Software Requirements Specification

for

Blood Donation Portal

Version 1.0 approved

Prepared by:- Morium Chowdhury(2252421008)

Sadia Iffat(2252421012)

Rifah Zakiah(2252421024)

Hamza Bin Arif(2252421032)

Tafsir Un Nahian(2252421094)

Bangladesh University of Professionals

Date created: 20/04/2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1.   Introduction

1.1   Purpose

The purpose of this Software Requirements Specification (SRS) report is to determine the functional and non-functional requirements of the Blood Donation Portal. The report provides the foundation on which the system will be established, declaring the design, implementation, testing, and maintenance phases of the software project's requirements.

This SRS covers the fundamental operation of the portal, e.g., user registration, handling of donation requests, donor blood tracking, and event management. It also addresses the administrative controls, reporting, and external system integration (e.g., health services). The system is designed to facilitate the process of blood donation by allowing for the registration of donors, monitoring of donation events, and requesting or donating blood. The system seeks to ease the process of blood donation through facilitating donor registration, monitoring donation activity, and donation of blood or requesting blood, as outlined by the Blood Donation Guidelines offered by the World Health Organization (WHO) [1]. There is no inclusion of hardware requirements, network infrastructure, and detailed database design in this report, which will be dealt with in the later stages of the project.

1.2   Document Conventions

This SRS conforms to the guidelines and standards of the IEEE Software Engineering Standards (IEEE 830-1998) [3], which defines the recommended structure and format of Software Requirements Specifications. This document conforms to these standards to ensure consistency, clarity, and completeness in documenting the functional and non-functional requirements.

Font Style: Titles are in bold. The section title is 14 pt font size, and the text is 12 pt font size. The body text is in Times New Roman.

Terminology: Some specific words, such as "donor," "admin," and "even,t" have uniform use throughout the document. They are also clarified in more detail in Section 2.3.

Hyperlinks and References: External document references are linked in underlined, blue text. Hyperlinks to internet resources are included where relevant.

Use of Tables and Diagrams: Requirements can be enumerated in tabular form or as diagrams to make it easier to understand, especially where there are complex workflows or data exchanges.

The Blood Donation Portal Design Team User Interface Style Guide [2] shall be followed in every user interface design to make it uniform and aligned to visual standards for all the pages and portal features.

1.3   Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) is intended for a general audience of stakeholders who participate in the development, deployment, and utilization of the Blood Donation Portal. The following are the recommended sections for each audience:

Developers: Developers can begin with the reading of Chapter 3: External Interface Requirements, Chapter 4: System Features, and Chapter 5: Other Nonfunctional Requirements. These chapters provide accurate descriptions of the system components, features, and technical information. Then, developers may refer to Chapter 6: Other Requirements for any other system constraints.

Project Managers: The project managers may start with Section 1.4: Product Scope, Section 2.1: Product Perspective, and Section 2.7: Assumptions and Dependencies. These sections give an overview of the project, business context, and assumptions required for planning, timeline, and resource planning. Then, Chapter 4: System Features will help in knowing the system functionality from a management perspective.

System Testers: Testers may begin with Chapter 3: External Interface Requirements to understand how the system communicates with users, hardware, and other software. Next, Chapter 5: Other Nonfunctional Requirements and Chapter 6: Other Requirements can be read in preparation for performance testing, security checks, and system validation.

Stakeholders and End Users: End users, marketing staff, and business analysts should begin with Section 1.4: Product Scope, which provides a concise understanding of what the system is for, does, and the benefits it provides. Section 2.3: User Classes and Characteristics in Chapter 2: Overall Description provides an overview of the users and their requirements. Chapter 4: System Features also gives an idea of who will be utilizing the system.

Technical Writers and Documentation Specialists: Technical writers should concentrate on Chapter 3: External Interface Requirements and Chapter 4: System Features for composing user guides, API documentation, and user manuals. These two chapters have extensive information that is critical in describing the system's operation, features, and user interactions.

To discover the overall goals and benefits of the system, all levels of readers are encouraged to begin with Section 1.4: Product Scope and Chapter 2: Overall Description before investigating in greater depth specific technical or functional components.

1.4   Product Scope

The Blood Donation Portal will be designed to have a seamless and streamlined blood donation process by providing a platform where donors are brought together with hospitals, blood banks, and healthcare professionals. Users will be registered as donors on the portal, track donation histories, and be informed of impending donation sessions. It will further enable healthcare units to order blood donations, manage donations, and coordinate donors.

The objectives of the portal are:

● Optimize blood donation processes for hospitals and blood banks using automated donor management and requests.
● Encourage more individuals to give blood using an easy-to-use portal with reminders, educational resources, and easy interfaces.

● Increase efficiency by integrating with current healthcare systems for donor management and blood management.

The portal aims to do the following:

● Streamline access for potential donors to track and manage donations.

● Increase participation via reminders, notifications, and educational materials on the importance of blood donation.

● Promote transparency by offering administrative access and reports to track donation activity.

Enabling the health sector as well as society as a whole, the Blood Donation Portal assists the corporate mission to improve public healthcare systems and guarantee blood supply to those in need.

1.5   References

[1] World Health Organization (WHO), Blood Donation Guidelines, 2020. [Online]. Available: https://www.who.int/bloodsafety/guidelines/en/. [Accessed: May 6, 2025].

[2] Blood Donation Portal Design Team, User Interface Style Guide, Version 1.2, March 2025. [Internal document].

[3] IEEE, Software Engineering Standards (IEEE 830-1998). [Online]. Available: https://standards.ieee.org/. [Accessed: May 6, 2025].

2. Overall Description

This chapter will give an overview of the Blood Donation Portal. It will outline the system's context to show how it will work and introduce its basic functionality for each stakeholder. The design constraints and assumptions for the system will also be explained.

2.1 Product Perspective

The "Blood Donation Portal" is an independent system that seeks to automate the blood donation process so that donors, hospitals, and blood banks are well-connected. It is a new system with no direct predecessor. Still, it seeks to improve existing manual systems or partially automated systems by providing a user-friendly, centralized online system for the management of blood donations.

The system will be used by non-profit organizations, clinics, and hospitals involved in blood donation drives to manage donor registration, match donors with blood requests, and maintain donation histories.

The application will be a web-based, stand-alone system with a simple interface for administrators and donors. Hospitals and blood banks will integrate it into their in-house systems, such as patient management and inventory systems, to manage blood supply efficiently.

A simplified component diagram is given below. This diagram is enough to explain how the components of the system interact with each other and the overall environment.
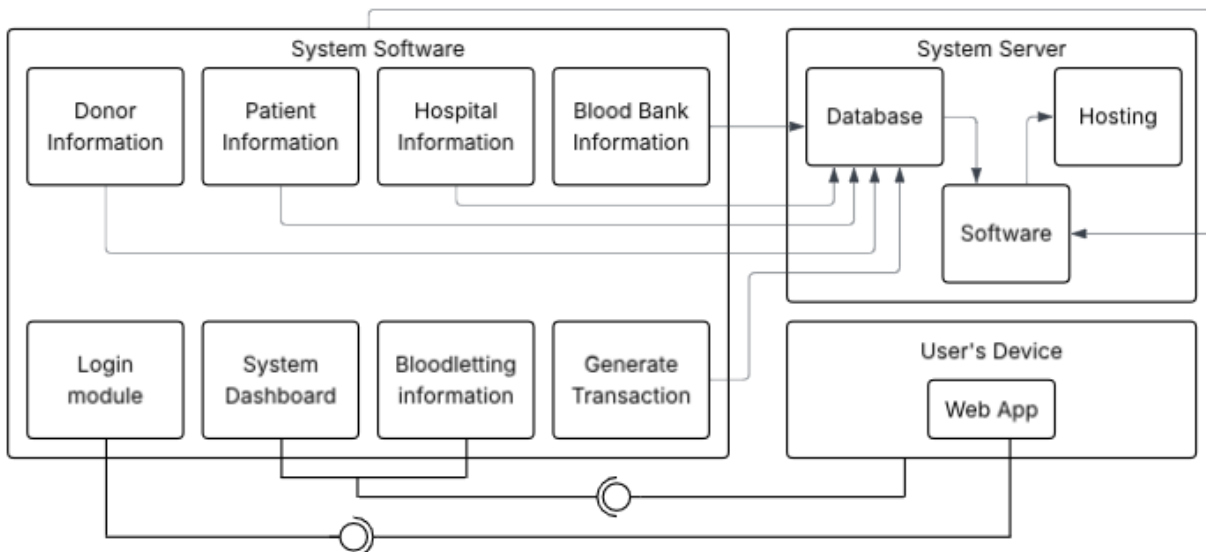
Fig. 2.1.1. Simplified component diagram

This product can be made as part of the healthcare system, which is designed to interface with several external entities such as hospitals, clinics, blood collection centers, patients, regulatory bodies, and donors. It must fulfill the relevant regulatory norms and also must be integrable with other health systems, such that the blood supply management can be made effectively and efficiently. It is composed of various internal components that work together to provide the intended functionality.

The components include the user interface, data management system, data storage system, data processing system, and security system. Each component has its role and functionality in the portal and works in conjunction with other components to make the overall system function as expected.

2.2   Product Functions

The system product functions are those attributes and capabilities that enable the successful and efficient handling of blood supply operations in a blood bank or a blood donation organization. The primary functions of the "Blood Donation Portal" are, but not restricted to:

●   **User Registration & Login:** Recipients, donors, hospitals, and blood banks can register and create a user account and log in to avail individualized options like donation history and blood requests.

●   **Donor Management:** The software should be able to manage the data of the donors, i.e., personal information, medical history, history of donations, blood group, and preferred donations.

●   **Blood Donation Request:** Blood donation requests may be initiated by users (patients, blood banks, or hospitals) whenever there is a demand in blood banks or hospitals. They may also request urgent blood requests during emergencies, which become visible immediately to donors.

●   **Request Management:** It must be feasible for the system to generate, update, and track blood requests, including urgency, blood group, quantity, compatibility, and receiving hospital.

●   **Donation History & Tracking:** Donors and the system can track their donation history as well as previous donations and preparedness for future donations.

●   **Blood Compatibility Checking:** It must be ensured by the system that there is compatibility between the donor's blood group and the patient's requirement.

●   **Inventory Management:** The system should be capable of managing blood inventory by monitoring inventory levels, expiration dates check, and blood usage management.

●   **Administration & Moderation:** The system should be capable of giving an admin interface to manage users (approve new users and requests), donations tracking, approve requests, spam/content filter, suspend users, and view system activity logs.

●   **Feedback & Rating:** The site must be able to give post-donation feedback collection and reliability scoring of donors and recipients. It must also allow people to rate their experience and provide feedback for the improvement of the site.

2.3 User Classes and Characteristics

User classes and characteristics of the Blood Donation Portal refer to the different types of users who interact with the system and their specific roles, responsibilities, and characteristics. The user classes and their characteristics of the proposed system are detailed in the table below:

Table 2.3.1: User classes with their characteristics:

User Class	Characteristics	Priority	Key Activities	Technical Expertise
Donors	- Volunteer individuals offering to donate blood - Frequent/occasional donors. - Requires a simple UI for scheduling and tracking.	High	Register, update profile, receive & respond to alerts	Low to moderate
Patients	- Patients or family members. - Urgent needs for medical treatment. - Medical personnel requesting on behalf of patients with emergency needs and no attendants.	High	Post requests, search for donors or blood banks, and track request status	Low to moderate
Hospitals	- Medical staff managing bulk requests. - Requires secure, high-volume access.	Medium	Request blood, proceed with donation procedures	High
Blood Banks	- Administrators for updating stock. - Managing workflows.	Medium	Update inventory, receive blood donations, and request blood donations	High
System Administrators	- Manages user roles, security, and reports. - Responsible for system oversight, content moderation, and user management.	Critical	Approve and verify users, monitor logs, and manage content	High

Guest	Unregistered visitors browsing general information about blood donation	Low	View FAQs, user manual, and learn about the donation procedure	Low

Each class will be given access to specific features depending on their role in the system, ensuring that sensitive data is only accessible by authorized users. Certain features—such as content moderation and audit logs—are restricted to the Administrator class, whereas Guests have read-only access to public information.

2.4   Operating Environment

Operating Environment defines a technical overview of our service specifications, together with the way the sub-components interoperate. The "Blood Donation Portal" will operate in a web environment, accessible through major browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. The system will be compatible with:

Hardware Platforms: Desktop,
- Processor: Intel(R) Core(TM) i3- OR more.
- RAM: 8GB OR more/ 4GB or more.
- Hard Disk: 12GB or more/ 8GB or more.

Operating System: Windows 10 OR more.
Web Server: Apache with PHP and MySQL backend.
Database: MySQL for storing user data, blood requests, and donation history.
Front end: HTML5, CSS, and JavaScript.
Client Applications: Modern web browsers (Chrome, Firefox, Edge, Safari).

2.5   Design and Implementation Constraints

The design and implementation of a Blood Donation may be subject to various constraints relevant to the design and implementation of the system. Some of the more common design and implementation constraints that are subject to design consideration during the implementation of the system are:

Technology Restrictions: A system may require certain technology environments or platforms, e. g. operating systems, databases, web browsers. Developers may need to be aware of the technology environment's compatibility as well as its scalability to make sure the system can support the expected workload.

Time constraints: There may be project deadlines or delivery schedules imposed on the system that require a certain level of system design or development prioritization (as well as other decisions) to ensure that the high-priority requirements of the system are being met on schedule.

Resource limitations: The system could be subject to budget constraints (whether in terms of development personnel, resources, or hardware), and the developer would have to look for low-cost development tools and methods to deliver the project on budget.

Regulatory Constraints: Depending on the scope of the system, there may be some regulatory standards and requirements (e. g. medical standards, data privacy laws, etc.) for which the developers need to make sure the system's design, and thus its implementation, is in line with the applicable standards and regulations.

User Constraints: The system needs to be user-driven. Developers will sometimes have to think about the usability and accessibility of the system so that the system works well for all users.

Security Requirements: Confidential user data (personal data, health information) must be encrypted, and the system should comply with data protection laws, like GDPR or HIPAA. Developers must follow OWASP Top Ten; use HTTPS, input validation, role-based access control, and encryption at rest.

Scalability: The system must be able to cope with lots of users and traffic, especially in times of crisis or massive blood donation campaigns.

Coding Standards: The code should follow industry best practices and internal style conventions; be accompanied by unit tests that are at least 80% coverage.

2.6 User Documentation

User documentation for an application is a written description or overview of how to utilize an application in a useful manner. It might include online help, tutorials, user guides, and FAQs. The function of user documentation is to help users learn how to utilize the application so that they can achieve their intended purpose and resolve problems if encountered.

The following user documentation will be made available to system users:

Online Help: Online help is an interactive, searchable guide to navigating an application. It can be provided locally on the platform or by third-party applications that describe how to use each feature.

Tutorials: Tutorials are a set of steps that lead users through common tasks or procedures in the application. They can show screenshots, videos, or interactive demos to allow users to learn how to use the application.

User Manual: A User manual is a formal document that completely documents an application's functions and features. User Guide (PDF and HTML) Document process of blood recipient and donor, registration, request for blood, etc. It generally contains comprehensive instructions, snapshots, and reference material.

FAQs: FAQs provide answers to the most common questions users have about the application. FAQs are generally located on the application's website, in the online help system, or in a separate document.

Administrator Guide: A step-by-step guide detailing how to install the system, how to deal with user profiles, authorize blood requests, and manage the system.

2.7 Assumptions and Dependencies

This section lays out key development assumptions and notable external dependencies the project relies on. These factors can influence system performance, user satisfaction, or general viability.

Assumptions:

The following assumptions were presumed in planning and designing the Blood Donation Portal:

1. **Error-Free Code:** The system's codebase is presumed to be free of critical runtime errors that would disrupt core functionality.
2. **User-Friendliness:** The system will be user-friendly and open to users who have varying degrees of technical savvy.
3. **System Capacity:** The system is assumed to support a high number of concurrent users with high-speed database access.
4. **Search Functionality:** The system will be equipped with robust search functionality between donors and blood requests.
5. **Access Requirements:** Users are expected to have access to an internet-equipped device with browser functionality and typical device literacy.
6. **Authentication:** The users should log in with valid credentials (username and password) to access their account and perform any customized activity.
7. **Inventory updates:** Inventory details need to be updated periodically by blood banks/hospitals according to their utilization. Donors will update regularly after each donation process is finished.

Dependencies:

The successful operation and utilization of the Blood Donation Portal are dependent on the following:

1. **Hardware & Software Requirements:** The system is dependent on some hardware infrastructure (storage, servers, modern browsers) and software systems (MySQL, etc.) to operate smoothly.
2. **Requirement Specifications:** Project development is carried out based on already set functional and non-functional requirements.
3. **Admin Expertise:** Proper functioning of the system depends on the administrator being adequately trained and familiar with the platform.
4. **Reporting System:** A reporting module for the storage, retrieval, and display of general summaries or insights should be present. This helps track donations, requests, and other activities.
5. **Centralized Database:** User-specific data(donors, patients, hospitals, blood banks) should be securely stored in a centralized database accessed by users according to their role.

Any changes from such assumptions or dependencies can lead to changes in scope, design, or deployment strategies.

3. External Interface Requirements

This chapter delineates the external interface requirements of the "Blood Donation Portal" web application that are different. The interfaces facilitate interaction among users, hardware, software, and communications systems effectively.

3.1 User Interfaces

The Blood Donation Portal will have five main categories of users, namely donors, patients, hospital representatives, blood bank representatives, and system administrators. Each of them will be displayed role-specific dashboards and access permissions.

Logical Characteristics:

The system will follow a responsive design to enable accessibility on desktops, tablets, and phones. Standard color schemes (grey, white, red) and iconography will be used to signify the medical and emergency intent of the application.

All screens will have standard UI elements consisting of

- Top navigation bar with profile, notification, help, and logout links
- Sidebar menu for dashboard-based navigation
- Confirmation and validation dialogs for actions like request submissions or profile updates
- Tooltip hints and inline error messages for form validation

GUI Standards:

- The interface design will follow Material Design principles and Bootstrap 5 principles for consistency and usability.
- Font and color accessibility will be WCAG 2.1 AA compliant.

Screen Layout Constraints:

- Minimum resolution support: 1366x768
- Responsive design support for viewports as small as 375x667 (mobile)

Sample Screens (see UI Design Specification):

- User Registration Page
- Emergency Blood Request Page
- Donor Profile Dashboard
- Admin Monitoring Pane

Shared Elements Across All Screens:

- Help button with an FAQs link
- Logout button
- Notification bell with real-time updates
- Universal footer with contact, terms, and privacy policy links

3.2 Hardware Interfaces

Operating System Compatibility:

The web application will be compatible with modern operating systems, including Windows 10+, macOS 11+, Ubuntu 20.04+, Android 11+, and iOS 13+.

Web Browsers Supported:

- Google Chrome (latest 3 versions)
- Mozilla Firefox (latest 3 versions)
- Safari (latest 2 versions)
- Microsoft Edge (latest 2 versions)

Server-Side Technologies:

- Backend Language: Node.js v18+ or Django (TBD)
- Framework: Express.js (for Node) or Django REST framework
- Database: PostgreSQL or MySQL 8.0
- Authentication: Firebase Authentication / JWT-based security

Third-Party Tools and Libraries:

- Map integration for location-based donor search (Google Maps API)
- Real-time communication (Firebase Cloud Messaging or Pusher)
- Chart rendering for analytics (Chart.js)

Shared Data Across Modules:

- User Profile Data (Name, Blood Group, Location, Availability Status)
- Blood Requests and Fulfillment Status
- Feedback and Ratings

Implementation Constraints:

- Shared data will be accessed through secure REST APIs using JSON format
- Global state managed via Redux (for React) or Vuex (for Vue)

3.3 Software Interfaces

Operating System Compatibility:

The web application will be compatible with modern operating systems, including Windows 10+, macOS 11+, Ubuntu 20.04+, Android 11+, and iOS 13+.

Web Browsers Supported:

- Google Chrome (latest 3 versions)
- Mozilla Firefox (latest 3 versions)

- Safari (latest 2 versions)
- Microsoft Edge (latest 2 versions)

Server-Side Technologies:

- Backend Language: Node.js v18+ or Django (TBD)
- Framework: Express.js (for Node) or Django REST framework
- Database: PostgreSQL or MySQL 8.0
- Authentication: Firebase Authentication / JWT-based security

Third-Party Tools and Libraries:

- Map integration for location-based donor search (Google Maps API)
- Real-time communication (Firebase Cloud Messaging or Pusher)
- Chart rendering for analytics (Chart.js)

Shared Data Across Modules:

- User Profile Data (Name, Blood Group, Location, Availability Status)
- Blood Requests and Fulfillment Status
- Feedback and Ratings

Implementation Constraints:

- Shared data will be accessed through secure REST APIs using JSON format
- Global state managed via Redux (for React) or Vuex (for Vue)

3.4   Communications Interfaces

Network Communication:

- All communication between client and server will use HTTPS with SSL/TLS encryption.
- RESTful API structure using JSON format for request/response cycles.

Email and Notification System:

- SMTP-based email service (e.g., SendGrid or Gmail SMTP) for sending registration confirmations, donation reminders, and password reset links
- Push notifications for emergency requests using Firebase Cloud Messaging

WebSockets (Optional for Real-Time Updates):

- WebSocket or Server-Sent Events for real-time updates on blood request statuses and admin notifications (implementation TBD)

Data Transfer and Security:

- Encryption: All data in transit will be protected using HTTPS

● Synchronization: AJAX-based polling and WebSocket-based updates will ensure data freshness

Communication Standards:

● REST API: For internal data communication
● HTTP/2: For faster multiplexed communication
● SMTP, TLS/SSL: For secure email services

The external interface requirements defined above ensure that the Blood Donation Portal remains accessible, secure, and efficient in interacting with all system entities and users.

4.   System Features

This section explains the most critical functional aspects of the blood donation management system. Each system feature is a major service or functionality the system provides to its users, e.g., donors, recipients, administrators, and hospital partners. For ease and usability, the features are explained individually, each explaining its purpose, interactions, and required system behaviors. Each requirement has a description of the priority level, stimulus/response pairs reflecting user interactions, user stories from the end-user perspective, and an exhaustive list of functional requirements. This presentation gives all stakeholders a clear notion of what the system must deliver to meet its desired goals and user expectations.

4.1   User Registration & Login

4.1.1 Description and Priority

This capability is the foundation for system login through enabling donors and recipients to register, securely authenticate, and manage their activities. Registration involves filling out personal and contact information, while login ensures that only verified users can access their personalized dashboard and features. The capability should be capable of supporting strong password policies, OTP/email verification, and encrypted credentials to prevent unauthorized access.
Priority: High

● Benefit: 9 (critical for personalized features and secure access)

● Penalty for Non-Compliance: 8 (without it, users cannot access core system features)

● Cost: 3 (moderate development cost due to common implementation patterns)

● Risk: 6 (risk of unauthorized access or credential leaks if not implemented securely)

● Benefit: 9

● Penalty: 8

● Cost: 3

4.1.2 Stimulus/Response Sequences

User Registration Trigger: A user accesses the registration page, enters personal, contact, and role-specific details, and submits.

○ **Response:** It validates input data, checks whether an account already exists, stores the information securely, and sends an email or SMS confirmation.

○ **Stimuli sequence:** User navigates to registration page and clicks on "Register" → System requests entry of form for email, phone number, and password → clicks "Submit" → System confirms that email and phone number are unique and correct → sends email or OTP to confirm user identity → User inputs verification code → Upon successful validation, system builds user account and displays welcome message → User navigates back to login page and enters credentials → System validates credentials and redirects to user dashboard.

Login Stimulus: User logs into the login page, enters username/email, and password.

○ **Response:** System checks credentials, grants access if correct, or displays an error message if wrong.

○ **Stimuli sequence:** In the event that the user is already registered and already has an account, then they can select "login" and enter credentials → the System checks the credentials and directs to the user dashboard.

Password Recovery Trigger: User selects the forgot password option and inputs their registered email or phone number.

○ **Response:** System verifies, sends a secure reset link or OTP, and requests new password creation.

○ **Stimuli sequence:** User clicks on forgot password → enters email or mobile number → reset link or OTP → enters new password → confirms password → logs in → takes to the user dashboard.

4.1.3 Functional Requirements

REQ-1: User Registration

A new user will have to register for the system, and while doing that, they have to select a role to get role-specific questions. As a new donor, I want to easily register for an account to securely participate in blood donation activities. As a new patient, I want to register for an account to get verified blood from either blood banks or directly from donors. As a new hospital, the management wants to register itself to participate in the blood donation process, which involves blood donation campaigns and blood transfusions to needy patients. As a new blood bank, the representatives want to expose themselves out there in front of the donors and patients as a reliable source for blood donation and reception.

Confirmation: Successful form submission stores user details securely, generates a unique User ID, and sends a confirmation notification. The form involves role-specific credentials like medical history for donors and license numbers for hospitals or blood banks. On the other hand, only a few credentials and verifications are required for patients, as they are potentially in need of an urgent blood transfusion.

Success: A successful account creation leads to the following:

- A "terms and conditions" page is shown.
- Users have to agree with the "terms and conditions".
- Their account confirmation is sent after 48 hours of admin review.

Failure: The failure to register leads to:
- Displays inline error messages (e.g., *"Email is required"*).
- Sends email: *"Your registration was declined. Contact support@blooddonationportal.com for details."*

REQ-2: User Login

As a registered user, I want to log into my account quickly and securely so I can access personalized system features. This feature needs to be very quick for patients. In case of hospitals and blood banks, their administrators will log in on the institution's behalf.
Confirmation: The system authenticates user credentials within two seconds, displaying the user dashboard upon success.
Success: A successful log-in leads to the following:
- User enters valid credentials (email/username + password).
- The system authenticates within 2 seconds and redirects to the personalized dashboard.
- Session token is generated and stored securely.

Failure: Failure to log in leads to the following:
- Invalid/Corrupted credentials entered → System displays: *"Invalid email/password. Try again."*
- Network/server error → System shows: *"Login temporarily unavailable. Try later."*

REQ-3: Password Complexity Enforcement

As a security-conscious user, I expect my password to meet strong complexity criteria to protect my data. In case of a blood bank and hospital, this feature is extremely important as much sensitive data is stored here.
Confirmation: Passwords must include at least eight characters, one uppercase letter, one lowercase letter, one digit, and one special character.
Success: A success leads to the following:
- User creates/updates password meeting criteria (8+ chars, uppercase, lowercase, digit, special char).
- System confirms: *"Password updated successfully."*

Failure: A failure leads to the following:
- Weak password entered → System highlights missing criteria (e.g., *"Add a special character (!@#)."*).
- Submission blocked until requirements are met.

REQ-4: Duplicate Account Prevention

As a system administrator, I want to prevent users from creating duplicate accounts to maintain system integrity. As the system deals with a very crucial task of blood donation, no duplicates accounts should be allowed.
Confirmation: System checks email or mobile numbers against the database, displaying an error for duplicates.
Success: A success leads to **the following:**
- New user registers with unique email/mobile → Account created.

- System confirms: *"Registration complete! Welcome, [user]."*

Failure: Failure leads to the following:

- Duplicate email/mobile entered → System rejects: *"An account already exists with this email/mobile."*

REQ-5: Session Management

As a user, I want my session to be managed securely and efficiently to prevent unauthorized access. This is very important as inactivity might lead to access to some third party to the device of the users.

Confirmation: User sessions automatically expire after 30 minutes of inactivity.

Success: A success leads to the following:

- Active session → User operates uninterrupted.
- After 30 minutes of inactivity, → System logs user out, displays: *"Session expired. Log in again."*

Failure: A failure leads to the following:

- Session token corrupted → System forces logout, prompts reauthentication.

REQ-6: Secure Password Recovery

As a user, I want to recover my account securely if I forget my password, minimizing disruption to my activities. Forgetting password is a normal phenomenon, but the mechanism for changing passwords need to ensure that no discrepancies occur.

Confirmation: System validates identity through email or SMS OTP, allows password reset, and confirms via notification.

Success: A success leads to the following:

- User requests recovery → Receives OTP via email/SMS.
- OTP validated → Password reset page opens.
- New password set → Confirmation email/SMS sent.

Failure: A failure leads to the following:

- Invalid OTP → System blocks reset: *"Incorrect OTP. Retry or request new OTP."*

REQ-7: Multifactor Authentication

As a user, I want an optional multifactor authentication (MFA) method to increase the security of my account. This feature is optional, as this might seem a bit too much for occasional users like patients, who do not need this level of security for their accounts. But this might be crucial for donor/hospital/blood bank accounts, as they handle sensitive data and need this authentication system.

Confirmation: When activated, MFA requires a secondary verification step via email or SMS upon login.

Success: A success leads to the following:

- MFA enabled → Secondary OTP sent post-login.
- OTP verified → Access granted.

Failure: A failure leads to the following:

- OTP expired/wrong → System denies access: *"Verification failed. Retry."*

REQ-8: Account Lockout Mechanism

As a system administrator, I want accounts temporarily locked after consecutive failed login attempts to prevent unauthorized access. This will help in case of malicious attempts to log in to an account, especially for business accounts of hospitals and blood banks as they provide critical and sensitive data which should not be manipulated.
Confirmation: The Account is locked after 5 failed attempts for 15 minutes, with user notification sent via email/SMS.
Success: A success leads to the following:
- 5 failed attempts → Account locked for 15 mins.
- Email/SMS sent: *"Account locked temporarily. Try again after [time]."*

Failure: A failure leads to the following:
- User retries before lockout expires → System rejects: *"Account locked. Wait 15 mins."*

REQ-9: User Consent & Data Privacy Agreement

As a new user, I must explicitly agree to the terms and privacy policy during registration. In case of not agreeing to this, the system will not let the user to register.
Confirmation: Registration cannot proceed without a checked consent box; agreement is recorded in user logs.
Success: A success leads to the following:
- User checks consent box → Proceeds to registration.
- Agreement logged in system.

Failure: A failure leads to the following:
- Unchecked box → System halts: *"You must accept terms to proceed."*

4.2   Donor Profile Management

4.2.1 Description and Priority

This feature enables donors to view and update their profiles, including name, contact details, blood group, availability status, and the date of their last donation. Keeping this information current is vital for accurate matching with recipients. It also helps to enforce safe donation practices by preventing donations before the required recovery period has passed.
Priority: High
- Benefit: 8 (essential for accurate donor-recipient matching)

- Penalty: 7 (incomplete or outdated data could delay or invalidate donations)

- Cost: 3 (low-to-moderate complexity)

- Risk: 5 (mismanagement of medical data can lead to safety issues)

4.2.2 Stimulus/Response Sequences

Profile Update Stimulus: Donor accesses the profile management page and updates personal information.

○   **Response:** The system validates and securely stores updated information, confirming the update via notification.

○   **Stimuli Sequence:** Donor logs in and clicks on "Profile" from the navigation menu → System retrieves and displays the current profile information. Donor edits fields such as phone number, blood type, or last donation date → System checks for valid input (e.g., correct format, donation interval). Donor clicks "Save" → System validates the data, updates the database, and shows a confirmation message.

Donation History Viewing Stimulus: Donor requests access to their donation history.

○   **Response:** The system retrieves and displays detailed past donation records.

○   **Stimuli Sequence:** Donor logs into their account → clicks on "Profile" → clicks on the "History" → Sees specific history information

Preferences Setting Stimulus: Donor modifies preferences for notifications and donation availability.

○   **Response:** The system updates preferences and sends a confirmation message

○   **Stimuli Sequence:** Donor logs into their account → clicks on "Profile" → clicks on "Edit Options" → System goes to profile edit mode → Donor edits information as required → Admin verifies the information within 48 hours.

4.2.3 Functional Requirements

REQ-1: Profile Update

As a donor, I want to easily update my details so my information remains accurate. I might have many changes in my information throughout the years(such as relocation), so this feature is highly helpful for me to keep up with my donation activities.

Confirmation: Updated details are validated, stored securely, and immediately reflected in the donor profile.

Success: A success leads to the following:
- Donor edits profile details (e.g., address, contact info).
- System validates and saves changes instantly.
- Confirmation message: *"Profile updated successfully!"*

Failure: A failure leads to the following:
- Invalid data format (e.g., wrong phone number) → System highlights errors: *"Please enter a valid 10-digit mobile number."*
- Server error → Displays: *"Update failed. Try again later."*

REQ-2: View Donation History

As a donor, I want to view my past donations clearly so I can track my contributions and schedule future donations effectively. This donation history will also help me to be able to track my next donation schedule, as a certain time period needs to pass between each donation.

Confirmation: Donation history is displayed chronologically with dates, donation locations, and quantities.

Success: A success leads to the following:
- Donor accesses history page → Sees chronological list with dates, locations, and donation amounts.
- Filter/sort options work flawlessly.

Failure: A failure leads to the following:
- No history found → System shows: *"No donations recorded yet."*
- Network issue → Displays: *"Unable to load history. Check connection."*

REQ-3: Preference Management

As a donor, I want to manage my notification and availability preferences easily to stay informed without being overwhelmed. As a donor, I might have many phases in my life, like a health issue or personal problems, such as examinations and family. At that times I should be able to manage my notifications to stay away from being drawn into the donation tasks.
Confirmation: Preferences are saved instantly, adjusting system notifications accordingly.
Success: A success leads to the following:
- Donor toggles notification/availability settings → Changes apply immediately.
- Confirmation: *"Preferences saved!"*

Failure: A failure leads to the following:
- Invalid selection (e.g., blank field) → System prompts: *"Please select at least one notification option."*

REQ-4: Privacy Settings

As a donor, I want control over the visibility of my profile information to maintain privacy. This should depend on the will of the donor, whether they want to display their information or not. There will be layers of privacy to decide for the donor which information they want to share and which ones they want to hide.
Confirmation: Donors can toggle visibility settings; the system enforces these settings strictly.
Success: A success leads to the following:
- Donor sets profile to "Private" → Details hidden from public view.
- Confirmation: *"Privacy settings updated."*

Failure: A failure leads to the following:
- Conflicting settings → System warns: *"Cannot hide required fields (e.g., name)."*

REQ-5: Profile Verification

As an administrator, I want donor profiles periodically verified for accuracy to ensure data reliability. This verification is very crucial as the donor will donate blood to the patients. This feature will able to obstruct many discrepancies like fraud or faulty blood donations.
Confirmation: The system flags profiles needing verification, prompting administrators for validation.
Success: A success leads to the following:
- System flags unverified profiles → Admin verifies/rejects with comments.
- Donor receives email: *"Profile verified successfully!"*

Failure: A failure leads to the following:
- Missing documents → Admin marks: *"Rejected: Upload ID proof."*

REQ-6: Profile Picture Upload

As a donor, I want to upload a profile picture to personalize my account. This picture will help to further personalize a donor's account.

Confirmation: Images uploaded are validated for format and size, and displayed correctly after approval.
Success: A success leads to the following:
- Donor uploads JPEG/PNG (<5MB) → Image displays after cropping.
- Confirmation: *"Photo uploaded!"*

Failure: A failure leads to the following:
- Invalid file (e.g., PDF) → Error: *"Only JPG/PNG allowed."*
- Oversized image → Prompt: *"Max size: 5MB."*

REQ-7: Emergency Contact Management

User Story: As a donor, I want to add emergency contact information so someone can be contacted quickly in case of an emergency. This will highly help in case of an urgent blood transfusion need.
Confirmation: Emergency contacts are recorded, and the system highlights missing or incomplete data.
Success: A success leads to the following:
- Donor adds/edits contacts → System saves and highlights completeness (e.g., *"100% complete"*).

Failure: A failure leads to the following:
- Missing phone number → Flags: *"Emergency phone required."*

REQ-8: Feedback Integration

As a donor, I want my feedback on donations recorded in my profile to improve future experiences. This feedbacks will contribute to the donor as a person so that patients can decide to reach out after seeing the feedbacks.
Confirmation: Feedback entries are linked to specific donation events, accessible to administrators for review.
Success: A success leads to the following:
- Donor submits feedback → Linked to donation ID in database.
- Admin sees feedback in dashboard.

Failure: A failure leads to the following:
- Blank feedback → System blocks submission: *"Please describe your experience."*

4.3   Blood Donation Request and Login

4.3.1 Description and Priority

Registered recipients (healthcare providers or patients) can place and submit detailed blood donation requests through this feature. Every request includes essential information such as blood group, degree of urgency, location, and special instructions, if any. The system processes the inputs to decide and inform prospective donors based on compatibility, proximity, and past donation history.

Efficient request processing is essential for emergency and non-emergency cases alike, so this functionality is at the heart of the platform's mission of saving lives.

Priority: High

Benefit: 9 (Mission-critical functionality to satisfy the platform's purpose)

Penalty: 9 (Not requesting blood negates the system's intent)

Cost: 4 (Moderate, as it deals with matching logic and notification mechanisms)

Risk: 6 (Delays or mismatches can have catastrophic outcomes)

4.3.2 Stimulus/Response Sequences

Request Creation:

Stimulus: The Recipient signs in and submits a blood request.

Response: The system validates the input, stores the request, and notifies nearby matching donors.

Request Update:

Stimulus: A recipient updates a requested update.

Response: The system stores the update and notifies previously notified donors about the change.

Request Cancellation:

Stimulus: A recipient cancels a request.

Response: The system stores the request as cancelled and sends cancellation notifications to matched donors.

4.3.3 User Stories

● As a recipient, we would like to make a blood request with the given blood type, urgency, and location, so that the system can notify nearby compatible donors instantly.
● As a donor, we would like to receive real-time alerts when a matching blood request is made nearby, so that we can respond instantly in emergencies.
● As a recipient, we would like to cancel or update my request if there are changes, and make sure donors are notified immediately.
● As users, we would like our blood request to be private or public according to our desire to keep it confidential.
● As an administrator, we would like requests to be automatically matched with appropriate donors so that they can be filled quickly, and there is less manual handling.
● As an admin, we need to auto-archive completed or old requests so that the database remains efficient and clean.

4.3.4 Functional Requirements

● The system must provide a secure recipient login for accessing the blood request functionality.

● The system must provide a formatted form for collecting blood type, level of urgency, quantity, location, and optional comments.

● The system must validate all required fields and prevent incomplete submissions.

● The system should allow recipients to set the visibility (public/private) of their request.

● The system should automatically match requests with available, compatible donors near them based on blood type and availability.

● The system should send real-time notifications (email, SMS, app alert) to matching donors.

● The system should update the status of the request dynamically (pending, matched, satisfied, cancelled).

● The system must allow recipients to cancel or edit their requests at any time.

● The system must maintain an audit trail of cancellations and edits for transparency purposes.

● The system must archive expired or completed requests automatically for improved database performance.

● The system must highlight high-urgency requests visually to improve donor prioritization.

4.4   Donation History and Tracking

4.4.1 Description and Priority

This feature provides users with a record of their past activities, including donation history and blood request logs. It allows users to track their contributions, ensure compliance with safe donation intervals, and review fulfilled or canceled requests. The feature enhances transparency and accountability.
Priority: Medium
● Benefit: 7 (improves user engagement and tracking)

● Penalty: 5 (absence limits historical reference but doesn't halt operations)

● Cost: 3 (moderate, mostly involves database queries and display logic)

● Risk: 4 (risk of inaccurate data if not updated properly)

4.4.2 Stimulus/Response Sequences

- **Donation History View Stimulus:** A user accesses their account dashboard to view past donation records.
- ○ **Response:** The system retrieves and presents the donation history with dates, locations, and statuses.
- **Donation Tracking Stimulus:** A user seeks real-time updates on the status of their current blood donation.
- ○ **Response:** The system provides immediate status updates on collection, testing, processing, and distribution.

4.4.3 Functional Requirements

REQ-1: Comprehensive Donation Records: As a donor, I would like to see a complete history of my previous donations or when I have donated to maintain a record of my donations and plan future donations.
Confirmation: The system displays appropriate and complete records, for instance, dates, types, locations, and statuses.
REQ-2: Real-Time Donation Tracking: As a donor, I want to track the real-time status of my blood donation from collection to distribution for transparency.
Confirmation: Real-time tracking statuses are updated continuously and displayed clearly.
REQ-3: Historical Data Export: As a donor or medical professional, I would want to be able to export my donation history for medical reasons or personal records.
Confirmation: Users can export their data in common formats such as PDF or Excel files.
REQ-4: Donation Status Notifications: As a donor, I want automated notifications regarding the status of my donations to remain informed without manual tracking.
Confirmation: Notifications are sent automatically at each stage of the donation lifecycle.
REQ-5: Analytical Insights: As an administrator, I want analytical insights into donation patterns to inform better decision-making and donor engagement strategies.
Confirmation: The system generates analytical reports, which are accessible to administrators.
REQ-6: User Feedback Integration: As a donor, I want my feedback, which I have submitted, to be recorded along with my donation history for comprehensive personal records.
Confirmation: Feedback linked explicitly to corresponding donations.
REQ-7: Privacy Settings: As a donor, I want control over who can view my donation history to maintain my privacy.
Confirmation: User-selected visibility settings are strictly followed by the system.
REQ-8: Donation Milestones Recognition: As a donor, I want to be recognized for reaching donation milestones to feel appreciated and motivated.
Confirmation: Milestone achievements trigger automated appreciative notifications or certificates.
REQ-9: Data Integrity Checks: As an administrator, I want regular integrity checks on historical donation data to ensure accuracy and reliability.
Confirmation: The system automatically performs scheduled data integrity validations and alerts administrators to irregularities.

4.5   Emergency Blood Requests

4.5.1 Description and Priority

With this feature, users can mark blood donation requests as emergencies. When flagged, the system sends high-priority alerts to compatible nearby donors to ensure rapid response. This function is critical in life-threatening scenarios and must bypass standard scheduling constraints.
Priority: High

● Benefit: 9 (lifesaving during emergencies)

● Penalty: 9 (delayed response may cause fatal outcomes)

● Cost: 4 (additional logic and UI for alerting mechanisms)

● Risk: 7 (incorrect handling could desensitize users to alerts)

4.5.2 Stimulus/Response Sequences

● **Emergency Request Stimulus:** A patient, hospital, or blood bank initiates an emergency blood request.
○ **Response: The system** immediately receives the request, broadcasts alerts to nearby donors, and provides real-time tracking.
● **Emergency Response Stimulus:** A donor responds to an emergency request notification.
○ **Response: The system** confirms donor availability and updates the requester and system administrators.

● Recipient fills out the blood request form and marks the "Emergency" checkbox → The system flags the request as high priority.

● Upon submission, the system scans for available nearby donors → The system sends real-time emergency alerts via push notifications, SMS, or email.

● Donors receive instant emergency alerts → They respond with availability, which the system relays to the requester.

● The system updates the request status accordingly and prioritizes it in admin and donor views.

4.5.3 Functional Requirements

REQ-1: Immediate Request Submission: As a medical worker, I am required to initiate emergency blood requests in a timely manner to provoke immediate donor response.
Confirmation: Requests are validated and transmitted immediately upon submission.
REQ-2: Priority Notices: As a donor, priority notices for emergency blood requests allow me to respond quickly to critical situations.

Confirmation: Emergency notifications override default notice settings and appear in a bold display.

REQ-3: Location-Based Alerts: I, being a system administrator, require alerts to be sent only to donors near the emergency so that I am readily available.

Confirmation: Alerts use correct geolocation to send alerts effectively.

REQ-4: Real-Time Response Monitoring: I, being a requester, require monitoring of real-time responses from donors to monitor availability and arrange logistics promptly.

Confirmation: Real-time data is available immediately on the emergency request dashboard.

REQ-5: Escalation Procedure: As an admin, I would prefer automatic escalation procedures when the response from donors on the first level is insufficient to handle critical shortages in a proper manner.

Confirmation: Escalation procedures are automatically triggered in 15 minutes in the event of insufficient responses.

REQ-6: Facilitating Direct Communication: As a requester, I would prefer direct communication with responding donors to hasten urgent coordination.

Confirmation: Secure and direct lines of communication are established instantly in response from donors.

REQ-7: Donor Confirmation and Commitment: As a donor, I want immediate and unambiguous confirmation of my commitment to donating in emergencies to be prepared effectively.

Confirmation: The System immediately confirms donor commitments via SMS or email.

REQ-8: Post-Emergency Reporting: As an administrator, I want full reports of emergency responses for continuous improvement of response strategies.

Confirmation: Full reports are automatically generated and saved upon emergency resolution.

REQ-9: Historical Emergency Data: Being a healthcare provider, I would prefer to have historical emergency request data to help forecast future requirements and plan.

Confirmation: Historical emergency data is neatly organized and presented readily for scrutiny.

4.6   Blood Compatibility Checker

4.6.1 Description and Priority

This feature helps users determine if a donor and a recipient are compatible based on their blood types. It provides a visual tool or table that maps acceptable pairings and prevents unsafe matches. This is an educational and preventive tool aimed at non-medical users.

Priority: Medium

- Benefit: 6 (helps ensure safe transfusions)

- Penalty: 6 (incorrect match can cause medical issues)

- Cost: 2 (minimal implementation complexity)

- Risk: 5 (incorrect logic could mislead users)

4.6.2 Stimulus/Response Sequences

Compatibility Check Stimulus: User inputs donor and recipient blood types.
- **Response:** The system immediately checks and displays compatibility results.

Compatibility Query Stimulus: User queries compatibility details for educational or informational purposes.
○ **Response:** The System provides comprehensive information about potential compatibility and incompatibility scenarios.

4.6.3 Functional Requirements

REQ-1: Quick Compatibility Assessment: As a medical practitioner, I require a quick compatibility checking tool to instantly verify donor-recipient compatibility.
Confirmation: The system instantly displays correct compatibility information on input of blood groups.
REQ-2: Detailed Compatibility Information: As a medical practitioner, I require detailed explanations of blood compatibility in order to properly inform patients and donors.
Confirmation: The system provides detailed explanations, including potential risks associated with incompatibilities.
REQ-3: Compatibility Chart Access: As a user, I must be able to access and view detailed blood compatibility charts for convenient reference.
Confirmation: Detailed charts are easily accessed and graphically simple to understand.
REQ-4: Automated Compatibility Alerts: As a requester, I would prefer the system to automatically warn me about compatibility issues when ordering blood requests so that errors can be avoided.
Confirmation: Alerts are automatically generated and displayed whenever there are potential compatibility issues.
REQ-5: Educational Resources: As a donor or patient, I would prefer easy access to materials explaining blood compatibility so that I can more fully appreciate its significance.
Confirmation: The System provides concise, easy-to-read educational materials incorporated into the compatibility checker.
REQ-6: Historical Compatibility Queries: As an administrator, I would prefer to track historical compatibility queries in order to trend frequent misunderstandings or educational needs.
Confirmation: System logs queries and produces reports at intervals for review.
REQ-7: Mobile Compatibility Checks: As a user, I would prefer compatibility checking easily accessible on mobile devices for convenience.
Confirmation: The compatibility checker is fully responsive and works flawlessly on all mobile devices.
REQ-8: Integration with Blood Request: As a medical practitioner, I would prefer automatic compatibility checking integrated into blood request workflows to make operations simpler.
Confirmation: Compatibility checks automatically execute during request submissions.
REQ-9: Error Handling and Correction: As a user, I would like step-by-step guidelines for the correction of erroneous blood type entries for the sake of accuracy and reliability.
Confirmation: The System immediately shows errors and prompts the user to correct them.

4.7   Donor Search with Filters

4.7.1 Description and Priority

This feature lets recipients find eligible donors by applying filters such as blood group, geographic proximity, availability, and donation frequency. It is essential for ensuring fast and relevant matches between patients and potential donors.
Priority: High

● Benefit: 8 (streamlines donor discovery and request fulfillment)

● Penalty: 7 (limits direct outreach if absent)

● Cost: 4 (requires backend querying and location handling)

● Risk: 4 (potential data overload or irrelevant results)

4.7.2 Stimulus/Response Sequences

Donor Search Stimulus: A user enters specific search criteria such as blood type, location, or availability.
○ **Response:** The system filters and displays a list of matching donors in real time.
Search Result Interaction Stimulus: A user selects a donor from the result list.
○ **Response:** The system displays a detailed donor profile with contact options and donation eligibility.

4.7.3 Functional Requirements

REQ-1: Filter-Based Donor Search: As a patient or medical professional, I want to search for donors using blood group, location, and availability filters so that I can find suitable matches efficiently.
Confirmation: Search results return only those matching all selected filters.
REQ-2: Radius-Based Location Filtering: As a requester, I want to define a radius around a location to find nearby donors to reduce response time.
Confirmation: Results dynamically update based on specified radius from the selected location.
REQ-3: Last Donation Date Filter: As a user, I want to filter out donors who recently donated to ensure donor health compliance.
Confirmation: Donors with recent donations (less than 3 months) are excluded automatically.
REQ-4: Search Result Visualization: As a user, I want results to be shown on both list and map views for better understanding and navigation.
Confirmation: Search results include geographic pins and detailed list entries.
REQ-5: Live Availability Filter: As a requester, I want to see only currently available donors for urgent needs.
Confirmation: Donors who mark themselves as unavailable are automatically excluded from results.
REQ-6: Direct Contact Access: As a patient or hospital, I want to contact matching donors instantly through the platform to accelerate coordination.
Confirmation: Contact buttons (call/email) are embedded within each donor profile.
REQ-7: Anonymous Matching Option: As a donor, I want to be searchable without exposing my identity until I choose to respond to a request.
Confirmation: Anonymous donor results show limited info with a system-handled request relay.

REQ-8: Smart Search Suggestions: As a user, I want intelligent search suggestions based on past search behavior to make my queries faster.
Confirmation: System suggests frequently searched filters and common donor locations.
REQ-9: Admin-Controlled Search Filters: As an administrator, I want to configure and expand search filters (e.g., vaccination status, weight range) for future needs.
Confirmation: Admin panel includes editable filter configurations with impact preview.

4.8   Blood Bank & Hospital Integration

4.8.1 Description and Priority

This feature integrates the system with licensed hospitals and blood banks to fetch and display real-time data on available blood units. It enforces compliance with the Safe Blood Transfusion Act, 2002, and the associated 2008 regulations by restricting access to verified institutions and requiring proof of disease testing (HIV, Hepatitis B/C, etc.).

Priority: Medium
- Benefit: 7 (improves access to reliable institutional inventories)

- Penalty: 5 (lack of integration reduces resource visibility during emergencies)

- Cost: 6 (requires secure APIs and legal compliance checks)

- Risk: 6 (non-compliance or outdated data can jeopardize safety)

4.8.2 Stimulus/Response Sequences

Admin Login Stimulus: An authorized admin accesses the admin panel via secure credentials.
○   **Response:** System authenticates access and loads the dashboard with relevant controls and analytics.
User Management Stimulus: Admin selects a user management option (e.g., approve, block, reset password).
○   **Response:** The System executes the command and logs the action with a timestamp.

4.8.3 Functional Requirements

REQ-1: System Integration Request: I, as a hospital or blood bank administrator, would like to request system integration to automate the availability of blood and operations.
Confirmation: System processes requests and notifies applicants of approval or required modifications.
REQ-2: Secure Inventory API: As a technical user, I would like an API securely to transmit our blood inventory information in real-time.
Confirmation: The System processes secure API requests and updates platform inventory accurately.
REQ-3: Access Management: From the perspective of a hospital administrator, I prefer to manage users and access levels in our integrated account for securing operations.
Confirmation: System employs role-based access control across the hospital/blood bank interface.

REQ-4: Inventory Auto-Sync: As an integrated organization, I prefer our inventory to auto-sync with the platform to avoid manual updates.
Confirmation: Inventory changes are automatically updated within 2 minutes of local system changes.
REQ-5: Emergency Inventory Flag: As a hospital, I want to mark blood units as emergency reserves so that visibility is restricted to only critical cases.
Confirmation: Flagged emergency units are visible only to authenticated users or situations.
REQ-6: Inter-Hospital Sharing Option: As a hospital administrator, I want to be able to see and order blood from other hospitals within the network when critically low.
Confirmation: Function exposed by role authorizations, logs transfer between hospitals.
REQ-7: Weekly Reporting Tool: As a blood bank, I would like automated weekly usage and inventory reports sent to me to schedule operations.
Confirmation: They are automatically emailed every Sunday morning at 6 AM.
REQ-8: Notification System: As a blood bank, I would like alerts when our level goes below a certain level so we can order in advance.
Confirmation: It provides alerts when thresholds could be set as required.
REQ-9: Data Compliance Logging: As a system administrator, I want to audit and log all data exchanges for security and compliance assurance.
Confirmation: The logs are time-stamped and encrypted and are retrievable via the admin console.

4.9 Admin Panel

4.9.1 Description and Priority

The admin interface enables centralized administration over donor records, emergency blood requests, user accounts, hospital compliance, and policy enforcement. Guaranteeing that the system is exclusively employed by authorized companies depends on enforcing legal criteria under the Safe Blood Transfusion Act of 2002.

Priority: High
- Benefit: 9 (ensures legal compliance, operational stability, and security)

- Penalty: 7 (lack of oversight may result in illegal or unsafe practices)

- Cost: 5 (requires complex access controls and auditing tools)

- Risk: 6 (admin error or malicious use may disrupt the system or violate regulations)

4.9.2 Stimulus/Response Sequences

- Admin logs in → System verifies credentials and loads the dashboard.

- Admin selects modules like "Hospital Oversight" or "Request Monitoring." → System displays logs, stats, or flagged entries.

- Admin reviews suspicious requests or policy breaches → System allows action (e.g., suspend, audit, flag, or freeze account).

- Admin modifies system rules → System logs change with admin ID and timestamp.

4.9.3 Functional Requirements

REQ-1: Admin Authentication: As an admin, I want to log in securely to the admin panel so that I can manage platform operations safely.
Confirmation: Admin login is secured with two-factor authentication and logs all access.
REQ-2: User Account Management: As an admin, I want to view, approve, suspend, or delete user accounts to maintain platform integrity.
Confirmation: System reflects all updates instantly and logs every user action taken.
REQ-3: Blood Request Oversight: As an admin, I want to view and manage all blood donation requests to ensure proper processing and resolve issues.
Confirmation: Admins can update request statuses and override errors.
REQ-4: Donor and Hospital Verification: As an admin, I want to verify the identities of donors and institutions before granting access to secure features.
Confirmation: Verification is completed manually or automatically and logged accordingly.
REQ-5: Dashboard Analytics: As an admin, I want to view platform analytics such as donor activity, request volumes, and regional trends to monitor performance.
Confirmation: Analytics loads automatically on the dashboard and updates daily.
REQ-6: System Settings Configuration: As an admin, I want to customize operational rules (e.g., donation eligibility interval) for adaptability.
Confirmation: Settings changes apply immediately and are recorded in system logs.
REQ-7: Notification and Alert Control: As an admin, I want to manage how and when system notifications are triggered to improve user experience.
Confirmation: Notification rules are configurable through a dedicated interface.
REQ-8: Feedback Moderation: As an admin, I want to view and moderate all user-submitted feedback and ratings to ensure constructive communication.
Confirmation: Admins can flag, approve, or delete feedback entries.
REQ-9: Audit Logging: As an admin, I want a full audit trail of all admin activities to maintain accountability and detect misuse.
Confirmation: Every admin action is time-stamped and stored securely in the audit logs.

4.10   Feedback and Rating

4.10.1 Description and Priority

Feedback and Rating is a module that allows users to provide post-interaction ratings both as star ratings and optional voluntary text comments. Its purpose is to enhance transparency, build trust with users, and support ongoing quality improvement in services provided by donors, hospitals, and other members of the platform. Despite no direct impact on core functionality, this module directly impacts platform credibility and long-term user experience significantly.

Priority: Low

Benefit: 5 (Encourages user activity and responsibility)

Penalty: 3 (Omission has no impact on primary operations)

Cost: 2 (Low cost of implementation using common UI components)

Risk: 3 (Risk of abuse if poorly moderated)

4.10.2 Stimulus/Response Sequences

Stimulus 1: User finishes a donation or is helped by a donor or hospital.
Response: The system detects completion and requests the user to rate and comment using a rating and comment form.

Stimulus 2: User provides a rating and/or comment.
Response: The system stores the feedback, updates the profile of the respective entity (e.g., donor or hospital), and, if selected, anonymizes the entry.

Stimulus 3: An admin opens the feedback dashboard.
Response: All records of feedback are shown with filtering, moderation, and analysis controls.

Stimulus 4: Other users read feedback.
Response: Tagged feedback and ratings aggregations appear on donor and hospital profiles to facilitate informed decision-making.

4.10.3 User Story

As a user, we want to be able to leave feedback and rate an individual, hospital, or service after providing support or making a donation, so that we can assist in making the system more transparent, reliable and offer improved services to future users.

4.10.4 Functional Requirements

● The system must support users to leave a star rating (1–5 stars) after the completion of a blood donation or support, with the possibility to leave a written comment.

● The system will calculate and display the average rating and review count for each donor and hospital profile.

● The system will prompt the feedback form at suitable points (e.g., on donation confirmation or request completion).

● The system will have a provision for users to leave anonymous feedback such that no personal identifiers are stored when anonymity is selected.

● Administrators must be capable of moderating posted feedback by flagging, hiding, or deleting offending posts.

● The system will generate periodic analytics reports to allow administrators to monitor user satisfaction and improvement areas.

● The system must support nested responses so that hospitals or donors can reply to feedback with time-stamped messages.

● The system should provide feedback tags (e.g., donor behavior, facility cleanliness, timing) to properly categorize and analyze feedback manually or via auto-categorization algorithms.

● The system should be able to accept multilingual submissions of feedback and provide translated summaries for administrative review where necessary.

5   Other Nonfunctional Requirements

5.1   Performance Requirements

The Blood Donation Portal should handle a maximum of 500 concurrent users without appreciable performance reduction. Page load time should be no greater than 3 seconds in typical network conditions. The system should process a donor search query in under 2 seconds. Real-time notifications (donation requests and alerts) must be delivered with a maximum lag of 5 seconds. The server uptime should satisfy a 99.9% availability standard per month, to ensure access in case of emergencies.

5.2   Safety Requirements

The system must ensure:

1.   Adequate donor-recipient matching to prevent inappropriate donations.

2.   Donor screening is required before allowing confirmation of a blood request.

3.   Backup and Recovery Plan: The database must automatically get backed up daily to prevent any data loss in the event of a system crash.

4.   In the event of system failure, graceful degradation (error messages and temporary offline access to emergency contact lists) must be present to maintain safety-critical information.

5.3   Security Requirements

The system will have to:

1.   Implement authentication processes for all users via email/OTP verification.

2.   Role-based access control (RBAC) to restrict access to sensitive medical and personal data (e.g., admins see full profiles; donors/recipients cannot see each other's full information without authorization).

3.   Data encryption (SSL/TLS) in transit and database encryption of stored sensitive data.

4.   Compliance with applicable data protection laws (e.g., GDPR principles or applicable local data privacy laws).

5.   There ought to be periodic security audits at least once every 6 months.

5.4   Software Quality Attributes

The Blood Donation Portal must possess the following quality attributes:

Availability: At least 99.9% uptime with support for auto failover.

Usability: The portal must be simple for users with the least digital literacy.

Reliability: System recovery in case of failure must not result in data loss of transactions.

Maintainability: The code must be modular with proper documentation to be easily patched and fixed for bugs.

Portability: The portal must be accessible from major browsers (Chrome, Firefox, Edge) and must be supported for mobile view.

Scalability: The Architecture should accommodate scaling to 5000 users without compromising on performance.

Interoperability: Should be able to integrate with SMS APIs, Google Maps API (for location tracking), and a healthcare database in the future.

5.5   Business Rules

●   Donors must have minimum eligibility requirements (age >18, weight >50kg, healthy status) verified at registration.

●   One blood donation request per user in a rolling 90-day window should be allowed (to respect safe donation practices).

●   Emergency requests should prioritize matching by blood group and location proximity within a 10 km radius.

●   Admins can accept or reject hospital registrations and monitor suspicious activity.

●   Donor profiles need to be anonymized until the donors consent to sharing information for a particular request.

●   Blood donation camps and events can only be generated by validated organizations (admins/hospitals).

6   Other Requirements

Database Requirements
The Blood Donation Portal will depend on a relational database, such as MySQL, to securely store user profiles, donation records, requests, and event details. The database design should follow best practices, including normalization, to minimize redundancy and ensure smooth performance.

Internationalization Requirements
Initially, the system will support English, keeping in mind the diversity of users in Bangladesh. It should also be flexible enough to add more languages in the future without major changes to the system.

Legal Requirements
The portal must obey relevant laws and regulations, including the Digital Security Act, 2018 of Bangladesh. Consent must be properly taken before collecting or processing any personal or medical data.

Reuse Objectives

Where possible, the project should leverage existing open-source libraries for functionalities like email verification, geolocation, SMS notifications, and data encryption. This will help in speeding up development while maintaining quality.

Backup and Recovery Requirements
The system must perform automatic database backups daily, storing them securely in multiple locations. A manual backup check should also be conducted once a week to ensure backup integrity.

Audit and Logging Requirements
All important activities—like user logins, profile updates, blood requests, and administrative actions—should be logged. These audit logs must be tamper-proof and kept for at least six months to support troubleshooting and security investigations if needed.

Appendix A: Glossary

1.   Blood Donation Portal - A web application that enables users to donate, collect, and manage blood donations.
2.   User - An individual or organization that uses the application to donate or collect blood.
3.   Donor Management - Add the new donor, update donor details, and all details about the donor..
4.   Search Donor - Search the donor location and Blood group-wise.
5.   Stock Blood - Check the Availability of blood Group.
6.   Delete Donor - Delete donor from the Database.
7.   SRS - Software Requirements Specification. A document that specifies the functional, performance, and other requirements for a software application.

Appendix B: Analysis Models

The figure attached below vaguely represents the top-level data flow diagram. This diagram shows the major groups related to the system and how they are related to each other, including a few features that they are supposed to handle in the system.
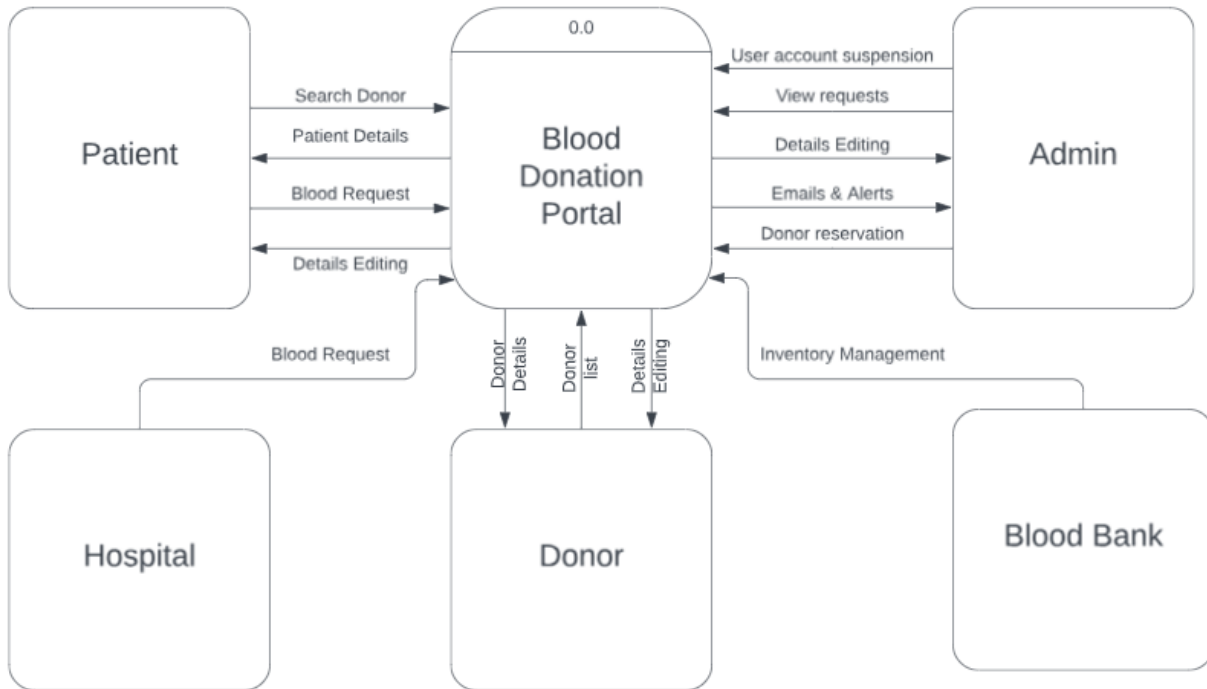

Fig. 1. Level 0 data flow diagram of the system

The basic class diagram of the users of the Blood donation portal is given below:
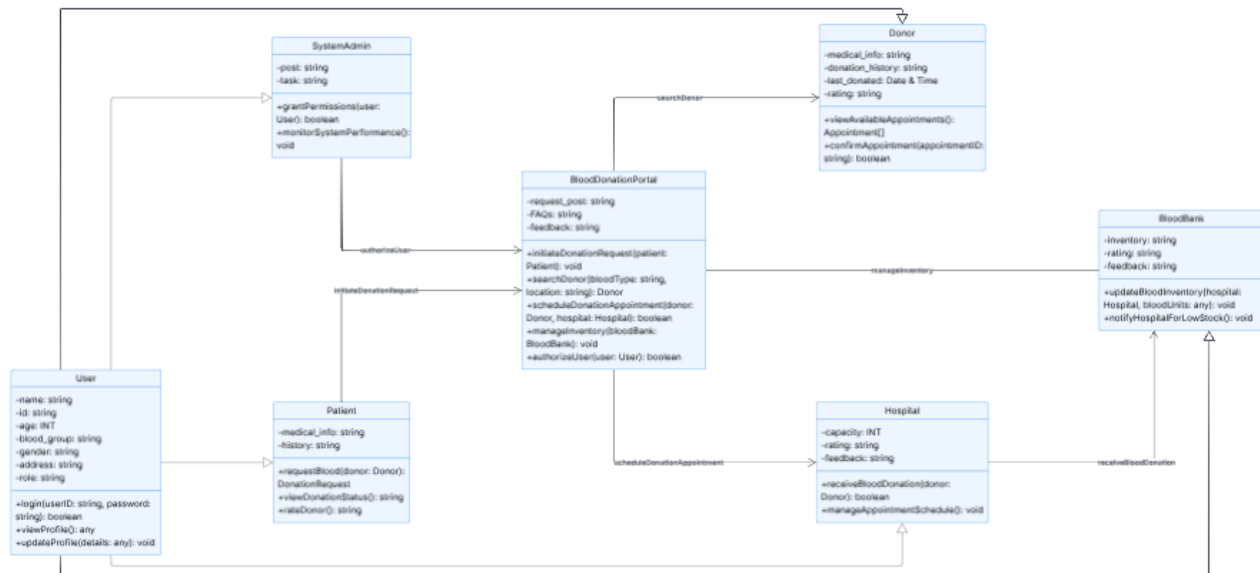

Fig. 2. Basic user class diagram

Appendix C: To Be Determined List

1.   Implementation of Modules.
2.   Testing.
3.   Limitations.
4.   Conclusion