



# **NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY**

## **DEPARTMENT OF COMPUTER SCIENCE AND TELECOMMUNICATION ENGINEERING**

### **A PROJECT REPORT ON Blood Donation Management System**

**Course Title** : Software Engineering and Information System Design Lab  
**Course Code** : CSTE 3210  
**Session** : 2020-21

**Year : 03, Term : 02.**

#### **Submitted by:**

**NAME** : Sahadat Hossain  
**ID** : MUH2101034M

**NAME** : MD.Nishad Ahmed Jibon  
**ID** : MUH2101021M

#### **Submitted to:**

**Dr. Nazia Majadi**  
**Associate Professor of**  
**Computer Science &**  
**Telecommunication Engineering.**

**Submission Date: 10/03/2025**

# 1. Introduction

The Blood Donation Management System is a comprehensive web application designed to streamline the process of blood donation and management. The system facilitates the registration of users, helps people find registered blood donors, allows donors to update their profiles, enables users to request blood in emergencies, and provides a platform for users to submit complaints. The goal of the project is to create a user-friendly, accessible, and efficient system for both blood donors and people in need of blood.

## 1.a) Project Scope

The **Blood Donation Management System** is designed to provide a platform for users to register as blood donors or individuals in need of blood, search for donors, request blood in emergencies, update their profiles, and submit complaints. The scope of the project focuses on creating a system that connects people in need of blood with registered donors, promotes blood donation awareness, and ensures ease of use for all users.

### Inclusions:

1. **User Registration & Profile Management:** Users can register as blood donors or seekers, update personal details, and manage their profiles.
2. **Blood Donor Search:** Users can search for blood donors based on blood type, location, and availability.
3. **Blood Request for Emergencies:** The system allows users to request blood for emergencies, which will be displayed on the homepage.
4. **Complaint Submission System:** Users can submit complaints about the system, which will be handled by administrators.
5. **Database Management:** A MySQL database will store user and donor information, blood requests, and complaints.
6. **Frontend Development:** The interface will be developed using HTML, CSS, and JavaScript, ensuring that it is responsive and user-friendly.

### Exclusions:

1. **Real-Time Blood Inventory Tracking:** The system does not include real-time tracking of blood inventory levels.
2. **Advanced Donation Scheduling:** There will be no scheduling or reminders for regular blood donations.
3. **Payment or Financial Transactions:** No payment-related features will be included in the system.

## 1.b) Project Objectives & Goals

The **Blood Donation Management System** aims to achieve several objectives that focus on usability, security, and efficient blood donation management.

1. **Create an Accessible Platform for Blood Donors and Seekers:** To provide an easy-to-use platform for users to register as donors or seekers and connect with each other.
2. **Enhance Blood Donation Awareness:** To raise awareness about blood donation by connecting individuals in need of blood with potential donors, especially in emergency situations.
3. **Offer an Efficient Emergency Blood Request System:** To facilitate quick and efficient blood requests during emergencies by displaying requests prominently on the homepage for all users.
4. **Ensure Data Security and Privacy:** To protect user data, including personal and medical information, through encryption and secure storage mechanisms.
5. **Improve User Experience with Simple Interface:** To create a user-friendly interface that makes the system easy to navigate for all users, regardless of their technical expertise.
6. **Provide Scalability for Growth:** To build a system that can scale as the platform grows, accommodating more users and data without performance degradation.
7. **Enable Easy Management of Complaints:** To allow users to submit complaints about the system, which will be addressed in a timely manner by administrators.

## 2. Defining Requirements

Defining requirements is an important part of the project as it ensures that the system's functions and features meet the user's needs and expectations.

### 2.a) Defining All Requirements

The system's overall requirements consist of both functional and non-functional aspects:

1. **User Registration and Profile Management:**
  - Users can register as blood donors or people seeking blood.
  - Donors can update their profile, including contact information and blood group.
2. **Blood Donor Search:**
  - The system allows users to search for blood donors based on their location, blood type, and availability.
3. **Blood Request:**
  - Users can request blood for emergency situations.
  - Blood requests are displayed on the homepage for all registered users to view.
4. **Complaint System:**
  - Users can submit complaints about the system or its services.
5. **Database Management:**

- A MySQL database will store user data, donor information, blood requests, and complaints.
  - The data will be accessible through Python-based APIs.
6. **Frontend Design:**
- The front-end will be developed using HTML, CSS, and JavaScript to ensure responsive design and smooth user interaction.
- 

## 2.b) Functional Requirements

Functional requirements define the specific behaviors and functions of the system, focusing on what the system should do:

1. **Registration:**
    - Users can register with their basic details (name, contact information, blood group, etc.).
    - The system must verify user details during registration and ensure that there is no duplicate data.
  2. **Login and Authentication:**
    - Registered users can log in using their credentials.
    - The system will authenticate user credentials securely.
  3. **Search Donors:**
    - Users can search for blood donors based on their location, blood group, and availability.
    - Results will display matching donors' names, contact details, and blood groups.
  4. **Profile Updates:**
    - Donors can update their profile, including contact information, availability, and blood group.
  5. **Blood Request Submission:**
    - Registered users can submit a request for blood in case of emergencies.
    - Blood requests will be visible to all registered users on the homepage.
  6. **Complaint Submission:**
    - Users can submit complaints related to the system.
    - Complaints will be stored and reviewed by administrators.
- 

## 2.c) Non-Functional Requirements

Non-functional requirements focus on the performance, usability, security, and other quality attributes of the system:

1. **Usability:**

- The system must have an easy-to-use interface, ensuring smooth navigation and accessibility for all users.
  - The design should be responsive to different screen sizes, from desktops to mobile devices.
2. **Performance:**
    - The system should be able to handle a large number of users and data without performance degradation.
    - The search functionality must return results in a timely manner.
  3. **Security:**
    - User data, including personal and medical information, must be encrypted and securely stored.
    - User authentication and authorization must be implemented to protect sensitive data.
  4. **Scalability:**
    - The system should be able to scale and accommodate more users and data as the platform grows.
  5. **Reliability:**
    - The system must be stable and operate without crashes or downtime.
    - It should recover quickly from any errors and continue to function properly.
  6. **Availability:**
    - The system should be available 24/7 with minimal downtime for maintenance.

## 3.Requirement analysis

### 3.a) Scenario based models

#### Use case description

##### 1. User Registration for Blood Donation Services

- **Actor:** Registered User (donor or requester)
- **Precondition:** The user must have access to the system (website/application).
- **Description:** This use case allows new users to create an account on the platform. The user must fill in their personal details, medical history, and location. Once registered, the user is added to the system's database and can access features such as finding donors, updating their profile, and requesting blood donations.

---

##### 2. Requesting Blood in an Emergency

- **Actor:** Registered User (blood requester)
  - **Precondition:** The user must be logged in and have an active account on the system.
  - **Description:** In case of an emergency, a registered user can request blood from available donors. The system will allow the user to specify the required blood type, urgency, and location. This request will then be displayed on the homepage for potential donors to view and respond to. The system ensures that the requester can receive the necessary blood from the closest available donor.
- 

### 3. Donor Profile Update

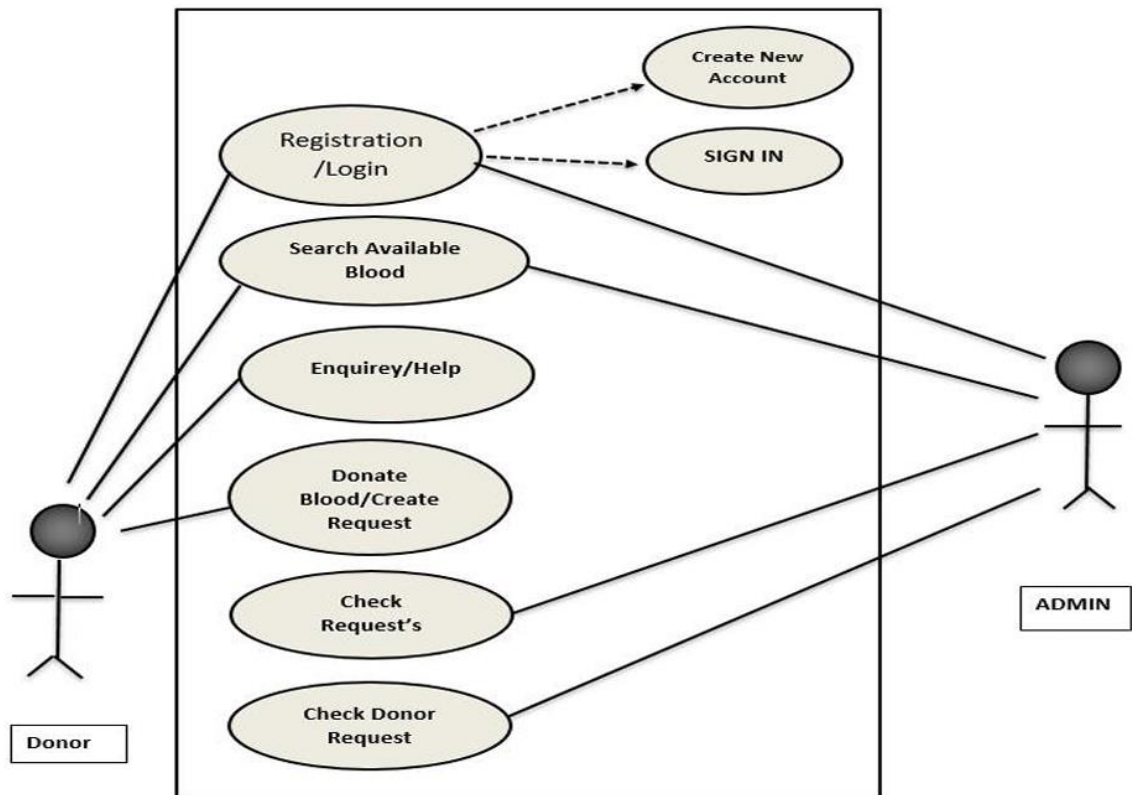
- **Actor:** Registered Donor
- **Precondition:** The user must be logged in to their account and have previously registered as a donor.
- **Description:** This use case allows a registered donor to update their personal information, including contact details, medical history, and blood donation history. The system stores these updates in the database to ensure the donor's profile is accurate for future donations and interactions with blood requesters. The donor can also choose to deactivate their profile or change their availability for future donations.

## b. Case Diagram

**Case Diagram** is a visual representation of the interactions between users (actors) and the system, showing how the system's functionalities (use cases) serve the needs of these users. It helps to identify the key functionalities of the system, the roles that interact with these functions, and how the system is used in different scenarios. The purpose of a Use Case Diagram is to provide a clear, high-level overview of what the system will do and how users will interact with it.

### USE CASE DIAGRAM

#### Use Case Diagram:-



## Activity Diagram

Activity Diagram An activity diagram for an online blood bank management system represents the flow of activities and actions within the system. It focuses on the behavior and sequence of actions that occur during the operation of the system. Here is a brief description of the main components typically.

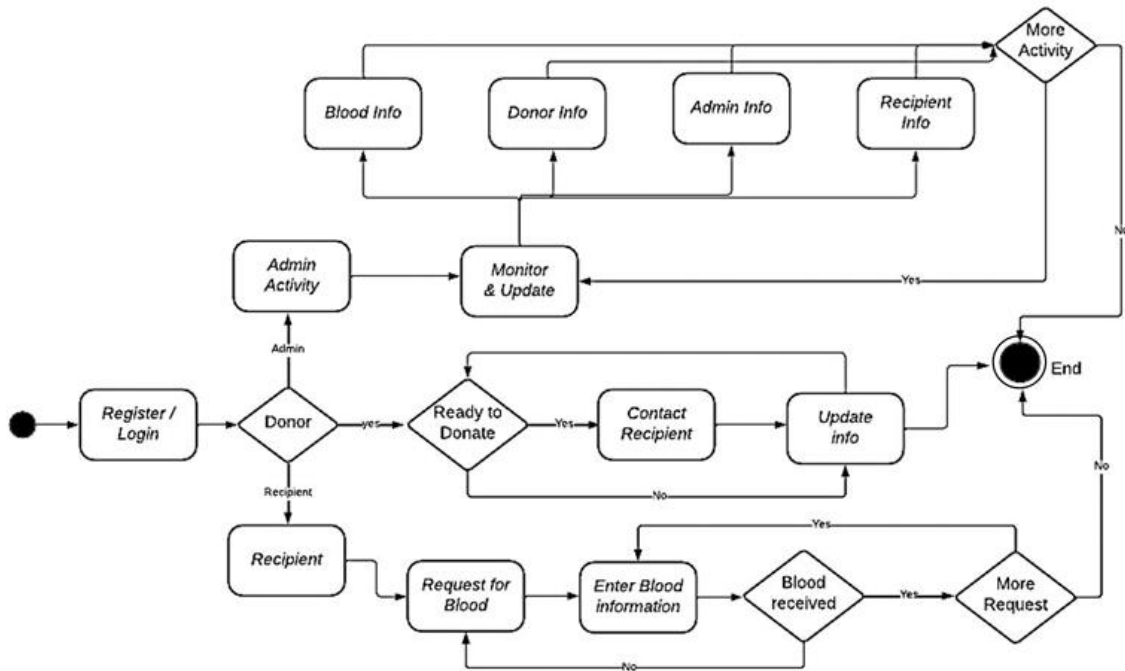


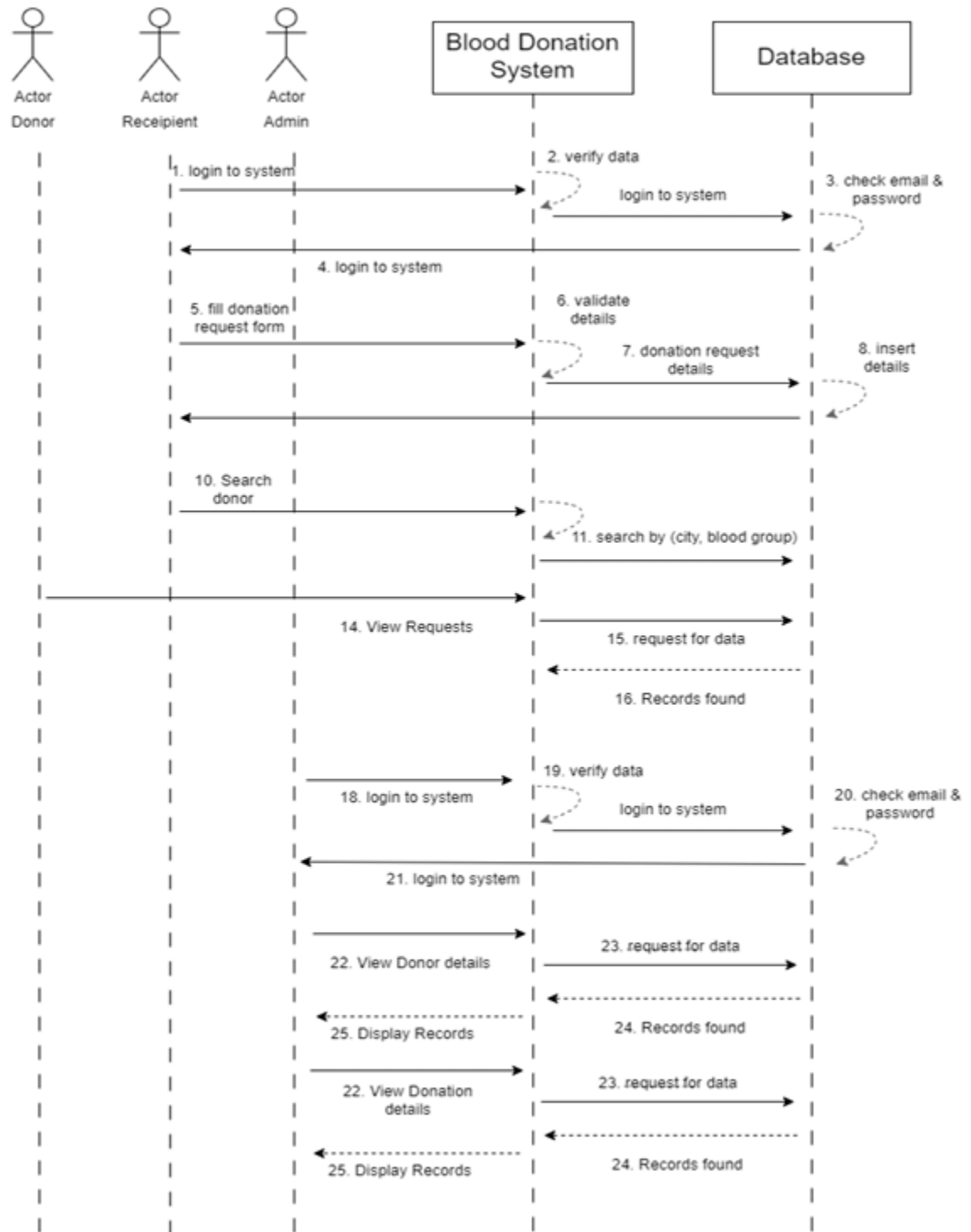
Fig: Activity Diagram

## Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that visually represents the interactions and order of messages exchanged between objects or components in a system. It illustrates



the flow of communication and the sequence of actions that occur during a particular scenario or use case. Here's a brief explanation of a sequence diagram



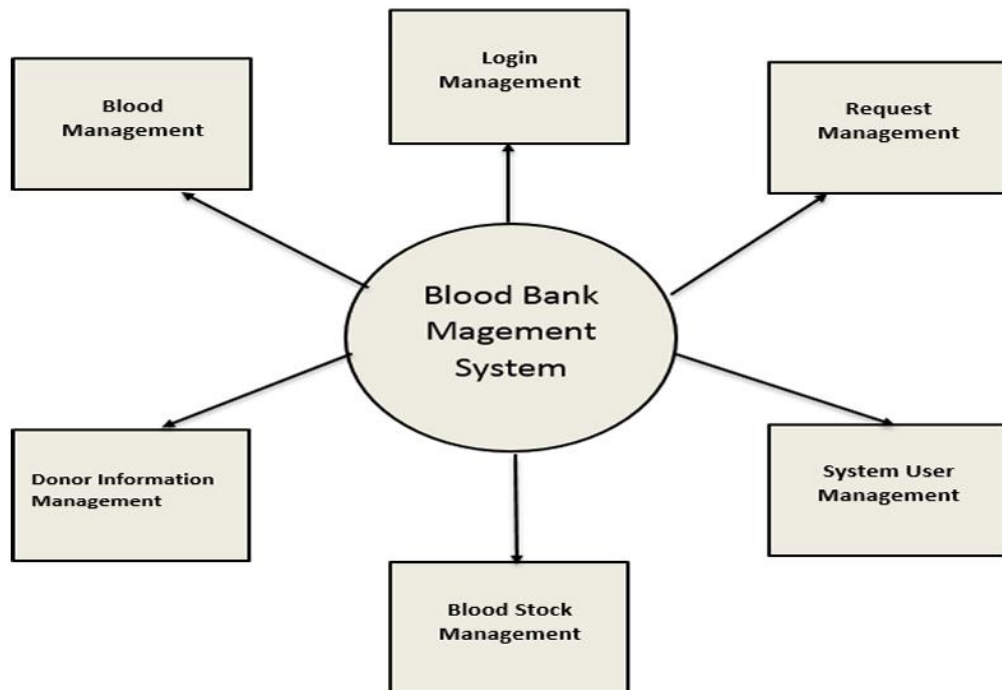
## Flow Model

**Flow Model** is a visual representation that shows the sequence of steps or actions that are taken in a process. It helps to understand the logic and flow of activities within a system, typically represented by a series of interconnected shapes (e.g., ovals, rectangles, diamonds). Each step or action is represented by a specific shape, and arrows are used to show the direction of the flow.

### Data Flow Diagram (Level 0)

DFD (Data Flow Diagram):-

Level 0:-



## Data Flow Diagram (Level 1)

Level 1:-

