**Software Requirements Specification (SRS)**  
**Blood Donation Management System  
Arpa Saha Barna  
1910628**

**1. Introduction**

**1.1 Purpose**

The purpose of the **Blood Donation Management System** is to create an online platform that connects blood donors with recipients in need. The system will facilitate donor registration, request management, and emergency notifications to streamline the blood donation process.

**1.2 Document Conventions**

This document follows standard **IEEE SRS formatting** and uses the following conventions:

* Functional requirements are prefixed with **FR-** (e.g., FR-1).
* Non-functional requirements are prefixed with **NFR-** (e.g., NFR-1).
* Tables and diagrams are used for clarity where applicable.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for:

* **Developers** to understand system requirements.
* **Testers** to validate functionality.
* **Project Managers** to track system progress.
* **End Users & Clients** to review system features.

Readers should focus on **Section 3 (System Features)** for key functionalities and **Section 5 (Nonfunctional Requirements)** for system constraints and security considerations.

**1.4 Project Scope**

The **Blood Donation Management System** aims to provide a user-friendly, efficient platform where donors can register and update their availability, and recipients can search for donors based on location and blood type. The system will also include an **admin panel** for request approvals and donor verification.

**1.5 References**

* IEEE Standard 830-1998 (Software Requirements Specification)
* PHP, MySQL, and JavaScript documentation
* Bootstrap/Tailwind CSS documentation for UI development

**2. Overall Description**

**2.1 Product Perspective**

The Blood Donation Management System is a **standalone web-based application** that functions as a centralized platform for blood donation activities.

**2.2 Product Features**

* User registration (Donor & Recipient)
* Search & filter donors by blood type and location
* Blood request submission
* Admin dashboard for donor verification and request approvals
* Emergency alerts & notifications
* Donation history tracking

**2.3 User Classes and Characteristics**

* **User (Donors** **& Recipients)** : Register, update profiles, and respond to requests, Search for donors and submit requests.
* **Admins**: Verify donors, manage requests, and monitor system activities.

**2.4 Operating Environment**

* **Frontend:** HTML, CSS (Bootstrap/Tailwind), JavaScript
* **Backend:** PHP (Vanilla)
* **Database:** MySQL
* **Browser Compatibility:** Chrome, Firefox, Edge

**2.5 Design and Implementation Constraints**

* Must use **PHP and MySQL** for backend.
* Must ensure data privacy and secure user authentication.

**2.6 User Documentation**

* **User Guide** (for donors and recipients)
* **Admin Manual** (for admin operations)

**2.7 Assumptions and Dependencies**

* Users must have an internet connection.
* The system will rely on **Google Maps API** (optional) for donor location services.

**3. System Features**

**3.1 User Registration & Authentication (FR-1)**

* Users can register and log in securely.
* Passwords will be encrypted.

**3.2 Donor Profile Management (FR-2)**

* Donors can update their availability and location.
* Option to temporarily deactivate profiles.

**3.3 Search & Filter Donors (FR-3)**

* Recipients can search for donors by blood type, location, and availability.

**3.4 Blood Request System (FR-4)**

* Recipients can submit a request specifying urgency and hospital details.
* Donors receive notifications of nearby requests.

**3.5 Admin Dashboard (FR-5)**

* Admins can verify donor profiles and approve/reject blood requests.
* Generate reports on donation trends.

**3.6 Donation History & Tracking (FR-7)**

* Users can track past donations and requests.

**4. External Interface Requirements**

**4.1 User Interfaces**

* **Donor Dashboard** (Profile update, availability status)
* **Recipient Dashboard** (Search for donors, request submission)
* **Admin Panel** (Manage users, requests, and system settings)

**4.2 Hardware Interfaces**

* The system requires a standard web server .

**4.3 Software Interfaces**

* **Google Maps API** (for donor location tracking, optional)

**4.4 Communications Interfaces**

* Secure HTTP (SSL/TLS) for encrypted data transfer.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements (NFR-1)**

* The system should handle **100+ concurrent users** without performance degradation.

**5.2 Safety Requirements (NFR-2)**

* Regular backups of the database should be maintained.

**5.3 Security Requirements (NFR-3)**

* Role-based access control (RBAC) for different user types.
* Data encryption for sensitive information.

**5.4 Software Quality Attributes (NFR-4)**

* **Maintainability**: Modular architecture for easy updates.
* **Scalability**: Ability to add new features with minimal disruption.

**6. Other Requirements**

* Support for multilingual functionality (optional future enhancement).

**Appendices**

**A. Glossary**

* **RBAC:** Role-Based Access Control
* **CRUD:** Create, Read, Update, Delete
* **SMTP:** Simple Mail Transfer Protocol

**B. Analysis Models**

* **Use Case Diagrams** (User interactions with the system)
* **Entity-Relationship Diagram (ERD)** (Database schema representation)

**C. Issues List**

* Pending decision on **Google Maps API** integration.
* Choice between **Laravel or Vanilla PHP** for backend.

This SRS document serves as a foundation for the development of the **Blood Donation Management System**, ensuring clarity and direction in the implementation process.