Software Requirements Specifications

ClaretCorridor

*Submitted By,*

Billal Hossain Nadia Jahan

ASH2125018M BFH2125017F

Year: 02, Term: 02 Year: 02, Term: 02

Institute of Information Technology Institute of Information Technology

*Submitted to,*

*Dipok Chandra Das*

Assistant Professor,

Institute of Information Technology

Noakhali Science and Technology University

[dipok.iit@nstu.edu.bd](mailto:dipok.iit@nstu.edu.bd)

1. [Introduction 3](#_bookmark0)
   1. [Problem Statement 3](#_bookmark1)
   2. [Purpose 3](#_bookmark2)
   3. [Project Scope 3](#_bookmark3)
   4. [Glossary 4](#_bookmark4)
   5. [References 4](#_bookmark6)
   6. Overview……………………………………………………………………..…………………..4
2. [Stakeholders and Characteristic 4](#_bookmark7)
   1. [Definition 4](#_bookmark8)
   2. [Classification based on Interest 4](#_bookmark9)
   3. [Classification based on Elicitation 5](#_bookmark10)
3. [Design and Implementation Constrains 6](#_bookmark11)

3.1 [Language 6](#_bookmark12)

* + 1. [HTML 6](#_bookmark13)
    2. [CSS 6](#_bookmark14)
    3. Bootstrap……………………………………………………………………………6
    4. [JavaScript 7](#_bookmark15)
    5. [PHP 7](#_bookmark16)
  1. [Server-Side Technology 7](#_bookmark12)

3.2.1 [Database Server 7](#_bookmark13)

1. [Requirement Specification 7](#_bookmark19)
   1. [Functional Requirement 7](#_bookmark20)
      1. [User registration and login 8](#_bookmark21)
      2. [Communication 8](#_bookmark22)
      3. [Posting 8](#_bookmark23)
      4. [Blood Bank Information 9](#_bookmark24)
      5. [Update Profile 9](#_bookmark25)
      6. [Send Notification 9](#_bookmark26)
      7. [Last Update Donation Time 10](#_bookmark27)
   2. [Performance Requirement 10](#_bookmark33)
      1. [Speed and Latency Requirements 10](#_bookmark34)
      2. [Safety Critical Requirements 10](#_bookmark34)
   3. [Maintainability and Supportability 11](#_bookmark40)
      1. [Maintenance Requirements 11](#_bookmark41)
      2. [Supportability Requirements 11](#_bookmark42)
   4. [Security Requirements 11](#_bookmark40)
      1. [Access Requirements 11](#_bookmark42)
      2. [Integrity Requirements 11](#_bookmark42)
2. [Requirement Engineering Process 11](#_bookmark53)
   1. [Requirement Elicitation Techniques 12](#_bookmark54)
      1. [Hold Interviews 12](#_bookmark55)
      2. [Perform Document Analysis 12](#_bookmark56)
      3. [Distribute Questionaries 12](#_bookmark55)
      4. [Sample Requirements Collection 12](#_bookmark56)
   2. [Requirement Validation 14](#_bookmark59)
      1. [Review the Requirements 14](#_bookmark60)
      2. [Simulate the Requirements 14](#_bookmark61)
3. [Use Case Diagram 15](#_bookmark63)
4. [Use Case Description 16](#_bookmark65)
5. [Activity Diagram 31](#_bookmark89)
6. [Sequence Diagram 44](#_bookmark113)
7. [Swimlane Diagram 56](#_bookmark114)

**List of Figure**

[Figure 1: Use-case Diagram 15](#_bookmark64)

[Figure 2: Registration 31](#_bookmark90)

[Figure 3: Log in 32](#_bookmark91)

[Figure 4: Authentication 33](#_bookmark92)

[Figure 5: Communication 34](#_bookmark93)

[Figure 6: Post 35](#_bookmark94)

[Figure 7: Blood Bank Information 36](#_bookmark95)

[Figure 8: Update Profile 37](#_bookmark96)

[Figure 9: Notification 38](#_bookmark97)

[Figure 10: Last Update Donation Date 39](#_bookmark98)

[Figure 11: Give Reward 40](#_bookmark99)

[Figure 12: Free Medical Service 41](#_bookmark100)

[Figure 13: Update Password 42](#_bookmark101)

[Figure 14: Log Out 43](#_bookmark102)

[Figure 15: Registration 44](#_bookmark103)

[Figure 16: Log in 45](#_bookmark104)

[Figure 17: Delete User 46](#_bookmark105)

[Figure 18: Communication 47](#_bookmark106)

[Figure 19: Post for Blood 48](#_bookmark107)

[Figure 20: Blood Bank Info 49](#_bookmark108)

[Figure 21: Update Profile 50](#_bookmark109)

[Figure 22: Sent Notification 51](#_bookmark110)

[Figure 23: Last Update Donation Date 52](#_bookmark111)

[Figure 24: Give Reward 53](#_bookmark112)

[Figure 25: Update Password………………………………………………………………...54](#_bookmark112)

[Figure 26: Log Out …………. ……………………………………………………………..55](#_bookmark112)

[Figure 27: Login……….. …………………………………………………………………..56](#_bookmark112)

[Figure 28: Delete User ……………………………………………………………………...57](#_bookmark112)

[Figure 29: Communication ………….…………………………………………..………….58](#_bookmark112)

[Figure 30: Post For Blood……….………………………………………………………….59](#_bookmark112)

[Figure 31: Blood Bank Info …………….…………………………………………………..60](#_bookmark112)

[Figure 32: Update Profile……….……………………………………………………….….61](#_bookmark112)

[Figure 33: Sent Notification …………………………………………………………….….62](#_bookmark112)

[Figure 34: Last Update Donation Date………………………………………………….….63](#_bookmark112)

[Figure 35: Post Get Reward…….……………………………………………….………….64](#_bookmark112)

[Figure 36: Free Medical Service …………….…………………………………….……….65](#_bookmark112)

[Figure 37: Update Password………….……………………………………………….…….66](#_bookmark112)

[Figure 38: Log Out……….. …………………………………………………………..…….67](#_bookmark112)

**List of Tables**

[Table 1: Requirement Collection 01……………………………………………….………..5](#_bookmark90)

[Table 2: Requirements Collection 02………………………………..……………………..10](#_bookmark91)

[Table 3: Requirements Collection 03 ……………….…….………………………………..11](#_bookmark92)

[Table 4: Registration………………………………………………………………………..16](#_bookmark93)

[Table 5: Log in…….…………….…………………………………………………………..17](#_bookmark94)

[Table 6: Delete Users and Blood Bank Account……….…….……………………………..18](#_bookmark95)

[Table 7: Communication……………..……………………………………………………..19](#_bookmark96)

[Table 8: Post for Blood 21](#_bookmark97)

[Table 9: Blood Bank Information 22](#_bookmark75)

[Table 10: Update Profile 23](#_bookmark76)

[Table 11: Send Notification 24](#_bookmark97)

[Table 12: Last Update Donation Date 25](#_bookmark75)

[Table 13: Give Rewards 26](#_bookmark76)

[Table 14: Free Medical Service 27](#_bookmark97)

[Table 15: Update Password 28](#_bookmark75)

[Table 16: Log Out 29](#_bookmark76)

1. **Introduction:**

The ClaretCorridor is a comprehensive and user-friendly software solution designed to streamline and enhance the operations of blood banks, donors, recipients, and medical service providers. This system addresses the critical needs of managing blood donations, transfusions, and related processes to ensure efficient and reliable blood supply management

* 1. **Problem Statement:**

"ClaretCorridor" is an innovative blood bank app that redefines the act of blood donation. Through seamless integration with the blood bank, donors earn points for their contributions, which can be redeemed for complimentary medical services. This user-friendly platform not only facilitates life-saving donations but also empowers donors to access vital healthcare resources at no cost.

* 1. **Purpose**

The "ClaretCorridor" project aims to streamline the blood donation and distribution process through an online platform. Its purpose is to facilitate easy access to blood for recipients by connecting them with nearby donors and blood banks. The project encourages regular blood donation through a rewards system and ensures efficient tracking and secure management of blood inventory. By leveraging digital technologies, it enhances accessibility and user experience while fostering a sense of community among donors, recipients, and blood banks. Ultimately, the goal is to save lives by creating a responsive and technologically advanced blood donation ecosystem.

* 1. **Project Scope:**

"The ClaretCorridor project is designed to enhance blood bank services, providing a comprehensive platform for blood donors and recipients. The system goes beyond individual blood banks, offering a versatile structure that can be implemented in multiple blood bank facilities. Users can seamlessly search for and donate blood, access information about the blood bank, and enjoy various features.

* **Multi-Blood Bank Support:** The system is not limited to a specific blood bank, allowing its structure to be employed across various blood bank facilities.
* **Efficient Blood Search:** Users can swiftly search for available blood types, making the process of finding and donating blood a quick and straightforward experience.
* **Remote Blood Request:** Donors and recipients can remotely request or offer blood donations, increasing accessibility and convenience.
* **Users Profile and Dues:** Users have personalized profiles, enabling them to monitor their donation history and check for any outstanding dues or obligations.
* **Flexible Donation Management:** Donors can manage their donations, extend the time period of availability, and view their contribution history.
* **Request for New Blood Type:** Users have the ability to send requests for adding new blood types to the blood bank, ensuring a diverse and comprehensive range of available blood.

**1.4 Glossary**

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

HTML – Hyper Text Markup Language

JS – JavaScript

CSS – Cascading Style Sheets

PHP – Hypertext Preprocessor

XML – Extensible Markup Language

SRS – Software Requirement Specification

UI – User Interface

API – Application Programming Interface

MB – Megabytes

**1.5 References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements  Specifications.* IEEE Computer Society, 1998.

**1.6 Overview**

The ClaretCorridor Blood Bank Project is a comprehensive initiative aimed at creating a robust platform for blood donation, storage, and distribution, with a unique twist that incentivizes and rewards donors for their life-saving contributions. The initiative also promotes community engagement through a dedicated platform for sharing donation experiences

1. **Stakeholders and Characteristics:**
   1. **Definition:** Stakeholders are individuals, groups, or enities that have a vested interest or concern in the success, outcomes, or activities of a particular project, organization, or system. In the context of your "ClaretCorridor" blood bank project, stakeholders would include anyone who is directly or indirectly affected by or can affect the project.
   2. **Classification Based on Interest:**

* **Primary Stakeholders:** These are individuals or groups directly impacted by the project and have a significant interest and influence on its success. In the case of ClaretCorridor project, primary stakeholders could include donors, recipients and actors.
* **Secondary Stakeholders:** Secondary stakeholders have an interest in the project but may not have as much direct influences as primary stakeholders. This category might include volunteers, community members, and non-profit organizations.
* **External Stakeholders:** External stakeholders are individuals or entities outside the immediate project team or organization but still have an interest in the project. This could encompass government health agencies, regulatory bodies.
* **Supporters:** These stakeholders may not be directly impacted, but they support the project’s goal and objectives. Supporters could include community members, businesses or organizations.
* **Operational Stakeholders:** Those involved in the day-to-day operations of the app, such as staff, volunteers are crucial for the project’s successful implementation.
  1. **Classification Based on Elicitation:**

Elicitation techniques refer to a set of methods or approaches used to gather information, requirements, or feedback from individuals or groups. These techniques are employed to extract knowledge, preferences, expectations, or opinions from stakeholders in order to understand their needs and perspectives.

1. **Directly Engaged Stakeholders:**

* Interview Participants: Donors, recipients, and medical professionals could actively participate in interviews to provide detailed insights into their experiences, expectations, and needs.
* Workshop or Focus Participants: Volunteers, community members, and nonprofit organizations might participate in collaborative sessions to share ideas and provide input on community engagement strategies.

1. **Document Contributors:**

* Document Providers: Government health agencies and regulatory bodies may contribute to document analysis by providing relevant guidelines, regulations, and policies that impact the blood bank project.

1. **Prototyping Collaborators:**

* Prototyping Feedback Providers: Donors and recipients might engage with prototypes or simulations to provide feedback on the user experience and identify any concerns.

1. **Creative Input Stakeholders:**

* Brainstorming Participants: Community members, volunteers, and nonprofit organizations could participate in brainstorming sessions to generate creative ideas for outreach and awareness campaigns.

1. **Design and Implementation Constraints**

In order to ensure the project's success, we used design and implementation limitations. It can also refer to a tool that enables testers and developers to view and interact with the user interface (UI) components of an application.

**3.1 Language**

User interface design, usually known as UI Design, is the visual organization of the parts of a website or technological product that a user could interact with. In other words, it is the visual layout of a website. On the other hand, the code that enables a computer program or application to run and cannot be viewed by a user is referred to as the back end. The back end of a computer system is where the majority of data and operating syntax are kept and accessed. Typically, the code is comprised of one or more programming languages.

**3.1.1 HTML**

HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. Precisely, the coding that organizes a web page's content is called HTML (Hypertext Markup Language). With the help of HTML, you can tell a web page whether it should be recognized as a paragraph, list, heading, link, image, multimedia player, form, or any other of the many other components that are now supported, or even a new element that you design. It is the programming language for formatting web pages that is widely accepted. Small and medium-sized businesses are the main users, as they do not actually require extensive functionality on their websites. The option to utilize HTML to design the structure of my web pages was made since it is free, works with all browsers on the client's machine, and is simple to use and understand.

**3.1.2 CSS**

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS specifies how items should be shown in various media, including speech, paper, screens, and other media. One of the fundamental languages of the open web, CSS is defined by the W3C (World Wide Web Consortium) specification and is supported by all major browsers.

**3.1.3 Bootstrap (Front-end framework)**

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It includes optional JavaScript extensions along with HTML and CSS-based design templates for navigation, buttons, forms, and other interface elements. It only addresses front-end development, unlike many web frameworks. Along with CSS, Bootstrap would be utilized to create the application's styling. Bootstrap is important in the application for the following reasons:

∙ **Easy to use:** Anyone can begin using Bootstrap with just a basic Knowledge of HTML and CSS.

∙ **Responsive features:** The responsive CSS in Bootstrap adapts to mobile devices, tablets, and desktops.

∙ **Mobile-first approach:** The fundamental Bootstrap framework provides mobile-first styling.

∙ **Browser compatibility:** All current browsers are compatible with Bootstrap (Chrome, Firefox, Internet Explorer, Safari, and Opera).

**3.1.4 JavaScript**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. JavaScript adds interactive elements to online pages that keep users engaged, whereas HTML and CSS are languages that give web pages structure and style.

The prototype is a built-in property that every JavaScript object has. The prototype is itself an object, so the prototype will have its own prototype, making what's called a prototype chain. When we get to a prototype that contains null for its own prototype, the chain comes to an end.

**3.1.5 PHP**

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It was one of the first server-side languages that HTML could incorporate, making it simpler to add functionality without having to call outside files for information. One of the best things about PHP is how beginner-friendly it is while still offering many advanced features to professional programmers.

**3.2 Server-Side Technology**

When an application is used, server-side development refers to the processes that happen in the back ground. Databases, scripting, website architecture, backend logic, APIs, and servers are the main topics covered.

* + 1. **Database Server**

MySQL is an open-source relational database management system (RDBMS). A relational database arranges data into one or more tables where it is possible for the data to be connected to one another. Programmers use the SQL language to create, change, and extract data from relational databases and to manage user access to the databases.

**4.Requirement Specification**

**4.1 Functional Requirement**

Functional requirements are those that serves as examples for the system’s internal operation, its description, and an explanation of each subsystem. I comprise of the task that the system should complete, the associated processes, the data that the system should store, and the user interfaces.

**4.1.1 User registration and login**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-1** | User (donor, receiver, system, blood bank) registration and login a registered account of ClaretCorridor. | | |
| **Description** | User should register his/her account for the first time and be able to login to the account which was registered once. Already registered users will not face this stage. At first the registration request is checked by the admin and then it is finally approved. | | |
| **Stakeholders** | User (donor, receive, blood bank), System | **Priority** | High |

**4.1.2 Communication:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-2** | User (donor, recipient, blood bank) can communicate by themselves or through blood bank. | | |
| **Description** | The user has to log in and goes to his/her profile. There will be a option of communication. They can communicate by themselves easily or through blood bank. | | |
| **Stakeholders** | User (donor, recipient), Blood bank. | **Priority** | High |

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-3** | User (donor, recipient, system, blood bank) can post for finding blood of specific blood group. | | |
| **Description** | User goes to their profile and select the option of post to create a post for finding any specific blood group. | | |
| **Stakeholders** | User (donor, recipient, blood bank, system) | **Priority** | High |

**4.1.3 Posting**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-4** | User (donor, recipient, admin, blood bank) can find any blood and health related information. | | |
| **Description** | Sometimes, it is important to see what blood group are available in blood bank and how much and also there will be the information of donor’s blood group. That’s why the system will perform this operation. | | |
| **Stakeholders** | User (donor, recipient, blood bank, system) | **Priority** | High |

**4.1.4 View Blood bank information**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-5** | User (donor, recipient, blood bank) can update their profile any time. | | |
| **Description** | To update the profile the users must be log in to their profile and if there is need to update any information, users and blood bank can update it. | | |
| **Stakeholders** | User (donor, recipient, blood bank) | **Priority** | High |

**4.1.5 Management**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-6** | The system will send notification. | | |
| **Description** | If the blood donating time (4 months) is over, the system will send notification to him/her to know it. Also, when someone post about to find blood, it will be notified to all. | | |
| **Stakeholders** | System | **Priority** | High |

**4.1.6 Send notification**

|  |  |  |  |
| --- | --- | --- | --- |
| **FR-7** | Blood bank will update the last donation time. | | |
| **Description** | When someone donate blood, the date of donating is updated by his/her profile by blood bank. | | |
| **Stakeholders** | Blood bank | **Priority** | High |

**4.1.7 Last update donation time**

**4.2 Performance Requirement**

It is important that maintain the performance of the system. To ensure the best performance of the system we must maintain the following steps:

**4.2.1 Speed and Latency Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **PR-1** | Faster searching for information of blood bank and fast notification system. | | |
| **Description** | When any authentic user of our system wants to search for any information of blood bank and get notified, then the user can feel the fast searching and notifying speed. | | |
| **Stakeholders** | User (donor, recipient, blood bank), system. | **Priority** | High |

**4.2.2 Safety Critical Requirements**

For our project there is no safety critical requirements.

**4.3 Maintainability and Supportability**

The term "maintenance" describes how simple it is to fix, enhance, and comprehend software code.  After the user has used the software, the software maintenance phase of the software development cycle begins.

**4.3.1 Maintenance Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **MR-1** | Develop maintainable code | | |
| **Description** | Maintainability must be ensured so that it can be modified later and will be readable. | | |
| **Stakeholders** | Developers | **Priority** | High |

**4.3.2 Supportability Requirements**

This system satisfies the supportability requirements for testability, maintainability, compatibility, configurability, serviceability, and install ability.

**4.4 Security Requirements**

Information security is far more crucial for a system to gain users' trust. Here are some security requirements are given below:

**4.4.1 Access Requirements**

The system will apply some authorization approaches when granting access to information to make sure the right user is using the right data.

**4.4.2 Integrity Requirements**

Integrity requirements relate to a security system that ensures an expectation of data quality. It also ensures that no data on the system will ever be exposed to malicious modification or accidental deletion.

**5.Requirement Engineering Process**

Software requirements are established using requirements engineering (RE), which takes into account customer wants or requirements. Requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management are all parts of the requirements engineering process.

**5.1 Requirement Elicitation Techniques**

Requirements elicitation, often known as "requirement gathering," is the process of investigating and discovering system requirements for users, clients, and other stakeholders. Contacting participants directly or conducting research, analysis, and testing are two ways to elicit requirements.

**5.1.1 Hold Interviews**

We have conversations that can be had alone or with a small group of people. They are a useful approach to accessing services without having to spend a lot of time with participants because we simply meet with them to go through a few key program criteria. Negotiations are useful for getting specific demands from participants in setting up workshops where those program participants gather to address any problems or conflicts. Our interviews are primarily conducted using a set of predetermined standards

**5.1.2 Perform Document Analysis**

Existing documentation can assist in demonstrating how systems are being used or what I should do with them. Documents contain textual details regarding existing programs, operational procedures, required specifications, and market research on competitors. Once again, textual analysis can be useful. Determine which features should be removed and which performances should stay by consulting the Software Requirements Specification.

**5.1.3 Distribute Questionnaires**

The questionnaire is a helpful tool for examining user satisfaction with priorities and preferences, changes in attitudes and ideas, and styles. We tried to keep our question lists to a minimum. The response can be worn out or angry. A fundamental justification for each question and grouped the subject areas together for the respondent to concentrate on. The primary benefit of the survey responses was that they were gathered in the expected manner. Many others provided summaries of the information.

**5.1.4 Sample of Requirement Collection**

*Table 1 Requirement Collection 01*

|  |  |
| --- | --- |
| Requirement Elicitation  Techniques | Interview,  Field Observation |
| Collected From | Noakhali Blood Bank and Transfusion Center, Maijdee, Noakhali. |
| Findings | 1. Third party should not get any information because they can hack the database of blood bank. 2. Blood bank can only delete the id of any user (donor, recipient). |

*Table 2 Requirement Collection 02*

|  |  |
| --- | --- |
| Requirement Elicitation  Techniques | Field Observation |
| Collected From | Prime Hospital, Maijdee, Noakhali. |
| Findings | 1. Blood bank should update the last donation time of donor.  2. any user can post for the specific blood group. |

*Table 3 Requirement Collection 03*

|  |  |
| --- | --- |
| Requirement Elicitation  Techniques | Interview |
| Collected From | Sadar Hospital, Maijdee, Noakhali. |
| Suggestion | If anyone donated blood four months ago then he/she will be notified by blood bank. |

**5.2 Requirement Validation**

Requirement validation criteria makes sure they are accurate and match the standard you desire from this program. Our requirements initially appeared to be good, but after reading them and attempting to implement them, we discovered that they contained gaps and ambiguities.

**5.2.1 Review the Requirements**

Among the techniques that produce the highest quality software now accessible, negative peer review, particularly the rigorous type known as evaluation, is exceptional. We carefully looked at documented needs, analysis models, and related disability information with a team of reviewers from various viewpoints.

**5.2.2 Simulate the Requirements**

We can use trading tools to simulate a suggested system in place or to add specifics to textual specifications in order to stimulate requirements. The simulation advances the concept of prototyping.

**6.Use Case Diagram**

**Actors:** Actors are entities or roles that interact with a system, typically in the context of software development or system design. In the realm of use case modeling, actors represent individuals, groups, or external systems that interact with a system to achieve specific goals. Actors can be users, other software systems, or even hardware devices that play a role in the system's functionality.

**Use Case Diagram:** A use case diagram, on the other hand, is a visual representation that depicts the functional requirements of a system from the end-user perspective. It illustrates how different actors interact with a system and its various use cases (specific functionalities or features). Use case diagrams are commonly used in software engineering and systems analysis to define, clarify, and organize system requirements.

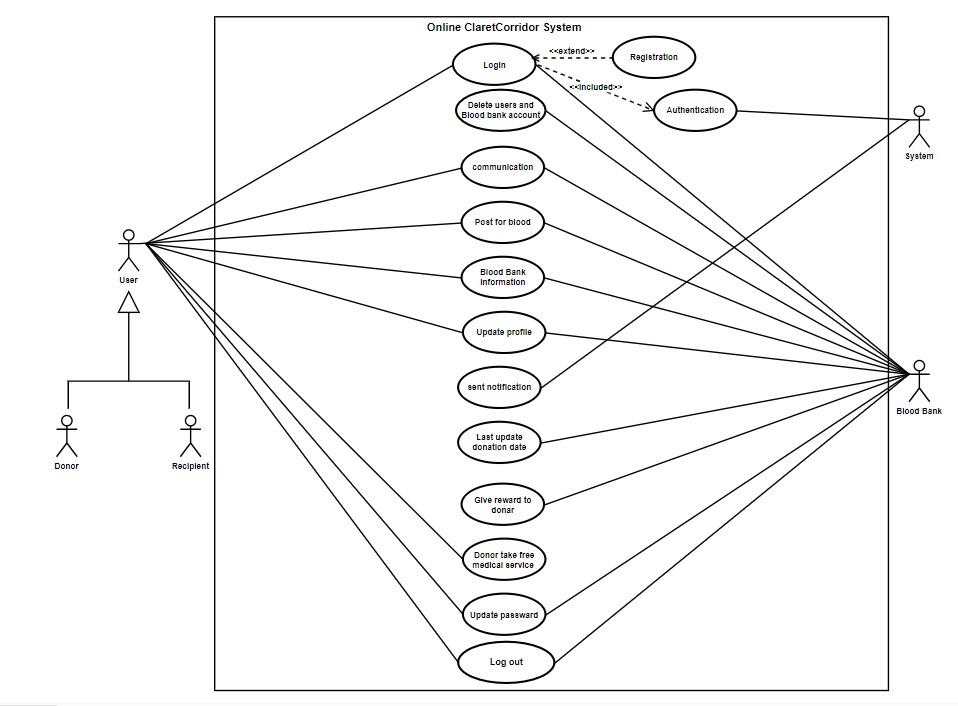


Figure 1 Use Case Diagram

**7. Use Case Descriptions**

*Table 04 Registration*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 01 | |
| **Use Case** | Registration | |
| **Goal** | A user (donor, recipient) request for registration and have got an id. | |
| **Preconditions** | Must be have NID card or birth certificate. | |
| **Success End Condition** | “User Registration Request” sent successfully. | |
| **Failed End Condition** | “User Registration request” sending process failed. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient)  System | |
| **Trigger** | The user clicks the “Register” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The user enters the first page of the “ClaretCorridor”. |
| 2 | The user clicks the “Register” button. |
| 3.1  3.2 | The user selects the “Register as Donor” option.  The user selects the “Register as Recipient” option.  . |
| 4 | The user provides the required information. |
| 5 | The user clicks the “Send Registration Request” button |
| 6 | The system sends a message as a response to the user with all required information. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 3a | The user did not select any option. |
| 3a1 | The registration page remains unchanged. |
| 4a | The user provided any invalid data. |
| 4a1 | The webpage shows “Invalid Data”. |
| 4a2 | The specific field gets cleared in which the user has given the wrong data. |
| **Quality Requirements** | **Step** | **Requirement** |
| 4 | The user should give valid information. |

*Table 5 Login*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 02 | |
| **Use Case** | Login | |
| **Goal** | A user (donor, recipient, blood bank) can log in to the system. | |
| **Preconditions** | Must be registered. | |
| **Success End Condition** | The user or blood bank has successfully logged in to the system. | |
| **Failed End Condition** | The user or blood bank could not log in to the system. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient), Blood bank  System | |
| **Trigger** | The primary actor clicks the “Login” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The primary actor enters the first page of the ClaretCorridor. |
| 2.1  2.1 | The primary actor clicks “Login as Donor” option.  The primary actor clicks “Login as Recipient” option.  The primary actor clicks “Login as Blood bank” option. |
| 3 | The primary actor enters username and password. |
| 4 | The system checks for validation. Use Case “Authentication”. |
| 5 | The primary actor successfully logs in to the system. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 4a | The primary actor provides invalid data. |
| 4a1 | The system prints a “Error” message. |
| 4a2 | The primary actor cannot log in to the system. |
| **Quality Requirements** | **Step** | **Requirement** |
| 3 | The system will not allow incorrect passwords more than 5 times and if  happens, keeps waiting for 20 minutes. |
| 4 | The primary actor authentication must be accurate. |

*Table 06 Delete users and blood bank account*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 03 | |
| **Use Case** | Delete users and blood bank account | |
| **Goal** | Blood bank can delete the account of user if the user is no longer want to keep the id. | |
| **Preconditions** | Log in to the system successfully. | |
| **Success End Condition** | Blood bank deletes the Id successfully. | |
| **Failed End Condition** | Blood bank doesn’t delete the id successfully. | |
| **Primary Actors:**  **Secondary Actors:** | Blood bank  Users (donor, recipient). | |
| **Trigger** | The user clicks the “Delete” button. | |
|  | 1 | The user should complete his/her registration. |
| 2 | The user log in to their profile. |
| 3 | Blood bank clicks the “Delete” button. |
| 4 | The system shows confirm delete. |
| 5 | The id has deleted from the ClaretCorridor app. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 4a | The system doesn’t show confirm delete. |
| 5a | The id has not removed or deleted. |
| **Quality Requirements** | **Step** | **Requirement** |
| 2 | The user must have an id. |

*Table 07 Communication*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 04 | |
| **Use Case** | Communication | |
| **Goal** | The user (donor, recipient) can communicate with each other through messaging, phone call through the app and also blood bank will communicate with them through the system. | |
| **Preconditions** | Log in to the system successfully. | |
| **Success End Condition** | The user has communicate with each other successfully. | |
| **Failed End Condition** | The user could not communicate with each other. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, recipient, blood bank)  System | |
| **Trigger** | The user clicks the “chat” button or “call” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The user enters the ClaretCorridor home page after login. |
| 2 | The user clicks the “” button. |
| 3 | The system shows a list of available donors’ and recipients’ id with their email and phone number. |
| 4 | The user searched for the specific donor or recipient. |
| 5 | The desired user has found. |
| 6 | The user clicks the persons id and shows the details. |
| 7 | The user clicks the desired option. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 4a | The user searched using the search option |
| 5a | The desired person is not found. |
| 5a1 | The system prints a “Sorry” message and shows a button called “search again”. |
| **Quality Requirements** | **Step** | **Requirement** |
| 4 | Searching should be fast and accurate. |
| 7 | All the details must be correct. |

*Table 08 Post for blood*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 05 | |
| **Use Case** | Post for blood | |
| **Goal** | The user and blood bank can post on the page of the app to find blood donor or specific blood group. | |
| **Preconditions** | Log in to the system successfully. | |
| **Success End Condition** | The user and blood bank has posted successfully. | |
| **Failed End Condition** | The user and blood bank could not post successfully. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient, Blood bank)  System. | |
| **Trigger** | The user clicks the “Post” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The user enters the ClaretCorridor home page after login. |
| 2 | The user clicks the “Post” button. |
| 3 | The system shows some options (photo, text) before posting. |
| 4 | The system show done option. |
| 7 | The system will approve the post. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 4a | The system does not show done option. |
| 4a1 | The webpage shows “Invalid”. |
| **Quality Requirements** | **Step** | **Requirement** |
| 4 | The user should post and click the done option for posting. |

*Table 09 Blood bank information*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 06 | |
| **Use Case** | Blood bank information | |
| **Goal** | The users (donor, recipient) find all the information of blood bank. | |
| **Preconditions** | Log in to the system successfully. | |
| **Success End Condition** | The user has checked the details successfully. | |
| **Failed End Condition** | The user could not check the details. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, recipient, blood bank)  System | |
| **Trigger** | The user clicks the “Details” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The user enters the ClaretCorridor home page after login. |
| 2 | The user enters into his/her profile. |
| 3 | The user clicks the “Details” button. |
| 4 | The system shows details with all required information. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 4a | The user could not click details button. |
| 4a1 | The system shows no information. |
| **Quality Requirements** | **Step** | **Requirement** |
| 4 | All details must be valid. |

*Table 10 Update profile*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 07 | |
| **Use Case** | Update profile | |
| **Goal** | A user (donor, recipient, blood bank) can update the profile any time. | |
| **Preconditions** | Log in to the system successfully | |
| **Success End Condition** | The user has updated the profile successfully. | |
| **Failed End Condition** | The user could not update the profile. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient, Blood bank)  System | |
| **Trigger** | The user clicks the “Update” icon on the top of a button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The user enters the first page of the ClaretCorridor . |
| 3 | The user clicks the “Update” icon on the top of a button. |
| 4 | The system shows the profile of users. |
| 5 | The user can update any specific information. |
| 6 | Profile is updated. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 3a | The profile is not found. |
| 3a1 | The system shows a “Sorry” message and back to the home page. |
| **Quality Requirements** | **Step** | **Requirement** |
| 6 | Updating must be faster. |
|  | 7 | The information in the profile must be valid after updating. |

*Table 11 Send notification*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 8 | |
| **Use Case** | Send notification | |
| **Goal** | The system sends the notification to the users (donor, recipient, blood bank). | |
| **Preconditions** | There should be any operation that requires a response such as “Notification bar”. | |
| **Success End Condition** | A system-generated notification has been sent to specific and all destination with all the required information (date over of donating blood, notification after post). | |
| **Failed End Condition** | A system-generated notification has not been sent to all. | |
| **Primary Actors:**  **Secondary Actors:** | System  User (Donor, recipient, Blood bank). | |
| **Trigger** | A notification request comes in. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The system sets the destination notification address. |
| 2 | The system sets the information. |
| 4 | The system sends the notification. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 1a | The destination notification address is not valid. |
| 1a1 | The system shows an “Error” message. |
| 1a2 | Notification is not sent. |
| **Quality Requirements** | **Step** | **Requirement** |
| 3 | The notification should be contained all required information. |

*Table 12 Last Update Donation Date*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 9 | |
| **Use Case** | Last update donation date | |
| **Goal** | Blood bank updates the last donation time in the users’ (donor, recipient) profile. | |
| **Preconditions** | The users’ (donor, recipient) must have an id. | |
| **Success End Condition** | A user’s last donation is updated successfully. | |
| **Failed End Condition** | A user’s last donation is not updated. | |
| **Primary Actors:**  **Secondary Actors:** | Blood bank.  User (Donor, Recipient). | |
| **Trigger** | Blood bank clicks “Update time” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | Primary actors click the “Update time” button. |
| 2 | Primary actor checks when the users donate last. |
| 3 | Primary actors update the profile of the users. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 2a | Primary actors does not check the last donation time. |
| 2a1 | Profile is not updated. |
| **Quality Requirements** | **Step** | **Requirement** |
| 5 | Primary actors must check the last donation time of the users. |

*table 13 Give rewards*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 10 | |
| **Use Case** | Give rewards. | |
| **Goal** | Blood bank gives rewards as stars to the users (Donor, Recipient) after every donating. | |
| **Preconditions** | The users must donate blood. | |
| **Success End Condition** | Blood bank gives rewards as stars to the users (donor) successfully. | |
| **Failed End Condition** | Blood bank does not give rewards as stars to the users (donor). | |
| **Primary Actors:**  **Secondary Actors:** | Blood bank  User (Donor). | |
| **Trigger** | Primary actor click’s the “Reward” button which is only shown for the blood bank. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The blood bank enters into the ClaretCorridor’s home page |
| 2 | Primary actor checks the last donation of donor. |
| 3 | Primary actor gives the rewards as star. |
| **Quality Requirements** | **Step** | **Requirement** |
| 2 | Users must be donate blood for gaining stars. |

*Table 14 Free Medical Service*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 11 | |
| **Use Case** | Free Medical Service. | |
| **Goal** | The users (Donor) who get rewarded can take free medical service by showing these stars. | |
| **Preconditions** | The users (donor) must be had enough stars. | |
| **Success End Condition** | The users (donor) get free medical service successfully. | |
| **Failed End Condition** | The users (donor) do not get free medical services. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor).  Blood bank, System. | |
| **Trigger** | Medical service. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The users (donor) enter into the ClaretCorridor home page. |
| 2 | The user (donor) checks if he/she has enough stars. |
| 3 | Request to blood bank for free medical services. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 2a | The user (donor) has no enough stars. |
| 2a1 | The system shows the “Sorry” message. |
| **Quality Requirements** | **Step** | **Requirement** |
| 2 | The user (donor) must have enough stars for taking free medical services. |

*Table 15 Update Password*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 12 | |
| **Use Case** | Update Password | |
| **Goal** | The user (donor, recipient), blood bank can change their password. | |
| **Preconditions** | Log in to the system successfully by the user (donor, recipient), blood bank. | |
| **Success End Condition** | The user (donor, recipient) and blood bank has updated the password successfully. | |
| **Failed End Condition** | The user (donor, recipient) and blood bank could not update the password. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient), Blood Bank  System | |
| **Trigger** | The primary actor clicks the “Update Password” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The primary actor enters the home page of ClaretCorridor after login. |
| 2 | The primary actor clicks the “Update Password” button. |
| 3 | The primary actor enters the old password. |
| 4 | The primary actor enters the new password. |
| 5 | The primary actor clicks the “Update” button. |
| 6 | The system finds that the old password is matched. |
| 7 | The system updates the database. |
| 8 | The system shows a “Successful” message. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 6a | The old password is not matched. |
| 6a1 | The system shows a “Error” message. |
| **Quality Requirements** | **Step** | **Requirement** |
| 7 | The database must be updated after every operation is performed. |

*able 16 Logout*

|  |  |  |
| --- | --- | --- |
| **Use Case No.** | 13 | |
| **Use Case** | Logout | |
| **Goal** | A user (donor, recipient), and blood bank can log out from the system. | |
| **Preconditions** | Log in to the system successfully and must be a user (donor, recipient) or blood bank. | |
| **Success End Condition** | The user (donor, recipient) or the blood bank has successfully logged out from the system. | |
| **Failed End Condition** | The user (donor, recipient), blood bank could not log out from the system. | |
| **Primary Actors:**  **Secondary Actors:** | User (Donor, Recipient), Blood bank    System | |
| **Trigger** | The user (donor, recipient), blood bank clicks the “Logout” button. | |
| **Main Success Flows** | **Step** | **Action** |
| 1 | The primary actor enters the home page of ClaretCorridor after login. |
| 2 | The primary actor clicks “Logout” button. |
| 3 | The primary actor has successfully logged out from to the system. |
| **Alternative Flows** | **Step** | **Branching Action** |
| 2a | The primary actor does not click “Log out” button. |
| 2a1 | The system performs according to the instructions of primary actor. |
| **Quality Requirements** | **Step** | **Requirement** |
| 2 | The system should ask “Are you Sure to Logout?” after clicking the logout button. |

**8. Activity Diagram**

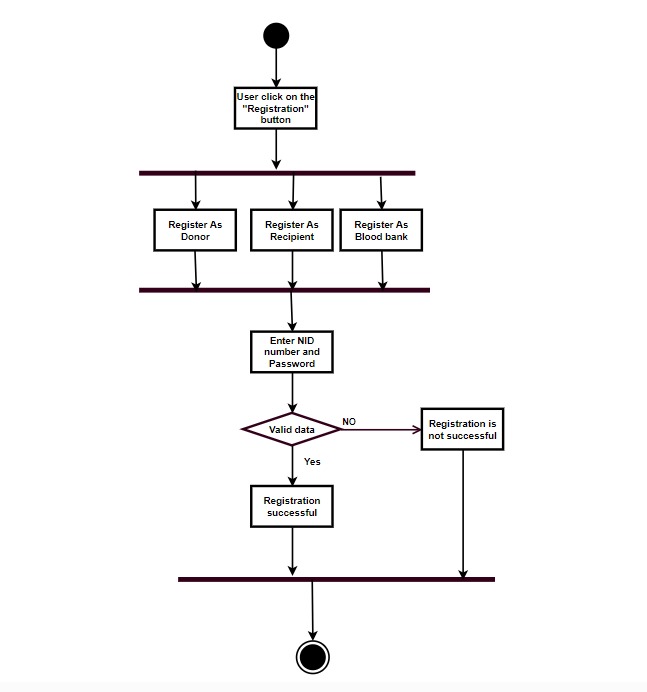


Figure 2 Registration

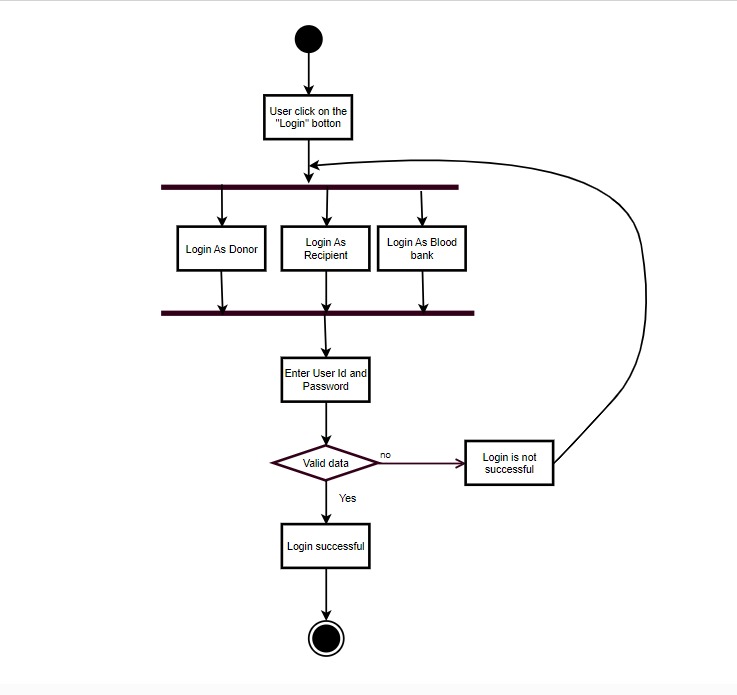


Figure 3 LogIn

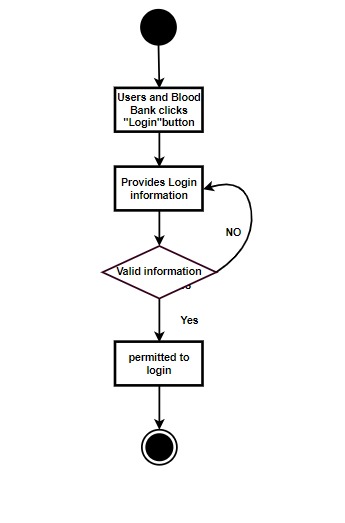


Figure 4 Authentication

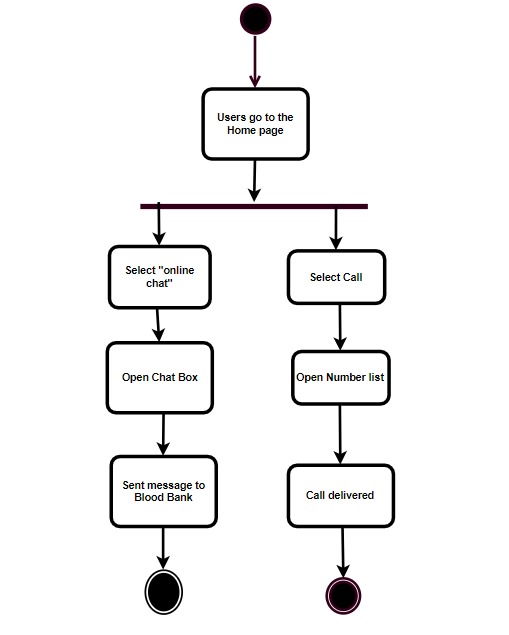


Figure 5 Communication

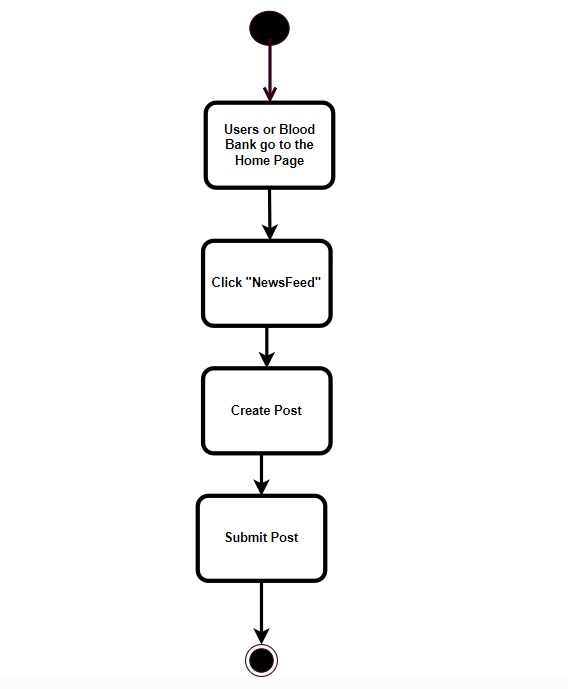


Figure 6 Post

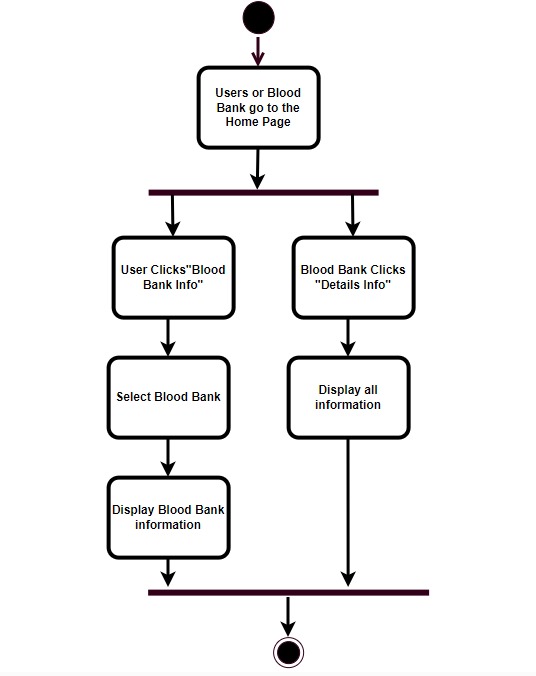


Figure 7 Blood bank Information

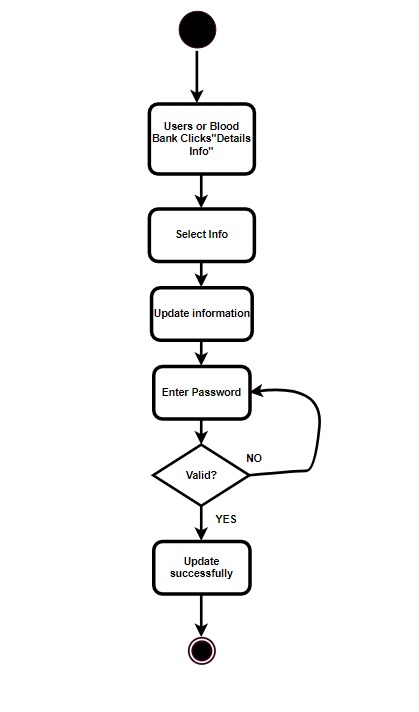


Figure 8 Management

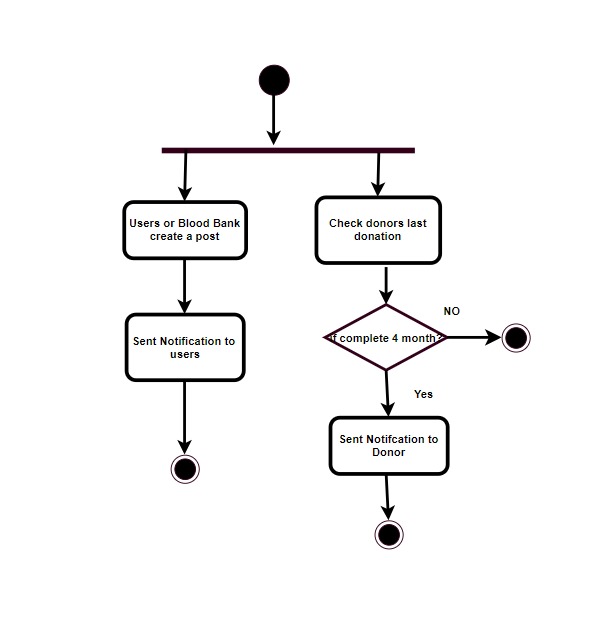


Figure 9 Notification

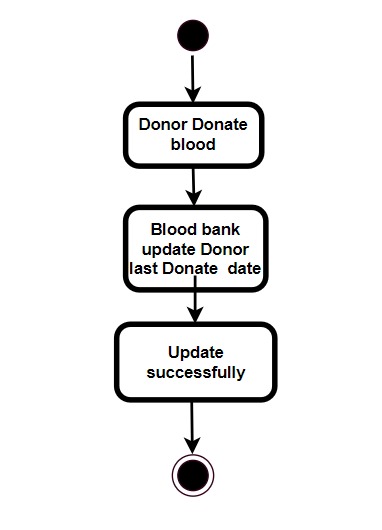


Figure 10 Last update donation date

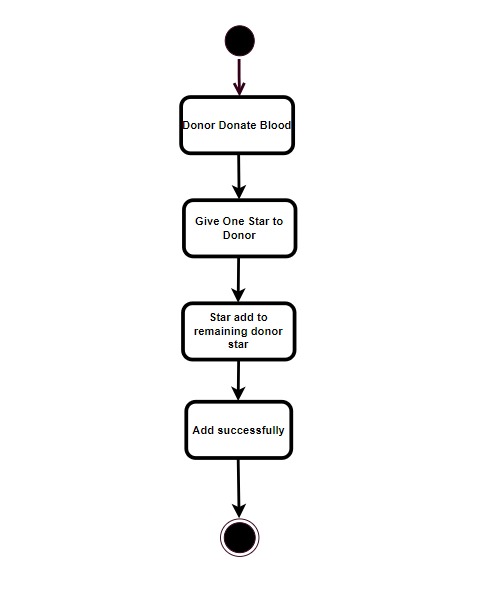


Figure 11 Give reward

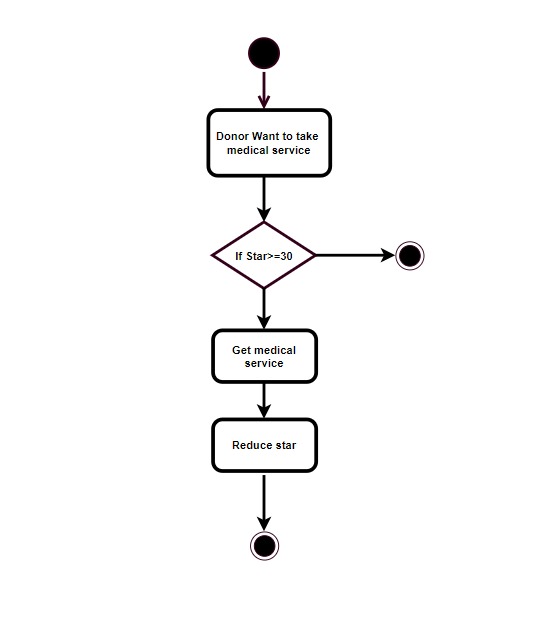


Figure 12 Free medical service

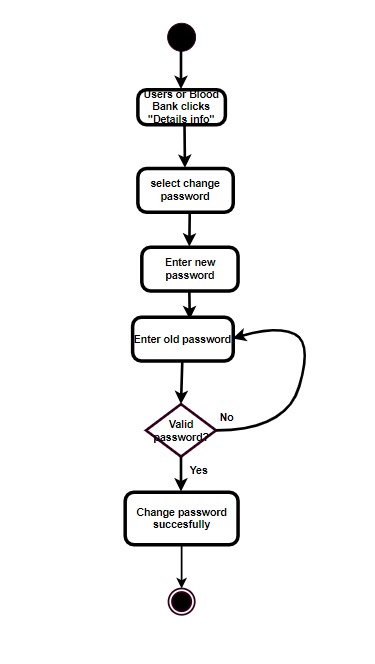


Figure 13 Update password

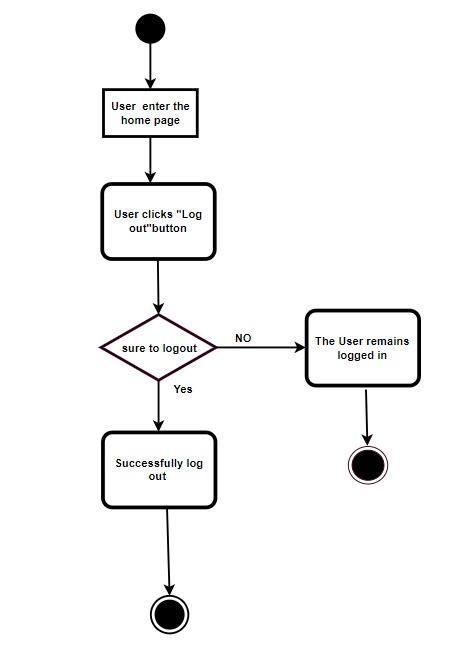


Figure 14 Log out

**9. Sequence Diagram**

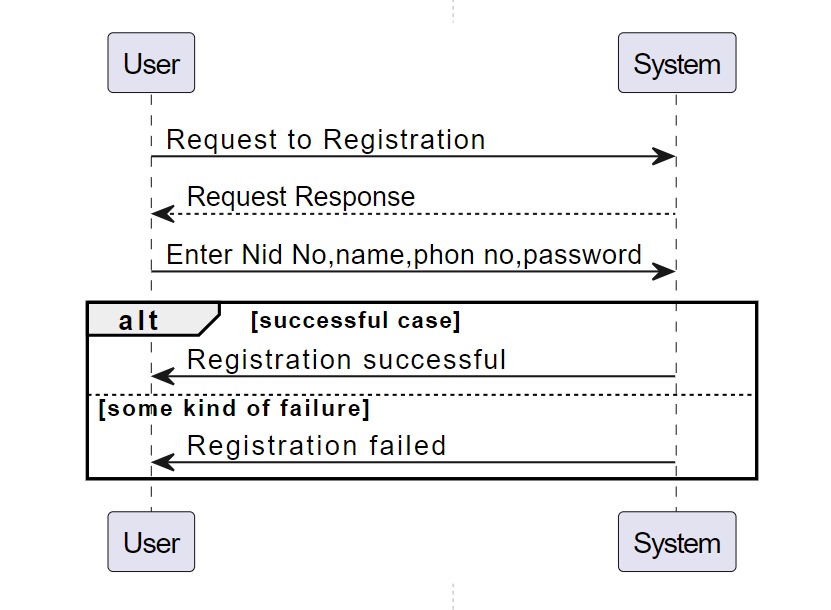


Figure 15 Registration

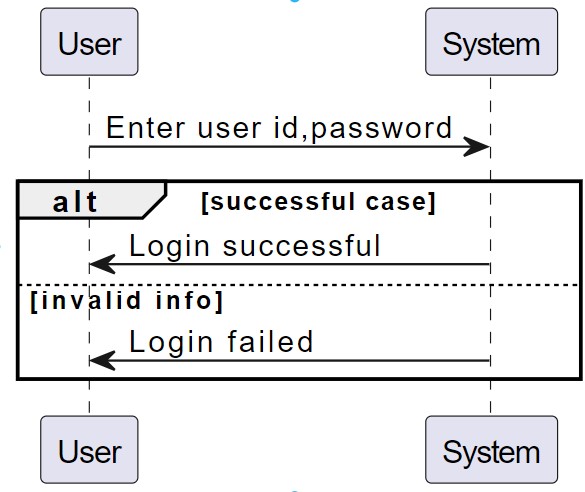


Figure 16 Log in

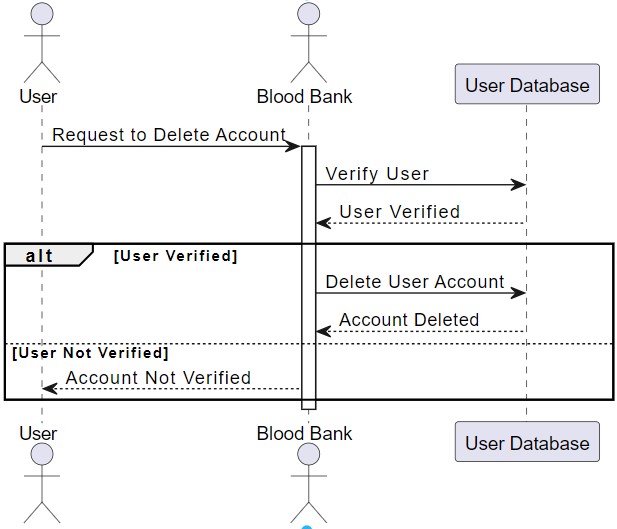


Figure 17 Delete user

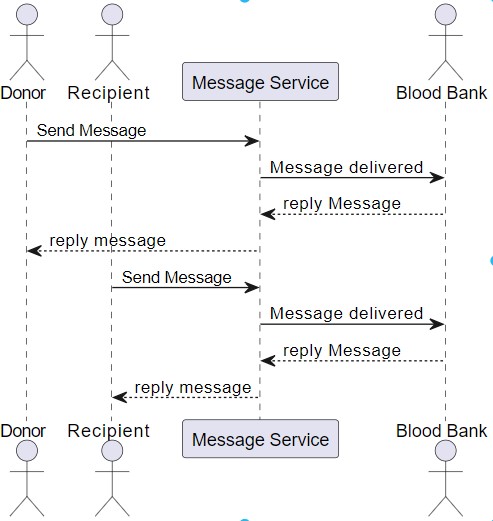


Figure 18 Communication

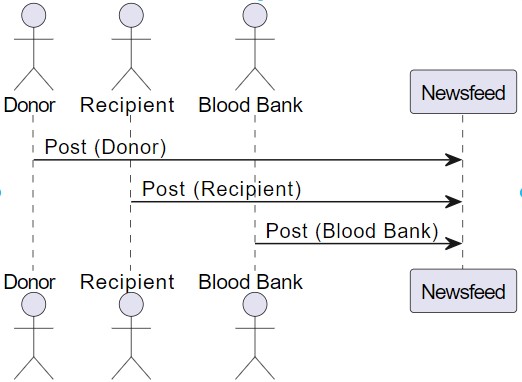


Figure 19 Post for blood

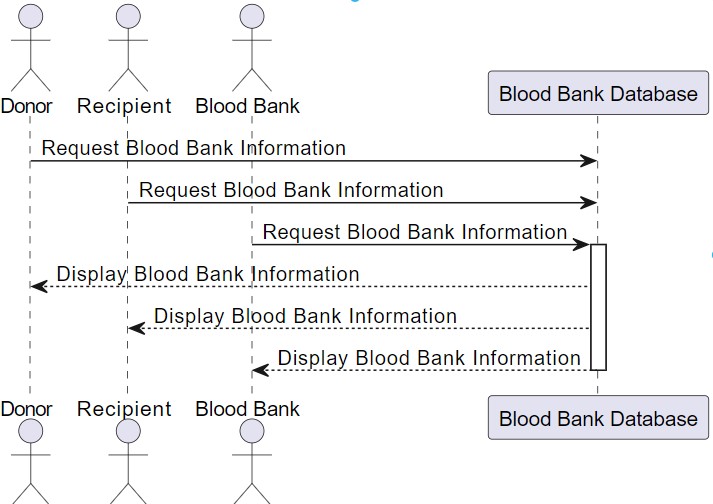


Figure 20 View blood bank info

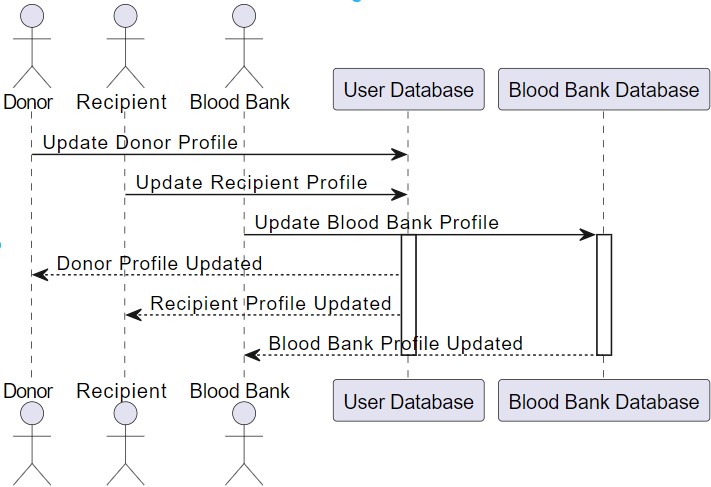


Figure 21 Management

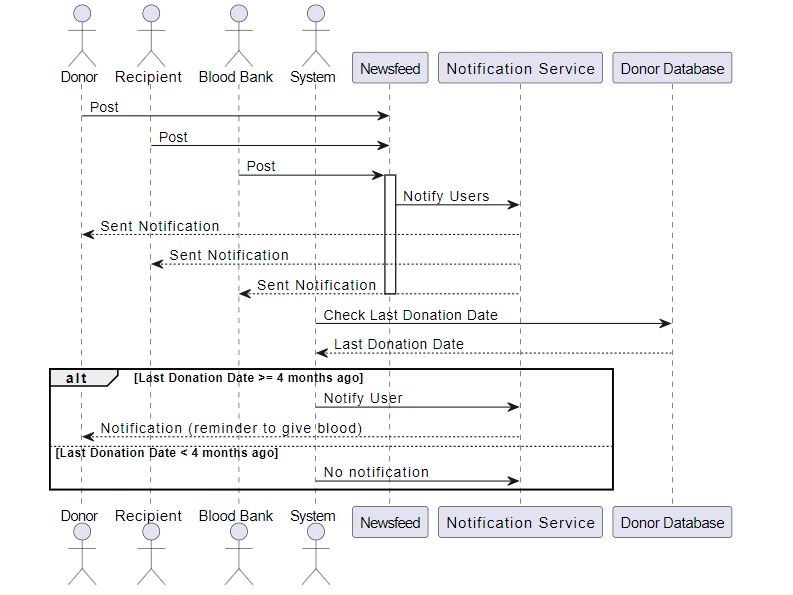


Figure 22 Send notification

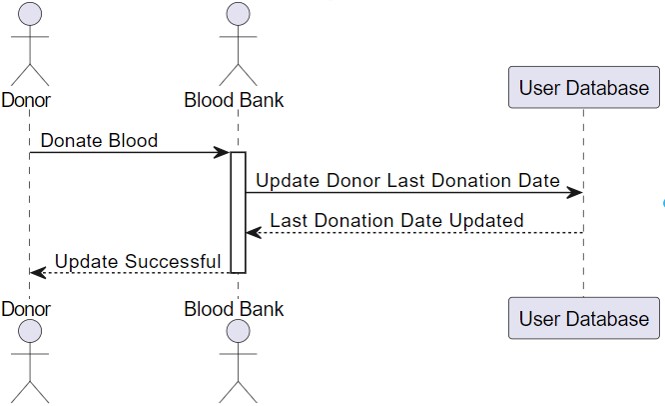


Figure 23 Last update donation date

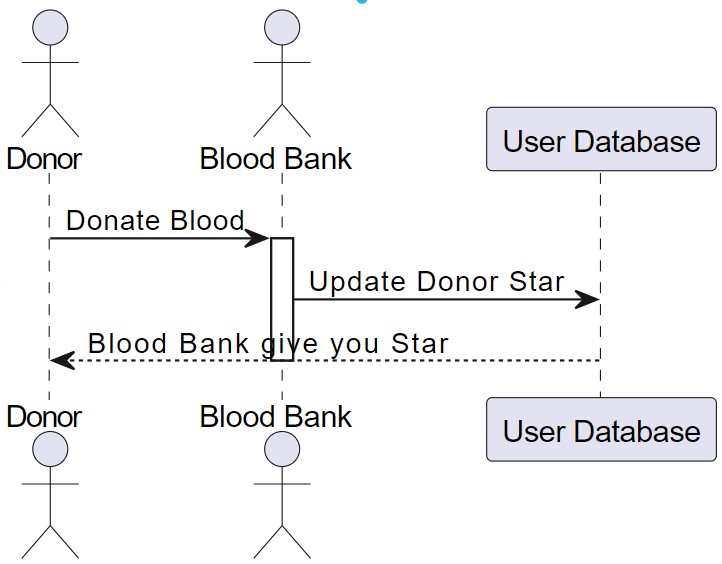


Figure 24 Give reward

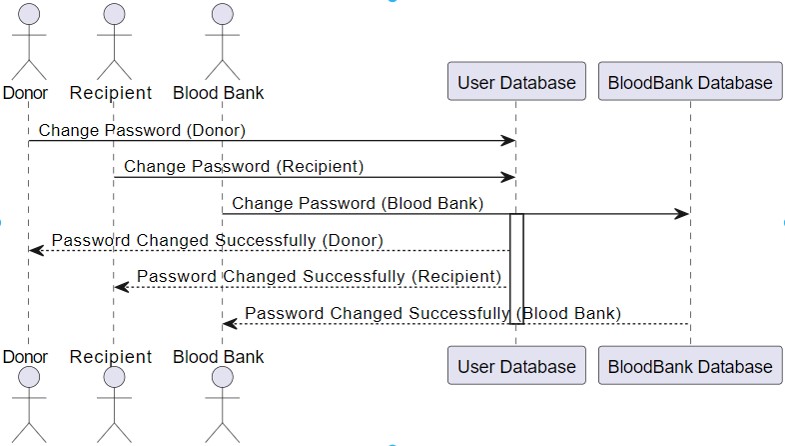


Figure 25 Management

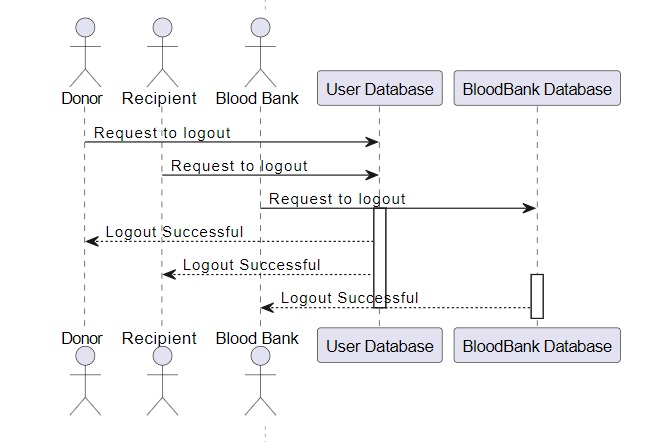


Figure 26 Logout

**10. Swimlane Diagram**

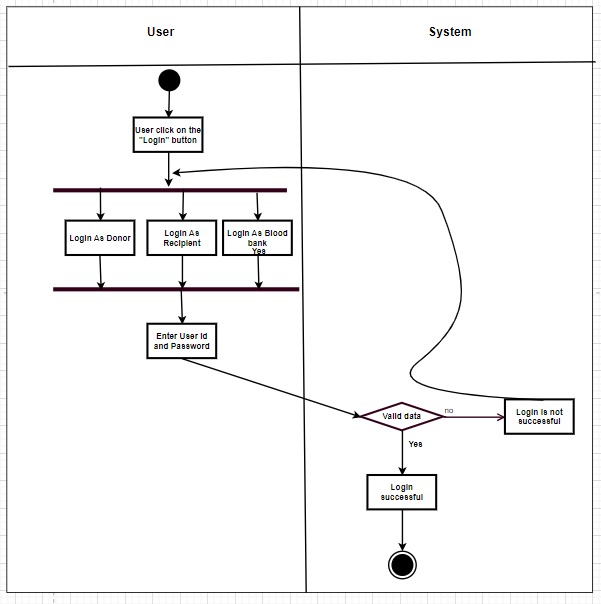


Figure 27Login

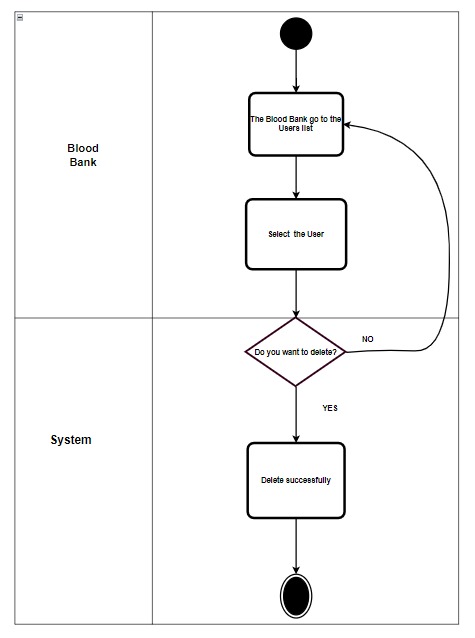


Figure 28 Delete user

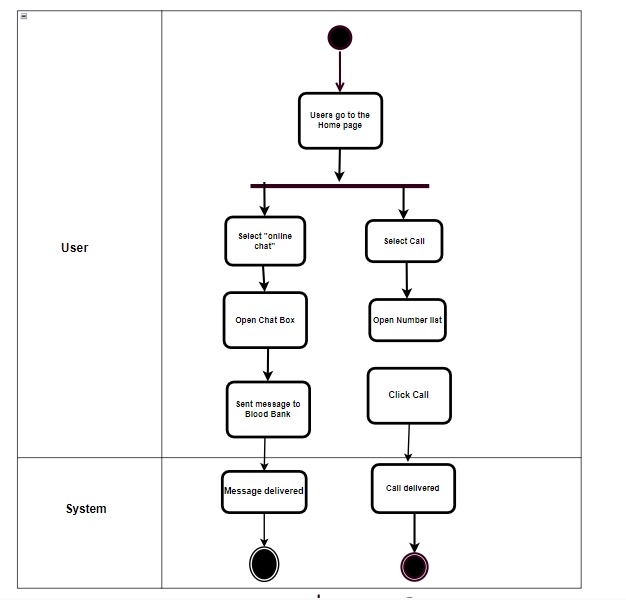


Figure 29 Communication

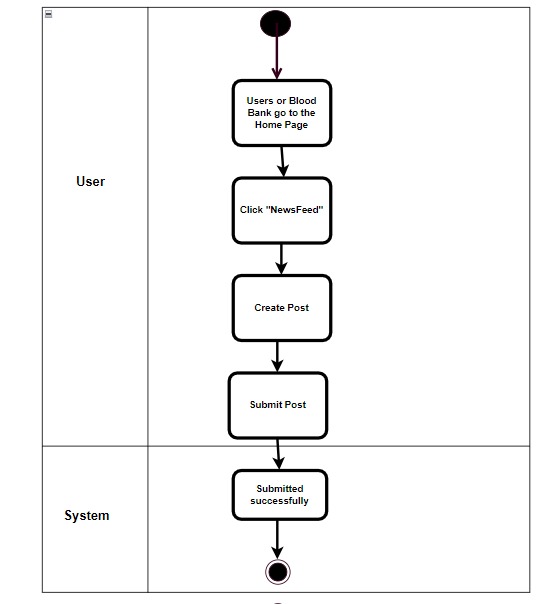


Figure 30 Post for blood

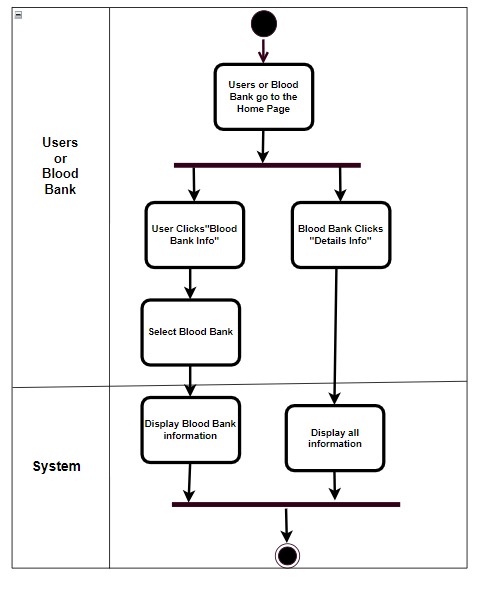


Figure 31 Blood bank info

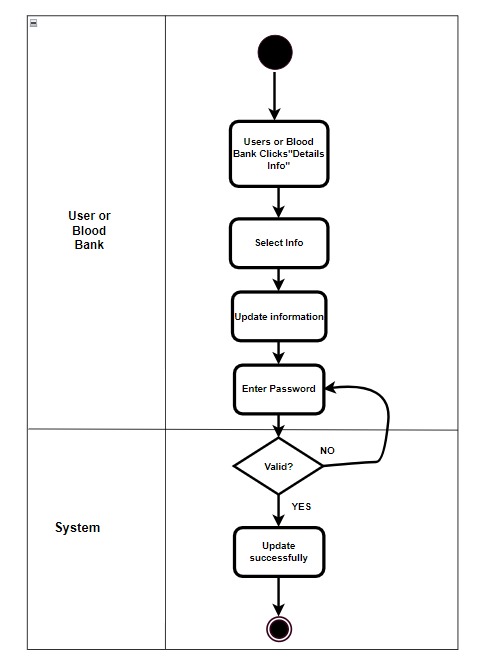


Figure 32 Update profile

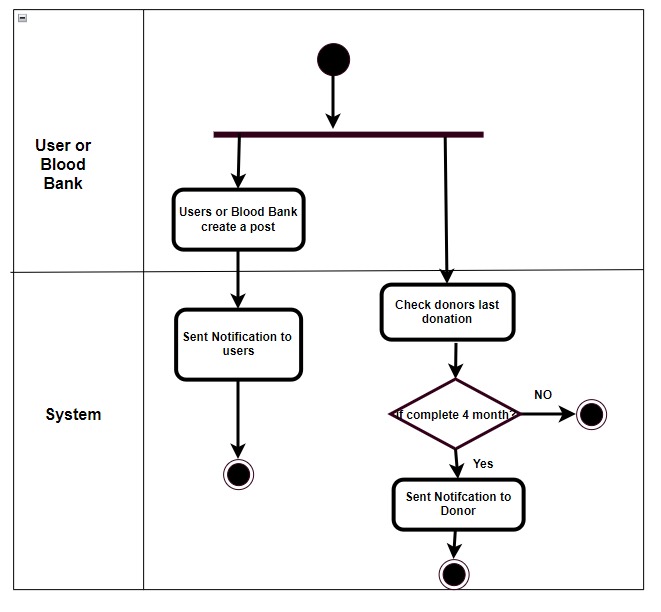


Figure 33 Sent notification

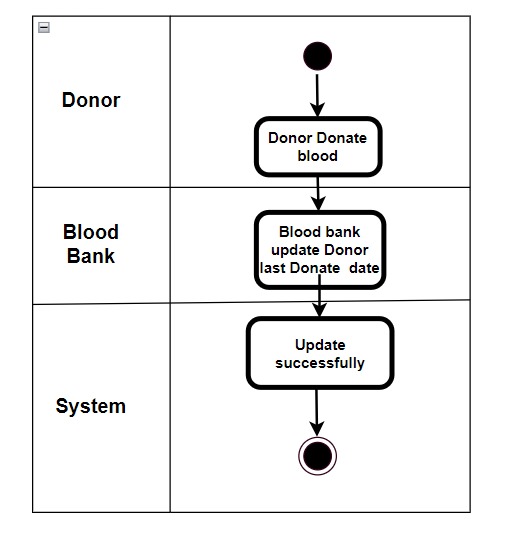


Figure 34 Last update donation time

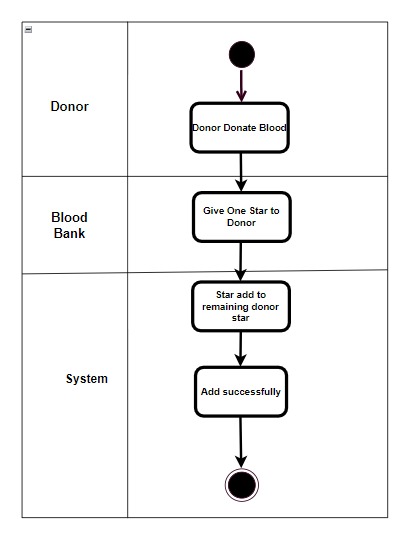


Figure 35 Get reward

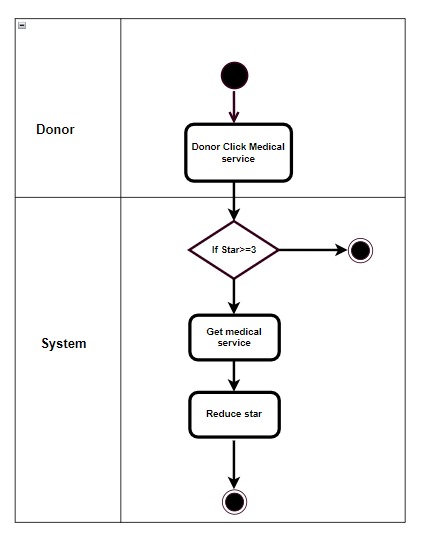


Figure 36 Free medical service

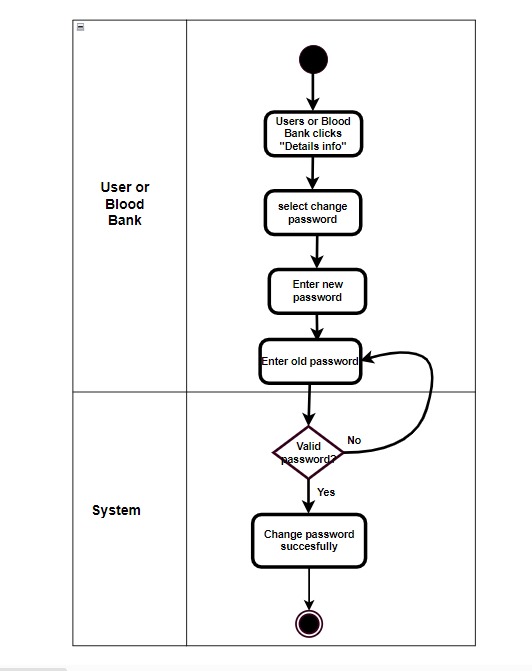


Figure 37 update password

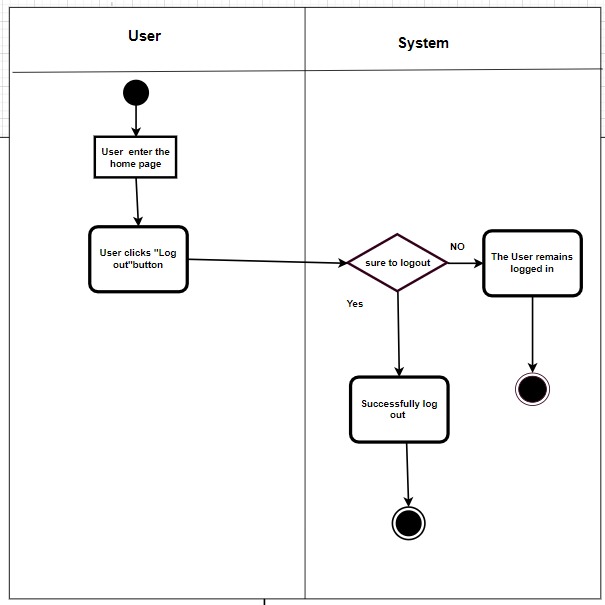


Figure 38 Log out

**11.**  **Conclusion**

In conclusion, "ClaretCorridor" stands as a testament to the collaborative spirit and collective impact of its stakeholders. The diverse array of contributors, from donors and recipients to medical professionals, regulatory bodies, and community members, forms the bedrock of this transformative blood bank initiative. Through seamless integration, we aim to create a community where every act of donation is not only recognized but reciprocated with access to vital healthcare services. Together, we build a corridor of care, uniting stakeholders in a shared mission for a healthier and more connected world.