

Software Engineering I

PROJECT NAME BLOOD BANK SYSTEM

Supervised By:

1. Prof. Marwa Refaai
2. T.A: Riham Fares
3. T.A: Abdelrahman Essam
4. T.A: Nada El Sayed
5. T.A: Nada Hussein
6. T.A: Dina Ahmed

Submitted By

Student Name
1) Omar Mahmoud Ahmed Khalil
2) Mohamed Osama Mostafa Ismail
3) Mohamed Ahmed Mahmoud Ahmed

Table Of Contents

- 1) Introduction**
- 2) Chapter 1: Problem Statement**
- 3) Chapter 2: Analysis & Design**
- 4) Chapter 3: UI & Prototype**
- 5) Chapter 4: System Testing**
- 6) Chapter 5: System Implementation**
- 7) Chapter 6: Tools**

Introduction

The following Section Provides an overview the software requirements specifications (SRS) for the

Blood Bank System

Purpose:

The purpose of SRS is to determine both the Functional and the Non-Functional requirements of software management system. Also, the documentation provided as over all descriptions about the

Blood Bank system with UML analysis models

Data collection:

Fundamental four phase model (planning, analysis, design, and implementation) common to all information system development projects.

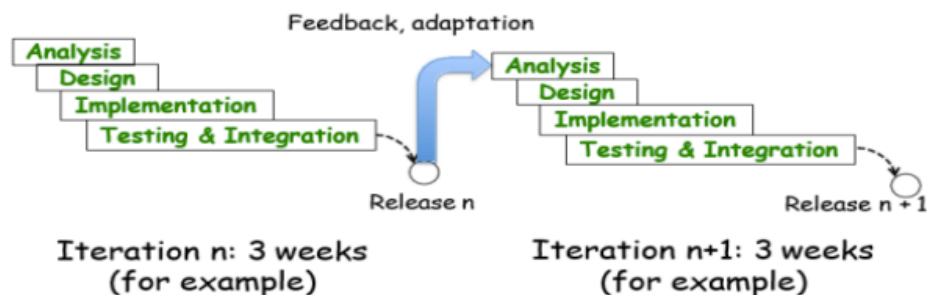
It describes the evolution of system development methodologies and discusses the roles and skills.

required of a system analyst.

Diagram of incremental mode used.

Incremental Development

► A sequential of waterfall models



Key Points of the Model :

Iterative: many releases/increments

- First increment: core functionality
- Successive increments: add/fix functionality
- Final increment: the complete product Require a complete definition of the whole system to break it down and build incrementally.

Pros & Cons :

Pros:

- Early discovery of software defects
- Early delivery of working software
- Less cost to change/identify requirements

Cons:

- Constant changes

Incremental Development Advantages:

1. Customer value can be delivered with each increment, so system functionality is available earlier
2. Early increments act as a prototype to help extraction of requirements for later increments
3. Lower risk of overall project failure
4. The highest priority system services tend to receive the most testing

When to use the model?

The requirements of the complete system are clear.

Major requirements must be defined while some details can evolve over time Need to get a product to the market early

Chapter 1: Problem Statement

Blood banks play a crucial role in healthcare by ensuring a stable and accessible supply of blood and blood-related products. To streamline and enhance the operations of a local blood bank, a Blood Bank Management System has been proposed and partially implemented. The system is designed to manage various aspects of blood donation, donor information, patient records, employee management, and blood stock monitoring.

System Overview:

The Blood Bank System is a comprehensive solution tailored to meet the specific needs of blood banks. The system comprises several interconnected modules, each dedicated to handling a specific aspect of blood bank operations.

These modules include:

1. Donor Management:

- Allows individuals to register as donors, providing personal details and blood group information.
- Facilitates the donation process, recording donor contributions and updating the blood stock.

2. Employee Dashboard:

- Provides a secure login for authorized personnel, including employees and administrators.
- Enables employees to manage donor and patient information, monitor blood stock levels, and perform administrative tasks.

3. Admin Panel:

- Exclusive access for administrators to perform high-level administrative functions.
- Empowers administrators to manage employee information, ensuring the integrity and security of the system.

4. Blood Stock Monitoring:

- Enables employees to monitor the current stock levels of various blood groups.
- Supports informed decision-making by providing real-time information on blood availability.

5. Patient Management:

- Allows for the efficient management of patient records, including additions updates.
- Integrates patient information seamlessly with the broader blood bank ecosystem.

Existing Modules and Functionality:

The system's existing implementation includes functional modules such as donor registration, employee login, blood stock monitoring, patient management, and more. Each module is designed to contribute to the overall efficiency and effectiveness of blood bank operations.

Role:

As a software engineer tasked with the completion of the Blood Bank System, our responsibilities include:

- Analyzing and enhancing existing code to ensure optimal functionality and security.
- Implementing additional features and modules as needed.
- Conducting thorough testing to identify and resolve any potential issues.
- Collaborating with stakeholders to gather requirements for future enhancements.

Chapter 2: System Requirements

Functional Requirements:

1. Donor Management Module:

- Allow individuals to register as donors by providing personal details (name, contact information, etc.).
- Capture and record the blood group information of donors.
- Facilitate the blood donation process, recording donations and updating the blood stock.
- Generate a unique donor identification number for each registered donor.

2. Employee Dashboard:

- Provide a secure login mechanism for authorized employees.
- Grant access to modules for managing donors, patients, and blood stock.
- Display relevant statistics and alerts on the employee dashboard.

3. Admin Panel:

- Implement an exclusive login for administrators with elevated privileges.
- Allow administrators to manage employee information, including addition, modification, and deletion of records.
- Provide administrative functions such as system configuration and user role management.

4. Blood Stock Monitoring:

- Display real-time information on current stock levels for different blood groups.
- Generate alerts or notifications when stock levels fall below a predefined threshold.
- Support querying of historical blood stock data for reporting purposes.

5. Patient Management:

- Allow employees to manage patient records, including additions and updates.
- Integrate patient information seamlessly with other modules in the system.
- Provide the ability to search and retrieve patient records efficiently.

Non-Functional Requirements:

1. Security:

- Implement secure authentication mechanisms for all user roles.
- Encrypt sensitive data, such as personal information and medical records.
- Maintain an access log for all system interactions to ensure accountability.

2. Performance:

- The system should respond to user interactions within 2 seconds under normal operating conditions.
- Support concurrent user access without significant degradation in performance.

3. Scalability:

- Design the system to accommodate an increase in the number of donors, patients, and employees without a significant impact on performance.

4. Usability:

- Ensure a user-friendly interface with intuitive navigation for all modules.
- Provide context-sensitive help and tooltips for users to understand system functionalities.

5. Reliability:

- The system should be available 99.9% of the time, allowing for scheduled maintenance.

These requirements will serve as a foundation for the design, development, and testing of the Blood Bank System, ensuring a robust and efficient solution.

Use Case: Blood Bank Operations

Scenario:

The donor launches the system, selects "Donate Blood," enters personal details, and blood group, then confirms the donation. System Records the donation, issues a confirmation, and updates the blood stock.

Employee Logs into the system, accesses the dashboard to manage donors, patients, and blood stock. System Validates credentials, grants access to the dashboard.

Admin Logs in, accesses the admin panel, manages employee information, and performs administrative tasks. System Validates admin credentials, provides access to admin functions.

Employee Monitors blood stock selects "Blood Stock," and reviews current levels of different blood groups. System Displays current stock levels, enabling informed decision-making.

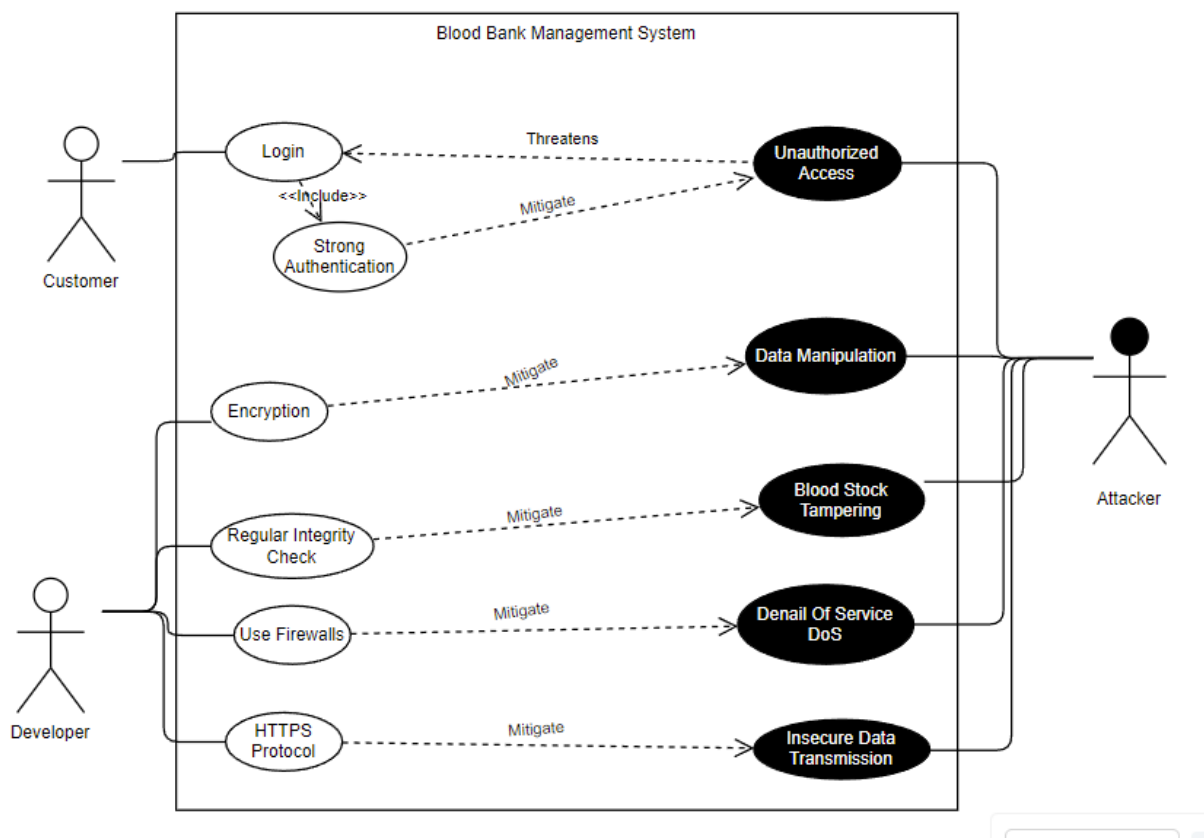
Employee Initiates blood transfer selects a patient, and completes the transfer process. System Records the transfer, updates stock, and notifies relevant parties. Employee Views detailed information about donors, selects "View Donors," and reviews comprehensive donor details. System Fetches and displays detailed donor information for reference or reporting.

Employee Manages patient information, selects "Patients," and updates or adds new patient records. System Allows the employee to manage patient records efficiently.



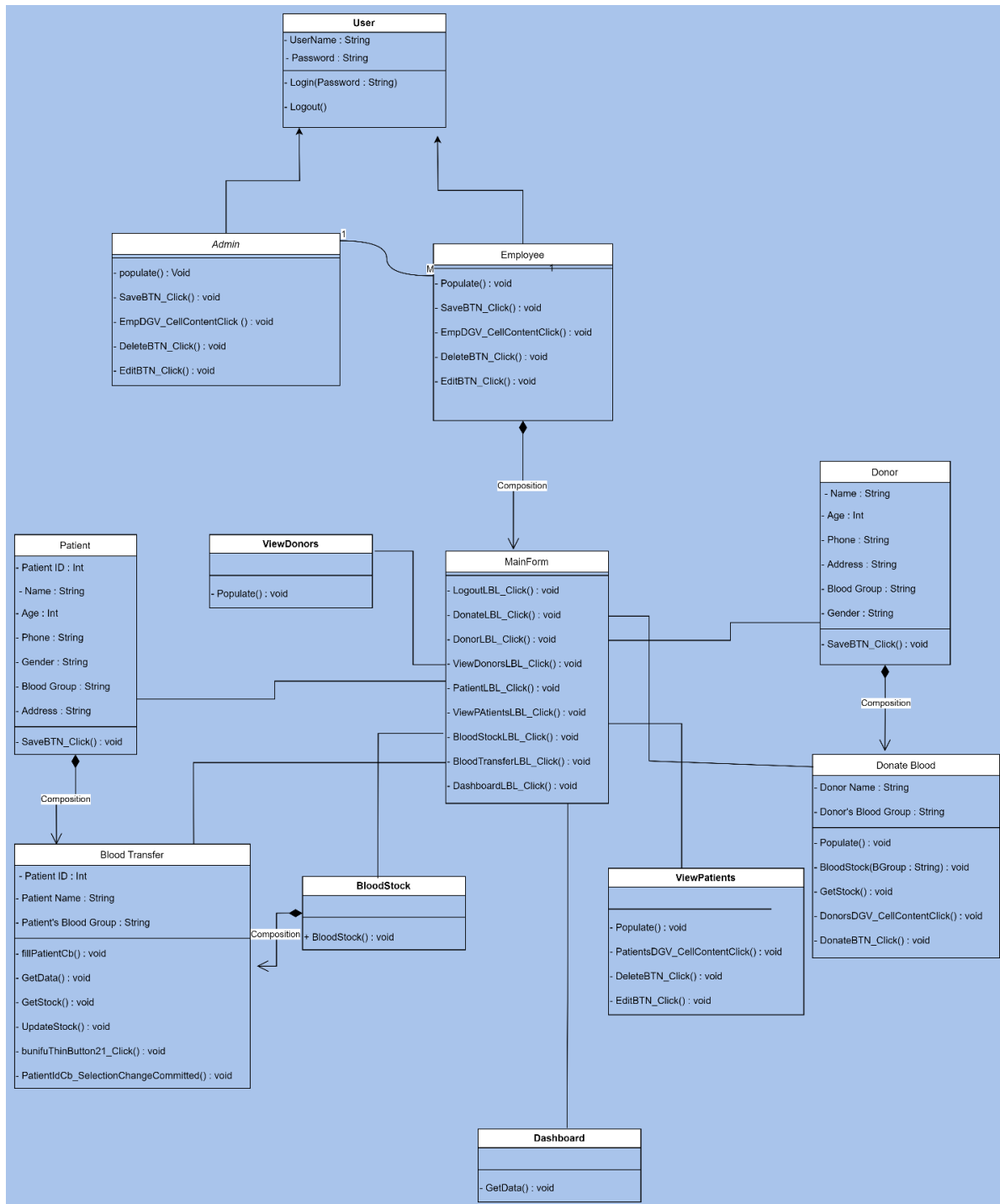
Misuse Case :

- An attacker attempts to gain unauthorized access to the system, potentially by exploiting vulnerabilities or using brute force attacks.
- An unauthorized user tries to manipulate or alter data within the system, such as modifying donor or patient records.
- An unauthorized user tries to manipulate blood stock levels, potentially causing misinformation or shortages.
- An attacker floods the system with requests, causing it to become unresponsive and denying service to legitimate users.
- Data transmitted between the client and server is intercepted by an attacker.



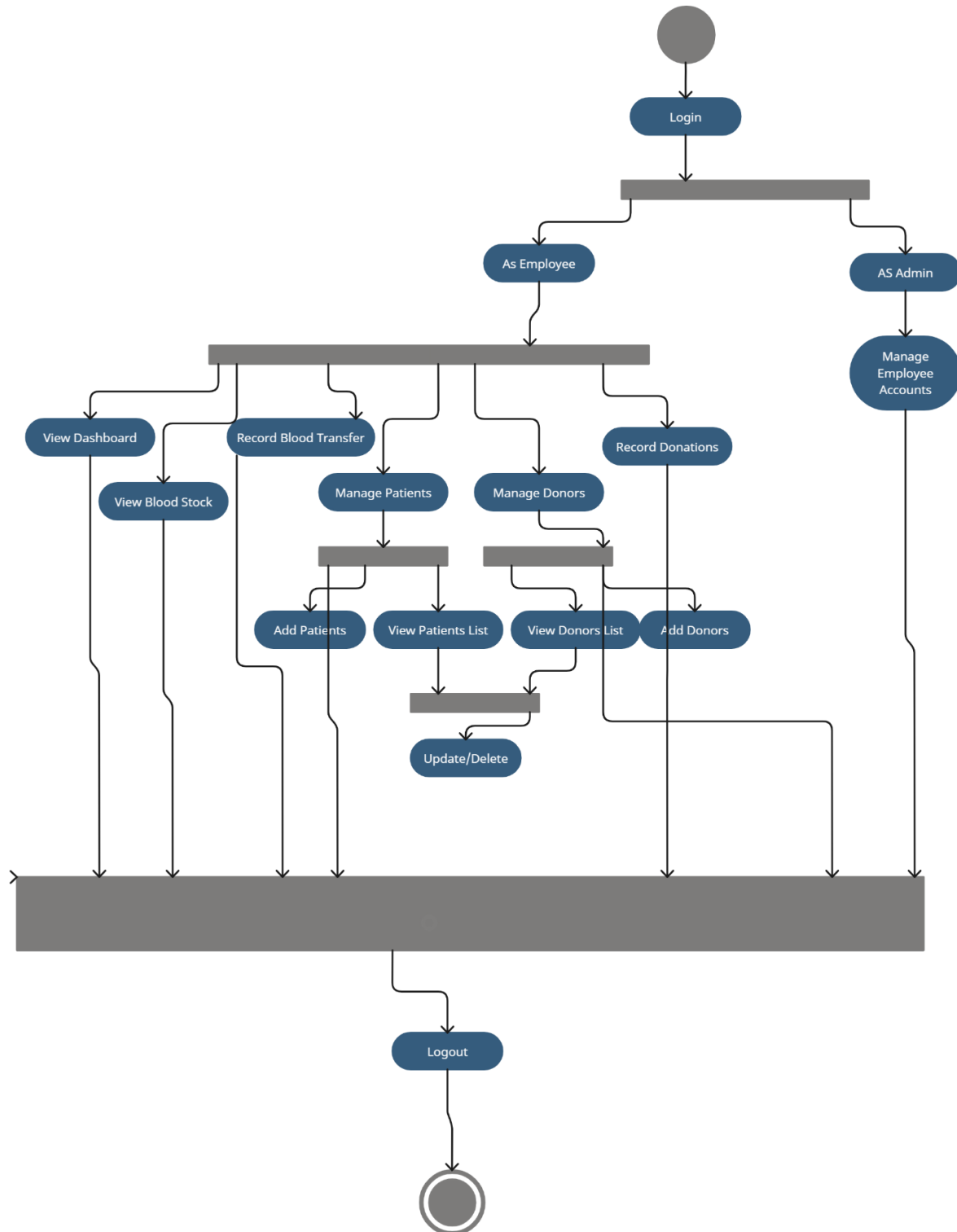
Class Diagram

Class diagram is a static diagram. It represents the static view of an application Class diagram shows a collection of classes, interfaces, associations , collaborations , and constraints Purpose of Class Diagrams Analysis and design of the static view of an application . Describe responsibilities of a system Base for component and deployment diagrams.

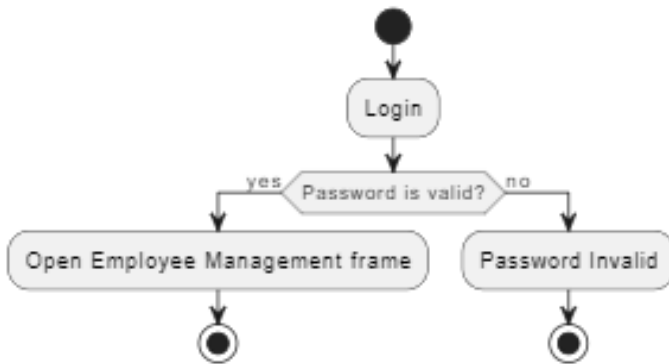


Activity Diagram

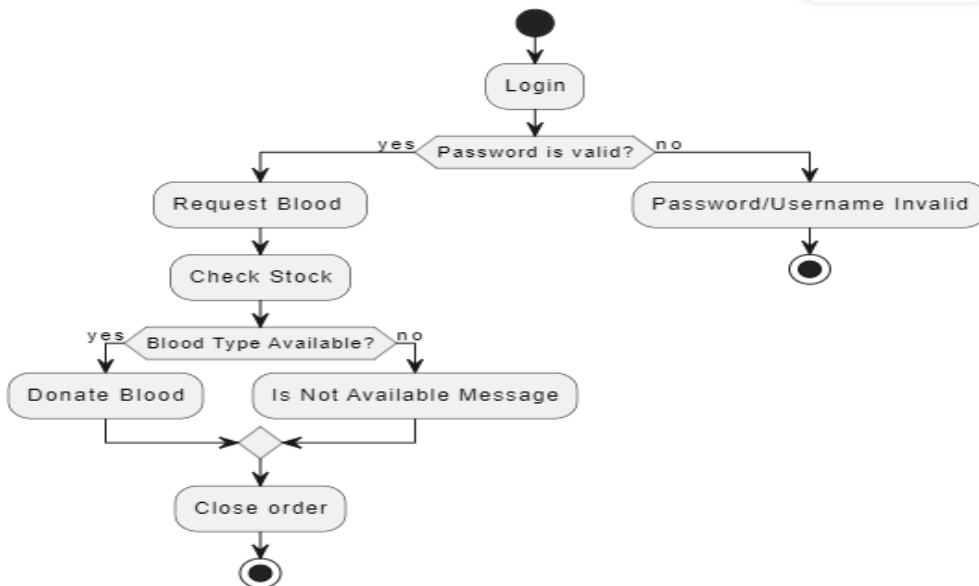
An activity diagram shows a process as a set of activities, showing their sequences , where activities can be carried out in parallel and show which person is responsible for which activity.



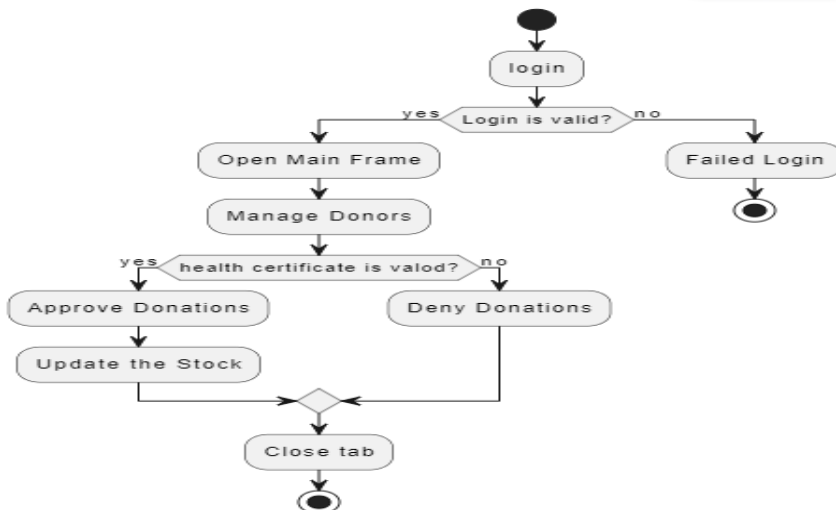
Admin Activity Diagram



Pateint Activity Diagram

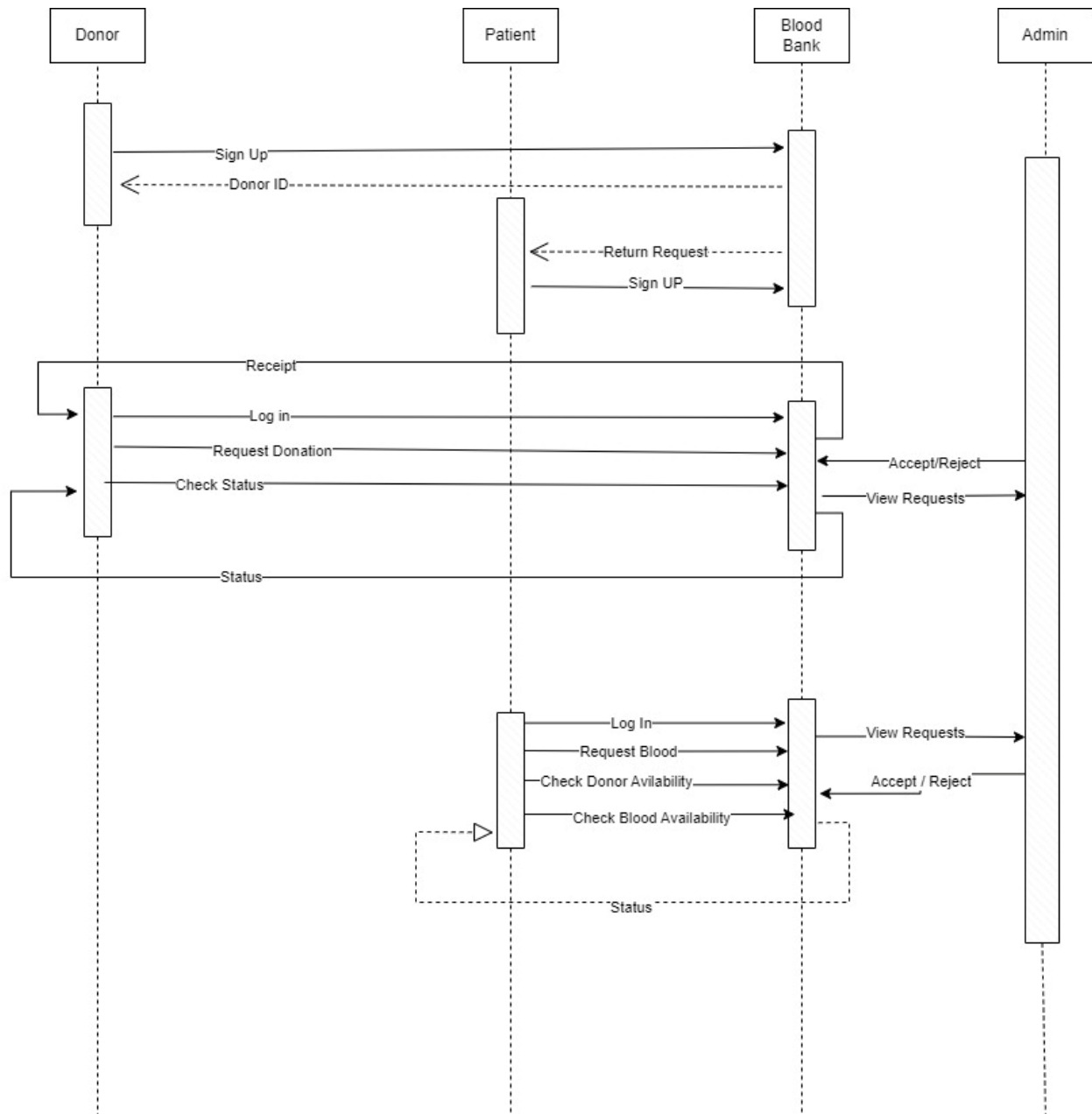


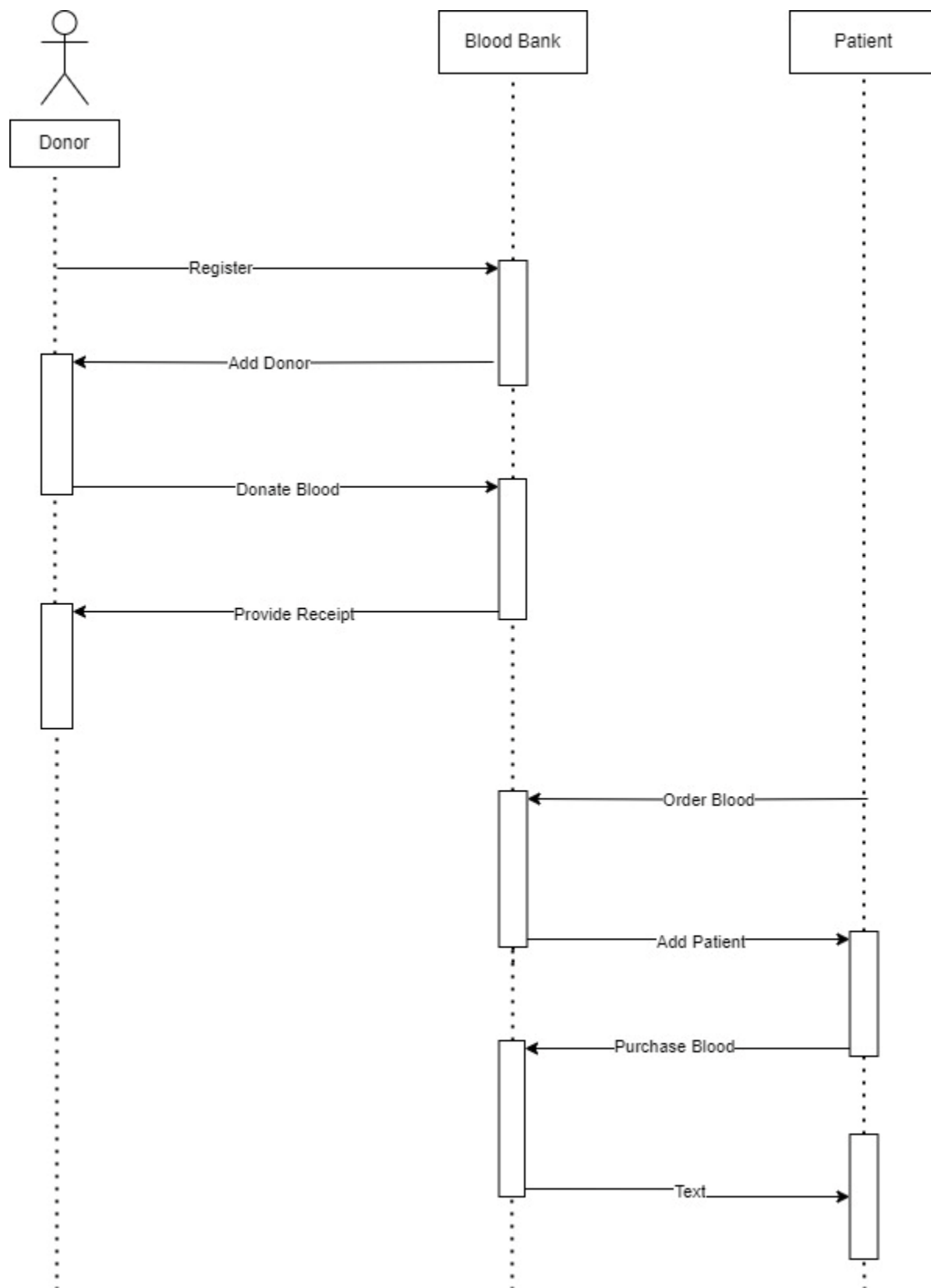
Employee Activity Diagram

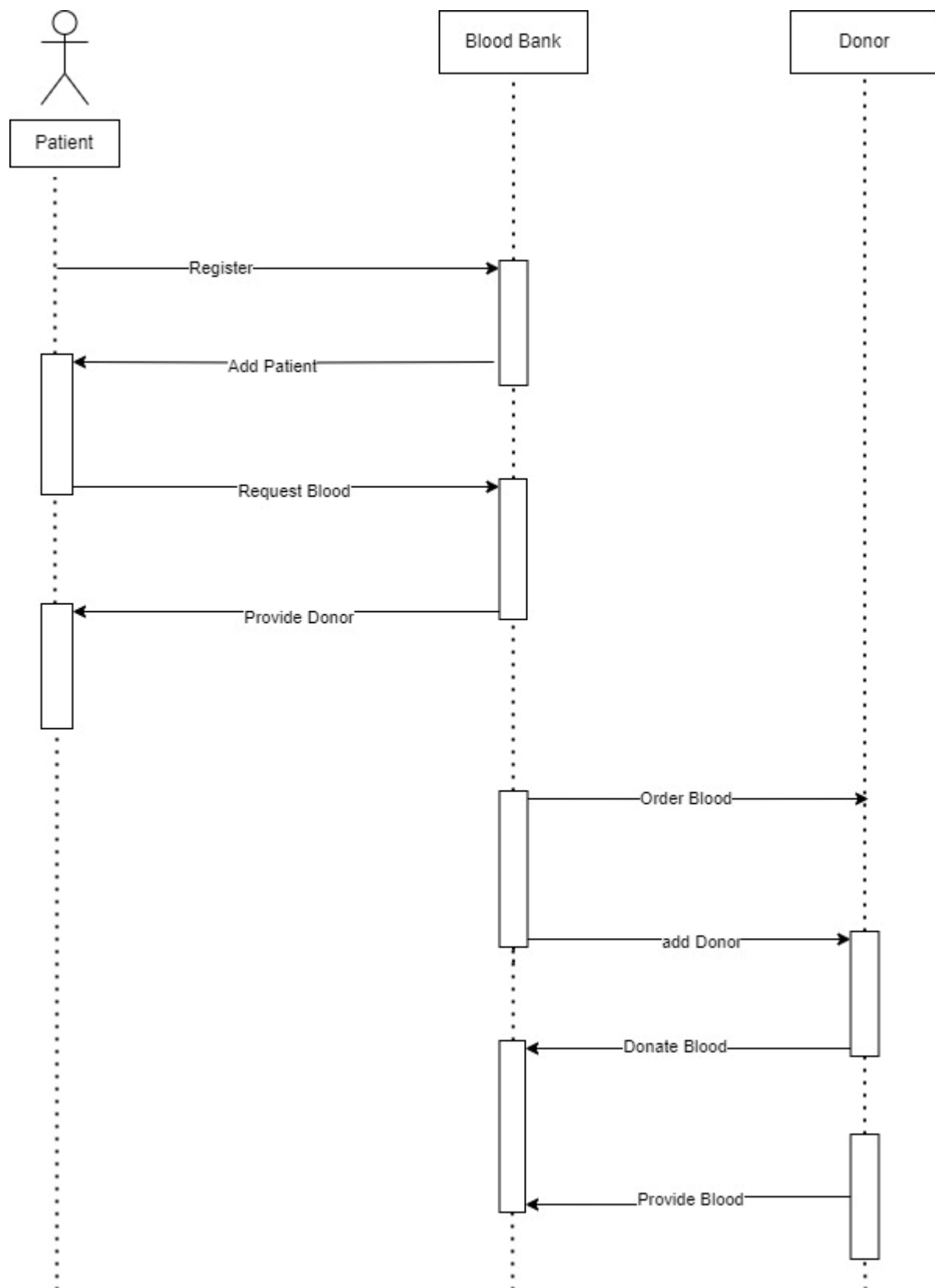


Sequence Diagram:

In the busy blood bank, patients register for transfusions, and staff checks records for compatible blood. Donors provide details, get screened, and those eligible donate. After donation, donors rest, and blood undergoes testing and processing. Labeled units are stored for future use. Patients receive transfusions in their rooms, thanks to donors and staff, ensuring a steady supply of safe blood, saving lives and fostering community well-being.

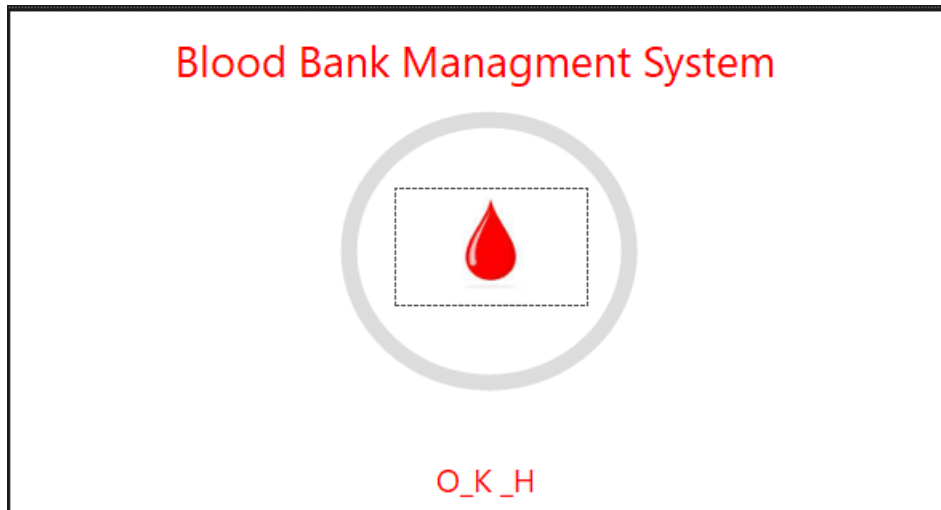




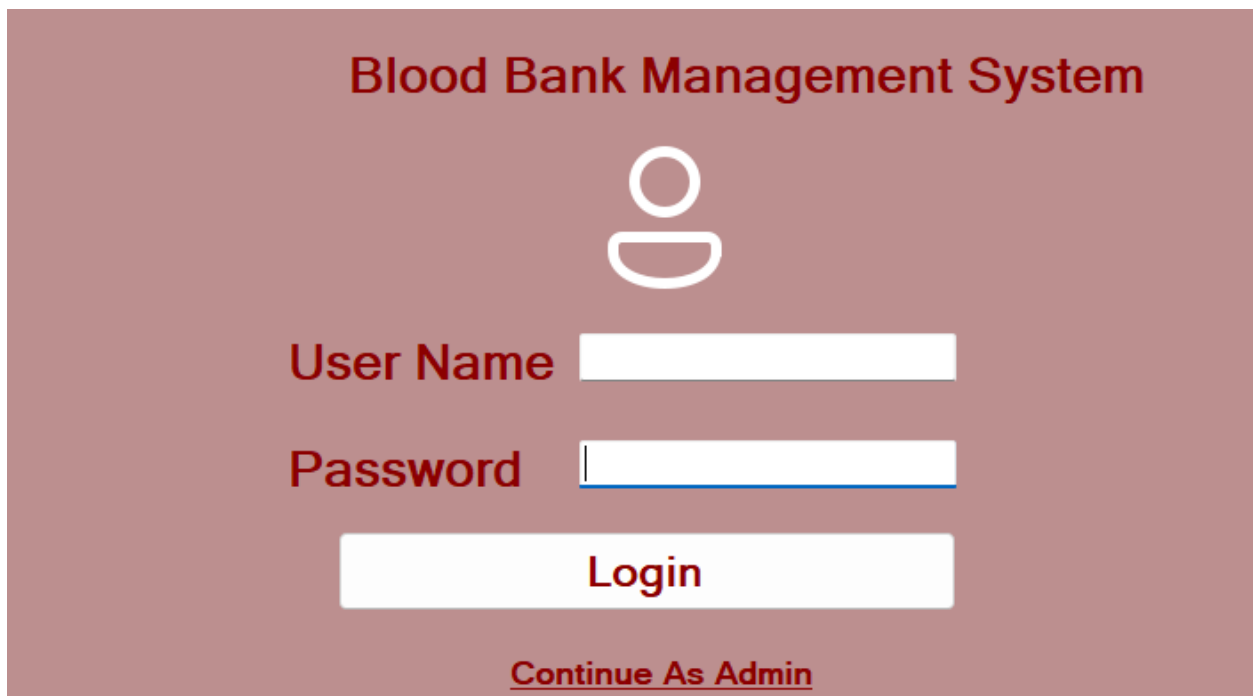


Chapter 3: UI & Prototype

Loading Page



Login Page



Home Page



Donor

Name	<input type="text"/>	Age	<input type="text"/>
Phone	<input type="text"/>	Gender	<input type="text"/>
Address	<input type="text"/>	Blood Type	<input type="text"/>

Save

View Donors

Donors List	
Name	<input type="text"/>
<div></div>	

Patient

Patient	
Name	Blood Type
<input type="text"/>	<input type="text"/>
Gender	Phone
<input type="text"/>	<input type="text"/>
Age	
<input type="text"/>	
Address	
<input type="text"/>	

View Patient

Patient List

Name

Blood Stock

Blood Stock



Blood Transfer

Blood Transfer



Patient ID

Patient Name

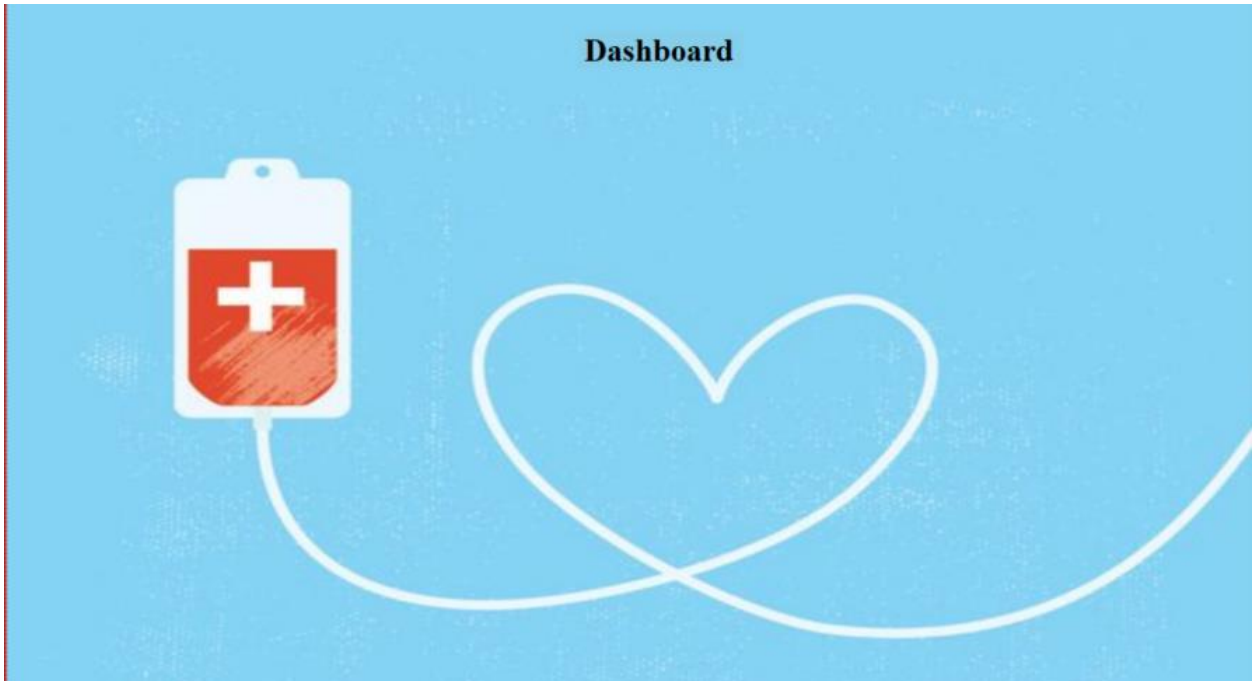
Blood Group

Available Or Not

Save

Dashboard

Dashboard



Chapter 4: System Testing

Intended:

- To show that a program does what it is intended to do.
- To discover program defects before it is into use.

Goals of the testing process:

- Validation testing: demonstrate that the software meets its requirements.
- Defect testing: discover incorrect undesirable or specification non conform behavior of the software.

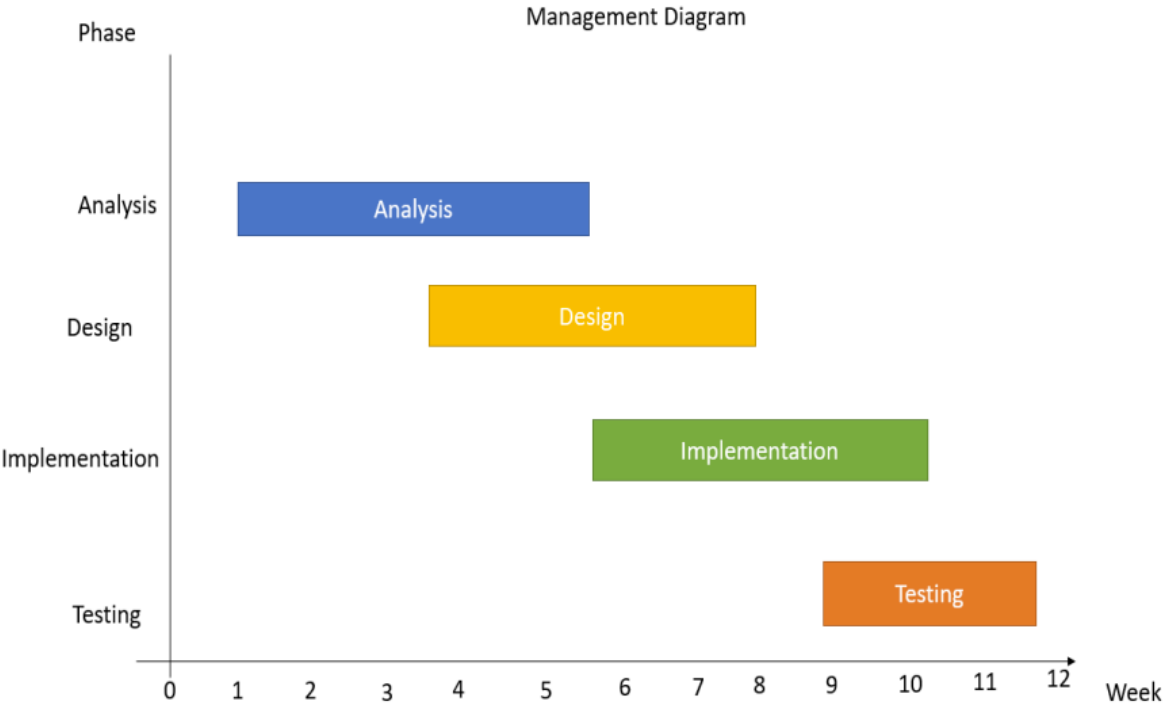
Test ID	Test Condition	Test Steps	Test Input	Expected result	Actual result	Status	Remark
TC-01	Check that if the correct username and password able to login	1- Enter username and password 2- Click login	Username : M Password : 1233	Open Main Frame	Open Main Frame	Pass	None
TC-02	Check that if the incorrect username and password can't login	1- Enter username and password 2- Click login	Username : hhh Password : 9876	Login Failed	Login Failed	Pass	None

TC-03	Check if the loading page is loading efficiency for employee	Click on program button	none	Login page	Login page	Pass	None
TC-04	Check if the loading page is loading efficiency for admin	Click on program button	none	Login page	Login page	Pass	None

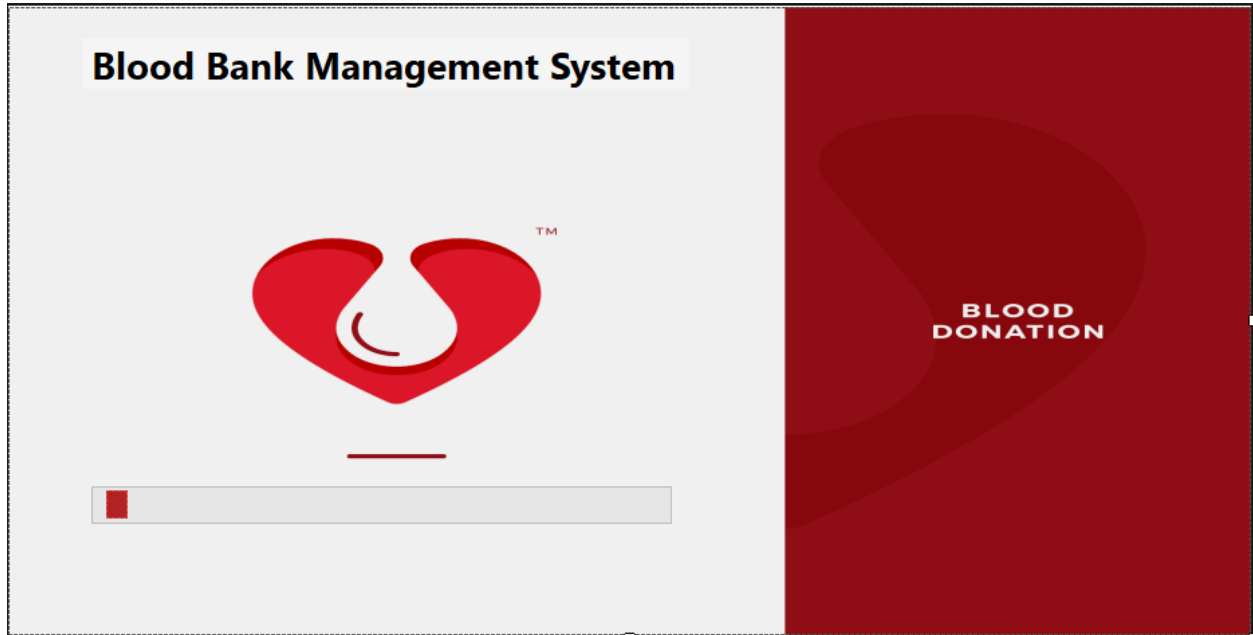
TC-05	Check if donor name field accepts the Input of 12 character	Give input	Khalil	Khalil	Khalil	Pass	None
TC-06	Check if patient name field accepts the Input of 12 character	Give input	Omar	Omar	Omar	Pass	None

TC-07	Check if employee can't access as an admin	1- Input username 2- Input password 3- login	Username : m Password : 1233	Main form	Main form	Pass	None
TC-08	Check if there is enough stock to transfer	1- input the Patient ID 2- Shows IsAvailable 3- Transfer Blood	Patient ID : 1	IsAvaialble	IsAvaialble	Pass	None

Management Diagram



Chapter 5: System Implementation



Blood Bank Managemt System

X



User Name

Password

Login

[Continue as Admin](#)

X

Blood Bank Managment System

Password

Login


Cancel

Employees

Logout

Blood Bank Managment System

Employees



Name

Password

Save

Edit

Delete

Donor
Donate
View Donors

Patient
View Patients
Blood Stock
Blood Transfer
DashBoard
Logout

Blood Bank Managment System




Donor
Donate
View Donors

Patient
View Patients
Blood Stock
Blood Transfer
DashBoard
Logout

Blood Bank Managment System

Donor



Name

Age

Gender

Phone

Blood Group

Address

Save

Blood Bank Managment System

Donor

Donate

View Donors

Patient

View Patients

Blood Stock

Blood Transfer

DashBoard

Logout

Blood Stock

Donors List

Name

Blood Group

Donate

Blood Bank Managment System

Donors List

Name

29 | Page

Donor

Donate

View Donors

Patient

View Patients

Blood Stock


Blood Transfer

DashBoard

Logout

Blood Bank Managment System

Patients



Name

Age

Phone

Gender

Blood Group

Adress

Save

Donor

Donate

View Donors

Patient

View Patients

Blood Stock

Blood Transfer

DashBoard

Logout

Blood Bank Managment System

Patients List

Name

Age

Phone

Gender

Blood Group

Address

Edit

Delete

Donor

Donate

View Donors

Patient

View Patients

Blood Stock


Blood Transfer

DashBoard

Logout

Blood Bank Managment System

Blood Stock



Filter

Donor

Donate

View Donors

Patient

View Patients

Blood Stock

Blood Transfer

DashBoard

Logout

Blood Bank Managment System

Blood Transfer

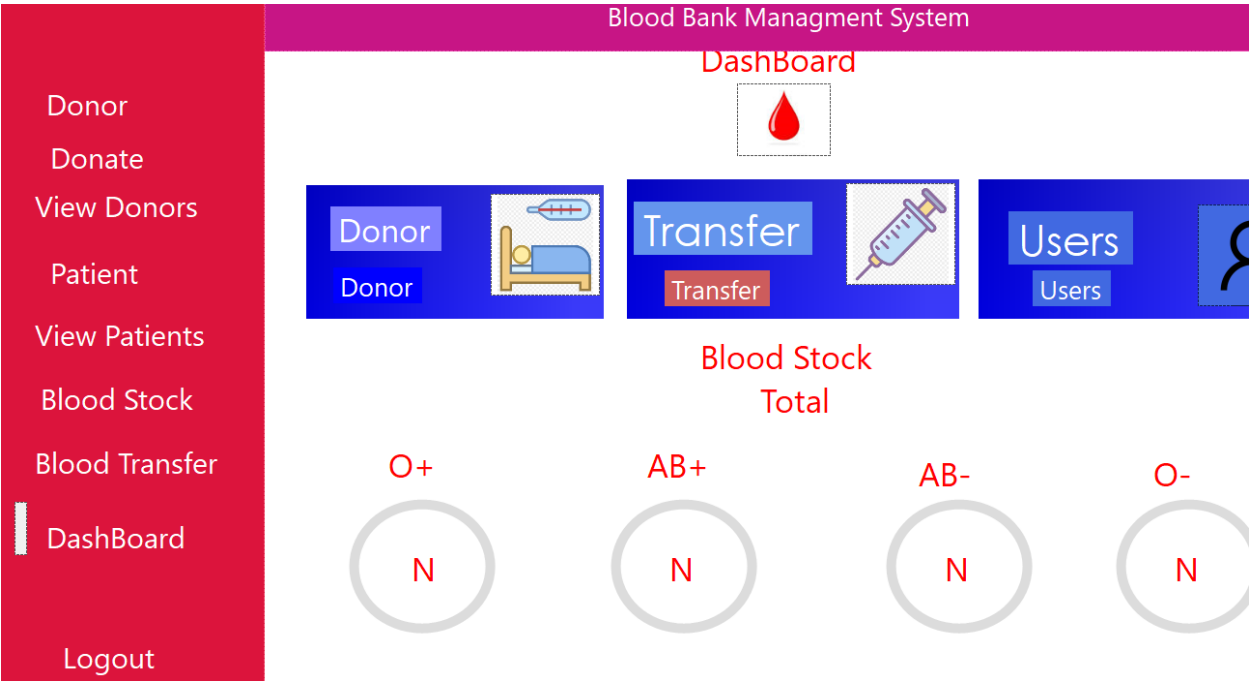
Patient Id

Patient Name

Blood Group

AvailableOrNot

Transfert



Chapter 6: Tools

Coding & Designing:

C# .Net Framework (Visual Studio Community 2022)

Database Manipulation:

SQL Server

Documenting:

MS Word