



RedCell

Object-Oriented Blood Bank Management System

Prepared By

Group-07: RedCell

Course

CSE221 - Object Oriented Programming

Instructor

Professor Dr. Syed Akhter Hossain
Dean and Professor of CSE, DIU

Date: May 25, 2025

01 Introduction

Access to safe and timely blood supply is critical for patient care in hospitals. Manual handling of donor records, blood inventory and request fulfillment often leads to inefficiencies, delays and errors. An Object Oriented Blood Bank Management System (BBMS) will automate these processes, ensuring accurate inventory tracking, donor eligibility checks and streamlined request handling.

02 Project Objectives

- **Automate Critical Workflows:** Digitize donor registration, blood inventory tracking and request fulfillment to minimize manual errors.
- **Enhance Data Reliability:** Implement real-time stock monitoring and expiration alerts to reduce wastage and shortages.
- **Simplify User Roles:** Provide tailored interfaces and permissions for Admins, Donors, Hospitals and Recipients.
- **Demonstrate OOP Principles:** Build a modular, maintainable codebase showcasing encapsulation, inheritance and polymorphism.
- **Generate Actionable Insights:** Offer reporting dashboards for donation trends, inventory status and request metrics.

03 System Overview

RedCell is a desktop application that connects blood donors, hospital staff and administrative users through a centralized platform.

- Admins manage user accounts, configure system thresholds and view analytics.
- Donors record new donations, view eligibility status and track their contribution history.
- Hospitals/Recipients place blood unit requests and monitor fulfillment in real time.

04 Key Features

Feature	Description
Donor Management	Register donors, check eligibility (based on age, last donation date), and log donations
Request Processing	Hospitals submit unit requests; system matches with available inventory and allocates
Role-Based Authentication & Access	Secure login with distinct dashboards and permissions for each user role
User Interface	Intuitive GUI built with JavaFX for seamless navigation
Data Persistence	Store data using SQLite as an embedded database system

05 Module Breakdown & Responsibilities

Module	Assigned Member	Responsibilities
User Role Management & Authentication	Shafayat Yeamin Jian	Define abstract User class, manage login and roles
Inventory Control & Request Allocation	Md. Asikujjaman	Track blood stock and fulfill hospital requests
GUI Design & Integration	Saber Mahmud Sarker	Design and connect the graphical user interface
Database & Persistence	Emdad Hossen Bhuyan	Manage data storage and SQLite integration
Documentation & UML Reports	Md. Rafiul Karim	Create system documentation and UML diagrams

06 Technology Stack

- **Java** – Core Programming
- **JavaFX** – GUI Framework
- **SQLite** – Data persistence
- **VS Code & IntelliJ IDEA** – Development Environment
- **GitHub** – Version Control & collaboration
- **Lucidchart** – UML Diagram creation

07 Project Timeline

Week	Milestones
1 st	Finalize requirements, draft UML diagrams, Module planning
2 nd	Code module implementation & development
3 rd	Integrate GUI with backend services
4 th	Testing – unit, integration and user acceptance
5 th	Documentation/Report writing, polish UI (If needed) and presentation preparation

08 Expected Outcomes

- Fully functional desktop program demonstrating end-to-end workflows.
- Well-documented OOP code with clear module separation.
- UML diagrams, data dictionary, API descriptions and user guide.
- Concise overview of development journey and system demo with all Source code hosted on GitHub.

09 Conclusion

By developing RedCell, we will achieve practical experience in OOP design, lightweight database integration using SQLite, and GUI development—while solving a vital challenge in healthcare. This system promises to improve blood bank efficiency, transparency, and responsiveness, making a meaningful impact on patient care.