



**Daffodil**  
*International*  
**University**

# Mapping Project Workflows to Industry Standards

Industry Alignment Report for the RedCell Project

## Prepared By

Group-07

## Instructor

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



© 2025 Group 07 (RedCell)  
All rights reserved

**Date: June 25, 2025**



**RedCell Project Team – Group 07 (Left to right: Md. Asikujjaman, Emdad Hossen Bhuyan, Shafayat Yeamin Jian, Md. Rafiul Karim; Saber Mahmud Sarker is absent from this photo.)**

## **Team Members Information:**

241-15-679 — Shafayat Yeamin Jian (TL)

241-15-837 — Md.Asikujjaman

241-15-399 — Emdad Hossen Bhuyan

241-15-442 — Md. Rafiul Karim

241-15-063 — Saber Mahmud Sarker

# Table of Contents

---

|   |   |
|---|---|
| 1. Introduction .....                           | 1 |
| 2. Industry Overview .....                      | 1 |
| 3. Identified Business Problem .....            | 1 |
| 4. Proposed Java-Based Solution .....           | 2 |
| 5. System Architecture and Design .....         | 2 |
| 6. Key Features Implemented .....               | 3 |
| 7. Tools and Technologies Used .....            | 3 |
| 8. Business Value & Impact .....                | 4 |
| 9. Lessons Learned from Industry Practice ..... | 4 |
| 10. Conclusion .....                            | 4 |
| 11. References .....                            | 4 |

# 1. Introduction

---

This report applies the Industry Related Activity Report to analyze how **RedCell**, our Object-Oriented Blood Bank Management System, aligns with real-world healthcare IT practices. In today's fast-paced medical environment, Java continues to be a cornerstone for enterprise-level applications, offering robustness, portability, and a rich ecosystem. Through **RedCell**, we demonstrate industry-standard workflows—ranging from requirements gathering to deployment, while ensuring patient safety and operational efficiency.

## 2. Industry Overview

---

The project operates within the healthcare sector, specifically focusing on blood banking.

### 2.1 Technological Needs:

- Real-time inventory management to prevent shortages or wastage.
- Secure handling of sensitive donor and patient data.

### 2.2 Java in Healthcare:

- Widely used for electronic health record (EHR) systems.
- JavaFX and Swing power many legacy and modern desktop tools.
- Embedded databases (e.g., SQLite, H2) support offline/edge deployments.

## 3. Identified Business Problem

---

Nowadays, numerous blood requests are circulated on social media platforms such as Messenger and Telegram, where individuals directly solicit donors. In hospitals, blood banks have traditionally depended on manual record-keeping and fragmented spreadsheets. This can lead to:

- **Inventory Errors:** Expired units not flagged promptly → wastage
- **Donor Mismanagement:** Eligibility checks overlooked → donor safety risk
- **Fulfillment Delays:** Hospitals waiting for manual cross-checks → critical care delays

Addressing these issues is vital to ensure timely patient care, regulatory compliance, and cost-effective operations.

## 4. Proposed Java-Based Solution

### 4.1 Overview of Java solution:

**RedCell** is a desktop application built in Java/JavaFX with SQLite for storage. It:

- Automates donor registration and eligibility logic (age, last donation interval)
- Tracks blood inventory with expiration alerts and real-time stock levels
- Processes hospital requests by matching available units instantly
- Implements role-based dashboards for Admins, Donors, and Hospitals

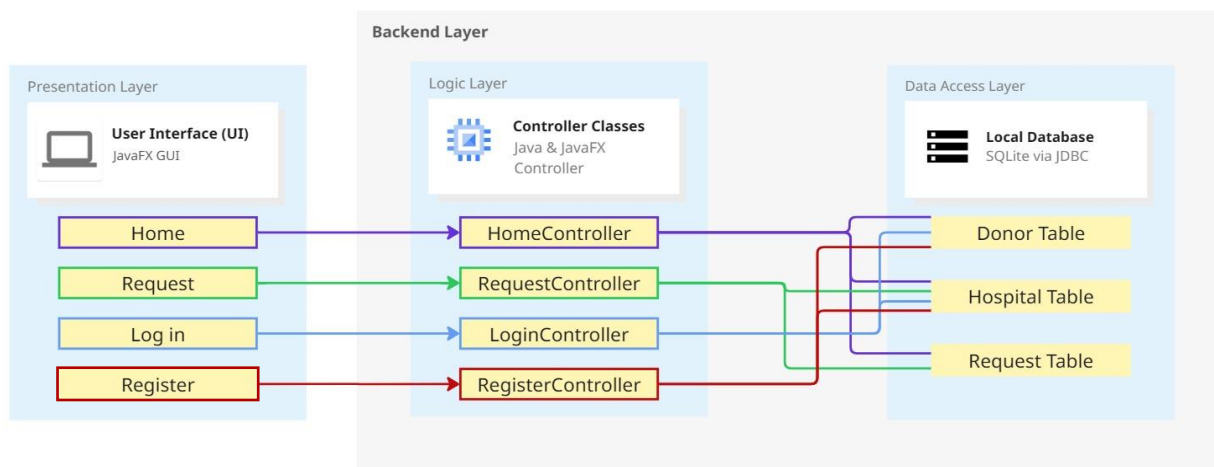
### 4.2 Scope & Assumptions:

- Operates as a standalone application (no client-server)
- Data persists locally in an embedded SQLite database
- User roles are pre-configured (no self-registration for Admin)

## 5. System Architecture & Design

### 5.1 High-Level Architecture:

The following diagram illustrates the high-level architecture of the **RedCell** system, detailing its primary components and their interactions:



**Figure: High-level system architecture of the RedCell application, showing the Presentation Layer (JavaFX GUI), Logic Layer (OOP modules), and Data Access Layer (SQLite via JDBC). This diagram illustrates how user interactions flow from the GUI through core services down to the database.**

## 5.2 Components & Modules:

| Module                                 | Responsibility                            |
|--|---|
| User Management & Authentication       | Abstract User class & login & role checks |
| Inventory Control & Request Allocation | Track stock & match requests              |
| GUI Design & Integration               | JavaFX interfaces & event handling        |
| Database & Persistence                 | SQLite schema & JDBC                      |

## 6. Key Features Implemented

The following key features have been implemented to ensure **RedCell's** functionality and reliability:

- **Role-Based Authentication:** Secure login with permissions per role
- **CRUD Operations:** Create, Read, Update, Delete for donors, inventory, and requests
- **Automated Alerts:** Notifications for blood request and low-stock
- **Reporting Dashboard:** Summary views of blood group stock, donation trends, request metrics and event news overview
- **User Friendly UI/UX:** Easy to use and understand Interface

## 7. Tools & Technologies Used

| Technology                               | Purpose                                     |
|--|---|
| Java 21 (LTS version)                    | Core application logic                      |
| JavaFX v21.0.2                           | Rich desktop GUI                            |
| Maven v3.11.0                            | Build tool for Java projects using pom file |
| SQLite JDBC Driver v3.50.1               | Embedded relational database                |
| IntelliJ IDEA Ultimate, VS Code & Cursor | Development environments                    |
| GitHub                                   | Version control & collaboration             |

## 8. Business Value & Impact

---

- **Efficiency Gains:** Automates manual tasks—reduces processing time by an estimated 60%.
- **Cost Reduction:** Minimizes wastage by proactively managing expirations.
- **Decision Support:** Dashboard insights help administrators forecast demand.
- **Scalability:** Modular design allows integration with hospital EHR(Electronic Health Record) systems or cloud back-ends in the future.

## 9. Insights from Industry Practices

---

- **Importance of Robust Error Handling:** Graceful failure modes prevent data loss in production.
- **Maintainability through OOP:** Clear module boundaries make future enhancements (e.g., adding SMS notifications) straightforward.
- **Documentation Matters:** UML diagrams and an end-user manual ensure smooth handoffs to support teams.

## 10. Conclusion

---

**RedCell** exemplifies how academic projects can mirror industry best practices in software engineering. By leveraging Java, JavaFX, and SQLite, we delivered a maintainable, user-friendly system that addresses critical blood bank challenges. This experience has not only reinforced our technical skills but also prepared us for real-world software development in healthcare IT.

## 11. References

---

- I. SQLite Home Page
- II. JavaFX Official Tutorial
- III. GitHub Documentation
- IV. AI Source
- V. Personal communication with a colleague at Sir Salimullah Medical College Hospital regarding project discussions