KIBUR COLLEGE

DEPARTMENT OF COMPUTER SCIENCE

COURSE-NAME: SOFTWARE ENGINEERING ASSIGNMENT

COURSE-CODE: COSC3061

PROJECT-TITLE: BLOOD DONATION MANAGEMENT SYSTEM

GROUP MEMBERES **ID**

1. NATNAEL ADEFRS …… URC/127
2. NATNAEL TESFAYE ……..URC/142
3. SAMUEL SETARIGE …… URC 136/14
4. ELIAS BERHANU …… URC/102

DATE OF SUBMITION: 21-07-2016 **SUBMITED TO MR. YECHALE D.**

Contents

ACKNOWLEDGMENT 4
ABBREVIATION i
CHAPTER ONE 1
INTRODUCTION OF THE PROJECT PROCESS 1
1.1 Background information 1
1.2 Statement of problem 2
1.3 Objectives 3
1.3.1 General Objectives 3
1.3.2 Specific objectives 3
1.4 Scope 3
1.4.1 In Scope 3
1.4.2 Out of scope 4
1.5 Description Of The Existing System 4
Chapter Two 6
PROPOSED SYSTEM 6
2.1 Overview 6
2.2 Functional Requirements 7
2.3 Non–Functional Requirements 8
2.4 Scenarios for using our system 9
Chapter Three 11
Unified Modeling Language 11
3.1 The use case diagram 11
3.1.1 Use case Description 12
3.2 Object Model 16
3.2.1 Data Dictionary 16
3.3 Class Diagram 17
3.4 Dynamic Model 18
3.4.1. Sequence Diagram 18
3.4.2 Activity Diagram 19
Chapter Four 22
System Design 22
4.1 Overview 22

4.1.1 Purpose of the System Design 22
4.1.2 Design Goal 22
4.2 Proposed system architecture 22
4.3 Database Design 23
4.4 Access control 23
4.5 User interface design 25
Chapter Five 27
Implementation 27
5.1 Overview 27
5.2 Development Tools 27
5.3 Prototypes 27

Figure 1 Manual donor registration paper 5
Figure 2 Use case diagram 11
Figure 3 Class diagram 17
Figure 4 Sequence diagram for scenario 1 18
Figure 5 Sequence diagram for scenario 2 18
Figure 6 Sequence diagram for scenario 3 19
Figure 7 Sequence diagram for scenario 4 19
Figure 8 New user registration 20
Figure 9 Selection activity 20
Figure 10 Editing profile 21
Figure 11 Logout profile 21
Figure 12 Data base design 23
Figure 13 Our home page 25
Figure 14 Our donation form 25
Figure 15 Our login(Sign up) page 26
Figure 16 Sample source code 27

Table 1 Use case description donor registration 12
Table 2 Use case description Authentication to the system 12
Table 3 Use case description blood registration 13
Table 4 Use case description Logout 13
Table 5 Use case description Hospital registration 14
Table 6 Use case description Update store 14
Table 7 Use case description edit profile 15
Table 8 Use case description View in store 15
Table 9 Data dictionary 16
Table 10 Development Tools 27

ACKNOWLEDGMENT

We would like to express our sincere gratitude and appreciation to Mr. Yechale Degu, our esteemed teacher, for his invaluable guidance, support, and mentorship throughout the duration of our software engineering project on the Blood Bank Donation System.

Furthermore, we extend our heartfelt thanks to Kibur College for providing us with a conducive learning environment and resources that have enabled us to undertake this project with enthusiasm and dedication. The college's support and encouragement have been crucial in fostering our academic growth and professional development.

We are grateful for the opportunity to work on such a meaningful project that aims to contribute to the vital cause of blood donation and saving lives. This project has been a rewarding experience, and we are thankful for the support and guidance we have received from Mr. Yechale and Kibur College.

Thank you once again to Mr. Yechale Degu and Kibur College for their unwavering support and mentorship throughout this software engineering project.

ABBREVIATION

- “SQL” = STRUCTURED QUERY LANGUAGE
- “DBMS” = DATABASE MANAGEMENT SYSTEM
- “ETC” = ET CETERA
- “IE” =ID EST
- “ID” = IDENTIFICATION
- “NO.” = NUMBER
- “BBDS” = BLOOD BANK DONATION SYSTEM
- “UC” = USE CASE
- “UML” = UNIFIED MODELING LANGUAGE
- “BBDS” = BLOOD BANK DONATION SYSTEM
- “WEB” = WEBSITE
- “FIG” = FIGURE
- “HTML” = HYPERTEXT MARKUP LANGUAGE
- “CSS” = CASCADING STYLE SHEETS

CHAPTER ONE

INTRODUCTION OF THE PROJECT PROCESS

Efficient and accurate registration systems are vital for the success of blood donation programs, enabling the collection and management of critical donor information. This abstract presents a comprehensive database solution specifically designed for blood donation registration systems, aiming to enhance the overall efficiency, accessibility, and security of the process.

The proposed database encompasses a wide range of features, including donor registration, eligibility test, medical history recording, and consent management. By centralizing and organizing donor information, the database minimizes duplication, eliminates manual paperwork, and ensures data integrity. This streamlines the registration process, enabling healthcare professionals and blood banks to access and update donor information in real-time.

Furthermore, the database incorporates advanced security measures to protect sensitive donor data. Robust encryption techniques, access controls, and audit trails safeguard the database against unauthorized access and ensure compliance with privacy regulations. Donors can have peace of mind knowing that their personal and medical information is secure and confidential.

The database's integrated approach also facilitates collaboration and communication among various stakeholders involved in the blood donation process. Donors can easily schedule appointments, receive reminders, and access their own donation history. Healthcare professionals and blood banks can efficiently search for suitable donors based on blood type, location, and other criteria. This enables timely coordination between donors and recipients, ensuring an uninterrupted supply of blood for medical treatments.

In conclusion, the database for blood donation registration systems presented in this abstract offers an integrated and secure solution to streamline the donor registration process. By centralizing donor information, enhancing data security, and enabling seamless communication, the database improves the efficiency and accessibility of blood donation programs. Ultimately, this empowers healthcare professionals and blood banks to provide prompt and accurate assistance to those in need, saving lives and contributing to the well-being of communities.

1.1 Background information

The management of blood donation processes and maintaining an efficient blood bank are critical aspects of ensuring a reliable and constant supply of blood for medical treatments.

Donor Management: The database system allows for the efficient registration and management of donor information. It includes the donor's personal details, medical history, donation records, and eligibility criteria. This enables the blood bank to maintain accurate information about donors, their blood types, and any specific requirements or restrictions.

Blood Inventory: The system facilitates the management of blood inventory by tracking the storage, availability, and expiration dates of blood units. It allows staff to monitor the quantity of different blood types, ensure proper storage conditions, and enable automatic notifications for replenishment or discarding of expired units.

Distribution and Requests: The database system enables efficient management of blood requests from hospitals, clinics, and other medical facilities. Staff can track requested blood types, quantities, and urgency. The system can match requested blood types with available units in the inventory, simplifying the process of allocating and distributing blood to meet various medical demands.

Reporting and Analysis: The database system provides reporting and analysis tools to generate comprehensive reports on donor statistics, inventory levels, usage patterns, and other key performance indicators. These reports assist in planning blood drives, identifying trends, and making informed decisions to optimize blood bank operations.

Security and Privacy: Given the sensitive nature of medical data, the database system ensures the security of donor information, complying with data protection regulations. Access controls, encryption, and other security measures are implemented to safeguard personal and medical data.

1.2 Statement of problem

The blood bank donation management database system in Ethiopia faces critical challenges that hinder its ability to efficiently manage and ensure a stable blood supply.

Limited Technical Infrastructure: Ethiopia encounters limitations in technical infrastructure necessary for effective implementation and maintenance of a robust blood bank donation management database system. Insufficient funding and resources, outdated technology infrastructure, inadequate network coverage, and unreliable power supply contribute to system instability and limited access to essential functionality.

Inadequate Data Management: Many blood bank donation management systems in Ethiopia rely on manual or paper-based record-keeping methods, leading to inaccuracies, difficulties in data retrieval, and insufficient information for proper donor management, blood inventory control, and decision-making. The absence of centralized databases and weak data security measures further exacerbate inefficiencies and inhibit effective blood bank operations.

Donor Engagement and Recruitment: Ethiopia faces challenges in actively engaging and expanding the pool of voluntary blood donors. Factors such as low awareness about blood donation benefits, cultural beliefs, and misconceptions hinder successful donor recruitment initiatives. Without comprehensive strategies for donor engagement, including awareness campaigns and targeted recruitment efforts, the blood bank donation management database system struggles to build and maintain a sufficient and diverse donor base.

Logistical Barriers: Ethiopia's vast and diverse geography, coupled with limited transportation infrastructure, poses logistical obstacles for the timely collection and distribution of blood. Insufficient refrigerated storage facilities, poorly optimized supply chain management, and inadequate transportation networks contribute to delays, possible spoilage, and improper handling of blood products, impacting availability and quality.

Limited Resources and Training: The availability of skilled staff, equipment, and financial resources in Ethiopia is often constrained. Insufficient funding allocation for blood bank systems hampers system improvement and maintenance. Furthermore, the lack of training opportunities and capacity building programs for blood bank personnel adversely affects their ability to operate, utilize, and maximize the potential of the database management system effectively.

1.3 Objectives

1.3.1 General Objectives

The general objective of this project is to implementing a blood bank donation management system is to improve the **efficiency**, **accuracy**, and **accessibility** of blood bank operations and processes.

1.3.2 Specific objectives

- To create a centralized database system that efficiently manages donor information, including personal details, medical history, donation records, and eligibility criteria.
- To establish a comprehensive blood inventory management system that tracks the storage, availability, and expiration dates of blood units. This includes real-time monitoring, inventory forecasting, and automated notifications for blood replenishment or discarding of expired units.
- To enable seamless distribution and retrieval of blood units based on demand from hospitals, clinics, and other medical facilities. This includes efficient tracking of blood requests, matching available blood types with requested units, and streamlining the logistics for timely and safe delivery to medical facilities.
- To generate comprehensive reports and analytic capabilities that provide critical insights into donor statistics, blood inventory levels, usage patterns, and quality control metrics.
- To enhance data security measures and compliance with data protection regulations to safeguard donor information and maintain confidentiality.

1.4 Scope

1.4.1 In Scope

A blood bank donation registration management system is responsible for efficiently managing the process of registering blood donors and maintaining their information. Here are some key components that fall within the system's scope:

- **Donor Registration**: The system should allow for the registration of blood donors by capturing their personal details, contact information, demographic information, and relevant medical history. It should ensure the confidentiality and security of donor data.
- **Eligibility Screening**: The system should include a module to check the eligibility of potential donors based on criteria such as age, weight, previous medical conditions, and lifestyle factors. This screening process helps ensure the safety and suitability of donors.
- **Consent Management**: The system should enable donors to provide informed consent for blood donation. It should record and store consent information securely, as per legal and ethical requirements.
- **Donor Health Assessment**: The system may include functionalities for assessing donor health, such as conducting medical questionnaires or physical examinations. This helps identify any potential health risks or reasons for deferral.
- **Appointment Scheduling**: The system should facilitate the scheduling of donation appointments, allowing donors to choose suitable dates and times. It should also provide reminders and notifications to donors about their upcoming appointments.
- **Donor Relationship Management**: The system can maintain a database of donor interactions and activities. This may include recording previous donations, sending donor-related updates and reminders, and maintaining communication with donors.

1.4.2 Out of scope

While a blood bank donation registration management database system primarily focuses on managing donor information and registration processes.

There are certain tasks that generally fall outside its scope. These include:

- **Public Awareness Campaigns**: While promoting blood donation is important, the system usually does not handle public awareness campaigns or marketing initiatives. Such activities are typically managed through separate channels, campaigns, and platforms.
- **Eligibility Screening**: The system should include a module to check the eligibility of potential donors based on criteria such as age, weight, previous medical conditions, and lifestyle factors. This screening process helps ensure the safety and suitability of donors.
- **Consent Management**: The system should enable donors to provide informed consent for blood donation. It should record and store consent information securely, as per legal and ethical requirements.
- **Donor Health Assessment**: The system may include functionalities for assessing donor health, such as conducting medical questionnaires or physical examinations. This helps identify any potential health risks or reasons for deferral.
- **Appointment Scheduling**: The system should facilitate the scheduling of donation appointments, allowing donors to choose suitable dates and times. It should also provide reminders and notifications to donors about their upcoming appointments.
- **Donor Relationship Management**: The system can maintain a database of donor interactions and activities. This may include recording previous donations, sending donor-related updates and reminders, and maintaining communication with donors.

1.5 Description Of The Existing System

The manual blood bank management system, although traditional, has its own set of drawbacks that can be quite a hassle to deal with. Here are some common drawbacks of manual blood bank management systems:

- Limited Accessibility:- Paper-based Records: Maintaining paper records makes it difficult to access information quickly and efficiently, especially during emergencies.
- Data Integrity:- Data Entry Errors: Manual data entry can lead to inaccuracies, duplication, or loss of crucial information, impacting the quality of patient care.
- Inventory Management:- Manual Inventory Tracking: Tracking blood inventory manually can result in errors, misplacement of units, and difficulty in identifying expired units.
- Efficiency and Productivity:- Time-Consuming Processes: Manual processes for donor registration, blood donation, and inventory management are time-consuming and may hinder overall efficiency.
- Search and Retrieval:- Search Complexity: Finding specific donor information or blood types manually can be challenging and time-consuming, impeding swift decision-making.
- Reporting and Analysis:- Lack of Data Analysis: Limited ability to generate comprehensive reports and analyze trends for effective decision-making and resource planning.

- Communication and Coordination:- Lack of Communication Channels: Inefficient communication between blood bank staff, healthcare providers, and recipients can lead to delays in blood transfusion processes.
- Security and Confidentiality:- Data Security Risks: Paper-based systems are vulnerable to loss, theft, or unauthorized access, compromising the confidentiality of donor and patient information.
- Scalability:- Inflexibility in Scaling: Manual systems may struggle to accommodate the increasing volume of donors, recipients, and blood inventory data as the blood bank grows.
- Regulatory Compliance:- Compliance Challenges: Meeting regulatory requirements and maintaining accurate records for audits can be cumbersome and error-prone in manual systems.

Figure 1 Manual donor registration paper

Chapter Two

PROPOSED SYSTEM

2.1 Overview

In the documentation section of our project on blood donation registration, we have outlined the process for individuals to sign up as blood donors. This includes detailed instructions on how to register online or in person, the information required for registration, and the steps involved in the screening process. Additionally, we have included forms and templates that donors can fill out to provide consent for donation and medical information. Our goal is to make the registration process as seamless and efficient as possible to encourage more people to become blood donors and help save lives.

Implementing a digital blood donor registration system can effectively address numerous challenges faced by traditional manual processes in blood donation management. Here are ten problems that can be solved by utilizing a digital blood donor registration system:

- Inefficient manual registration process: A digital blood donor registration system can streamline the registration process, making it quicker and more convenient for donors to sign up.
- Limited access to donor information: With a digital system, healthcare providers and blood banks can easily access donor information, including blood type and medical history, which can help in matching donors with patients in need of specific blood types.
- Difficulty in tracking donor eligibility: A digital system can track donor eligibility criteria, such as age, weight, and health conditions, ensuring that only eligible donors are registered and preventing potential health risks.
- Lack of communication with donors: A digital system can send automated reminders to donors about upcoming donation appointments, helping to increase donor retention rates.
- Inaccurate donor records: A digital system can maintain accurate and up-to-date donor records, reducing the risk of errors in tracking donations and ensuring that donors receive proper recognition for their contributions.
- Limited reach to potential donors: A digital system can expand the reach of blood donation campaigns by allowing donors to register online from anywhere, increasing the pool of potential donors.
- Difficulty in tracking donation trends: A digital system can provide real-time data on donation trends, including the number of donations, blood types in demand, and donation locations, helping blood banks better plan for future needs.
- Inadequate donor privacy protection: A digital system can implement robust security measures to protect donor information and ensure compliance with data privacy regulations.
- Manual scheduling challenges: A digital system can automate the scheduling of donation appointments based on donor availability and blood bank needs, reducing scheduling conflicts and improving efficiency.
- Limited donor engagement: A digital system can facilitate communication between blood banks and donors, providing updates on donation impact, upcoming events, and opportunities for involvement, fostering a sense of community and encouraging continued participation.

2.2 Functional Requirements

Functional requirements describe what the system should do to meet the needs of its users. Functional requirements outline functions, features, and capabilities that the system must possess to support specific operations. Based our system these are the 15 functional requirements that our donor registration system meets.

1. User registration: Allow donors to create an account by providing basic information such as name, contact details, blood type, and medical history.
2. Donor eligibility check: Verify donor eligibility based on criteria such as age, weight, health conditions, and recent travel history.
3. Appointment scheduling: Enable donors to schedule donation appointments online based on their availability and blood bank capacity.
4. Blood type matching: Match donors with patients in need of specific blood types based on compatibility and urgency of the request.
5. Donation history tracking: Record and track each donor's donation history, including dates of donations, blood volumes donated, and any adverse reactions experienced.
6. Donor communication: Send automated reminders and notifications to donors about upcoming appointments, donation milestones, and relevant news or events.
7. Donation center locator: Provide a feature to help donors locate nearby donation centers or mobile blood drives based on their location.
8. Social media integration: Allow donors to share their donation experiences on social media platforms to raise awareness and encourage others to donate.
9. Donation progress tracking: Display real-time updates on the overall donation progress, including the number of donations received, blood types in demand, and donation goals.
10. Donor feedback collection: Gather feedback from donors about their donation experience, satisfaction with the process, and suggestions for improvement.
11. Privacy settings: Implement security measures to protect donor information, including encryption of data, secure login procedures, and compliance with data privacy regulations.
12. Reporting and analytics: Generate reports on donation trends, donor demographics, and performance metrics to help blood banks make informed decisions and plan future campaigns.
13. Volunteer opportunities: Provide information on volunteer opportunities related to blood donation events, fundraising activities, and community outreach programs.
14. Donor rewards program: Implement a rewards system to incentivize regular donors with perks such as gift cards, badges, or recognition for their contributions.
15. Mobile app compatibility: Ensure that the blood donation registration system is compatible with mobile devices, allowing donors to access the platform on-the-go and make donations more convenient.
16. Blood inventory management: Enable blood bank staff to monitor and manage the inventory of blood products in real-time, including tracking expiration dates, quantities available, and prioritizing requests based on urgency.
17. Emergency response coordination: Facilitate communication between blood banks, hospitals, and emergency services to coordinate rapid responses in case of emergencies or disasters requiring immediate blood donations.

2.3 Non–Functional Requirements

Non-functional requirements define the qualities that affect the system as a whole, such as performance, usability, security, scalability, and reliability.

1. Performance:

- The system should have a response time of less than 2 seconds for donor registration and appointment scheduling.
- The system should be able to handle concurrent user registration requests and maintain responsiveness under peak loads.

2. Security:

- Data encryption should be implemented to secure donor information during transmission and storage.
- Role-based access control should be in place to restrict access to sensitive donor data based on user roles.
- Regular security audits and vulnerability assessments should be conducted to ensure data security compliance.

3. Usability:

- The user interface should be intuitive, user-friendly, and accessible on various devices and screen sizes.
- The system should provide clear instructions and guidance for donors to complete registration and scheduling processes easily.

4. Reliability:

- The system should have a 99% uptime rate to ensure availability for donors and healthcare providers at all times.
- Automated backup and recovery procedures should be in place to prevent data loss in case of system failures.

5. Scalability:

- The system should be able to accommodate a growing number of donors and donation centers without compromising performance.
- Load balancing mechanisms should be implemented to distribute incoming traffic efficiently across server nodes during peak usage.

6. Data Integrity:

- Data validation checks should be implemented to ensure the accuracy and completeness of donor information entered into the system.
- Regular data backups and data hygiene processes should be in place to maintain data integrity over time.

7. Regulatory Compliance:

- The system should comply with data protection regulations, such as GDPR, HIPAA, or other relevant laws governing donor data privacy.
- Audit trails should be maintained to track changes to donor records and ensure compliance with data governance requirements.

8. Compatibility:

- The system should be compatible with a wide range of devices and browsers to ensure a seamless user experience for all users.

9. Maintenance:

- The system should be easy to maintain and update, with regular backups and monitoring to prevent any downtime or data loss.

10. Documentation:

- The system should have comprehensive documentation available for users and administrators, outlining the functionality and features of the system.

2.4 Scenarios for using our system

Scenario 1: New User Registration

a. User visits the blood donation registration website.
b. User clicks on the "Register Now" button.
c. User fills out the registration form with their personal information, such as name, contact details, and blood type.
d. User agrees to the terms and conditions of the blood donation event.
e. User clicks on the "Submit" button to complete the registration process.
f. User receives a confirmation message on the screen indicating successful registration.

Scenario 2: Event Selection

a. User logs in to the blood donation registration system.
b. User navigates to the list of upcoming blood donation events.
c. User selects a preferred event from the list based on location and date.
d. User clicks on the event to view more details, such as time and venue.
e. User confirms their attendance by clicking on the "Register" button next to the event.
f. User receives a confirmation email with event details and instructions.

Scenario 3: Editing Personal Information

a. User logs in to their account on the blood donation registration system.
b. User navigates to the "Profile" section of their account.
c. User clicks on the "Edit" button next to the personal information they wish to update, such as phone number or email address.
d. User makes the necessary changes in the form provided.
e. User clicks on the "Save Changes" button to update their information.
f. User receives a confirmation message indicating that their information has been successfully updated.

Scenario 4: Cancellation of Registration

a. User logs in to their account on the blood donation registration system.
b. User navigates to the list of upcoming blood donation events they have registered for.
c. User selects the event they wish to cancel their registration for.
d. User clicks on the "Cancel Registration" button next to the event.
e. User confirms the cancellation by providing a reason for not attending.
f. User receives a confirmation email confirming the cancellation of their registration for the event.

Chapter Three

Unified Modeling Language

UML is a standardized modeling language used in software engineering to visually represent a system's design. UML provides a way to visualize a system's architecture, components, relationships, and behaviors using various diagrams.

3.1 The use case diagram

A use case diagram in UML (Unified Modeling Language) is a type of behavioral diagram that depicts the interactions between actors (users) and a system to achieve specific goals or functionalities. Use case diagrams are used to illustrate the high-level requirements of a system and the ways in which users interact with the system.

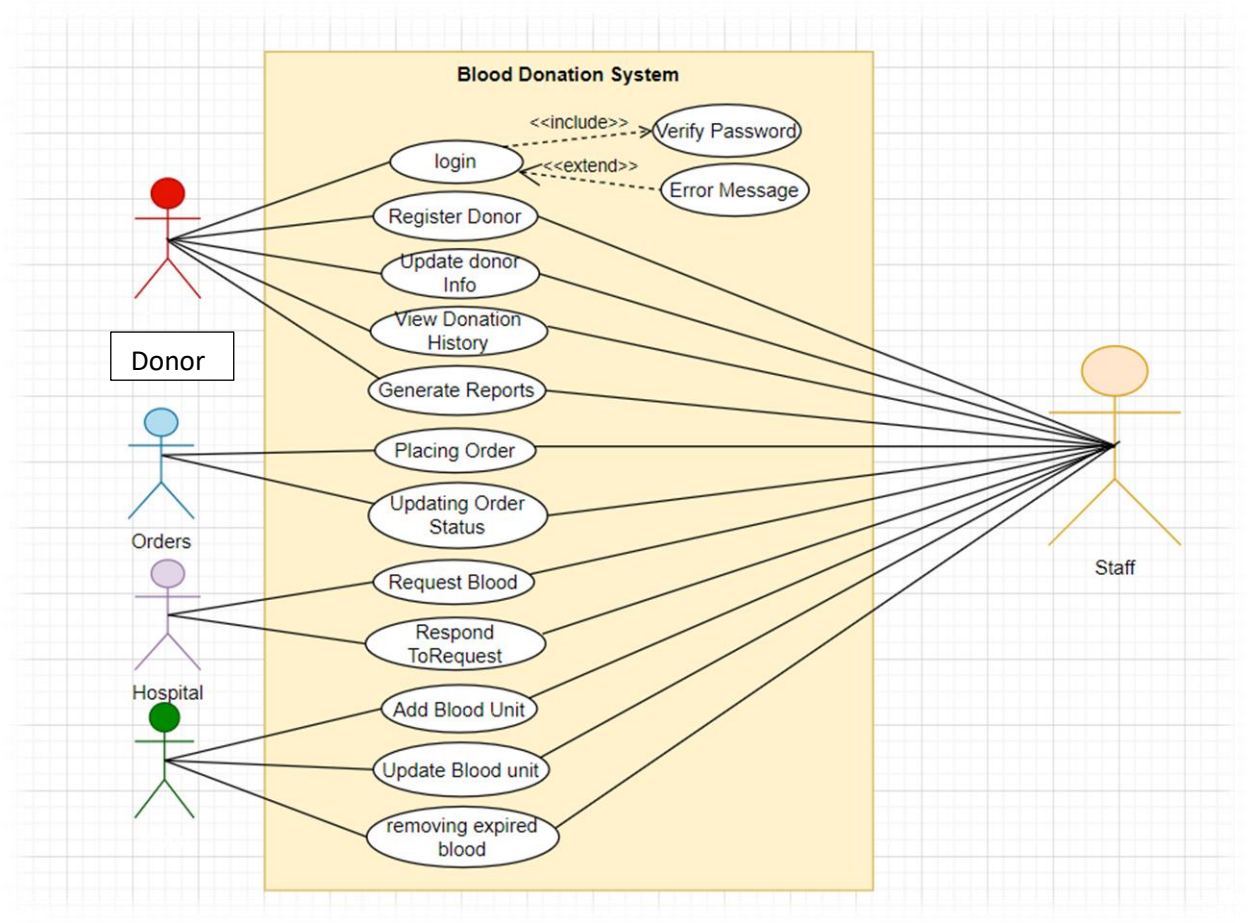

Figure 2 Use case diagram

3.1.1 Use case Description

Table 1 Use case description donor registration

Description : 01	
Use case name	Donor registration
Use case code	UC01
Use case description	Add the donor to the system.
Actor	The Donor
Pre-condition	• F&B manager need to have an internet connection • F&B manager need to have a web browser to access our website
Main flow	• The donor open browser • The donor navigate to our website • The donor click login button • The donor click don't have an account link • The donor fill all required registration form • The donor click register button
Post-condition	The donor will create the account, and then system will save the donor to the database.
Exceptional flow	If the Donor enters incorrect information the system displays an error message

Table 2 Use case description Authentication to the system

Description : 02	
Use case name	Login
Use case code	UC02
Use case description	Authentication to the system
Actor	Staff
Pre-condition	• The staff need to have an internet connection • The staff need to have a web browser to access our website • The staff must have an account
Main flow	• The staff open browser • The staff navigate to our website • The staff click login button • The donor enter password and username. • The donor click login button
Post-condition	System will save the donor to the database.
Exceptional flow	If the staff enters incorrect information the system displays an error message

Table 3 Use case description blood registration

Description : 03	
Use case name	Blood registration
Use case code	UC03
Use case description	Add blood to the store
Actor	Blood and Staff
Pre-condition	• The staff need to have an internet connection • The staff need to have a web browser to access our website • The staff must login to the web.
Main flow	• The staff must login to the system and click store menu. • The staff click add new blood button. • The staff fills the form and then click save button.
Post-condition	System will save the donor to the database.
Exceptional flow	If the staff enters incorrect information the system displays an error message

Table 4 Use case description Logout

Description : 04	
Use case name	Logout
Use case code	UC04
Use case description	Used to leave the system
Actor	Donor
Pre-condition	• The donor need to have an internet connection • The donor need to have a web browser to access our website • The donor must login to the web.
Main flow	• The donor clicks the profile button • The donor clicks on logout link.
Post-condition	The user will leave the page.
Exceptional flow	

Table 5 Use case description Hospital registration

Description : 05	
Use case name	Hospital registration
Use case code	UC05
Use case description	Add hospital to the system
Actor	Hospital
Pre-condition	• The hospital need to have an internet connection • The hospital need to have a web browser to access our website
Main flow	• The hospital open browser • The hospital navigate to our website • The hospital click login button • The hospital click don't have an account link • The hospital fill all required registration and blood ordering form • The hospital click register button
Post-condition	The hospital create an account and wait its order, and then system will save the user to the database.
Exceptional flow	If the hospital enters incorrect information the system displays an error message

Table 6 Use case description Update store

Description : 06	
Use case name	Update store
Use case code	UC06
Use case description	Add new blood unit
Actor	Staff and blood
Pre-condition	• The staff need to have an internet connection • The staff need to have a web browser to access our website • The staff must login into our system
Main flow	• The staff clicks on store menu. • The staff head over to the stored blood to be updated and click the update button. • The staff change details in the forms and click update changes.
Post-condition	The blood detail updated and saved into the database.
Exceptional flow	If the staff enters incorrect information the system displays an error message

Table 7 Use case description edit profile

Description : 07	
Use case name	Edit profile
Use case code	UC07
Use case description	User(Donor and hospital) update their information
Actor	Donor and hospital
Pre-condition	• User need to have an internet connection • User need to have a web browser to access our website • User must login into our system
Main flow	• User clicks the settings button • User clicks on edit profile link. • User change details in the form. • User clicks submit changes button. If the user fills the form correctly then the user will be redirected to home page.
Post-condition	Personal information of the user gets updated.
Exceptional flow	If the user enters incorrect information the system displays an error message

Table 8 Use case description View in store

Description : 08	
Use case name	View in the store
Use case code	UC08
Use case description	View the current amount of blood in the store
Actor	Staff
Pre-condition	• The staff need to have an internet connection • The staff need to have a web browser to access our website • The staff must login into our system
Main flow	• The staff clicks the store and see the remaining blood amount
Post-condition	The current amount of blood in the store will displayed.
Exceptional flow	

3.2 Object Model

Object modeling is a technique used to represent real-world objects, concepts, or entities within a software system. Object diagrams show instances of classes and the relationships between these instances at a specific point in time.

3.2.1 Data Dictionary

Table 9 Data dictionary

Class	Attribute	Operation	Description
Donor	• ID • Name • Address • Blood group • Username • Password	• Register Donor () • Update() • Logout()	This class allows the donor to register, get new registered user, verify new user, edit the user profile , reset the user password and retrieve all the users ,login and logout into and from the system respectively
Blood	• ID • Type • Quantity	• Add Blood Unit () • Update Blood Unit ()	This class allows the blood to be add to the store and get updated whenever the new blood unit get in the system
Staff	• ID • Name • Address • Contact Information	• Update Donor Info () • View Donation History () • Respond To Request ()	This class allows the staff to manipulate and access all the related data in the system after they are signed in as a staff
Hospital	• ID • Name • Address • Contact information	• Register() • Request Blood ()	This class allows hospital to be registered in the system and request for blood unit form the bank
Order	• Hospital ID • Hospital Name • Hospital Address • Contact information	• Placing Order () • Updating Order Status ()	This class allows the hospital to place order in the system and change the pre ordered after the request have been accepted by the staff

3.3 Class Diagram

The Class diagram captures the logical structure of the system; the classes and things that make up the model. It is a static model, describing what exists and what attributes and behavior it has, rather than how something is done. Class diagrams are most useful to illustrate relationships between classes and interfaces.

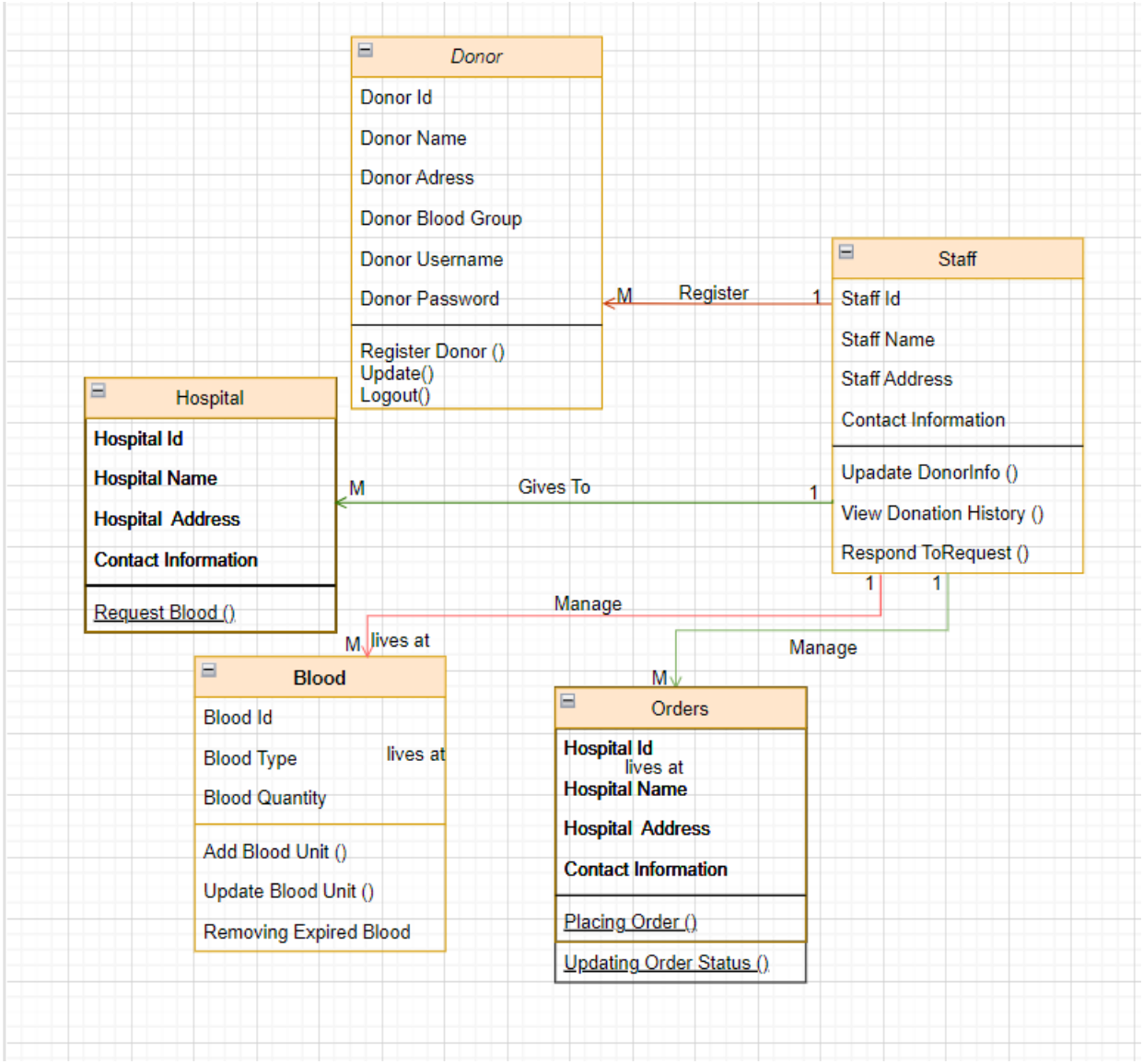

Figure 3 Class diagram

3.4 Dynamic Model

Dynamic Model refers to the representation of the behavior or functionality of a system over time. It focuses on how objects or components interact with each other and the sequence of actions that occur during the execution of the system. Dynamic models are typically used to depict the flow of control, data, and events within a software system.

3.4.1. Sequence Diagram

A sequence diagram in a UML is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

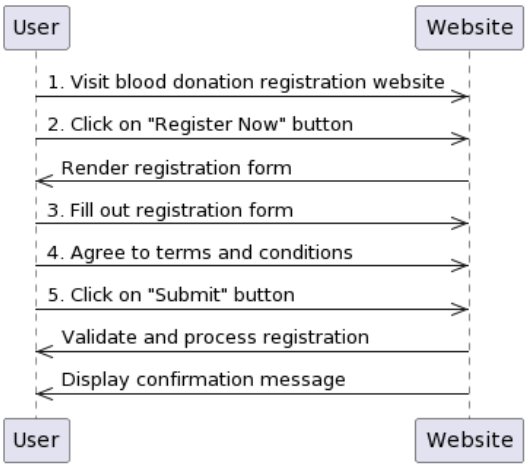

Figure 4 Sequence diagram for scenario 1

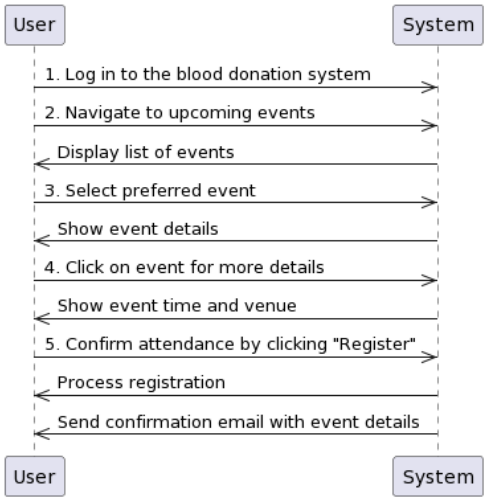

Figure 5 Sequence diagram for scenario 2

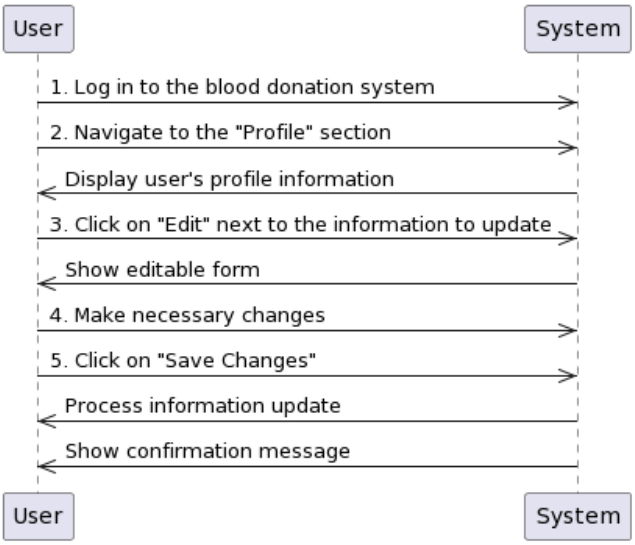

Figure 6 Sequence diagram for scenario 3

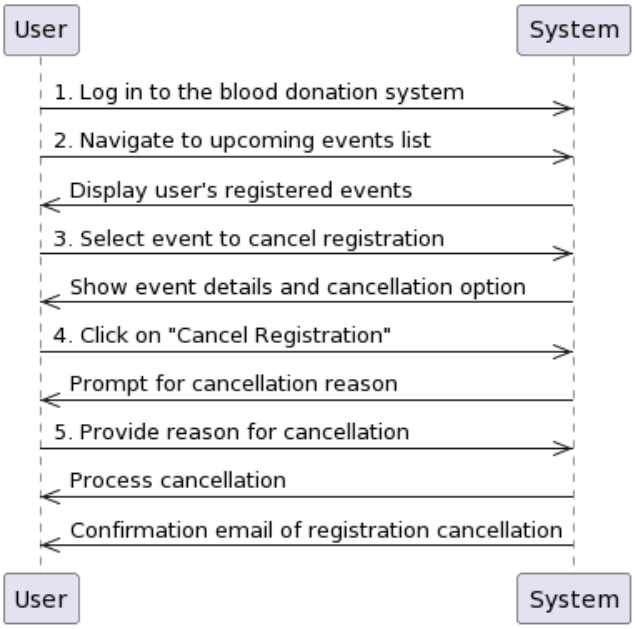

Figure 7 Sequence diagram for scenario 4

3.4.2 Activity Diagram

An activity diagram describes a system in terms of activities. Activities are states that represent the execution of a set of operations. The completion of these operations triggers a transition to another activity. Activity diagrams are similar to flowchart diagrams in that they can be used to represent control flow (i.e., the order in which operations occur) and data flow (i.e., the objects that are exchanged among operations).

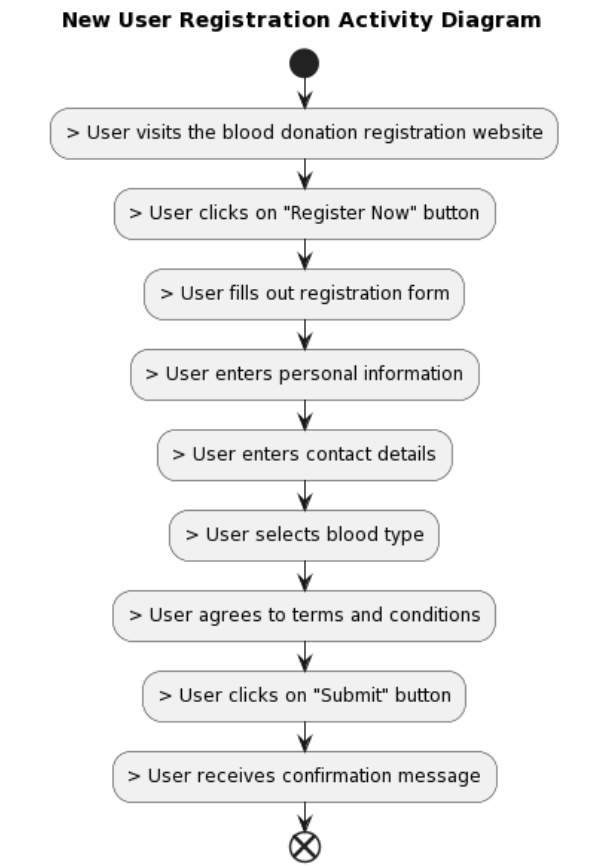

Figure 8 New user registration

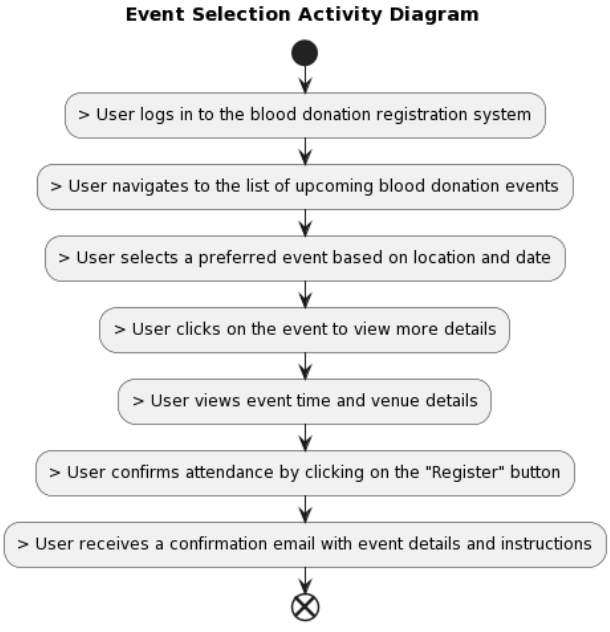

Figure 9 Selection activity

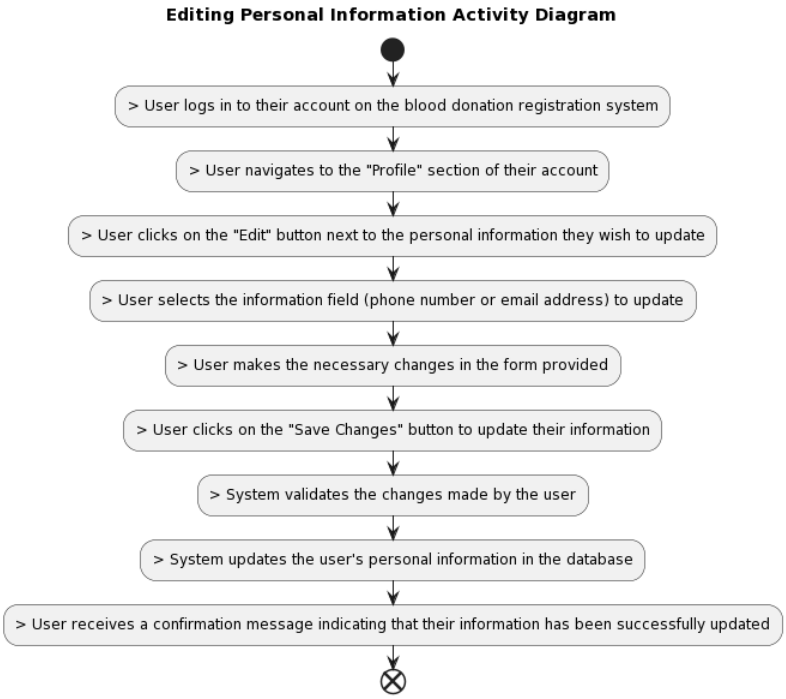

Figure 10 Editing profile

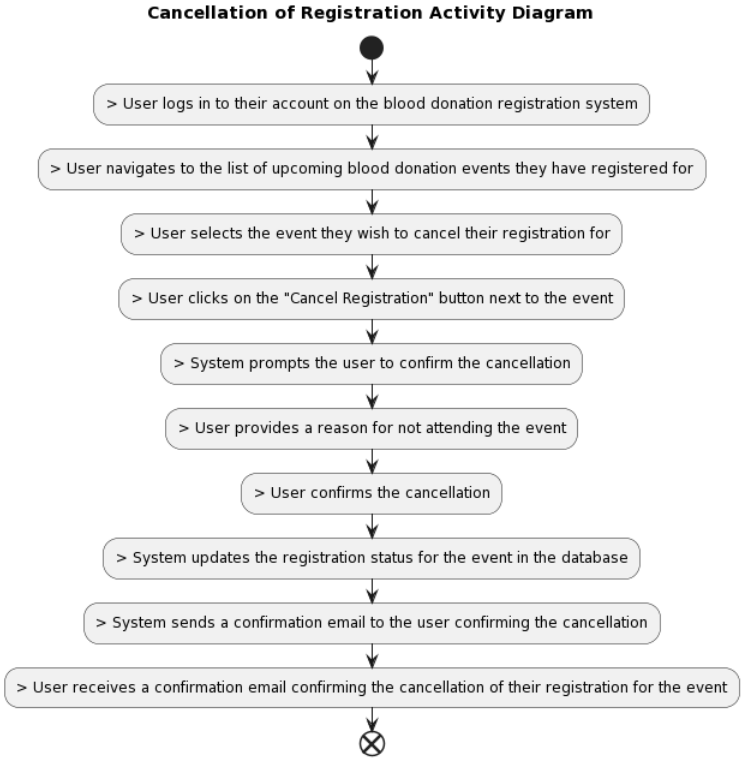

Figure 11 Logout profile

Chapter Four

System Design

4.1 Overview

The Donor Registration System is designed to streamline the process of registering donors for blood donation events. The system aims to provide a user-friendly interface where individuals can visit the blood donation website, fill out a registration form with their personal information, such as name, contact details, and blood type, and agree to the terms and conditions of the donation event. Upon submitting the form, the system validates the information provided and stores it securely in a database. An automated confirmation message is then displayed on the screen to notify the user of successful registration. The system ensures data accuracy, confidentiality, and efficient communication with donors, ultimately facilitating the registration process and encouraging more individuals to participate in lifesaving blood donation events.

4.1.1 Purpose of the System Design

The purpose of an overview system design for a Donor Registration System is to provide a high-level understanding of how the system functions and its key components. It serves as a roadmap for developers, designers, and stakeholders to grasp the system's architecture, interactions, and data flow, ensuring clarity on the system's objectives, scope, and functionality before delving into detailed design and implementation phases. This overview facilitates effective communication, decision-making, and alignment among project team members, guiding them towards developing a well-structured and efficient system for donor registration.

4.1.2 Design Goal

The design goal of an overview system design for a Donor Registration System is to establish a clear and cohesive blueprint that outlines the system's purpose, functionality, and interactions in a concise manner. This design aims to provide a holistic view of the system architecture, highlighting the key modules, data flows, and user interactions to ensure a structured approach towards development. By defining the system's scope, requirements, and high-level design elements, this overview design sets the foundation for a well-organized, scalable, and user-centric donor registration system.

4.2 Proposed system architecture

The Donor Registration System will be designed using a three-tier architecture comprising the presentation layer, application layer, and data layer.

1. Presentation Layer:

-The front-end will be developed using modern web technologies such as HTML, CSS, and JavaScript for a responsive and user-friendly interface.

- User interactions, form validations, and feedback messages will be handled here to ensure a seamless registration process.

2. Application Layer:

- A server-side application layer will be implemented using a suitable programming language like Java, Python, or Node.js for processing user inputs and business logic.

- This layer will manage user authentication, form submission handling, validation, and communication with the data layer.

3. Data Layer:

- A relational database management system (e.g., MySQL, PostgreSQL) will store donor information securely.

- The database will be normalized to ensure data integrity and efficiency in managing donor records, registration details, and event information.

4.3 Database Design

This our design for the database.

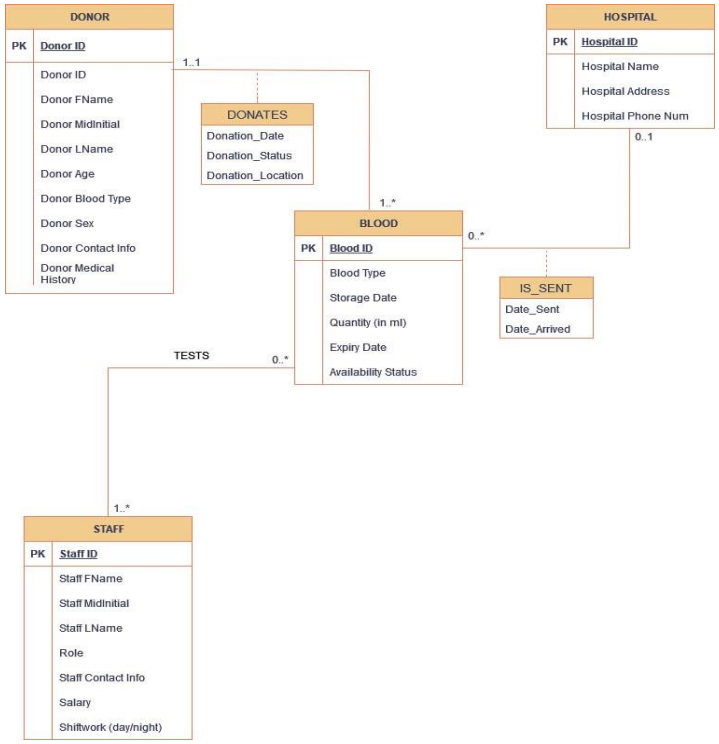

Figure 12 Data base design

4.4 Access control

In this system there are different actors with different access privileges. This part shows the privilege each actor has.

"The STAFF" has the following privileges.

- ✓ registration
- ✓ registering blood
- ✓ view analytics
- ✓ view organization growth
- ✓ view stock ratings
- ✓ update store detail
- ✓ delete profile
- ✓ view amount in store
- ✓ view report in store
- ✓ manage stuffs
- ✓ login
- ✓ logout
- ✓ edit profile

"THE DONOR" has the following profile

- ✓ see profile
- ✓ update profile
- ✓ view the update post on the page
- ✓ see last donated date
- ✓ login
- ✓ logout
- ✓ edit profile

"THE HOSPITAL" has the following profile

- ✓ see profile
- ✓ update profile
- ✓ view the update post on the page
- ✓ see the available blood that are distributed for the hospitals
- ✓ see the nearest blood bank store
- ✓ login
- ✓ logout
- ✓ edit profile

4.5 User interface design

Figure 13 Our home page

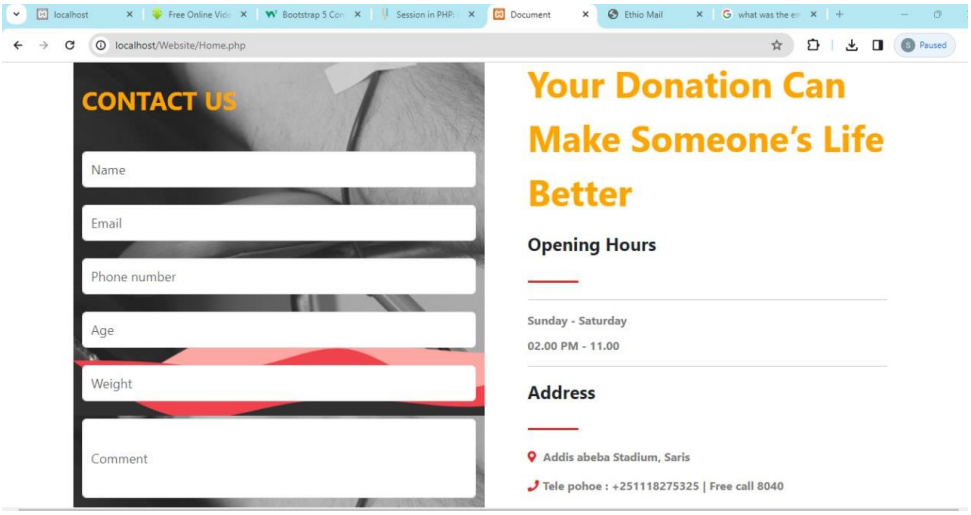

Figure 14 Our donation form

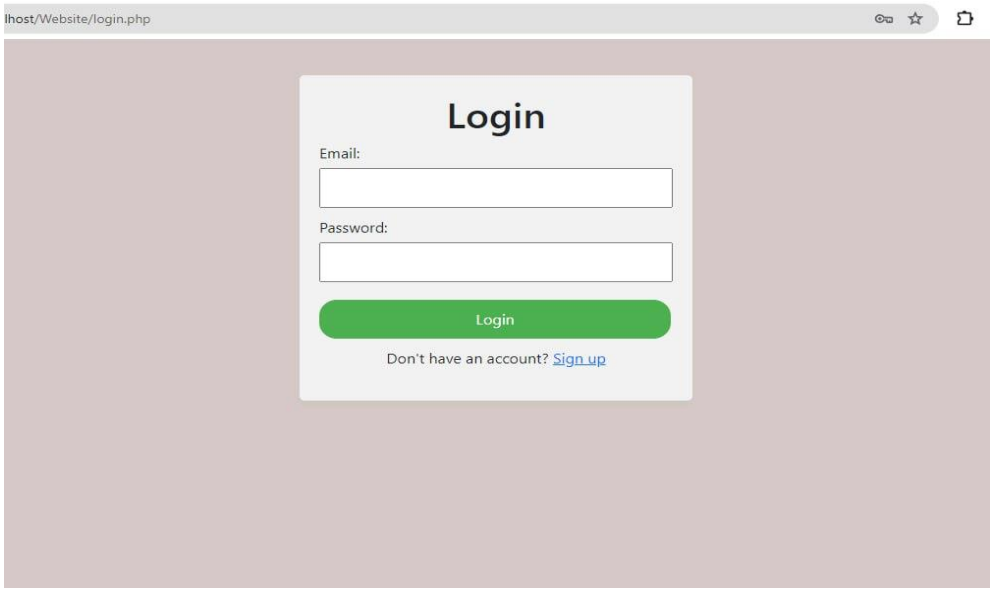

Figure 15 Our login(Sign up) page

Chapter Five

Implementation

5.1 Overview

Implementation is the process of integrating the system functions or the development of software and hardware based on the functional and non-functional requirements of the project. Our project implements the functional and non-functional requirements of the proposed system.

5.2 Development Tools

We use these tools to complete our projects.

Table 10 Development Tools

Tools	Use
Draw.Io	For drawing the UMLs
Vs Code	For coding programs
Microsoft word	For documenting
Microsoft powerPoint	For the slides
MySQL	For background database
Google Chrome	For our search engine

5.3 Prototypes

```php
<?php
    if (isset($_POST["submit"])) {
    $fullname = $_POST["fullname"];
    $email = $_POST["email"];
    $password = $_POST["password"];
    $passwordRepeat = $_POST["repeat_password"];

    $passwordHash = password_hash($password, PASSWORD_DEFAULT);

    $errors = array();

    if (empty($fullname) OR empty($email)OR empty($password) OR empty($passwordRepeat)) {
        array_push($errors, "All fields are required");
    }

    if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
        array_push($errors, "Email is not valid");
    }

    if (strlen($password) < 8) {
        array_push($errors, "Password must be at least 8 characters long");
    }

    if($password !== $passwordRepeat){
        array_push($errors, "Password does not much");
    }

    require_once "database.php";
    $sql = "SELECT * FROM users WHERE email = '$email'";
    $result = mysqli_query($conn, $sql);
    $rowCount = mysqli_num_rows($result);
    if ($rowCount > 0) {
        array_push($errors,"Email Already Exist!");
    }
```

```php
        if (count($errors) > 0) {
            foreach($errors as $error){
                echo "<div class='alert alert-danger'>$error </div>";
            }
        }else {
            $sql = "INSERT INTO users(full_name, email, password) VALUES( ?, ?, ?)";
            $stmt = mysqli_stmt_init($conn);
            $prepareStmt = mysqli_stmt_prepare($stmt, $sql);
            if ($sql) {
                mysqli_stmt_bind_param($stmt, "sss", $fullname, $email, $passwordHash);
                mysqli_stmt_execute($stmt);
                header("Location: Home.php");
            }else {
                die("Something Went Wrong");
            }
        }
    }
?>
```

Figure 16 Sample source code