



## **SOFTWARE REQUIREMENTS SPECIFICATION**

---

HemaTrack

---



**Version 1.0 approved**  
**Submitted by: Eisha Khan**  
**Roll number: 23L-7805**  
**Submitted to: Ms. Moomna Zenab**

**MARCH 31, 2024**

**EISHA KHAN**  
**23L-7805**

# **Software Requirements Specification**

## **Table of Contents**

### **ii Revision History**

#### **1. Introduction**

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 References

#### **2. Overall Description**

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

#### **3. External Interface Requirements**

- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

#### **4. System Features**

- 4.1 System Feature
- 4.2 System Feature 2 (and so on)

#### **5. Other Nonfunctional Requirements**

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes
- 5.5 Business Rules

#### **6. Other Requirements**

#### **Appendix A: Glossary**

#### **Appendix B: Analysis Models**

#### **Appendix C: To Be Determined List**

#### **Revision History**

# **1. Introduction**

## **1.1 Purpose**

**Product:** HemaTrack

**Scope:** The SRS outlines the software requirements for the HemaTrack system, a blood management and donation tracking application. This document covers the entire scope of the HemaTrack system, including all major features and functionalities. It describes the requirements for both the frontend user interface and the backend system components. Additionally, it encompasses administrative tools and interfaces for managing users, blood stock, and system settings. This SRS serves as a comprehensive guide for the development and implementation of the HemaTrack application.

## **1.2 Document Conventions**

The document follows standard document conventions to ensure clarity, consistency, and readability throughout the Software Requirements Specification (SRS). Each requirement statement is structured with a unique identifier and includes a clear description of the requirement, along with its priority. Priorities assigned to higher-level requirements are assumed to be inherited by detailed requirements within the same section unless explicitly stated otherwise. The document employs consistent language and terminology, with key terms highlighted for emphasis. Section headings are utilized to organize the document logically, facilitating navigation and comprehension. Technical terms and acronyms are defined where necessary to aid understanding. Additionally, requirement statements are written in clear, concise language to minimize ambiguity, and each requirement is formulated with specific intent and clarity of purpose. Overall, these document conventions contribute to the effectiveness of the SRS in conveying the project requirements accurately to stakeholders and development teams.

## **1.3 Intended Audience and Reading Suggestions**

The intended audience for this Software Requirements Specification (SRS) includes developers, project managers, testers, and documentation writers involved in the development and implementation of the software system. Additionally, stakeholders such as users and marketing staff may find value in understanding the system's requirements and functionalities.

The SRS contains an overview of the software system, including its purpose, scope, and high-level features. It outlines functional and nonfunctional requirements, detailing specific functionalities, performance criteria, and constraints. The document also includes use cases, describing various scenarios and interactions between users and the system.

For readers, it is suggested to begin with the overview sections to gain a comprehensive understanding of the software system's purpose and scope. Developers and project managers can then focus on the detailed functional requirements and use cases to understand specific features and interactions. Testers may refer to both functional and nonfunctional requirements to plan and execute testing procedures. Documentation writers can use the entire document as a reference to create user manuals and technical documentation.

## **1.4 Product Scope**

HemaTrack is a comprehensive software solution designed to streamline blood management and donation tracking processes for healthcare organizations, blood banks, and donation centers. The primary purpose of HemaTrack is to facilitate the efficient and secure management of blood inventory, donor information, and blood donation records, ultimately contributing to improved patient care and safety.

The software's **main objectives and goals** include:

1. **Efficient Blood Inventory Management:** HemaTrack enables organizations to maintain accurate and up-to-date records of blood supplies, including blood types, quantities, expiration dates, and storage locations. By providing real-time visibility into blood inventory levels, the software helps prevent shortages and ensures that adequate supplies are available for patient transfusions.
2. **Enhanced Donor Management:** HemaTrack facilitates the registration, screening, and management of blood donors, allowing organizations to maintain comprehensive donor profiles and track donation histories. The software helps streamline the donor recruitment process, identify eligible donors, and promote regular blood donations.
3. **Seamless Donation Tracking:** HemaTrack enables organizations to track the entire lifecycle of blood donations, from donor registration to product processing and distribution. The software automates donation tracking workflows, reduces manual errors, and improves traceability, thereby enhancing the safety and quality of blood products.
4. **Compliance and Reporting:** HemaTrack helps organizations comply with regulatory requirements and quality standards governing blood banking and transfusion medicine. The software generates customizable reports, audits, and documentation to support regulatory inspections, quality assurance initiatives, and accreditation processes.

By providing a centralized platform for blood management and donation tracking, HemaTrack aligns with corporate goals and business strategies aimed at optimizing healthcare delivery, reducing costs, and improving patient outcomes. The software empowers organizations to achieve greater efficiency, transparency, and compliance in their blood management practices, ultimately contributing to the mission of saving lives through safe and effective blood transfusions.

## 1.5 References

This SRS document stands alone in detailing the software requirements for HemaTrack, a system designed for managing blood and tracking donations. While there are no specific references provided here, additional documents such as user interface style guides, contracts, and system requirements specifications may be utilized or developed as necessary throughout the development phase.

## 2. Overall Description

### 2.1 Product Perspective

HemaTrack is a standalone product developed to modernize blood management and donation tracking processes within healthcare organizations. It is not part of a product family or a replacement for existing systems, but rather a new solution tailored specifically for blood transfusion medicine. HemaTrack offers comprehensive functionality, including donor management, blood inventory tracking, patient transfusion management, and administrative tools, all within a single, integrated platform. While it does not directly interface with other systems, it may integrate with external services to ensure interoperability. Overall, HemaTrack aims to streamline operations and enhance efficiency in blood management practices.

### 2.2 Product Functions

**User Authentication and Authorization:**

- Allow users to register, log in, and manage their accounts securely.
- Implement role-based access control to ensure appropriate permissions for different user roles.

#### **Blood Donation Management:**

- Enable donors to register, update their information, and schedule donation appointments.
- Track the entire donation process, from donor screening to blood collection and processing.
- Manage blood inventory, including blood typing, storage, and expiration tracking.

#### **Donor and Patient Management:**

- Maintain comprehensive donor profiles, including medical history, donation records, and eligibility status.
- Enable healthcare providers to request blood products for patients, track transfusion histories, and manage patient records.

#### **Administrative Tools:**

- Provide administrators with tools to manage user accounts, permissions, and system settings.
- Generate reports and analytics on blood donation trends, inventory levels, and patient outcomes.

#### **Communication and Notification:**

- Send notifications to donors for appointment reminders, eligibility updates, and donation milestones.
- Notify healthcare providers of blood product availability, pending requests, and critical alerts.

These functions form the core capabilities of HemaTrack, addressing the needs of donors, healthcare providers, administrators, and other stakeholders involved in blood management and donation tracking. They ensure efficient operation of the system and support the overarching goal of improving patient care and safety.

### **2.3 User Classes and Characteristics**

HemaTrack anticipates the following user classes:

1. **Donors:** Individuals who donate blood. They may access the system to schedule appointments, update their information, and view their donation history.
2. **Healthcare Providers:** Medical professionals responsible for patient care. They may use the system to request blood products, track transfusions, and access patient records.
3. **Patients:** Individuals receiving medical treatment, including blood transfusions. They may access the system to view their medical history, track transfusions, and communicate with healthcare providers.
4. **Administrators:** Staff members responsible for managing the HemaTrack system. They have elevated privileges to configure system settings, manage user accounts, and generate reports.
5. **System Managers:** IT personnel responsible for maintaining and updating the HemaTrack system. They ensure the system operates smoothly, perform backups, and troubleshoot technical issues.

Each user class may have varying levels of technical expertise, security privileges, and frequency of use. Donors, patients, and healthcare providers interact with the system to manage medical records and blood-related information, while administrators and system managers

handle system administration tasks.

## **2.4 Operating Environment**

HemaTrack operates in a web-based environment and is platform-independent. It is compatible with common web browsers such as Chrome, Firefox, and Safari. The system is hosted on a server infrastructure meeting minimum hardware requirements, and it interacts with databases (MySQL) for data storage. HemaTrack requires an internet connection for user access and may integrate with external services or APIs for additional functionality.

## **2.5 Design and Implementation Constraints**

Constraints for HemaTrack include compliance with regulatory standards for healthcare data privacy and security. The system must adhere to web development best practices and accessibility standards to ensure usability for all users. Integration with external systems or databases must follow established protocols and APIs. Additionally, the technology stack and development tools are predetermined based on project requirements and organizational standards.

## **2.6 User Documentation**

User documentation for HemaTrack includes:

- user manuals,
- online help resources, and
- tutorials.

Documentation will be provided in digital formats accessible through the web interface, with tutorials available for key functionalities. The documentation will adhere to standard formatting and style guidelines for clarity and usability, catering to users with varying levels of technical proficiency.

## **2.7 Assumptions and Dependencies**

### **Assumptions:**

1. **Availability of Third-Party Components:** We assume that the necessary third-party components or libraries, such as authentication mechanisms or data visualization tools, will be accessible and compatible with the HemaTrack system.
2. **Compliance with Regulatory Standards:** It is assumed that the development team will adhere to regulatory standards for healthcare data privacy and security throughout the development process.

### **Dependencies:**

1. **External APIs and Services:** The project relies on the availability and reliability of external APIs and services for functionalities like geolocation services, email notifications, and payment processing.
2. **Integration with Healthcare Information Systems:** The project's success depends on seamless integration with existing healthcare information systems or databases to exchange patient data and ensure interoperability.

## **3. External Interface Requirements**

### **3.1 User Interfaces**

The software product requires user interfaces to facilitate interaction between the users and the system. The logical characteristics of the user interfaces can include the following:

1. **Graphical User Interface (GUI):** The software product may utilize a graphical user interface that presents visual elements, controls, and screens to users. The GUI should

follow appropriate design principles, usability standards, and guidelines to ensure a consistent and intuitive user experience. Specific GUI standards or product family style guides, if applicable, should be followed to maintain a cohesive look and feel across the software product.

2. **Screen Layout and Navigation:** The screen layout of the user interfaces should be designed to optimize usability and efficiency. This may include organizing information, controls, and functionalities in a logical and intuitive manner. Navigation elements, such as menus, tabs, buttons, or hyperlinks, should be provided to allow users to move between screens or access different features of the software.
3. **Standard Buttons and Functions:** The user interfaces may include standard buttons and functions that appear consistently across screens. These standard buttons and functions may include features like Save, Cancel, OK, Apply, Back, Help, or Download. They should adhere to established conventions and provide expected behaviors to users.

These user interfaces enable users to interact with HemaTrack effectively, whether accessing it through web browsers, mobile devices, or administrative tools. The design and functionality of each interface are tailored to meet the specific needs and preferences of different user roles within the system.

### 3.2 Hardware Interfaces

Hardware Interfaces for HemaTrack:

#### 1. Server Hardware:

- **Logical Characteristics:** HemaTrack is deployed on a server infrastructure, which can be physical hardware or virtualized environment..
- **Physical Characteristics:** Servers should meet minimum hardware requirements, including sufficient CPU, memory, and storage resources to support the application's workload and user base.
- **Supported Device Types:** Any server hardware capable of running the required operating system and hosting web applications.
- **Data and Control Interactions:** The server hardware hosts the HemaTrack application, managing data storage, processing user requests, and executing background tasks.
- **Communication Protocols:** TCP/IP protocols are used for network communication between the server hardware and client devices, enabling HTTP/HTTPS communication for web-based access to HemaTrack.

#### 2. Client Devices (e.g., Desktops, Laptops, Mobile Devices):

- **Logical Characteristics:** Client devices access HemaTrack through web browsers or dedicated client applications.
- **Physical Characteristics:** Client devices vary in form factor, processing power, memory, and display capabilities.
- **Supported Device Types:** HemaTrack supports a wide range of client devices, including desktop computers, laptops, tablets, and smartphones, running standard web browsers (e.g., Chrome, Firefox, Safari).
- **Data and Control Interactions:** Client devices interact with the HemaTrack server to retrieve data, submit user inputs, and display user interfaces.
- **Communication Protocols:** Client devices communicate with the server using HTTP/HTTPS protocols over the internet or local network connections, facilitating

data exchange and user interactions.

### 3.3 Software Interfaces

Software Interfaces for HemaTrack:

#### 1. Django Framework:

- **Version:** 3.0.5
- **Purpose:** Django is a high-level Python web framework used for rapid development of web applications. HemaTrack utilizes Django for implementing the backend logic, handling HTTP requests, managing user authentication, and interacting with the database.
- **Data items/messages:** HTTP requests/responses, form data, user authentication tokens.
- **Services needed:** Request handling, authentication, database interaction.
- **Nature of communications:** HTTP requests/responses, function calls within the Django framework.

#### 2. Python Libraries:

- Django forms library
- **Purpose:** The Django forms library provides a convenient way to create HTML forms and handle form data within Django applications. HemaTrack uses Django forms to define and process user input for various functionalities such as user registration, blood donation, and profile updates.
- **Data items/messages:** Form data submitted by users.
- **Services needed:** Form validation, data processing.
- **Nature of communications:** Function calls within the Django forms library.

#### 3. Django Models:

- **Purpose:** Django models represent database tables and are used to define the structure and behavior of data within the application. HemaTrack defines Django models to represent entities such as users, donors, and blood donations, specifying their fields and relationships.
- **Data items/messages:** Database queries, data manipulation commands.
- **Services needed:** Database interaction (CRUD operations), data validation.
- **Nature of communications:** SQL queries executed by Django ORM (Object-Relational Mapping).

#### 4. HTML Templates:

- **Purpose:** HTML templates are used to generate the user interface for HemaTrack web pages. These templates contain placeholders for dynamic data and are rendered by Django to produce the final HTML pages displayed to users.
- **Data items/messages:** HTML content, template variables.
- **Services needed:** Rendering dynamic content, handling user interactions.
- **Nature of communications:** HTTP responses containing HTML content.

#### 5. Database (MySQL):

- **Purpose:** The database stores persistent data for HemaTrack, including user accounts, donor information, and blood donation records. HemaTrack interacts with the database through Django models using the Django ORM.
- **Data items/messages:** Database queries, data records.
- **Services needed:** Data storage, retrieval, and manipulation.



- **Nature of communications:** SQL commands executed by the database management system.

These software interfaces facilitate the interaction between different components of HemaTrack, enabling seamless data flow, user authentication, and user interface rendering. The integration of these components ensures the smooth functioning of the application and the efficient handling of user interactions.

### 3.4 Communications Interfaces

Communications Interfaces for HemaTrack:

#### 1. Web Browser Communication:

- **Requirement:** HemaTrack interacts with users through web browsers, allowing them to access the application's features and functionalities via HTTP or HTTPS protocols.
- **Message Formatting:** HTTP requests and responses follow standard formats, such as JSON or XML, for data exchange between the client (web browser) and the server.
- **Communication Standards:** HTTP/HTTPS protocols are used for client-server communication, ensuring compatibility with web browsers and adherence to web standards.
- **Communication Security:** HTTPS protocol is used to encrypt communication between the web browser and the server, ensuring data confidentiality and integrity.
- **Data Transfer Rates:** Data transfer rates depend on factors such as network bandwidth, server load, and client-side processing capabilities. Efforts should be made to optimize performance for efficient data transfer.

#### 2. Network Server Communications:

- **Requirement:** HemaTrack may interact with external servers or APIs for various purposes, such as geolocation services, payment processing, or integration with healthcare systems.
- **Message Formatting:** Data exchanged with external servers may use standard formats such as JSON or XML, depending on the communication protocol and API specifications.
- **Communication Standards:** HTTP/HTTPS protocols are commonly used for communicating with external servers or APIs over the internet.
- **Communication Security:** Secure communication protocols (HTTPS) and authentication mechanisms (e.g., API keys) are implemented to secure interactions with external servers and protect sensitive data.
- **Data Transfer Rates:** Data transfer rates depend on factors such as network latency, server responsiveness, and API throttling limits. Efforts should be made to optimize performance and minimize latency in server communications.

#### 3. Email:

- **Requirement:** HemaTrack should be able to send email notifications for various events such as account registration, password reset, and appointment reminders.
- **Message Formatting:** Emails should be formatted in plain text or HTML format, depending on the notification type.
- **Communication Standards:** SMTP (Simple Mail Transfer Protocol) will be used for sending emails.

- **Security:** Email communication will utilize TLS (Transport Layer Security) encryption for secure transmission of messages.
- **Data Transfer Rates:** Email notifications should be sent promptly upon triggering the corresponding events.
- **Synchronization Mechanisms:** Email sending operations should be asynchronous to prevent blocking the user interface.

These communications interfaces ensure seamless interaction between HemaTrack and its users, as well as external services, while maintaining security, data integrity, and efficient data transfer.

## 4. System Features

### 4.1 User Accounts (Donor/Patient)

#### 4.1.1 Description and Priority

This functionality enables users (whether they're donors or patients) to sign up, sign in, and oversee their account information, which encompasses personal details and donation records.

*Priority: High*

#### 4.1.3 Functional Requirements

REQ-1: The User should be able to login and logout with a unique username & password.

REQ-2: The system should add a new user (register user) to the database & system.

#### Use Case: a. Login

Use Case Identification and History			
Use Case ID:	UL:1		
Use Case Name:	Login User		
End Objective	The user should be able to log in to the website		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Donor/Patient		
4.1.2 Stimulus/Response Sequences			
Basic Flow			
Step	User Actions	System Actions	
1.	The user will enter their registered username or password and click on the login button	The system validates the credentials and logs in the user successfully.	
2.	The user will enter their personal information and create new account	The user will register on the application successfully.	

3.	After successful login, the user will see account dashboard	The system will successfully display his/her profile information and provide access to features like a make Donation request and donation history.
Alternate Flow		
Step	User Actions	System Actions
1.	The user will enter their username or password and click on the login button	The system will display an error message that “login Failed please enter correct username or password”
2.	The user will enter their personal information and create his/her account	The User already registered
Pre-conditions:		
The Donor or Patient should be in the logged-out state.		
Post-condition:		
The user will be in logged in state and granted access to their account.		

#### B. Register:

Use Case Identification and History			
Use Case ID:	UL:1		
Use Case Name:	Register User		
End Objective	The user should be able to log in to the website		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Donor/Patient		
4.1.2 Stimulus/Response Sequences			
Basic Flow			
Step	User Actions	System Actions	

1.	The user will enter their registered username or password and click on the login button	The system validates the credentials and logs in the user successfully.
2.	The user will enter their personal information and create new account	The user will register on the application successfully.
3.	After successful login, the user will see account dashboard	The system will successfully display his/her profile information and provide access to features like a make Donation request and donation history.
Alternate Flow		
Step	User Actions	System Actions
1.	The user will enter their username or password and click on the login button	The system will display an error message that "login Failed please enter correct username or password"
2.	The user will enter their personal information and create his/her account	The User already registered
Pre-conditions:		
The Donor or Patient has not been registered yet.		
Post-condition:		
System User has successfully registered to the system.		

## 4.2 Download/View History

### 4.2.1 Description and Priority

The "Download History" feature is specifically tailored for the administrator's use. It empowers the administrator to access and review the history of downloads made from the platform. This functionality provides the administrator with valuable insights into the platform's usage patterns, facilitating better management of resources and monitoring of user activity.

**Priority:** *Medium*

### 4.2.3 Functional Requirements

REQ-1: The system should authenticate the administrator's credentials for access to the "Download History" feature.

REQ-2: The system must retrieve and display the download history data upon request by the administrator.

### Use Case: a. Authentication

#### Use Case Identification and History

Use Case ID:	UL:2		
Use Case Name:	Authenticate Administrator Credentials		
End Objective	to ensure that only authorized administrators can access the "Download History" feature within the system		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Admin		

#### 4.2.2 Stimulus/Response Sequences

##### Basic Flow

Step	User Actions	System Actions
1.	The user navigates to the "Download History" section within the admin dashboard.	The system verifies the user credentials for access to the "Download History" feature.
2.	The administrator view specific download history data.	The system retrieves and displays the download history data.
3.	The administrator may choose to download the complete history or view it directly on the dashboard.	If the user opts to download the history, the system generates a downloadable file in the .csv format or if the user chooses to view the history directly, the system presents the data in a user-friendly format within the admin dashboard.

##### Alternate Flow

Step	User Actions	System Actions
1.	Encountering an error message due to incorrect credentials, the user may choose to re-enter the credentials or reset the password.	If the credentials are incorrect, the system prompts the user to re-enter them or offers the option to reset the password.
2.	After successfully logging in, the user navigates to the "Download History"	After successful validation or password reset, the system logs in the user. When the user accesses the "Download History"

	section within the admin dashboard.	section, the system verifies the user's credentials for access to this feature.
Pre-conditions:		
The administrator has been registered in the system with a unique username and password. The administrator has navigated to the admin dashboard section where the "Download History" feature is accessible.		
Post-condition:		
Upon successful authentication, the administrator gains access to the "Download History" feature and can proceed to view or interact with the download history data.		

## B. Download/view

### Use Case Identification and History

Use Case ID:	UL:3		
Use Case Name:	Retrieve and Display /Download History Data		
End Objective	The end objective of this use case is to enable administrators to effectively retrieve and review download history data from the system.		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Admin		

### 4.2.2 Stimulus/Response Sequences

Basic Flow		
Step	User Actions	System Actions
1.	The user navigates to the "Download History" section within the admin dashboard.	The system verifies the user credentials for access to the "Download History" feature.
2.	The administrator view specific download history data.	The system retrieves and displays the download history data.
3.	The administrator may choose to download the complete history or view it directly on the dashboard.	If the user opts to download the history, the system generates a downloadable file in the .csv format or if the user chooses to view the history directly, the system presents the

		data in a user-friendly format within the admin dashboard.
Alternate Flow		
Step	User Actions	System Actions
1.	Encountering an error message due to incorrect credentials, the user may choose to re-enter the credentials or reset the password.	If the credentials are incorrect, the system prompts the user to re-enter them or offers the option to reset the password.
2.	After successfully logging in, the user navigates to the "Download History" section within the admin dashboard.	After successful validation or password reset, the system logs in the user. When the user accesses the "Download History" section, the system verifies the user's credentials for access to this feature.
Pre-conditions:		
The administrator has successfully authenticated their credentials and gained access to the "Download History" feature within the system.		
Post-condition:		
The administrator has successfully retrieved and reviewed the download history data. The system remains accessible for further administrative tasks.		

## 4.3 Update Blood Stock

### 4.3.1 Description and Priority

The "Update Blood Stock" feature allows authorized users, typically blood bank administrators or staff, to manage and update the inventory of blood stock within the system. This functionality enables users to add new blood donations, remove expired or unusable units, and update the quantities of available blood types. Maintaining an accurate and up-to-date blood stock inventory is crucial for ensuring that blood supplies meet the demands of patients and healthcare facilities. Therefore, this feature is of high priority to support efficient blood management and facilitate timely responses to blood donation needs.

*Priority:High*

### 4.3.3 Functional Requirements

REQ-1: The system should provide functionality for authorized users to update the quantities of available blood types in the inventory.

#### Use Case: a. Update Blood Stock Quantities

Use Case Identification and History	
Use Case ID:	UL:4
Use Case Name:	Update Blood Stock Quantities

End Objective	Update blood stock quantities in the system's inventory to reflect the latest availability of each blood type.		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Admin		
4.3.2 Stimulus/Response Sequences			
Basic Flow			
Step	User Actions	System Actions	
1.	The authorized user navigates to the "Update Blood Stock" section within the system.	The system displays the interface for managing blood stock.	
2.	The user selects the option to add a new blood donation	The system prompts the user to enter details such as blood type, quantity, and expiration date for the new donation.	
3.	The user submits the information for the new blood donation..	The system validates the data and updates the blood stock inventory accordingly.	
4	The user submits the updated quantities of available blood types.	The system verifies the data and updates the blood stock inventory accordingly.	
5	The user completes the blood stock management tasks and exits the "Update Blood Stock" section.	The system saves the changes made to the blood stock inventory and returns the user to the main menu or dashboard.	
Alternate Flow			
Step	User Actions	System Actions	
1.	Encountering an error message due to incorrect credentials while the authorized user navigates to the "Update Blood Stock" section within the system.	If the credentials are incorrect, the system prompts the user to re-enter them or offers the option to reset the password.	



2.	After successfully logging in, the user navigates to the "Update Stock" section within the admin dashboard.	After successful validation or password reset, the system logs in the user. When the user accesses the "Update Stock" section, the system verifies the user's credentials for access to this feature.
Pre-conditions:		
The authorized user has successfully logged into the system.		
The user has navigated to the "Update Blood Stock" section within the system.		
Post-condition:		
The blood stock inventory reflects the updated quantities entered by the user.		
The system remains accessible for further blood stock management tasks.		

#### 4.4 Donate Blood

##### 4.4.1 Description and Priority

The "Donate Blood" feature allows users to register as blood donors and participate in blood donation drives or events..

*Priority: High*

##### 4.4.3 Functional Requirements

REQ-1: When a user donates blood, the system should accept/reject donation.

REQ-2: When a user donates blood, the system should record the donation and update the donor's status accordingly.

##### Use Case: a.Blood Donation Approval

###### Use Case Identification and History

Use Case ID:	UL:5		
Use Case Name:	Blood Donation Approval		
End Objective	To facilitate the blood donation process by allowing users to submit blood donation requests and ensuring that the system accurately evaluates and either accepts or rejects these requests based on predefined eligibility criteria.		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	donor		

###### 4.4.2 Stimulus/Response Sequences

Basic Flow		
Step	User Actions	System Actions

1.	User navigates to the "Donate Blood" section on the platform or website.	System displays the "Donate Blood" section with registration options.
2.	User selects the option to register as a blood donor.	System prompts the user to fill out the necessary personal information for donor registration.
3.	User fills out the required personal information and completes the registration process.	System validates the entered information and registers the user as a blood donor.
4	User donates the blood	System approves the blood donation request.
Alternate Flow		
Step	User Actions	System Actions
1.	User navigates to the "Donate Blood" section on the platform or website.	System displays the registration form for blood donation.
2.	User selects the option to register as a blood donor.	System validates the entered information and detects errors.
3.	User fills out the required personal information for registration.	System prompts the user to correct the errors in the registration form.
4.	User encounters an error due to missing or incorrect information.	System updates the registration form with the corrected information.
5.	User corrects the errors in the registration form.	System revalidates the updated information and registers the user as a blood donor.

6.	User resubmits the registration form.	System confirms the successful registration and provides further instructions for blood donation participation.
Pre-conditions:		
The user has completed the registration and login process.		
Post-condition:		
If accepted, the blood donation request is successfully processed and recorded in the system. If rejected, the user is informed of the rejection reason and may be provided with guidance on how to proceed.		

## B. Record Blood Donation and Update Donor Status

### Use Case Identification and History

Use Case ID:	UL:5		
Use Case Name:	Record Blood Donation and Update Donor Status		
End Objective	To maintain accurate records of blood donations made by users and update the donor's status within the system to reflect their donation activity.		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Donor		

### 4.4.2 Stimulus/Response Sequences

Basic Flow		
Step	User Actions	System Actions
1.	User navigates to the "Donate Blood" section on the platform or website.	System displays the "Donate Blood" section with registration options.
2.	User selects the option to register as a blood donor.	System prompts the user to fill out the necessary personal information for donor registration.
3.	User fills out the required personal information and completes the registration process.	System validates the entered information and registers the user as a blood donor.

4	User donates the blood	System approves the blood donation request.
Alternate Flow		
Step	User Actions	System Actions
1.	User navigates to the "Donate Blood" section on the platform or website.	System displays the registration form for blood donation.
2.	User selects the option to register as a blood donor.	System validates the entered information and detects errors.
3.	User fills out the required personal information for registration.	System prompts the user to correct the errors in the registration form.
4.	User encounters an error due to missing or incorrect information.	System updates the registration form with the corrected information.
5.	User corrects the errors in the registration form.	System revalidates the updated information and registers the user as a blood donor.
6.	User resubmits the registration form.	System confirms the successful registration and provides further instructions for blood donation participation.
Pre-conditions:		
The user has successfully donated blood.		
Post-condition:		
The system updates the database accordingly.		

#### 4.5 Request for Blood

##### 4.5.1 Description and Priority

The "Request for Blood" feature allows users to register as Patient and request for blood

*Priority: High*

#### 4.4.3 Functional Requirements

REQ-1: When a user request blood, the system should accept/reject the request.

REQ-2: When the user accepts or rejects the request, the system should record the request and update the patient's status accordingly.

##### Use Case: a.Blood Request Approval

Use Case Identification and History			
Use Case ID:	UL:7		
Use Case Name:	Blood Request Approval		
End Objective	To facilitate the blood donation process by allowing users to submit blood requests and ensuring that the system accurately evaluates and either accepts or rejects these requests based on predefined eligibility criteria.		
Created by:	Eisha Khan	On (date):	30/March/2024
Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	Patient		

#### 4.5.2 Stimulus/Response Sequences

Basic Flow		
Step	User Actions	System Actions
1.	User navigates to the "Request Blood" section on the platform or website.	System displays the "Request Blood" section with registration options.
2.	User selects the option to register as a Patient.	System prompts the user to fill out the necessary personal information for Patient registration.
3.	User fills out the required personal information and completes the registration process.	System validates the entered information and registers the user as a Patient.
4	User request for blood	System approves the blood request.
Alternate Flow		
Step	User Actions	System Actions

1.	User navigates to the "Request for Blood" section on the platform or website.	System displays the registration form for blood request.
2.	User selects the option to register as a patient.	System validates the entered information and detects errors.
3.	User fills out the required personal information for registration.	System prompts the user to correct the errors in the registration form.
4.	User encounters an error due to missing or incorrect information.	System updates the registration form with the corrected information.
5.	User corrects the errors in the registration form.	System revalidates the updated information and registers the user as a blood donor.
6.	User resubmits the registration form.	System confirms the successful registration and provides further instructions for blood request.

Pre-conditions:

The user has completed the registration and login process.

Post-condition:

If accepted, the blood request is successfully processed record in the system.

If rejected, the user is informed of the rejection reason and may be provided with guidance on how to proceed.

## B. Record Blood request and Update Patient Status

### Use Case Identification and History

Use Case ID:	UL:8		
Use Case Name:	Record Blood request and Update Patient Status		
End Objective	To maintain accurate records of blood requests made by users and update the patient's status within the system to reflect their request activity.		
Created by:	Eisha Khan	On (date):	30/March/2024

Approved By:	Eisha Khan	On (date):	30/March/2024
User/Actor	patient		
4.5.2 Stimulus/Response Sequences			
Basic Flow			
Step	User Actions	System Actions	
1.	User navigates to the "Request Blood" section on the platform or website.	System displays the "Request Blood" section with registration options.	
2.	User selects the option to register as a Patient.	System prompts the user to fill out the necessary personal information for Patient registration.	
3.	User fills out the required personal information and completes the registration process.	System validates the entered information and registers the user as a Patient.	
4	User request for blood	System approves the blood request.	
Alternate Flow			
Step	User Actions	System Actions	
1.	User navigates to the "Request for Blood" section on the platform or website.	System displays the registration form for blood request.	
2.	User selects the option to register as a patient.	System validates the entered information and detects errors.	
3.	User fills out the required personal information for registration.	System prompts the user to correct the errors in the registration form.	

4.	User encounters an error due to missing or incorrect information.	System updates the registration form with the corrected information.
5.	User corrects the errors in the registration form.	System revalidates the updated information and registers the user as a blood donor.
6.	User resubmits the registration form.	System confirms the successful registration and provides further instructions for blood request.
Pre-conditions:		
The user has successfully requested for blood.		
Post-condition:		
The system updates the database according to rejection/approval.		

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

Performance Requirements:

1. **Response Time:** The system should respond to user actions within 2 seconds under normal load conditions to ensure a smooth and responsive user experience. This requirement is essential to prevent user frustration and maintain user engagement with the platform.
2. **Transaction Processing Time:** The system should process blood donation requests and record donations within 5 seconds to ensure timely updating of donor records and availability of blood inventory information. This requirement is crucial for efficient blood management and quick response to patient needs.
3. **Scalability:** The system should be capable of handling a significant increase in user activity, such as during blood donation drives or emergencies, without significant degradation in performance. It should scale horizontally by adding additional resources to handle increased load effectively.
4. **Reliability:** The system should maintain a high level of reliability, with a target uptime of 99.9% to minimize downtime and ensure continuous availability of blood donation services to users. This requirement is critical for uninterrupted access to blood donation facilities, especially during emergencies.
5. **Data Retrieval Time:** The system should retrieve download history data within 3 seconds upon request by the administrator to provide timely access to essential information for monitoring platform usage and user activity.
6. **Concurrency:** The system should support concurrent user access, allowing multiple users to register as blood donors, submit blood donation requests, and access blood donation



records simultaneously without performance degradation or data inconsistency issues.

**Rationale:**

- Timely response and processing times ensure that critical actions, such as blood donation requests and recording donations, are handled efficiently, contributing to the overall effectiveness of blood management and patient care.
- Scalability is crucial to accommodate fluctuations in user activity, particularly during peak periods or emergencies, without compromising system performance or user experience.
- Reliability is paramount to maintain continuous access to blood donation services, especially during critical situations where timely access to blood can be a matter of life and death.
- Efficient data retrieval time for download history data enables administrators to monitor platform usage effectively, facilitating informed decision-making and resource allocation.

## **5.2 Safety Requirements**

### **Software Requirements Specification for Hematrack**

1. **Safety Testing and Validation:** The system shall undergo thorough safety testing and validation to identify and mitigate potential safety risks. This may include functional testing, security testing, and risk assessments to ensure the product's safety and reliability.
2. **User Safety Training:** The system shall provide appropriate user safety training materials or resources to educate users on the safe and proper use of the product. These materials may include user manuals, instructional videos, or online resources.
3. **Error Prevention:** The system shall incorporate mechanisms to prevent or minimize errors that could potentially lead to user harm or data loss. This may include input validation, error-checking routines, and clear error messages to guide users in taking corrective actions.
4. **Data Integrity and Reliability:** The system shall ensure the integrity and reliability of user data to prevent potential harm caused by data corruption or loss. This can be achieved through backup mechanisms, data validation, and error detection and recovery mechanisms.

These safety requirements address the prevention of harm or damage resulting from the use of the product. They emphasize user safety, data integrity, compliance with safety standards and regulations, and the implementation of safeguards against potential risks. The specific safety requirements may vary depending on the nature of the product and the industry in which it operates.

## **5.3 Security Requirements**

1. **User Authentication:** The system shall require users to authenticate themselves with a unique username and password combination before accessing any sensitive information or performing privileged actions.
2. **Role-Based Access Control:** The system shall enforce role-based access control, ensuring that users can only access functionalities and data relevant to their assigned roles.
3. **Password Policy:** The system shall enforce a strong password policy, requiring users to create passwords that meet specified complexity requirements (e.g., minimum length,

- inclusion of alphanumeric and special characters) and periodically change their passwords.
4. **Data Encryption:** The system shall encrypt sensitive data, such as user credentials and personal health information, to protect it during transmission and storage. Encryption algorithms and key management practices shall adhere to industry standards.
  5. **Secure Communication:** The system shall use secure communication protocols (e.g., HTTPS) to ensure the confidentiality and integrity of data exchanged between the client and the server.
  6. **Access Control:** The system shall implement access controls to restrict unauthorized access to sensitive functionalities and data. Access controls shall be based on the principle of least privilege, granting users only the permissions necessary to perform their tasks.

These security requirements address the protection of user data, secure access control, secure communication, compliance with regulations, and adherence to industry best practices. The specific security requirements may vary depending on the context and the applicable policies or regulations governing the product's use.

#### 5.4 Software Quality Attributes

Additional quality characteristics for the HemaTrack product:

1. **Adaptability:** The system should be adaptable to accommodate changes in blood donation regulations, medical protocols, and user requirements. This includes the ability to easily integrate new features or modules without significant disruption to existing functionality.
2. **Availability:** The system should aim for high availability to ensure that users can access it whenever needed. This can be quantified as a percentage of uptime over a specific period, with a target availability of at least 99.9%.
3. **Correctness:** The system should consistently produce accurate results and perform actions as expected. This can be verified through rigorous testing procedures, including unit tests, integration tests, and system tests, with a target defect rate of less than 1% in production.
4. **Maintainability:** The system should be designed and implemented in a way that facilitates ease of maintenance and updates by developers. This includes clear code documentation, modular design principles, and adherence to coding standards, with a target code maintainability index of at least 80.
5. **Reliability:** The system should be reliable in handling user interactions and processing data without unexpected failures or errors. This can be measured by mean time between failures (MTBF), with a target MTBF of at least 10,000 hours.
6. **Usability:** The system should be user-friendly and intuitive, allowing users to accomplish tasks efficiently and effectively. This can be assessed through user feedback surveys and usability testing, with a target user satisfaction score of at least 4 out of 5.
7. **Robustness:** The system should be robust in handling unexpected inputs or adverse conditions without crashing or malfunctioning. This can be verified through stress testing and boundary testing, with a target of no critical failures under extreme conditions.
8. **Testability:** The system should be designed with testability in mind, allowing for comprehensive testing of all components and functionalities. This includes providing adequate logging and debugging capabilities, with a target test coverage of at least 80%.
9. **Interoperability:** The system should be interoperable with other healthcare systems and services, enabling seamless data exchange and integration. This can be verified through interoperability testing with industry-standard protocols, with a target of passing interoperability tests with major healthcare platforms.

These quality characteristics are essential to both customers and developers, ensuring that the HemaTrack product meets high standards of performance, reliability, and usability while also being adaptable to evolving needs and technologies in the healthcare domain.

## **5.5 Business Rules**

List of operating principles about HemaTrack:

1. **User Authentication:** Only registered users with valid credentials can access the HemaTrack system.
2. **User Roles:** The system supports different user roles, including Administrator, Superuser, Donor, and Patient. Each role has specific privileges and restrictions.
3. **Superuser Access:** Only the Superuser has the authority to perform administrative tasks such as managing users, approving/rejecting blood donation requests, and managing blood stock.
4. **Blood Donation Eligibility:** Donors must meet certain eligibility criteria, such as age, health condition, and previous donation history, to be eligible to donate blood.
5. **Blood Stock Management:** The system enforces rules for managing blood stock, such as maintaining an adequate supply of various blood types, tracking expiration dates, and ensuring proper handling and storage.
6. **Request for Blood:** Patients can request blood only if they meet the specified criteria and have a valid medical need for it.
7. **Approval/Rejection of Blood Donation Requests:** The Superuser reviews and approves/rejects blood donation requests based on the availability of matching blood types and the patient's medical condition.
8. **Donor Privacy:** Donor information, including personal and medical details, should be kept confidential and accessible only to authorized personnel.
9. **Request Tracking:** The system tracks the status of blood requests, including pending, approved, and rejected requests, to ensure transparency and accountability.
10. **Data Security:** The system should implement appropriate security measures to protect sensitive data, including encryption, access controls, and regular backups.
11. **Compliance with Regulatory Standards:** The system should adhere to relevant regulatory standards and guidelines for blood donation and patient healthcare, ensuring legal and ethical compliance.
12. **Error Handling:** The system should handle errors gracefully, providing informative error messages and logging error details for troubleshooting and system improvement.

These business rules influence the functional requirements of the HemaTrack system, such as user authentication and authorization, blood donation request handling, user role management, data privacy, and security measures.

## **6. Other Requirements**

1. **Database Requirements:** The system shall utilize an RDBMS for data storage, adhering to normalization principles for data integrity. It shall support efficient data retrieval and querying.
2. **Internationalization Requirements:** The system shall support multiple languages and cultural norms, with easily translatable user interface elements.
3. **Legal Requirements:** The system shall comply with data privacy laws such as GDPR and implement proper consent mechanisms for user data collection.

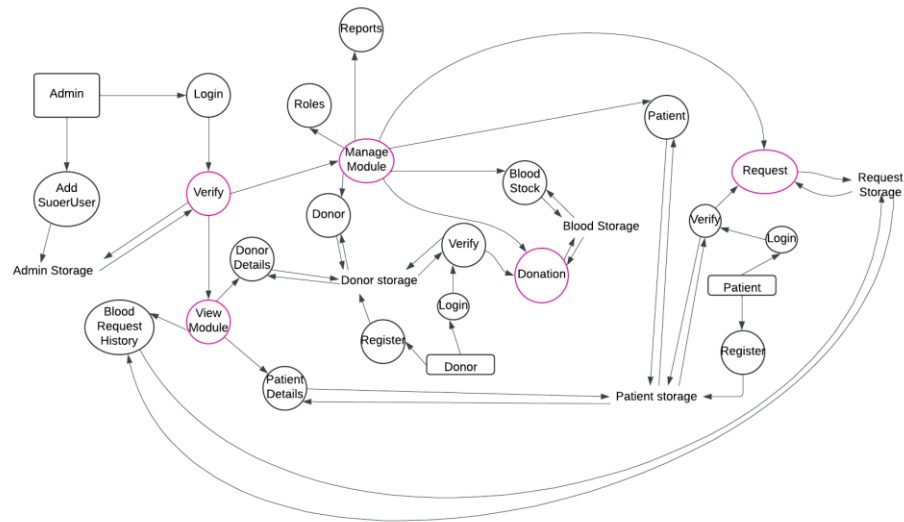
4. Reuse Objectives: The system shall be designed with modularity for code reuse and encapsulation of common functionalities into reusable modules.
5. Security Requirements: The system shall implement encryption, secure authentication, access control policies, and undergo regular security audits.
6. Accessibility Requirements: The system shall adhere to WCAG standards for accessibility, ensuring operability for users with disabilities.
7. Performance Requirements: The system shall be optimized for responsiveness and scalability, meeting predefined performance targets for critical functions.

## **Appendix A: Glossary**

All the terms necessary to properly interpret the SRS:

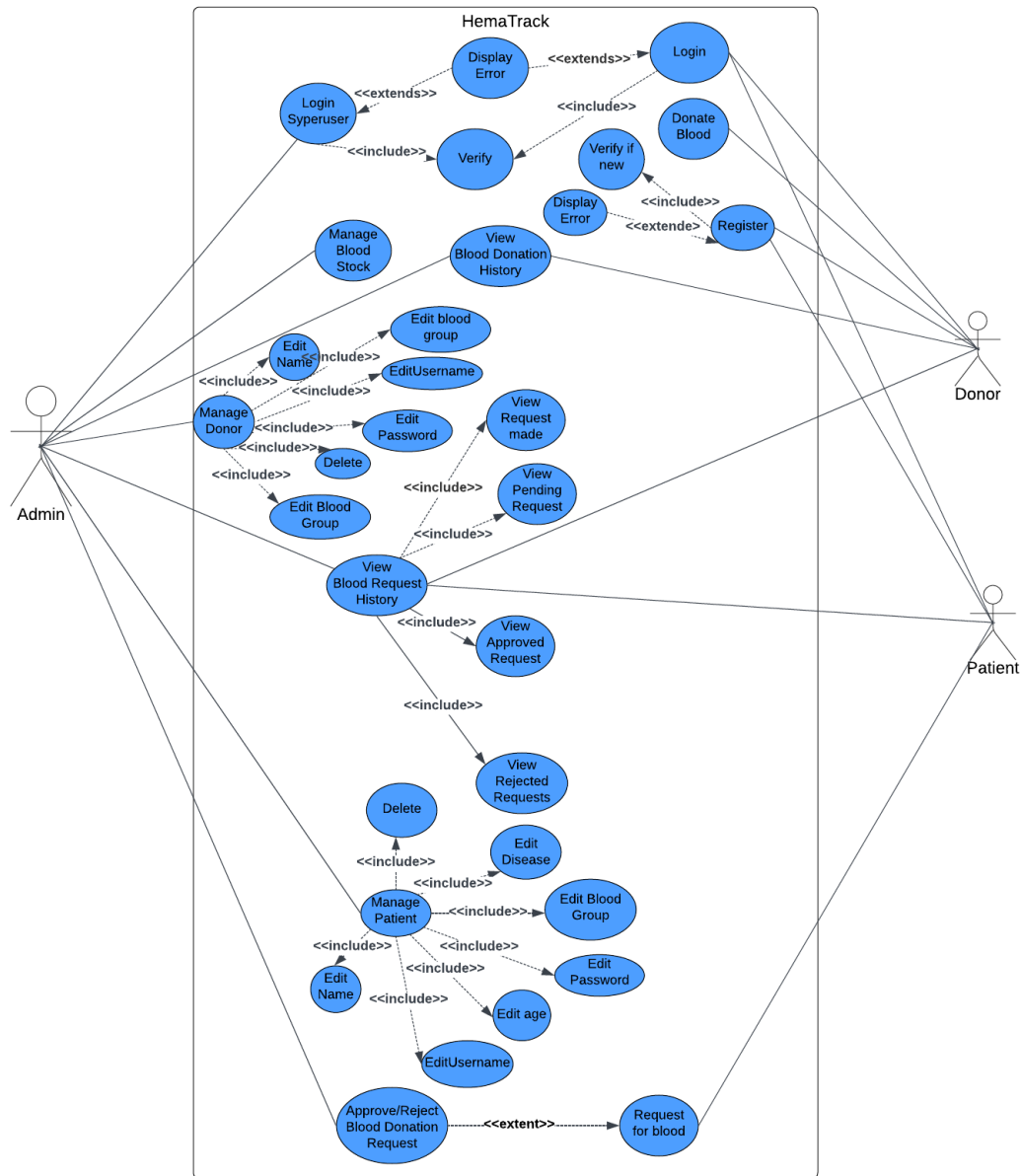
1. SRS: Software Requirements Specification - A document that describes the intended behavior and functionalities of a software system, including user requirements, system requirements, and constraints.
2. GUI: Graphical User Interface - A user interface that allows users to interact with electronic devices through graphical icons and visual indicators, as opposed to text-based interfaces.
3. CRUD: Create, Read, Update, Delete - A set of basic operations used in database management and application development for manipulating data.
4. API: Application Programming Interface - A set of protocols, tools, and definitions that allow different software applications to communicate with each other.
5. HTTP: Hypertext Transfer Protocol - An application protocol for distributed, collaborative, and hypermedia information systems, commonly used for transferring data over the World Wide Web.
6. HTTPS: Hypertext Transfer Protocol Secure - An extension of HTTP that adds encryption to secure the data being transferred between a client and a server, commonly used for secure communication over the internet.
7. UI: User Interface - The point of interaction between users and a computer program, including graphical user interfaces, command-line interfaces, and voice interfaces.
8. UX: User Experience - The overall experience of a person using a product such as a website or computer application, especially in terms of how easy or pleasing it is to use.
9. CPU: Central Processing Unit - The primary component of a computer responsible for executing instructions and performing calculations.
10. OS: Operating System - System software that manages computer hardware, software resources, and provides common services for computer programs.

### 1.Data Flow Diagram: (Level1):



## Level 1 DataFlow Diagram For HemaTrack

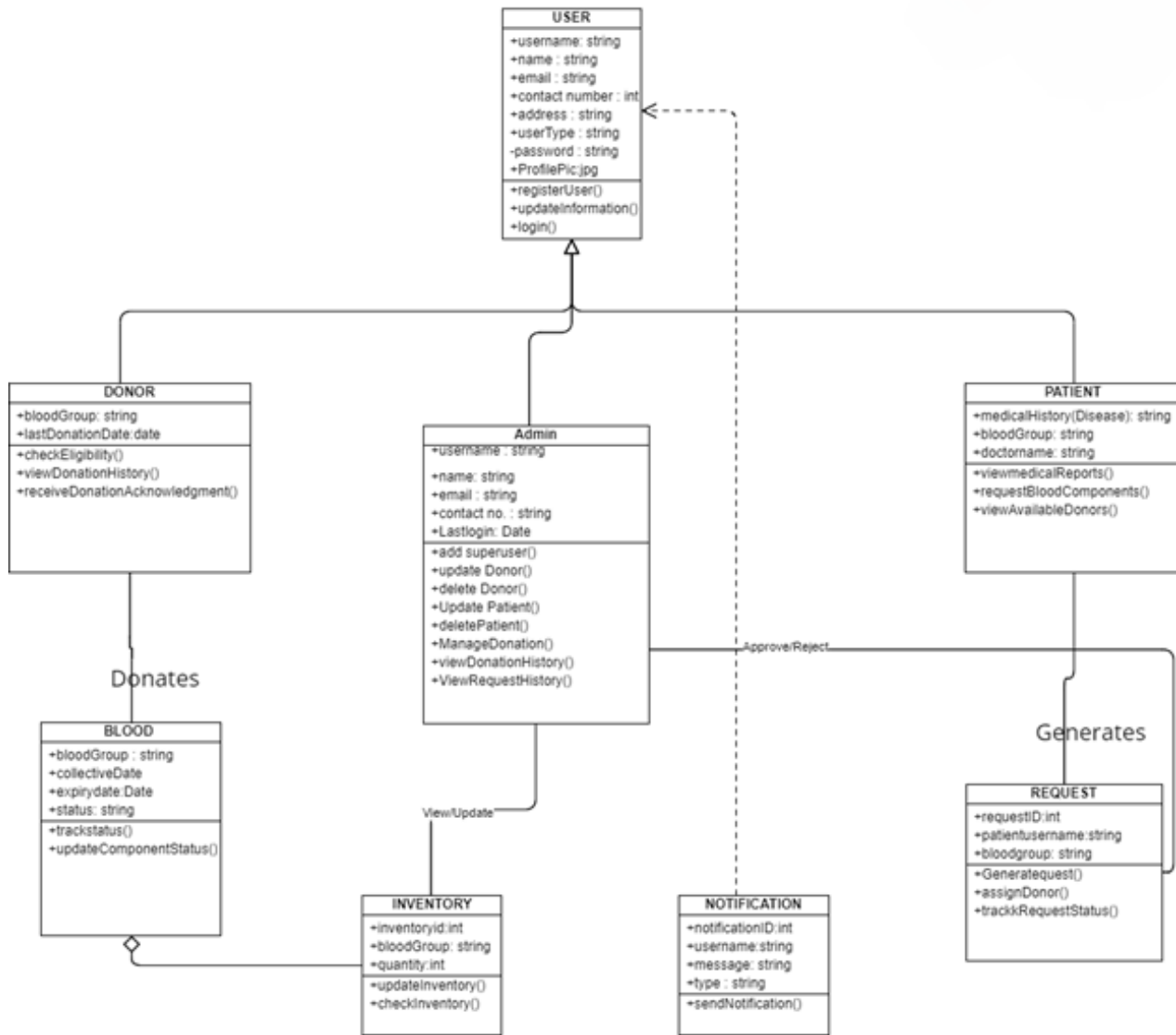
## 2. Use case Diagram:



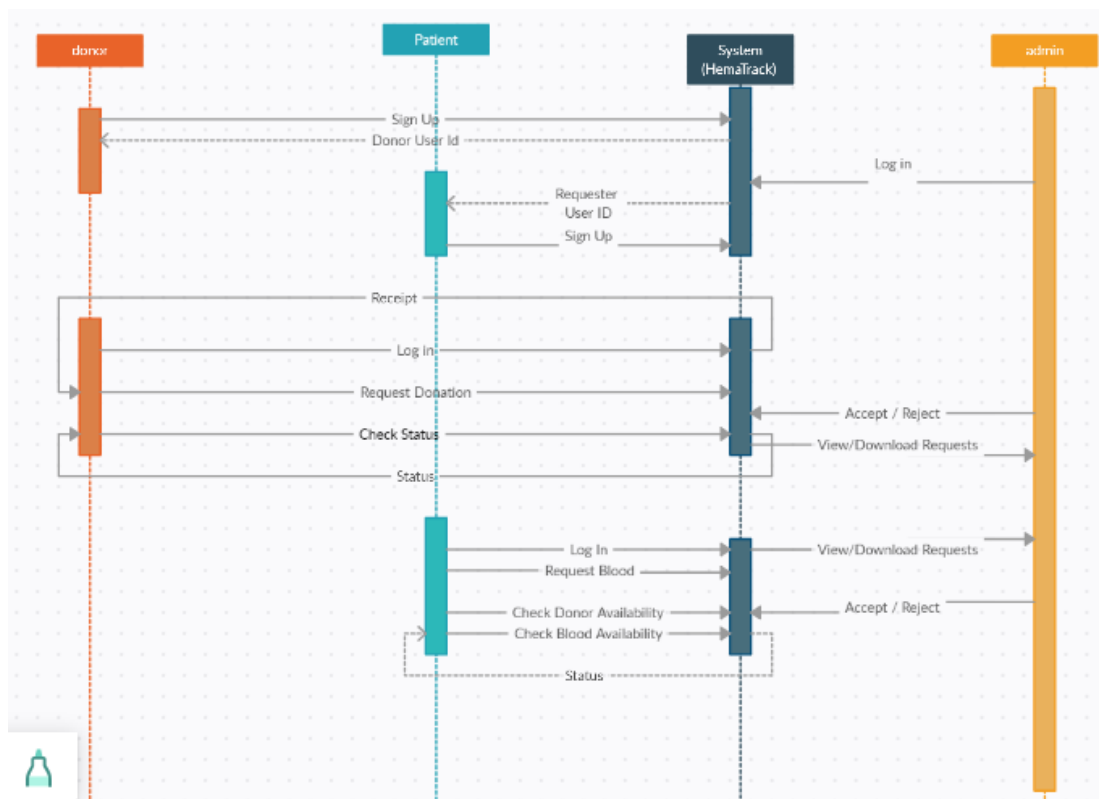
### Use case Diagram for Hematrack

### 3. Class Diagram:

ClassDiagram for HemaTrack



## 4. Sequence Diagram:



## Appendix C: To Be Determined List

1. TBD: Define specific encryption algorithms and key lengths for securing user passwords.
2. TBD: Specify the version of the Django framework to be used for development.
3. TBD: Identify any additional external servers or APIs that HemaTrack will interact with and define communication protocols.
4. TBD: Determine the exact data transfer rates and synchronization mechanisms for email notifications.
5. TBD: Define any standard buttons and functions to appear on every screen in the administrative interface.
6. TBD: Determine the error message display standards for the mobile app interface.
7. TBD: Identify any specific GUI standards or product family style guides to be followed in the design of the web-based user interface.
8. TBD: Specify any keyboard shortcuts to be implemented for frequently used actions in the web-based user interface.
9. TBD: Determine the platform-specific design guidelines to be followed for the mobile app interface.
10. TBD: Identify any additional screen layout constraints for the administrative interface.
11. TBD: Define any communication security or encryption issues for interactions with external servers or APIs.