

# **SOFTWARE REQUIREMENTS SPECIFICATION**

for

## **Blood Donation Management System**

Version 1.0 approved

Prepared by

Felix R-29

Bijo Thomas-21

Ben Baby-20

Jaidon Gill Shajan-39

Aldrin Immanuel P.S-08

**TABLE OF CONTENTS**

TABLE OF CONTENTS

REVISION HISTORY

## **1. INTRODUCTION**

- 1.1 Purpose
- 1.2 Document conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 Definitions, acronyms, and abbreviations
- 1.6 References

## **2. OVERALL DESCRIPTION**

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User characteristics
- 2.4 Constraints
- 2.5 Assumptions and dependencies

## **3. EXTERNAL INTERFACE REQUIREMENTS**

- 3.1 User interfaces
- 3.2 Hardware interfaces
- 3.3 Software interfaces
- 3.4 Communication interfaces

## **4. SYSTEM FEATURES**

- 4.1 Donor database

## **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

Appendix A: Glossary

Appendix B: Group Log

## **INTRODUCTION**

## **1.1 Purpose**

- The purpose of this document is to specify the requirements for a Blood Donation Management System that will be developed using HTML, CSS, PHP, and MySQL. The system will be used to manage the collection, processing, and distribution of blood donations.

## **1.2 Document conventions**

- This document was created based on the IEEE template for System Requirement Specification Documents

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

- This SRS document is intended for programming developers, product testers, project managers, and marketing staff; and our stakeholders may review it as a guideline for understanding the requirements and maintaining the main goal of the application. By reading this document, the reader will get a general understanding of the product including the application's functional and non-functional requirements.

## **1.4 Product Scope**

- The scope of this system includes the management of donor information, scheduling of appointments, processing of donated blood, and distribution of blood products to hospitals and other medical facilities. The system will be accessed via a web browser.

## **1.5 Definitions, acronyms, and abbreviations**

- Blood donation: The process of donating a unit of blood for use by others.
- Blood product: A component of blood that has been processed for use, such as red blood cells, plasma, or platelets.
- Donor: An individual who has undergone the process of donating blood.
- Recipient: An individual who receives a blood product.

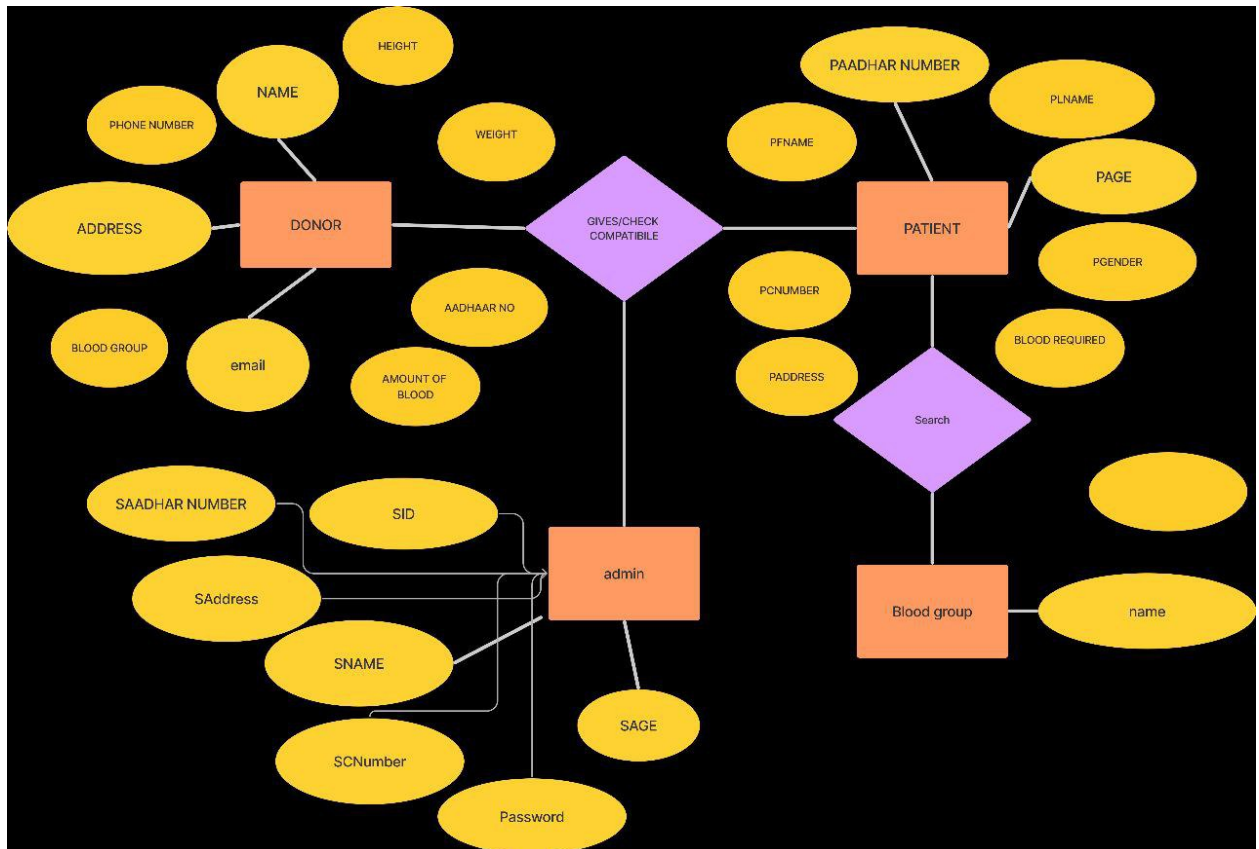
## **1.6 References**

- <https://docs.python.org/3/library/tkinter.html>
- <https://www.python-course.eu/>
- <https://stackoverflow.com/questions>
- <https://www.geeksforgeeks.org/python-gui-tkinter/>
- <https://www.javatpoint.com/sqlite-tutorial>
- <https://www.sqlite.org/index.html>

# **OVERALL DESCRIPTION**

## **2.1 Product perspective**

- The blood donation management system will be used by staff at blood collection centers, hospitals, and other medical facilities to manage the collection and distribution of blood donations. It will be accessed via a web browser.



## **2.2 Product functions**

The system will allow users to perform the following functions:

- Register and authenticate donors
- Schedule appointments for blood donations
- Process donated blood and create blood products
- Distribute blood products to hospitals and other medical facilities
- Track the location and expiration date of blood products
- Generate reports and analytics on the supply and demand for blood products

## **2.3 User characteristics**

- Users of the system will include staff at blood collection centres, hospitals, and other medical facilities. They will have varying levels of computer literacy and will be responsible for managing the collection and distribution of blood donations.

## **2.4 Constraints**

- The system must comply with all relevant regulations and laws regarding the collection and distribution of blood donations. It must also be compatible with modern web browsers and have a responsive design for use on mobile devices.

## **2.5 Assumptions and dependencies**

- It is assumed that users of the system will have access to a computer with a web browser and an internet connection. The system will be developed using HTML, CSS, PHP, and MySQL.

## EXTERNAL INTERFACE REQUIREMENTS

### **3.1 User interfaces**

- The system will have a web-based interface that is accessible through a modern web browser. The user interface will be intuitive and easy to use, with clear instructions and prompts for users. It will also have a responsive design for use on mobile devices.

### **3.2 Hardware interfaces**

- The system will be accessed via a web browser and will require an internet connection.

### **3.3 Software interfaces**

- The system may integrate with other hospitals or healthcare systems as needed, such as for recording donations in a donor's medical record or for identifying potential recipients of donated blood. Any such integration will be done using standard web technologies (e.g. APIs) and will be thoroughly tested.

### **3.4 Communication interfaces**

- The system will have a secure login system for authorized personnel to access the system. It will also be able to send notifications and alerts to users, such as appointment reminders or alerts about low blood supply levels. These notifications will be sent via email or through a messaging system within the application.

## SYSTEM FEATURES

### **4.1 Donor database**

- Donor database: The system will maintain a database of donors using MySQL, including contact information, medical history, and previous donation history. Donors will be able to register and authenticate themselves through the system using PHP.
- Appointment scheduling: The system will allow users to schedule appointments for blood donations using PHP, either online or through a call centre. It will send appointment reminders to donors using PHP and allow users to cancel or reschedule appointments as needed.
- Blood processing: The system will include a system for processing donated blood using PHP and MySQL, including screening for infections and separating the blood into components such as red blood

## **OTHER NONFUNCTIONAL REQUIREMENTS**

### **5.1 Performance Requirements**

- **Speed:** The system should be able to process transactions quickly, with minimal delays for hospital staff.
- **Reliability:** The system should be available for use at all times, with minimal downtime for maintenance or updates.

- User-friendliness: The system should be easy to use for hospital staff, with clear instructions and intuitive navigation.
- Customization: The system should allow the receiver to customize various features and settings to suit their specific needs.
- Scalability: The system should be able to handle a large volume of data, especially during disaster situations.

## **5.2 Safety Requirements**

- Physical security: The system should be installed in a secure location, with measures in place to prevent tampering or unauthorized access to the hardware and software.
- Disaster recovery: The system should have a plan in place to ensure that data and operations can be recovered in the event of a disaster such as a power outage, hardware failure, or natural disaster.
- Compliance: The system should be compliant with any relevant laws and regulations, such as data protection laws and industry-specific standards.
- Training hospital staff should be provided with training on how to use the system safely and securely. This can include instructions on how to handle sensitive data, how to identify and report potential security threats, and how to respond to system failures or errors.

## **5.3 Security Requirements**

- Only legitimate users are allowed to use the application using authentication
- Will ensure privacy and user safety.

## **5.4 Software Quality Attributes**

- Reliability
- The system should be reliable and able to handle large amounts of data without crashing or experiencing errors.
- Performance: The system should be fast and responsive, allowing hospital employees to quickly access and update information as needed.
- Ease of use: The system should be intuitive and easy to use, with a clear, user-friendly interface.
- Maintainability: The system should be easy to maintain, with regular updates and the ability to fix bugs and issues as they arise.
- Scalability: The system should be able to handle an increasing number of donor entries and customers as the hospital grows.

## **5.5 Business Rules**

All the rules for the design and development of the system will be under the jurisdiction.

## **Appendix A: Glossary**

- UI: User Interface
- GUI: Graphical User Interface

## **Appendix B: Group Log**

A group of four people was made to prepare the Project. A topic was decided upon as the Blood Donation Management System. The topic and the group members' names were then given to

the Instructor. All four members then read the example srs document provided and searched for the requisite documentation for the system. The Problem description, the purpose and the scope of the document were finalized. Each section was divided among the four members and each member was given the task to complete their sections with proper theory. The entire document was then clubbed to form the SRS for the Blood Donation Management System.