Software Requirements Specification

for

Blood Bank Management System

Version : 1.0

Group : 01

Abhishek Kumar	B220628CS	abhishek_b220628cs@nitc.ac.in
Abhay Kumar Bharti	B220088CS	abhay_b220088cs@nitc.ac.in
Adarsh Kharwar	B220107CS	adarsh_b220107cs@nitc.ac.in

Instructor	:Dr. Chandramani Chaudhary ,Dr. K. A . Abdul Nazeer
Course	: Database Management Systems
Date	: 30-10-2024

Table Of Contents

1. Introduction

1.1 Purpose

1.2 Product Scope

The scope of the Blood Bank Management System is to provide a comprehensive and efficient solution for managing blood bank operations. It encompasses the complete lifecycle of blood donation, inventory management, and blood request processing. The system allows for the registration of donors, tracking of donor information, recording of blood donations, and management of blood inventory, including blood type, quantity, and expiry dates.It also facilitates the processing of blood requests, ensuring timely and accurate responses.The system is intended to enhance the overall functionality of blood banks, improve blood inventory management, and streamline the process of blood collection, storage, and distribution, ultimately contributing to the provision of life-saving healthcare services.

1.3 Intended Audience and Document Overview

1.4 Definitions, Acronyms, and Abbreviations

This section aims to clarify and standardize the terminology used throughout the SRS, ensuring that all stakeholders have a common understanding of the key terms, acronyms, and abbreviations relevant to the project.

- **Donor**: An individual who voluntarily contributes blood to the blood bank.
- **BBMS**: Blood Bank Management System being developed to manage blood bank operations.
- **RDBMS**: Relational Database Management System, such as MySQL used to store and manage data.
- **SRS**: Software Requirements Specification, the document you are currently reading.
- **GUI**: Graphical User Interface, the visual components and interactions that users will employ to interact with the system.
- **API**: Application Programming Interface, used for integrating with external systems or data sources.

- **User**: Any person or entity interacting with the Blood Bank Management System, including donors, blood bank staff, and administrators.
- **Authentication**: The process of verifying the identity of users before granting access to the system.
- **Authorization**: The process of specifying what actions or resources users are allowed to access.
- **Blood Type**: A classification of blood based on the presence or absence of specific antigens on the surface of red blood cells.
- **Inventory**: The stock of blood and blood products available for distribution.
- **Data Dictionary**: A repository that defines the data elements used within the system, specifying their names, attributes, and relationships.

1.5 Document Conventions

- **Font**: The document uses Arial font with the following font sizes for various textual content:
- **Main Headings**: Font size 16
- **Subheadings**: Font size 14
- **Sub-subheadings and Paragraphs**: Font size 12

The SRS document aims to provide a clear and standardized presentation of information in accordance with the IEEE formatting requirements, ensuring readability and accessibility for all stakeholders involved in the Blood Bank Management System project.

1.6 References and Acknowledgements

- IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 6/e, Pearson Education, 2011

2. Overall Description

2.1 Product Overview

The Blood Bank Management System is a software solution designed to streamline blood bank operations, focusing on efficient blood inventory management, donor registration, and swift response to blood requests. It addresses the challenges of maintaining an accurate inventory, enabling timely and secure blood supply to healthcare facilities, and ensuring data confidentiality.

The system features a user-friendly web-based interface for administrators, staff, and donors, promoting engagement and user satisfaction. Its implementation promises improved healthcare services, enhanced donor participation, and reliable data security, ultimately contributing to efficient and effective healthcare delivery.

2.2 Product Overview

The product functionality of the Blood Bank Management System includes:

Donor Management: Allows for the registration, tracking, and management of donor information, ensuring a readily available pool of potential donors. **Blood Inventory Control**: Provides real-time monitoring of blood units, their types, quantities, and expiry dates, minimizing shortages and waste. **Blood Request Processing**: Facilitates efficient processing of blood requests, ensuring that the required blood type is promptly delivered to healthcare facilities. **Data Security**: Prioritizes data security through encryption, access control, and secure communication to protect sensitive donor and recipient information. **User-Friendly Interface**: Features an intuitive web-based interface for administrators, staff, and donors, enhancing user engagement and satisfaction. **Reporting**: Enables the generation of various reports, such as donor statistics and blood inventory status, for informed decision-making.
Extensibility: Designed to accommodate future enhancements, integrations, and scalability, ensuring adaptability to evolving requirements and technology.
Performance Monitoring: Provides tools for performance monitoring to proactively address potential issues and optimize system performance.

2.3 Design and Implementation Constraints

Design and implementation constraints for the Blood Bank Management System include:

Platform Compatibility: The system needs to work seamlessly on multiple platforms (Windows, macOS, Linux) to accommodate diverse user environments.
Reliability and Availability: The system must maintain a high level of reliability with at least 99.9% uptime, and scheduled maintenance activities should be communicated to users in advance.

Data Privacy Regulations: Compliance with data privacy regulations (e.g., GDPR, HIPAA) is essential to protect donor and recipient data, which imposes constraints on data handling and storage.

Resource Limitations: The system should be designed to operate efficiently with standard hardware and not require specific or high-end hardware components.

Scalability: The architecture must support scalability to accommodate a growing number of users, donors, and blood units without compromising performance.

Integration Compatibility: Integration with external systems or databases, such as RDBMS, should be seamless to ensure interoperability.

Legacy System Integration: In some cases, integration with existing legacy systems may be necessary, which poses constraints on data mapping and communication protocols.

Usability Constraints: The system's design should prioritize usability, providing an intuitive interface for users of varying technical proficiencies.

Regulatory Compliance: Compliance with healthcare regulations and standards is crucial for blood bank management, which may necessitate specific design considerations and constraints.

2.4 Assumptions and Dependencies

Donor Eligibility: The assumption that donors meet the necessary eligibility criteria for blood donation, including age, health, and medical history.

Reliable Internet Access: Dependence on a stable internet connection for system access and data transmission.

Availability of Blood Testing Facilities: Assuming the availability of laboratories for blood testing and screening to determine donor eligibility and ensure the safety of collected blood.

Regulatory Compliance: The system's functionality is dependent on compliance with local and national healthcare regulations, including those related to blood banking and data privacy.

Database Availability: Dependence on the continuous availability and reliability of the underlying database management system (e.g., MySQL) for data storage.

User Training: Assumption that users, including administrators and staff, receive adequate training to use the system effectively and securely.

External Integration: Depending on external systems and databases for the exchange of information, such as integrating with hospital or healthcare facility databases.
Donor Participation: The assumption that donors will actively engage with the system for registration, appointment scheduling, and donation.
IT Infrastructure: Dependency on a robust IT infrastructure, including servers, network equipment, and backup systems, to maintain system availability.
Budget and Resources: Availability of the necessary budget and resources for system development, maintenance, and regular updates.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Blood Bank Management System will have a user-friendly web-based interface. It will include the following components:

Dashboard: Provides an overview of blood inventory, recent donations, and pending requests. Allows quick access to key functionalities.
Donor Registration Form: A form to capture donor information including name, contact details, blood type, and eligibility criteria.
Donor Management Page: Enables administrators to view, edit, and delete donor records. Includes search and filter options.
Blood Inventory Display: Shows a detailed list of available blood units, including type, quantity, and expiry date.
Request Processing Interface: Allows staff to process blood requests, specifying the recipient details and required blood type.
Report Generation: Provides options to generate reports on donor statistics, blood inventory levels, and request fulfillment.

3.1.2 Hardware Interfaces

The Blood Bank Management System will be accessible through standard web browsers (e.g., Chrome, Firefox) on devices such as desktops, laptops, tablets, and smartphones. It does not require any specific hardware components beyond a stable internet connection.

3.1.3 Software Interfaces

The system will interact with the following software components:

Database Management System (DBMS): Utilizes MySQL for efficient storage and retrieval of donor, inventory, and request data.

Web Server: Runs on Apache to host the web application and handle client-server interactions.
Operating System: Compatible with Windows, macOS, and Linux environments.
Programming Languages: Developed using HTML, CSS, JavaScript for the front-end, and PHP for server-side processing.

3.2 Functional Requirements

3.2.1 F1: Create Donor

The system shall allow administrators to create a new donor record. Details including name, contact information, blood type, and eligibility criteria will be captured. Upon submission, the information will be stored in the database.

3.2.2 F2: Role-Based Login

Donor: Registered donors can log in with their credentials.
Patient: Registered patients can log in with their credentials.
Administrator: Registered administrators can log in with their credentials.

This revision emphasizes role-based logins for donors, patients, and administrators.

3.2.3 F3: Invalid Login

The system shall not permit access for users with invalid login credentials. An 'Invalid Credentials' error message will be displayed, and the user will be redirected to the login page.

3.2.4 F4: Display Donor Information and Patient Information

The system will display detailed information about registered donors and patients, including their contact details, blood type, and eligibility status.

3.2.5 F5: Donor Management

Administrators can view, edit, or delete donor records. They will have the ability to search for specific donors and apply filters for easy access.

3.2.6 F6: Display Blood Inventory

The system will provide a comprehensive list of available blood units, indicating blood type, quantity in stock, and expiry date.

3.2.7 F7: Blood Request Processing

Authorized staff will be able to process blood requests, specifying recipient details and the required blood type. The system will track pending and fulfilled requests.

3.2.8 F8: Update Blood Inventory

As new donations are received, the system will update the blood inventory, reflecting the addition of new units along with their respective details.

3.3 Use Case Model

3.3.1 Use Case #1 (Login - U1)

Author: Abhishek kumar

Purpose: To allow registered users to log in to the system.

Requirements Traceability: F2, F3

Priority: High

Preconditions:

1.The user must be a registered donor.
2.The user must have valid login credentials.

Post Conditions:

1.The user successfully logs into the Blood Bank Management System.

Actors: Donor

Extends: None

Flow of Events:

1.The donor enters their login credentials.
2.The system verifies the provided information.
3.If the credentials are valid, the user is logged in.
4.If the credentials are invalid, an error message is displayed, and the user is redirected to the login page.

Includes: None

Notes/Issues: None

3.3.2 Use Case #2 (Create Donor - U2)

Author: Abhishek kumar

Purpose: To allow administrators to create a new donor record.

Requirements Traceability: F1

Priority: High

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.

Post Conditions:

1.A new donor record is successfully created.
2.The donor's information is saved in the system's database.

Actors: Administrator

Extends: None

Flow of Events:

1.The administrator navigates to the "Create Donor" section.
2.The administrator enters the donor's information including name, contact details, blood type, and eligibility criteria.
3.The administrator submits the form.
4.The system validates and stores the provided information in the database.

Includes: None

Notes/Issues: None

3.3.3 Use Case #3 (Display Donor Information - U3)

Author: Abhishek kumar

Purpose: To display detailed information about registered donors.

Requirements Traceability: F4

Priority: Medium

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.

Post Conditions:

1.The administrator can view a list of all registered donors along with their information.

Actors: Administrator

Extends: U4

Flow of Events:

1.The administrator logs in to the system.
2.The administrator navigates to the "Display Donor Information" section.
3.The system retrieves and displays a list of registered donors along with their contact details, blood type, and eligibility status.

Includes: U4

Notes/Issues: None

3.3.4 Use Case #4 (Donor Management - U4)

Author: Abhay kumar

Purpose: To allow administrators to manage donor records, including adding, editing, or deleting donor information.

Requirements Traceability: F5

Priority: High

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.

Post Conditions:

1.New donor records can be added to the system's database.
2.Existing donor records can be edited or deleted.
3.The database is updated accordingly.

Actors: Administrator

Extends: None

Flow of Events:

1.The administrator logs in to the system.
2.The administrator navigates to the "Donor Management" section.
3.The administrator selects the desired action (add, edit, or delete).
4.The system processes the action and updates the database.

Includes: U3

Notes/Issues: None

3.3.5 Use Case #5 (Display Blood Inventory - U5)

Author: Adarsh kharwar

Purpose: To provide a comprehensive list of available blood units, indicating blood type, quantity in stock, and expiry date.

Requirements Traceability: F6

Priority: Medium

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.

Post Conditions:

1.The administrator can view a detailed list of available blood units.

Actors: Administrator

Extends: U6

Flow of Events:

1.The administrator logs in to the system.
2.The administrator navigates to the "Display Blood Inventory" section.
3.The system retrieves and displays a list of available blood units along with their details.

Includes: None

Notes/Issues: None

3.3.6 Use Case #6 (Blood Unit Management - U6)

Author: Adarsh kharwar

Purpose: To allow administrators to manage blood units, including adding, updating, or deleting blood unit information.

Requirements Traceability: F7

Priority: High

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.

Post Conditions:

1.New blood units can be added to the system's inventory.
2.Existing blood units can be edited or deleted.
3.The database is updated accordingly.

Actors: Administrator

Extends: None

Flow of Events:

1.The administrator logs in to the system.
2.The administrator navigates to the "Blood Unit Management" section.
3.The administrator selects the desired action (add, edit, or delete).
4.The system processes the action and updates the database.

Includes: U5

Notes/Issues: None

3.3.7 Use Case #7 (Process Blood Request - U7)

Author: Adarsh kharwar

Purpose: To facilitate the processing of blood requests from medical facilities.

Requirements Traceability: F8, F9, F10

Priority: High

Preconditions:

1.The administrator must be logged in to the Blood Bank Management System.
2.A blood request must have been received.

Post Conditions:

1.The status of the blood request is updated (delivered or not delivered).
2.The quantity of blood units delivered is recorded.

Actors: Administrator

Extends: None

Flow of Events:

1.The administrator logs in to the system.
2.The administrator navigates to the "Process Blood Request" section.
3.The administrator selects the appropriate blood request.
4.The system updates the status of the request and records the quantity delivered.

Includes: U8, U9

Notes/Issues: None

4. Other Non-functional Requirements

4.1 Performance Requirements

To avoid any issues, the DBMS software produced should be able to work efficiently to provide information when needed and to store data without any latency. Among the many aspects that influence performance, the system resources must be adequate and meet the baseline requirements for the software to execute smoothly.

To avoid problems when a request comes in, the performance should be accurate and the response time should be as short as possible.To avoid data loss when a server fails or a corrupted file causes a total data loss, the data should be backed up as log files on a regular basis.To be more effective, the DBMS should be able to manage a huge amount of data and conduct activities in less time.To connect to the software, the password and username will be matched to the password and name kept in the database, allowing only authenticated users to login.

4.2 Safety and Security Requirements

For reasons of safety and security, user information will be kept private and will not be shared with any other third-party organizations, ensuring that user privacy and information are protected. The data is backed up on a regular basis to ensure that it is not lost in the event of a database crash or other data loss event. The data is also saved on a private storage system,which means it cannot be viewed from the outside.

For security reasons, the database is secured, and system users have varied restrictions on accessing it.Users cannot update the database; only administrators are permitted to do so.Admins and users should have distinct accounts so that only admins can make changes to the database.

4.3 Software Quality Attributes

Adaptability - This developed DBMS software is adaptable by any organization.

Availability - The availability of the software is easy and for everyone.

Correctness - The results of the function are pure and accurate.

Flexibility - The operation may be flexible and reports can be presented in many ways.

Maintainability - After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

Portability - The software can be deployed at any machine.

Reliability - The performance of the software is better which will increase the reliability of the software.

Reusability - The data and record that are saved in the database can be reused if needed.

Robustness- If there is any error in any window or module then it does not affect the remaining part of the software.

Usability- To perform any operations and to understand the functioning of software is very easy.

Productivity- This software will produce every desired result with accuracy.

Timelines- The time limit is very important. It will save much time and provide fast access.

5. Hardware Requirements

- Processor: 1GHz
- RAM: 512 MB
- Storage: 1 GB or higher
- Internet connectivity for communication interfaces.

6. Software Requirements

- Operating System: Compatible with Windows, Linux, and MacOS.
- RDBMS: MySQL or equivalent.
- Web Browser: Latest versions of Google Chrome, Mozilla Firefox, and Safari.

APPENDIX - A DATA DICTIONARY

- **HTML** : HYPER TEXT MARKUP LANGUAGE.
- **CSS** : CASCADING STYLE SHEETS.
- **BOOTSTRAP** : OPEN-SOURCE CSS FRAMEWORK.
- **JS** : JAVA SCRIPT.
- **PHP** : HYPERTEXT PREPROCESSOR.
- **MYSQL :** OPEN-SOURCE RELATIONAL DATABASE MANAGEMENT SYSTEM.
- **SQL** : STANDARD QUERY LANGUAGE FOR RDBMS.

APPENDIX - B GROUP LOG

Work

● Adarsh Kharwar	-	Introduction
● Abhay kumar bharti	-	Overall Description
● Abhishek Kumar	-	Specific Requirements , Hardware and Software Requirements