|  |
| --- |
|  |
| SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT ON |
| BLOOD BANK MANAGEMENT SYSTEM |

|  |
| --- |
| By Nzoalefack Syvera  17/01/2024 |

Contents

[1. Introduction 2](#_Toc188014003)

[1.1 Purpose of the project 2](#_Toc188014004)

[1.2 Scope of the document 2](#_Toc188014005)

[2. Overview of BBMS 2](#_Toc188014006)

[3. General Description 2](#_Toc188014007)

[4. Functional Requirements 3](#_Toc188014008)

[4.1 Login 3](#_Toc188014009)

[4.2 Donor Profile 4](#_Toc188014010)

[4.3 Blood Stock Management 4](#_Toc188014011)

[4.4 Donor/Recipient Management 4](#_Toc188014012)

[4.5 Report/History 4](#_Toc188014013)

[5. Non Functional requirements 4](#_Toc188014014)

[6. Interface Requirements 5](#_Toc188014015)

[7. Performance Requirements 5](#_Toc188014016)

[8. DESIGN CONSRTRAINTS 6](#_Toc188014017)

[9. PROJECT TIMEELINE AND SCHEDULE 6](#_Toc188014018)

# Introduction

A blood bank is a centralized place that stores blood for its distribution. It collects blood from donors and other sources and provides blood to needy people or hospitals. There would be various processes and tasks in a blood bank between the collection and distribution of blood. It is designed for handling those tasks and processes effectively and easily.

## **1.1 Purpose of the project**

The purpose of the blood bank management system is to simplify and automate the process of searching for blood in case of emergency and maintain the records of blood donors, recipient, blood donation information and blood stocks.

## **1.2 Scope of the document**

This system will facilitate the management of blood inventory, donor information, recipient information, and blood donation drives. It aims to streamline the operations of a blood bank by providing an efficient and reliable platform.

# Overview of BBMS

The blood bank management system is a web-based application. It provides an interface and powerful tools to simplify the process of connecting recipient to blood donors with opportunity across hospitals and the public.

The system will be build using HTML, CSS, JS, PHP and other web frameworks.

# General Description

* **Objectives**
* To develop a system that makes management of blood banks easy and to provide a platform for the donors and receivers.
* To improve the efficiency of bloodstock management by alerting the blood bank staff when the blood quantity is below its par level or when the bloodstock is expired.
* To check the availability of blood anytime
* To store a proper computerized database of blood, donors, and receiver.
* To give quick responses to the patients when they need the blood in emergency cases.
* **Key Features**
* Login interface.

Users should enter the valid username and password to get access to its profile.

* Donor Profile.

User will be able to see its account No, The receipt of the blood donation in the bank, Donation of to the bank, Need of blood to the bank, and Request

* Blood Stock Management

It will show the blood detailed of the specific bottle with it full donor detail or account No. if he/she requested to the bank report.

It will be available on the Admin’s profile and will show the ability of blood groups with its number of available bottle as per admin’s choice to view the report as day, month, or year.

* Patient Management
* Blood Request Management
* Donation History
* Make Blood Request
* **User Characteristics**
* Donors: Individuals registering to donate blood.
* Recipients: Patients or hospitals requesting blood.
* Administrators: Blood bank staff managing inventory and operations.
* **Benefits of the system**
* Automation of data entry, tracking, and reporting reduces the likelihood of human errors.
* Efficient matching of donors and recipients based on blood type and availability.
* An intuitive and easy-to-navigate interface enhances the user experience for administrators, medical staff, and donors.
* Automation reduces the need for manual data entry and administrative tasks, resulting in cost savings.

# Functional Requirements

## 4.1 Login

**User Input**

The system provides security features through username and password where only authorized user can access the system.

**Output**

Register himself, search blood Donor, Request for blood Donation, check bloodstock, Request for blood and logout

Admin

Input: -username, password

Output: - invalid or update blood details, logout

## 4.2 Donor Profile

This allow healthy public to register as volunteer donor

## 4.3 Blood Stock Management

The blood bank staffs can manage the blood stock starting from the blood collection, to blood screening, storage, and transfusion through the system, each process or work can be traced from the database

The system will also raise alert to the staff whenever the blood quantity is below it’s per level or when the blood in stock has expire.

## 4.4 Donor/Recipient Management

The record of all donors/recipient and their history are kept in one centralized database and thus reducing duplicate data in the database. The record of donation is maintained by the system.

## 4.5 Report/History

The system is able to generate reports such as the list of donors, staffs, the blood quantity in the bank.

# Non Functional requirements

**Availability**

* The system should be available at all times, meaning the user can access it using application.
* In case of a hardware failure or corruption in the database, backups of the database should be retrieved from the application folder and saved by the administrator.

**Security**

* Implement data encryption for sensitive information.
* Use role-based access control for user authorization.
* Ensure compliance with data protection regulations.

**Performance**

* The system is interactive and the delays involved are less

**Reliability**

* As the system provide the right tools for problem solving it is made in such a way that the system is reliable in its operations and for securing the sensitive details.

# Interface Requirements

* + **User Interface:**

Front end: HTML, CSS, JS

Back-end: PHP, Database

* + **Hardware Interface**: The system will be compatible with a variety of hardware platforms, including Windows, MacOS and Linux,

# Performance Requirements

Performance requirements define how well the system should perform under various conditions. Here are some key performance requirements:

**Response Time:**

* + The system should respond to user actions within 2 seconds for most operations.
  + For complex queries and reports, the response time should not exceed 5 seconds.

**Throughput:**

* + The system should be able to handle a high volume of transactions, including concurrent user sessions, data processing, and report generation.

**Reliability:**

* + The system should be up to time, ensuring availability for critical operations.
  + It must include failover mechanisms and redundancy to minimize downtime and data loss.

1. DESIGN CONSRTRAINTS

The Graphical User Interface (GUI) will only be in English

1. PROJECT TIMEELINE AND SCHEDULE

**Phase 1: Project Planning and Requirement Gathering**

Duration: 1 Week

Activities

* Review and finalize the requirements
* Design the database schema and user interface
* Create a detailed system design document

**Phase 2: Development and Implementation**

Duration: 2 Week

Activities:

* Frontend development using HTML, CSS, and JavaScript
* Implement the server-side using PHP, MySQL
* Create the client-side logic for interacting with the serve
* Connect to the database and implement
* Test and debug the front-end and back-end code

**Phase 3: Integration and Testing**

Duration: 1 week

Activities:

* Integrate the front-end and back-end code
* Test the entire system for functionality and performance
* Identify and fix any bugs or issues

**Phase 4: Deployment and Launch**

Duration: 1 Week

Activities:

* Set up cloud hosting
* Final system check and launch

Duration: 1 Week

**Maintenance**

Duration: Ongoing after launch

Activities:

* Regular updates and bugs fix
* Configure the system for security