Software Requirement Specification(SRS) for

Sahayak -For the people by the People

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of an Online Blood Management System. This system aims to facilitate the management and distribution of blood resources efficiently by connecting blood donors, recipients, and blood banks through an online platform.

1.2 Scope

The Online Blood Management System will provide a comprehensive platform for:

Blood donors to register, schedule appointments, and donate blood.

Blood banks to manage inventory, requests, and donations.

Recipients to search for and request blood units as per their requirements.

Admins to oversee the entire system, manage users, and ensure smooth operation.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

DB: Database

1. Overall Description

2.1 Product Perspective

The Online Blood Management System will be a standalone web-based application interacting with users through a browser-based UI. It will integrate with existing databases of blood banks for inventory management and facilitate communication between donors, recipients, and blood banks.

2.2 Product Features

User Registration,Admin Registration and Authentication:

Users can register with the system as donors, recipients.

Admin can register BloodBanks,add the camps and show lists of bloodBanks..

Authentication mechanisms will ensure secure access to user accounts.

Donor Management:

Donors can create and manage their profiles, including personal details and blood type.

They can schedule donation appointments.

Blood Bank Management:

Blood banks can manage their inventory, including blood type, quantity, and expiration dates.

Admin can handle incoming blood donations, update inventory, and track usage.

Admin Panel:

Admins will have access to an administrative dashboard to manage users, blood bank accounts, and system settings.

They can monitor system activity and generate reports.

2.3 User Classes and Characteristics

Donors:

Individuals willing to donate blood.

Should have valid personal information and be medically eligible to donate.

Recipients:

Individuals in need of blood transfusions.

Should have valid medical requirements for blood.

Admins:

System administrators overseeing the entire platform.

Should have access to all features and system settings.

2.4 Operating Environment

The system will be deployed on a web server with support for modern web technologies. It should be accessible via popular web browsers such as Google Chrome, Mozilla Firefox, and Safari.

2.5 Design and Implementation Constraints

The system should comply with relevant data protection regulations, ensuring the privacy and security of user data.

It should be designed with scalability in mind to accommodate potential growth in user base and data volume.

3. Specific Requirements

3.1 External Interface Requirements

User Interface:

The UI should be intuitive and user-friendly, accessible from both desktop and mobile devices.

Different UI views for donors, recipients and admins.

APIs:

The system should expose APIs for integration with external systems, such as blood bank databases and notification services.

Database:

Interaction with the database should be efficient and secure.

Proper indexing and normalization to ensure data integrity and performance.

3.2 Functional Requirements

User Registration:

Users should be able to register with the system by providing necessary details.

Validation of user input to prevent incorrect or incomplete registration.

Authentication:

Donation Appointment:

Donors should be able to schedule appointments for blood donation.

Availability of time slots based on donor preferences and blood bank schedules.

Inventory Management:

Blood banks should be able to add, update, and remove blood units from inventory.

Request Management:

Recipients should be able to search for blood units based on type, location, and availability.

Submit requests for blood units with necessary details.

Admin Controls:

Admins should be able to manage user accounts, blood bank profiles, and system settings.

Access to system logs and audit trails for monitoring.

3.3 Non-functional Requirements

Security:

Data encryption in transit and at rest.

Role-based access control to restrict unauthorized access.

Performance:

Fast response times for user interactions.

Scalability to handle concurrent users and large datasets.

Reliability:

High availability with minimal downtime.

Regular backups and disaster recovery mechanisms.

Usability:

Intuitive UI with clear navigation and informative error messages.

Accessibility features for users with disabilities.

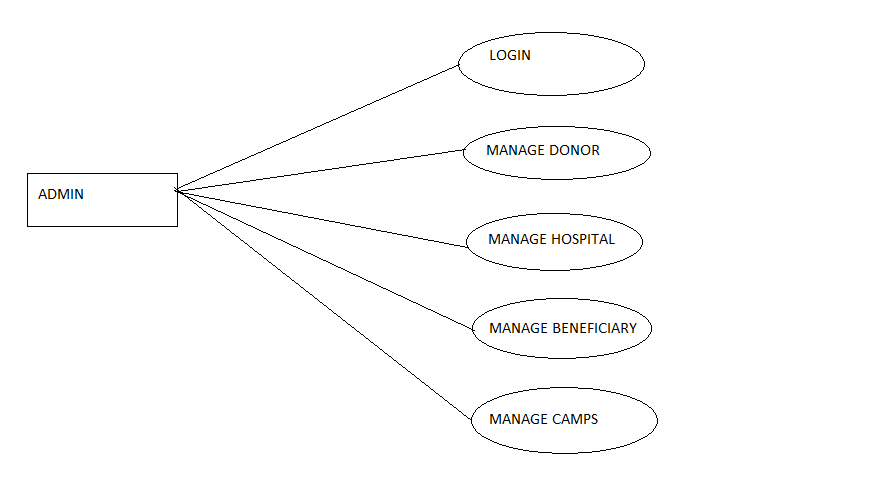
Compatibility:

Compatibility with major web browsers and devices.

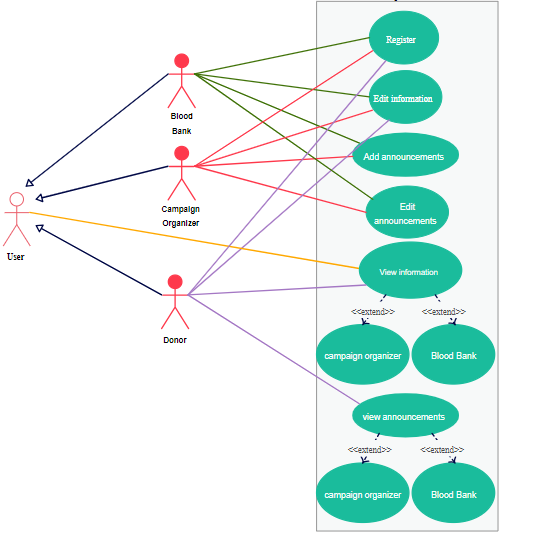
Responsive design for seamless user experience across different screen sizes.

**Data Flow Diagrams:**

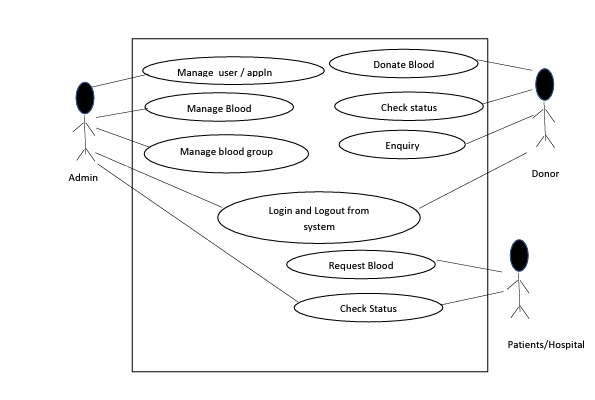
**Admin**



**For User:**

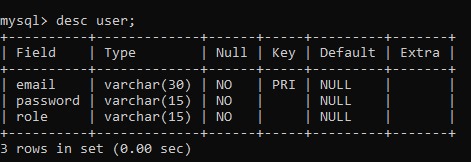


**Use Case Diagram:**

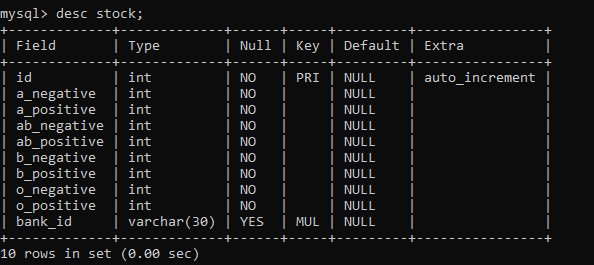


**Table Structure:**

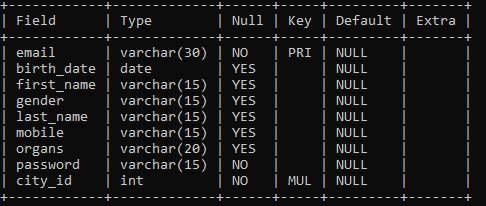
1. **User Table**

****

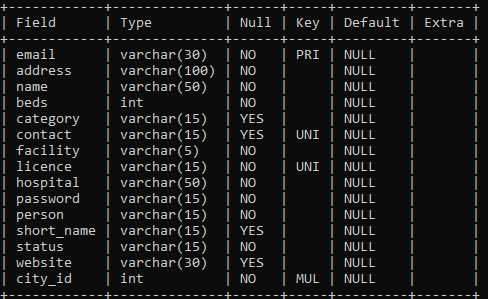
1. **Stock table:**



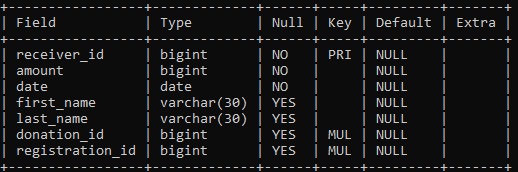
1. **Donor Table:**



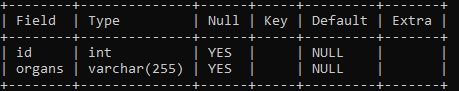
1. **Blood Bank Table**

****

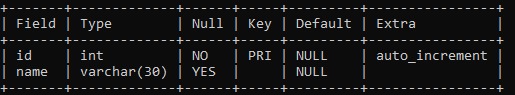
1. **Receiver table**

****

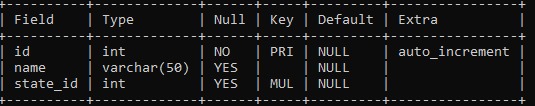
1. **Organ Table**

****

1. **State Table**

****

1. **District Table:**

****

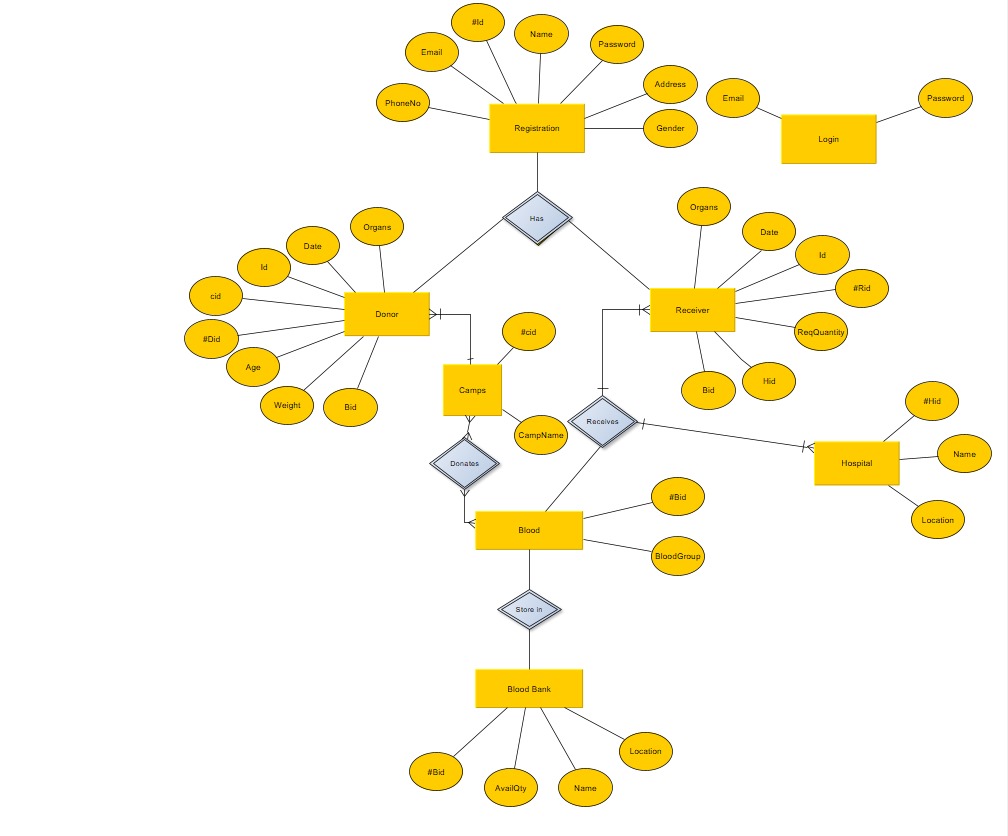
**ER DIAGRAM**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

.

ER Diagram



4. Appendices

4.1 Glossary

Donor: Individual willing to donate blood.

Recipient: Individual in need of blood transfusion.

Blood Bank: Facility responsible for collecting, storing, and distributing blood.

Admin: System administrator with full access to system features.

4.2 References

World Health Organization - Blood Transfusion Safety

American Red Cross - Blood Donation FAQs

This Software Requirements Specification document outlines the necessary features, constraints, and functionalities for the development of the Online Blood Management System. It serves as a guideline for the development team and stakeholders throughout the software development lifecycle.