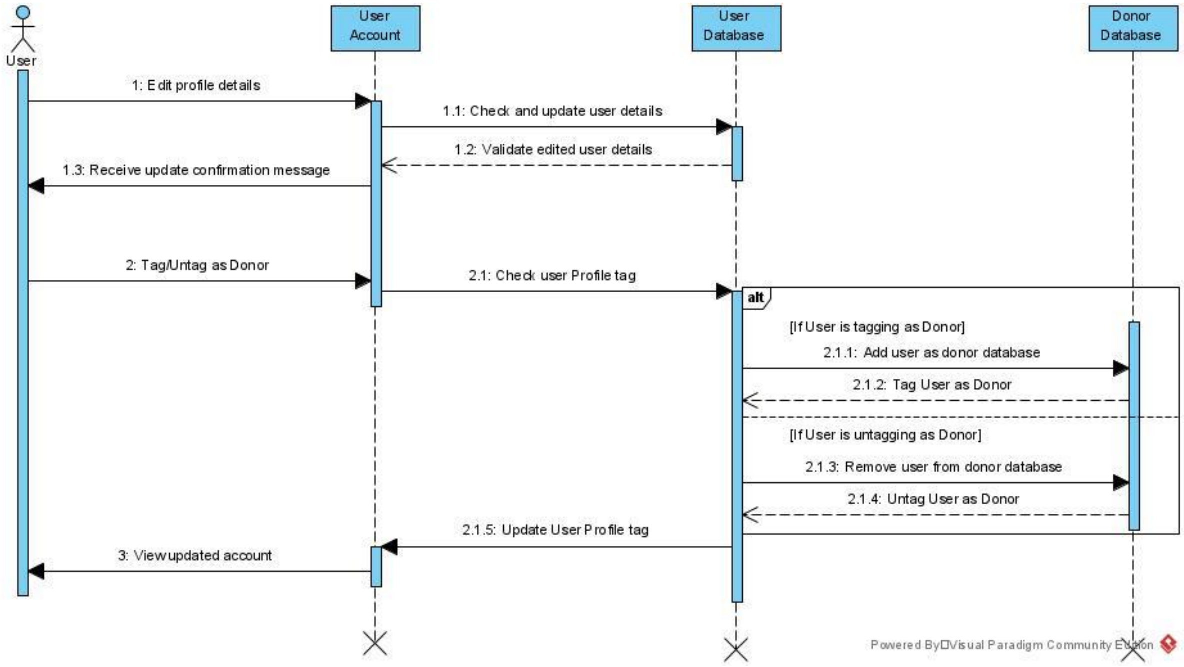


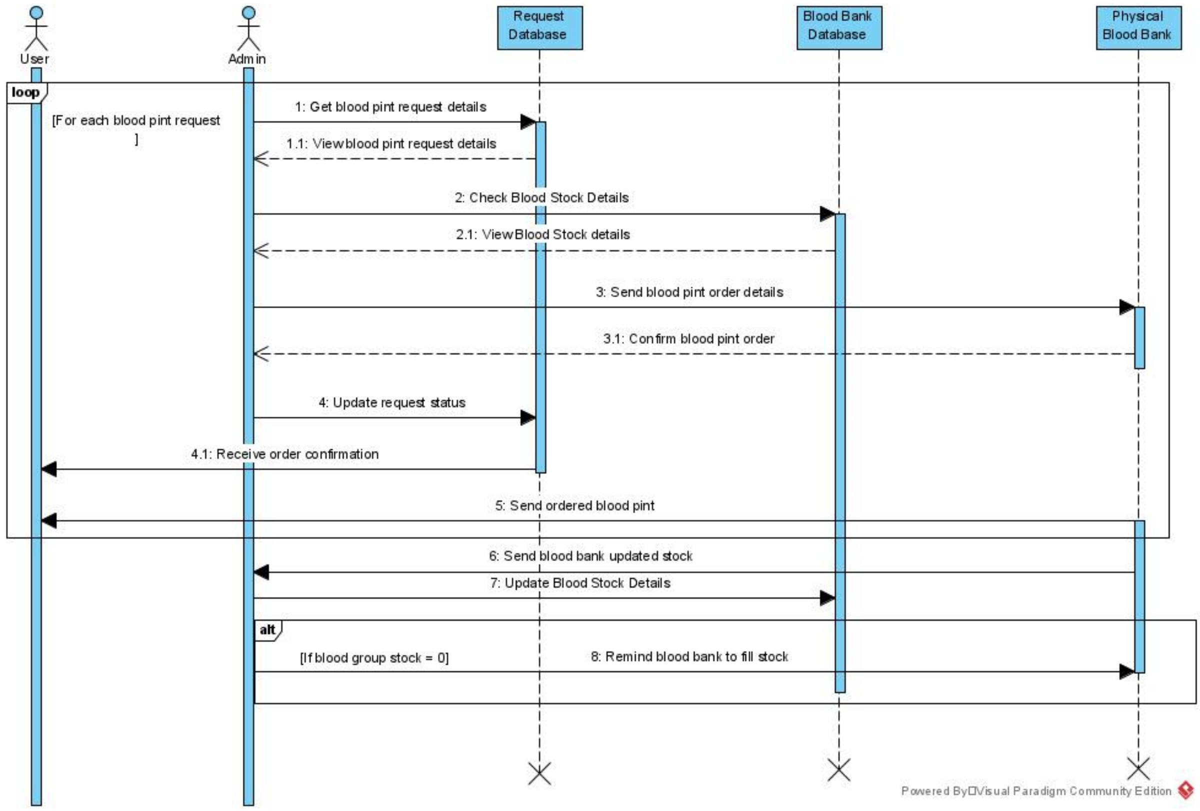
Class Diagram



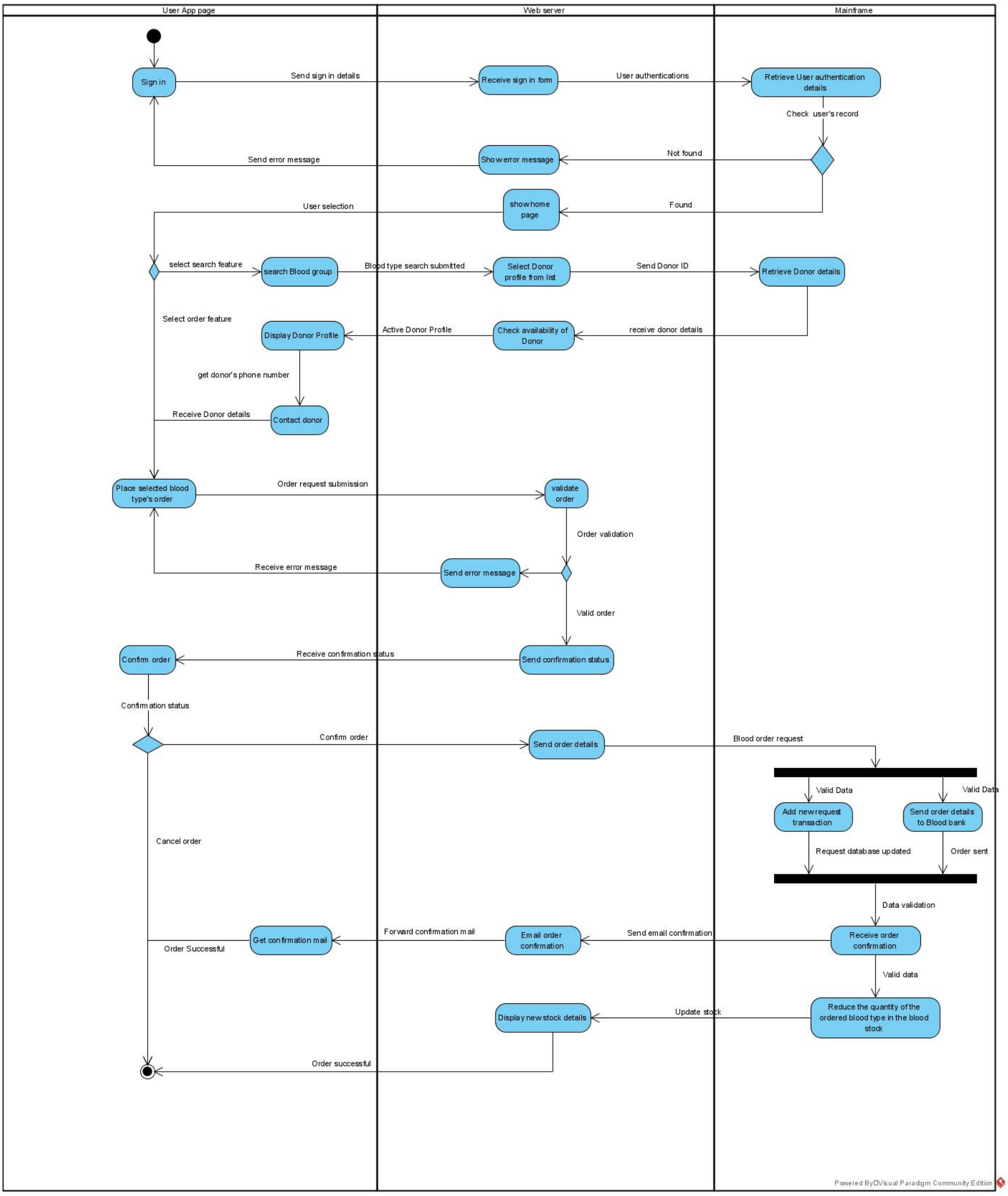
Sequence Diagrams


Donor
User
User Database
Donor Account
BloodBank Database
Request Database
1: Select Blood Type for search
loop
[For User = Donor]
1.1: Check Donors details
1.2: Return Donors Profiles
1.3: View filtered Donors List
2: Select Donor Profile
alt
[If Selected Donor = User]
2.1: Check Donors details
2.2: Return Donors details
2.3: View Donor Account
3: Get Donor phone number
3.1: Donor Phone Number
4: Call
5: Check Blood Stock Details
5.1: View Blood Stock details
6: Request for Blood pint
6.1: Request confirmation
Powered By Visual Paradigm Community Edition


User
User Account
User Database
Donor Database
1: Edit profile details
1.1: Check and update user details
1.2: Validate edited user details
1.3: Receive update confirmation message
2: Tag/Untag as Donor
2.1: Check user Profile tag
alt
[If User is tagging as Donor]
2.1.1: Add user as donor database
2.1.2: Tag User as Donor
[If User is untagging as Donor]
2.1.3: Remove user from donor database
2.1.4: Untag User as Donor
2.1.5: Update User Profile tag
3: View updated account
Powered By Visual Paradigm Community Edition


User
Admin
Request Database
Blood Bank Database
Physical Blood Bank
loop
[For each blood pint request]
1: Get blood pint request details
1.1: View blood pint request details
2: Check Blood Stock Details
2.1: View Blood Stock details
3: Send blood pint order details
3.1: Confirm blood pint order
4: Update request status
4.1: Receive order confirmation
5: Send ordered blood pint
6: Send blood bank updated stock
7: Update Blood Stock Details
alt
[If blood group stock = 0]
8: Remind blood bank to fill stock
Powered By Visual Paradigm Community Edition

Activity Diagram



Conclusion

The implementation of the +Blood project was meticulously discussed throughout the report by explaining the software process models, requirement engineering stages, software objective and scope, software requirements, software testing, and prototype analysis. I hope that this report gives you a better grasp on the +Blood application implementation.