**Project Document: Blood Management System**

**Table of Contents**

1. Introduction

1.1. Purpose of the Document

1.2. Project Overview

1.3. Scope

2. System Requirements

2.1. Functional Requirements

3. Architecture

3.1. Activity Diagram

3.2. Sequence Diagrams

4. User Interface

5. Technologies Used

6. Testing

6.1. Test Cases

6.2. Unit Testing

7. Conclusion

8. References

**1. Introduction**

**1.1. Purpose of the Document**

This document aims to offer an outline of the Blood Management System project, which was created using Core Java. It delineates the project's requisites, framework, functionalities, and text-based user interface.

**1.2. Project Overview**

The Blood Management system constitutes a text-based application designed for administrative control over the organization's donor and receiver database. Admin-initiated posts manage donor and receiver requests, enabling receivers to access donor information either directly or through the admin. This project effectively oversees diverse records of blood contributions from various donors.

**1.3. Scope**

The scope of the Blood Management System project includes the following functionalities:

**- Login credential for Admin –** Only authenticated admin will be able to add,update or delete the Donor or Receiver data.

- **Donor Portal –** In Donor portal admin will be able to add , update or delete the donor details that are present in the database .

- **Receiver portal –** In Receiver portal admin will be able to see all the request ;

- **Report –** Report will generate all the records ( Donor and Receiver ) present in the Database;

**2. System Requirements**

**2.1. Functional Requirements**

1. Login Credential: The Admin should be able to login with the username and password provided by the management.

2. Donor portal:

2.1 Add the donor details

2.2 Delete the donor details

3.3 Update the donor details

3. Receiver portal:

3.1 Add the receiver details

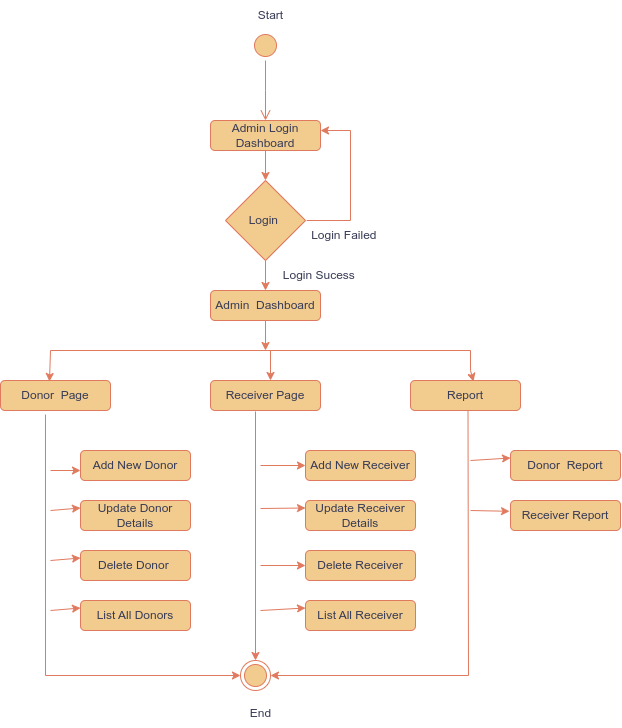
3.2 Delete the receiver details

3.3 Update the receiver details

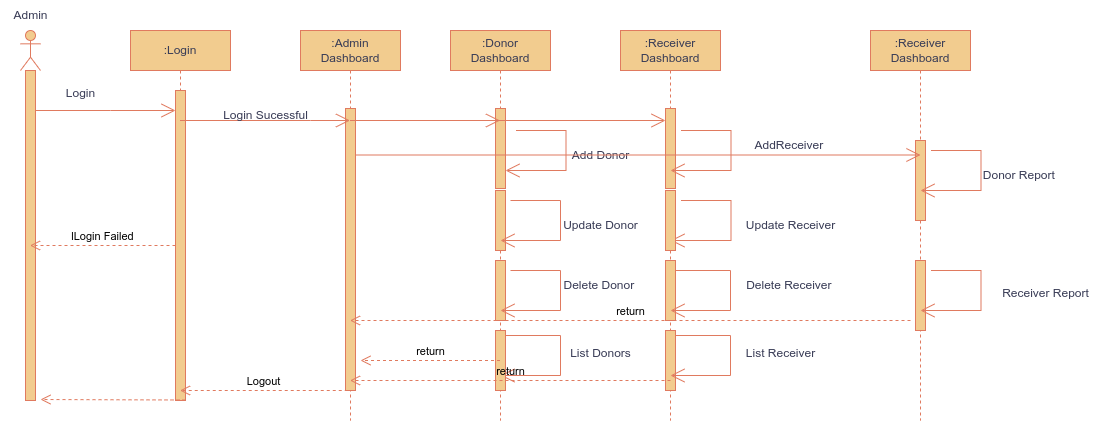
4. Report: Admin should generate the report consisting of total blood available for each blood group.

**3. Architecture**

**3.1. Activity Diagram**



**3.2. Sequence Diagrams**



**4. User Interface**

The admin interface operates through a console-based system. Within this interface, the admin is presented with menu-driven choices, and upon selecting a specific menu, corresponding operations are executed. As there is no graphical user interface (GUI) provided, the admin engages with the database solely through the console.

**5. Technologies Used**

1. For storing the data MySQL is used in local host with default port number 3306 .

2. MySql workbench is used to for database visualization (as a GUI interface for Database);

3. All business logics are written in Java and to establish connection with mysql dependency is included . This dependency provides built-in methods for easily connection with the mysql database.

4. intellij IDEA is used for project .

**6. Testing**

Manual Testing is done for all the operations performed by the Admin.

**7. Conclusion**

Developing a Java and SQL-powered, console-based Blood Management System results in a refined approach to regulating blood inventory. The Java interface within the system empowers healthcare experts to handle donations, monitor blood types, and supervise distribution seamlessly. Through the amalgamation of an SQL database, data precision is guaranteed, facilitating effective reporting and analysis. This system, characterized by its lightweight nature and accessibility, lays the groundwork for potential enhancements in terms of user-friendliness and expanded features. Ultimately, this system exemplifies the significance of adept blood inventory management.

**8. References**

1. <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

2. <https://docs.oracle.com/cd/E17952_01/index.html>