# Blood donation

SOFTWARE ENGINEERING

Ahmed aboelnaga 225258
Mohamed Ashraf 225257
Manar basouni 225180
ENG. YOUSRA EMAM | DR. SARA AHMED

# Blood Bank Management System Documentation

## Table of Contents:



## 1. Introduction

The Blood Bank Management System is designed to streamline the process of managing blood donations and requests, ensuring that patients in need receive timely and appropriate support. This system provides functionalities for users to register as donors or recipients, log in to their accounts, update their personal information, request or donate blood, and perform various other operations related to blood management.

## 2. System Overview

The Blood Bank Management System consists of two main types of users: donors and recipients. Each user type has specific functionalities tailored to their needs. Donors can register, log in, request donations, update their information, and delete their accounts. Recipients can register, log in, update their information, search for specific blood types,

request blood, and delete their accounts. The system uses file storage to save user data and employs appropriate data structures to manage information efficiently and ensure data integrity.

# 3. Functional Requirements

## Donor Functionalities:

1. Register: Donors can register by providing their personal information, including name, email, password, age, gender, blood type, and medical details such as disease information and latest donation date.

2. Login: Donors can log in to the system using their unique ID and password.

3. Request Donation: Donors can record a request for blood donation, specifying the details of the donation.

4. Update Data: Donors can update their personal and medical information as needed.

5. Delete Account: Donors can delete their account from the system, removing all associated data.

## Recipient Functionalities:

1. Register: Recipients can register by providing their personal information, including name, email, password, age, gender, blood type, hospital name, and doctor's name.

2. Login: Recipients can log in to the system using their unique ID and password.

3. Update Data: Recipients can update their personal and medical information as needed.

4. Delete Account: Recipients can delete their account from the system, removing all associated data.

5. Search Blood Type: Recipients can search for the availability of specific blood types in the system's database.

6. Display All Blood Data: Recipients can view all available blood data, including the types and quantities of blood available.

7. Request Blood: Recipients can request a specific blood type and quantity, providing details such as the hospital and doctor involved in the request.

# 4. Class Diagram

The class diagram represents the structure of the Blood Bank Management System, illustrating the classes, their attributes, methods, and relationships. The primary classes in the system are Donor and Recipient, each with specific attributes and methods.

Classes:

- Donor

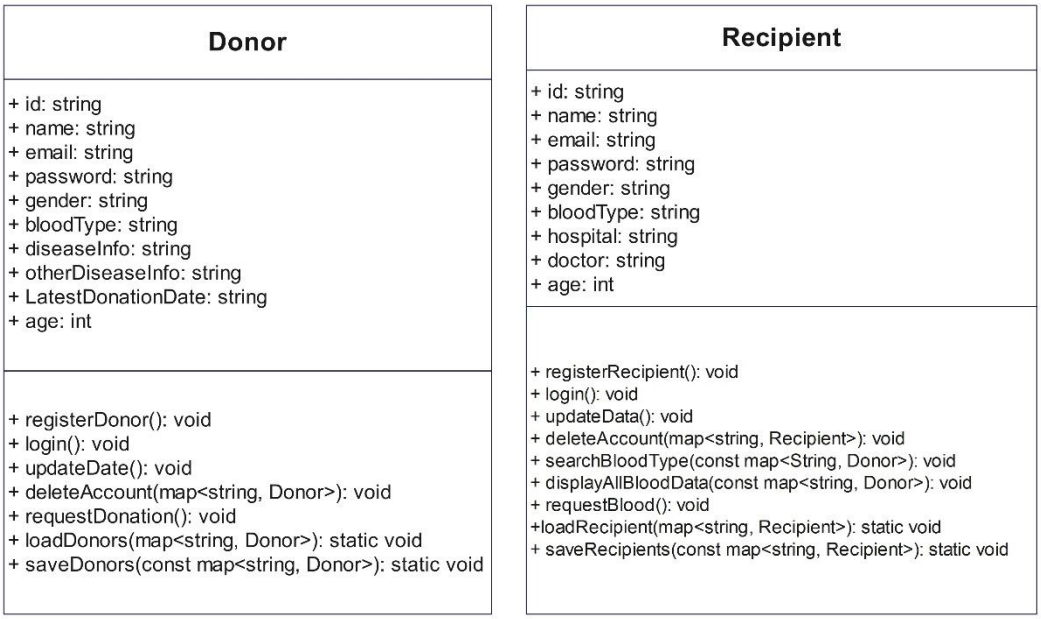
  - Attributes: id, name, email, password, age, gender, bloodType, diseaseInfo, otherDiseaseInfo, latestDonationDate

  - Methods: registerDonor(), login(), updateData(), deleteAccount(), requestDonation(), loadDonors(), saveDonors()

- Recipient

  - Attributes: id, name, email, password, age, gender, bloodType, hospital, doctor

  - Methods: registerRecipient(), login(), updateData(), deleteAccount(), searchBloodType(), displayAllBloodData(), requestBlood(), loadRecipients(), saveRecipients()



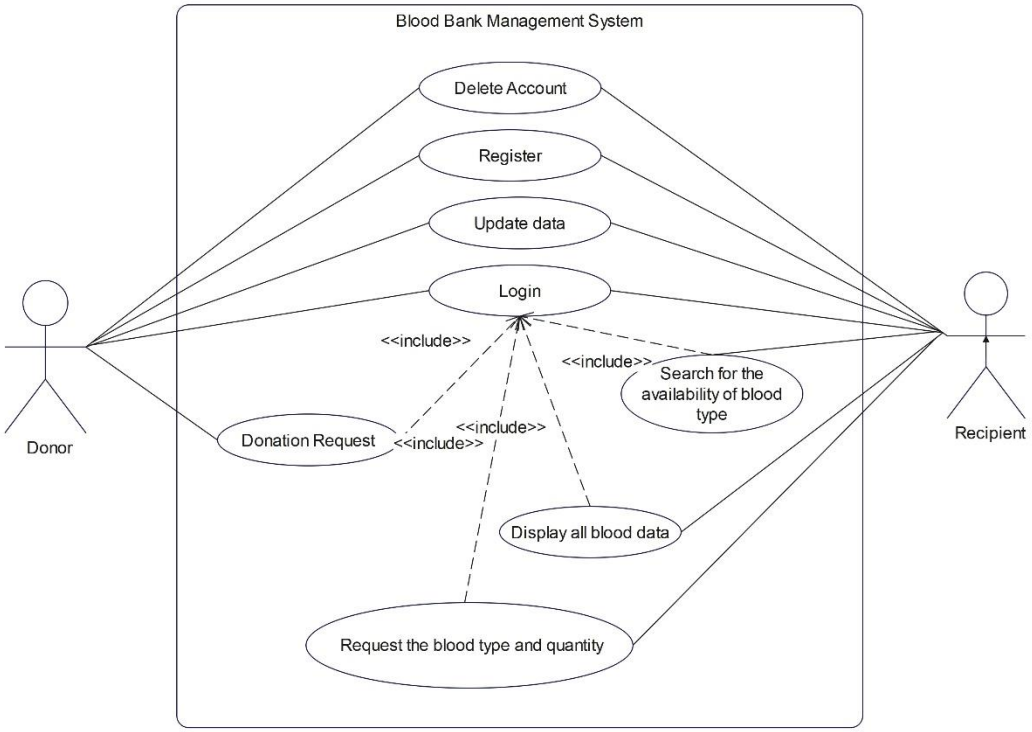
# 5. Use Case Diagram

The use case diagram illustrates the interactions between users (donors and recipients) and the system, highlighting the functional requirements of the system. It depicts the various actions that each type of user can perform and how they interact with the system to accomplish these tasks.

Use Cases:

- Donor
  - Register
  - Login
  - Request Donation
  - Update Data
  - Delete Account

- Recipient
  - Register
  - Login
  - Update Data
  - Delete Account
  - Search Blood Type
  - Display All Blood Data
  - Request Blood




# 6. Sequence Diagram

The sequence diagram details the interaction between objects in the system to perform specific tasks. It shows the flow of messages between objects and the order in which these messages are exchanged to achieve a particular functionality.

Donor Registration Sequence:

1. Donor selects the registration option.
2. System prompts for personal information.
3. Donor enters personal information.
4. System validates and saves the information.
5. Confirmation message is displayed to the donor.

Recipient Blood Request Sequence:

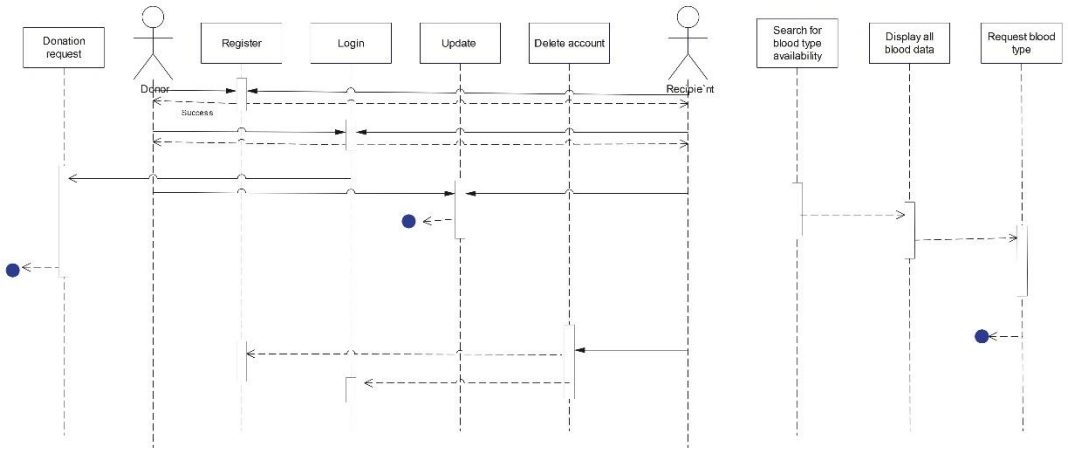
1. Recipient logs into the system.
2. Recipient selects the option to request blood.

3. System prompts for blood type and quantity.

4. Recipient enters the required details.

5. System validates the request and updates the database.

6. Confirmation message is displayed to the recipient.




# 7. State Diagram

The state diagram depicts the various states an object (donor or recipient) can be in and the transitions between these states based on events. It provides a clear view of how objects change states in response to different actions performed by the users.

Donor States:

- Registered

- Logged In

- Updating Data

- Deleting Account

- Requesting Donation

Recipient States:

- Registered

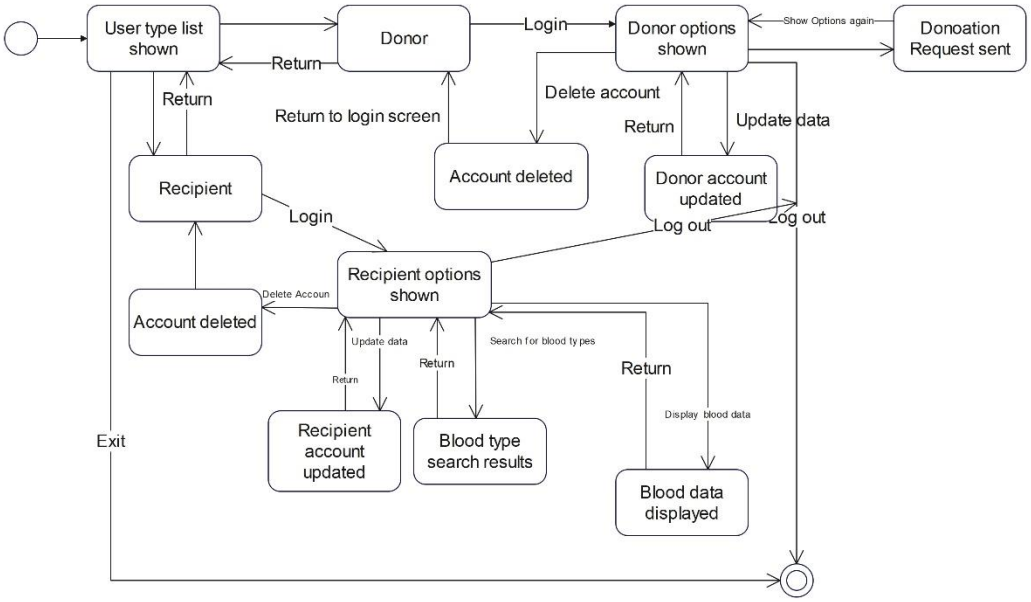
- Logged In

- Updating Data

- Deleting Account

- Searching Blood Type

- Requesting Blood




# 8. Activity Diagram

The activity diagram illustrates the workflow of activities and actions within the Blood Bank Management System, showing the control flow from one activity to another. It provides a visual representation of the processes involved in various user interactions.

Donor Workflow:

1. Donor selects registration option.

2. Donor enters personal information.

3. System validates and saves information.

4. Donor receives confirmation.

Recipient Workflow:

1. Recipient logs in.

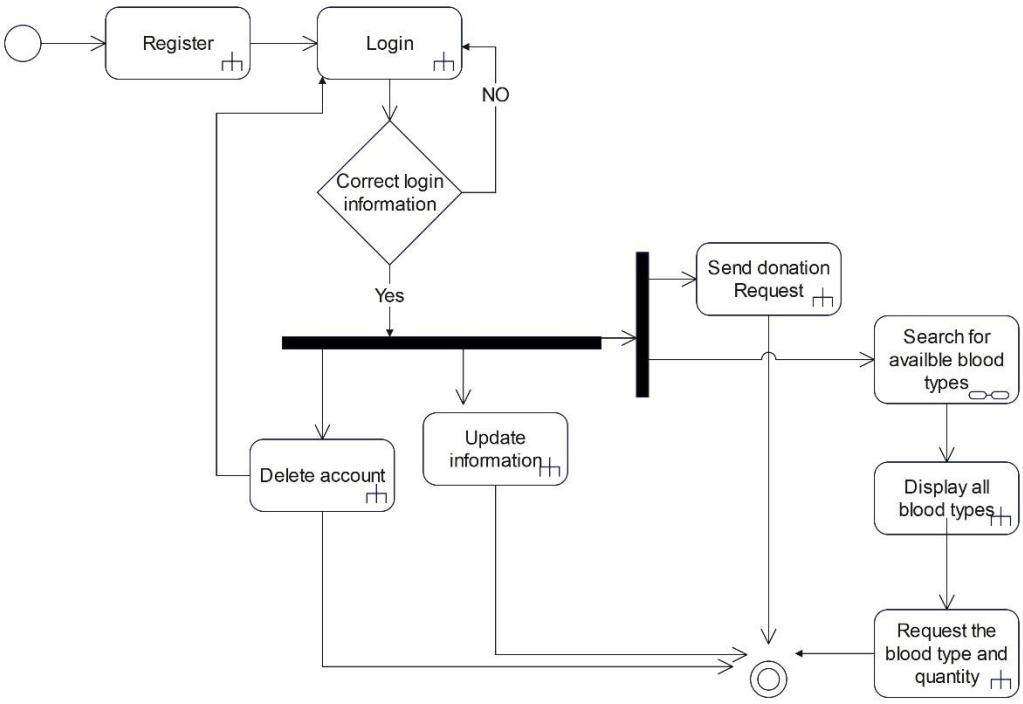
2. Recipient searches for blood type.

3. System displays search results.

4. Recipient requests blood.

5. System validates request and updates database.

6. Recipient receives confirmation.




# 9. Implementation Details

Donor Class:

- registerDonor(): Collects donor information and stores it in the donors map.

- login(): Authenticates donor using ID and password.

- updateData(): Allows donors to update their information.

- deleteAccount(): Removes donor from the system.

- requestDonation(): Records a donation request.

- loadDonors(): Loads donor data from a file.

- saveDonors(): Saves donor data to a file.

Recipient Class:

- registerRecipient(): Collects recipient information and stores it in the recipients map.
- login(): Authenticates recipient using ID and password.
- updateData(): Allows recipients to update their information.
- deleteAccount(): Removes recipient from the system.
- searchBloodType(): Searches for available blood types in the donor data.
- displayAllBloodData(): Displays all blood data from donors.
- requestBlood(): Records a blood request.
- loadRecipients(): Loads recipient data from a file.
- saveRecipients(): Saves recipient data to a file.

# 10. File Structure

The file structure of the Blood Bank Management System includes the following files:

- main.cpp: Contains the main function and menus.
- Donor.h: Header file for the Donor class.
- Recipient.h: Header file for the Recipient class.
- Donor.cpp: Implementation file for the Donor class.
- Recipient.cpp: Implementation file for the Recipient class.
- donors.txt: File for storing donor data.
- recipients.txt: File for storing recipient data.

# 11. Conclusion

The Blood Bank Management System is designed to manage blood donations and requests efficiently, ensuring that patients receive timely support. The system's design, encompassing various UML diagrams and detailed implementation, ensures clarity and maintainability. By providing functionalities for both donors and recipients, the system facilitates the management of blood resources and supports the overall goal of saving lives through effective blood donation and distribution processes.