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 Class Diagram
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

The purpose of this document is to provide an overview of the Blood Management System project developed using Core Java. It outlines the system's requirements, architecture, features, console-based user interface.

1.2. Project Overview

The Blood Management system is a console-based application that allows the admin to manage database of donor and receiver present in organization. Donor and Receiver requests are posted by the admin and Receiver can get the Donor details through admin or directly. This project manages various records of the blood donated by different 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 toadd, 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

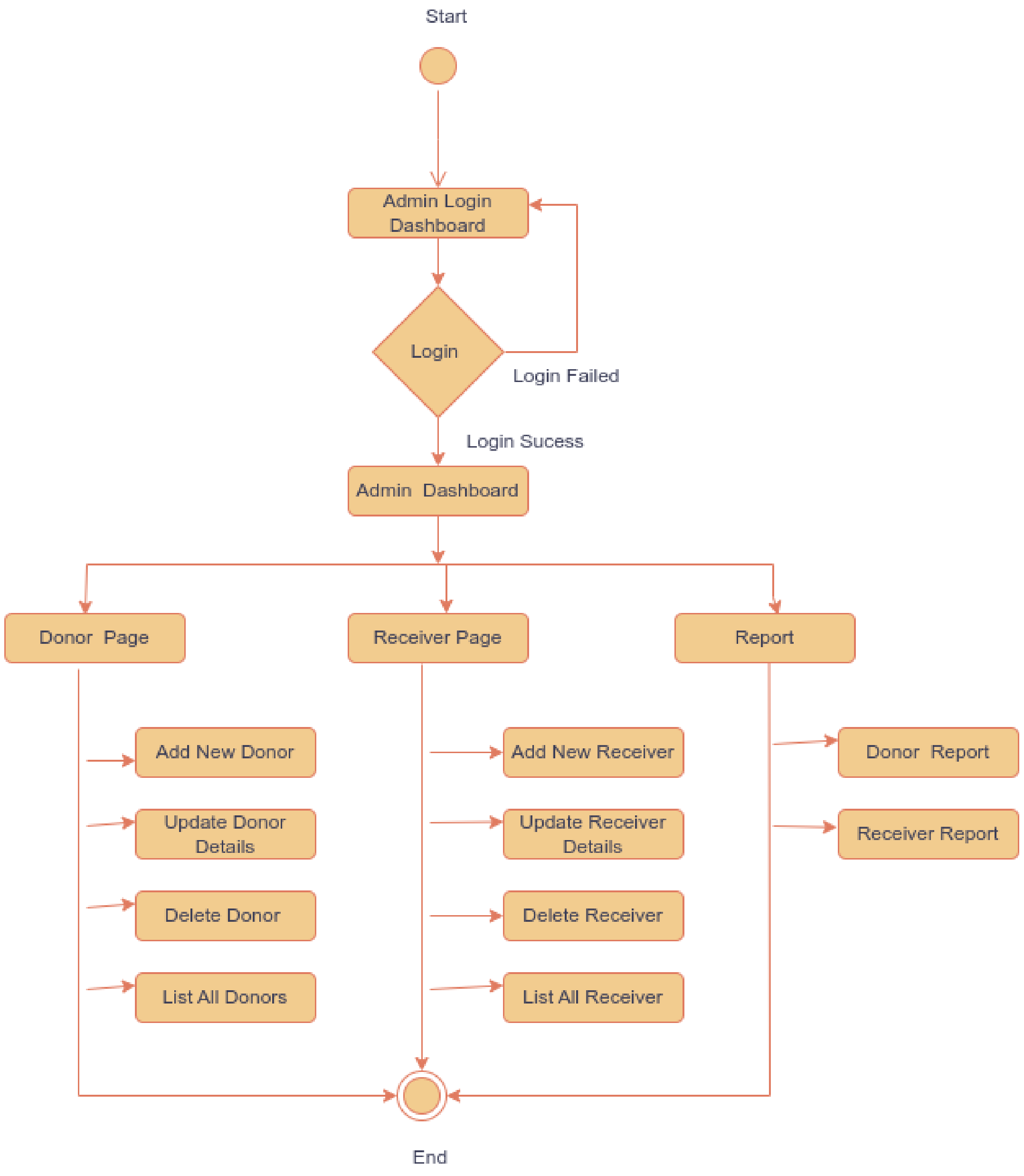
3.2 Delete the receiver details

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



4. User Interface

Admin interface is a console-based interface. In this interface admin will get menu driven options and based on selection of menu operation will beperformed. No GUI is available so admin will use console to interact with the database.

5. Technologies Used

1. For storing the data MySQL is used in local host.

2. MySQL workbench is used to for database visualization (as a GUIinterface 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

The creation of a console-based Blood Management System using Java and SQL offers streamlined blood inventory control. The system’s java interface allows medical professionals to manage donations, track blood types and oversee distribution. The integration of an SQL database ensures data accuracy for efficient reporting and analysis. It is light weight and easy to access. Future improvements could explore more user-friendlyinterface with more functionalities. This system overall demonstrates the importance of efficient 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
3. https://dev.mysql.com/doc/workbench/en/