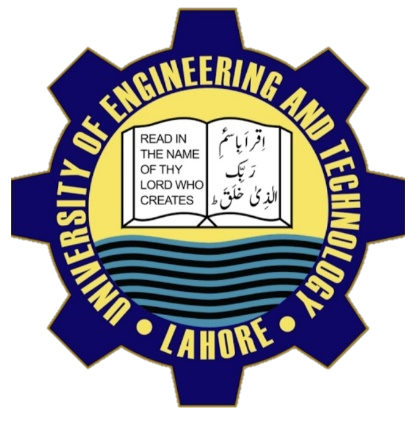
**Blood Donation Management System**

**Design Document**

****

**Submitted By:**

Wali Muhammad 2021-SE-39

Zohra Amna 2022-SE-03

Muhammad Noman 2022-SE-31

**Submitted To:**

Ma’am Alina Munir

**Submission Date:**

December 12, 2024

Department of Computer Science

**University of Engineering and Technology, New Campus**

**Lahore**

**Table Of Contents**

[**1. Introduction** 6](#_Toc184750793)

[**2. Goals and Objectives** 6](#_Toc184750794)

[**2.1. High-Level Goals** 6](#_Toc184750795)

[**2.2. Specific Objectives** 6](#_Toc184750796)

[**3. Scope** 6](#_Toc184750797)

[**4. Audience** 6](#_Toc184750798)

[**4.1. Primary Stakeholders** 6](#_Toc184750799)

[**4.2. Secondary Stakeholders** 6](#_Toc184750800)

[**5. Functional and Non-Functional Requirements** 7](#_Toc184750801)

[**5.1. Functional Requirements** 7](#_Toc184750802)

[**5.1.1. User Features:** 7](#_Toc184750803)

[**5.1.2. Admin Features:** 7](#_Toc184750804)

[**5.2. Non-Functional Requirements** 7](#_Toc184750805)

[**6. System Architecture** 7](#_Toc184750806)

[**6.1 Architecture Overview** 7](#_Toc184750807)

[**6.1.1. Presentation Layer:** 7](#_Toc184750808)

[**6.1.2. Business Logic Layer:** 7](#_Toc184750809)

[**6.1.3. Data Layer:** 7](#_Toc184750810)

[**7. Functional Design** 8](#_Toc184750811)

[**7.1 Use Case Descriptions** 8](#_Toc184750812)

[**7.1.1. Actors:** 8](#_Toc184750813)

[**7.1.2. Use Case List:** 8](#_Toc184750814)

[**8. Database Design** 8](#_Toc184750815)

[**8.1. Entity Relationship Diagram** 8](#_Toc184750816)

[**8.2. Database Schema** 9](#_Toc184750817)

[**8.2.1. Database Creation** 9](#_Toc184750818)

[**8.2.2. Database Relations Creation** 9](#_Toc184750819)

[**8.3. Data Storage Design** 10](#_Toc184750820)

[**8.3.1. Database Type** 10](#_Toc184750821)

[**8.3.2. Data Model** 10](#_Toc184750822)

[**8.3.3. Data Consistency** 11](#_Toc184750823)

[**8.3.4. Data Scalability and Performance** 11](#_Toc184750824)

[**8.3.5. Data Security** 11](#_Toc184750825)

[**8.3.6. Data Backup and Retention** 11](#_Toc184750826)

[**8.3.7. Storage Location** 11](#_Toc184750827)

[**9. UML Diagrams** 12](#_Toc184750828)

[**9.1. Use Case Diagram** 12](#_Toc184750829)

[**9.2. Class Diagram** 13](#_Toc184750830)

[**9.3. Object Diagram Diagrams** 14](#_Toc184750831)

[**9.4. Sequence Diagrams** 14](#_Toc184750832)

[**9.5. Activity Diagrams** 21](#_Toc184750833)

[**9.6. Deployment Diagram** 29](#_Toc184750834)

[**9.7. Package Diagram** 30](#_Toc184750835)

[**9.8. State Machine Diagram** 30](#_Toc184750836)

[**9.9. Collaboration Diagram** 36](#_Toc184750837)

[**9.10. Component Diagram** 43](#_Toc184750838)

[**10. User Interface Design (UI)** 44](#_Toc184750839)

[**10.1. User Wireframes** 44](#_Toc184750840)

[**10.2. Admin Wireframes** 47](#_Toc184750841)

[**11. Technology Stack** 49](#_Toc184750842)

[**11.1. Backend** 49](#_Toc184750843)

[**11.2. Frontend** 50](#_Toc184750844)

[**12. Security** 50](#_Toc184750845)

[**13. Deployment** 50](#_Toc184750846)

[**14. Testing** 50](#_Toc184750847)

[**15. Future Enhancements** 50](#_Toc184750848)

[**16. References** 50](#_Toc184750849)

# 

**Figures**

[Figure 1: Entity Relationship Diagram 8](#_Toc184719272)

[Figure 2: Use Case Diagram 12](#_Toc184719273)

[Figure 3: Class Diagram 13](#_Toc184719274)

[Figure 4: Object Diagram 14](#_Toc184719275)

[Figure 5: User Sign up Sequence Diagram 14](#_Toc184719276)

[Figure 6: User Login Sequence Diagram 15](#_Toc184719277)

[Figure 7: User View Blood Inventory Sequence Diagram 15](#_Toc184719278)

[Figure 8: User Search Blood by City Sequence Diagram 16](#_Toc184719279)

[Figure 9: User Search Blood by Blood Group Sequence Diagram 16](#_Toc184719280)

[Figure 10: User Request Blood Donation Sequence Diagram 17](#_Toc184719281)

[Figure 11: User Make Blood Appeals Sequence Diagram 17](#_Toc184719282)

[Figure 12: Admin Login Sequence Diagram 18](#_Toc184719283)

[Figure 13: Admin View Blood Inventory Sequence Diagram 18](#_Toc184719284)

[Figure 14: Admin Manage Blood Inventory Sequence Diagram 19](#_Toc184719285)

[Figure 15: Admin Search Blood by City Sequence Diagram 19](#_Toc184719286)

[Figure 16: Admin Search Blood by Blood Group Sequence Diagram 20](#_Toc184719287)

[Figure 17: Admin Approve or Reject Blood Appeals Sequence Diagram 20](#_Toc184719288)

[Figure 18: Admin Approve or Reject Blood Donation Requests Sequence Diagram 21](#_Toc184719289)

[Figure 19: User Sign up Activity Diagram 22](#_Toc184719290)

[Figure 20: User Login Activity Diagram 23](#_Toc184719291)

[Figure 21: User View Blood Inventory Activity Diagram 23](#_Toc184719292)

[Figure 22: User Search Blood by City Activity Diagram 24](#_Toc184719293)

[Figure 23: User Search Blood by Blood Group Activity Diagram 24](#_Toc184719294)

[Figure 24: User Request Blood Donation Activity Diagram 25](#_Toc184719295)

[Figure 25: User Make Blood Appeals Activity Diagram 25](#_Toc184719296)

[Figure 26: Admin Login Activity Diagram 26](#_Toc184719297)

[Figure 27: Admin View Blood Inventory Activity Diagram 26](#_Toc184719298)

[Figure 28: Admin Manage/Update Blood Inventory Activity Diagram 27](#_Toc184719299)

[Figure 29: Admin Search Blood by City Activity Diagram 27](#_Toc184719300)

[Figure 30: Admin Search Blood by Blood Group Activity Diagram 28](#_Toc184719301)

[Figure 31: Admin Approve or Reject Blood Appeals Activity Diagram 28](#_Toc184719302)

[Figure 32: Admin Approve or Reject Blood Donation Requests Activity Diagram 29](#_Toc184719303)

[Figure 33: Deployment Diagram 29](#_Toc184719304)

[Figure 34: Package Diagram 30](#_Toc184719305)

[Figure 35: User Sign up State Machine Diagram 30](#_Toc184719306)

[Figure 36: User Login State Machine Diagram 31](#_Toc184719307)

[Figure 37: User View Blood Inventory State Machine Diagram 31](#_Toc184719308)

[Figure 38: User Search Blood by City State Machine Diagram 31](#_Toc184719309)

[Figure 39: User Search Blood by Blood Group State Machine Diagram 32](#_Toc184719310)

[Figure 40: User Request Blood Donation State Machine Diagram 32](#_Toc184719311)

[Figure 41: User Make Blood Appeals State Machine Diagram 33](#_Toc184719312)

[Figure 42: Admin Login State Machine Diagram 33](#_Toc184719313)

[Figure 43: Admin View Blood Inventory State Machine Diagram 33](#_Toc184719314)

[Figure 44: Admin Manage Blood Inventory State Machine Diagram 34](#_Toc184719315)

[Figure 45: Admin Search Blood by City State Machine Diagram 34](#_Toc184719316)

[Figure 46: Admin Search Blood by Blood Group State Machine Diagram 35](#_Toc184719317)

[Figure 47: Admin Approve or Reject Blood Appeals State Machine Diagram 35](#_Toc184719318)

[Figure 48: Admin Approve or Reject Blood Donation Requests State Machine Diagram 36](#_Toc184719319)

[Figure 49: User Sign up Collaboration Diagram 36](#_Toc184719320)

[Figure 50: User Login Collaboration Diagram 37](#_Toc184719321)

[Figure 51: User View Blood Inventory Collaboration Diagram 37](#_Toc184719322)

[Figure 52: User Search Blood by City Collaboration Diagram 38](#_Toc184719323)

[Figure 53: User Search Blood by Blood Group Collaboration Diagram 38](#_Toc184719324)

[Figure 54: User Request Blood Donation Collaboration Diagram 39](#_Toc184719325)

[Figure 55: User Make Blood Appeals Collaboration Diagram 39](#_Toc184719326)

[Figure 56: Admin Login Collaboration Diagram 40](#_Toc184719327)

[Figure 57: Admin View Blood Inventory Collaboration Diagram 40](#_Toc184719328)

[Figure 58: Admin Manage Blood Inventory Collaboration Diagram 41](#_Toc184719329)

[Figure 59: Admin Search Blood by City Collaboration Diagram 41](#_Toc184719330)

[Figure 60: Admin Search Blood by Blood Group Collaboration Diagram 42](#_Toc184719331)

[Figure 61: Admin Approve or Reject Blood Appeals Collaboration Diagram 42](#_Toc184719332)

[Figure 62: Admin Approve or Reject Blood Donation Requests Collaboration Diagram 43](#_Toc184719333)

[Figure 63: Component Diagram 43](#_Toc184719334)

[Figure 64: User Sign up Wireframe 44](#_Toc184719335)

[Figure 65: User Login Wireframe 44](#_Toc184719336)

[Figure 66: User Search Blood Wireframe 45](#_Toc184719337)

[Figure 67: User make Blood Appeal Wireframe 45](#_Toc184719338)

[Figure 68: User Donation Request Wireframe 46](#_Toc184719339)

[Figure 69: User View Blood Inventory Wireframe 46](#_Toc184719340)

[Figure 70: Admin Login Wireframe 47](#_Toc184719341)

[Figure 71: Admin Search Blood Wireframe 47](#_Toc184719342)

[Figure 72: Admin Accept/Reject Donation Requests Wireframe 48](#_Toc184719343)

[Figure 73: Admin Accept/Reject Blood Appeals Wireframe 48](#_Toc184719344)

[Figure 74: Amin View Inventory Wireframe 49](#_Toc184719345)

[Figure 75: Admin Update Inventory Wireframe 49](#_Toc184719346)

# **1. Introduction**

The Blood Management System is designed to streamline blood donation and distribution processes. It connects donors, recipients, and administrators to efficiently manage blood inventory, handle donation requests, and approve appeals.

This document outlines the architecture, design choices, and implementation plan for the system. Spaces are reserved for visual representations such as diagrams and database schemas.

# **2. Goals and Objectives**

Here are the goals and objectives of our project which we will achieve:

## **2.1. High-Level Goals**

* Provide a platform for users to view blood inventory and submit blood appeals.
* Enable efficient inventory management for admins to ensure blood availability.
* Facilitate donation requests and tracking in a user-friendly manner.

## **2.2. Specific Objectives**

* Provide an intuitive platform for users to search and request blood.
* Enable admins to manage blood inventory and approve/reject requests.
* Ensure smooth operations with secure and robust data handling.
* Offer easy-to-use interfaces for all stakeholders.

# **3. Scope**

The Blood Management System will include:

* User functionalities like sign-in, sign-up, and viewing/searching blood inventory.
* Admin functionalities for inventory management and appeals approval/rejection.
* Real-time inventory tracking and updates.
* Secure and role-based access to the system.

**Exclusions**:

* Payment processing for blood donations.
* Integration with external blood bank systems.

# **4. Audience**

Audience and Stakeholders of the project includes:

## **4.1. Primary Stakeholders**

* End-users in need of blood (patients, relatives, hospitals).
* Blood bank administrators.

## **4.2. Secondary Stakeholders**

* Developers and maintainers of the system.

# **5. Functional and Non-Functional Requirements**

## **5.1. Functional Requirements**

### **5.1.1. User Features:**

* Users must Sign up.
* User must Login.
* User must View Blood Inventory.
* Users must Search Blood by City.
* User must Search Blood by Blood Group.
* User must Request Blood Donation.
* Users must Make Blood Appeals.

### **5.1.2. Admin Features:**

* Admin must Login.
* Admin must View Blood Inventory.
* Admin must Manage Blood Inventory.
* Admin must Search Blood by City.
* Admin must Search Blood by Blood Group.
* Admin must Approve or Reject Blood Appeals.
* Admin must Approve or Reject Blood Donation Requests.

## **5.2. Non-Functional Requirements**

Non-Function requirements include:

* **Performance**: The system should manage up to 10 concurrent users.
* **Reliability**: 99.99% uptime expected.
* **Security**: Role-based access control.
* **Scalability**: Ability to scale horizontally to accommodate increased users.
* **Usability**: Intuitive interface for both users and admins.

# **6. System Architecture**

## **6.1 Architecture Overview**

The system follows a 3-Tier Architecture:

### **6.1.1. Presentation Layer:**

Frontend for user interaction.

### **6.1.2. Business Logic Layer:**

Backend logic, validation, and processing.

### **6.1.3. Data Layer:**

Database for storing and managing data.

# **7. Functional Design**

## **7.1 Use Case Descriptions**

### **7.1.1. Actors:**

* **User:** Interacts with the system to view inventory, search for blood, or make requests/appeals.
* **Admin:** Manages the system by handling appeals, requests, and inventory.

### **7.1.2. Use Case List:**

* User Signup.
* User Login.
* View Blood Inventory.
* Search Blood by City
* Search Blood by Blood Group.
* Make Blood Appeals.
* Request Blood Donation.
* Admin Login.
* Approve/Reject Blood Appeals.
* Approve/Reject Blood Donation Requests.
* Update Blood Inventory.

# **8. Database Design**

## **8.1. Entity Relationship Diagram**

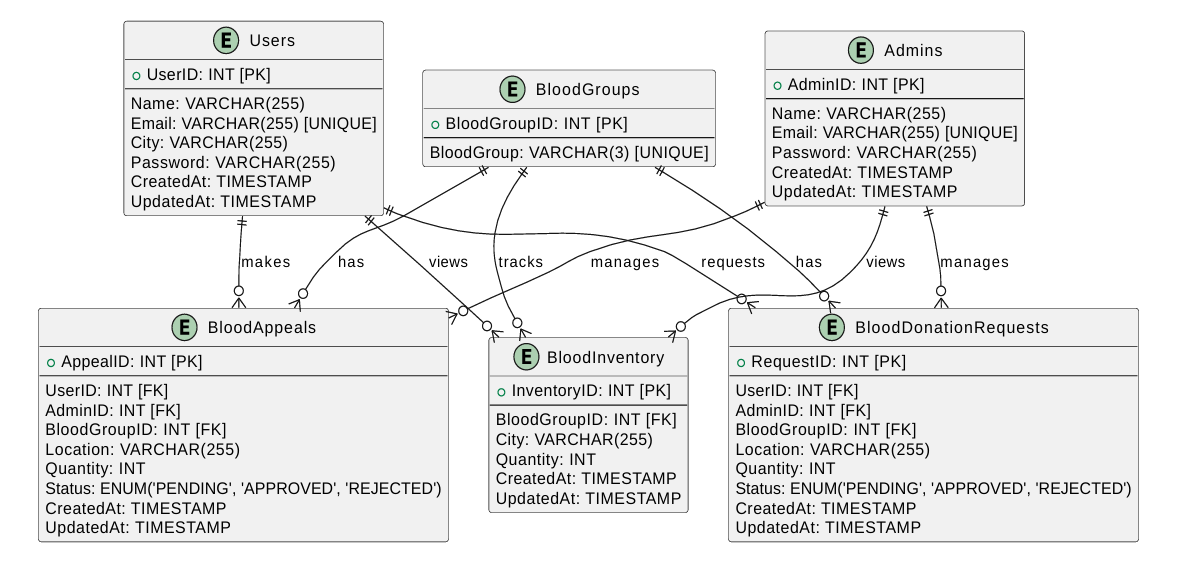


Figure 1: Entity Relationship Diagram

## **8.2. Database Schema**

### **8.2.1. Database Creation**

CREATE DATABASE BloodManagementSystem.

USE BloodManagementSystem.

### **8.2.2. Database Relations Creation**

* CREATE TABLE Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR (255) NOT NULL,

Email VARCHAR (255) UNIQUE NOT NULL,

City VARCHAR (255) NOT NULL,

Password VARCHAR (255) NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

* CREATE TABLE Admins (

AdminID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR (255) NOT NULL,

Email VARCHAR (255) UNIQUE NOT NULL,

Password VARCHAR (255) NOT NULL,

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

* CREATE TABLE BloodGroups (

BloodGroupID INT PRIMARY KEY AUTO\_INCREMENT,

BloodGroup VARCHAR (3) UNIQUE NOT NULL

);

* CREATE TABLE BloodInventory (

InventoryID INT PRIMARY KEY AUTO\_INCREMENT,

BloodGroupID INT NOT NULL,

FOREIGN KEY (BloodGroupID) REFERENCES BloodGroups (BloodGroupID),

City VARCHAR (255) NOT NULL,

Quantity INT DEFAULT 0 NOT NULL,

UNIQUE (BloodGroupID, City),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

* CREATE TABLE BloodAppeals (

AppealID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT NOT NULL,

FOREIGN KEY (UserID) REFERENCES Users (UserID),

AdminID INT DEFAULT NULL,

FOREIGN KEY (AdminID) REFERENCES Admins (AdminID),

BloodGroupID INT NOT NULL,

FOREIGN KEY (BloodGroupID) REFERENCES BloodGroups (BloodGroupID),

Location VARCHAR (255) NOT NULL,

Quantity INT NOT NULL,

Status ENUM ('PENDING', 'APPROVED', 'REJECTED') DEFAULT 'PENDING',

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

* CREATE TABLE BloodDonationRequests (

RequestID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT NOT NULL,

FOREIGN KEY (UserID) REFERENCES Users (UserID),

AdminID INT DEFAULT NULL,

FOREIGN KEY (AdminID) REFERENCES Admins (AdminID),

BloodGroupID INT NOT NULL,

FOREIGN KEY (BloodGroupID) REFERENCES BloodGroups (BloodGroupID),

Location VARCHAR (255) NOT NULL,

Quantity INT NOT NULL,

Status ENUM ('PENDING', 'APPROVED', 'REJECTED') DEFAULT 'PENDING',

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

UpdatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

## **8.3. Data Storage Design**

In this project, the data storage strategy is carefully designed to ensure reliability, security, and scalability. The following decisions outline how data is stored and managed for the Blood Management System:

### **8.3.1. Database Type**

We have selected MySQL, a relational database management system, due to its robustness, ACID compliance, and compatibility with the Spring Boot framework. MySQL's ability to manage structured data and enforce relationships between entities makes it ideal for this project.

### **8.3.2. Data Model**

The database is modeled using an Entity-Relationship Diagram (ERD) to represent entities like Users, Admins, BloodInventory, BloodGroups, BloodAppeals, and BloodDonationRequests. Each table includes primary keys, foreign keys, and constraints to ensure data integrity and relationships.

### **8.3.3. Data Consistency**

Relational constraints like foreign keys are used to maintain the integrity of data relationships. For instance:

* The BloodGroupID in the BloodInventory table references the BloodGroupID in the BloodGroups table.
* Similarly, the UserID in the BloodAppeals table references the UserID in the Users table.

### **8.3.4. Data Scalability and Performance**

Data Scalability and Performance include:

**Scalability:** The database is designed to accommodate future growth, with the potential for horizontal scaling (partitioning) and vertical scaling (server upgrades).

**Performance Optimization:** Indexing is applied to frequently queried columns, such as City and BloodGroup, to speed up search operations.

### **8.3.5. Data Security**

To protect sensitive information:

* Passwords are hashed using secure algorithms (e.g., bcrypt) before storage.
* Data access is controlled at the application level to prevent unauthorized access.

### **8.3.6. Data Backup and Retention**

Regular database backups are scheduled to prevent data loss in case of hardware or software failure. Policies are defined for retaining and archiving records, such as completed blood appeals, after a specified period.

### **8.3.7. Storage Location**

Initially, the database will be hosted locally during development for ease of use. For deployment, it will be migrated to a cloud-based platform like AWS RDS or Google Cloud SQL to ensure high availability and accessibility for all stakeholders.

**8.3.8. Future-Proofing**

The system is designed with flexibility to integrate additional features, such as advanced reporting and analytics, without major changes to the data model. This approach ensures that data is stored in a dependable, secure, and efficient manner while supporting the system's current and future needs.

# **9. UML Diagrams**

## **9.1. Use Case Diagram**

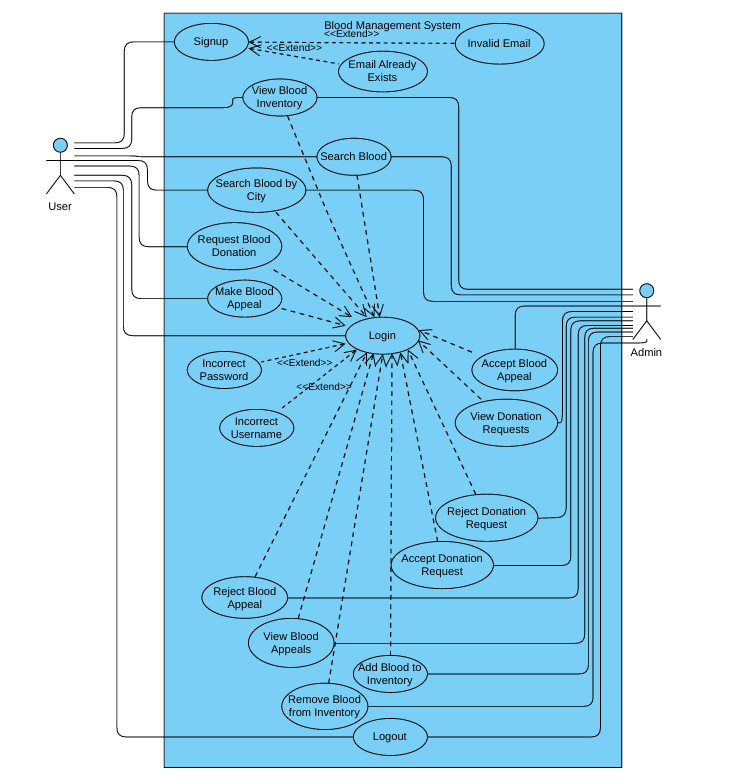


Figure 2: Use Case Diagram

## **9.2. Class Diagram**

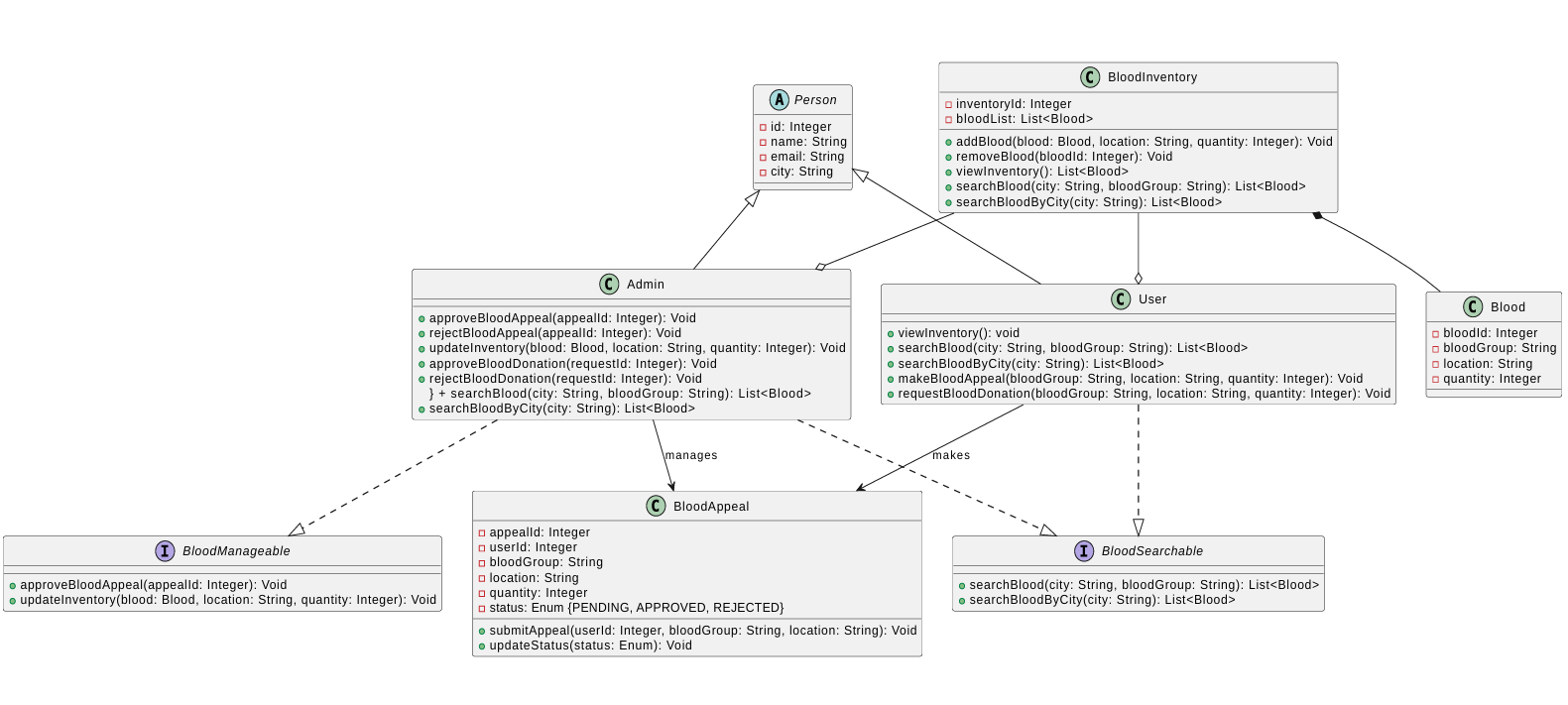


Figure 3: Class Diagram

## **9.3. Object Diagram Diagrams**

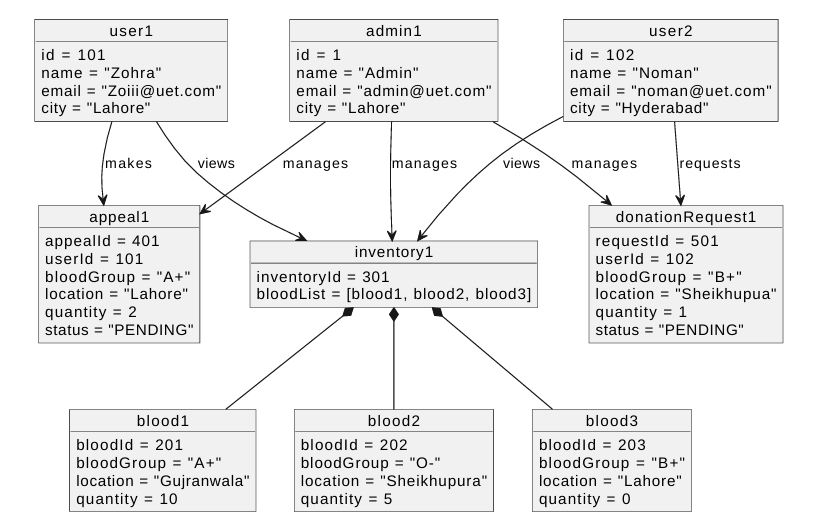


Figure 4: Object Diagram

## **9.4. Sequence Diagrams**

**User Features:**

* Signup.

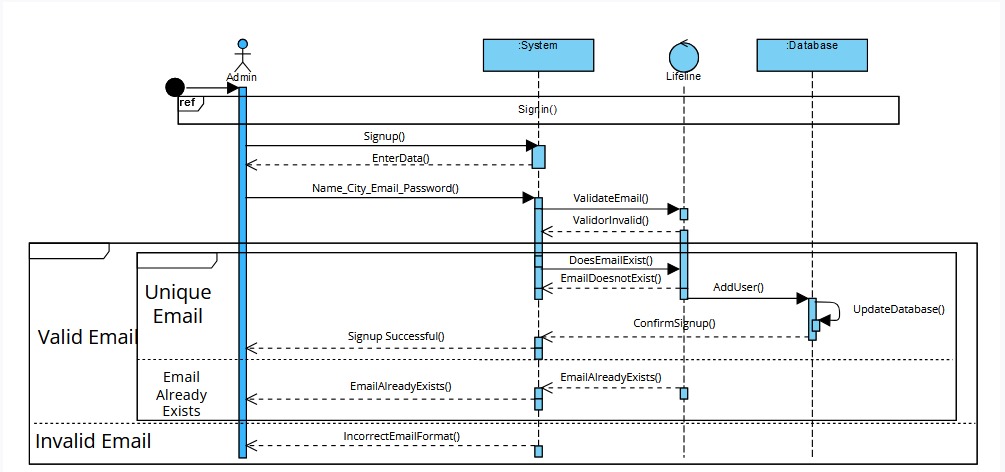


Figure 5: User Sign up Sequence Diagram

* Login.

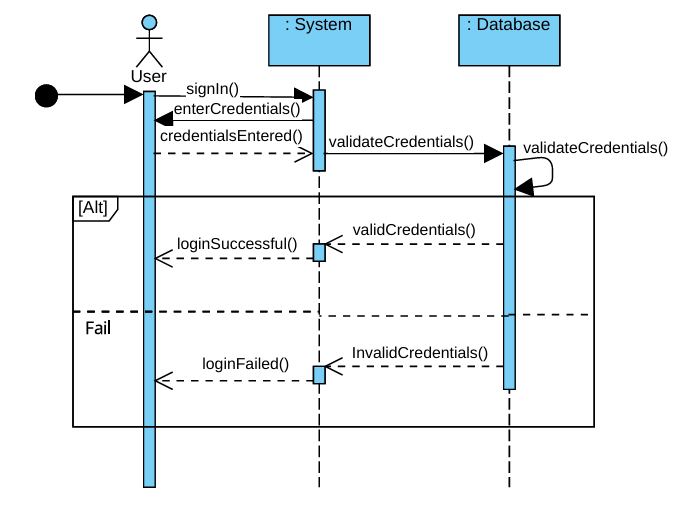


Figure 6: User Login Sequence Diagram

* View Blood Inventory.

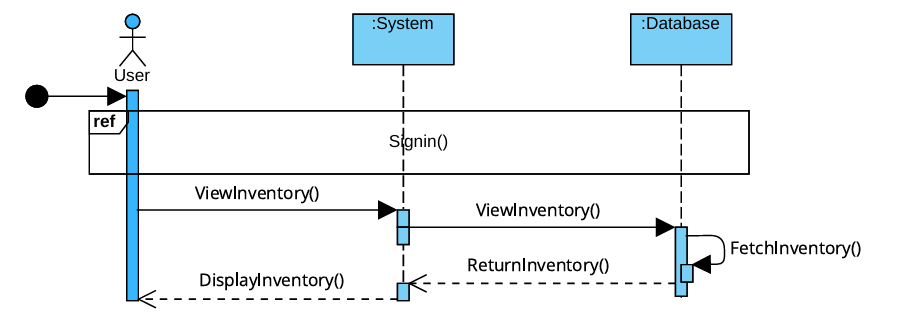


Figure 7: User View Blood Inventory Sequence Diagram

* Search Blood by City.

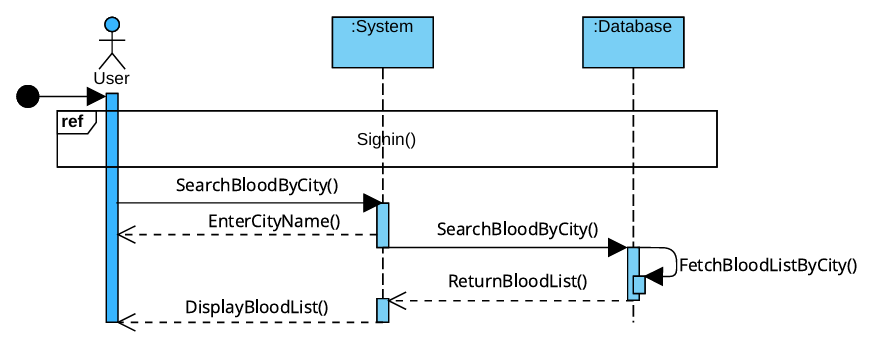


Figure 8: User Search Blood by City Sequence Diagram

* Search Blood by Blood Group.

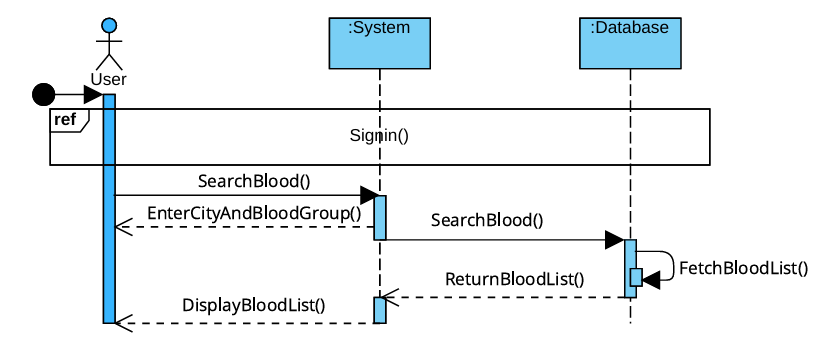


Figure 9: User Search Blood by Blood Group Sequence Diagram

* Request Blood Donation.

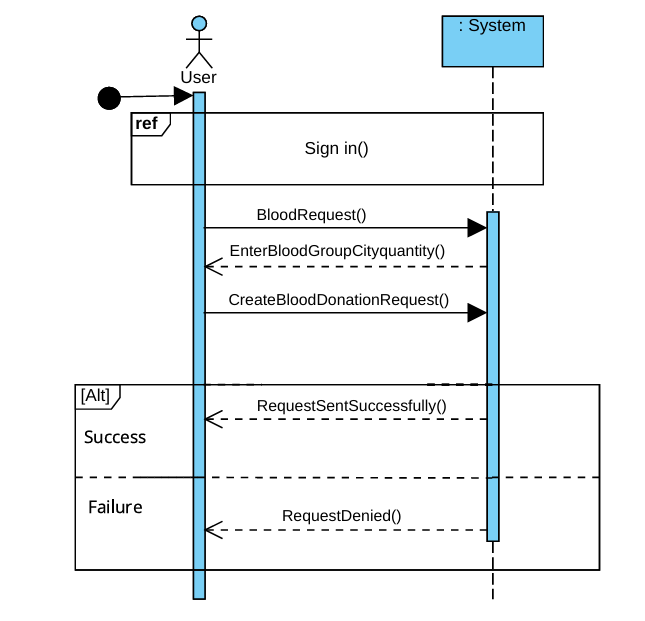


Figure 10: User Request Blood Donation Sequence Diagram

* Make Blood Appeals.

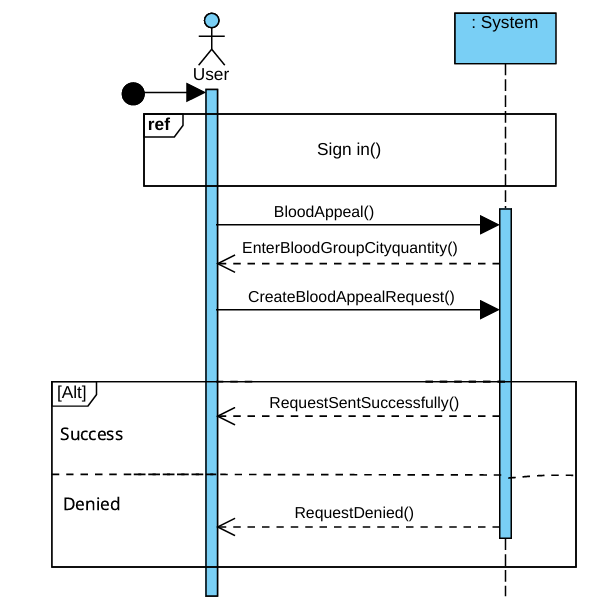


Figure 11: User Make Blood Appeals Sequence Diagram

**Admin Features:**

* Login.

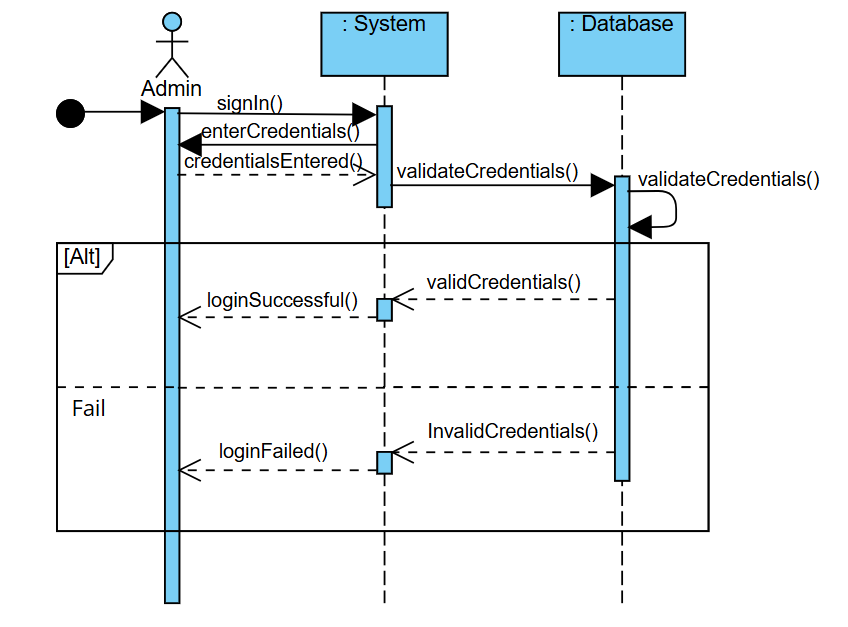


Figure 12: Admin Login Sequence Diagram

* View Blood Inventory.

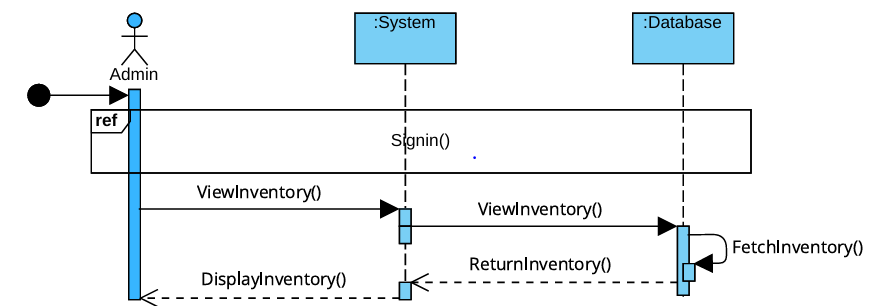


Figure 13: Admin View Blood Inventory Sequence Diagram

* Manage Blood Inventory.

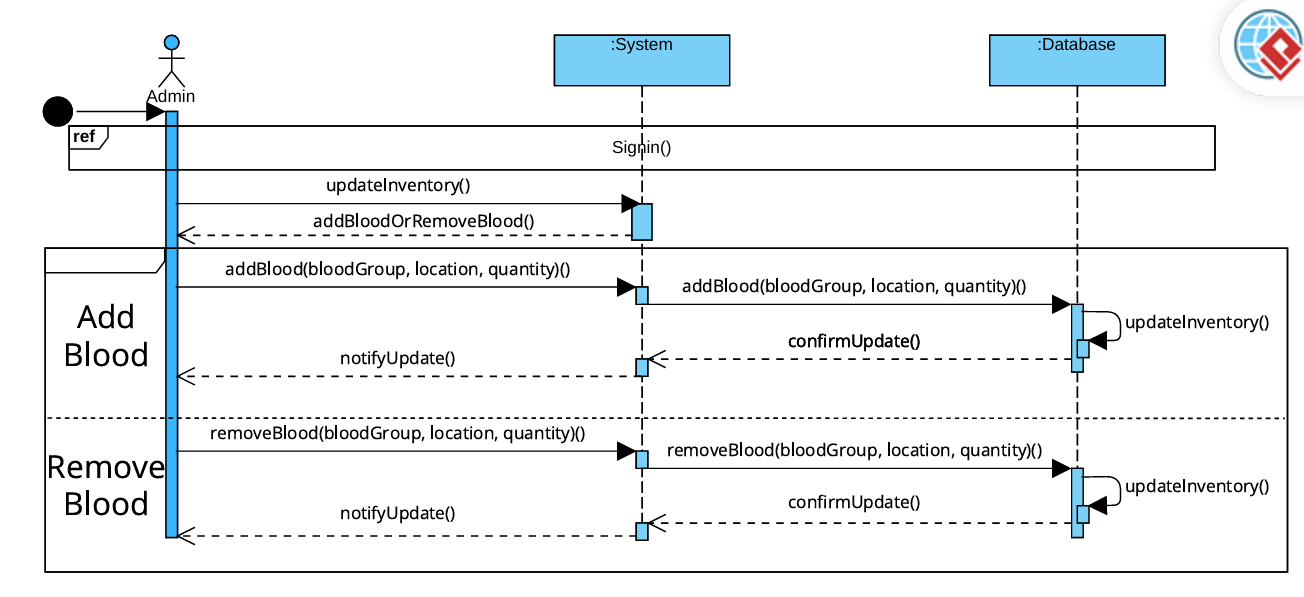


Figure 14: Admin Manage Blood Inventory Sequence Diagram

* Search Blood by City.

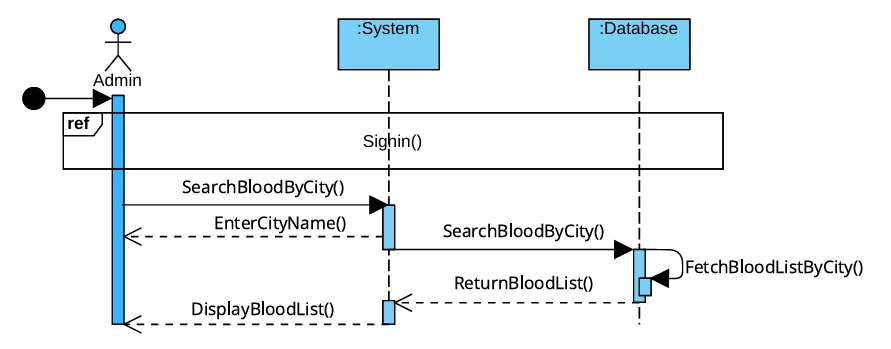


Figure 15: Admin Search Blood by City Sequence Diagram

* Search Blood by Blood Group.

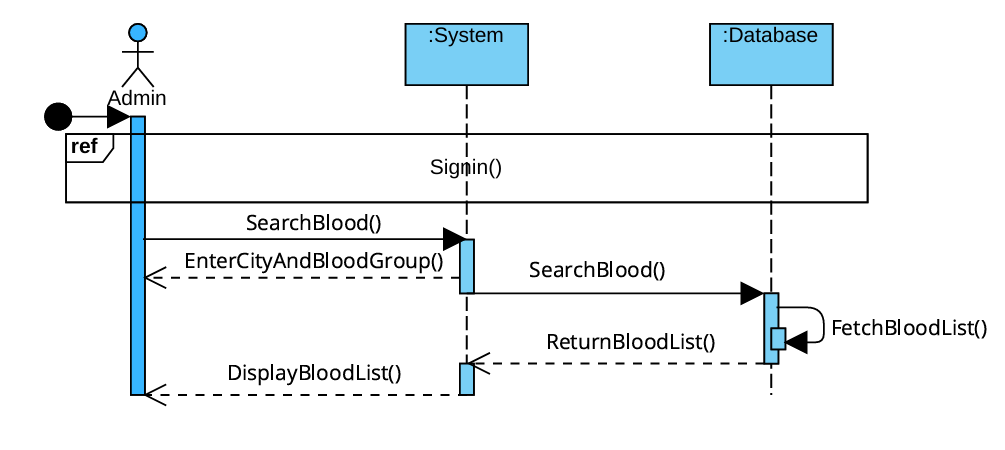


Figure 16: Admin Search Blood by Blood Group Sequence Diagram

* Approve or Reject Blood Appeals.

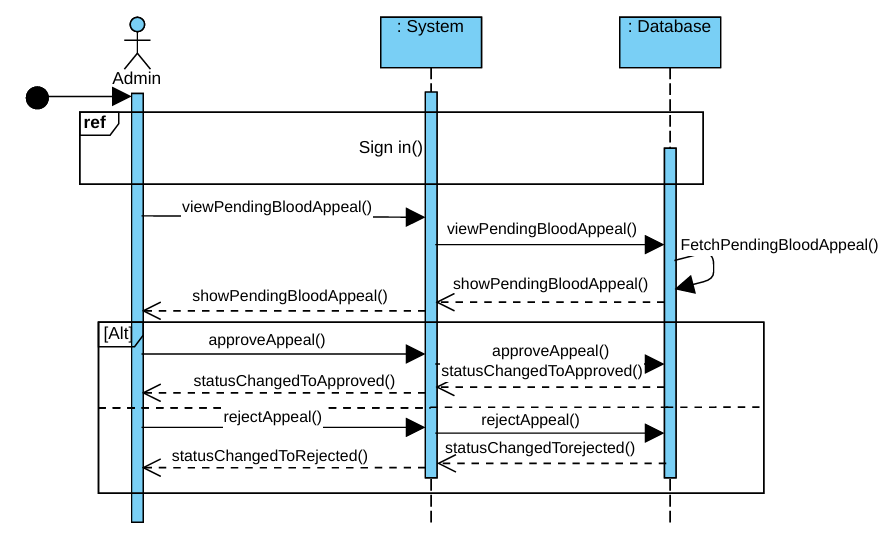


Figure 17: Admin Approve or Reject Blood Appeals Sequence Diagram

* Approve or Reject Blood Donation Requests.

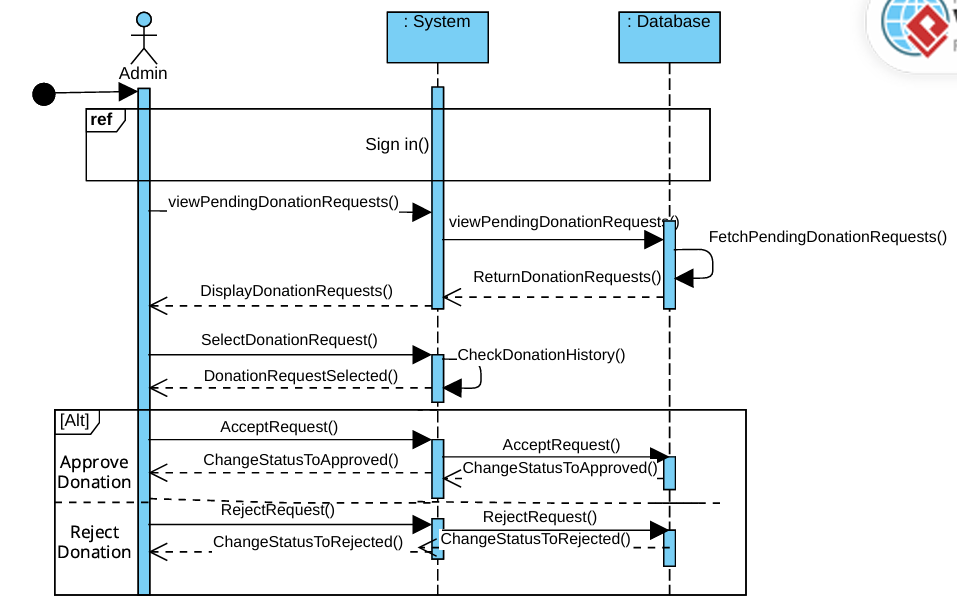


Figure 18: Admin Approve or Reject Blood Donation Requests Sequence Diagram

## **9.5. Activity Diagrams**

**User Features:**

* Signup.

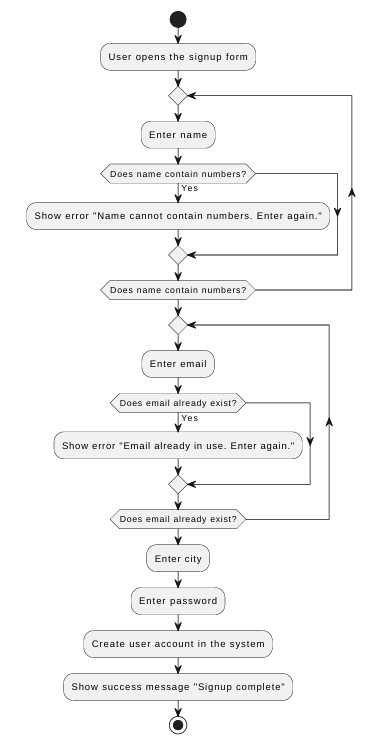


Figure 19: User Sign up Activity Diagram

* Login

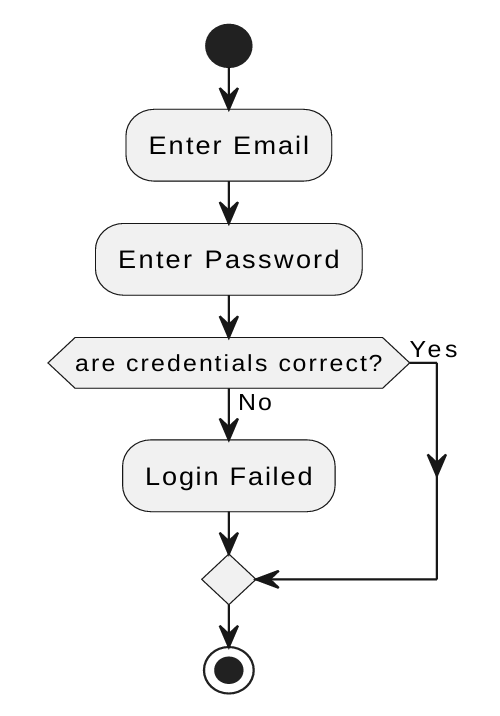


Figure 20: User Login Activity Diagram

* View Blood Inventory.

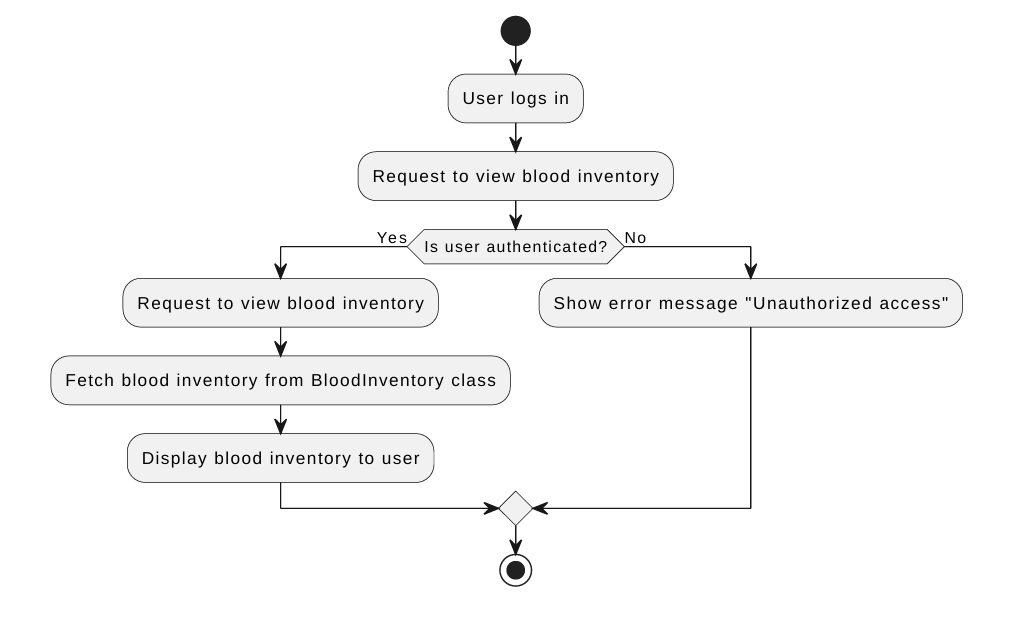


Figure 21: User View Blood Inventory Activity Diagram

* Search Blood by City and Blood Group.

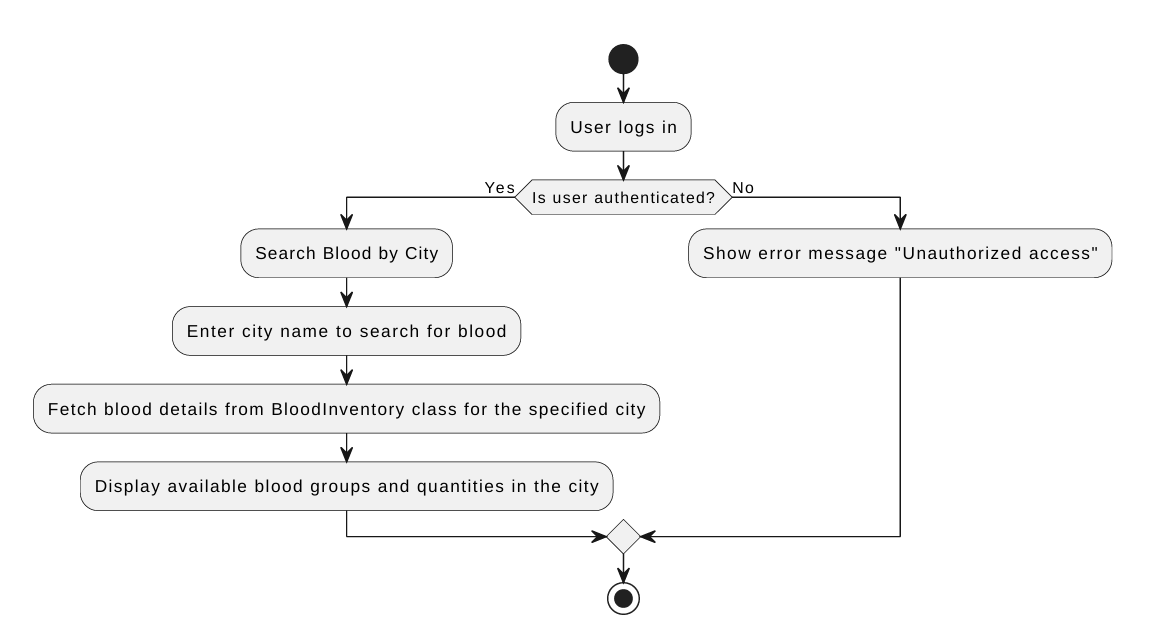


Figure 22: User Search Blood by City Activity Diagram

* Search Blood by Blood Group

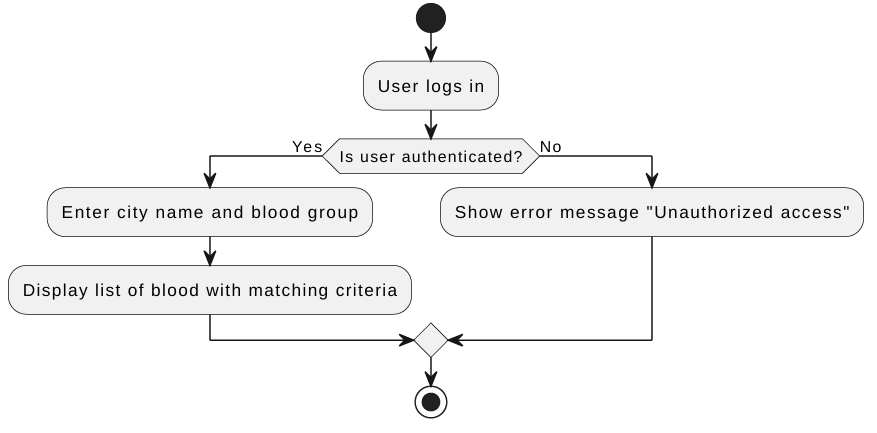


Figure 23: User Search Blood by Blood Group Activity Diagram

* Request Blood Donation.

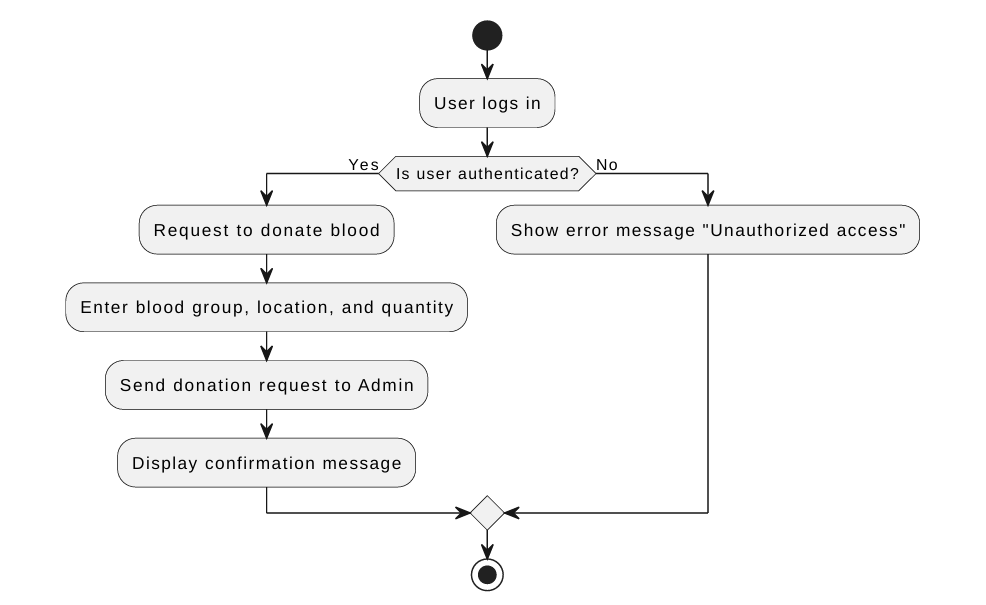


Figure 24: User Request Blood Donation Activity Diagram

* Make Blood Appeals.

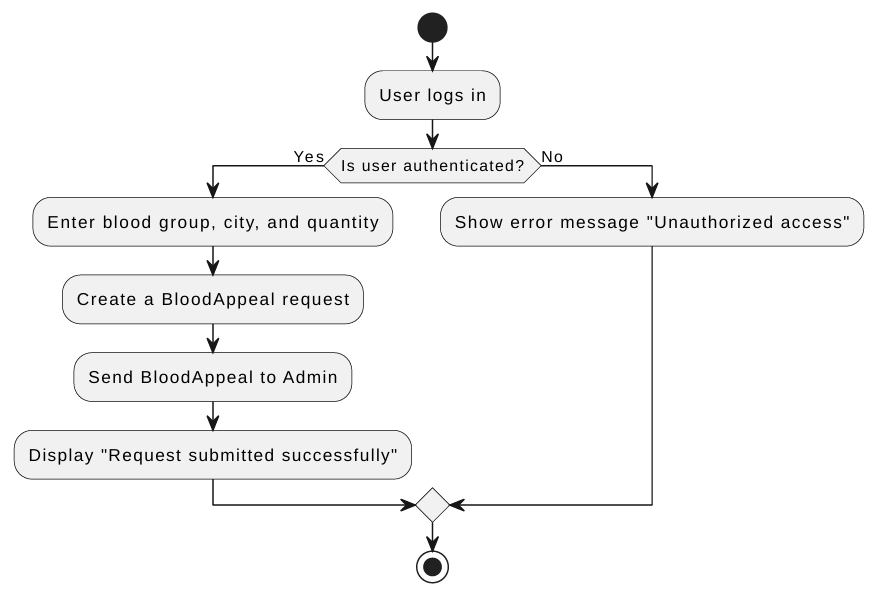


Figure 25: User Make Blood Appeals Activity Diagram

**Admin Features:**

* Login.

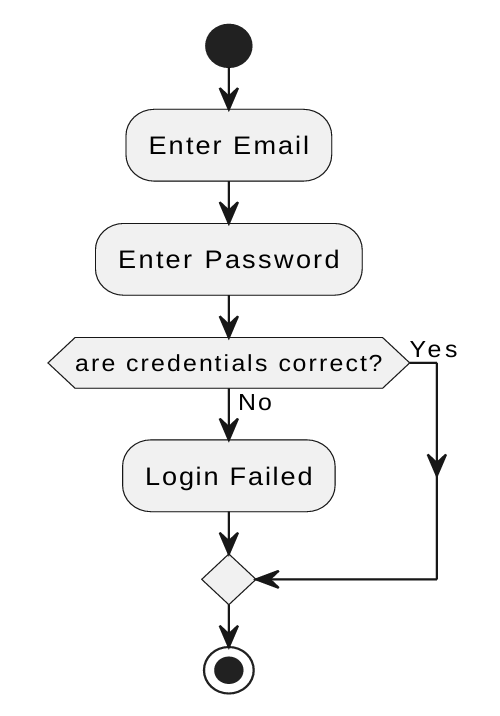


Figure 26: Admin Login Activity Diagram

* View Blood Inventory.

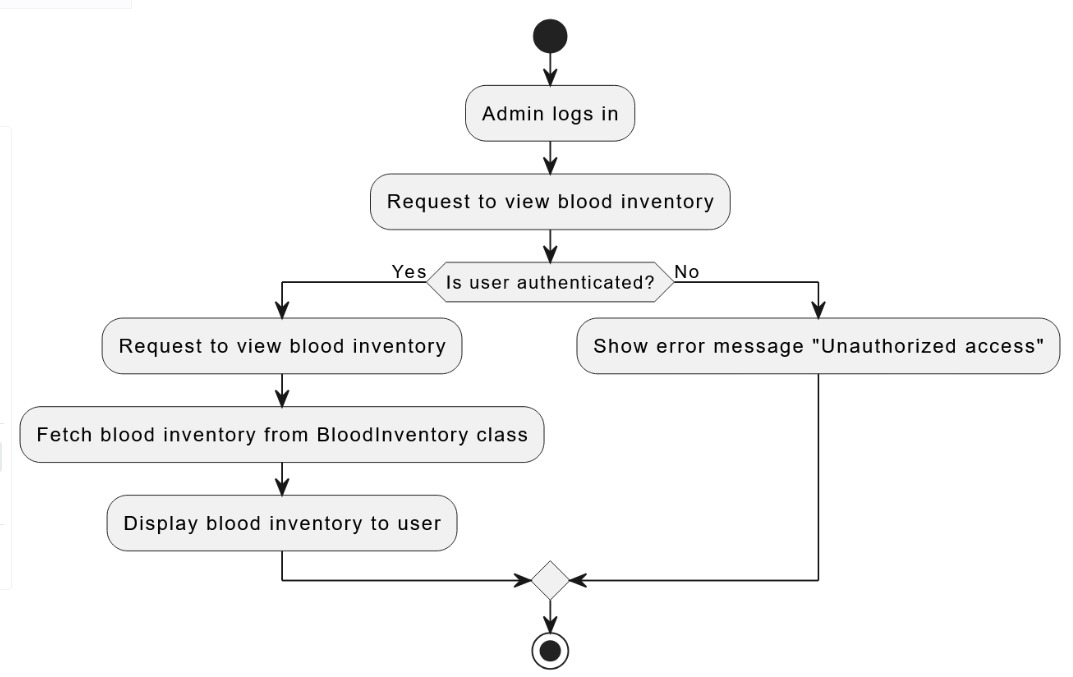


Figure 27: Admin View Blood Inventory Activity Diagram

* Manage/Update Blood Inventory.

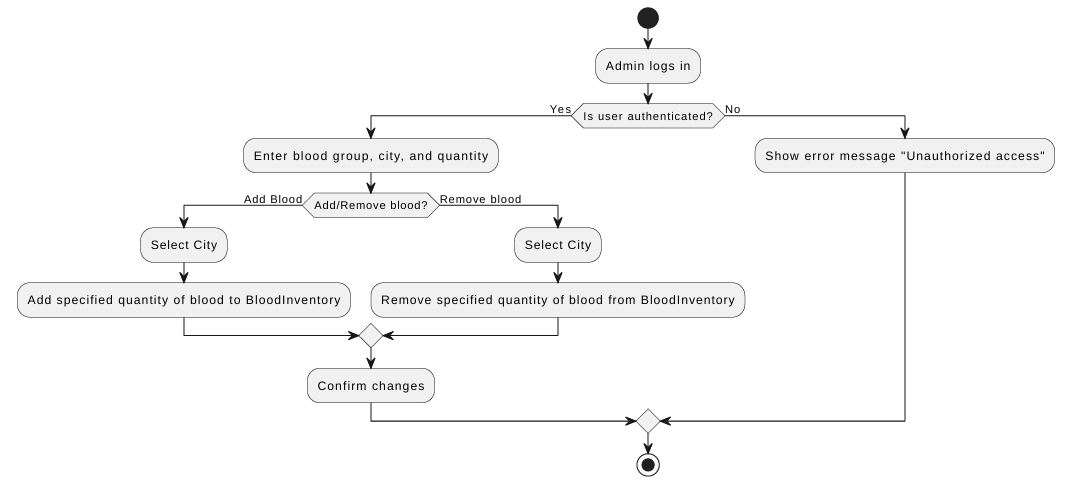


Figure 28: Admin Manage/Update Blood Inventory Activity Diagram

* Search Blood by City.

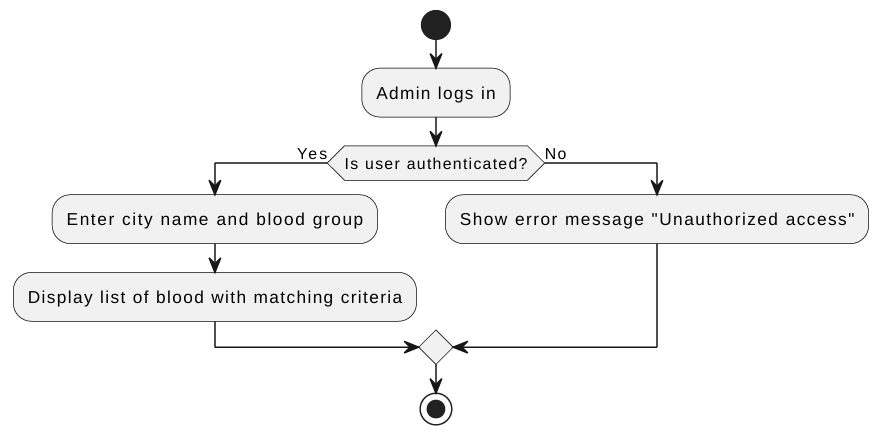


Figure 29: Admin Search Blood by City Activity Diagram

* Search Blood by Blood Group.

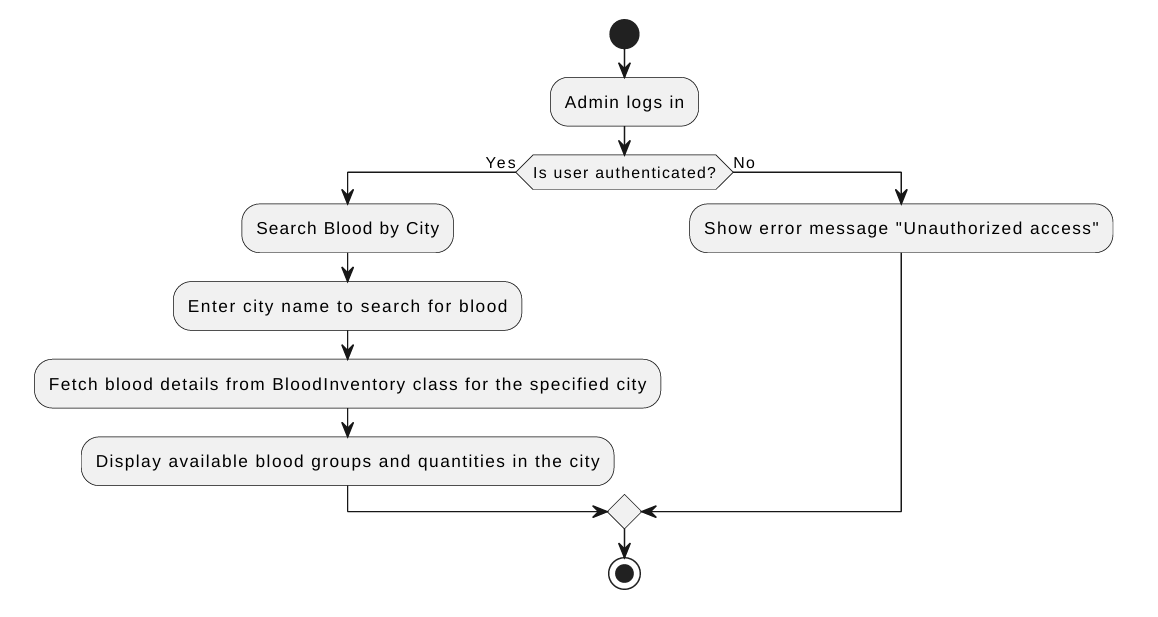


Figure 30: Admin Search Blood by Blood Group Activity Diagram

* Approve or Reject Blood Appeals.

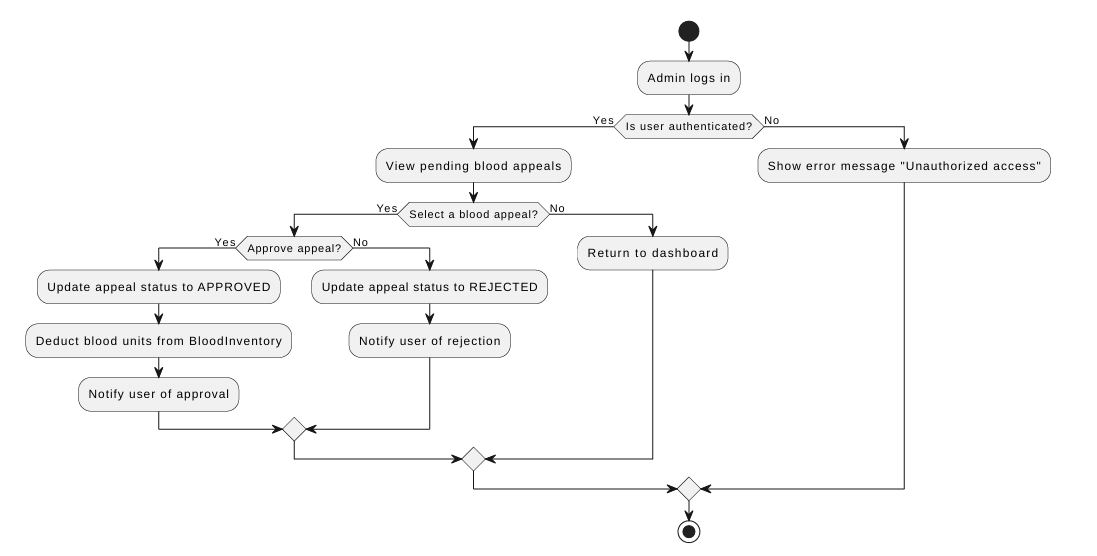


Figure 31: Admin Approve or Reject Blood Appeals Activity Diagram

* Approve or Reject Blood Donation Requests.

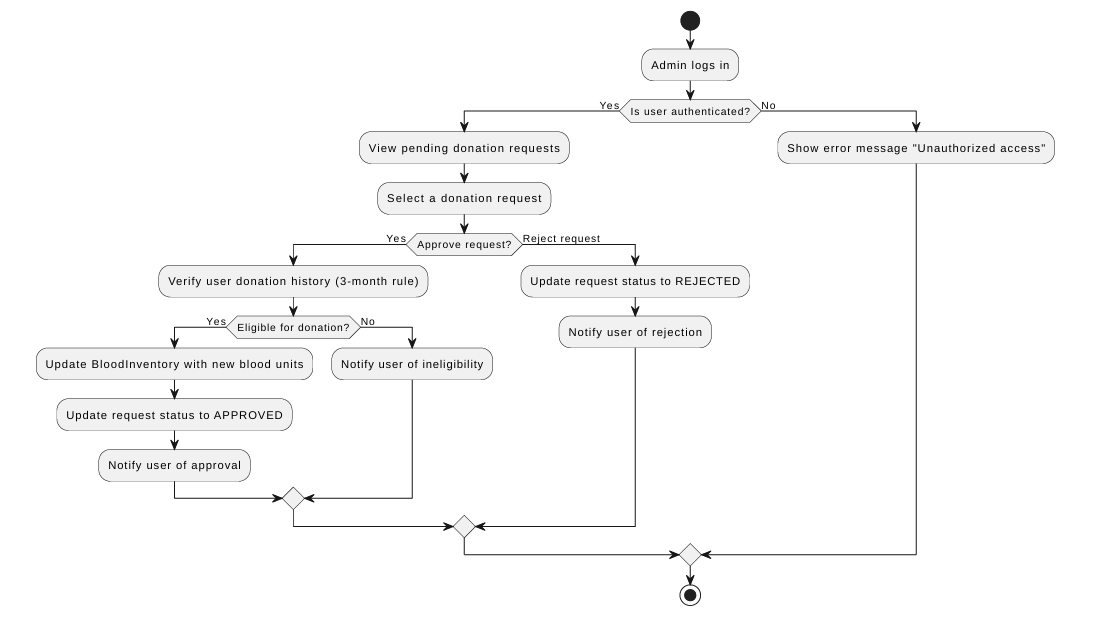


Figure 32: Admin Approve or Reject Blood Donation Requests Activity Diagram

## **9.6. Deployment Diagram**

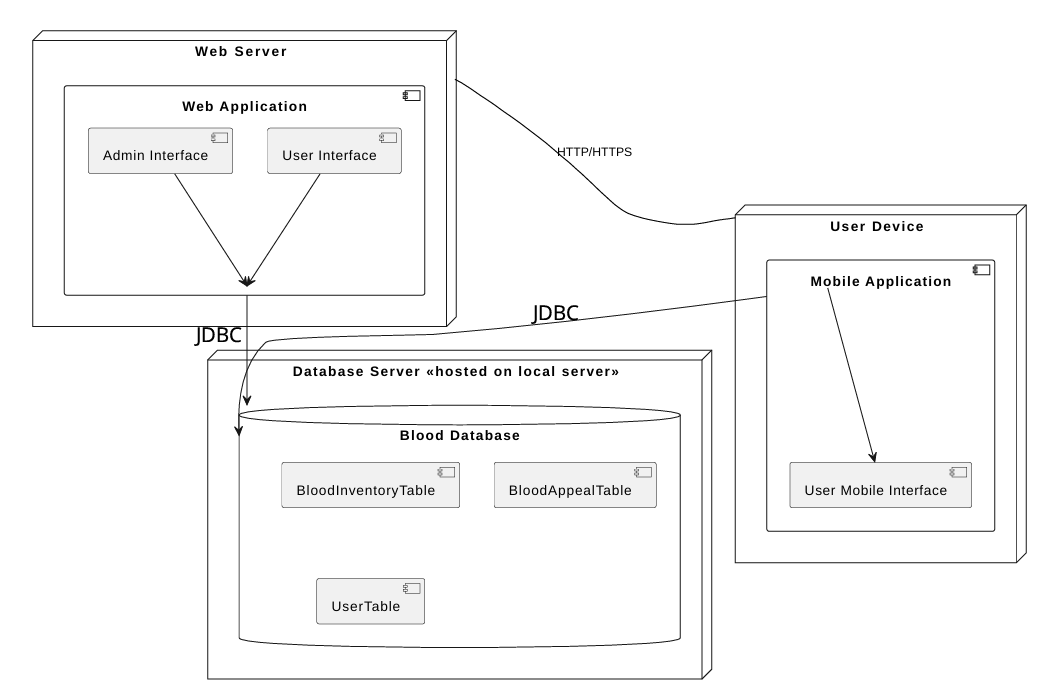


Figure 33: Deployment Diagram

## **9.7. Package Diagram**

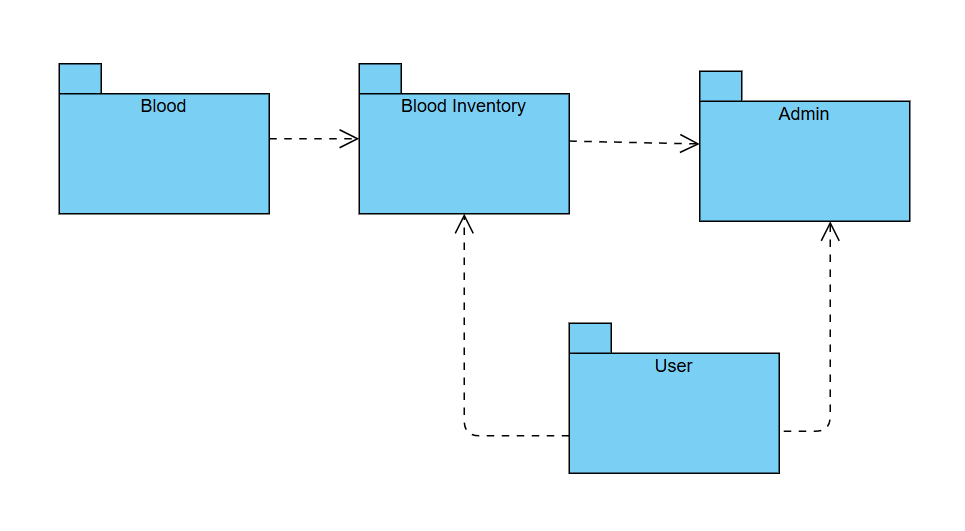


Figure 34: Package Diagram

## **9.8. State Machine Diagram**

**User Features:**

* Signup.

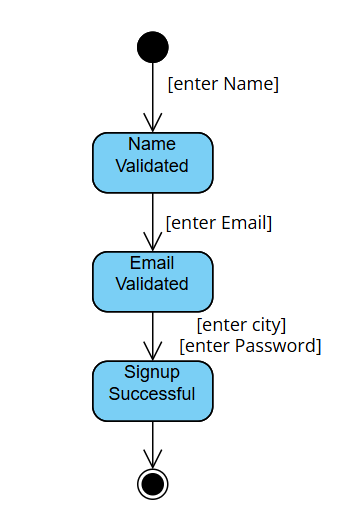


Figure 35: User Sign up State Machine Diagram

* Login

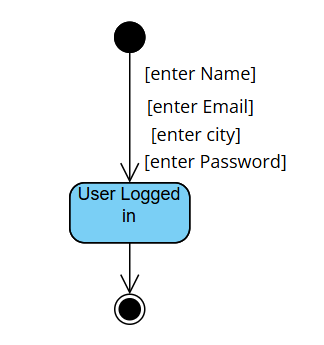


Figure 36: User Login State Machine Diagram

* View Blood Inventory.

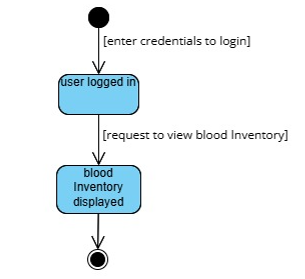


Figure 37: User View Blood Inventory State Machine Diagram

* Search Blood by City.

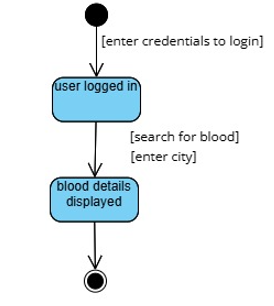


Figure 38: User Search Blood by City State Machine Diagram

* Search Blood by Blood Group.

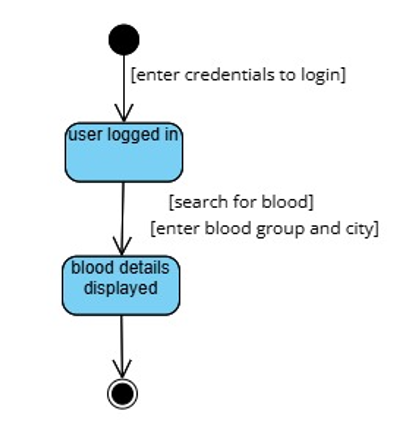


Figure 39: User Search Blood by Blood Group State Machine Diagram

* Request Blood Donation.

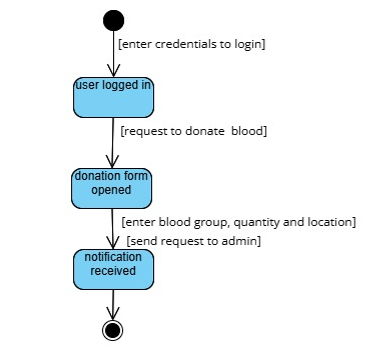


Figure 40: User Request Blood Donation State Machine Diagram

* Make Blood Appeals.

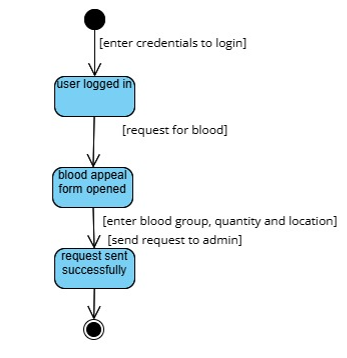


Figure 41: User Make Blood Appeals State Machine Diagram

**Admin Features:**

* Login.

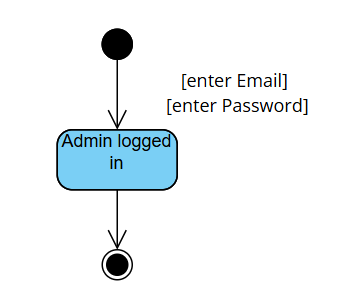


Figure 42: Admin Login State Machine Diagram

* View Blood Inventory.

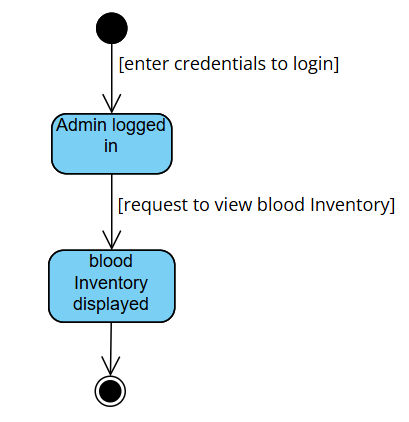


Figure 43: Admin View Blood Inventory State Machine Diagram

* Manage Blood Inventory.

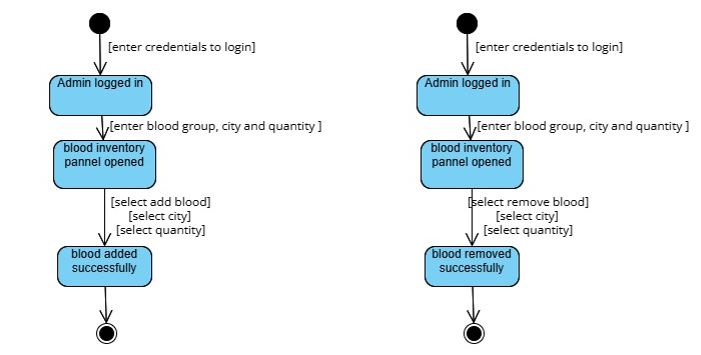


Figure 44: Admin Manage Blood Inventory State Machine Diagram

* Search Blood by City.

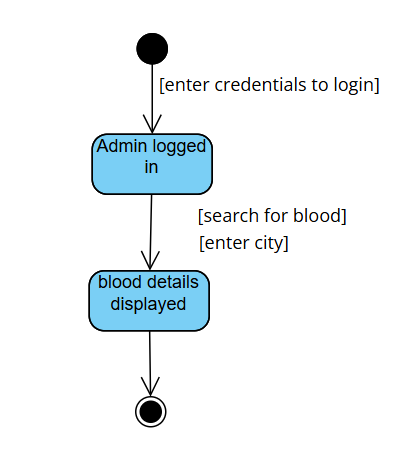


Figure 45: Admin Search Blood by City State Machine Diagram

* Search Blood by Blood Group.

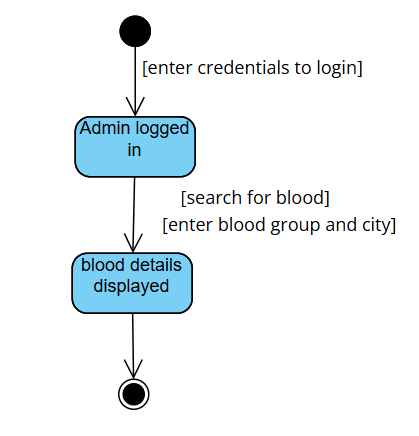


Figure 46: Admin Search Blood by Blood Group State Machine Diagram

* Approve or Reject Blood Appeals.

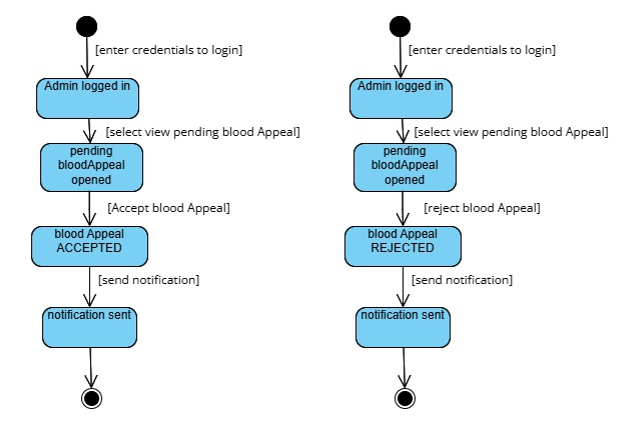


Figure 47: Admin Approve or Reject Blood Appeals State Machine Diagram

* Approve or Reject Blood Donation Requests.

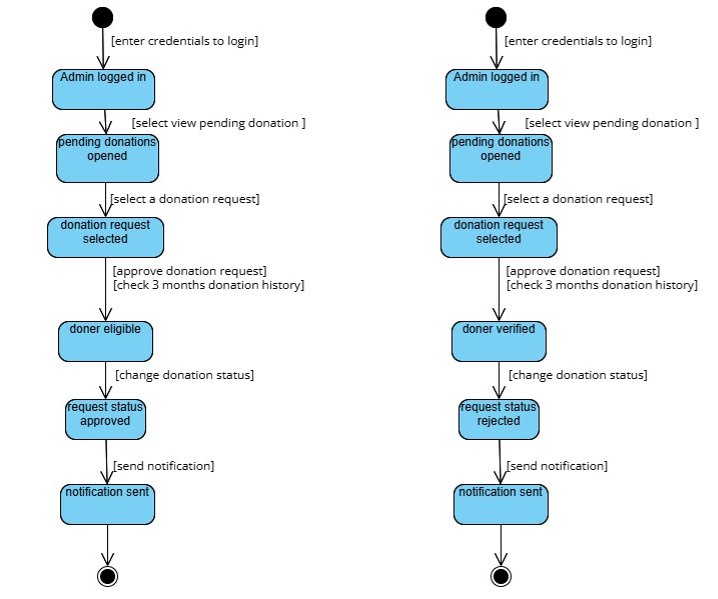


Figure 48: Admin Approve or Reject Blood Donation Requests State Machine Diagram

## **9.9. Collaboration Diagram**

**User Features:**

* Signup

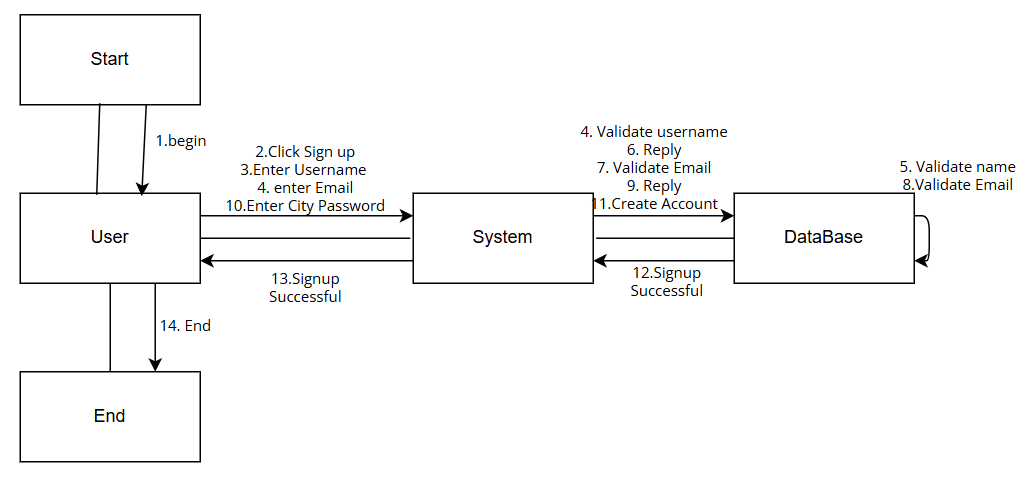


Figure 49: User Sign up Collaboration Diagram

* Login.

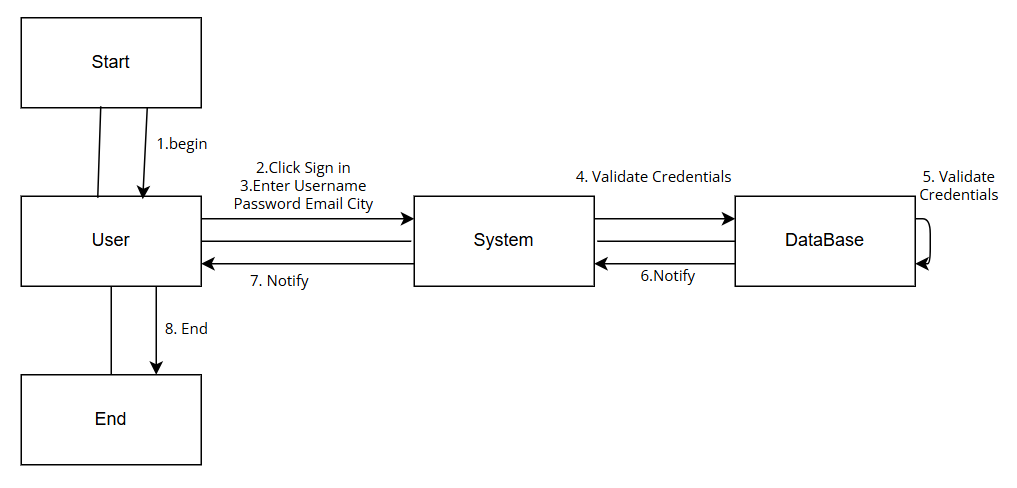


Figure 50: User Login Collaboration Diagram

* View Blood Inventory.

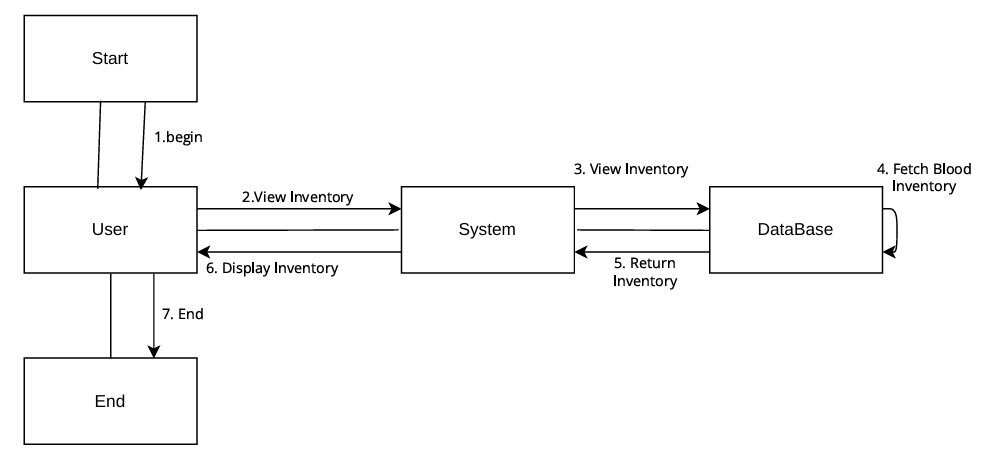


Figure 51: User View Blood Inventory Collaboration Diagram

* Search Blood by City.

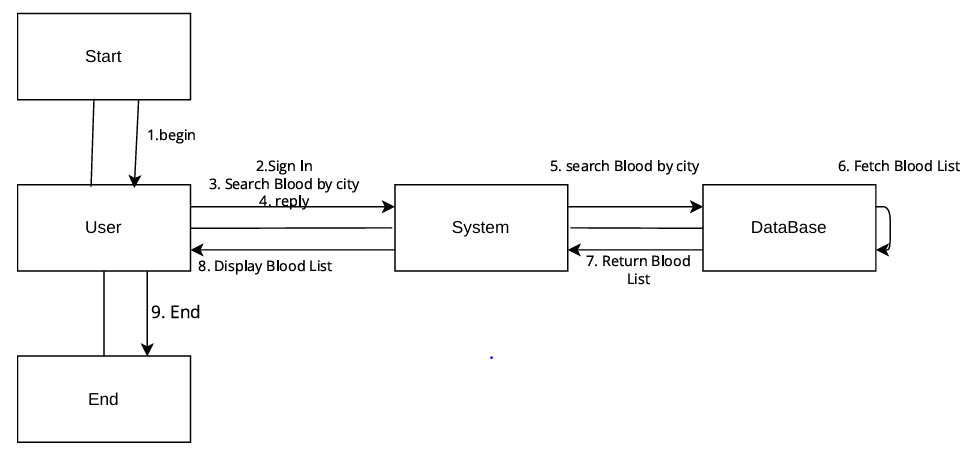


Figure 52: User Search Blood by City Collaboration Diagram

* Search Blood by Blood Group.

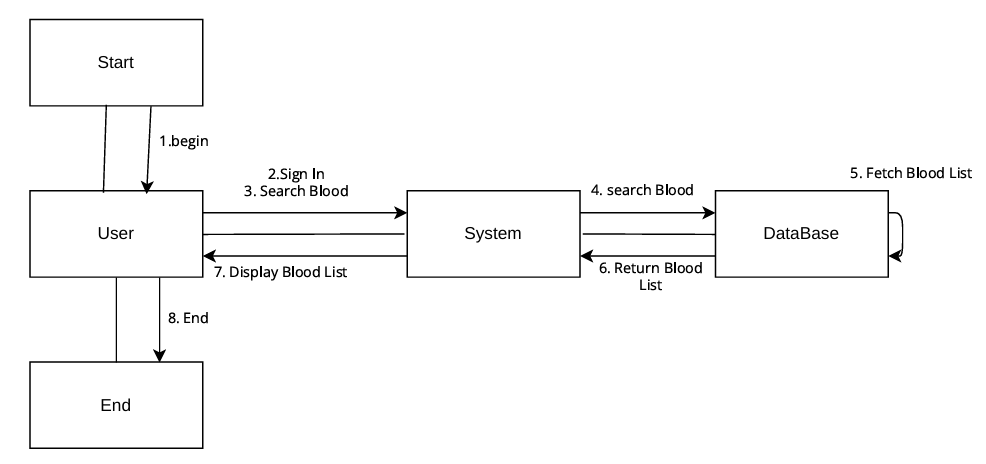


Figure 53: User Search Blood by Blood Group Collaboration Diagram

* Request Blood Donation.

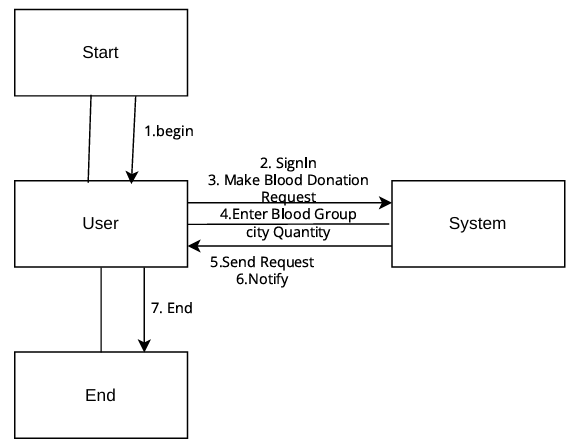


Figure 54: User Request Blood Donation Collaboration Diagram

* Make Blood Appeals.

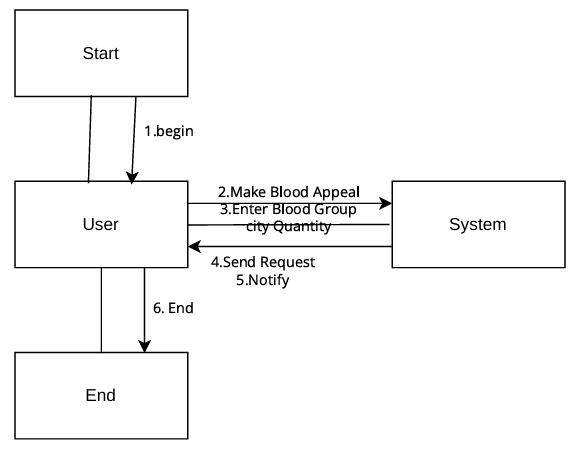


Figure 55: User Make Blood Appeals Collaboration Diagram

**Admin Features:**

* Login.

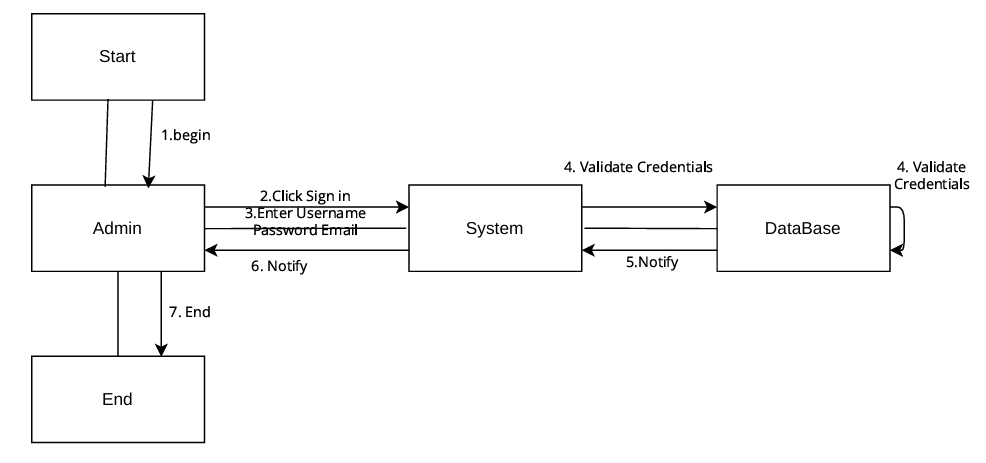


Figure 56: Admin Login Collaboration Diagram

* View Blood Inventory.

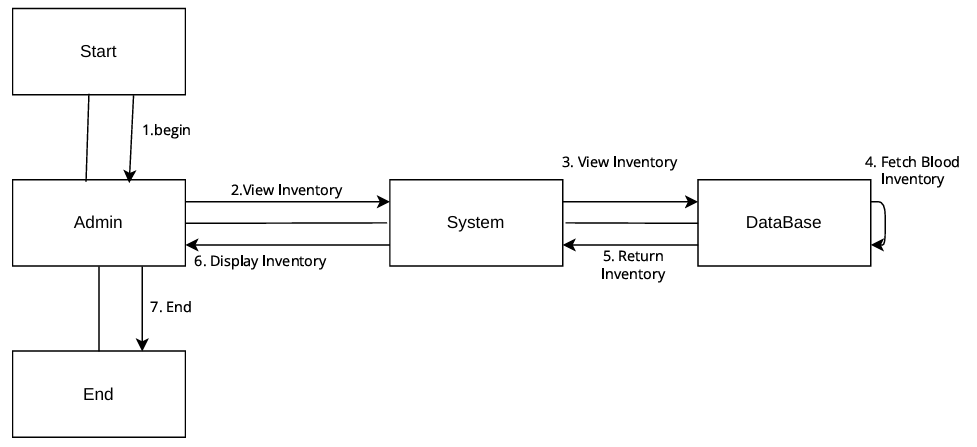


Figure 57: Admin View Blood Inventory Collaboration Diagram

* Manage Blood Inventory.

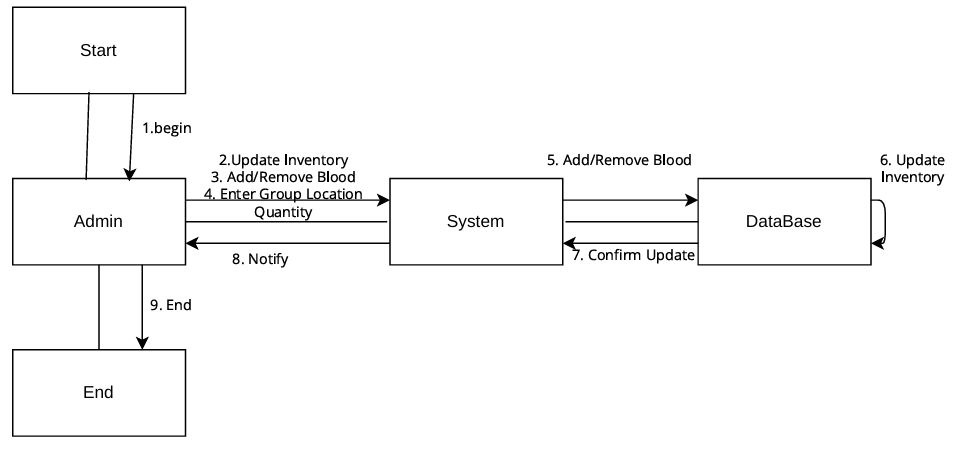


Figure 58: Admin Manage Blood Inventory Collaboration Diagram

* Search Blood by City.

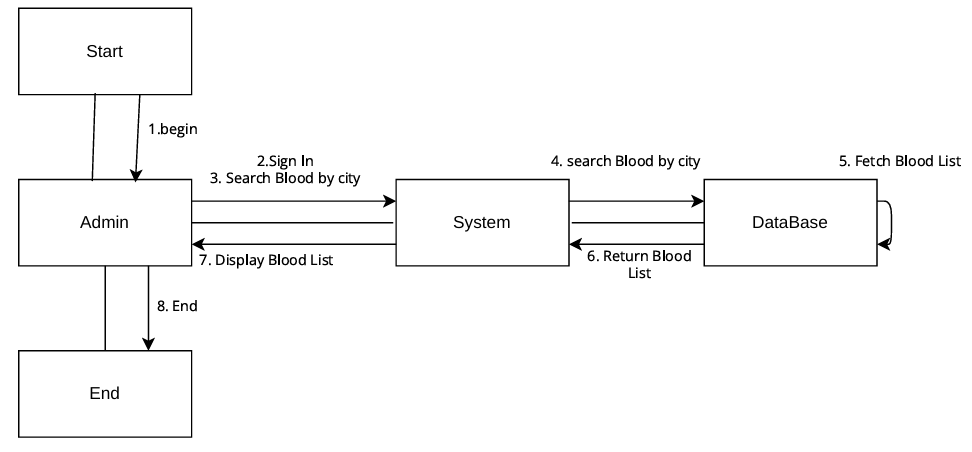


Figure 59: Admin Search Blood by City Collaboration Diagram

* Search Blood by Blood Group.

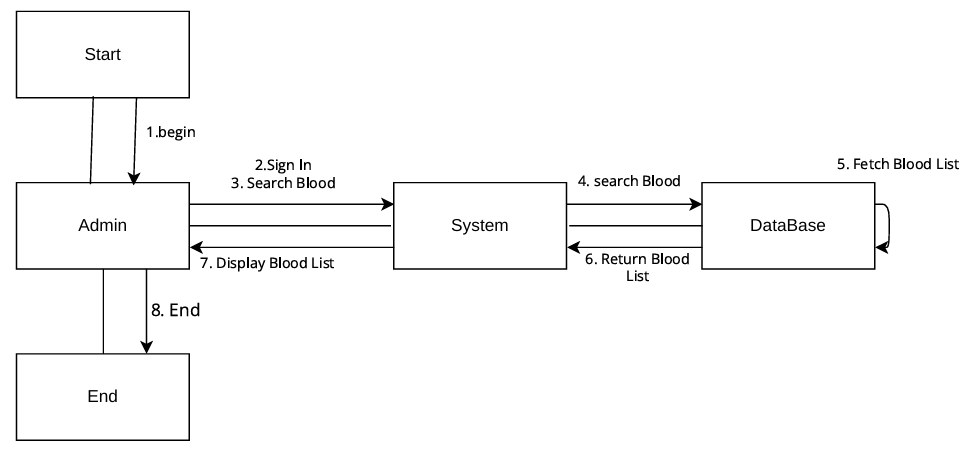


Figure 60: Admin Search Blood by Blood Group Collaboration Diagram

* Approve or Reject Blood Appeals.

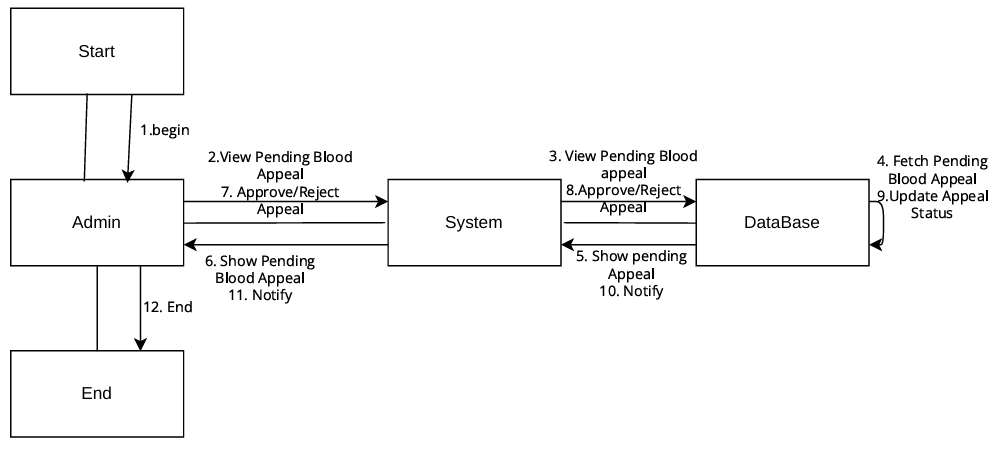


Figure 61: Admin Approve or Reject Blood Appeals Collaboration Diagram

* Approve or Reject Blood Donation Requests.

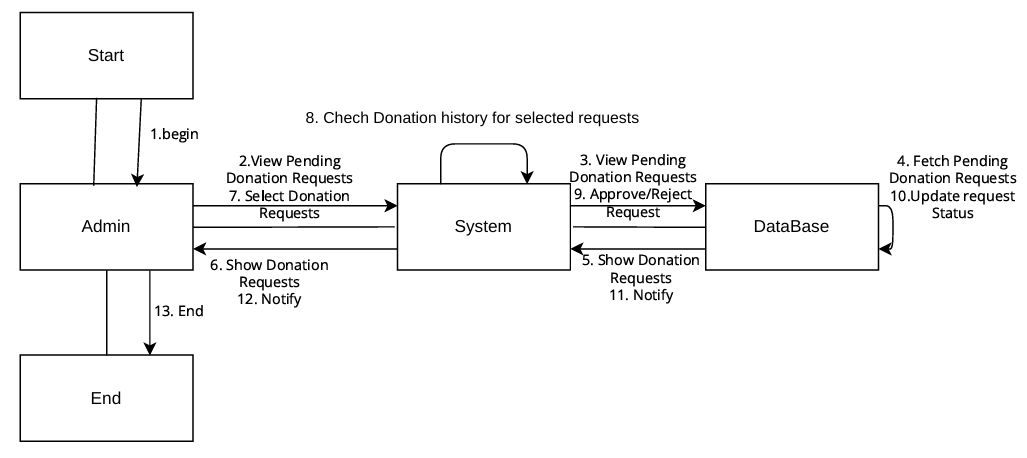


Figure 62: Admin Approve or Reject Blood Donation Requests Collaboration Diagram

## **9.10. Component Diagram**

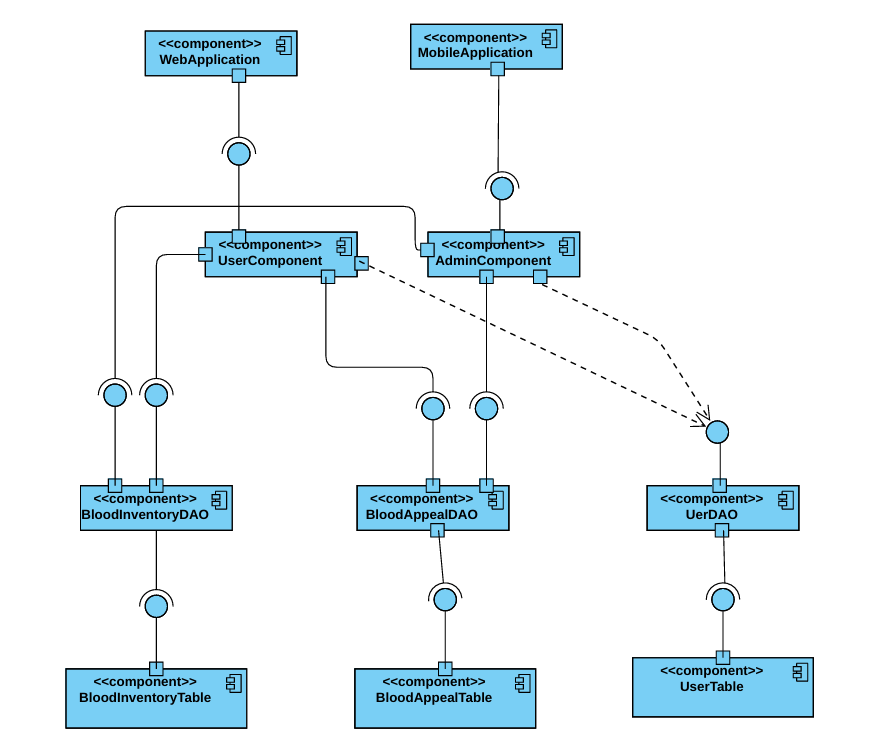


Figure 63: Component Diagram

# **10. User Interface Design (UI)**

## **10.1. User Wireframes**



Figure 64: User Sign up Wireframe.

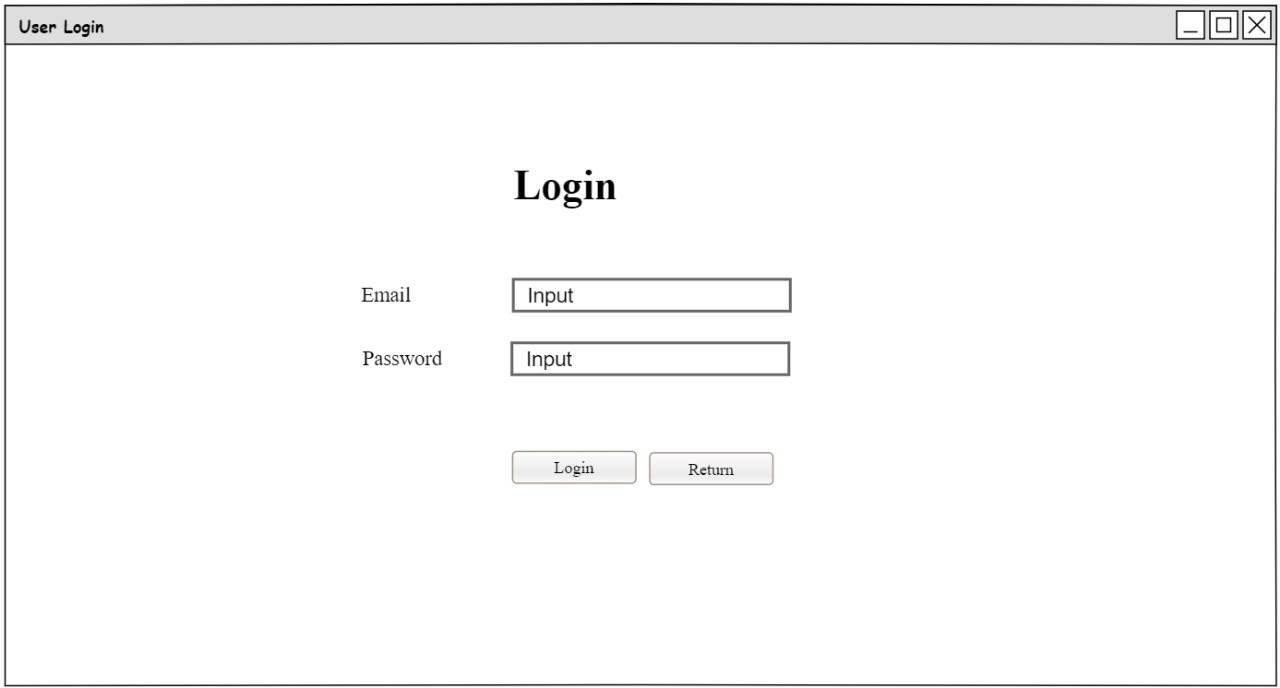


Figure 65: User Login Wireframe

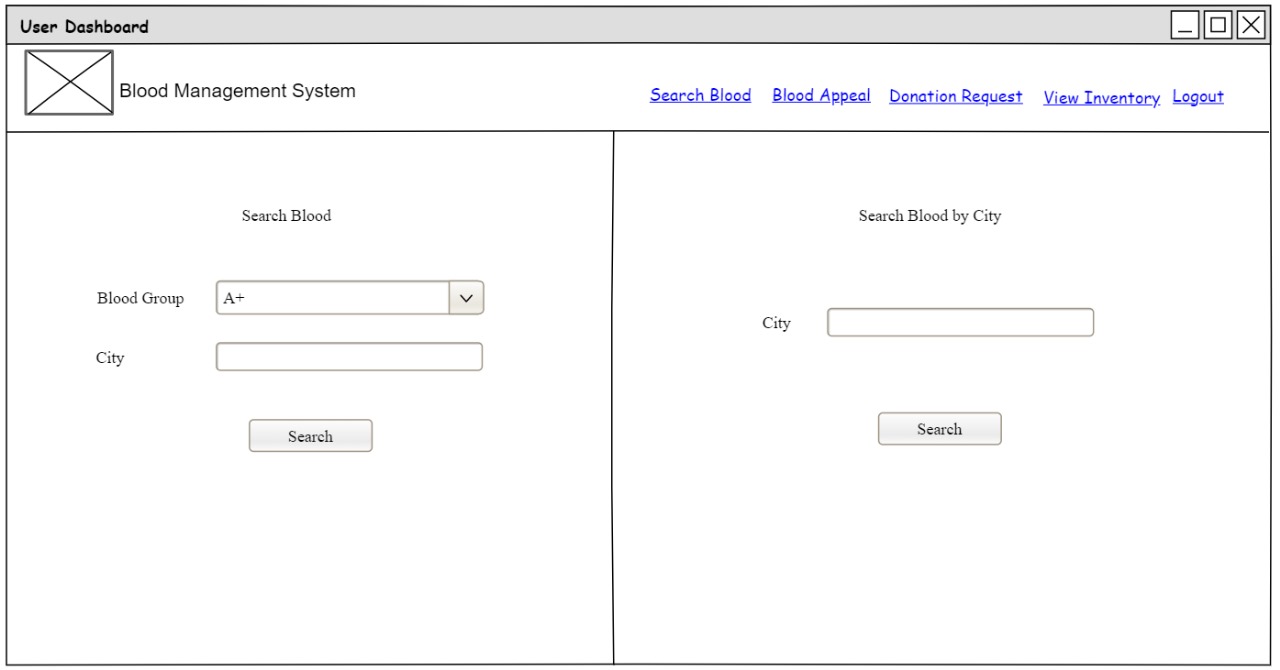


Figure 66: User Search Blood Wireframe

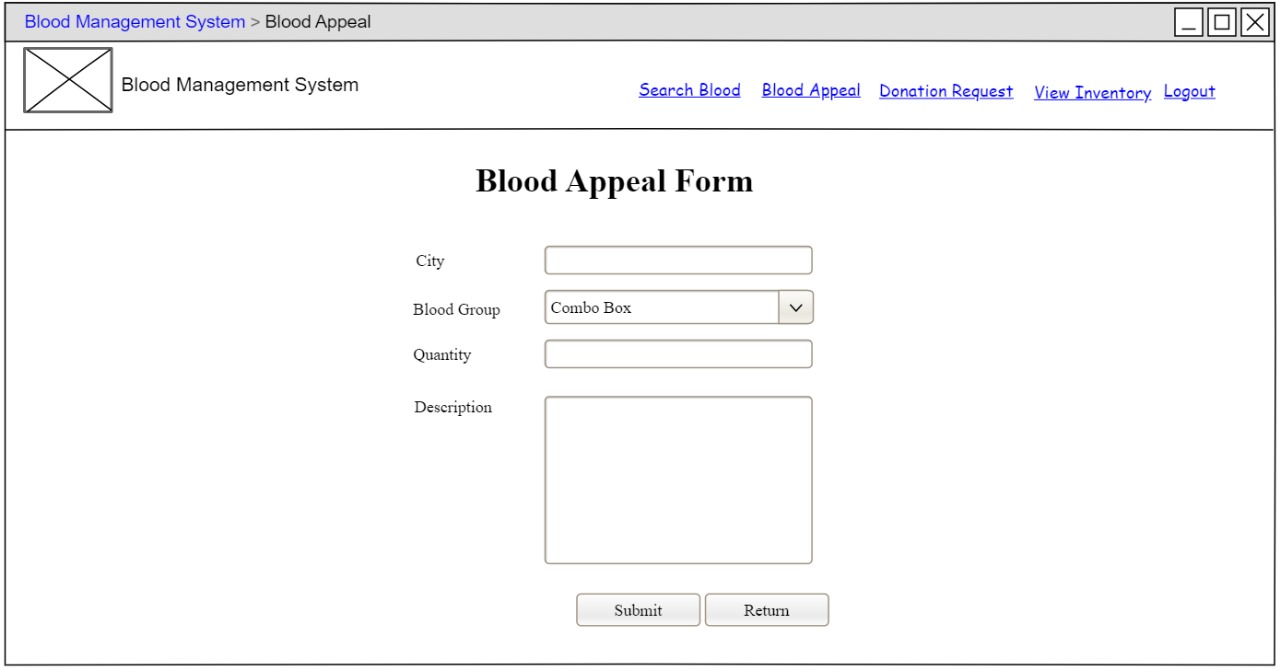


Figure 67: User make Blood Appeal Wireframe



Figure 68: User Donation Request Wireframe

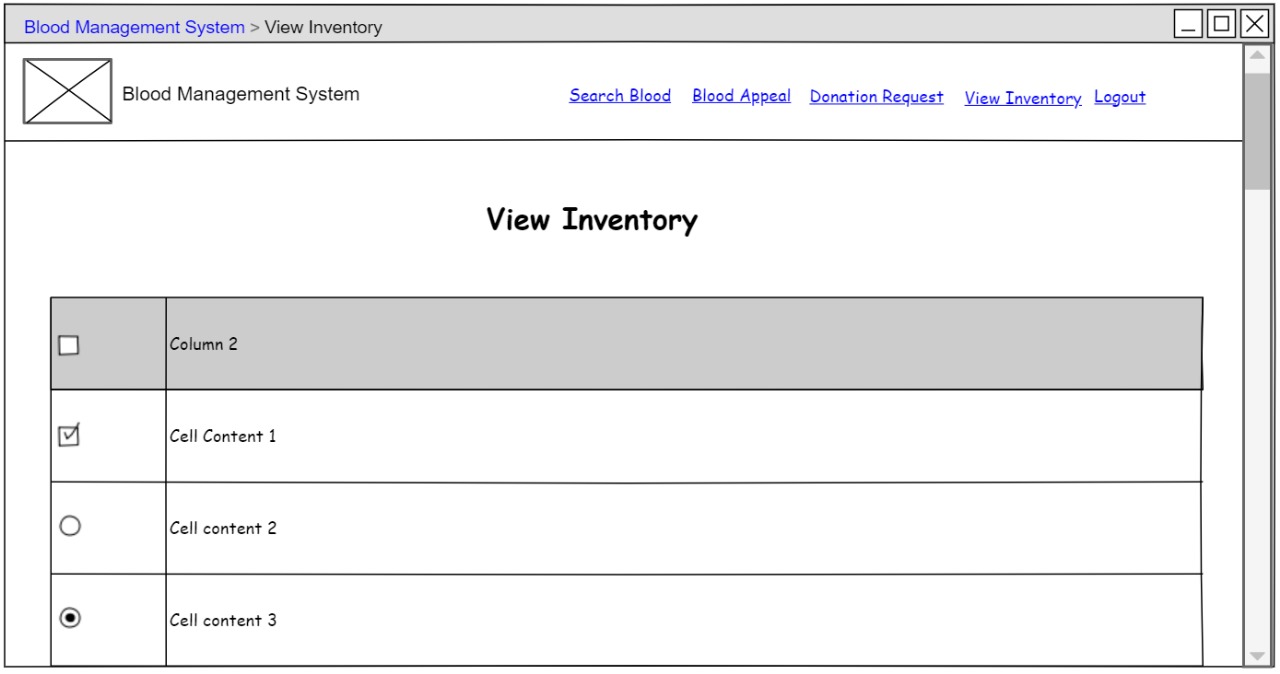


Figure 69: User View Blood Inventory Wireframe

## **10.2. Admin Wireframes**

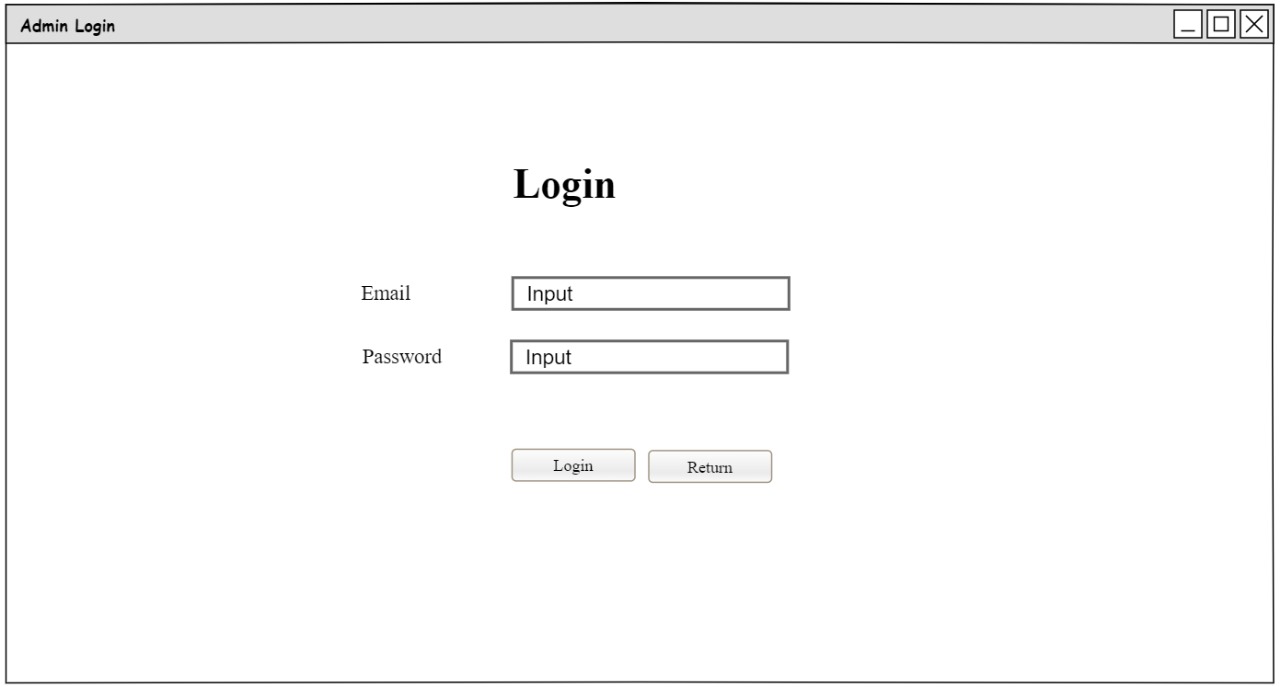


Figure 70: Admin Login Wireframe

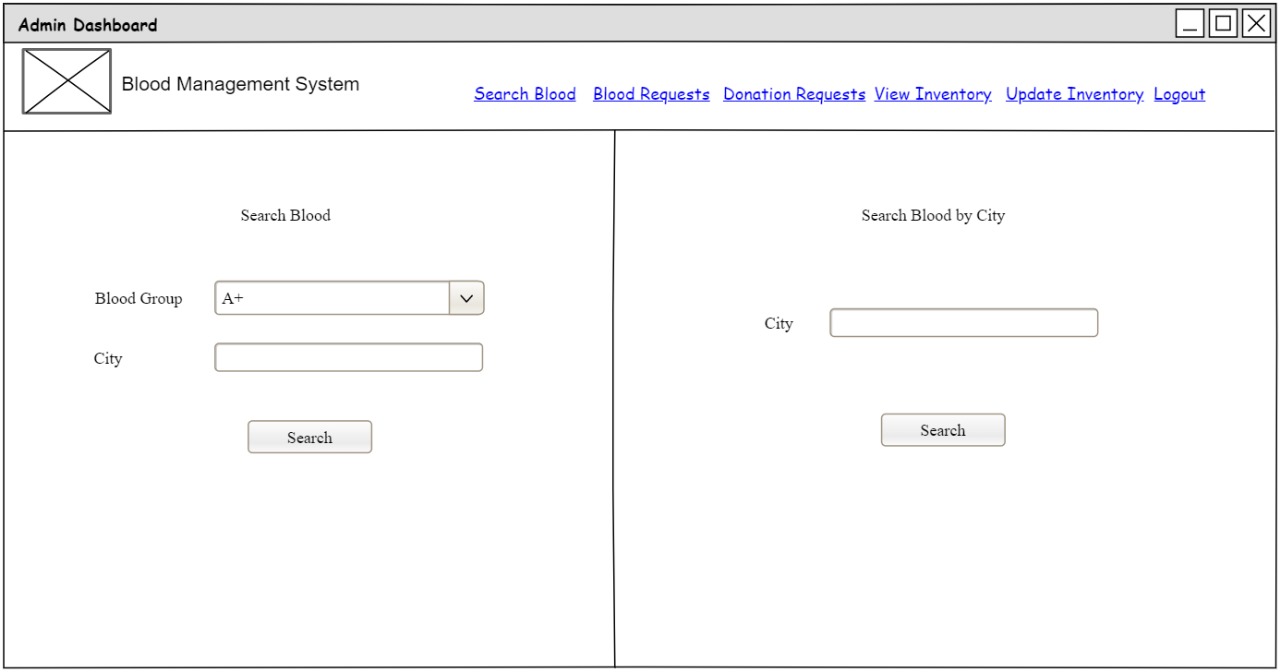


Figure 71: Admin Search Blood Wireframe

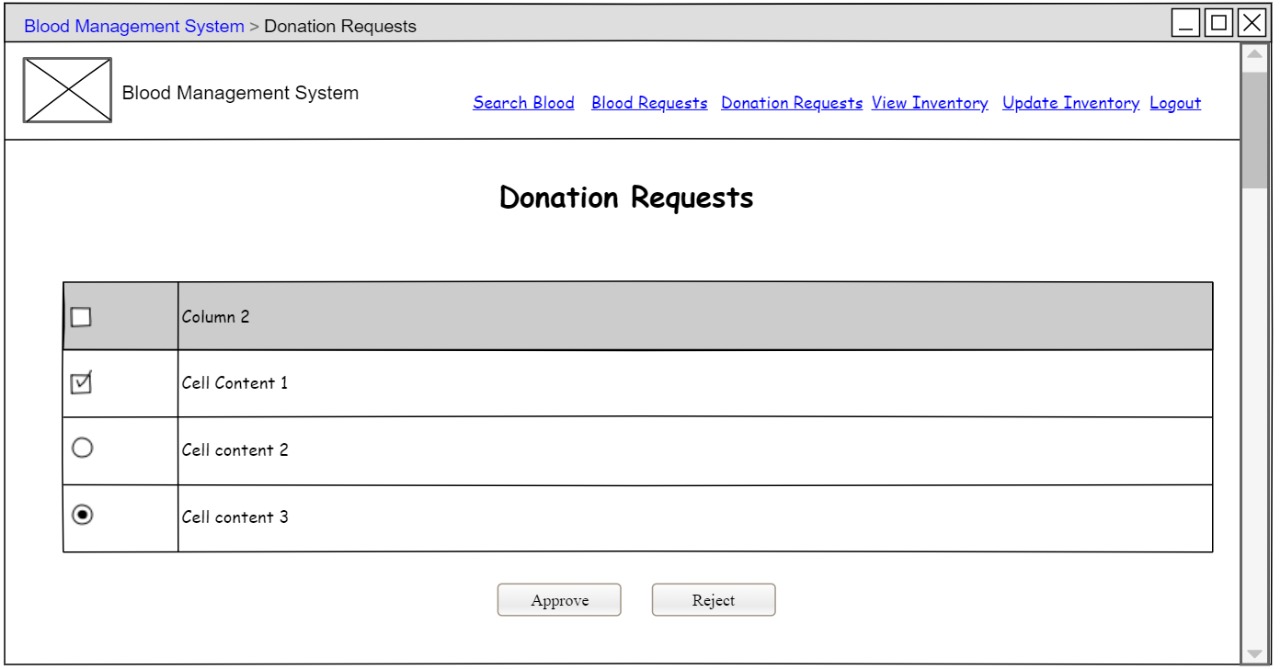


Figure 72: Admin Accept/Reject Donation Requests Wireframe

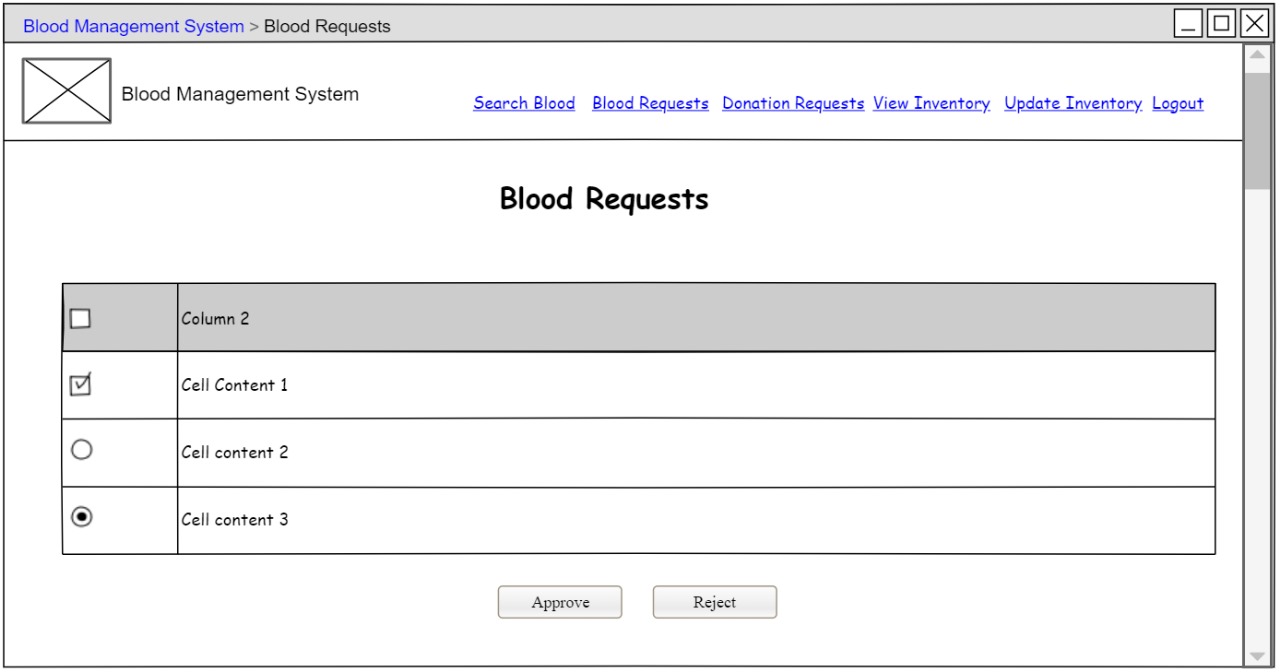


Figure 73: Admin Accept/Reject Blood Appeals Wireframe

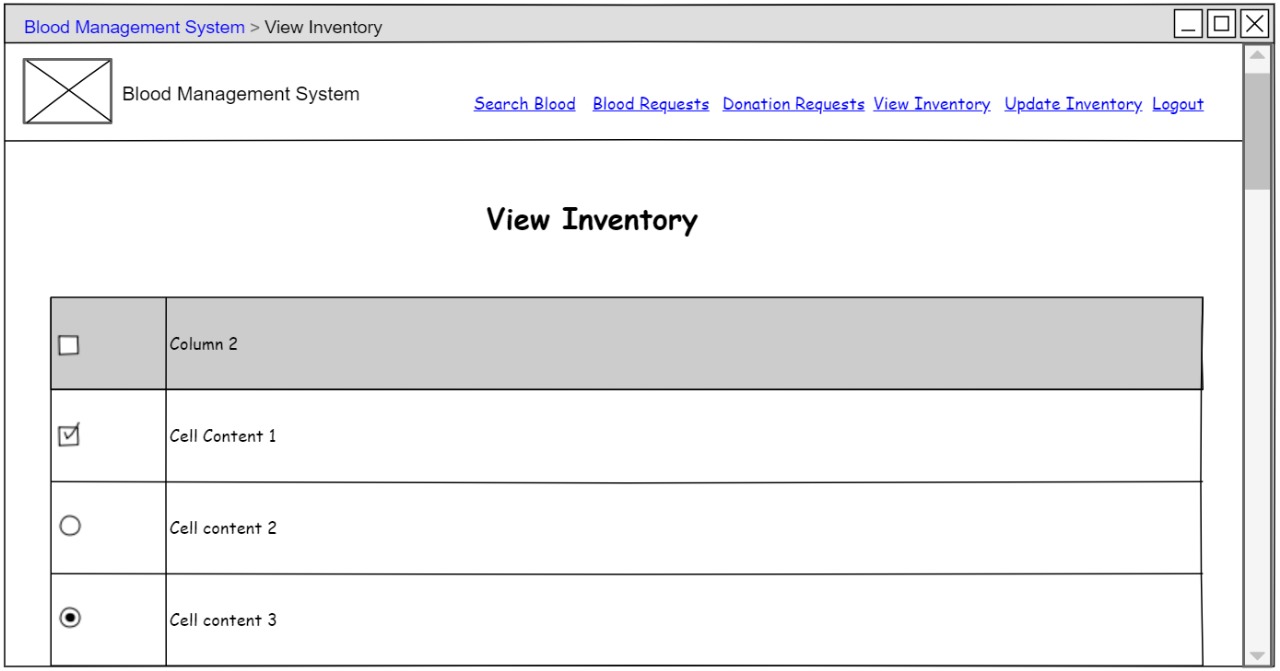


Figure 74: Amin View Inventory Wireframe

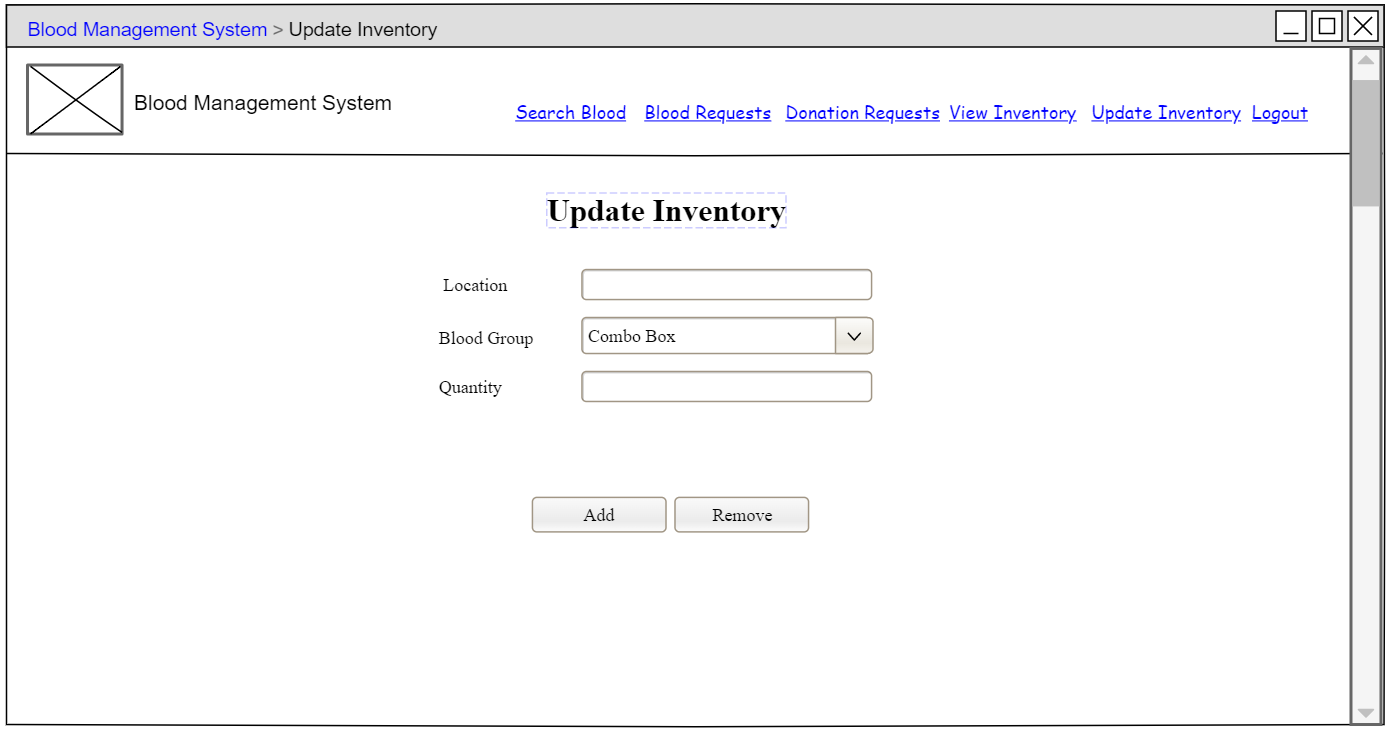


Figure 75: Admin Update Inventory Wireframe

# **11. Technology Stack**

## **11.1. Backend**

* **Language:** Java
* **Framework:** Spring Boot
* **Database:** MySQL
* **Architecture:** RESTful APIs
* **Libraries/Tools:** Hibernate, Lombok, and Logback.

## **11.2. Frontend**

* Language: JavaScript (with JSX syntax)
* **Framework:** React.js
* **Libraries: Tailwind CSS, and Daisy UI**.

# **12. Security**

* User passwords will be hashed using bcrypt.
* Data validation on both client and server sides.
* Role-based access control (RBAC) for admins and users.

# **13. Deployment**

* **Server:** AWS EC2 instance for hosting the backend.
* **Frontend:** Hosted on AWS S3 or a similar service.
* **Database:** AWS RDS with MySQL.

# **14. Testing**

* **Unit Testing:** JUnit for backend logic.
* **Integration Testing:** Postman and Rapid API.
* **Frontend Testing:** Selenium.
* **Stress Testing:** Apache JMeter.

# **15. Future Enhancements**

* Using TypeScript in Frontend for robust development and easy debugging.
* Adding support for donor registration and management.
* Enabling SMS/email notifications for requests and appeals.
* Integrating location-based services for better blood search results.

# **16. References**

UML Notations

Best practices for database schema design.