Detailed Design

Project Code:	**BBMS**
Project Name:	**Blood Bank Management System**

Prepared by/Date **Reviewed by/Date** **Approved by/Date**

1. Introduction:

The Blood Bank Management System is a comprehensive software solution designed to streamline the process of blood donation, request, and management. This system is aimed at connecting donors, patients, and blood bank administrators to ensure efficient blood supply management while saving lives.

1.1. Objective:

The primary objective of this project is to create a user-friendly platform that facilitates the donation of blood, allows users to request blood when needed, and enables blood bank administrators to efficiently manage donor appointments and patient requests.

1.2. Benefits:

- **Efficiency:** Streamlines the blood donation process, reducing manual paperwork.
- **Timeliness:** Enables quick responses to blood requests, potentially saving lives.
- **Transparency**: Provides transparency in blood availability.
- **Data Analysis:** Allows administrators to make data-driven decisions for resource allocation.

2. System Analysis:

2.1. Existing System:

The problem at hand is the inefficiency and lack of transparency in the blood donation and reservation process. Donors, patients, and administrators face challenges in managing the workflow, checking blood availability, and updating patient profiles.

- Mostly manual records are maintained.
- It takes a lot of time for getting exact address or information of donor/blood bank is a huge difficult and time seeking.
- The growth in number of recipients of blood now-a-days.
- The boom in number of accidents now-a-days is increased.

2.2. Proposed System:

The proposed solution is the development and implementation of a comprehensive Blood Donation and Reservation System, which aims to address these issues and streamline the entire process:

Implement an online platform where donors can register and schedule appointments. This reduces wait times and ensures a steady supply of blood.

Develop user-friendly patient profiles that store relevant information, including accepted requests and donor profiles. This simplifies tracking and management.

Enable patients with accepted requests to easily access the blood bank administration, ensuring a smooth and organized process for blood reservation and retrieval.

- The human beings in want of blood can look for the donors through giving their blood group and registering.

- The person’s time and difficulty is reduced a lot which resides inside the present system.

- Easy and Helpful.

- The humans aren't restricted to acquire or offer services in the website, is simplest; is serviced 24/7.

3. Key Features:

- **Donor Registration:** Donors can register to donate blood, providing essential information such as name, contact details, and blood type.

- **Blood Collection:** Trained administrators collect blood from registered donors, ensuring proper safety measures and documentation.

- **Blood Availability Check**: Administrators check the availability of the requested blood type when a patient submits a request.

- **Request Acceptance:** If the requested blood type is available, administrators accept the patient's request.

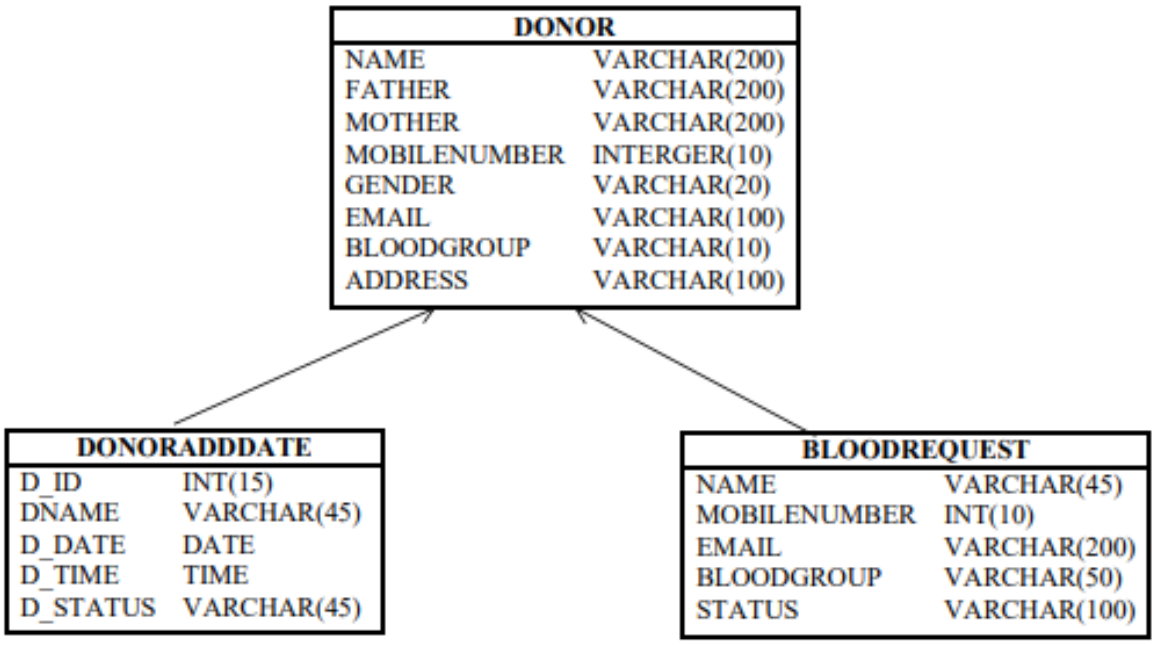
- **Patient Profile:** Patients can access a secure profile where they can view information about accepted requests.

- **Blood Bank Administration:** Patients with accepted requests can visit the blood bank administration to receive the reserved blood.

- **Reservation Update:** After the blood is taken, administrators update the patient's profile to indicate that blood has been reserved and the transaction is complete.

4. Global Data Structures and Shared Data Functions:

This section describes the structure of the tables to be used for the implementation of requirements as stated in the specifications.



5. Functional Requirements:

The contents of the "Blood Bank Management System" are:

1. **Administrator** – Add/View/Search information related to blood, donor and patient.
2. **Donor**–View and update details.
3. **Patient** – View details.

5.1. Interface Requirements

The Blood Bank Management System is been developed as a Web Application using HTML/JSP screens.

5.2. Non-functional Requirements

- The application should be intuitive to use and easy-to-maintain
- Designer/Programmer need to consider suitable conditions/constraints and assumptions.
- The application should be able to handle simultaneous requests from multiple clients
- Upon installation, the administrator, donor and patient are expected to have User IDs and Passwords
- The system may force the administrator/donor / patient to change their passwords to a non-default one, upon initial login.

Software Environment:

Windows Operating System
Server – Apache Tomcat 9 or higher
Database – MYSQL
JRE 1.9 or higher
Eclipse IDE

6. Functionalities:

- User Registration and Profiles:
    - Donors can register and create profiles with their personal and medical information.
    - Users (patients or their representatives) can also register.
- Donor Management:
    - Donors can view and update their profiles.
    - Schedule blood donation appointments and view their donation history.
- Blood Bank Administration Management:
    - Users can request blood by providing blood type and personal details like gender, mobile number.
    - Blood bank administrators can review and approve/reject requests.
    - Access the system with authentication.
    - View Donors and Patient Lists.
    - Display a list of all donors.
    - Display a list of blood requests from patients.
    - Accept or reject blood requests from patients.
    - If accepted, update the patient's profile to show the request as "accepted."
    - Patients can collect blood from the administrative after approval.
    - Update the patient's profile to indicate that blood has been taken by the patient.
    - Administrative can view and manage donor appointments.
    - When donors donate blood, update their status to "donated blood."

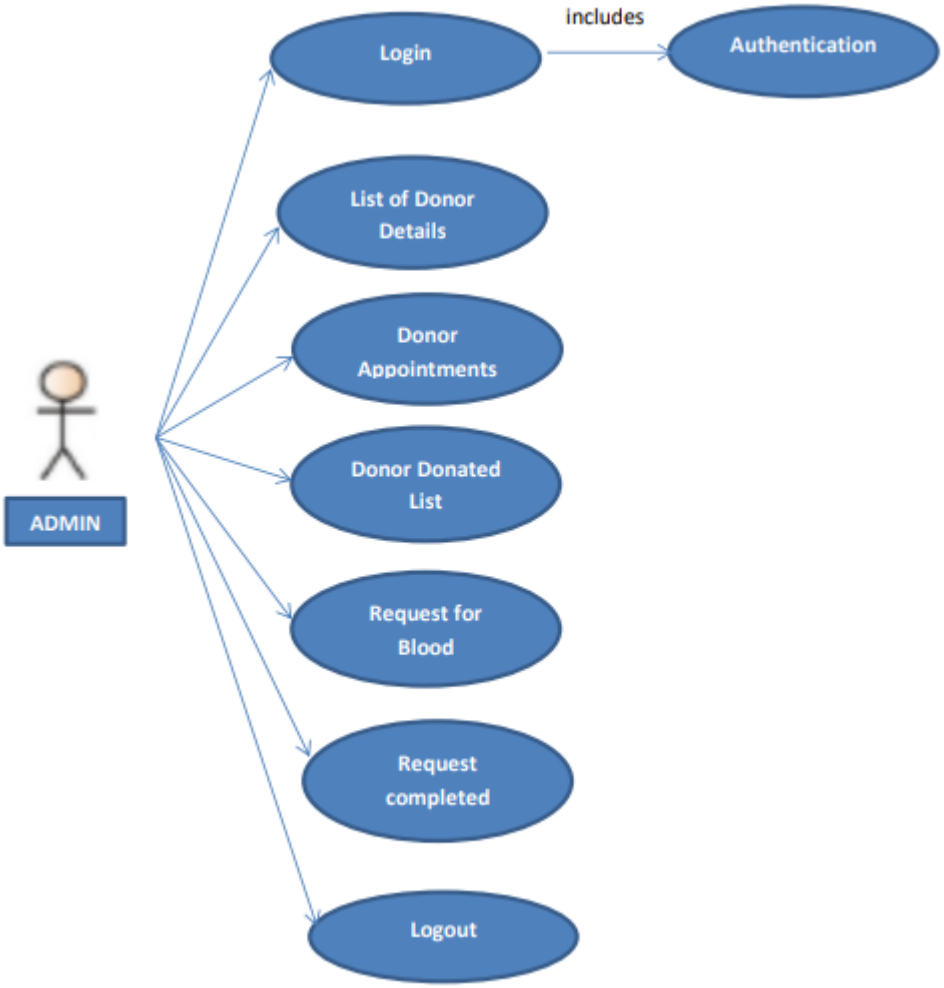
7. High Level Design:

This section describes the high level design diagrams. Use case diagram with Use Case definition, Sequence Diagram and Class Diagram which provides a visual representation of the requirements, logical flow and their class representations.

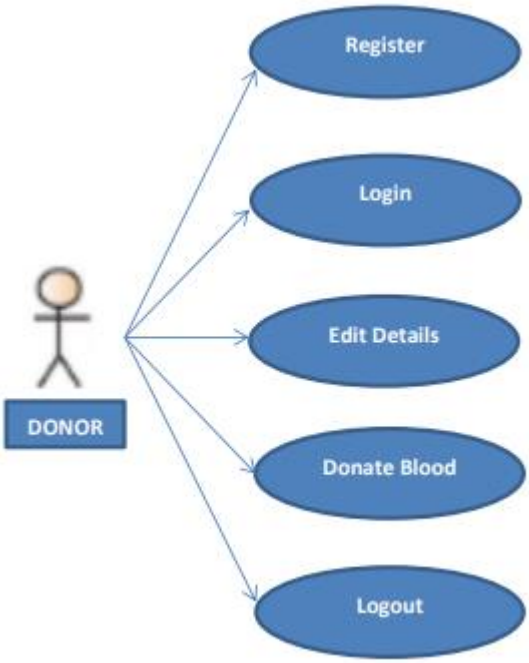
7.1. Use Case Diagrams:

The requirements of a system can be represented using a use case model in the Use Case Diagram. The use case diagram for the actors of this case study is given as below.

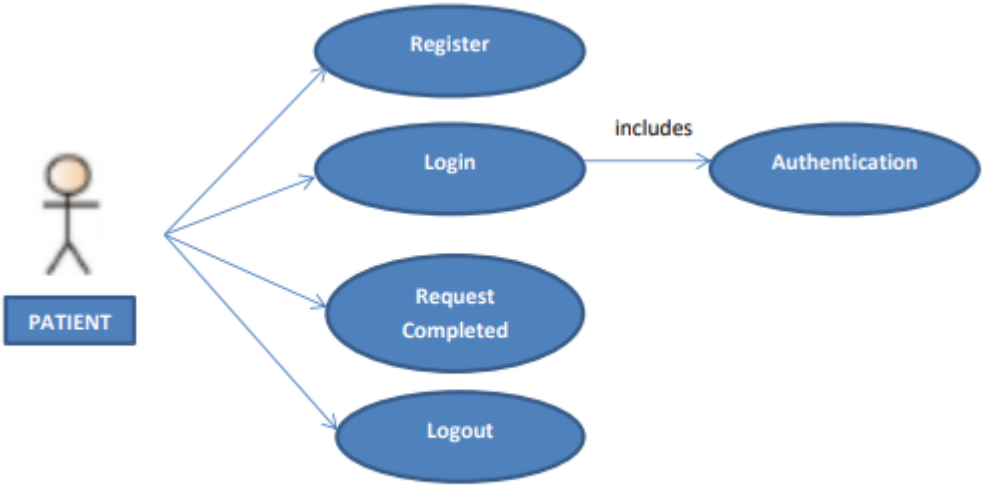
7.1.1. Use Case Diagram for Admin:



7.1.2. Use Case Diagram for Donor:



7.1.3. Use Case Diagram for Patient:



7.2. Use case Definition:

Generally, in a design document, Use case definitions should be written for all the Requirements of the system.

Note: Participants are expected to document use case definitions for all requirements. However, for few requirements documented below for reference.

Below table explains ‘Use Case’ definition for requirement - Login operation for all users.

7.2.1. Login

USE CASE	Login
Goal	All users logging into the system should be authenticated using a unique login-id and password (operations to be supported based on type of user)
Preconditions	If the user type is ‘Admin’, credential details should exist.
Success End Condition	If the user type is ‘Admin’, then redirect to the Admin page. If the user type is ‘Donor’, then redirect to the Donor login page. If the user type is ‘Patient’, then redirect to the Patient login page.
Failed End Condition	The end user is redirected to a Register Page, and/or is asked to in-valid login credentials.
Actors	Admin, Donor, Patient.

Trigger	Login button	
DESCRIPTION	**Step**	**Action**
	1	Enter Login credentials (id & password)
	2	Click on Login button
	3	If id & password is Success, then identify user type Display appropriate(Admin/Donor/Patient) home page
	Step	**Branching Action**
	1	If 'id' is not existing then return with requesting for
	2	If password is not matching return with suitable error message say 'Re-enter id & password'
Assumptions	Admin/Donor/Patient login credentials are available in the Database and others are already registered with their credentials	

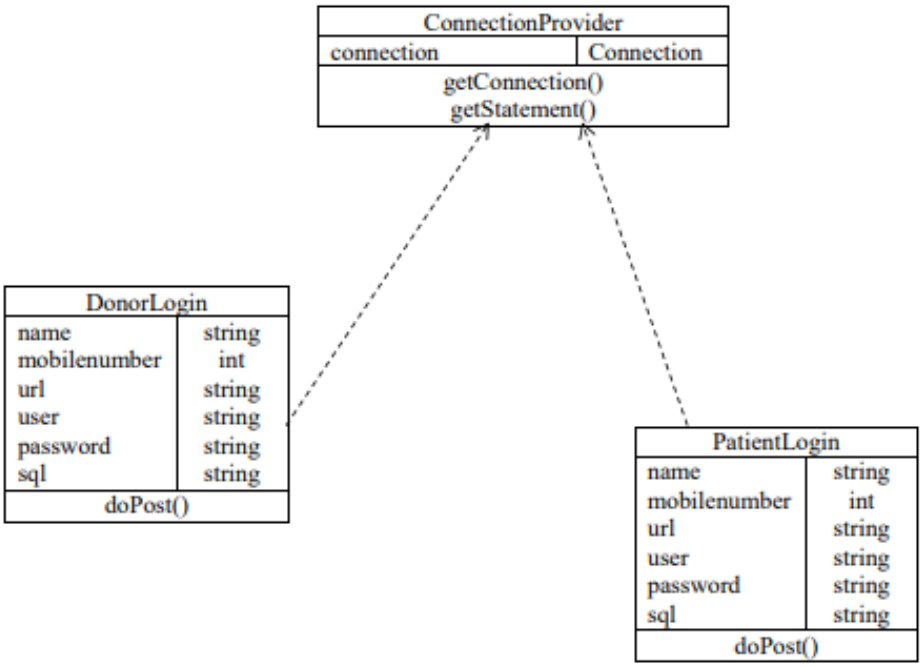
Below table explains ‘Use Case’ definition for requirement – view donor Details operation for Admin user only.

7.2.2. View donor Details:

USE CASE	View donor details
Goal	To enable Administrator to view donor
Preconditions	Administrator must be logged in to be able to view Candidate.
Success End Condition	“Redirect to Admin home page”
Failed End Condition	"Error page"
Actors	Admin
Trigger	Logout
DESCRIPTION	Provide appropriate Donor details

7.3. Class Diagram:

The class diagram is a very basic concept in object-oriented world. Class diagrams demonstrate a model, describing what attributes and behaviour it has rather than describing the methods for accomplishing operations. Class diagrams are very useful in representing relationships between classes and interfaces.

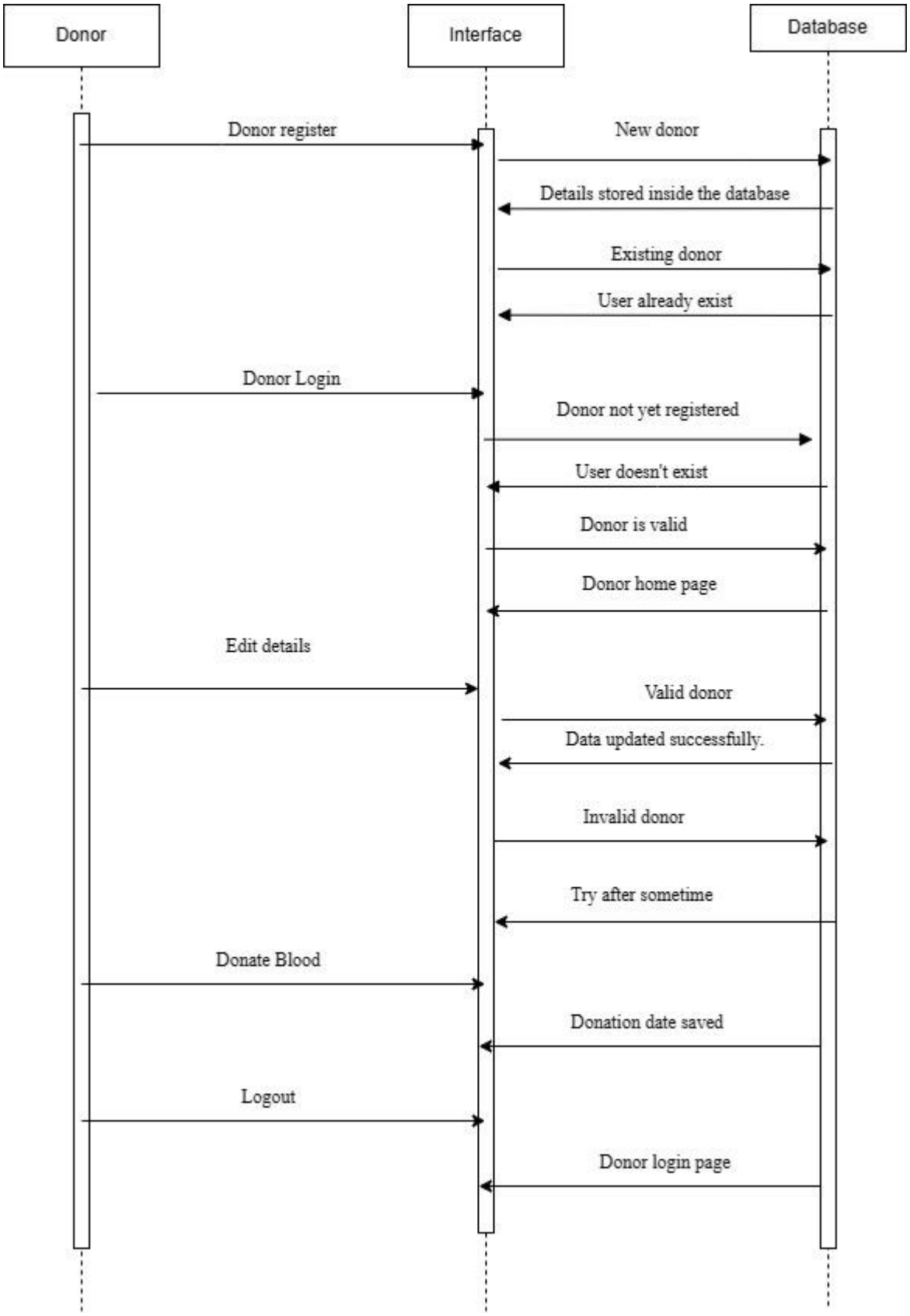


7.4. Sequence Diagram:

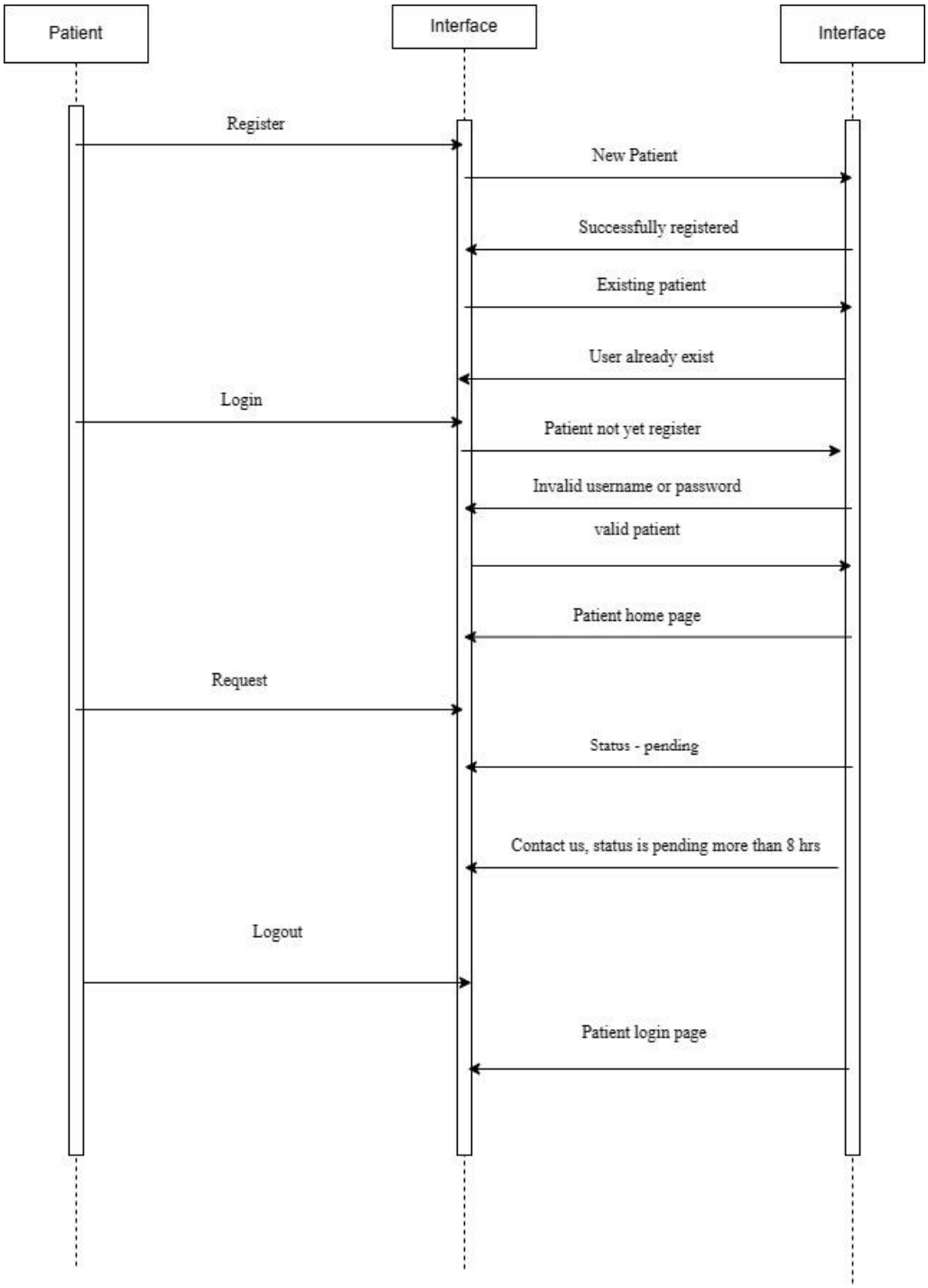
A graphical representation of a module’s function invoking functions of other modules in order to achieve a task (specific user requirement) is called a sequence diagram.

A sequence diagram for the authentication process is given below for reference. The below example is for a Web Application using jsp.

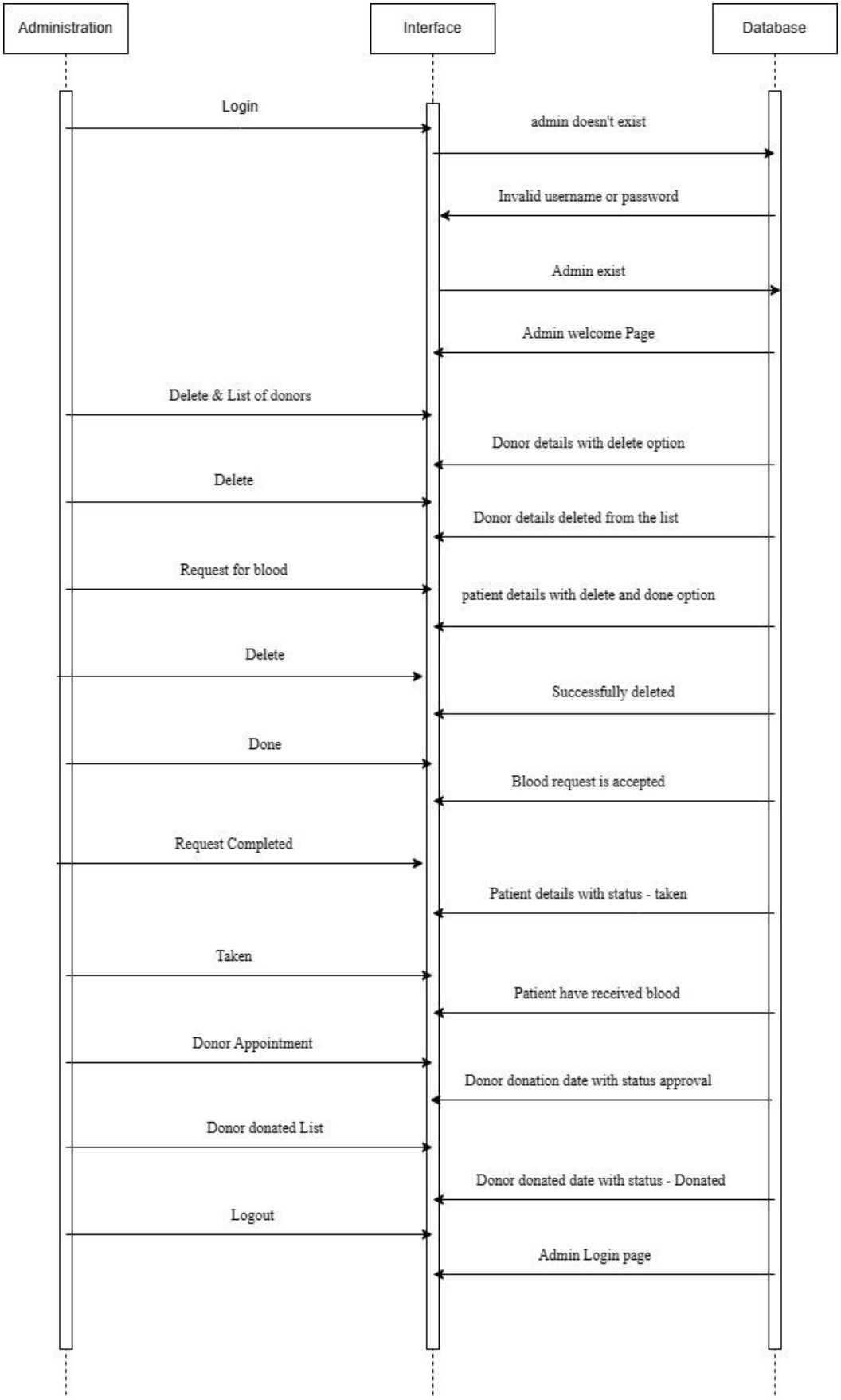
Donor:



Patient:


Patient
Interface
Interface
Register
New Patient
Successfully registered
Existing patient
User already exist
Login
Patient not yet register
Invalid username or password
valid patient
Patient home page
Request
Status - pending
Contact us, status is pending more than 8 hrs
Logout
Patient login page

Admin:


Administration
Interface
Database
Login
admin doesn't exist
Invalid username or password
Admin exist
Admin welcome Page
Delete & List of donors
Donor details with delete option
Delete
Donor details deleted from the list
Request for blood
patient details with delete and done option
Delete
Successfully deleted
Done
Blood request is accepted
Request Completed
Patient details with status - taken
Taken
Patient have received blood
Donor Appointment
Donor donation date with status approval
Donor donated List
Donor donated date with status - Donated
Logout
Admin Login page

8. Tables:

8.1. Table of Donor:

DONOR		
NAME	VARCHAR(200)	Either(A-Z /a-z/)
FATHER	VARCHAR(200)	Either(A-Z /a-z/)
MOTHER	VARCHAR(200)	Either(A-Z /a-z/)
MOBILENUMBER	INTERGER(10)	Only(0-9)
GENDER	VARCHAR(20)	Male / female
EMAIL	VARCHAR(100)	Primary key
BLOODGROUP	VARCHAR(10)	Only(A+/A-/B+/B-/O+/O-/AB+/AB-)
ADDRESS	VARCHAR(100)	Not null

8.2. Table of BloodRequest:

BLOODREQUEST		
NAME	VARCHAR(45)	Either(A-Z /a-z/)
MOBILENUMBER	INT(10)	Only(0-9)
EMAIL	VARCHAR(200)	Primary key
BLOODGROUP	VARCHAR(50)	Only(A+/A-/B+/B-/O+/O-/AB+/AB-)
STATUS	VARCHAR(100)	Either [1 ,0]

8.3. Table of DonorAdddate:

DONORADDDATE		
D_ID	INT(15)	Primary key
DNAME	VARCHAR(45)	Either(A-Z /a-z/)
D_DATE	DATE	Not null
D_TIME	TIME	time
D_STATUS	VARCHAR(45)	Either [1 ,0]

9. Packages / Classes / Interface:

This section provides a brief outlook on the packaging hierarchy along with the respective classes to be used for the implementation.
The 3 packages mentioned below are for both GUI and Web Application.

com.java.project

ConnectionProvider	
connection	Connection
getConnection() getStatement()	

com.java.donor

DonorLogin	
name	string
mobilenumber	int
url	string
user	string
password	string
sql	string
doPost()	

com.java.patient

PatientLogin	
name	string
mobilenumber	int
url	string
user	string
password	string
sql	string
doPost()	

10 .UI Templates:

10.1. UI Principle:

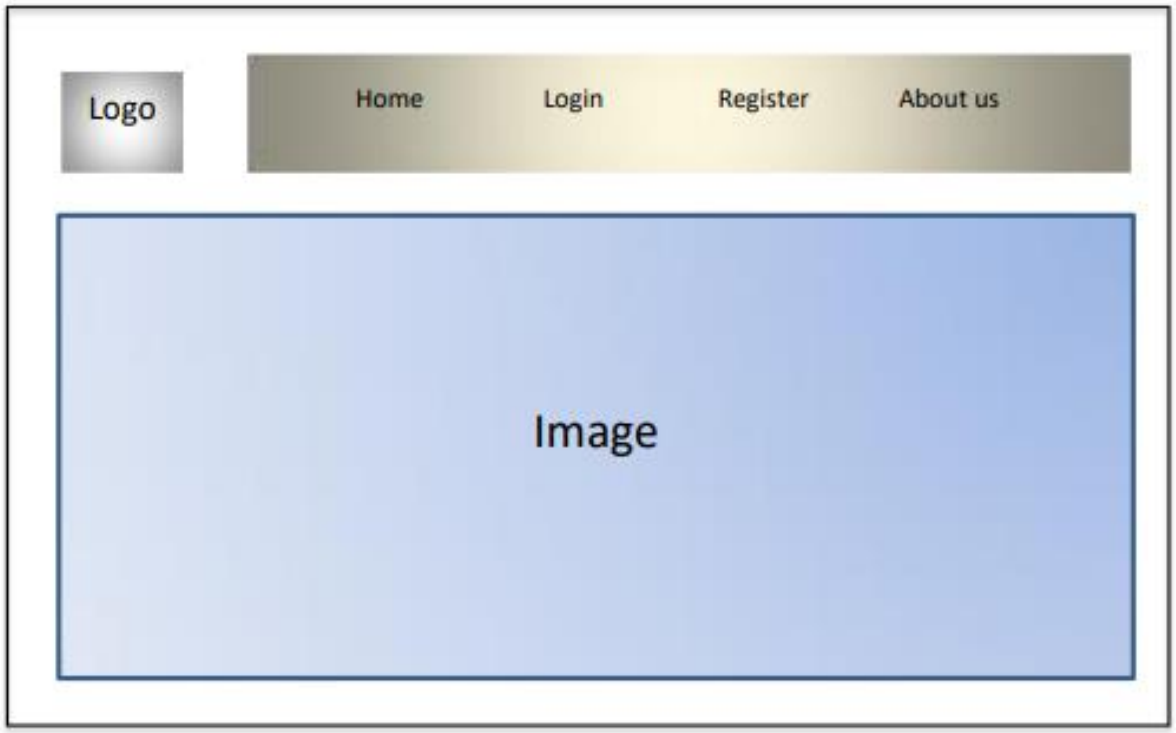
The UI [Presentation Layer] should be designed with the below mentioned principles which helps easy interaction by the user to the application.

10.2. UI controls and Usage Principle:

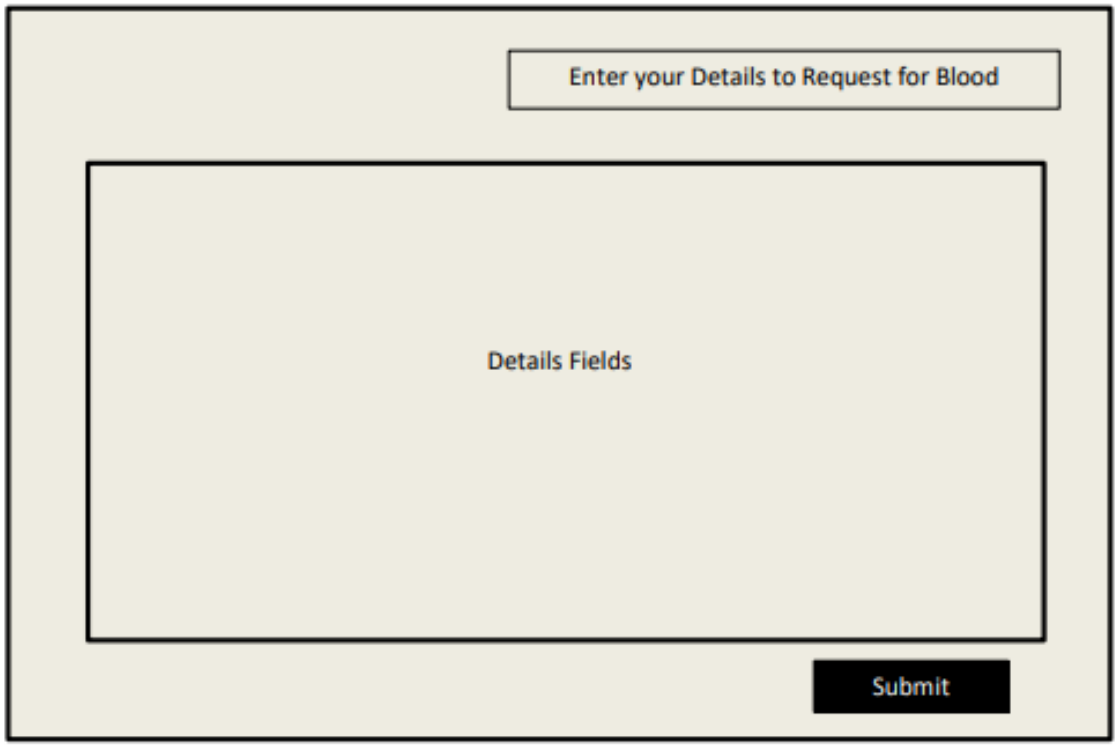
UI Type	Controls	Description
Direct Entry	Text Box, Text Area	Any input that cannot be predicted and needs the user to key in. e.g. Name, Address, contact no etc.
Static Selection	Option Button, Check Box, Drop Down	Should be used where the input can be predefined. E.g. gender, month [Jan – Dec] etc. If number of items is more, drop down is preferred.
Dynamic Selection	Drop Down	The items for the drop down should be retrieved from a stored Data. E.g. Displaying Districts in a drop down from places table.
Automation	Label, Text Field [Read Only]	Data's that are calculative or an output of a function. e.g. : Displaying system date, showing total amount etc.
Decision Control	Button	Operations like submit, save, clear should be executed only upon clicking respective buttons.

10.3. UI Template:

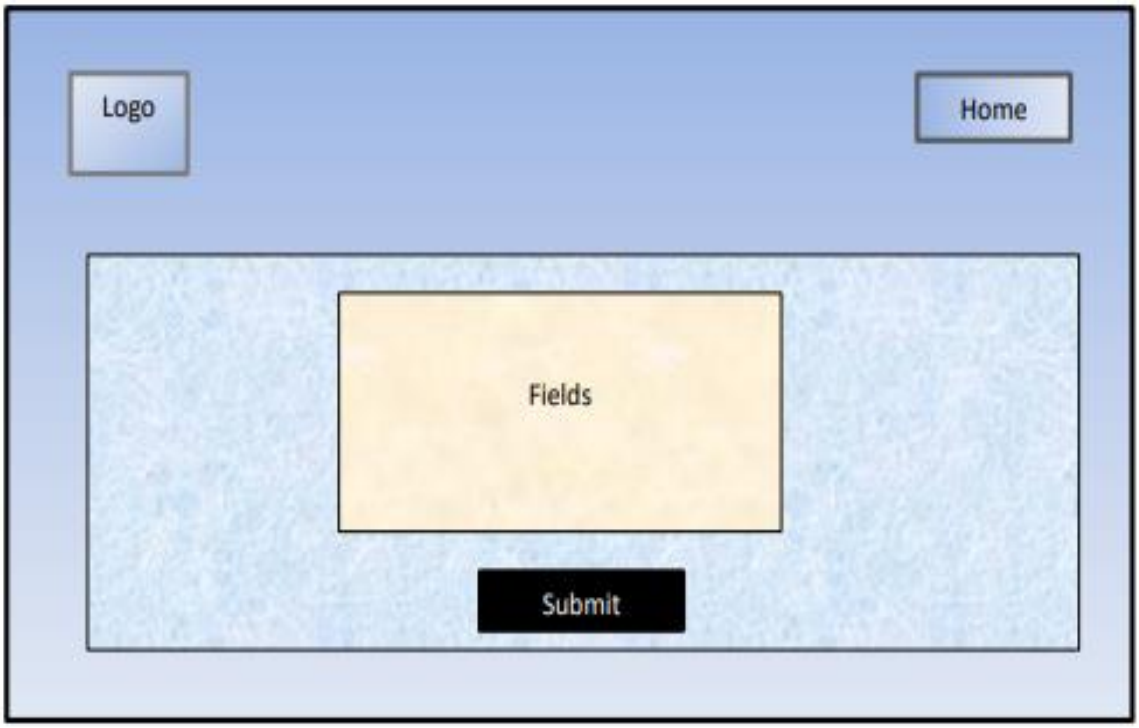
Main Page Template:



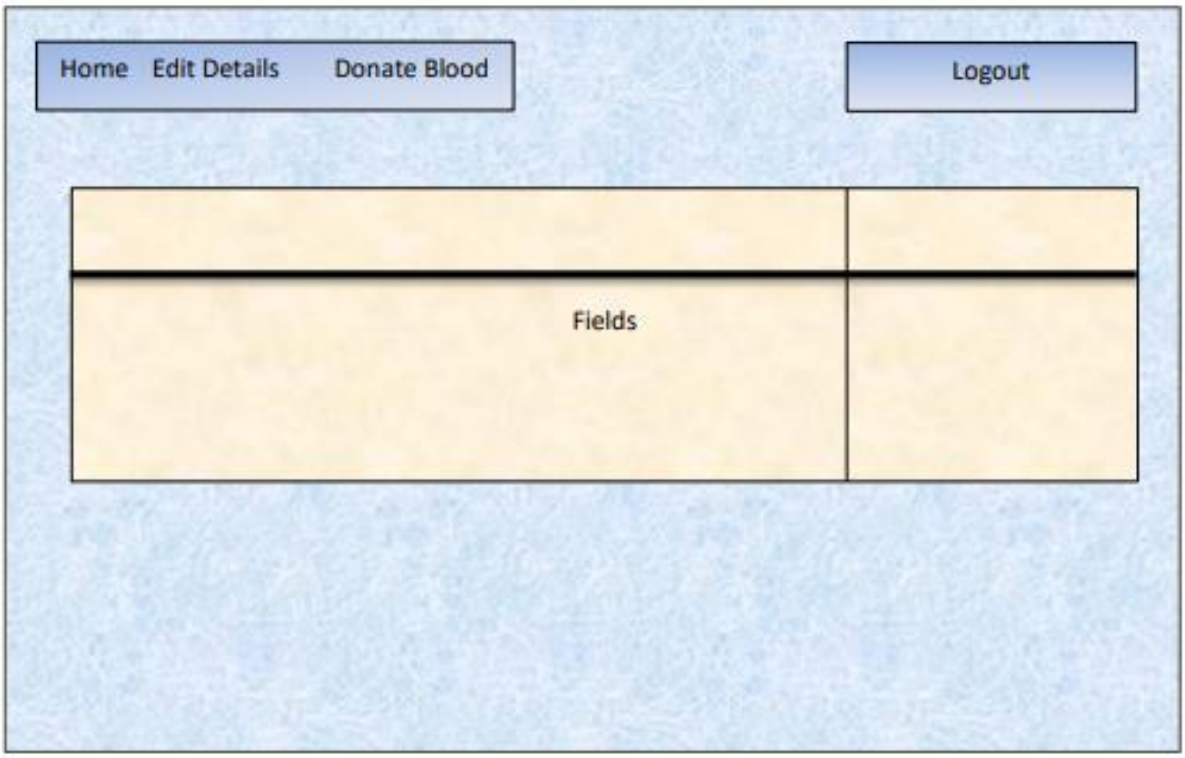
Register Page Template:



Login Page Template:



Edit Details Page Template:



11. Critical Functions and Focus for Testing:

Login ()
viewDonor ()
request ()
acceptRequest ()

12. Limitations:

- Less Security
- Internet connectivity is needed for the online blood management system. Internet speed may affect the perception of the systems users with regards to the system effectiveness and efficiency.

13. Conclusion:

The Blood Bank Management System serves as a bridge between donors and patients, ensuring that the critical need for blood is met effectively and efficiently. By connecting these stakeholders through a user-friendly digital platform, the project aims to make a significant impact on the healthcare sector, ultimately contributing to the well-being of communities.

Overall, The Blood Bank Management System is an essential tool for managing the blood donation process and improving the efficiency and effectiveness of blood banks and hospitals.

14. Future Scope:

Support for various regional languages for better reach. The size of the database may increase exponentially, so our BBMS is made such that it is scalable and can be deployed on cloud storage systems like Amazon Elastic Compute Cloud (EC2) or Google's Kubernetes Engine (GKE) after containerizing the application.

The system can be improved by adding new features like integration with other systems, mobile application, analytics and reporting, online blood bank, and integration with IoT devices. These improvements can help in the efficient utilization of blood samples, reduce wastage, and ultimately save more lives.