### Blood Bridge

Final Year Project Report

by

|  |  |
| --- | --- |
| **Sehar Akbar** | **2145** |
| **Iqra Khokhar** | **2124** |
| **Komal Praveen** | **2102** |

Project Advisor:

**Mr. Abdul Islam Bajwa**

**Department of Computer Science and Information Technology,**

**Jinnah Islamiya Graduate College, Sialkot, Pakistan.**

**(2021-2025)**

Executive Summary

Blood Bridge is a web-based blood donation and management system designed to add donors, recipients and blood banks. This simplifies the process of finding and donating blood, especially during the emergency. The platform automatically automatches the donor registration, recipient request and inventory updates, ensuring safe communication and real time notifications. Using a tool such as Google Maps API, blood bridge helps users to find nearby donors quickly. The purpose of the system is to improve blood donation efficiency, reduce delays and save life by making the process fast, more accessible and reliable.

.

## FYDP Overview

FYDP Title

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Roll Numbers** | **Name** | **Signatures** |
| 1. | 069582 / 2145 | Sehar Akbar |  |
| 2. | 069593 / 2124 | Iqra Khokhar |  |
| 3. | 069639 / 2102 | Komal Parveen |  |

***Table 1 Proposal Project summary***

|  |
| --- |
| **FYDP Goals** |
| * To develop an efficient online platform that connects blood donors with recipients in realtime. * To simplify the process of requesting and donating blood through a user-friendly web application**.** |
| **FYDP Objectives** |
| * To design and implement an online system for efficient blood donation management. * To enable users to easily find and connect with nearby blood donors or recipients. * To ensure data security and privacy through secure user authentication. * To provide real-time notifications and updates to users and administrators. * To maintain a centralized database for managing users, donations, and requests. |
| **FYDP Success Criteria** |
| * Users can successfully register, login, and manage their profiles. * Donors and recipients can easily connect through accurate donor search and request features. * The system ensures secure and reliable data management. * Real-time notifications keep users informed about requests and approvals.. |
| **Assumptions:** |
| * Users have internet access. * Information provided by users is accurate. * Users know how to use web apps. * Location services (like Google Maps API) work reliably. * Admins actively manage user activity. |
| **Risks & Obstacles** |
| * Data privacy breaches. * Server downtime or failures. Fake or inaccurate user data.  * Low donor participation. * Dependence on third-party APIs. |
| **Organization Address: Department of Computer Science, Govt. Jinnah Islamia Graduate College, Sialkot, Pakistan.** |
| **Target End Users** |
| **Suggested Project Supervisor: Sir Abdul Islam Bajwa** |
| **Approved By:** |

|  |
| --- |
| **Date: October 07, 2024** |

**Table of Contents**

[Executive Summary 2](#_TOC_250063)

[*FYDP Overview* *3*](#_TOC_250062)

[*Chapter No 1* *8*](#_TOC_250061)

[*Project Proposal* *8*](#_TOC_250060)

* 1. [*Introduction* *9*](#_TOC_250059)
  2. [*Background* *9*](#_TOC_250058)
  3. [*Problem Statement* *10*](#_TOC_250057)
  4. [*Stakeholders & Interests* *10*](#_TOC_250056)
  5. [*Objectives* *11*](#_TOC_250055)
  6. [*Scope* *11*](#_TOC_250054)
  7. [*Assumptions* *12*](#_TOC_250053)
  8. [*Risks* *12*](#_TOC_250052)
  9. [*Success Criteria* *12*](#_TOC_250051)
  10. [*Tools, Libraries & Technologies* *13*](#_TOC_250050)
  11. [*Work Division* *14*](#_TOC_250049)
  12. [*Summary* *14*](#_TOC_250048)

[*Chapter No 2* *15*](#_TOC_250047)

[*Literature Review* *15*](#_TOC_250046)

* 1. [*Literature Survey* *16*](#_TOC_250045)
  2. [*Related Work:* *16*](#_TOC_250044)
  3. [*Gap Analysis* *16*](#_TOC_250043)
  4. [*Summary* *17*](#_TOC_250042)

[*Chapter No 3* *18*](#_TOC_250041)

[*Software Requirements Specification* *18*](#_TOC_250040)

* 1. [*Requirements Analysis* *19*](#_TOC_250039)
  2. [*User classes and characteristics* *19*](#_TOC_250038)
  3. [*Requirement Identifying Technique* *19*](#_TOC_250037)
  4. [*Functional Requirements* *20*](#_TOC_250036)
  5. [*Non-Functional Requirements* *24*](#_TOC_250035)

1. [Reliability 24](#_TOC_250034)
2. [*Usability* *24*](#_TOC_250033)
3. [*Performance* *24*](#_TOC_250032)
4. [*Security* *25*](#_TOC_250031)
   1. [*External Interface Requirements* *25*](#_TOC_250030)
5. [*User Interfaces Requirements* *25*](#_TOC_250029)
6. [*Software Interfaces* *25*](#_TOC_250028)
7. [*Hardware Interfaces* *26*](#_TOC_250027)
8. [*Communications Interfaces* *26*](#_TOC_250026)
   1. [*Use case Analysis* *26*](#_TOC_250025)
9. [*Use Case #1 (User Registration)* *26*](#_TOC_250024)
10. [*Use Case #2(User Login)* *27*](#_TOC_250023)

None 27

1. [*Use Case #3(Find a donor)* *27*](#_TOC_250022)
2. [*Use Case #4(Request Blood)* *28*](#_TOC_250021)
3. [*Use Case #5(Approve/Reject Request)* *29*](#_TOC_250020)
4. [*Use Case #6(Notifications for Donors)* *30*](#_TOC_250019)
5. [*Use Case #7(Contact a Donor)* *30*](#_TOC_250018)
6. [*Use Case Diagram* *31*](#_TOC_250017)
   1. [*Storyboarding:* *34*](#_TOC_250016)
   2. [*Summary* *34*](#_TOC_250015)

[*Chapter No 4* *35*](#_TOC_250014)

[*Software Design Specification* *35*](#_TOC_250013)

* 1. [*System Design*](#_TOC_250012)

*.* *36*

* 1. [*Design Considerations*](#_TOC_250011)

*.* *36*

* 1. [*Requirements Traceability Matrix*](#_TOC_250010)

*.* *36*

* 1. [*Design Models*](#_TOC_250009)

*.* *38*

1. [*Design Class Diagram (DCD)* *38*](#_TOC_250008)
   1. [*Architectural Design*](#_TOC_250007)

*.* *39*

1. [*UML Component diagram* *40*](#_TOC_250006)
   1. [*Data Design* *41*](#_TOC_250005)
2. [*Data Dictionary* *44*](#_TOC_250004)
   1. *User Interface Design* *46*
      1. *Screen Images* *46*

*50*

* 1. [*Design Decisions*](#_TOC_250003)

*.* *52*

* 1. [*Behavior Model*](#_TOC_250002)

*.* *52*

* 1. [*Summary*](#_TOC_250001)

*.* *53*

[*Reference* *54*](#_TOC_250000)

### List of Tables

**Table 1 Proposal Project summary** 3

**Table 2 Tools , Libraries and Technologies** 13

**Table 3 Work Devision** 14

**Table 4 Gap Analysis** 17

Table 5 User classes and characteristics 19

Table 6 Use Case and their Description 19

Table 7 FR-1 20

Table 8 FR-2 21

Table 9 FR-3 21

Table 10 FR-4 22

Table 11 FR-5 22

Table 12 FR-6 22

Table 13 FR-7 23

Table 14 FR-8 23

Table 15 FR-9 24

Table 16 Use Case 1(User Registration) 26

Table 17 Table 17 Use Case 2 (User Login) 27

Table 18 Use Case 3 (Find a Donor) 28

Table 19 Use Case 4 ( Request Blood) 28

Table 20 Use Case 5(Approve/Reject Request) 29

Table 21 Use Case 6 ( Notification for Donors) 30

Table 22 Use Case 7 ( Contact a Donor) 30

Table 23 Requirements Traceability Matrix 36

Table 24 (MVC + Multi-Tier) 41

Table 25 User Table 41

Table 26 Donor Table 42

Table 27 Blood Request Table 42

Table 28 Notification Table 43

Table 29 Admin Table 44

Table 30 Terminology and Description 44

Table 31 Function Name and Function Parameter 45

Table 32 Screen Object and Action 51

###### Tables of Figures

**Figure 1 Use Case Diagram for Admin** 31

**Figure 2 Use Case Diagram for Donor** 32

**Figure 3 Use Case Diagram for Recipient** 33

**Figure 4 Story boarding** 34

Figure 5 Class Diagram 38

Figure 6 Architecture Design 39

Figure 7 UML Component diagram 40

Figure 8 Sequence Diagram 52

Figure 9 State Diagram 53

Chapter No 1

Project Proposal

#### Introduction

In today's world, every person life can be saved with the click of a mouse. Blood Bridge is an internet-based website that mechanizes blood donation, management, and distribution processes. It serves as an intermediary between the donors, recipients, and blood banks to enable communication and blood resource allocation. The system provides donor registration, recipient requests, updates on blood inventory, blood availability and even location. This answer addresses major problems such as the ease of access, accurate tracking of blood allocation, and most critically, the availability of the blood. Technology is applied to address major problems that in the process ease the limit of long processes. It ensures that life-saving units are providing to those in need without any interruptions. It is, plain and simple, life saving. This project helps to foster a culture of voluntary blood donation and help to enhance the efficiency of operation of blood banks and health care certifies providers. In times of emergencies, this system is a lifesaver as it ensures the availability of the right blood type at all times.

A blood bank is an institution where blood is collected and stored. Most of the blood banks' inventory derives from voluntary donors giving blood through blood drives, hospitals, or national blood centers. Blood banks have the tasks of blood collection, management, and testing before storage, updating the inventory of blood available, and authorizing all requests for blood. In the process of blood donation, blood is passed on to blood banks, and even in the process of blood transfusion, some issues arise. Due to the lack of sufficient staff, outdated machinery, poor storage of blood, partial automation of the blood management process, and low apparatus supply, the process of blood donation and transfusion has become sluggish. The shortage and wastage of blood units may be due to the absence of a quality management system for blood banks. The blood bank should prioritize any emergency hospitals or patients who require it most before issuing the blood unit to anyone on the waiting list**.**

#### Background

The demand for a well-organized blood donation management system has grown because of the escalating healthcare demands and the inefficiencies of associated with conventional blood donation practices. Manual record keeping is commonly practiced in the conventional approach, leading to errors, delays, and mismanagement of crucial donor and recipient information.

A BDMS offers an online solution that computerizes blood donation drives management, donor registrations, blood bank inventories, and recipient requests. Through the application of current technologies such as databases, cloud computing, and mobile applications, the system increases accessibility, accuracy, and security in the blood donation.  **Current Challenges in Blood Donation Management:**

* + - **Lack of Awareness and Donor Engagement** - Numerous potential donors are not informed of donation requirements and opportunities.
    - **Inefficient Record-Keeping** - Manual system cause inaccurate or outdated data.
    - **Difficulty in Locating Donors** – In case of emergencies, finding compatible donors can be time-consuming. o **Blood Wastage and Shortages** – Ineffective inventory control leads to excess supply wastage or critical shortages.
    - **Security and Privacy Issues** - Protecting donor and recipient information must be secured unauthorized person.

#### Problem Statement

When there is a medical emergency, it is really hard for a patient to search for donors. What if a person needs blood but cannot have any? There are many gaps in data between patients (who receive the blood), donors (who give the blood), blood banks, and clinics in our areas. Not parts of the country have the same accessibility to the medical faculty. If someone needs blood, they should first check within their family members, then the nearest hospitals, clinics, and blood banks. If they cannot get the necessary information they need, it is very difficult for them to reach other people in an effort to get blood quickly.

The main objective of the project is to eradicate or overcome the problems in finding blood, especially during emergency cases. By using this software, people can easily find blood contributors in close proximity to their current location. A combination of a unique username and password will provide access to security features, ensuring that only authorized personnel can utilize the program. The program will motivate healthy people to donate blood, while management of the blood stock will be undertaken by blood bank personnel. The application will maintain records in relation to donors, recipients, and blood group, as well as donor records. In general, Google Maps will prove beneficial for locating nearby and distant areas (in meters/kilometers).

#### Stakeholders & Interests

|  |  |
| --- | --- |
| **Stakeholder** | **Roles** |
| Donor | Register his/her self and update his/her availability Approve/reject blood requests  Manage their profiles and charity history. |
| Recipient | Search for donors based on blood type and location.  Send blood requests to donor and communicate with donors. |
| Admin | Manage platform operations and manage user accounts. Monitor all the system activities.  Handling system issues, complaints and technical assistance. |
| System Developer | Design, develop and maintenance to the website. Apply safety measures to protect user data.  Make sure the system moves efficiently with minimal downtime. |
| Healthcare  Organization (Future Scope) | Healthcare Organization (Future Scope) \*\* Can cooperate to verify donors and recipients.  Help promote the platform for broader access. |

* 1. **Objectives:**
     + To design and implement an online system for efficient blood donation management.
     + To enable users to easily find and connect with nearby blood donors or recipients.
     + To ensure data security and privacy through secure user authentication.
     + To provide real-time notifications and updates to users and administrators.
     + To maintain a centralized database for managing users, donations, and requests.

#### Scope:

The Blood Bridge is designed to simplify the process of blood donation, donor management and request. The system will allow donors and recipients to communicate with each other while enabling efficient system.

Inclusions:

* + - **User Authentication:**

Safe login and registration for donor, recipients, and administrator. Donor and

Recipient Profiles, management of user data, donation history, and blood request report.

* + - Blood Donation Requests:

Handling and Monitoring recipient and blood donation requests.

* + - Find a Donor:

Donor search and filtering for compatible blood types and locations.

* + - Blood Bank Management:

Admin features to manage blood stock, availability and donations.

* + - Notification System:

Notifications for recipient, donor, and admin on blood requests and approval.

* + - Google API integration:

Location based completely services to enable donor looking and blood financial institution route.

* + - Dark Themed Consumer Interface:

A beautiful UI for improved consumer revel in.

* + - Social Media Integration:

The device will come with immediate sharing of donation activities on social media websites.

Exclusions:

* + - **Cell application:**

The initial scope may can be a web-based platform, without having a particular cellular map.

* + - Multilanguage assistance:

The machine can be developed in English mainly with no multilingual assistance in the initial model.

* + - Third-Party payment Processing:

There won’t be any payment gateway for service fees or donations.

#### Assumptions

* + - Users have internet access.
    - Information provided by users is accurate.
    - Users know how to use web apps.
    - Location services (like Google Maps API) work reliably.
    - Admins actively manage user activity.

#### Risks

* + - . Data Privacy and Security Risks:
      * Unauthorized access to donor and recipient information. o Risk of data breaches exposing sensitive personal details.
      * Possibility of cyberattacks like hacking or phishing.
    - Fake or Misleading User Information:
      * Donors or recipients may provide false medical or contact details.
      * Others will pretend to be potential donors and never intend giving blood at all.
    - Health and Eligibility Issues:
      * Donors are not within the good standard of health and attempt donation anyway.
      * Insufficient medical check-up before donation.
    - Health and Eligibility Issues:
      * Donors are not within the good standard of health and attempt donation anyway.
        + Insufficient medical check-up before donation.

#### Success Criteria

* + - Users can successfully register, login, and manage their profiles.
    - Donors and recipients can easily connect through accurate donor search and request features.
    - The system ensures secure and reliable data management.
    - Real-time notifications keep users informed about requests and approvals.
    - The platform is user-friendly and responsive across different devices.

#### Tools, Libraries & Technologies

***Table 2 Tools , Libraries and Technologies***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tools,**  **Libraries and Technologies** | **Components** | **Tools** | **Version** | **Rationale** |
| Front-end | HTML, CSS,  and JavaScript | Latest | Develops a user-friendly interface. |
| Framework | React.js | Latest | For interactive and dynamic UI. |
| Backend | PHP | Latest | Widely supported on almost every server.  Open-source and free. |
| Frame work | larva | Latest | Built-in  authentication  system with  password encryption and role-based access  control. |
| Database | MySQL | Latest | Real-time data management. |

#### Work Division

Clear work division among group members to be shown

***Table 3 Work Devision***

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Roll Number** | **Name** | **Role Assignment and Work Division** |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | 2145 | Sehar Akbar | Database integration and API implementation. |
| 2. | 2124 | Iqra Khokhar | Prototype creation and front-end development. |
| 3. | 2102 | Komal Parveen | Backend development. |

#### Summary

Blood Bridge streamlines blood donation by connecting donors and recipients through an easyto-use online platform. It offers features like secure registration, donor search by blood type and location, real-time notifications, and inventory management. The system addresses key issues such as manual errors, delays, and lack of donor awareness. Developed using React.js and MySQL, Blood Bridge ensures data security, efficient operation, and user-friendly access, making it a vital tool for improving blood donation and saving lives

# Chapter No 2

# Literature Review

#### Literature Survey

Blood donation is an important process that saves millions of lives in the world. However, the traditional blood donation system often faces challenges such as the donor's unavailability, real time lack of communication and disabled request handling. Many existing platforms try to address these issues but come up with their limits. For instance, Sundas Foundation Sialkot, Seal Blood Donor, Sialkot Blood Donor, and UMT Blood Donor Society type platforms offer blood donation services, but tend to use manual coordination, which is slow and unreliable. Some lack of real-time donor tracking of the donors, while others have outdated interfaces that make the donor communication difficult. Blood Bridge enhances these systems, which by providing real-time donor search, location-based matching via Google API, and by providing a user-friendly web-based platform with instant notifications for blood requests. In contrast to conventional methods, it does away with the need for an intermediary blood bank, which provide quick and more direct communication between donors and recipients. By overcoming the limitations of current solutions, blood bridges offer more effective, safe and method of blood donation management.

#### Related Work:

* + - Existing Solutions:

To develop a competitive and worthy system, an analysis of existing blood donation platforms was conducted. The results include their strengths, weaknesses and room for improvement.

* + - Existing Platforms Examined:
      1. Sundas Foundation
      2. Sialkot Blood Donor
      3. Sial Blood Donor
      4. UMT Blood Donor Society
    - Strengths:
      1. Existing donor and recipient networks.
      2. Strong community trust and presence.
      3. Quick response to emergency blood request.
    - Weaknesses:
      1. Inadequate user-friendly and modern web interfaces.
      2. Limited real-time tracking and notifications.
      3. No Google Maps integration for donor search.
      4. Manual donor-recipient matching causing delays.

#### Gap Analysis:

**Gap Analysis** assists in identifying the difference between existing blood donation platform and our proposed **Blood Bridge**. Below, we indicate the main gaps in current solutions and how our system will be address them.

***Table 4 Gap Analysis***

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Existing Solutions (Sundas Foundation, Sial Blood Donor, Sialkot Blood Donor, UMT**  **Blood Donor Society)** | **Gaps Identified** | **Proposed**  **Solution in My FYP** |
| **Online Donor Registration** | Primarily manual registration or no organized database | No good donor records or history  tracking | A full online registration system with donor profiles |
| **Tracking Blood Availability in Real Time** | No computerized system to verify available blood units | Patients find it hard to find blood in time | Live tracking of available blood  across various blood banks |
| **Mobile Responsive Website** | No separate websites or only social media based groups | Not easy to access using mobile devices | A responsive web application for easier accessibility |
| **Emergency**  **Blood Request Feature** | Primarily volunteers responses on social media | Delays in emergency cases | One click emergency appeal to alert  passing  donors. |
| **Secure Donor Database** | No organized database to track donor history | It is difficult to locate repeat donors quickly | Safe system for maintaining donor profiles to be accessed later. |
| **Public Access for Blood**  **Requests** | Some platforms have limited access to specific communities or universities | General public unable to request blood | Donor lists & blood requests with open access, and verified users |

#### Summary

Blood Bridge is an advanced blood donation platform that solves the delays and inefficiencies of traditional systems. Unlike existing platforms that rely on manual processes, it offers realtime donor search, location tracking with Google Maps, and instant notifications. Blood Bridge directly connects donors and recipients, ensuring fast, secure, and efficient blood donation, especially in emergencies.

**Chapter No 3**

**Software Requirements Specification**

#### Requirements Analysis

Our Blood Bridge's Software Requirement Specification (SRS) document thoroughly outlines all the necessary functional and non-functional requirements for the project. We've carefully analyzed and pinpointed the essential features, user roles, and system interactions, making sure the platform effectively serves both donors and recipients. We've also established the system's boundaries, performance criteria, and security protocols to lay the groundwork for a practical and user-friendly solution. This detailed analysis acts as the foundation for our development process, ensuring every aspect of the system aligns with our project objectives and is meticulously organized.

#### User classes and characteristics

***Table 5 User classes and characteristics***

|  |  |
| --- | --- |
| User Class | User Characteristics |
| Donor | Users who can register on the system and respond to donation requests are able to donate blood. |
| Recipient | Users who can register on the system and respond to donation requests are able to donate blood. |

|  |  |
| --- | --- |
| Administrator | They have full access to the platform. They're responsible for verifying users, handling complaints, and keeping the system running smoothly. |
| Guest User | independent users can check out general information without any hassle |

##### Requirement Identifying Technique

###### Use Case Modeling:

* + - **Use case and their description:**

***Table 6 Use Case and their Description***

|  |  |
| --- | --- |
| User Registration | Users Like donor, recipient register their Accounts by providing details. |
| Blood Request | Allow recipient to request blood according to his/her need to  the donor. |
| Blood Request reject or approve | Allow donor to accept or reject recipient request |
| Notification Management | Notify the users regarding to blood request, approval, rejection and emergency request etc. |
| Admin Management | Admin can manage all the data, security and ensure system  rum smoothly. Admin can block a person who is not active while a time. |
| Contact Support | If users find any Kind of difficulties to manage account they can contact to the admin |
| Logout | Allow users to logout his/her accounts. If they are not interested to donate more. |
| **Use case** | **Description** |

.

##### Functional Requirements

The system will provide the following functional requirements:

Functional Requirement-1

***Table 7 FR-1***

|  |  |
| --- | --- |
| **Identifier** | **FR-1** |
| **Title** | User Registration |
| **Requirement** | The user shall be able to register by providing necessary details such as name, email, phone number, blood type, and location. The system shall  validate the information before storing it. |
| **Source** | End Users (Donors & Recipients) |
| **Inputs** | Name, Email, Phone Number, Password, Blood Type, Location |
| **Destination** | Stored in the database and displayed on the dashboard |
| **Outputs** | Confirmation message, User profile created |
| **Rationale** | Required for users to access system features such as finding donors or  requesting blood. |
| **Business Rule** | Users must provide a valid email and phone number for verification. |
| **Dependencies** | None |
| **Priority** | High |

Functional Requirement-2

***Table 8 FR-2***

|  |  |
| --- | --- |
| **Identifier** | **FR-2** |
| **Title** | User Login & Authentication |
| **Requirement** | The user shall be able to log in using a registered email and password. The  system shall authenticate the credentials and allow access to the dashboard. |
| **Source** | End Users (Donors & Recipients) |
| **Inputs** | Email, Password |
| **Destination** | Standard display (Dashboard) |
| **Outputs** | Authentication success or failure message |
| **Rationale** | Ensures secure access to system features |
| **Business Rule** | Users must enter valid credentials. If login fails 3 times, the system shall  lock the account for security. |
| **Dependencies** | FR-1 (User Registration) |
| **Priority** | High |

Functional Requirement-3

***Table 9 FR-3***

|  |  |
| --- | --- |
| **Identifier** | **FR-3** |
| **Title** | Find a Donor |
| **Requirement** | The recipient shall be able to search for donors by blood type and location.  The system shall display a list of matching donors. |
| **Source** | Recipient Users |
| **Inputs** | Blood Type, Location |
| **Destination** | Standard display (List of donors) |
| **Outputs** | List of available donors with contact details |
| **Rationale** | Helps recipients find matching donors quickly |
| **Business Rule** | Donors’ contact details shall be shown only to verified recipients. |
| **Dependencies** | FR-1 (User Registration) |
| **Priority** | High |

Functional Requirement-4

***Table 10 FR-4***

|  |  |
| --- | --- |
| **Identifier** | **FR-4** |
| **Title** | Request Blood |
| **Requirement** | The recipient shall be able to create a blood request specifying the required blood type,  urgency level, and location. The system shall notify suitable donors. |
| **Source** | Recipient Users |
| **Inputs** | Blood Type, Urgency Level, Location |
| **Destination** | Notifications to matching donors |
| **Outputs** | Request confirmation, Notifications sent to donors |
| **Rationale** | Facilitates quick blood donation and emergency support |
| **Business Rule** | Only registered users can request blood. Donors shall be notified only if they match the blood  type and location. |
| **Dependencies** | FR-1 (User Registration) |
| **Priority** | High |

Functional Requirement-5

***Table 11 FR-5***

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | Notifications for Donors |
| **Requirement** | The system shall send notifications to donors when a blood request matches  their profile. |
| **Source** | System |
| **Inputs** | Blood Request Details (Blood Type, Location, Urgency Level) |
| **Destination** | Donor Dashboard, Email, and Mobile Notifications |
| **Outputs** | Notification alert for donors |
| **Rationale** | Ensures quick response from donors when a request is made |
| **Business Rule** | Donors shall be notified only if they meet the request criteria. |
| **Dependencies** | FR-4 (Request Blood) |
| **Priority** | High |

Functional Requirement-6

***Table 12 FR-6***

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | User Dashboard |
| **Requirement** | The system shall provide a dashboard for users to manage their profile,  check requests, and track donation history. |
| **Source** | End Users (Donors & Recipients) |
| **Inputs** | User authentication (Login) |
| **Destination** | Standard display (Dashboard) |
| **Outputs** | User profile, Request history, Notifications |
| **Rationale** | Centralized access to all user activities and data |

|  |  |
| --- | --- |
| **Business Rule** | Users can edit only their own profiles and cannot access others' data. |
| **Dependencies** | FR-1 (User Registration), FR-2 (Login) |
| **Priority** | Medium |

Functional Requirement-7

***Table 13 FR-7***

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | Blood Donation History Tracking |
| **Requirement** | The system shall store and display the history of donations made by each  donor. |
| **Source** | Donors |
| **Inputs** | Donor ID, Date of Donation, Recipient Details |
| **Destination** | Standard display (Donor Profile) |
| **Outputs** | List of past donations |
| **Rationale** | Helps donors track their contributions and encourages future donations. |
| **Business Rule** | Only verified donations shall be recorded in the history. |
| **Dependencies** | FR-6 (User Dashboard) |
| **Priority** | Medium |

Functional Requirement-8

***Table 14 FR-8***

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Contact a Donor |
| **Requirement** | The recipient shall be able to contact a donor via the system’s messaging  feature. The system shall ensure privacy and security of communication. |
| **Source** | Recipients |
| **Inputs** | Recipient ID, Donor ID, Message Content |
| **Destination** | Donor’s Inbox |
| **Outputs** | Message notification for the donor |
| **Rationale** | Ensures a direct and secure communication method between donors and  recipients |
| **Business Rule** | Only verified recipients can send messages to donors. |
| **Dependencies** | FR-3 (Find a Donor) |
| **Priority** | Medium |

Functional Requirement-9

***Table 15 FR-9***

|  |  |
| --- | --- |
| **Identifier** | ER-9 |
| **Title** | Contact Support |
| **Requirement** | Users shall be able to submit queries or complaints to the system  administrator. The system shall log the request and respond accordingly. |
| **Source** | End Users |
| **Inputs** | Name, Email, Message Content |
| **Destination** | Admin Dashboard, Email |
| **Outputs** | Confirmation of request, Admin response |
| **Rationale** | Provides user assistance and resolves issues promptly |
| **Business Rule** | The admin must respond within 48 hours. |
| **Dependencies** | None |
| **Priority** | Low |

#### Non-Functional Requirements

#### Reliability

* + **Automatic Data Backup: Backup** of the database should be taken every Sunday 9 PM to prevent data loss
  + **System Uptime:** The system should be available **99% of the time** except during maintenance.
  + **Error Handling: If** any type of error occurs like server failure a friendly message appear instead of crashing the system

#### Usability

The system should be easy to use for all type of users including Donor, Recipient, Admin

* + **Mobile-Friendly Design:** The website should be **responsive**, its means it works properly on both computers and mobile devices.
  + **User Instructions:** Each page should have small **instructions** to guide users on how to use the system.
  + **Simple Navigation:** The system should have clear menu and button to facilitate users to find what they need.

#### Performance

. Performance show how system should response

* + **Search time:** User should be able to find the donor within **5,6** second after selecting Blood Type and Location
  + **Login Response time:** User should be authenticating within 5 sec**.**
  + **Data load time:** Pages load within 3,4 seconds

#### Security

Security ensure that the system and User data is secure from unauthorized access.

* + **User Authentication: Only** registered users can access their dashboard.
  + **Data Encryption:** Sensitive data like passwords should be **stored securely** using encryption techniques like **hashing**.

**Failed Login Attempts:** If a user enters the wrong password **3 times**, their account should be **temporarily locked for 10 minutes** to prevent hacking attempts

#### External Interface Requirements

As Our project is a web base so, it mainly requires a browser and a strong internet connection to run smoothly. Below are the necessary interface requirements to ensure the smooth Communication between system, Users, and external components.

#### User Interfaces Requirements

* + Design Standards

The website will employ a **dark theme** to ensure improved readability and a contemporary look.

There will be a consistent font style such as Roboto or Poppins throughout.

Icons and buttons will be reliably styled with simple labels to support navigation. It will be responsive scaling to suit mobile phones, tablets, and desktops.

* + Layout and Navigation

A fixed navigation bar will exist, containing:

Home

Find a Donor

Become a Donor Login/Register Contact Us

About us

Every page will have a footer containing contact info and social media links.

#### Software Interfaces

* Operating System: Window/Linux.
* Database: MYSQL for storing users' information.
* Frontend Technologies: HTML, CSS, JS for designing website.
* Google API: Google API use for location base donor search search
* IDEs: Visual studio
* internet connection: require for real time data update and communication

#### Hardware Interfaces

* Processor: Minimum intel core i3 or equivalent
* RAM: At least 4GB
* Storage:20GB

#### Communications Interfaces

* **Web Communication:** Web site communicate with server using HTTP/HTTPs protocols for secure data Transfer
* **Forms and button:** It will be used for interaction like login, registration, blood request etc.
* **Email notifications**: Email Notification can be sent to users for account verification and request update.
* **User Communication:** A contact us page allow user to communicate with admin Users can receive **notifications on their dashboard** about new blood requests or approvals.
* **External API:** Google Map API will be help user to find nearby a donor**.**

#### Use case Analysis

1. **Use Case #1 (User Registration)**

***Table 16 Use Case 1(User Registration)***

|  |  |
| --- | --- |
| **UC Identifier** | **UC1** |
| **Requirements**  **Traceability** | FR-1(User Registration functionally) |
| **Purpose** | Allow new users like donor, recipient create account to access the system |
| **Priority** | High |
| **Preconditions** | The user must have valid email and phone number |
| **Post conditions** | A new user account is created, and the user can log in |
| **Actors** | User (donor, recipient) |
| **Extends** | None |
| **Main Success Scenario** | 1. User clicks on the "Register" button. 2. System displays the registration form. 3. User enters required details and submits the form. 4. System validates the input data. 5. System sends a verification email. 6. User clicks the verification link. 7. System activates the user account. 8. User receives confirmation and can log in. |
| Alternate Flows | If the email already exists, the system prompts an error. |
| Exceptions | Network issue - System displays an error message.  Verification email not received - User can request a new verification email. |
| Includes | None |

#### Use Case #2(User Login)

***Table 17 Table 17 Use Case 2 (User Login)***

|  |  |
| --- | --- |
| **Use case**  **identifier** | **UC2** |
| Requirements Traceability | FR-2 (User Authentication) |
| Purpose | Allows registered users to log into their accounts. |
| Priority | High |
| Preconditions | User must be registered and email verified |
| Post conditions | User is successfully logged in. |
| Actors | User (Donor/Recipient) |
| Extends | None |
| Main Success Scenario | 1. User enters email and password. 2. System verifies credentials. 3. If valid, the user is redirected to the dashboard. |
| Alternate Flows | If the password is incorrect, the system displays an error message |
| Exceptions | Account is inactive - System prompts user to verify their email. |
| Includes | None |

#### Use Case #3(Find a donor)

***Table 18 Use Case 3 (Find a Donor)***

|  |  |
| --- | --- |
| **Use case**  **identifier** | **UC3** |

|  |  |
| --- | --- |
| Requirements Traceability | FR-5(search donor) |
| Purpose | Allows recipients to search for nearby donors by blood type and  location. |
| Priority | High |
| Preconditions | User must be logged in. |
| Post conditions | System displays a list of available donors. |
| Actors | Recipient |
| Extends | None |
| Main Success Scenario | 1. Recipient enters blood type and location. 2. System filters and displays matching donors. |
| Alternate Flows | If no donors are available, the system suggests alternative locations. |
| Exceptions | Network issue - System displays an error message. |
| Includes | Google Maps Integration |

#### Use Case #4(Request Blood)

***Table 19 Use Case 4 ( Request Blood)***

|  |  |
| --- | --- |
| **Use case**  **identifier** | **UC4** |
| Requirements Traceability | FR-6(Submit request) |
| Purpose | Allows recipients to request blood from available donors. |
| Priority | High |
| Preconditions | User must be logged in and verified. |
| post conditions | Request is sent to selected donor(s). |
| Actors | Recipient, Donor |

|  |  |
| --- | --- |
| Extends | None |
| Main Success Scenario | 1. Recipient selects a donor and submits a request. 2. System notifies the donor. 3. Donor responds to the request. |
| Alternate Flows | If no donor responds, the system sends a reminder. |
| Exceptions | Donor declines - System notifies the recipient. |
| Includes | Notification |

#### Use Case #5(Approve/Reject Request)

***Table 20 Use Case 5(Approve/Reject Request)***

|  |  |
| --- | --- |
| **Use case**  **identifier** | **Uc5** |
| Requirements Traceability | FR-7(**A**pprove/Reject Reques**t)** |
| Purpose | Allows donors to accept or reject blood donation requests. |
| Priority | High |
| Preconditions | User must be logged in. |
| post conditions | Recipient is notified of the donor’s decision. |
| Actors | Donor |
| Extends | None |
| Main Success Scenario | 1. Donor receives a request notification. 2. Donor accepts or rejects the request. 3. System updates the request status. |
| Alternate Flows | If the donor does not respond within 24 hours, the request is marked as "pending." |
| Exceptions | Donor's account is inactive - System informs the recipient. |
| Includes | None |

#### Use Case #6(Notifications for Donors)

***Table 21 Use Case 6 ( Notification for Donors)***

|  |  |
| --- | --- |
| **Use Case ID** | **UC-5** |
| **Use Case**  **Name** | Notifications for Donors |
| **Actors** | System, Donor |
| **Preconditions** | A blood request exists matching the donor's criteria. |
| **Triggers** | A recipient submits a blood request. |
| **Flow of**  **Events** | 1. The system checks for matching donors. 2. The system sends  notifications via email, SMS, and dashboard alerts. 3. The donor receives the notification and can respond. |
| **Post**  **conditions** | Donor is informed about the blood request. |
| **Exceptions** | If notification sending fails, the system retries after a short delay. |

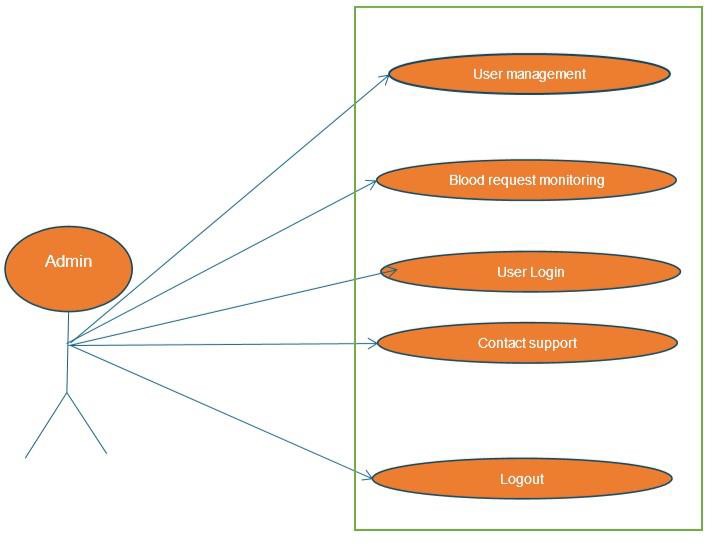
#### Use Case #7(Contact a Donor)

***Table 22 Use Case 7 ( Contact a Donor)***

|  |  |
| --- | --- |
| **Use Case ID** | **UC-6** |
| **Use Case**  **Name** | Contact a Donor |
| **Actors** | Recipient, Donor, System |
| **Preconditions** | Recipient must have found a matching donor. |
| **Triggers** | Recipient wants to contact a donor. |
| **Flow of**  **Events** | 1. Recipient selects a donor from the search results. 2. Recipient sends a message through the system. 3. System notifies the donor about the  message. 4. Donor can reply to the message. |
| **post**  **conditions** | Recipient and donor can communicate securely. |
| **Exceptions** | 1. If the donor is inactive, the system notifies the recipient. 2. If messaging  fails, the system retries later. |

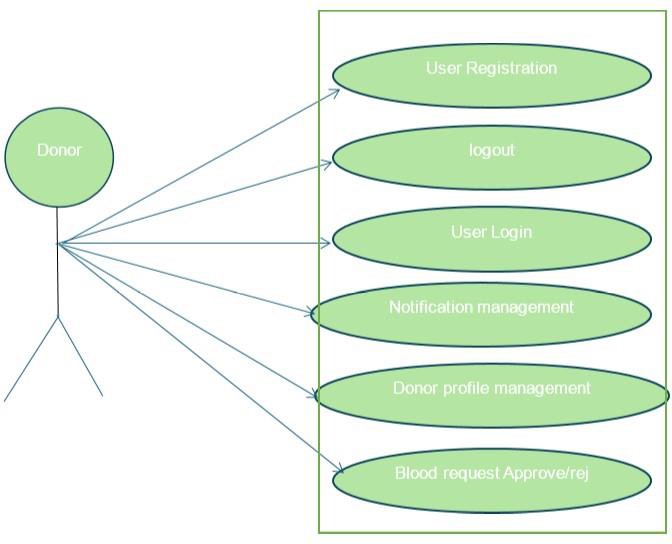
#### Use Case Diagram

Use case diagram for admin



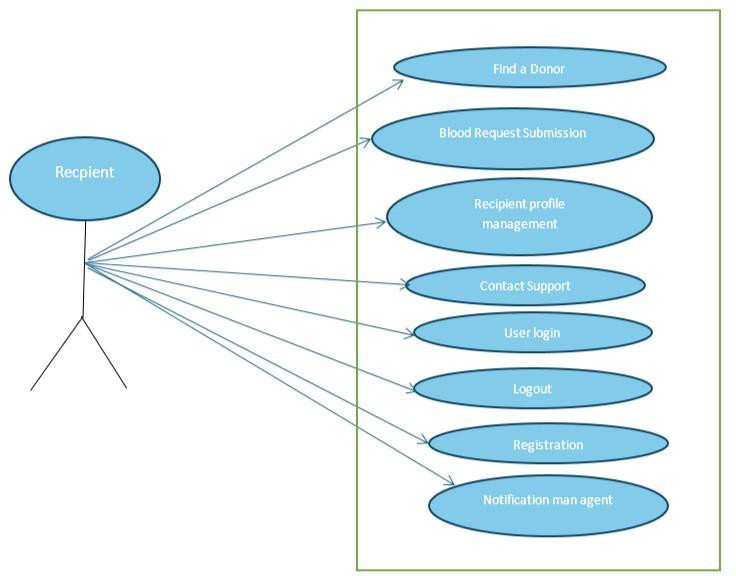
**Figure 1 Use Case Diagram for Admin**

Use case Diagram for donor



**Figure 2 Use Case Diagram for Donor**

Use Case Diagram for Recipient:



**Figure 3 Use Case Diagram for Recipient**

# Storyboarding:

***Figure 4 Story boarding***

# Summary

The process of identification of requirements It is important to define the functionality and expectations of Blood Bridge. This includes the system to meet functional and non-functional needs to meet the needs of the system. To identify the requirements, we used techniques using case analysis, which helps to break up system interactions in a structured manner. Use of each case underlines the stages involved in various system functionalities, ensuring clarity in development. These use cases help to design user interactions, which smoothly ensure DonorRisking connections, blood request handling, and user management. Requirements Engineering Process plays an important role.

**Chapter No 4**

**Software Design Specification**

#### System Design

The Blood Bridge is a web-based application that facilitates direct communication between blood donors and recipients. It operates as a standalone system but interacts with external services such as email and SMS APIs for notifications and the Google API for locationbased services.

#### Design Considerations

|  |  |
| --- | --- |
|  | **Assumption and Dependence** |
| o | Users have a stable internet connection to reach the stage. |
| o | The donor and recipients provide accurate information during registration. |
| o | Google Map API is available and is functioning correctly. |
| o | Email and SMS services are reliable to send information. |
| o | The system will be hosted on a server that supports high availability and safety. |
|  | **Limitations:** |
| o | The platform does not support offline functionality. |
| o | Users must have an internet connection to access services. |
| o | If a donor does not respond to the request, the system cannot guarantee blood  availability. |
| o | The search functionality depends on the accuracy of the data provided by the donors. |
| o | The initial version can only support cities or regions in limited numbers. |
|  | **Risks:** |
| o | **System down time: If** the server crashes, users will not be able to reach the platform. |
| o | **Data Breach:** There is a risk of unauthorized access to user information. |
| o | **False information**: Users can enter incorrect details. |
| o | **High Traffic Load**: Increase in users can slow down the system. |

#### Requirements Traceability Matrix

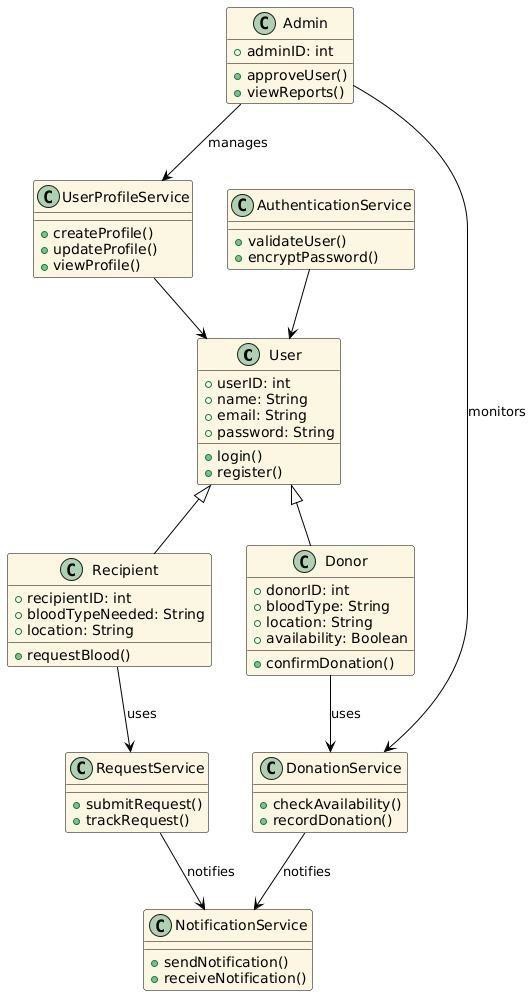
Here a requirements traceability matrix (RTM) for our Blood Bridge, ensuring that each requirement is mapped for a specific design specification

***Table 23 Requirements Traceability Matrix***

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Requirement Description** | **Design Specification** |
| R1 | The system should allow users (donors and recipients) to register and log in safely | Component: **User authentication module** |
| R2 | The system will have to store and recover donor details including name, blood type,  location and availability status | Component: Donor Management Module |
| R3 | Users should be able to find blood donors  depending on blood type and location | Components: Search and  Filter Algorithm |
| R4 | The system should allow recipients to  request blood from the available donors. | Component: **Blood request**  **module** |
| R5 | The donors should accept/reject recipient  request | Component: **Notification**  **System** |
| R6 | The system should allow users to update their profiles including contact information  and medical history. | Components: **Profile Management Modul**e |
| R7 | The administrator must be able to verify  and manage users (approval, deactivate or remove accounts). | Component: **Admin Management Pane**l |
| R8 | Google Maps API should be integrated to  show the system at donor locations. , | Components: **Google Maps**  **API Integration** |
| R9 | The system should encrypted sensitive user  data for safety purposes | Components: **security** |
| R9 | The platform should be accessible on  mobile and desktop browsers | Components: **responsible**  **web design** |

#### Design Models

##### Design Class Diagram (DCD)



***Figure 5 Class Diagram***

#### Architectural Design

It describes the pattern and techniques used to design and build an application. The architectural design of this

* + - High-level system architecture:

The system consists of three primary layers;

* + - Presentation layer:

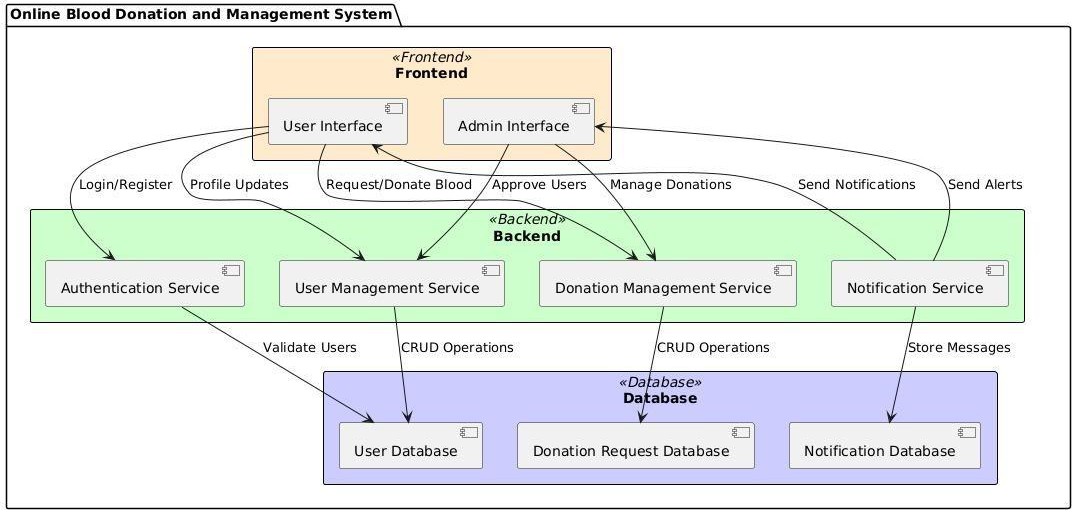
Responsible for user interaction and UI/UX. HTML, CSS, is designed using JavaScript. Users can find donors, request blood, register and log in.

* + - Application layer:

Request of procedures from the front. The user applies main functionalities such as authentication, blood request handling, donor search and information. Node.JS, is developed. Communication with database.

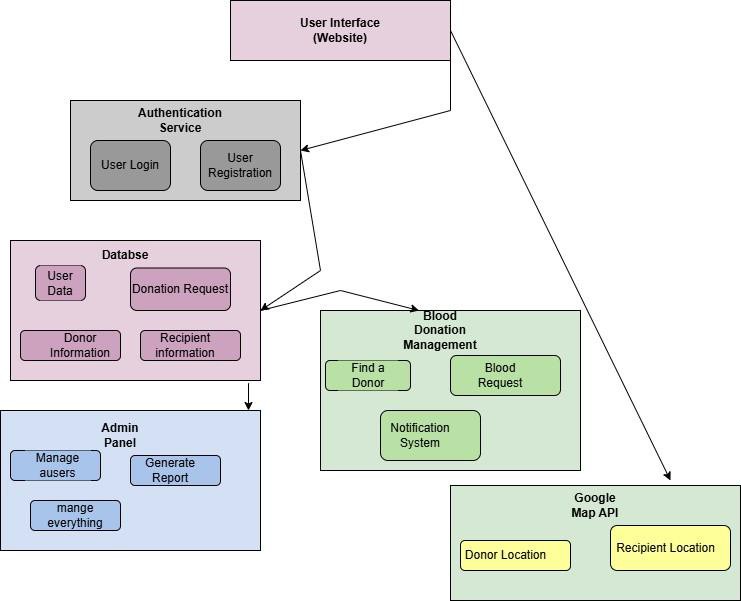
* + - Data layer:

Manages information about stores and users, donors, requests and history. MySQL uses PostgreSQL. Data ensures stability, safety and recovery efficiency.



***Figure 6 Architecture Design***

##### UML Component diagram



***Figure 7 UML Component diagram***

###### Mapping Modules to Architecture Pattern (MVC + Multi-Tier)

***Table 24 (MVC + Multi-Tier)***

|  |  |
| --- | --- |
| **Architecture Layers** | **Mapped Components** |
| Presentation Layer (View  - Frontend) | HTML, CSS, JavaScript for UI, user interactions, and data display. |
| Application Layer (Controller - Backend) | Handles authentication, blood request processing, donor search, and notifications using APIs. |
| Data Layer (Model - Database) | Stores and retrieves users, blood requests, and donation history. |

#### Data Design

The database consists of many entities, represents a major component of each system. Below are the major tables and their objectives:

1. **Users Table** *(Stores general user information, including donors, recipients, and admins.)*

***Table 25 User Table***

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| User\_id | INT (PK, AUTO\_INCREMENT) | Unique identifier for each user. |
| Name | VARCHAR (100) | Full name of the user. |
| Email | VARCHAR (255) (Unique) | Email for login and notifications. |

|  |  |  |  |
| --- | --- | --- | --- |
| Password | VARCHAR (255) | | Encrypted password for security. |
| Phone No | VARCHAR (15) | | Contact number. |
|  |  | |  |
| Location | VARCHAR (255) | | User’s city or area for donor search |
| Role | ENUM ('donor',  'admin') | 'recipient', | Defines the user’s role in the system. |

1. **Donor Table** *(Stores information specific to donors.*

***Table 26 Donor Table***

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| user\_id | INT (FK) | Linked to the users table. |
| Donor-id | Unique donor ID. | Unique donor ID. |
| Blood-Type | ENUM ('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+', 'AB-') | . Blood group of the donor. |
| Availability- status | BOOLEAN | is |
| Indicates whether the donor available. |

1. **Blood Requests Table** *(Tracks blood requests made by recipients.)*

***Table 27 Blood Request Table***

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| request\_id | INT  AUTO\_INCREMENT) | (PK, | Unique request ID. | | |
| Recipient-id | INT (FK) | | table | (who | is |
| Linked to users  requesting). |  |  |
| Blood- type |  | | Required blood type.. | | |
|  | ENUM ('A+', 'A-', 'B+', 'B-', 'O+',  'O-', 'AB+', 'AB-') | |  | | |
| Location | VARCHAR (255) | | Where blood is needed. | | |
| status | ENUM ('pending', 'matched', 'completed', 'canceled') | | Status of the request. | | |
| Match-donor-id | INT (FK, NULL) | | Linked to donor table when matched. | | |

1. **Notifications Table** *(Stores system notifications for users.)*

***Table 28 Notification Table***

|  |  |  |  |
| --- | --- | --- | --- |
| ***Column Name*** | ***Data Type*** | | ***Description*** |
| Notification\_id | INT  AUTO\_INCREMENT) | (PK, | Unique request ID. |
| User\_id | INT (FK) | | Linked to users table (who will receive it). |
| Message | TEXT | | Notification content. |
| Status | ENUM ('unread', 'read') | | Notification status. |

|  |  |  |
| --- | --- | --- |
| Created-at | TIMESTAMP | When the notification was sent. |

1. **Admin Table** *(Stores information about admin users who manage the system.)*

***Table 29 Admin Table***

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| Admin\_id | INT (PK,  AUTO\_INCREMENT) | Unique ID. |
| User\_id | INT (FK) | Linked to users' table |
| Role | ENUM ('super\_admin', 'moderator') | Admin privileges. |

#### Data Dictionary

The **Data Dictionary** covering system entities, functions, and object-oriented components.

Alphabetical List of System Entities or Major Data with Types and Descriptions

***Table 30 Terminology and Description***

|  |  |
| --- | --- |
| **Terminology** | **Description** |
| Admin (Object) | The person who is responsible for the management of users and  system operations. |
| Blood Donation (Object) | Record of a donation program including donor details and location |

|  |  |
| --- | --- |
| Blood Group (Attribute) | The blood type of a donor or recipient (e.g., A+, B-B-O+). Data **Type: String.** |
| Contact (Attribute) | User contact number for communication. **Data Type: String.** |
| Donor (Object) | A user who donates blood, including personal details and charity  history |
| Email (Attribute) | User's email for certification and information **Data Type: String.** |
| Location (Object) | Geographical coordination or address of a donor or recipient. |
| Notification  (Object) | Alert for donation requests, approval and other updates. |
| Password  (Attribute) | Authentication of a user credentials. **Data Type: String**. |
| Recipient (Object) | A user who requests blood donation. |
| Request (Object) | A blood donation request presented by a recipient. |
| Status (Attribute) | The current state of a blood request (e.g., pending, approved, full). Data Type: String. **Data Type: String.** |
| User (Object) | A common unit that represents both donors and recipients, including  certification details. |
| User ID (Attribute) | Unique identifier for a user. **Data Type: Integer**. |

Structured Approach: Functions and Function Parameters:

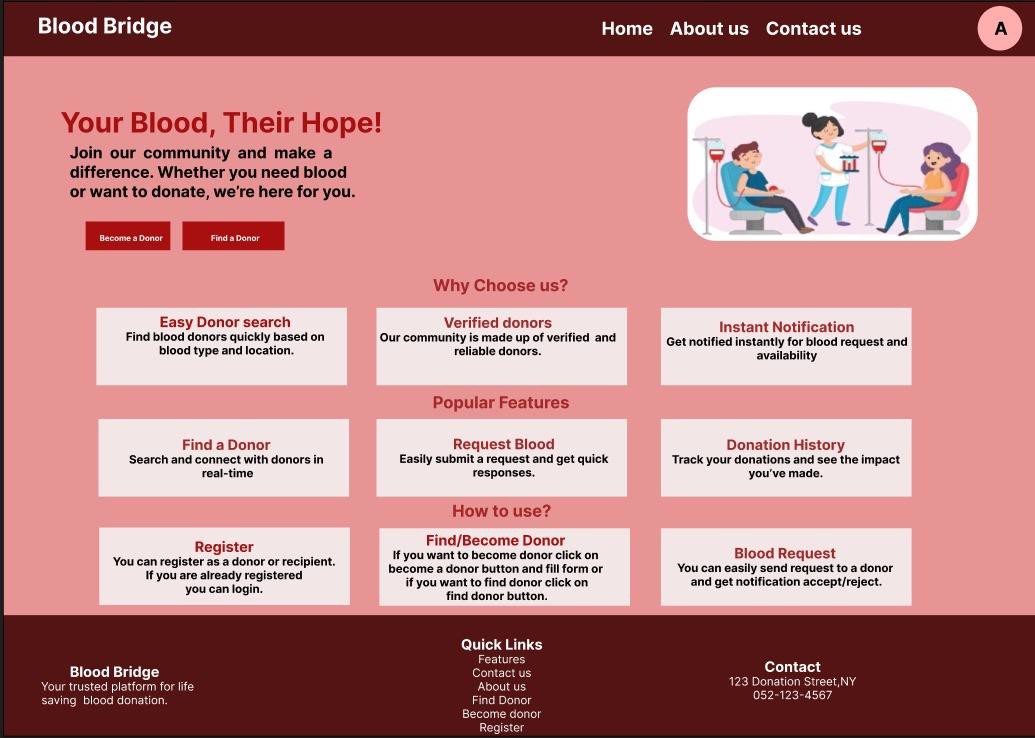
***Table 31 Function Name and Function Parameter***

|  |  |  |
| --- | --- | --- |
| **Function Name** | **Description** | **Parameters** |
| Register User () | A new user registers by storing its details in the system. | **User name** (string) - user name, **emai**l (string) - user email, **password** (string) - password for safety, **usertype** (string) - it defines whether the user is a donor or  recipient |
| Login User () | Authenticates a user and grants access if credentials match. | **email** (String) – Registered email of the user, **password** (String) – Password for authentication. |
| Find a Donor () | Search the donors provided by blood type and location. | **blood Group** (String) – Required blood type, **location** (String) – Recipient’s location for donor search. |

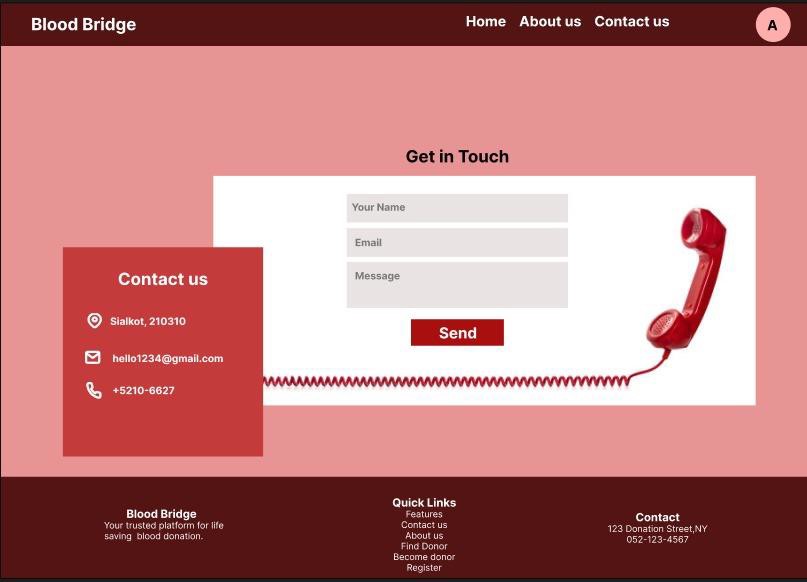
|  |  |  |
| --- | --- | --- |
| Send Request () | Sends a blood request to a donor. | **recipient ID** (Integer) – ID of the recipient, **donor ID** (Integer) – ID of the donor, **blood Group** (String) – Required blood type, **location** (String) – Location  for donation. |
| Update Profile () | Allows users to update their profile details. | **userID** (Integer) – ID of the user, **name**  (String) – Updated name, **contact** (String)   * Updated contact info, **address** (String) * Updated address. |

#### User Interface Design 4.7.1Screen Images

**Home page :**

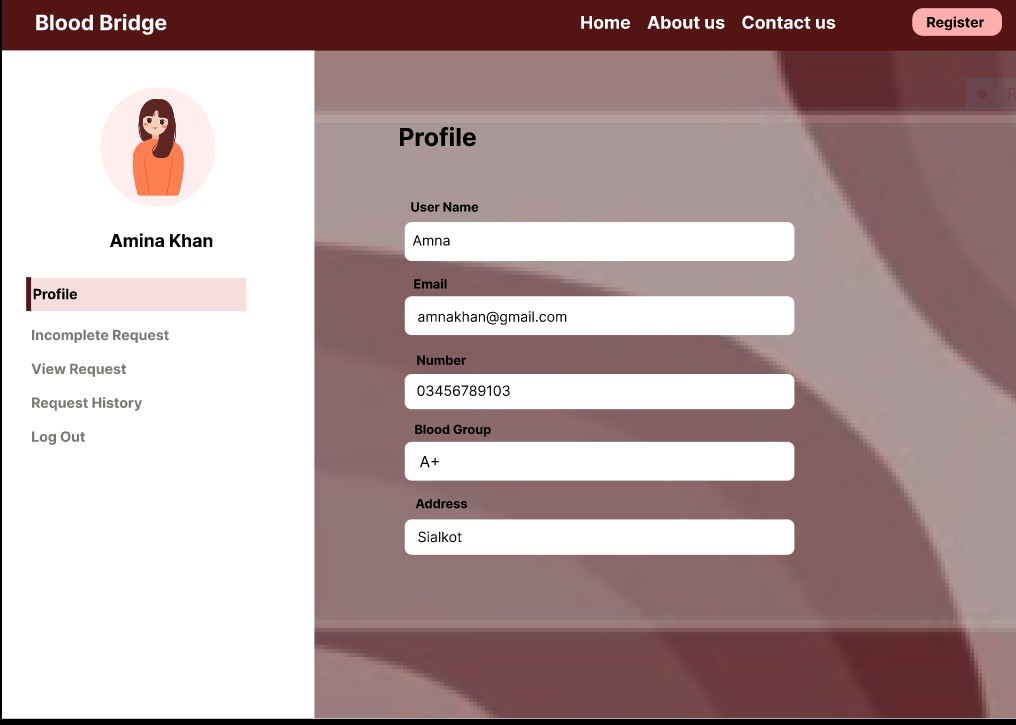


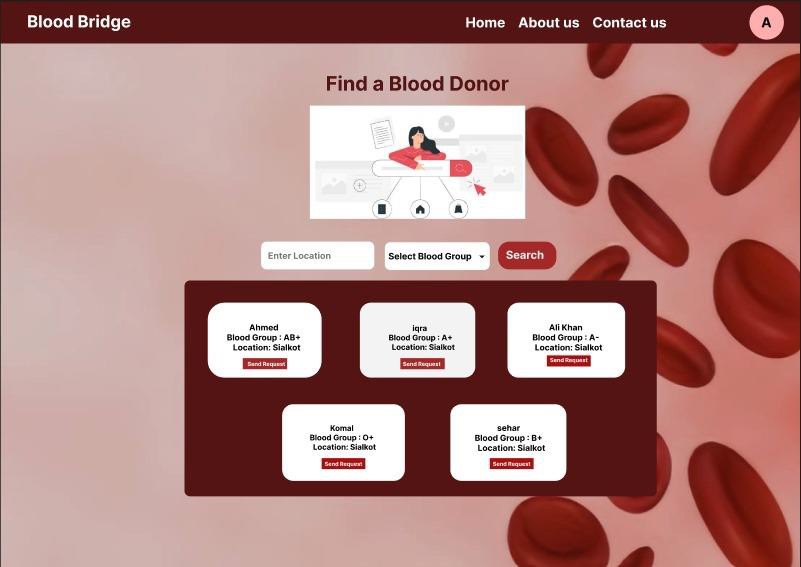
**Contact Us page**

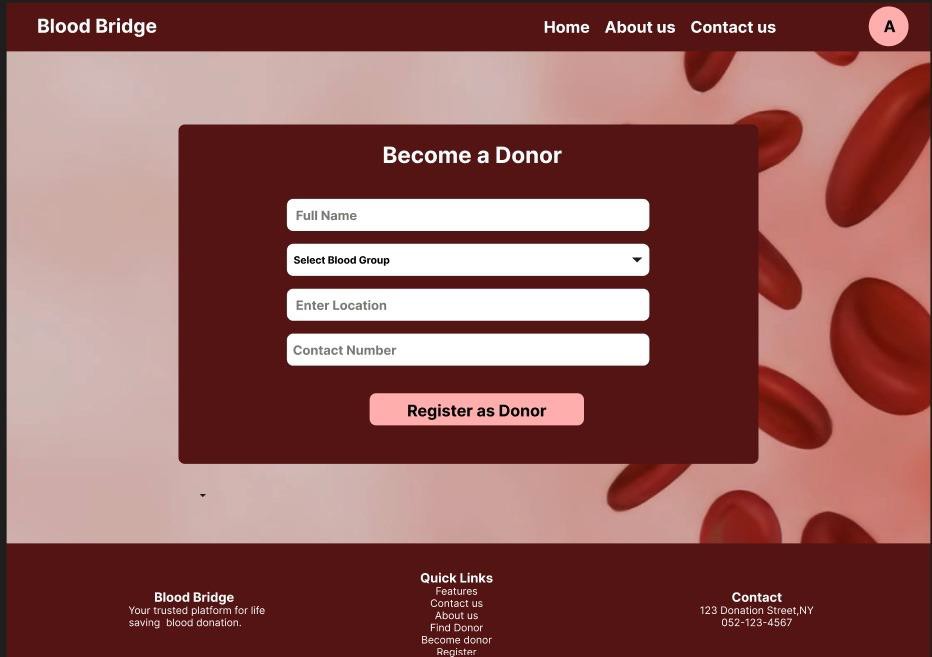


**Registration \ login page**









#### 4.7.2 Screen Objects and Actions

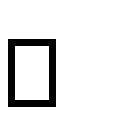
***Table 32 Screen Object and Action***

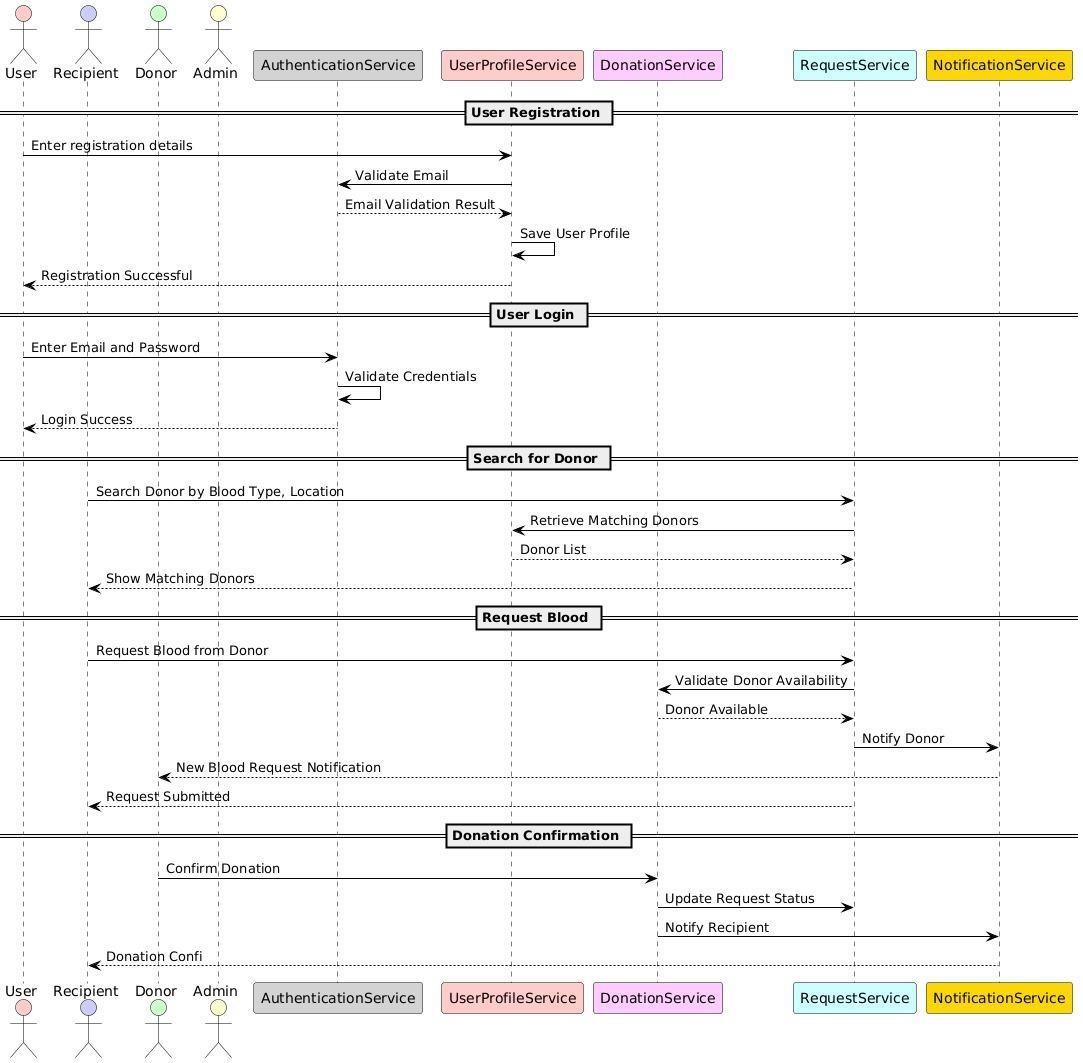
|  |  |
| --- | --- |
| **Object** | **Action** |
| Search bar, donor list, request button, information. | The user searches for donors, selects one, and submit a request. The system informs the donor and updates the request position. |
| Registration/login form, submit button, error message, dashboard link | User registers or logs in. The system verification  redirects on credentials and dashboards or shows an error. |

#### Design Decisions

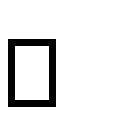
* + - **Integrated request to find a donor:** simplifies navigation and improves user experience.
    - **Real time notifications:** Accelerated donor ensures reaction.
    - **Google Maps AP**I: The accurate donor provides tracking.
    - **Customized database**: Manages users and requests efficiently.
    - **Safe authentication:** protects user data.
    - **User friendly UI:** Ensures simple navigation.
    - **Administrator Panel:** Easy System Manage Allows

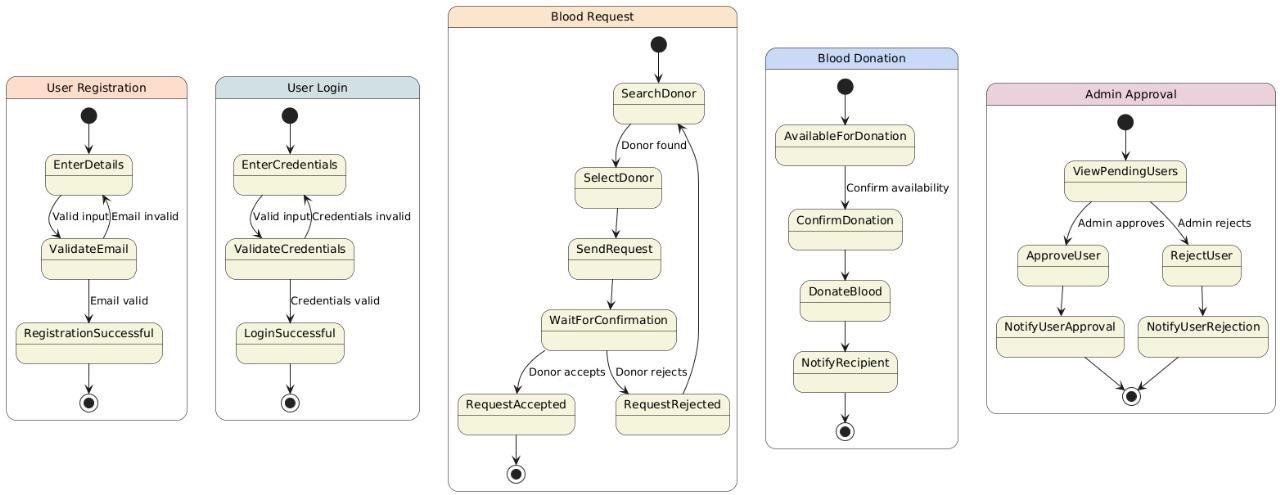
#### Behavior Model

Sequence Diagram



***Figure 8 Sequence Diagram***

State Diagram



***Figure 9 State Diagram***

#### Summary

This chapter outline the key design elements of on-line blood donation and management system, that focus on component diagrams, screen objects and interactions. Integration of the request feature in a donor enhance useability, while real time notification improves donor response time. The use of Google Maps API guarantees correct donor area tracking, and safe authentication protects person information. The database is structured for efficiency, supporting seamless operations. These design choices align with the cause of the project to create a user- friendly and reliable platform for blood donation control.

##### Reference

From Internet.