Experiment 2

Prepare an SRS document in line with the IEEE recommended standards for the specified Case Study. (Non-Functional Requirements.

Creating a blood donation system involves several components, including database design, user

interface, backend logic, and frontend design. Below is a simplified outline of how you can approach building a basic blood donation system using PHP, HTML, CSS.

1. **Introduction**

* **Purpose**: This document outlines the requirements for the development of a Blood Donation Management System.
* **Scope**: The system will facilitate the management of blood donation events, donor records, and recipient requests.

1. **Overall Description**

* **Product Perspective:**
* The system will be a standalone web-based application.
* **Product Functions:**
* User registration and login.
* Donor registration and profile management.
* Recipient request submission and management.
* Reporting and analytics for blood donation events.

1. **Specific Requirements**

* **Non-Functional Requirements**
* **Performance**
* The system should handle concurrent users efficiently.
* **Security**
* User data, including personal and medical information, must be securely stored and transmitted.
* Authentication and authorization mechanisms should be robust.
* **Usability**
* The user interface should be intuitive and user-friendly.
* **Compatibility**
* The system should be compatible with modern web browsers (e.g., Chrome, Firefox, Safari).

# Database Design (MySQL):

* + Create a database to store information about donors, recipients, and blood donations.
  + Design tables like `donors`, `recipients`, and `donations`.
  + Define appropriate fields for each table (e.g., donor\_name, donor\_email, blood\_type, etc.).

# Backend (PHP):

* + Establish a connection to the database.
  + Create PHP scripts to handle various functionalities:
  + Registering donors and recipients.
  + Logging in and managing user sessions.
  + Adding, updating, and deleting donor and recipient records.
  + Recording blood donations and updating donor information.
  + Retrieving data for display.

# User Interface (HTML/CSS):

* + Design web pages for different functionalities (e.g., registration, login, dashboard, donation form, etc.).
  + Use HTML for structuring the content and CSS for styling.
  + Ensure that the interface is user-friendly and responsive.

# Authentication and Authorization:

* + Implement user authentication to ensure that only registered users can access certain features.
  + Set up user roles (e.g., admin, donor, recipient) and manage permissions accordingly.

# Search and Filter Functionality:

* Implement search and filter options to allow users to find donors or recipients based on criteria like blood type, location, etc.

# Testing:

* + Conduct thorough testing of the application to identify and fix any bugs or issues.
  + Ensure that all features work as expected and that user input is handled appropriately.

# Deployment:

* + Choose a web hosting service and deploy your application.
  + Configure the server environment (e.g., PHP version, MySQL database setup).
  + Secure the server, set up SSL, and implement regular backups.

Remember that this is a high-level overview, and the actual implementation may involve more detailed steps and considerations depending on the complexity and specific requirements of your blood donation system. Additionally, always prioritize user privacy and data security in your

development process.