BLOOD BANK MANAGEMENT SYSTEM

UCS503 Software Engineering Project Report
End Semester Evaluation

Submitted by:
102017093 Nirbhay Makhija
102017095 Anjali Rana
102017098 Sanskar Kapoor
102017099 Naman Khurana

BE Second year , CSE

Group number : A

Submitted to:
Ms. Harkiran Kaur

Computer Science and Engineering Department
TIET , Patiala
May 2022

TABLE OF CONTENTS

1.1 Software Bid/ Project Teams

UCS 503- Software Engineering Lab

Group: 2CS5 Dated: 31st Jan,2022

Team Name: Bits && Bytes
Team ID (will be assigned by Instructor): Team A

Please enter the names of your Preferred Team Members. :

You are required to form **a three to four person** teams, with at-least one male and one female member (wherever feasible).
Choose your team members wisely. You will not be allowed to change teams.

Name	Roll No	Project Experience	Programming Language used
Anjali Rana	102017095	Portfolio site	JavaScript
Naman Khurana	102017099	Weather Forecasting app	Java, XML
Nirbhay Makhija	102017093	Notefy, Url shortener app	Flutter , dart
Sanskar Kapoor	102017098	Encourage discord bot	JavaScript

Programming Language / Environment Experience

List the languages you are most comfortable developing in, **as a team**, in your order of preference. Many of the projects involve Java or C/C++ programming.

1. HTML, CSS
2. JavaScript
3. Python

Choices of Projects:

Please select **4 projects** your team would like to work on, by order of preference: *[Write at- least one paragraph for each choice (motivation, reason for choice, feasibility analysis, etc.)]*

First Choice	**Blood Bank** Every day 12,000 people in India die due to the sheer lack of donated blood. To overcome this, we think of creating a software to find local blood donors in any geographical area, helping many individuals to get the required blood group within reasonable time. We don't think of any economic/technical hurdles in our project, though the major hurdle is getting the local blood donors to sign up on our app.
Second Choice	**Airline Reservation and Tracker**

	A software for users to book airline tickets and also provides the users to track status of their flights. Moreover, this software provides personalized assistance to the users such as expected time for checking and on boarding.
Third Choice	**Food Delivery and Tracker** A software to order food from different restaurants in the nearby location and also a tracker to track your food.
Fourth Choice	**Price Compare** A software which helps the users to compare the price of different items from different e-commerce sites.

Additional Remarks/ Inputs

Please tell us about any other factors that we should take into consideration (e.g., if you really would like to work on a project for some particularly convincing reason).

1.2 Project Overview

Introduction

Every day 12,000 people in India die due to the sheer lack of donated blood. To overcome this, we think of creating a software to find local blood donors in any geographical area, helping many individuals to get the required blood group within reasonable time. We don't think of any economic/technical hurdles in our project, though the major hurdle is getting the local blood donors to sign up on our app.

Problem Statement

The logistic hurdles and stock management issues have always proved to be an obstacle in blood donation campaigns. Out of 234 million major operations carried out globally each year, many times blood requirements don't get fulfilled, leading to serious casualties. The patient's family members, who are already in a state of mental trauma, undergo pressure for finding blood donors, which turns out to be even difficult for rural areas. Moreover, it is difficult for the organizations to keep track of the total successful blood donations, and it is tiresome for a donor to visit centers manually for collecting their certificates or editing their personal information.

Objectives

The goal of this project is to develop an application for:

- Catering the needs of the patients by finding eligible donors/banks in the required location.
- Blood donors to find opportunities to donate blood through blood donation campaigns.
- Blood banks to manage their blood stock and a record of successful donations.
- To provide users a one stop destination for various different activities such as digital certification, accessing personal profile and getting their medical history.

Scope of the project

The features of the system will include the following:

- Registration
- Find nearby donors/banks.
- Emergency notifications by e-mail and sms
- Notifying donors about nearby blood camps
- Manage blood inventory
- Record of Donor history
- Donation certification

2.1 USE CASE DIAGRAM

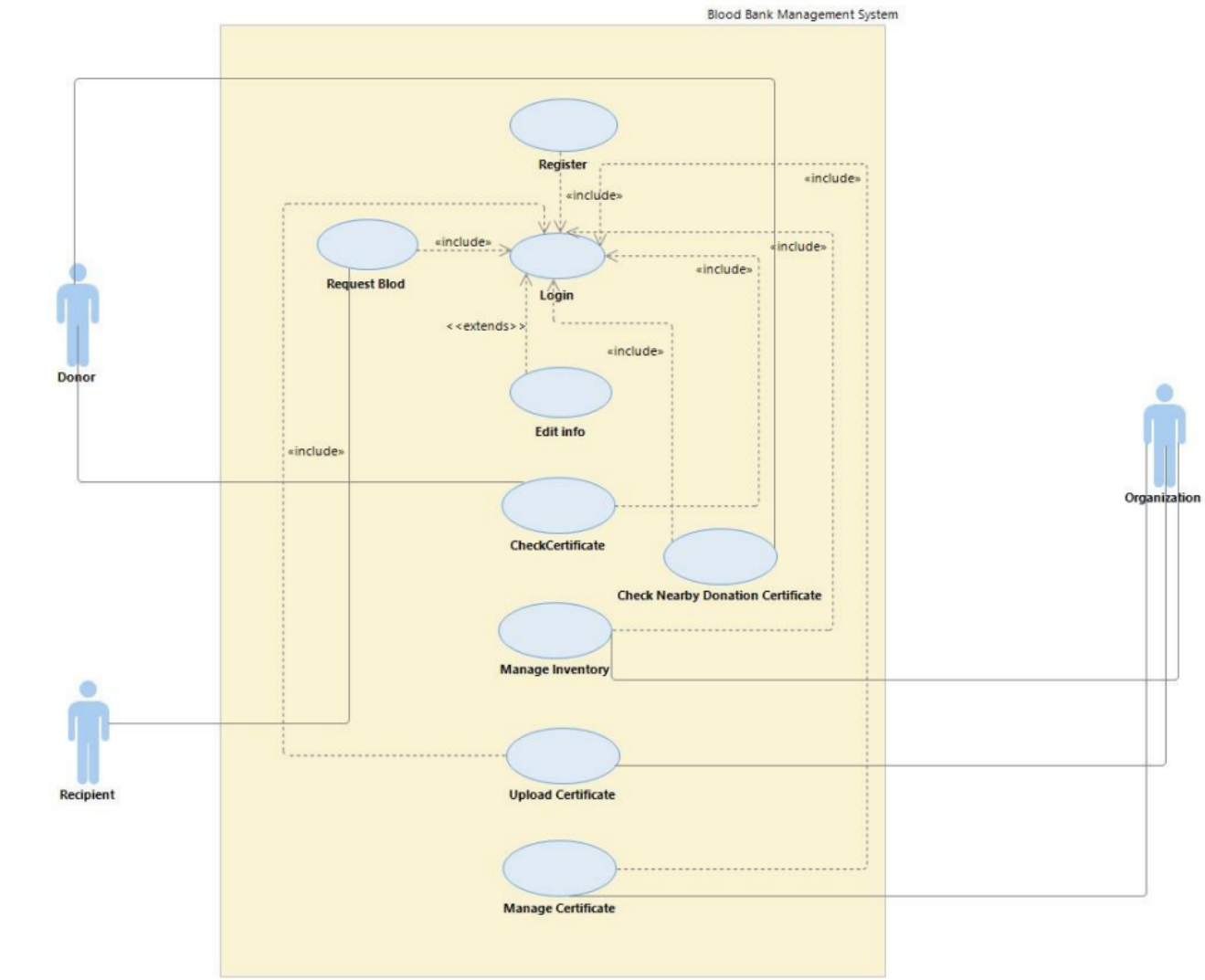

Fig 2.1: Use case diagram

2.2 ACTIVITY DIAGRAM

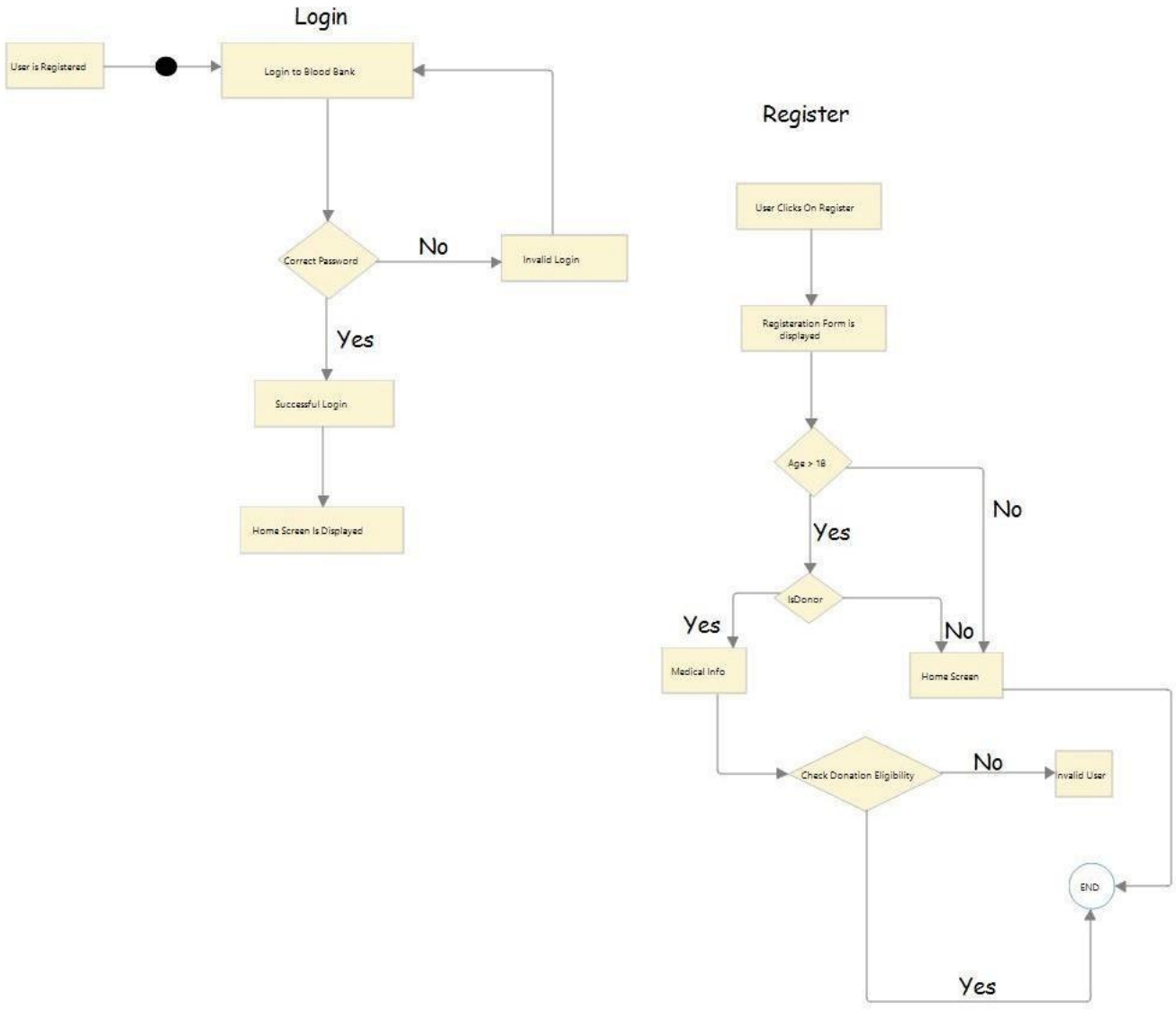

Fig 2.2: Activity diagram

2.2 SWIMLANE DIAGRAM

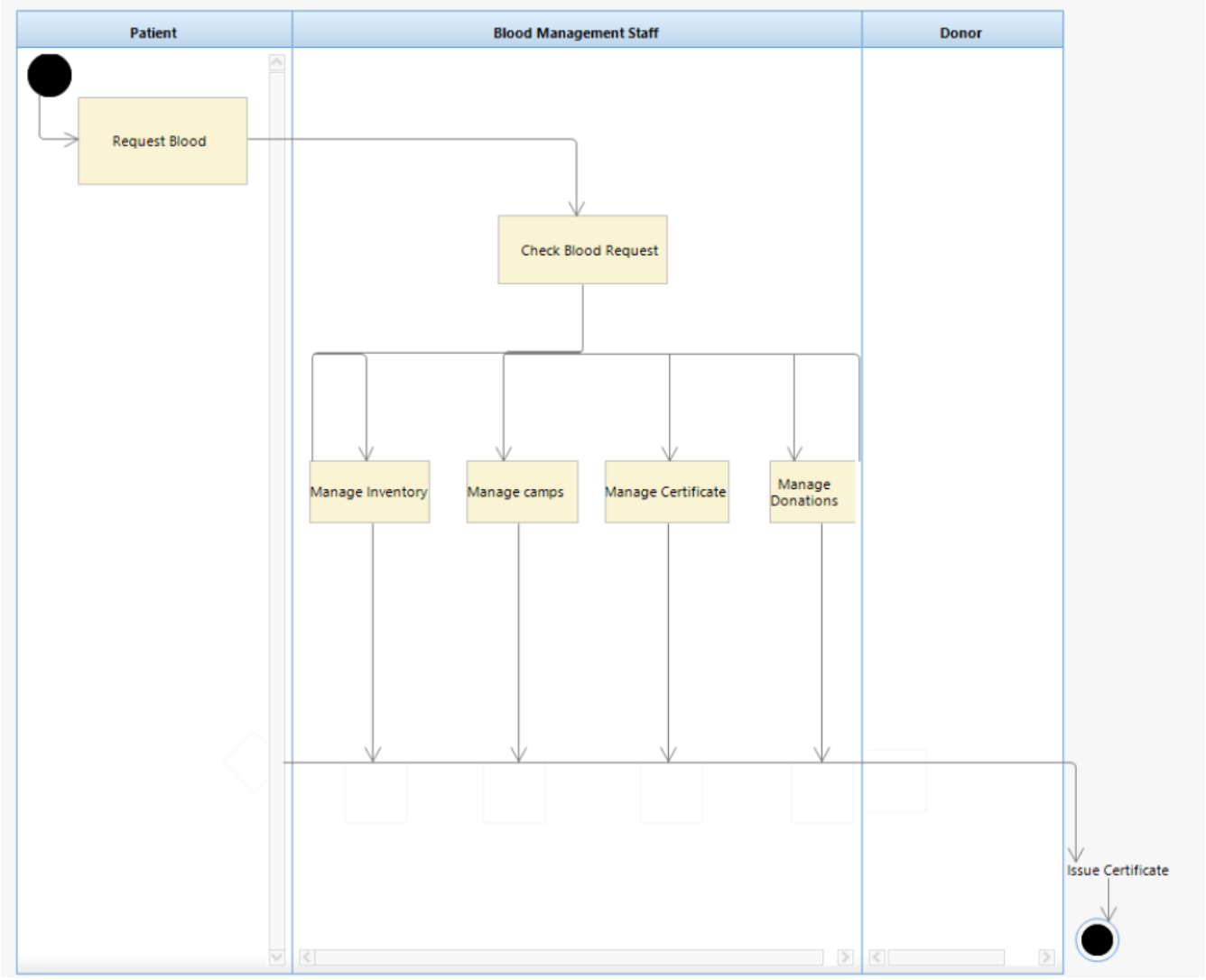

Fig 2.3: Swimlane diagram

2.3 FLOW DIAGRAM

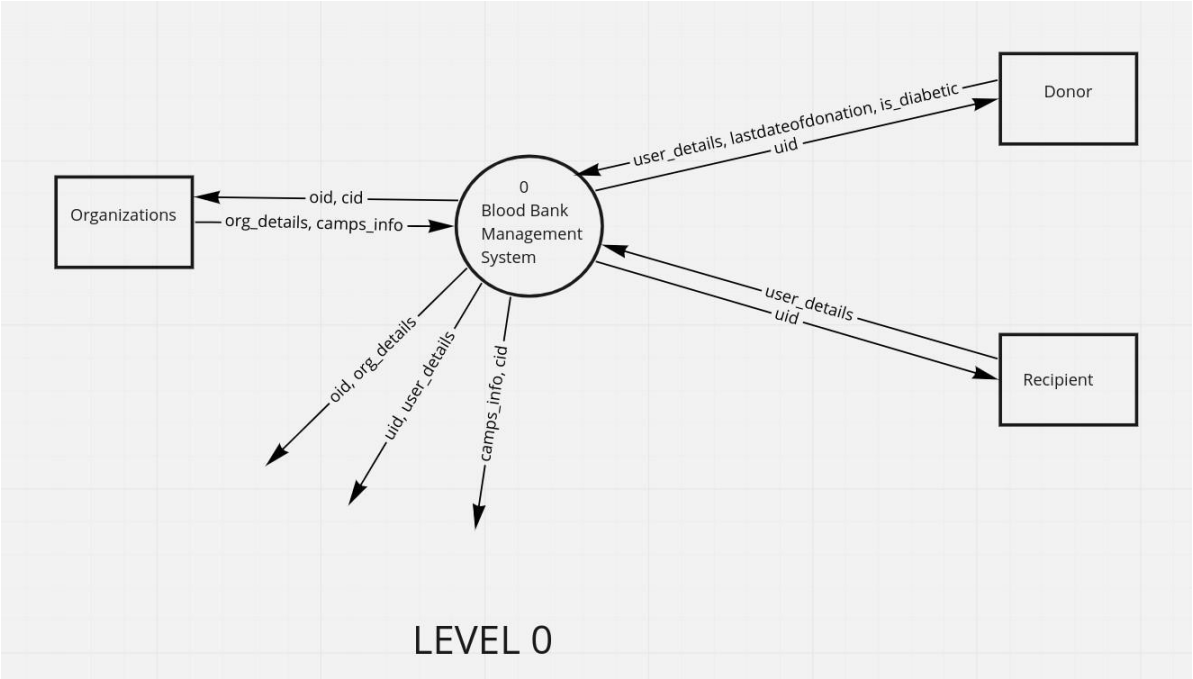

Fig 2.4.1: Data flow diagram level 0

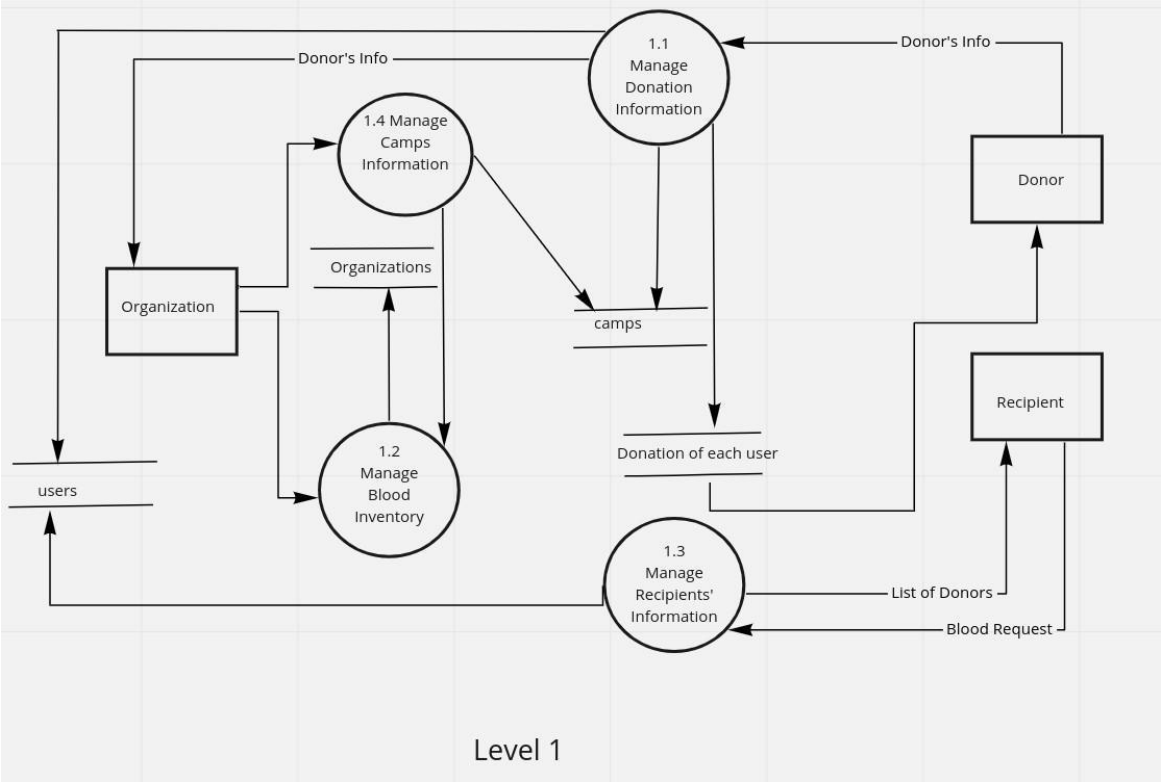

Fig 2.4.2: Data flow diagram level 1

Software Requirements Specification Document

Version 1.0

Blood Bank Management System

TABLE OF CONTENTS

1. Introduction

1.1 Purpose of this Document

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. A web based blood donation system is mainly used for helping the patient who needs blood . So this SRS document consists of a simple explanation about the system and its features

1.2 Scope of the Development Project

The purpose of this system is to create a convenient and easy-to-use online system for users, trying to get or donate blood and for organizations to manage their blood inventory and also organize blood donation camps. We hope to provide a comfortable user experience for each user type. A user can search for nearby donors or blood banks for a particular blood type and contact them for the same. The donors will find opportunities to donate blood in nearby locations in camps or to the patients. The donors will be notified by email/SMS in case a request is marked as an emergency. Organizations can manage their blood inventory and donors. Camps can be organized for donors in nearby locations. All this data that's being accessed will be stored in a database server.

The system must be able to perform the following operations:

1. **Register a user as a donor/patient:** A user will be able to register as a donor / patient by filling a simple form for the same.
2. **Register an organization:** An organization can register themselves and they need to verify themselves.
3. **Requesting for blood:** A user can request for a particular blood group and get a list of nearby donors/blood banks.
4. **Emergency requests**: A request can be marked as an emergency and the donors will be notified by email/SMS.
5. **Nearby camps**: Donors will be shown a list of nearby donation camps.
6. **Managing blood inventory**: Organizations will be shown a page to edit and update their blood inventory.
7. **Organizing a camp**: Organizations can organize camps and add related information on the system.
8. **Maintaining a database of the donations**: Organizations can maintain a database for the donations made to them.
9. **Uploading and downloading certificates:** Organizations can upload certificates for each donor after a successful donation and the donors will be able to see them and download them onto their system.

1.3 Definitions, Abbreviations and Acronyms

Table 1 gives the full form of most commonly used abbreviations in this SRS document

Table 1: Full form of the most commonly used abbreviations

Serial number	Abbreviation	Meaning
1	**Oid**	Organization id: which is a unique id given to every organization during commencement with the Blood Bank
2	**Uid**	User id: which is given to every user who register's in our Blood bank if he/she wants to donate or requires blood.
3	**Cid**	Camp's id: which is the id given to every blood donation camp.
4	**Certid**	Certificate id: a unique id given to each certificate
5	**Bid**	Blood id: a unique id given to each sample of blood collected.

1.4 References

1. *Blood bank information*. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Blood_bank
2. *Blood bank NGOs*. (n.d.). Retrieved from https://www.bloodconnect.org/
3. *Blood bank organisations*. (n.d.). Retrieved from http://www.sankalpindia.net/blood-donation-organization

1.5 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 2 gives the functional requirements, data requirements and constraints and assumptions made while designing the multi-utility system. It also gives the user viewpoint of product use. Section 3 gives the specific requirements of the product. Section 3.0 also discusses the external interface requirements and gives detailed description of functional requirements.

2. Overall Description

This section will give an overview of the entire system. This will show how the system will work and introduce the basic functionality of it. It will describe all the users who will access the system and what functionality is available for each type of user. The constraints and assumptions for the system will also be discussed.

2.1 Product Perspective

This system includes both offline and online components. The collection of blood will be manual through Blood Donation Camp. The donor can either register on the Blood Bank website on his own or can visit the Blood Donation Camp and the responsible authority at the camp can do the registration for the donor. An online database is maintained with all the information about the donors. Once the blood is collected it is stored in a safe place. An online Blood Inventory Database is maintained as well for the Blood Units collected. Patients can request for blood according to their needs, and nearby availability is shown to the patient in terms of nearby donors and blood banks. A patient can also put in an emergency request whereby the available donors in the nearby location are notified by email/sms for the same.

2.2 Product Function

The product should be able to perform the following functions

1. Register a user as a donor or a patient.
2. Register an organization and verify them
3. Manage an organization's blood inventory
4. Manage camps and update information related to the same
5. Manage the donations made to an organization
6. Request blood to a particular blood group
7. Mark a request as an emergency request.
8. Show the nearby camps and patients to the donors
9. Uploading of certificates by organizations.
10. Downloading certificates for donors.

2.3 User Characteristics

The following are the users interacting with each other in this system:

1. User (can be a donor or a patient)
2. A member from the organization to manage their inventory and information related to the camps

2.4 General Constraints, Assumptions and Dependencies

The Donor and the acceptor are constrained to create an account first to avail the services. The internet connection is also a constraint for this web application.

It is assumed that the users have enough resources to run the web application i.e a mobile phone or a computer that supports the required functions.

The front end is designed with the help of HTML, CSS, Bootstrap and Javascript. The backend is done in the Django framework of python.

3. Specific Requirements

3.1 Detailed Description of Functional Requirements

Table 3 shows a template that will be used to describe functional requirements for all the types of users.

Table 2: Template for describing functional requirements

Purpose	A description of the functional requirements and its reasons
Inputs	What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input.
Processing	Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations
Outputs	The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced as output.

3.1.1 Functional requirements for Registration Screen for patient/donor

Table 3: Functional requirements for Registration Screen for patient/donor

Purpose	This screen displays a registration form for the user to register themselves as a donor or patient
Inputs	The screen requires the following inputs: User information such as email, phone number, DOB etc. and whether they are registering as a donor or a patient. This data will arrive in the form of text.
Processing	Registers the user and checks their age validity and also their medical conditions in case the user is a donor.
Outputs	If the registration is successful then the user is taken to the next screen. And in case age validation is false, then an error is shown for the same.

3.1.2 Functional requirements for Registration Screen for organization

Table 4: Functional requirements for Registration Screen for organization

Purpose	This screen displays a registration form for a member of an organization to register the organization.
Inputs	The screen requires the following inputs: Organization information such as name, contact info, address etc. and some kind of verification of the organization. This data will arrive in the form of text.
Processing	Registers the organization and verifies the organization if the verification is successful.
Outputs	If the registration is successful then the organization is taken to the next screen.

3.1.3 Functional requirements for requesting blood of a particular type

Table 5: Functional requirements for requesting blood of a particular type

Purpose	This screen helps a user to search for donors in the nearby location for a particular blood group.
Inputs	The screen requires the following inputs: Blood group. This data will arrive in the form of text.
Processing	Queries the database for donors and blood banks for a particular blood group.
Outputs	Displays the nearby donors and blood banks according to the distance and the availability

3.1.4 Functional requirements for managing blood inventory

Table 6: Functional requirements for managing blood inventory

Purpose	This screen helps an organization to maintain their blood inventory
Inputs	The screen requires the following inputs: Blood group, date of storage, amount, donorID, bloodID(in case of editing). This data will arrive in the form of text.
Processing	Stores/updates the details of blood inventory in a database.
Outputs	Displays a message if the updation of the database is successful or not.

3.1.5 Functional requirements for managing information related to camps

Table 7: Functional requirements for managing information related to camps

Purpose	This screen helps an organization to manage information related to camps organized by them.
Inputs	The screen requires the following inputs: Information related to the particular camp such as name,location,time,etc. This data will arrive in the form of text.
Processing	Stores/updates the details of camp in a database.
Outputs	Displays a message if the updation of the database is successful or not.

3.1.6 Functional requirements for managing certificates

Table 8: Functional requirements for managing certificates

Purpose	This screen helps an organization to manage certificates of donors.
Inputs	The screen requires the following inputs: Certificate. This data will arrive in the form of a pdf.
Processing	Stores/updates the data in the database.
Outputs	Displays a message if the updation of the database is successful or not.

3.2 Non-functional Requirements

1. Availability : The system will be available 24/7.

2. Security: Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

4. Correctness: The Blood Unit sent by the Blood Bank should be matched with the requested Blood Unit by the Hospital, which should reach the correct destination(Requested Hospital).

5. Maintainability: The Blood Inventory Manager should maintain correct records of the Blood Inventory Stock.

6. Usability: The system can be used by many organizations to manage their blood inventories.

7. Extensibility: Requirements for website extensibility in case there is a need to add new functional requirements.

3.3 Performance requirements

- The software can be run from any computer with access to the internet.
- The software supports simultaneous user access.
- For normal conditions, most of the transactions would be processed in less than 5 seconds.

3.4 Logical Database Design

Figure 1 shows the E-R diagram of the entire system.

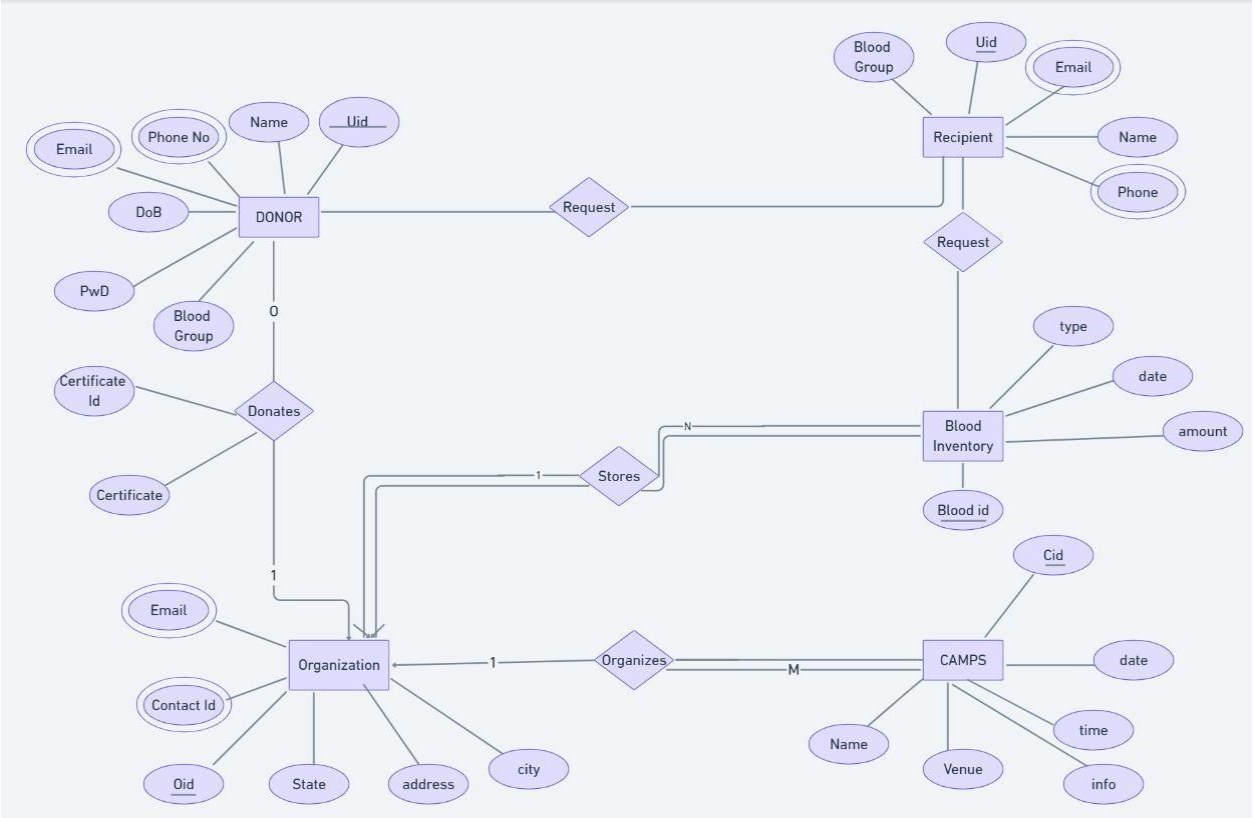

Figure 1: E-R diagram of the blood bank management system

3.5 Quality attributes

The product is targeted towards a wide variety of users such as donor, recipient,inventory manager, etc. The product must load quickly. It must also tolerate a wide variety of input possibilities from a user, such as incorrect responses.

3.6 Other requirements

None at this time

4. Change History

200209	**Version 1.0 – Initial Release**

5. Document Approvers

SRS for blood bank management system approved by:

_____

(name)
Designation:
Date:

3.1 CLASS DIAGRAM

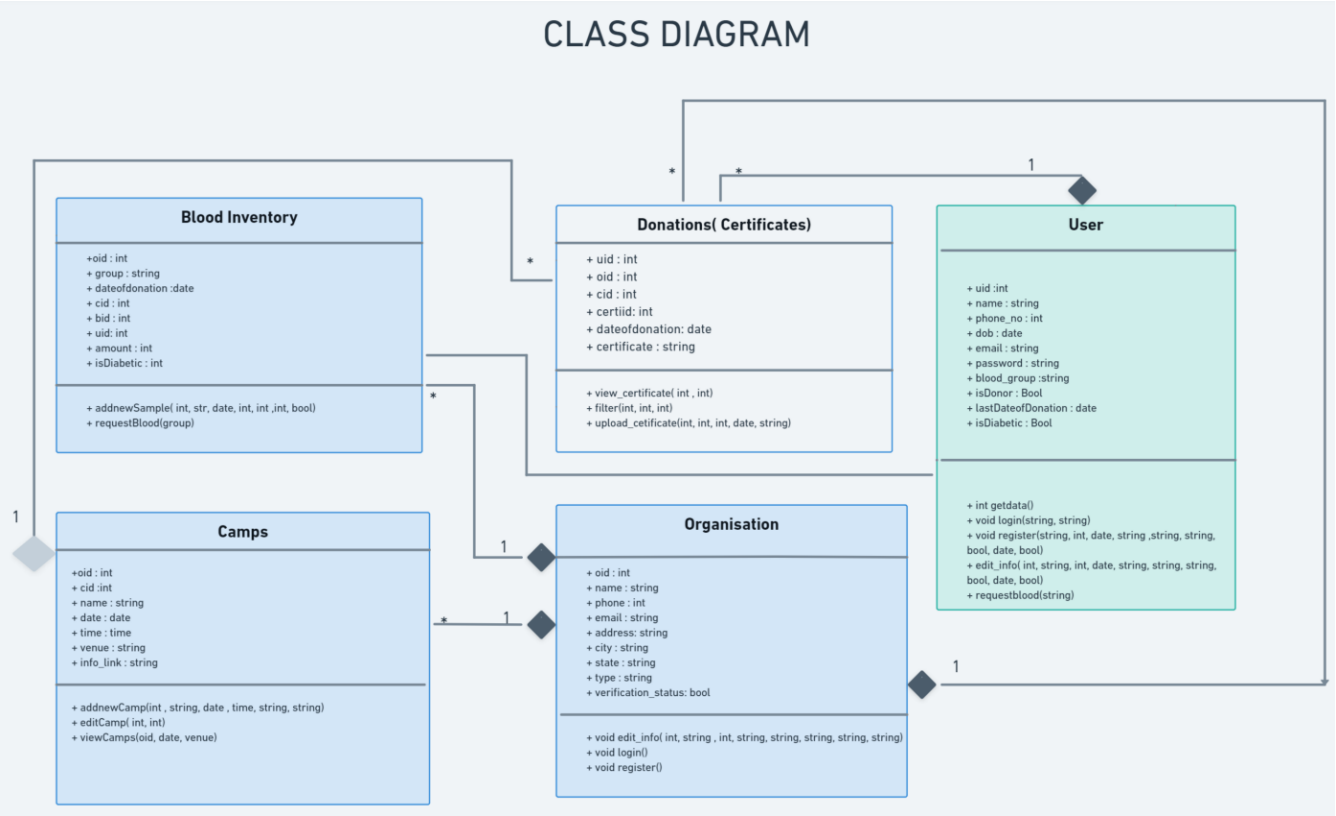

Fig 3.1: Class diagram

3.2 SEQUENCE AND COLLABORATION DIAGRAM

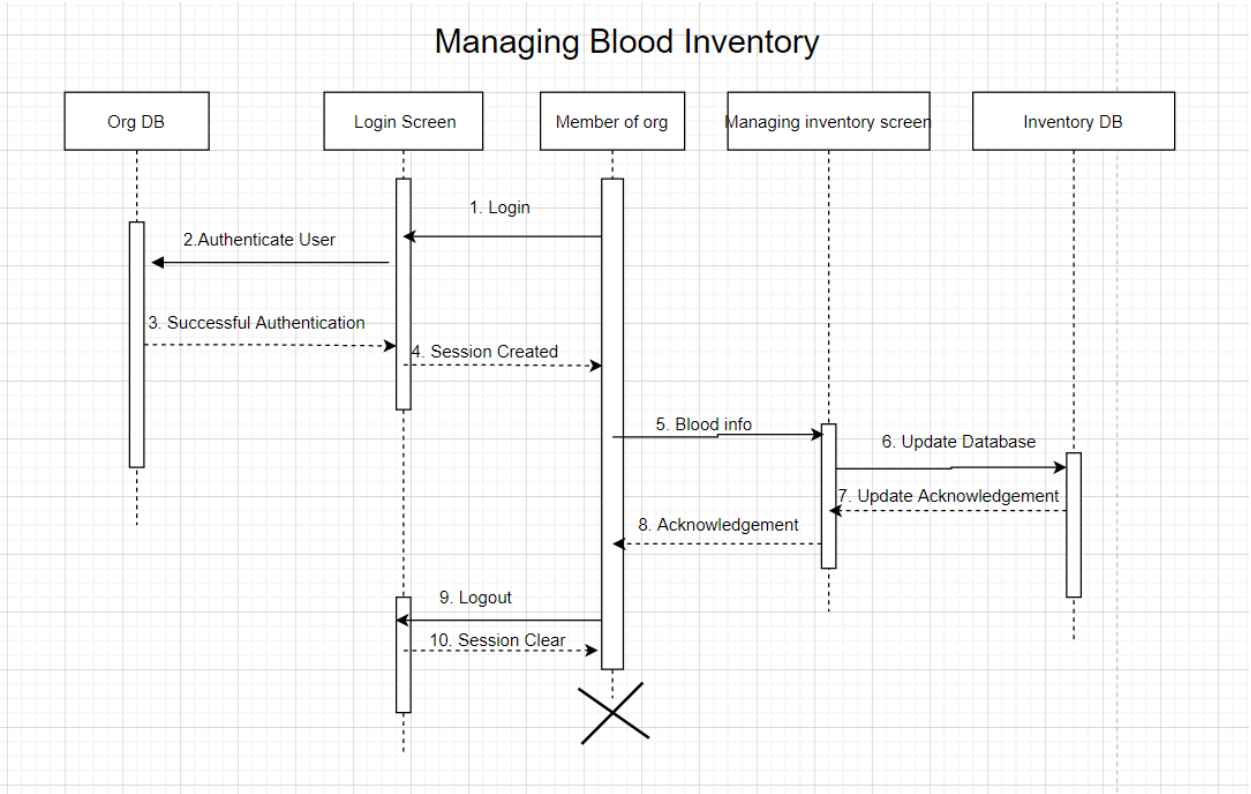

Fig 3.2.1 : Sequence diagram for Managing blood inventory

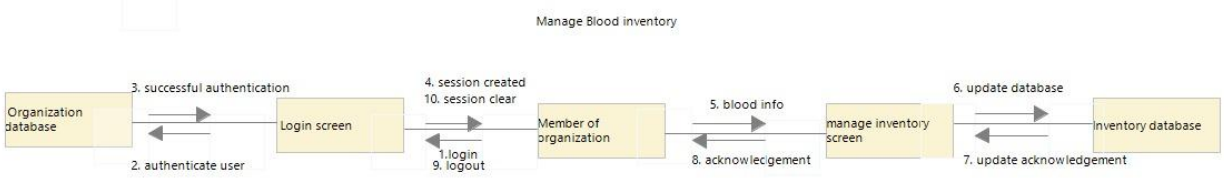

Fig 3.2.2 : Collaboration diagram for Managing blood inventory

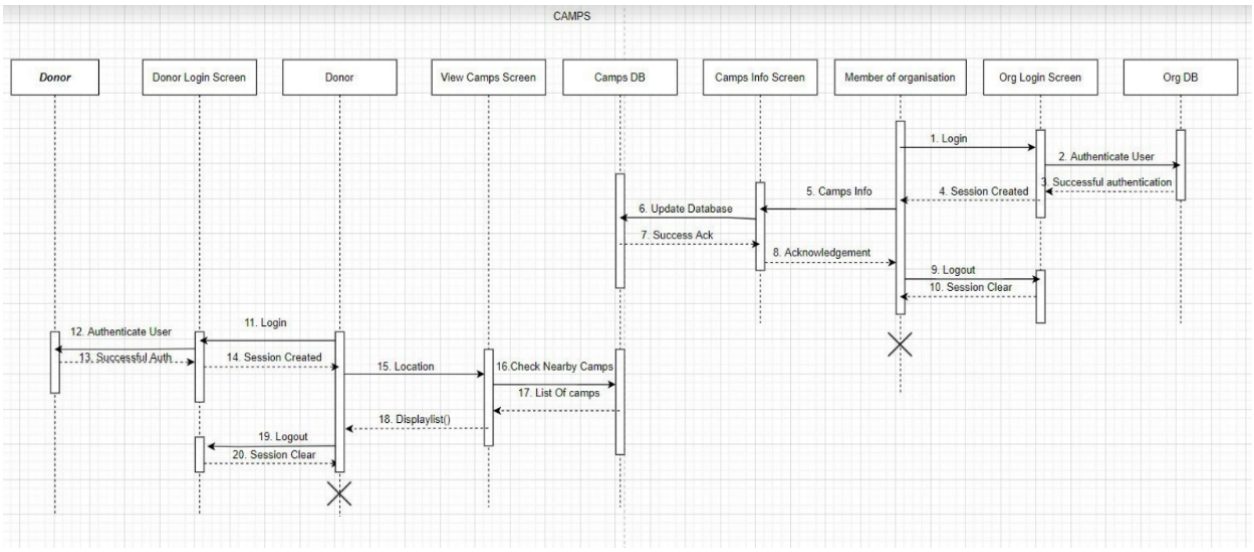

Fig 3.2.3 : Sequence diagram for managing camps info and displaying camps data

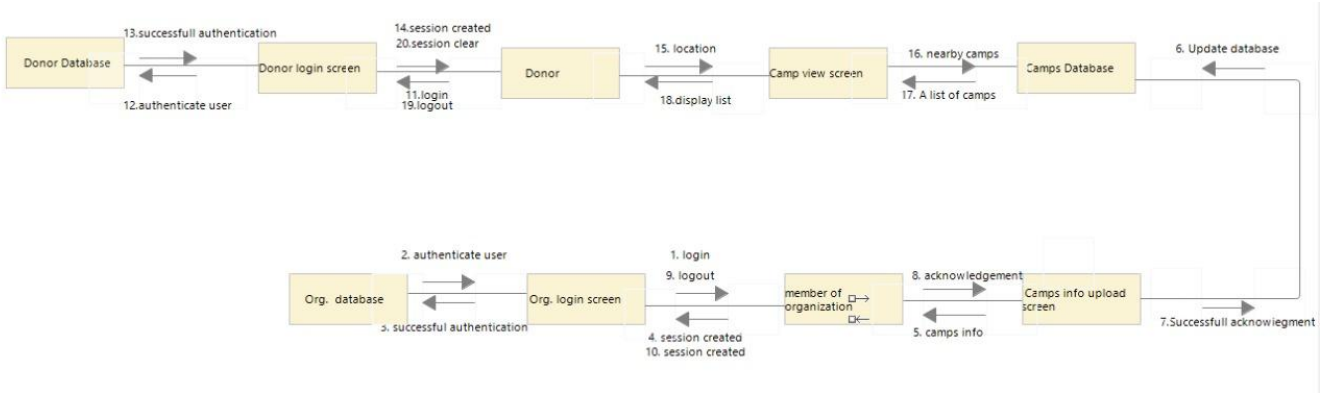

Fig 3.2.4 : Collaboration diagram for Managing camps info and displaying camps data

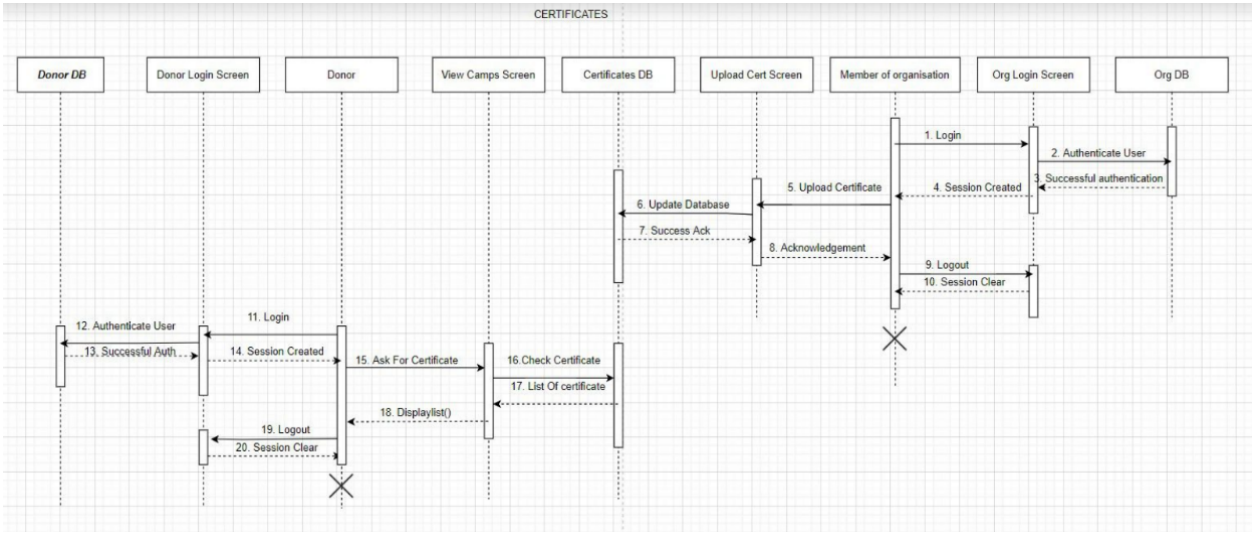

Fig 3.2.5 : Sequence diagram for managing certificates

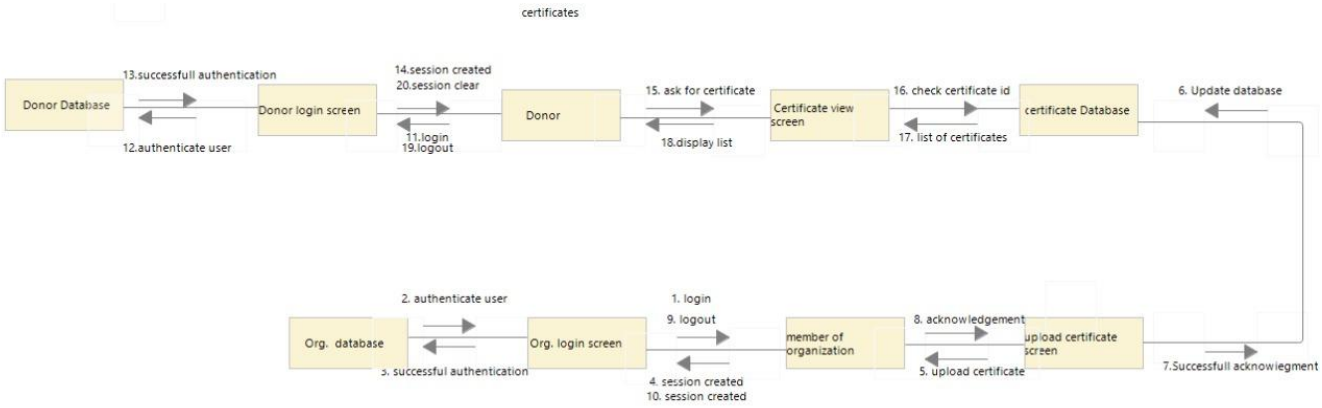

Fig 3.2.6: Collaboration diagram for managing certificates

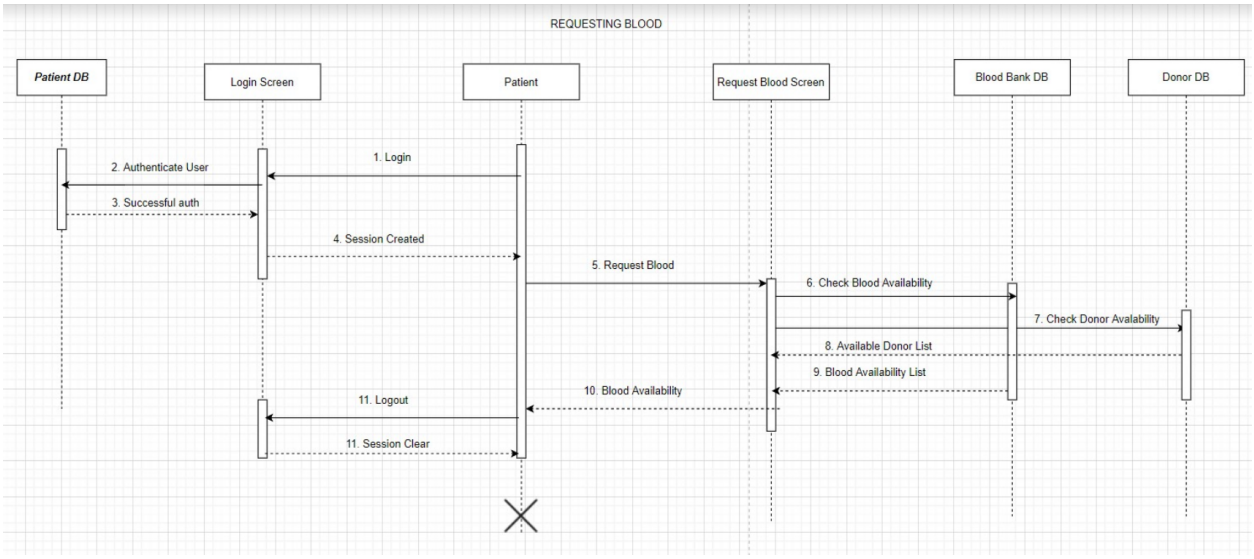

Fig 3.2.7 : Sequence diagram for requesting blood

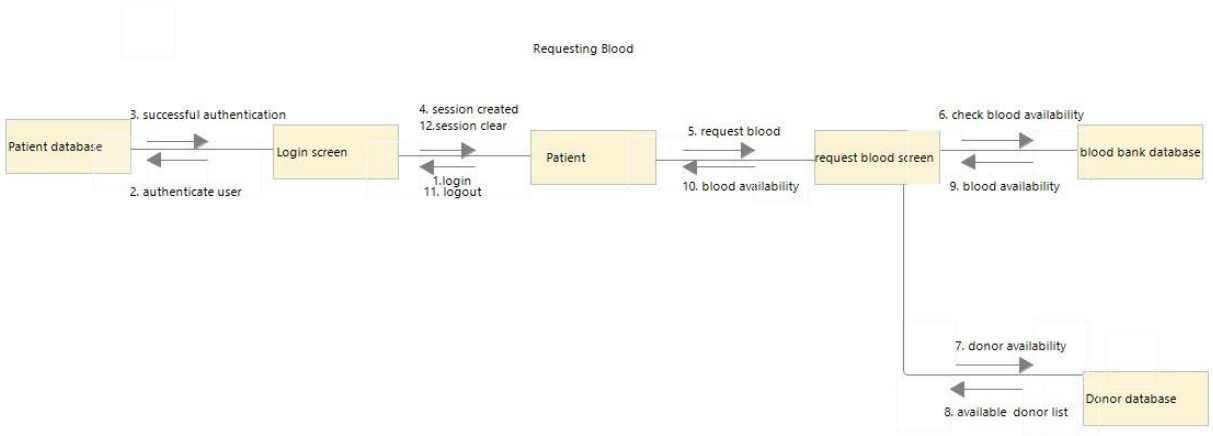

Fig 3.2.8 : Collaboration diagram for requesting blood

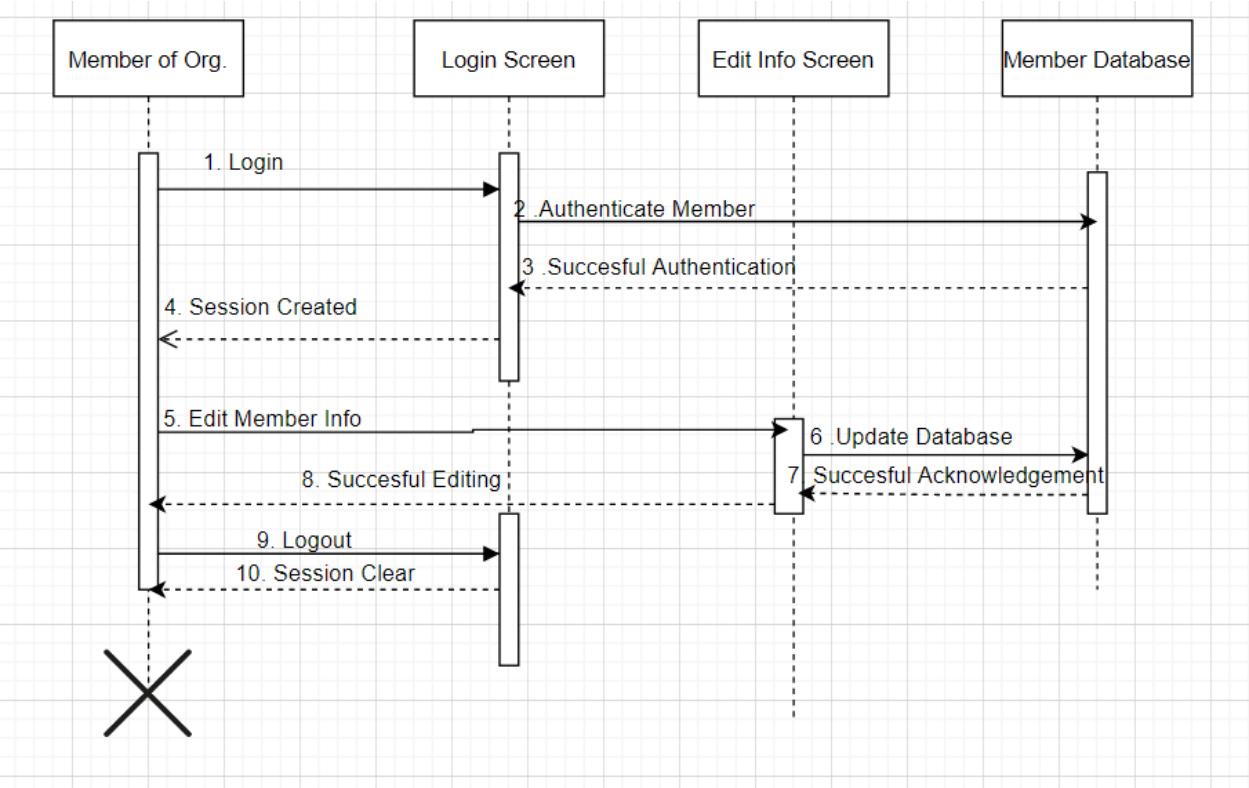

Fig 3.2.9 : Sequence diagram for editing user information

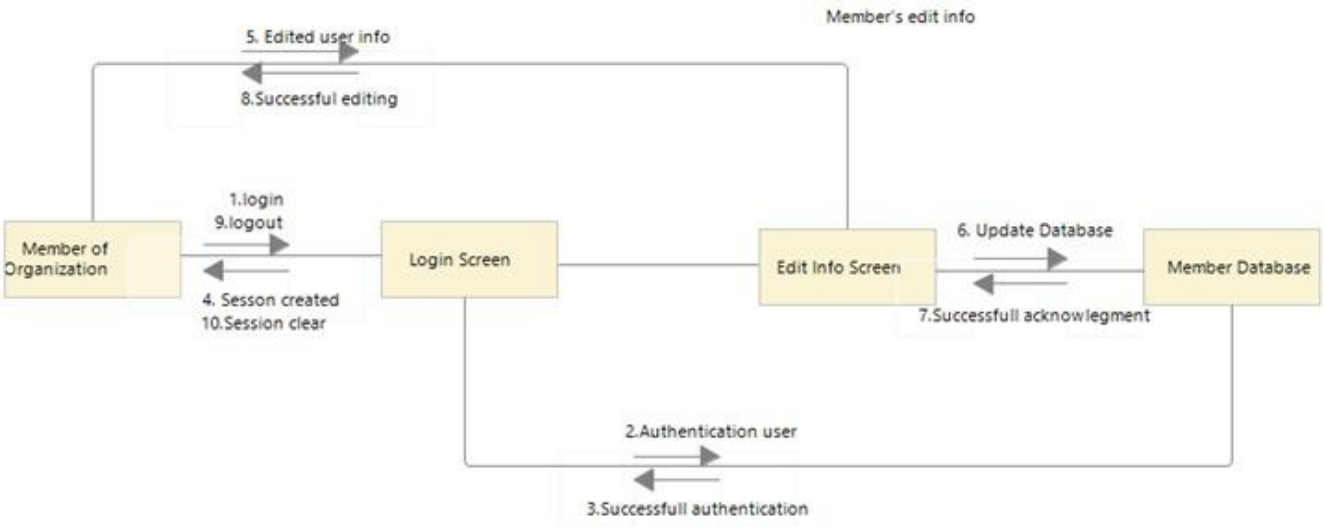

Fig 3.2.10 : Collaboration diagram for editing user information

3.3 ER DIAGRAM

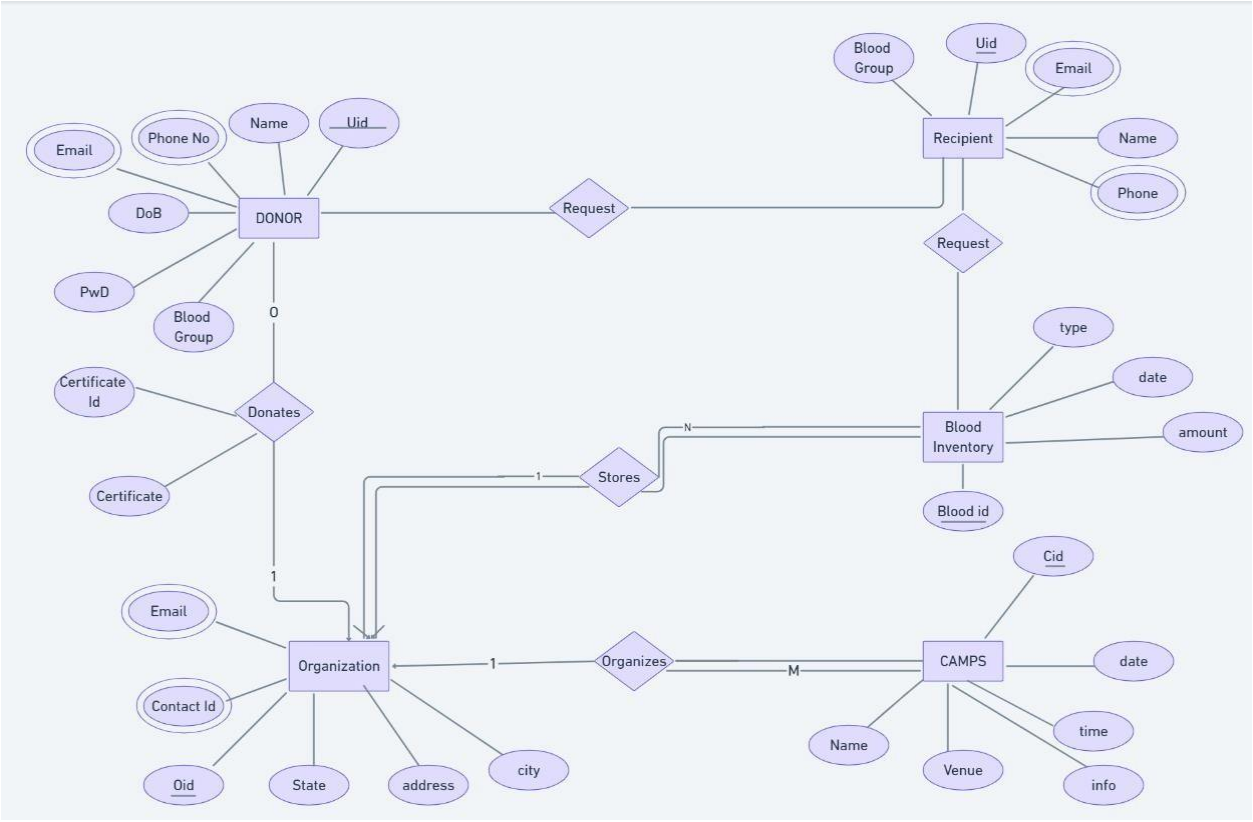

Fig 3.3: ER diagram for blood bank management system

3.4 STATE CHART DIAGRAM

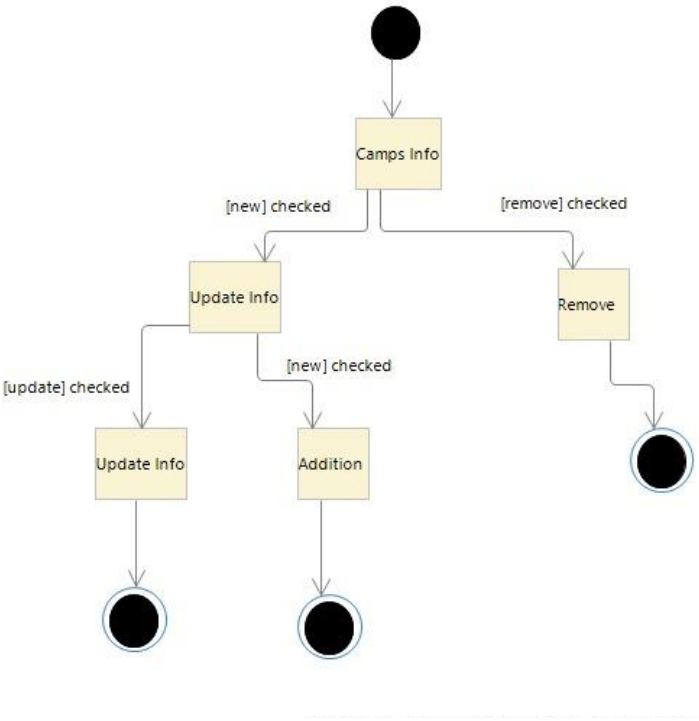

Fig 3.4.1: State chart diagram for camps information

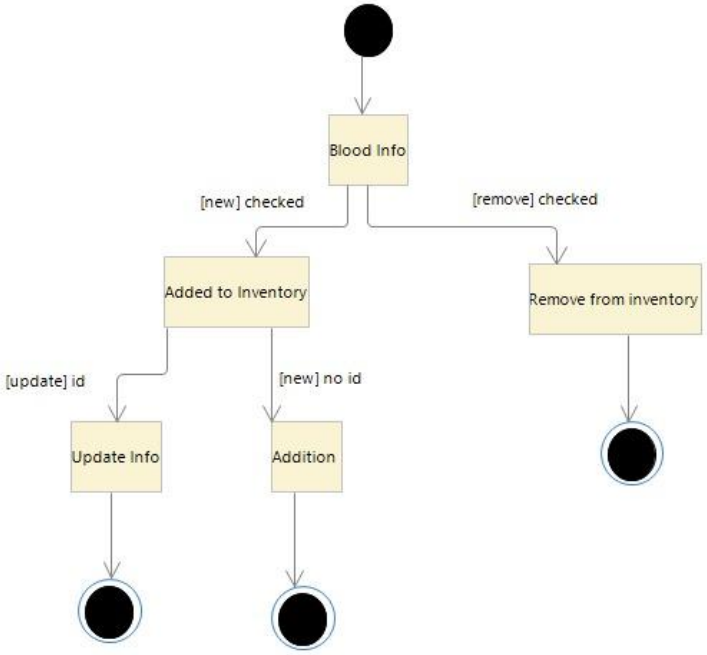

Fig 3.4.2: State chart diagram for blood information

4.1 COMPONENT DIAGRAM

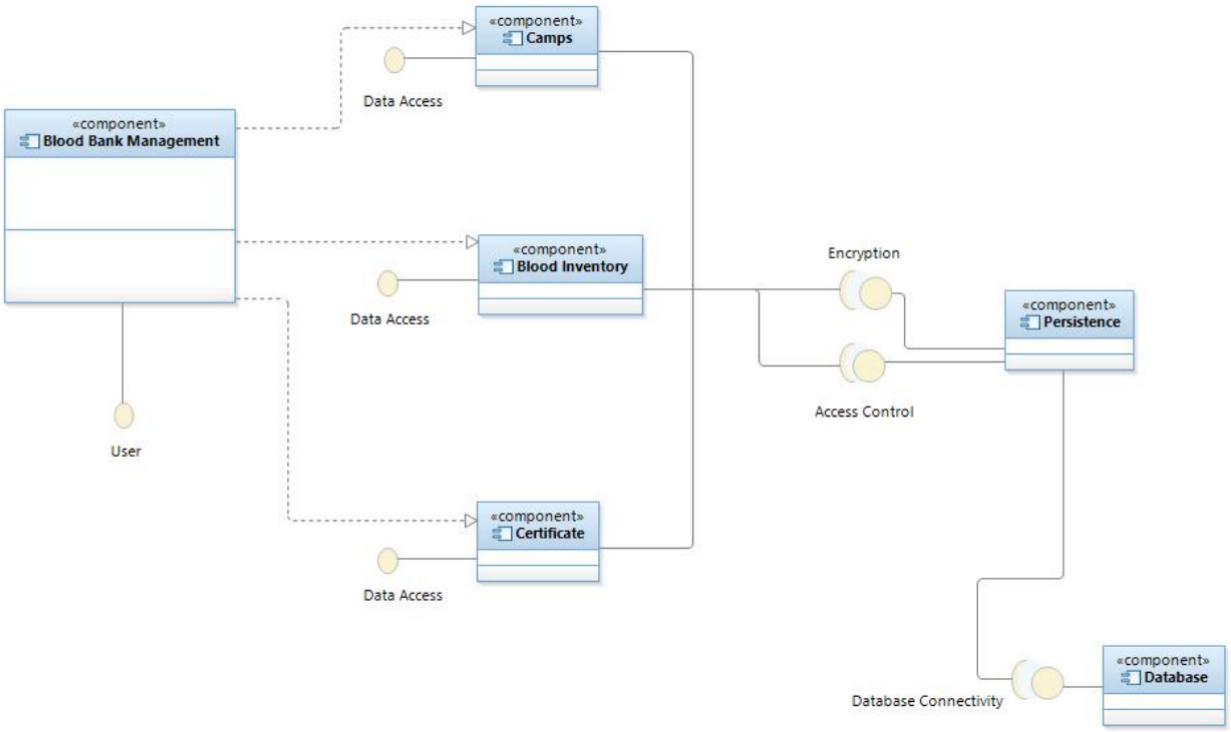

Fig 4.1: Component diagram

4.2 DEPLOYMENT DIAGRAM

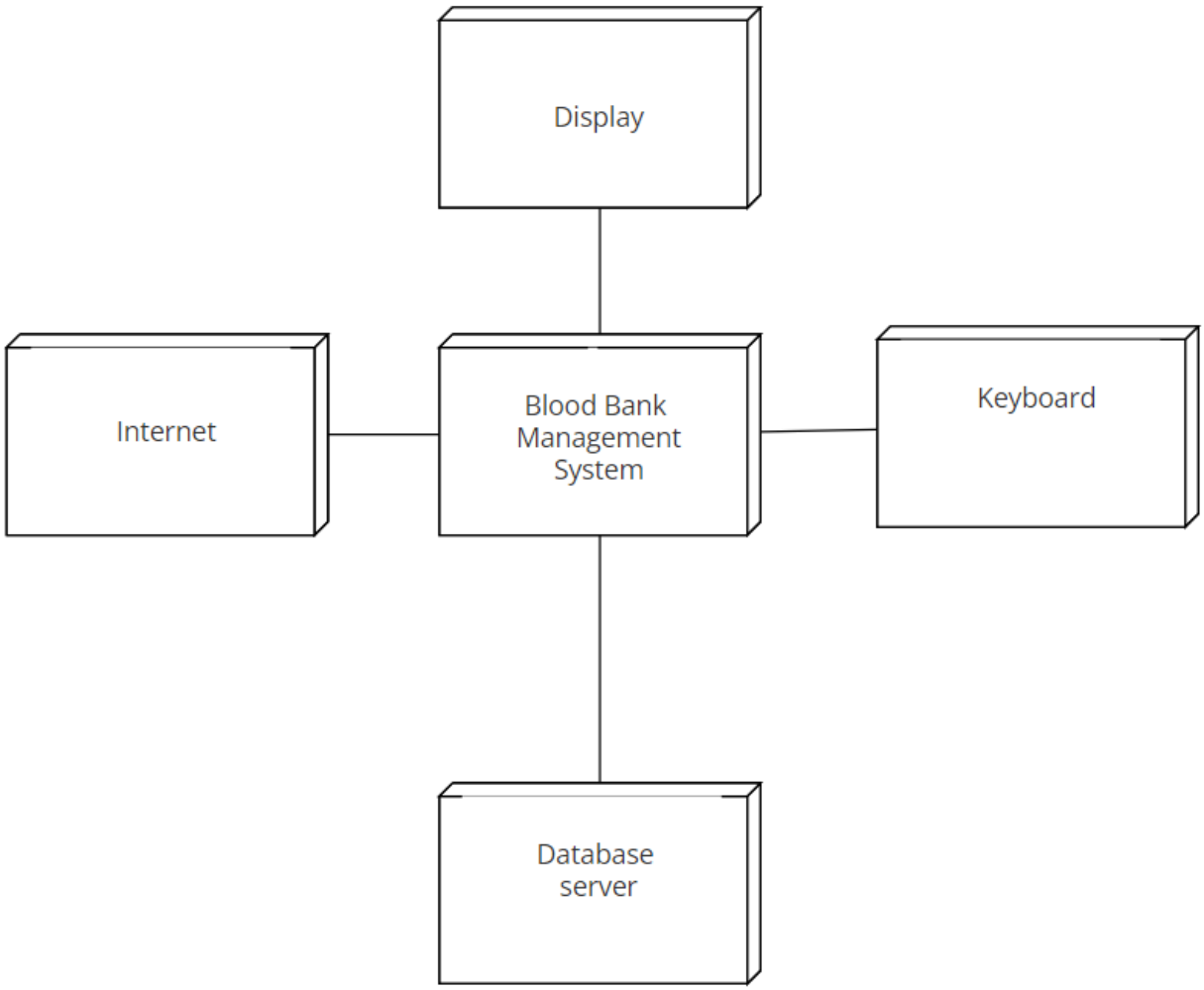

Fig 4.2 Deployment diagram

4.3 SCREENSHOTS

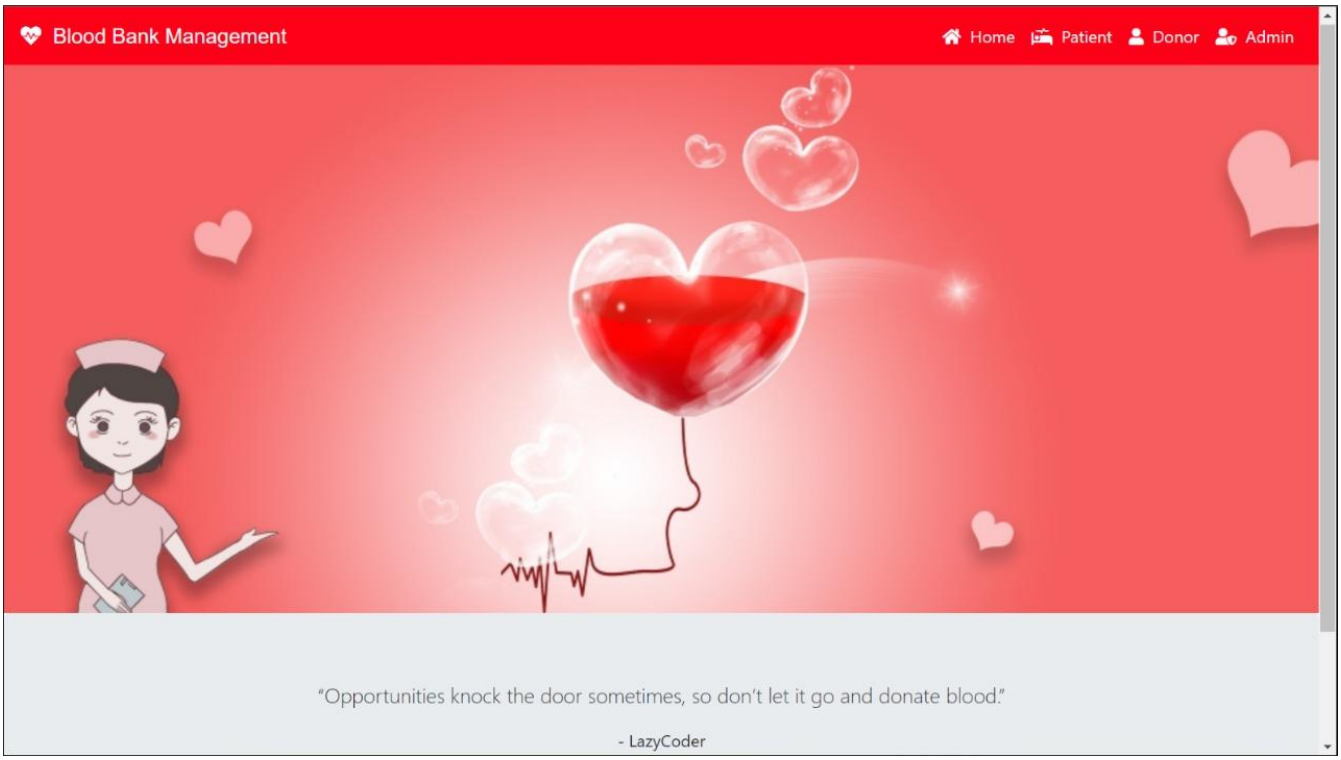

Fig 4.3.1: Home Screen

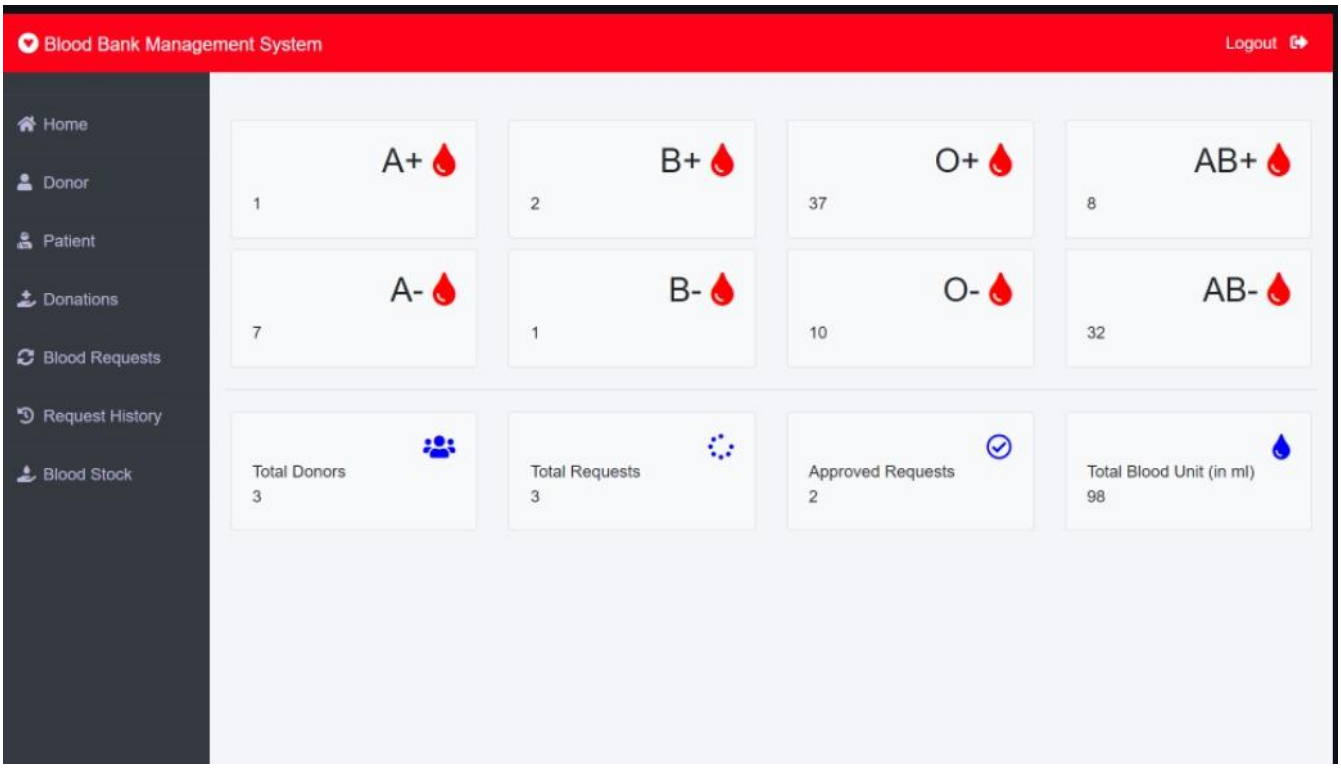

Fig 4.3.2: Admin home screen

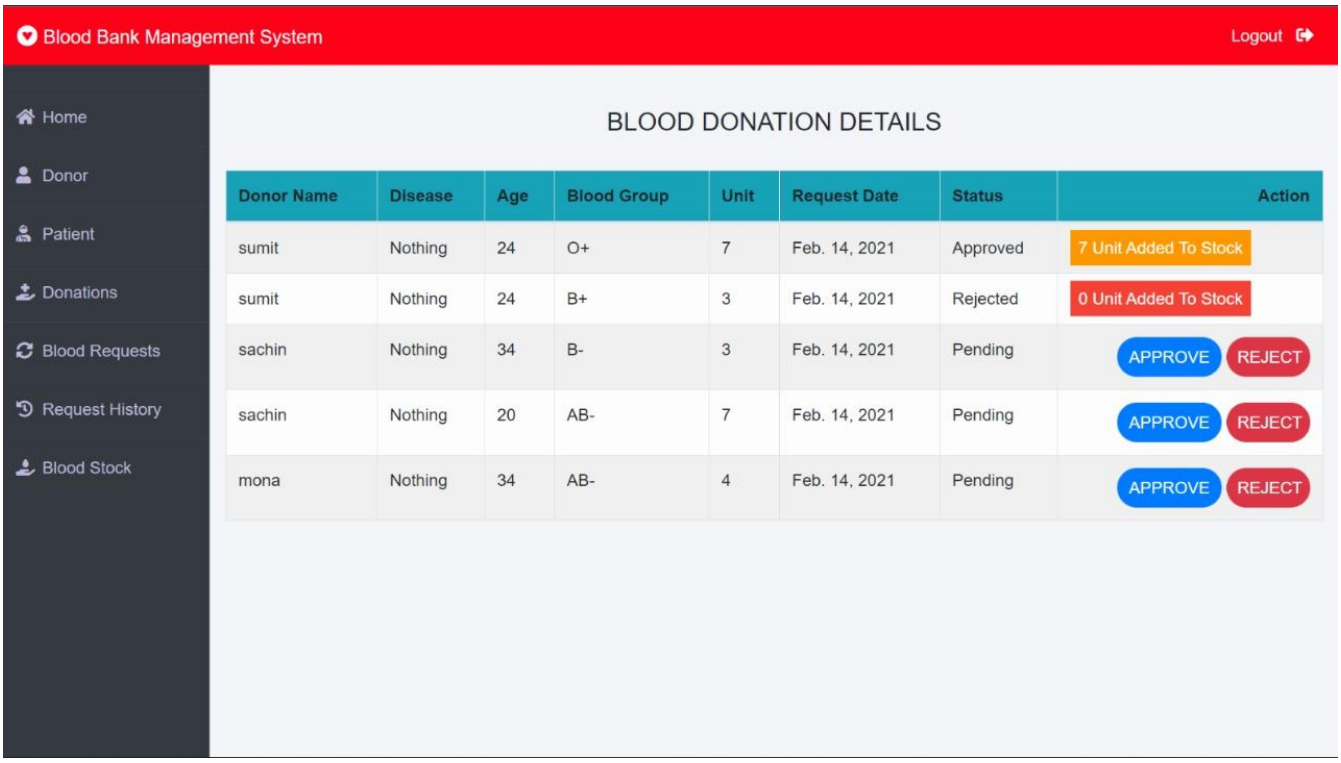

Fig 4.3.3: Blood donation details (admin)

Fig 4.3.4: Blood request screen (admin)

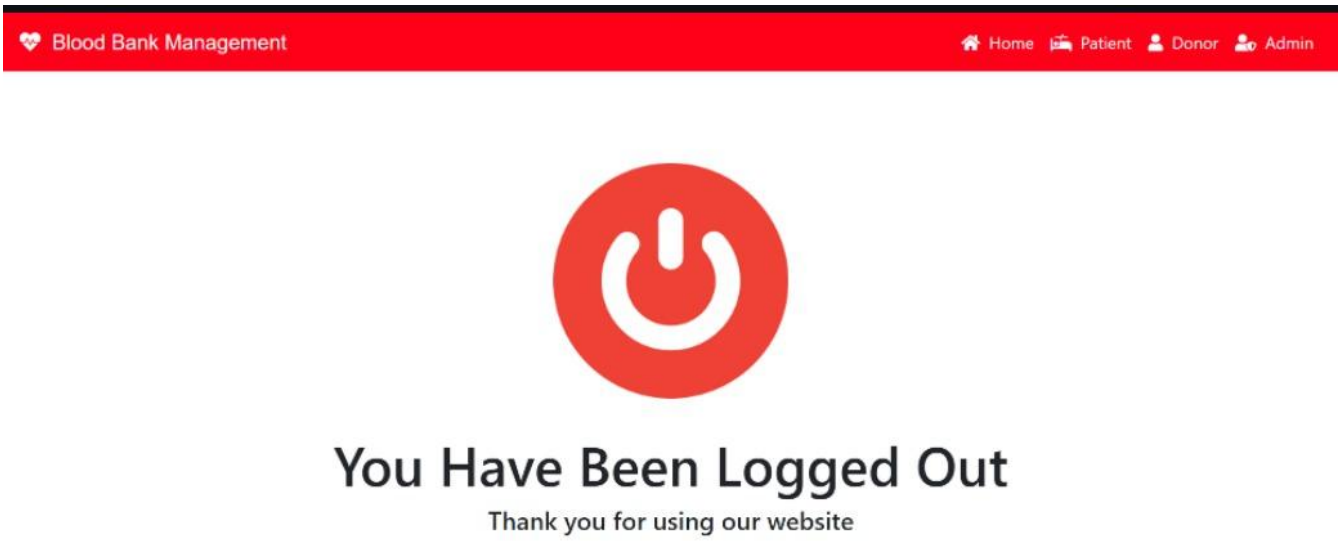

Fig 4.3.5: Logout screen

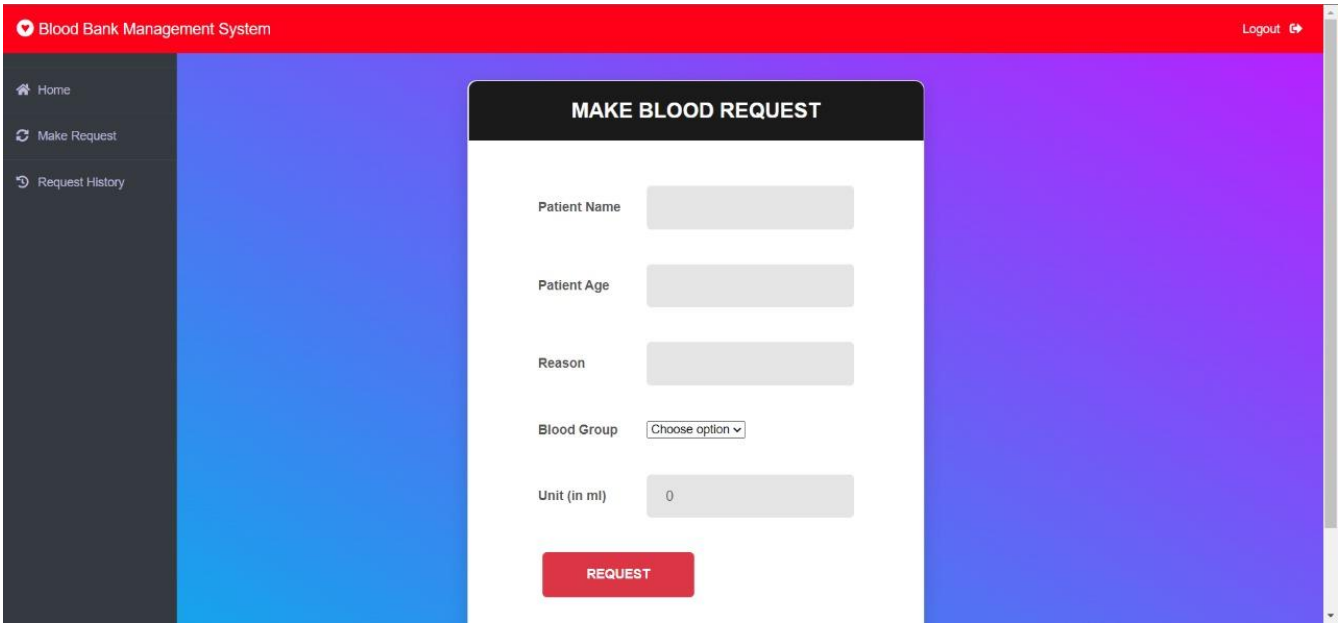

Fig 4.3.6: Patient request blood screen

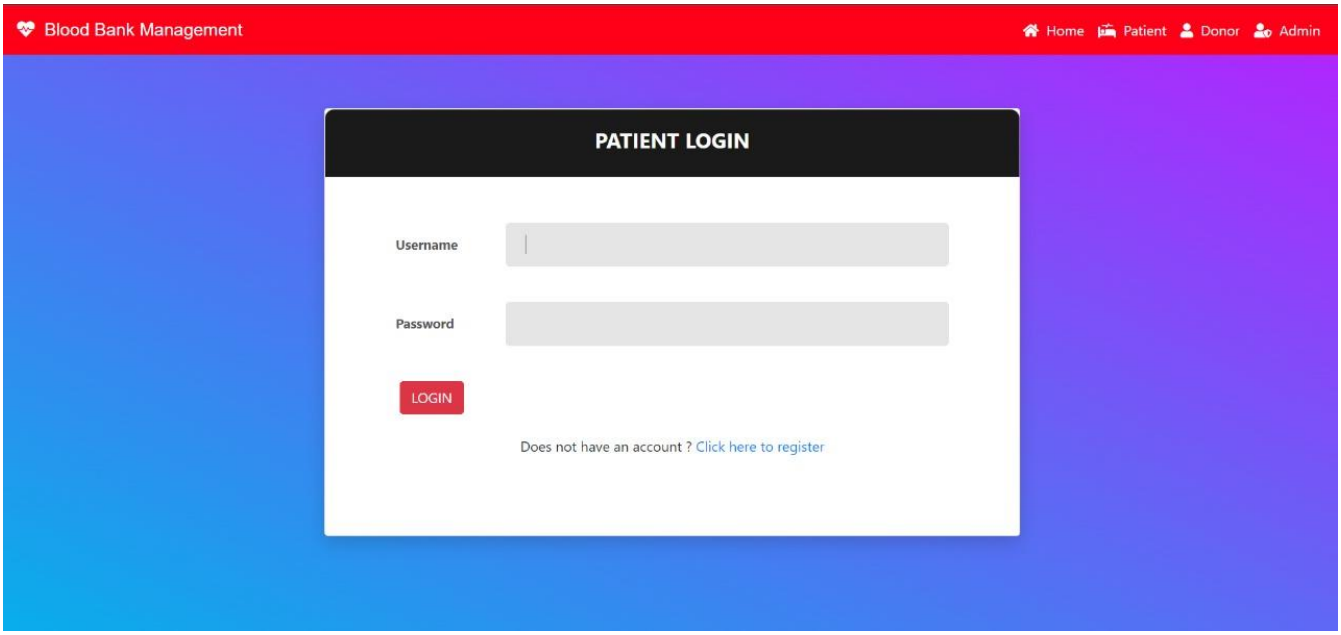

Fig 4.3.7: Patient login

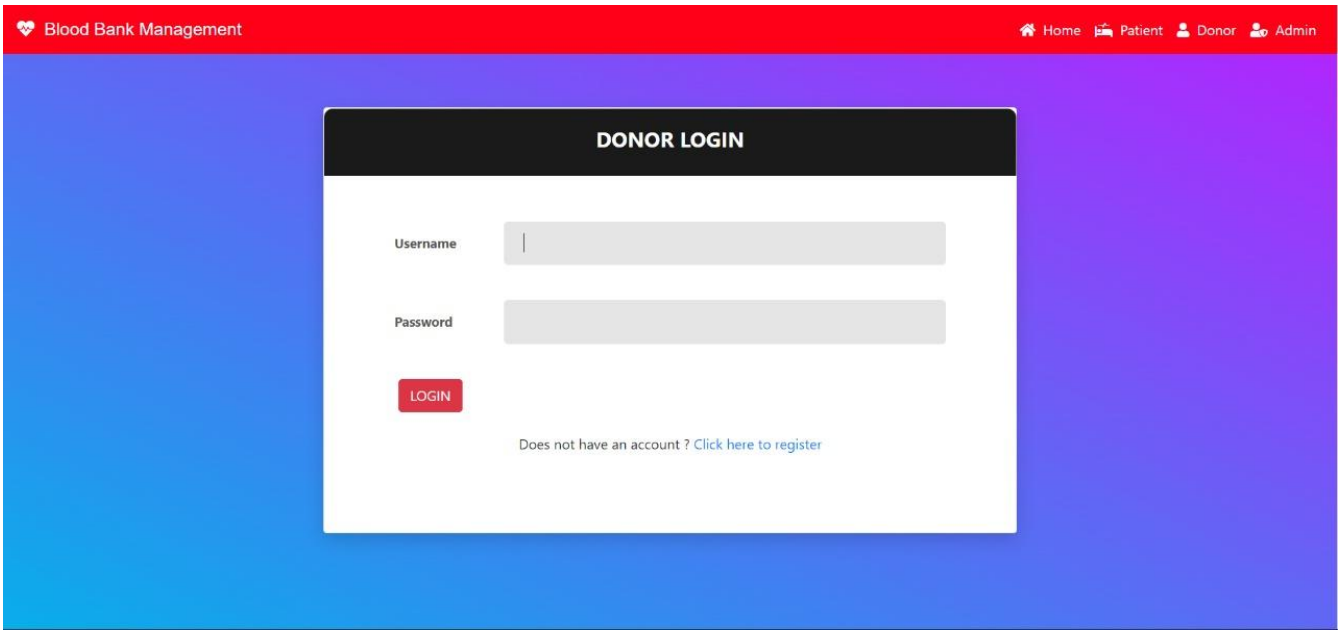

Fig 4.3.8: Donor login

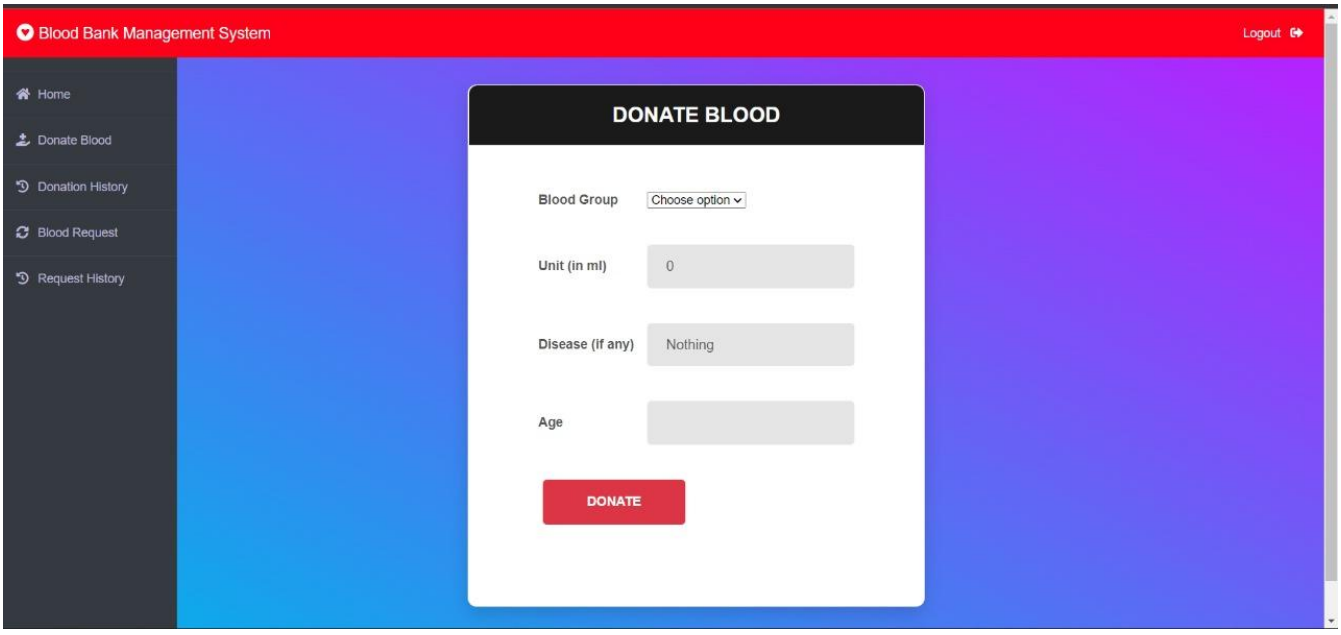

Fig 4.3.9: Donor donate blood

5.1 TEST PLAN

1. **Requesting blood:**
   Once logged in, the patient/donor can request for a blood of particular type.
2. **Donating blood:**
   Once logged in, the donor can make an entry for a donation they have made.
3. **Update inventory:**
   The admin can update the blood stock by entering the blood details.
4. **Accept/Reject Blood request/donation**
   The admin can accept/reject a particular request or a donation
5. **Update Patient/Donor information**
   The admin can edit a particular user’s details and also delete that user.

5.2 TEST CASES

Test Case #: 1.1
Test Case Name: Request Blood
System: Blood bank management system
Subsystem: Donor/Patient portal
Designed by: Nirbhay,Anjali,Sanskar,Naman
Design date: 16th may
Executed by:
Execution date: 17th may
Short description: Patient requests blood of a particular blood group

Pre Conditions The user should be logged in to the system to make a request

Step	Action	Expected system response	pass/fail	comment
1.	Click make request	Make blood request form is displayed		
2.	Enter the details	No response		
3.	Click Request button	Request history is shown showing your requests		

Post Conditions Request is shown in the history section

Test Case #: 1.2 **Test Case Name:** Donate Blood
System: Blood bank management system **Subsystem:** Donor portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: **Execution date:** 17th may
Short description: Donor puts its in the details of the blood he donated

Pre Conditions
The donor should be logged in to the system to make an addition in the system

Step	Action	Expected system response	pass/fail	comment
1.	Click donate blood	Donate blood form is displayed		
2.	Enter the details	No response		
3.	Click Donate button	Donation history is shown showing your donations		

Post Conditions
Donation is shown in the history section

Test Case #: 1.3
Test Case Name: Update inventory
System: Blood bank management system
Subsystem: Admin/Organization portal
Designed by: Nirbhay,Anjali,Sanskar,Naman
Design date: 16th may
Executed by:
Execution date: 17th may
Short description: The admin updates the blood inventory

Pre Conditions The admin must be logged in to the system

Step	Action	Expected system response	pass/fail	comment
1.	Click blood stock	Blood stock screen is displayed		
2.	Enter the details	No response		
3.	Click Update button	The particular blood type is updated		

Post Conditions No post conditions

Test Case #: 1.4 **Test Case Name:** Accept/Reject Blood Request/Donations

System: Blood bank management system **Subsystem:** Admin/organiztion portal

Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may

Executed by: **Execution date:** 17th may

Short description: Admin can accept/ receive blood requests and donations

Pre Conditions The admin should be logged in to the system to accept/reject blood

Step	Action	Expected system response	pass/fail	comment
1.	Click donations	A list of all donations is displayed		
2.	Click accept/reject	The particular donation is updated		
3.	Click blood requests	A list of all blood requests is displayed		
4.	Click accept/reject	The particular request is updated		

Post Conditions No post conditions

Test Case #: 1.5 **Test Case Name:** Update Donor/Patient info
System: Blood bank management system **Subsystem:** Admin/ organization portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: **Execution date:** 17th may
Short description: Admin can edit the information of a donor or a patient

Pre Conditions The admin should be logged in to the system to update a user's information

Step	**Action**	**Expected system response**	**pass/fail**	**comment**
1.	Click Donor/Patient	A list of all the users is displayed		
2.	Click on Edit	An edit page is displayed		
3.	Enter details	No response		
4.	Click Update button	Donor is updated and the screen is refreshed		

Post Conditions No post conditions

5.3 TEST REPORTS

Test Case #: 1.1 **Test Case Name:** Request Blood
System: Blood bank management system **Subsystem:** Donor/Patient portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: Shreyansh **Execution date:** 17th may
Short description: Patient requests blood of a particular blood group

Pre Conditions The user should be logged in to the system to make a request

Step	Action	Expected system response	pass/fail	comment
1.	Click make request	Make blood request form is displayed	**pass**	
2.	Enter the details	No response	**pass**	
3.	Click Request button	Request history is shown showing your requests	**pass**	

Post Conditions Request is shown in the history section

Test Case #: 1.2 **Test Case Name:** Donate Blood
System: Blood bank management system **Subsystem:** Donor portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: Shreyansh **Execution date:** 17th may
Short description: Donor puts its in the details of the blood he donated

Pre Conditions The donor should be logged in to the system to make an addition in the system

Step	Action	Expected system response	pass/fail	comment
1.	Click donate blood	Donate blood form is displayed	**pass**	
2.	Enter the details	No response	**pass**	
3.	Click Donate button	Donation history is shown showing your donations	**pass**	

Post Conditions Donation is shown in the history section

Test Case #: 1.3 **Test Case Name:** Update inventory
System: Blood bank management system **Subsystem:** Admin/Organization portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: Shreyansh **Execution date:** 17th may
Short description: The admin updates the blood inventory

Pre Conditions The admin must be logged in to the system

Step	**Action**	**Expected system response**	**pass/fail**	**comment**
1.	Click blood stock	Blood stock screen is displayed	**pass**	
2.	Enter the details	No response	**pass**	
3.	Click Update button	The particular blood type is updated	**pass**	

Post Conditions No post conditions

Test Case #: 1.4 **Test Case Name:** Accept/Reject Blood Request/Donations
System: Blood bank management system **Subsystem:** Admin/organiztion portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: Shreyansh **Execution date:** 17th may
Short description: Admin can accept/ receive blood requests and donations

Pre Conditions The admin should be logged in to the system to accept/reject blood

Step	**Action**	**Expected system response**	**pass/fail**	**comment**
1.	Click donations	A list of all donations is displayed	**pass**	
2.	Click accept/reject	The particular donation is updated	**pass**	
3.	Click blood requests	A list of all blood requests is displayed	**pass**	
4.	Click accept/reject	The particular request is updated	**pass**	

Post Conditions No post conditions

Test Case #: 1.5 **Test Case Name:** Update Donor/Patient info
System: Blood bank management system **Subsystem:** Admin/ organization portal
Designed by: Nirbhay,Anjali,Sanskar,Naman **Design date:** 16th may
Executed by: Shreyansh **Execution date:** 17th may
Short description: Admin can edit the information of a donor or a patient

Pre Conditions The admin should be logged in to the system to update a user's information

Step	**Action**	**Expected system response**	**pass/fail**	**comment**
1.	Click Donor/Patient	A list of all the users is displayed	**pass**	
2.	Click on Edit	An edit page is displayed	**pass**	
3.	Enter details	No response	**pass**	
4.	Click Update button	Donor is updated and the screen is refreshed	**pass**	

Post Conditions No post conditions