

SMART BLOOD BANK MANAGEMENT SYSTEM



Project Advisor: Mubbarra Samreen

Submitted By

Mohsin Abbas	BSF2102877
Muhammad Haris	BSF2102966
Haroon	
Savaira Saeed	BSF2103017

University of Education, Lahore

SBMS

Smart Blood Bank Management System

BS in Information Technology 2021-25

A project submitted in partial fulfillment of the requirements for the award of
the degree of BS in information Technology.

DEPARTMENT OF INFORMATION TECNOLOGY

UNIVERISTY OF EDUCATION, LAHORE

D. G. KHAN CAMPUS

SBMS

“I hereby declare that I have read this project documentation and in my opinion this project is sufficient in terms of scope and quality for the award of the degree of BS in Information Technology.”

Project Primary Supervisor

Name: Mubbarra Samreen

Designation: Assistant Professor

Project Examiner

Name: Write name of Project Examiner here

Designation: Write designation of Examiner

STUDENT DECLARATION

I declare that this project title entitled “**Smart Blood Bank Management System**” is the result of my own research and development except as cited in the references. This project has not been accepted for any degree and is not concurrently submitted in candidate for any other degree. At any time if my statement is found to be incorrect even afterwards of BS in Information Technology/MSc in Information Technology, the university has the right to withdraw my BS in Information Technology.

Signature: _____

Name: Mohsin Abbas

Date: _____

Signature: _____

Name: Muhammad Haris Haroon

Date: _____

Signature: _____

Name: Savaira Saeed

Date: _____

PLAGIARISM UNDERTAKENING

I solemnly declare that project work presented in this documentation entitles “Smart Blood Bank Management System” is solely my work with no significant contribution from any other person. Small contribution/help wherever taken has been acknowledged and that complete project has been written by me.

I understand that zero tolerance policy of the HEC and University of Education, Lahore towards plagiarism. Therefore, we as an author of the above titled project declare that no portion of my project documentation and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled project even after award of BS degree, the University reserve the rights to withdraw/revoke my BS degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized projects.

Signature: _____

Name: Mohsin Abbas

Date: _____

Signature: _____

Name: Muhammad Haris Haroon

Date: _____

Signature: _____

Name: Savaira Saeed

Date: _____

CERTIFICATE OF APPROVAL

This is to certify that the project work presented in this documentation entitled, **“Smart Blood Bank Management System”**, was conducted by **“Mohsin Abbas”**, **“Muhammad Haris Haroon”**, **“Savaira Saeed”**, under the supervision of **“Mubbarra Samreen”**. No part of this project has been submitted anywhere else for any degree. This project is submitted to the **“DG Khan Campus, University of Education, Lahore”** is partial fulfillment of the requirements of the degree of BS in Information Technology.

Signature: _____

Name: Mohsin Abbas

Date: _____

Signature: _____

Name: Muhammad Haris Haroon

Date: _____

Signature: _____

Name: Savaira Saeed

Date: _____

OFFICE OF CONTROLLER OF EXAMINATION

NOTIFICATION

No: _____

Date: _____

It is notified for the nomination of all the concerned that Mr. Mohsin Abbas, Mr. Muhammad Haris Haroon, and Ms. Savaira Saeed BS student of Information Technology University of Education has completed all the requirements for the award of BS Degree in the discipline of Information Technology as per detail given here under:

BS in Information Technology

Cumulative Result

Credit Hours: _____					Cumulative Grade Point Average (CGPA)
Registration No	Complete Name	Course work	Project	Total	
21-UE-04232	Mohsin Abbas		SBMS		
21-UE-04335	Muhammad Haris Haroon		SBMS		
21-UE-04393	Savaira Saeed		SBMS		

Project Title: Smart Blood Bank Management System

Name of Supervisor: Mubbarra Samreen

Signed by
Controller of Examination

LIST OF CONTENTS

Sr. #	DESCRIPTION	Page #
1	Student Declaration	I
2	Plagiarism Undertaking	II
3	Certificate of Approval	III
4	Office of Controller of Examination Notification	IV
4	List of Contents	V
5	List of Figures	VIII
6	Acknowledgement	IX
7	Abstract	X

CHAPTERS

1	GATHERING AND ANALYZING INFORMATION	1-06
	1.1 Introduction	01
	1.2 Problem Statement	02
	1.3 Project Goal and Objectives	02
	1.4 Research Questions	02
	1.5 Methodology	03
	1.5.1 Available Methodologies.....	03
	1.5.2 Chosen Methodology.....	03
	1.5.3 Reasons for Chosen Methodology.....	03
	1.6 Website Architecture	04

	1.7 Definitions, Acronyms and Abbreviations.....	06
2	SOFTWARE REQUIREMENTS SPECIFICATION	07-08
	2.1 Functional Requirements	07
	2.2 Non Functional Requirements	08
3	ANALYSIS	09-16
	3.1 Use Case Scenario	09
	3.2 UC Name: Login Management.....	9
	3.3 UC Name: Donor Directory Access.....	10
	3.4 UC Name: Blood Request System.....	11
	3.5 UC Name: Admin Panel Access.....	12
	3.6 UC Name: Notifications.....	13
	3.7 UC Name: Educational Resources.....	15
	3.8 Use Case Diagram.....	16
4	DESIGN	17-21
	4.1 Architecture Diagram Description.....	17
	4.2 ER Diagram Description.....	18
	4.3 DB Schema Diagram Description.....	19
	4.4 Class Diagram Description.....	20
	4.5 Sequence Diagram Description.....	21
5	GRAPHICAL USER INTERFACE	22-28
	5.1 Design Principles	22

	5.2 Key GUI Screens	22
	5.2.1 Signup Page	22
	5.2.2 Login Page	23
	5.2.3 Donor Dashboard.....	24
	5.2.4 Blood Bank Admin Dashboard.....	25
	5.2.5 System Admin Panel.....	26
	5.3 Navigation and User Flow.....	27
	5.4 Interface Tools and Technologies.....	28
6	TESTING	29-31
	6.1 Introduction.....	29
	6.2 Test Scenario (Sample Attached).....	29
	6.3 Test Plan.....	29
	6.4 Definition of Test Cases.....	30
	6.5 Test Case Specifications.....	30
	6.6 Test Case Results.....	30
	6.6.1 Black Box Test Cases.....	30
	6.6.2 White Box Test Cases.....	31
7	CONCLUSION AND FUTURE WORK	32-33
	7.1 Conclusion.....	32
	7.2 Future Work.....	32
	REFERENCES	34
	Appendix	35

LIST OF FIGURES

Figure #	DESCRIPTIONS	Page #
1.1	Figure 1.1: Website Architecture.....	05
3.1	Figure 3.1: Use Case Diagram	16
4.1	Figure 4.1: Architecture Diagram.....	17
4.2	Figure 4.2: ER Diagram.....	18
4.3	Figure 4.3: DB Schema Diagram.....	19
4.4	Figure 4.4: Class Diagram.....	20
4.5	Figure 4.5: Sequence Diagram.....	21
5.1	Figure 5.1: Signup Page	23
5.2	Figure 5.2: Login Page	24
5.3	Figure 5.3: Donor Dashboard	25
5.4	Figure 5.4: Blood Bank Admin Dashboard	26
5.5	Figure 5.5: System Admin Panel	27
5.6	Figure 5.6: Navigation and User Flow	27

ACKNOWLEDGEMENT

We truly acknowledge the cooperation and help make by Mubbarra Samreen, Assistant Professor, University of Education, Lahore, DG Khan Campus. She has been a constant source of guidance throughout the course of this project.

We are grateful to her for the help and guidance she provided throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

Date: May 09, 2025

ABSTRACT

The Smart Blood Bank System is a web-based application designed to streamline and enhance the process of blood donation and distribution. Traditional blood bank systems often suffer from inefficiencies such as manual record keeping, lack of real-time data, and communication gaps between donors, hospitals, and blood banks. This project aims to overcome these limitations by providing a centralized platform that enables real-time tracking of blood inventory, automated donor notifications, and seamless communication among users.

The system allows hospitals and blood banks to update and monitor blood stock levels, while registered donors can receive alerts based on their blood type and location when their donation is needed. By integrating features such as donor eligibility checks, location-based search, and secure user management, the system ensures faster response times in emergencies and better utilization of available blood units. Ultimately, the Smart Blood Bank System seeks to save lives by improving accessibility, transparency, and efficiency in the blood donation ecosystem.

CHAPTER 1

GATHERING AND ANALYZING INFORMATION

1.1 Introduction

The Smart Blood Bank Management System aims to create a platform that connects blood donors with recipients in need, facilitating seamless access to blood supplies. With an easy-to-use interface, this platform provides users with a searchable directory of blood donors, a blood request system, and educational resources about blood donation. The system will also support hospital and blood bank operations by simplifying donation management and improving response times for emergencies.

- a. **Project Title:** Smart Blood Bank Management System
- b. **Overview Statement:** The Smart Blood Bank Management System is a website that helps to connect people who need blood with donors who are willing to give it.
- c. **The Main Goal:** The main goal of Smart Blood Bank Management System is to create an efficient, user-friendly platform that connects blood donors with recipients in need, making the process of finding, requesting, and donating blood faster and more accessible.
- d. **Project Objectives:** Develop a user-friendly platform for donors and recipients to easily register, manage profiles, and track blood donations and requests.
- e. **High Level System Components:** User Interface, Appointment Scheduling, Communication Module, Central Database.
- f. **List of Optional Functional Units:** Mobile Application, Social Media integration.
- g. **Exclusions:** Direct Blood Transfusion Management, Third-Party Payment Processing, Real-Time Inventory Management, Support for Non-Blood Donation Services
- h. **Application Architecture:** HTML/AJAX/JQuery/JavaScript for frontend and PHP for backend.

- i. **Gantt Chart:** describe the detailed timeline for project implementation in the given time.

1.2 Problem Statement

The traditional blood bank management system faces challenges such as inefficient donor-recipient matching, lack of real-time data, manual record-keeping, and delayed communication between hospitals and donors. These issues can lead to critical delays in blood availability, especially during emergencies. The Smart Blood Bank Management System aims to digitize and streamline the blood donation process by providing a centralized platform that ensures real-time tracking of blood inventory, automated notifications to eligible donors, and efficient coordination between blood banks, hospitals, and donors.

1.3 Project Goal and Objectives

- Develop a user-friendly platform for donors and recipients to easily register, manage profiles, and track blood donations and requests.
- Create a searchable directory where recipients can find blood donors based on blood type and location.
- Provide real-time notifications to donors when a matching blood request is submitted.
- Establish an admin panel for hospitals and administrators to manage user accounts, oversee blood requests, and monitor donation activities.
- Promote awareness about blood donation through an information hub that educates users on donation eligibility and health benefits.
- Implement a blood request system that allows recipients to submit requests and receive timely responses from matching donors.

1.4 Research Questions

1. How can real-time tracking and automation improve the efficiency of blood inventory management in blood banks?

2. What technologies can be used to ensure accurate and secure donor-recipient matching?
3. How can automated notifications and communication tools enhance coordination among donors, hospitals, and blood banks?
4. What measures can be taken to protect sensitive health information within a smart blood bank system?
5. What are the main challenges in implementing a Smart Blood Bank Management System across different regions or healthcare settings?

1.5 Methodology

1.5.1 Available Methodologies

Several software development methodologies were considered for this project, including:

- **Waterfall Model** – A linear and sequential approach with clearly defined phases.
- **V-Model** – An extension of the Waterfall Model with a strong emphasis on validation and verification.
- **Agile Methodology** – An iterative approach that promotes flexibility, continuous feedback, and rapid delivery.
- **Spiral Model** – A risk-driven model combining iterative development with systematic aspects of the Waterfall model.

1.5.2 Chosen Methodology

The Agile Methodology was selected for this project.

1.5.3 Reasons for Chosen Methodology

- **Flexibility:** Agile allows for frequent updates and iterative improvements based on feedback.
- **User Involvement:** Continuous communication with stakeholders ensures the system aligns with real-world requirements.
- **Faster Delivery:** Agile promotes rapid development cycles (sprints), allowing features to be released and tested quickly.

- **Improved Quality:** Regular testing and integration at each sprint help ensure a more stable and reliable system.

1.6 Website Architecture

Presentation Layer (Front-End)

- **Technology:** HTML, CSS, JavaScript (with optional frameworks like React)
- **Purpose:** Interface where users interact with the system
- **Key Components:**
 - 1) Donor Registration & Login
 - 2) Blood Availability Search
 - 3) Request Blood Form
 - 4) Admin Dashboard
 - 5) Notifications Panel

Application Layer (Back-End)

- **Technology:** PHP / Node.js / Python (e.g., Django or Flask)
- **Purpose:** Handles business logic, user authentication, and data processing
- **Key Functions:**
 - 1) Validate donor eligibility
 - 2) Process blood requests
 - 3) Update inventory
 - 4) Manage user roles and access
 - 5) Send notifications (email/SMS)

Database Layer

- **Technology:** MySQL / PostgreSQL / MongoDB
- **Purpose:** Stores and manages system data
- **Key Data Stored:**

- 1) Donor profiles and history
- 2) Blood inventory records
- 3) Hospital/blood request data
- 4) Admin logs and reports

Hosting & Deployment

- **Web Server:** Apache / Nginx
- **Hosting:** Cloud platform (e.g., AWS, Heroku, or shared hosting)
- **Security Features:** SSL, data encryption, role-based access control

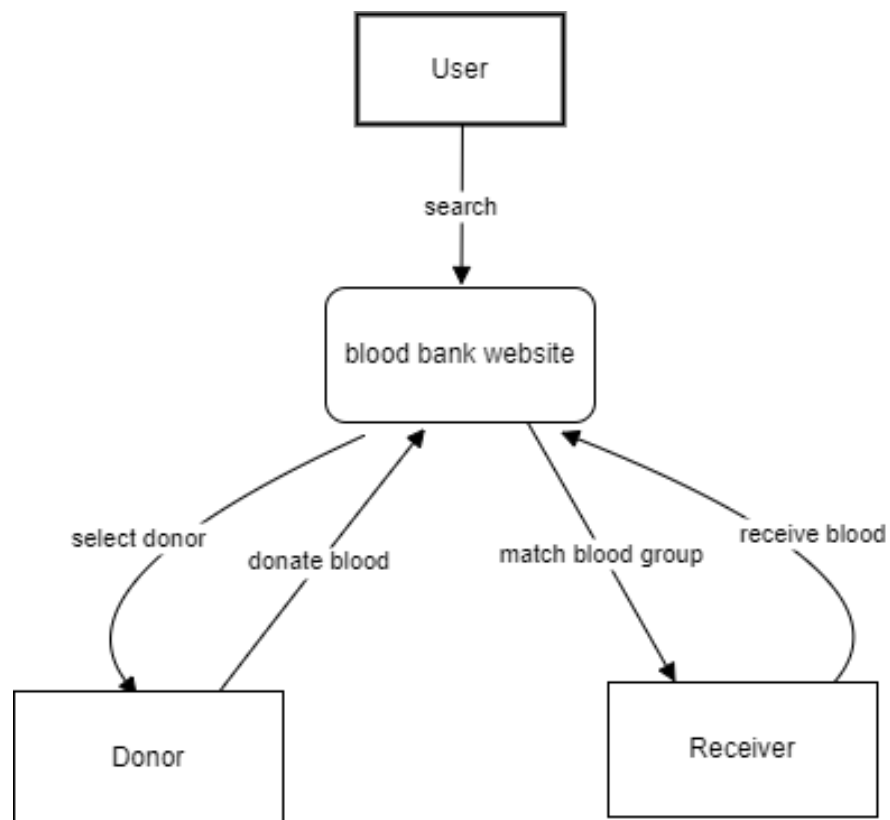


Figure 1.1: Website Architecture

Architecture Overview:

1. Blood donor and receiver register their accounts and submit details on the website.
2. Receiver search for their blood group and send request after matching blood with donor.
3. A notification will be received to the donor for donating blood.
4. If donor agrees, it accept the request and then notification will be send to the receiver.

5. Donor donate the blood and receiver receive it.

1.7 Definitions, Acronyms and Abbreviations

Term / Acronym Definition

SBMS	Smart Blood Bank Management System
Donor	A person who voluntarily gives blood for transfusion purposes
Recipient	A patient who receives a blood transfusion
Inventory	The stored blood units available at a blood bank
Real-time Tracking	The live monitoring of blood stock and related data
API	Application Programming Interface – used for communication between system components
UI	User Interface – the part of the system users interact with
Database	A structured collection of data used to store donor, recipient, and blood stock information
Notification System	A feature that sends alerts to users (e.g., when a donation is needed)

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Functional Requirements

FR No.	Functional Requirement	Description
FR1	Login Management	<p>Allow new users (donors, recipients, hospital staff) to register.</p> <p>Enable secure login with role-based access for different user types.</p>
FR2	Donor Directory	<p>The system must provide a searchable directory for recipients to find donors based on blood type and location.</p> <p>Donors must be able to update their availability and blood donation status.</p>
FR3	Blood Request System	<p>The system must allow recipients to submit blood requests, specifying their location and required blood type.</p> <p>When a request is made, the system must automatically notify matching donors via email or SMS.</p> <p>Donors can accept or decline requests, with notifications sent to the recipient upon acceptance.</p>
FR4	Admin Panel Access	<p>Hospitals and administrators must have access to an admin panel to manage user accounts, oversee blood requests, and monitor donation activities.</p>

		Admins must be able to generate reports on donation statistics, user activity, and blood inventory.
FR5	Notifications	Real-time notifications must be sent to donor when a recipient requests a matching blood type. Notifications must be delivered via email or SMS and include details about the recipient's location and urgency.
FR6	Educational Resources	The system must provide an educational section for users to learn about blood donation, eligibility, and health benefits.

2.2 Non-Functional Requirements

NFR No.	Non-Functional Requirement	Description
NFR1	Performance	The system should be able to handle 500 concurrent users with a response time of less than 3 seconds for key actions.
NFR2	Security	The system should follow encryption standards (e.g., HTTPS, SSL/TLS) to secure sensitive data during transfer.
NFR3	Availability	The platform must have 99.5% uptime and be accessible 24/7 to ensure uninterrupted access to learning materials.
NFR4	Usability	The interface should be intuitive and user-friendly, supporting responsiveness on different devices (desktop, mobile, tablet).

CHAPTER 3

ANALYSIS

3.1 Use Case Scenario

3.2 UC Name: Login Management

Functional Requirement No: FR1

Primary Actors/Stakeholders: Donor, Receiver

Secondary Actors/Stakeholders: Admin

Description: Users must log in with their account credentials to access the Smart Blood Bank System. Upon successful authentication, the system identifies the user's role (Admin, Blood Donor, or Blood Receiver) and provides access to role-specific features.

Preconditions: The user must have valid account with correct username and password.

Main Success Scenario (MSS):

1. The user connects to the system.
2. The user enters his/her username and password.
3. The system validates the username and password.
4. The system determines the user's role.
5. The system displays a list of actions the user can perform based on the user's role.

Alternative Scenario:

1. Invalid account user or pass
2. User already logged into the system

Post Conditions:

1. The admin will log in to the system
2. The admin has access to the functions of the system

Extensions:

1. The system determines that the password is incorrect for the username entered.
The system prompts the user to re-enter the password.
 - a) The system determines that the re-entered password is incorrect.

- b) The system provides the option for the user to retrieve a forgotten password.
- 2. The system determines that the username does not match a username for any registered account.
 - a) The system displays an error message indicating that username is invalid.
- 3. The system determines that the user has no role assigned in the system.
 - b) The system does not allow the user to access the system.

3.3 UC Name: Donor Directory Access

Functional Requirement No: FR2

Primary Actors/Stakeholders: Blood Receiver

Secondary Actors/Stakeholders: Admin, Blood Donor

Description: The Donor Directory feature allows blood receivers to search for available blood donors based on specific criteria such as blood type, location, and availability. Admins manage the donor directory, while blood donors can update their personal details and availability status.

Preconditions:

- a. The user (Blood Receiver, Blood Donor, or Admin) must be logged into the system.
- b. The Donor Directory must contain up-to-date donor information.

Main Success Scenario (MSS):

- 1. Blood Receiver navigates to the Donor Directory.
- 2. The system displays the search criteria.
- 3. The system determines the user's role.
- 4. Blood Receiver enters search criteria.
- 5. The System processes the search.
- 6. The system displays a list of matching blood donors.
- 7. Blood Receiver selects a donor and requests contact.

Alternative Scenario:

- 1. No matching donors found.

2. Invalid search criteria

Post Conditions:

1. Blood Receiver successfully views a list of matching donors.
2. Admin can manage and update the donor directory.
3. Blood Donor can update their availability.

Extensions:

Blood Donor updates their availability.

- a. The blood donor logs in and accesses their profile.
- b. The donor updates their availability status and saves the changes.
- c. The system refreshes the donor directory to reflect the updated availability.

Admin manages the donor directory.

- a. Admin logs into the system and navigates to the donor directory management section.
- b. Admin can add, delete, or update donor information.
- c. The system reflects the changes in the directory in real-time.

3.4 UC Name: Blood Request System

Functional Requirement No: FR3

Primary Actors/Stakeholders: Blood Receiver

Secondary Actors/Stakeholders: Admin, Blood Donor

Description:

The Blood Request System allows blood receivers to request specific blood types and quantities. The system notifies matching blood donors and admins about the request. Blood donors can respond to the requests based on their availability, and admins can manage and track requests.

Preconditions:

1. The blood receiver must be logged into the system.
2. The blood receiver must have an active profile with complete contact information.
3. The blood request system must be active, and blood donor availability should be updated.

Main Success Scenario (MSS):

1. Blood Receiver navigates to the Blood Request section.
2. The system displays a request form.
3. Blood Receiver enters the blood request details.
4. The system processes the blood request.
5. The system sends notifications.
6. Blood Donors respond to the request.

Alternative Scenario:

1. No matching donors available.
2. Incomplete blood request form.
3. Blood Donor does not respond.

Post Conditions:

1. Blood Receiver successfully submits a blood request.
2. Blood Donor responds to the request.
3. Admin manages the request system.

Extensions:

Blood Receiver cancels or modifies the request.

- a. The blood receiver navigates to their request history and selects an active request.
- b. The blood receiver can either cancel the request or modify details such as quantity or urgency.
- c. The system sends updated notifications to donors and admin, if applicable.

Admin monitors blood request activity.

- a. Admin logs into the system and navigates to the blood request management section.
- b. Admin can view all active, pending, and completed requests, as well as track response times from donors.
- c. Admin can manually escalate or reassign requests if necessary.

3.5 UC Name: Admin Panel Access

Functional Requirement No: FR4

Primary Actors/Stakeholders: Admin

Secondary Actors/Stakeholders: Doctors, Administrator

Description: The Admin Panel allows the administrator to manage the system's core functionalities, including managing blood donor and receiver profiles, monitoring blood requests, viewing system statistics, and handling notifications.

Preconditions: The admin must be logged into the system with proper credentials and admin privileges.

Main Success Scenario (MSS):

1. Admin logs into the system.
2. Admin accesses the Admin Panel.
3. Admin manages donor and receiver profiles.
4. Admin monitors blood requests.
5. Admin manages notifications.

Alternative Scenario:

1. Invalid login credentials.
2. Admin encounters an error while updating profiles.

Post Conditions:

1. Admin successfully manages profiles, requests, and notifications.
2. Admin reviews system performance.

Extensions:

Admin escalates urgent blood requests.

- a. Admin manually assigns priority to urgent blood requests, ensuring that donors are immediately notified.

Admin audits system activity.

- a. Admin reviews logs of system changes, blood request histories, and notifications sent.

3.6 UC Name: Notifications

Functional Requirement No: FR5

Primary Actors/Stakeholders: Donor, Receiver

Secondary Actors/Stakeholders: Admin

Description: The notification system alerts blood donors, blood receivers, and admins about important system events, such as new blood requests, donor responses, or updates to the system. Notifications ensure timely communication and action between stakeholders.

Preconditions:

1. The user (donor, receiver, or admin) must be logged into the system.
2. The notification system must be active, and user profiles must have valid contact details (email, phone number, etc.).

Main Success Scenario (MSS):

1. The system triggers a notification event.
2. The system generates the appropriate notification.
3. The system sends notifications to the stakeholders.
4. Stakeholders receive and respond to notifications.
5. Admin monitors notifications

Alternative Scenario:

1. Notification delivery failure.
2. User does not respond to notifications.

Post Conditions:

1. Blood donors and blood receivers are notified
2. Admin monitors notification status.

Extensions:

Customizable notification preferences.

- a. Users (donors, receivers, or admin) can configure their notification preferences (e.g., receive notifications only via SMS, email, or in-app alerts).

Urgent notifications

- a. In cases of urgent blood needs, the system prioritizes notifications and sends them with high importance, ensuring that donors respond quickly.

3.7 UC Name: Educational Resources

Functional Requirement No: FR6

Primary Actors/Stakeholders: Donor, Receiver

Secondary Actors/Stakeholders: Admin

Description: The Educational Resources section provides blood donors and receivers with access to informative content about blood donation, health guidelines, and best practices. It includes articles, videos, and guides that help users understand the importance of blood donation, the process, and health tips.

Preconditions: The educational content must be available and up-to-date.

Main Success Scenario (MSS):

1. User navigates to the Educational Resources section.
2. The system displays the available educational content.
3. User selects and accesses a resource.
4. User reads or watches the content.

Alternative Scenario:

1. Content is unavailable.
2. User experiences loading issues.

Post Conditions:

1. User successfully accesses educational resources.
2. Admin reviews and updates content.

Extensions:

Admin reviews and updates content.

- a. The admin can upload new articles, videos, or health tips and remove outdated materials.

User provides feedback on resources.

- a. Users can provide feedback or rate the educational resources, helping improve the quality of future content.

3.8 Use Case Diagram

Description:

This System allows users to register as donors, recipients, or hospitals, manage profiles, and log in to access personalized dashboards. Recipients can search for donors based on blood type and location, and submit blood requests, which automatically notify matching donors via email or SMS. Donors can accept or decline these requests, while hospitals and admins can manage the system via an admin panel, track blood requests, and generate reports. Additionally, the system provides educational resources on blood donation, ensuring users stay informed.

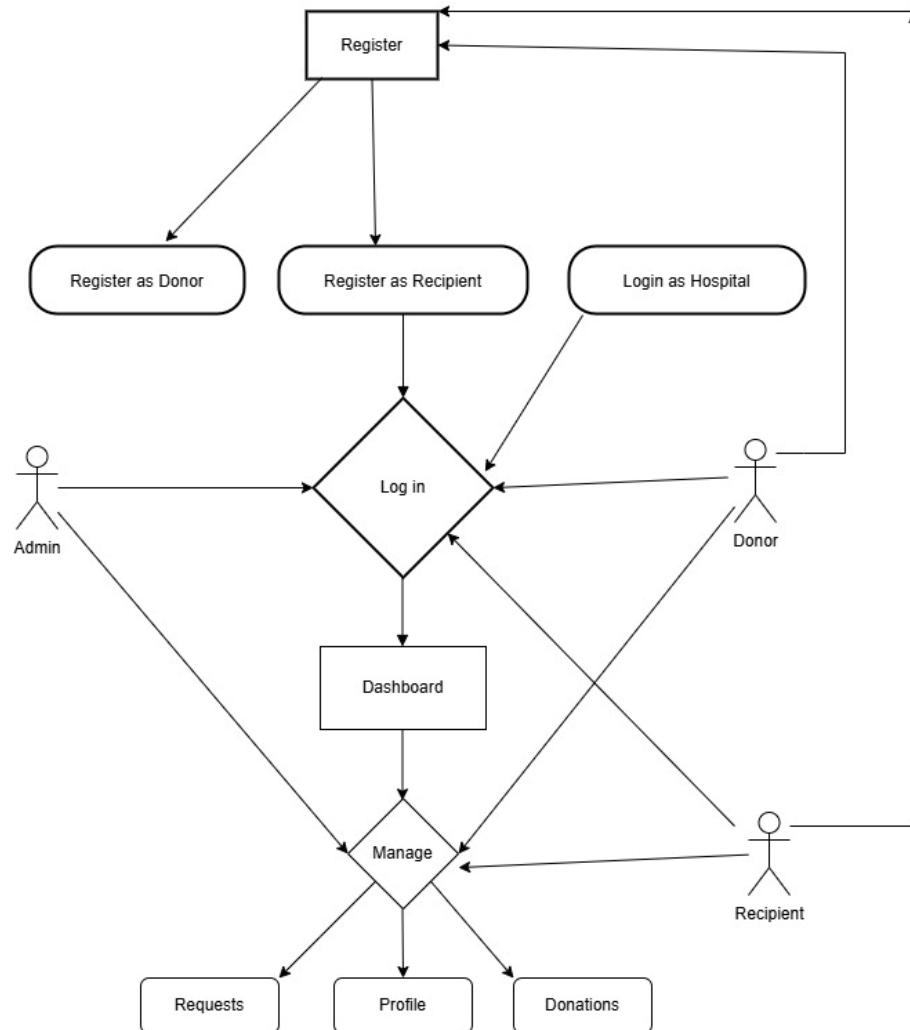


Figure 3.1: Use Case Diagram

CHAPTER 4

DESIGN

4.1 Architecture Diagram Description

The **user interface layer** provides separate portals for donors, recipients, and administrators to interact with the system. The **application logic layer** handles core functionalities, including login and registration with role-based access, recipient and donor directory management, notifications (via email/SMS), admin management (reports and statistics), and an educational section for blood donation resources. The **database layer** stores critical data, including user information (recipients, donors, admins), blood requests, and notification logs. The **infrastructure layer** supports the system with cloud hosting for servers and storage, backup systems for data protection, and third-party APIs for sending emails and SMS notifications. This layered design ensures scalability, security, and ease of maintenance.

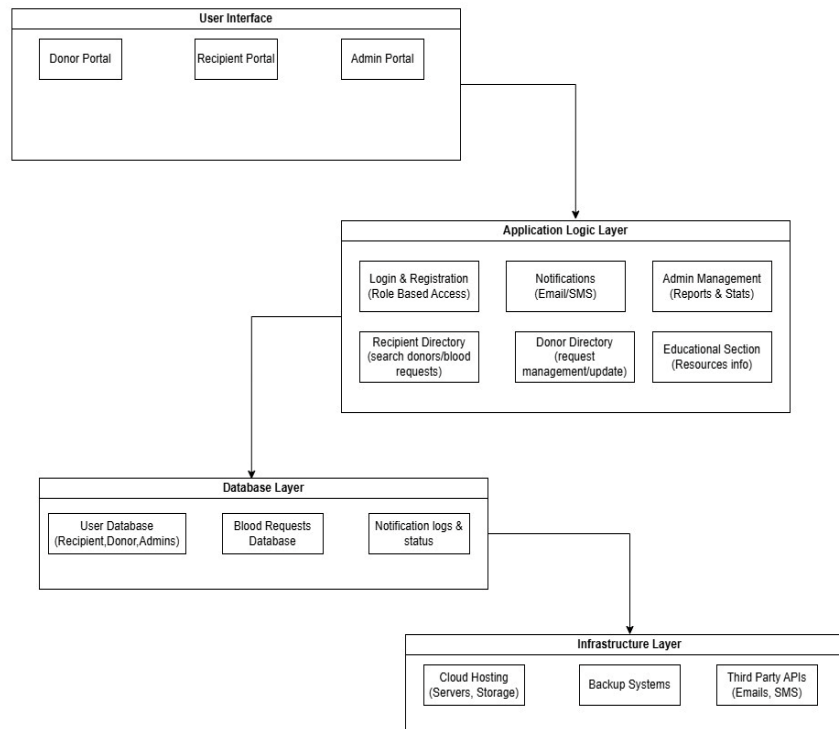


Figure 4.1: Architecture Diagram

4.2 ER Diagram Description

The **USER** entity is central, containing attributes like name, email, gender, and role, and links to other entities such as **Admin**, **Donor**, and **Recipient**. The **admin** entity manages users and reports, where reports are generated through the **report** entity, which includes attributes like report Id and type. The **donor** and **recipient** entities are connected via the **blood request** entity, which tracks blood requests with attributes like blood type, urgency, and status. Donors fulfill requests through the **donation** entity, which logs donation details, including date and status. The relationships illustrate workflows such as donors donating blood to recipients, admins managing users, and the system tracking and generating reports on activities. This model represents the operational and functional structure of the blood bank system.

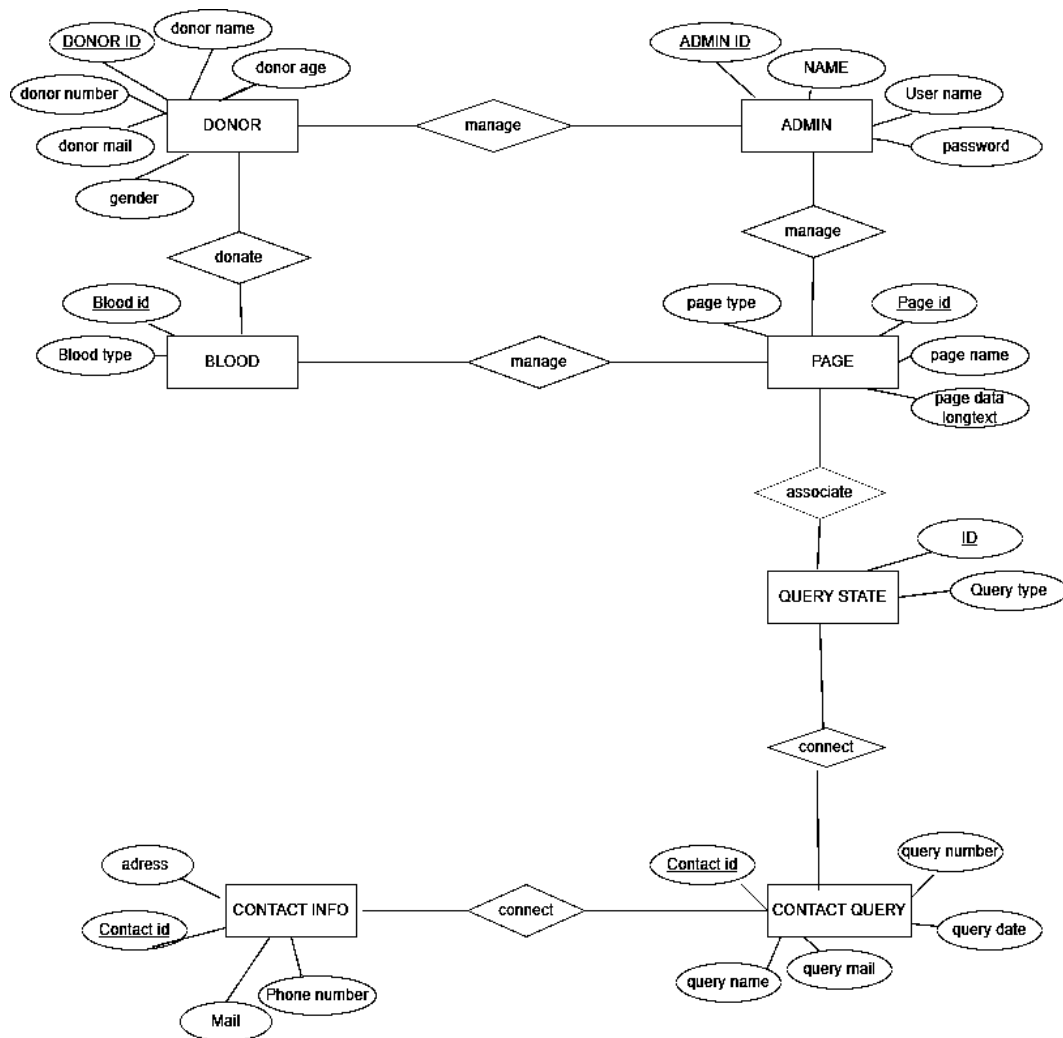


Figure 4.2: ER Diagram

4.3 DB Schema Diagram Description

User, which acts as a central table linked to **admin**, **recipient**, and **donor** tables, defining roles and personal information. The **blood request** table records requests made by recipients, linked to both recipients and donors. The **donation** table tracks donation activities, associating donors with specific blood requests. The **report** table enables administrators to generate and manage system reports. Relationships between tables ensure efficient data management, including tracking user roles, donation statuses, blood requests, and report generation.

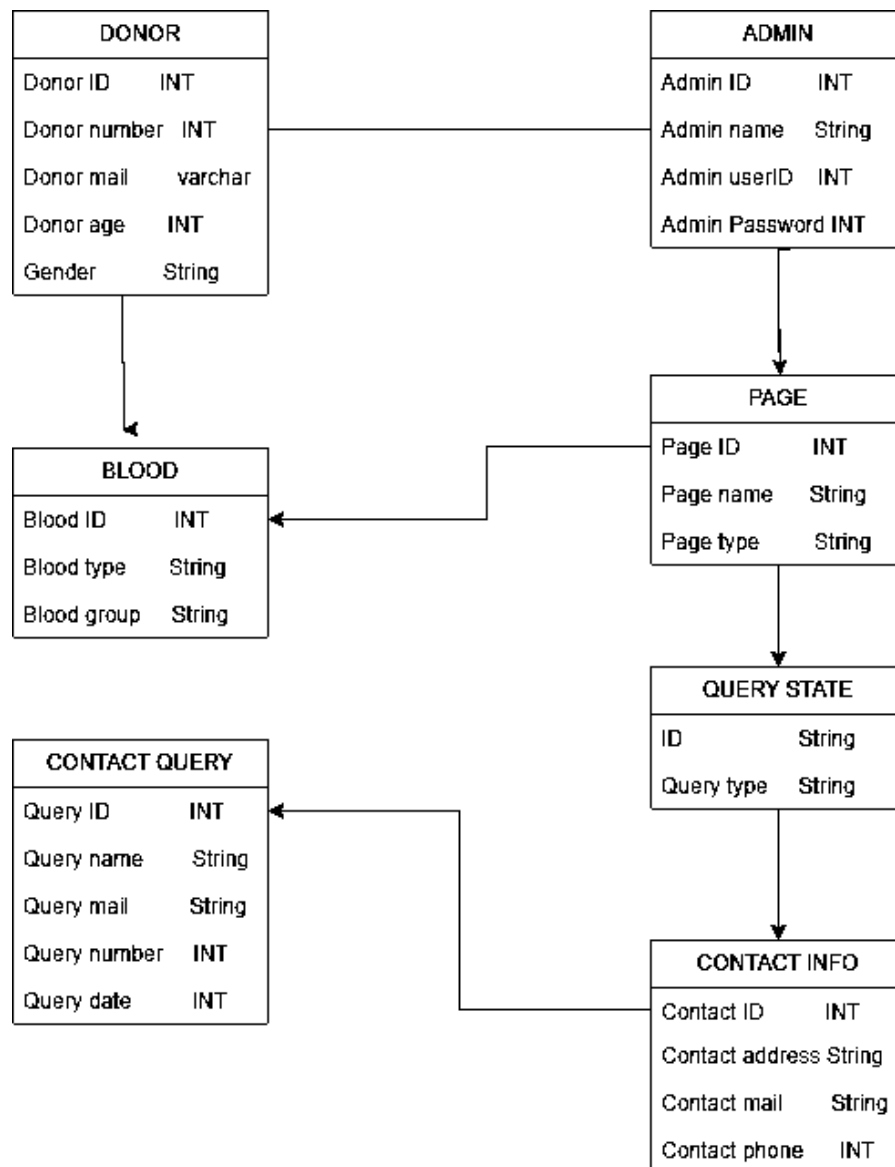
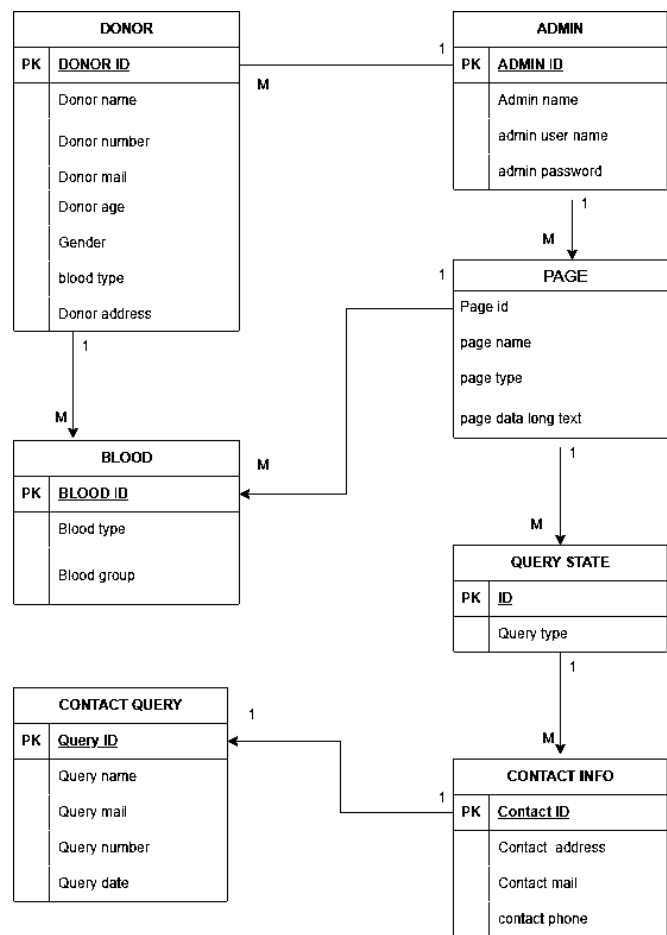


Figure 4.3: DB Schema Diagram

4.4 Class Diagram Description

It consists of several interconnected entities: **User**, **Admin**, **Recipient**, **Donor**, **Blood Request**, **Donation**, and **Report**. The **user** table acts as a central entity, containing general user details such as name, email, role, and gender, and links to specific roles like **admin**, **donor**, and **recipient** through their respective foreign keys. The **recipient** table stores details of blood recipients, such as blood type, location, and gender. The **donor** table includes blood donors' details, such as blood type, location, and availability, while also associating with **recipient** for potential matches. The **blood request** table tracks requests made by recipients, detailing blood type, urgency, location, and request status. The **donation** table records the history of donations made by donors to satisfy blood requests, capturing donation dates and statuses. At the end, the **report** table, linked to **admin**, allows for generating reports on the system's operations and user management.



1

Figure 4.4: Class Diagram

4.5 Sequence Diagram Description

This diagram illustrates a sequence of interactions among users, the system, and the database for three main processes: login and registration, profile updating, and blood requests. In the login & registration section, users either log in with credentials or register if not already a user. The system validates the credentials or registration details with the database and provides appropriate feedback. The profile updating section shows users (or admins) updating profile details, with the system validating and saving the changes in the database. Admins can also manage user profiles (adding, updating, or deleting accounts). Finally, the Blood Request section outlines the process where recipients search for matching donors, the system checks availability from the database, and notifies the recipient. If a donor is available, the request is sent via email or SMS, and the donor can accept or reject it. If accepted, the match is finalized; otherwise, another donor is sought.

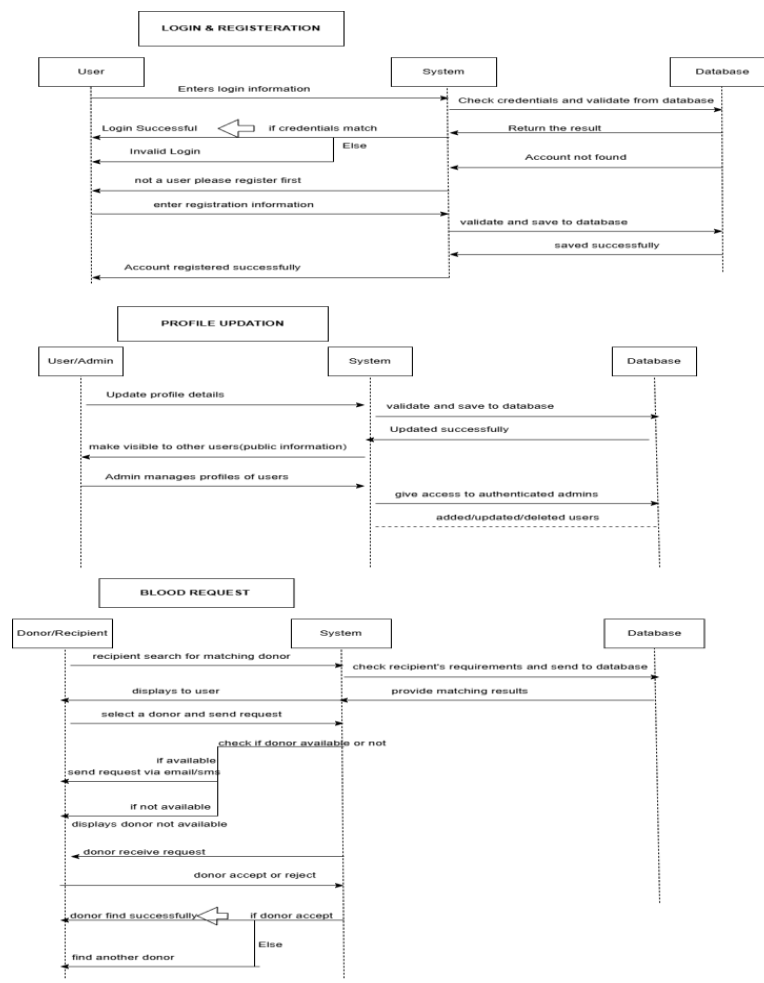


Figure 4.5: Sequence Diagram

CHAPTER 5

GRAPHICAL USER INTERFACES

Graphical User Interface (GUI) design plays a vital role in ensuring that the system is user-friendly, accessible, and efficient. A well-designed interface allows users to navigate the system intuitively, reducing the learning curve and improving the overall experience. In the **Smart Blood Bank Management System**, GUIs have been carefully developed for each user role—donors, hospitals, blood bank administrators, and system admins—to match their specific needs and responsibilities.

5.1 Design Principles

The GUIs were developed following key user interface design principles:

- **Simplicity:** Interfaces are clean, uncluttered, and focused on the user's primary tasks.
- **Consistency:** Common elements (like buttons, fonts, and colors) are reused across the system for familiarity.
- **Responsiveness:** Pages adapt to various screen sizes including desktops, tablets, and mobile devices.
- **Feedback:** Users receive confirmation or alerts for every action (e.g., successful donation request, failed login).
- **Accessibility:** High-contrast text, tooltips, and simple layouts help users of all backgrounds and abilities.

5.2 Key GUI Screens

5.2.1 Signup Page

The **Signup Page** is the entry point for new donors to register in the Smart Blood Bank Management System (SBMS). It collects essential information such as name, contact details, blood group, and location to create a donor profile. This data is crucial for

SBMS

managing blood donations, sending alerts, and ensuring the availability of compatible blood types during emergencies.

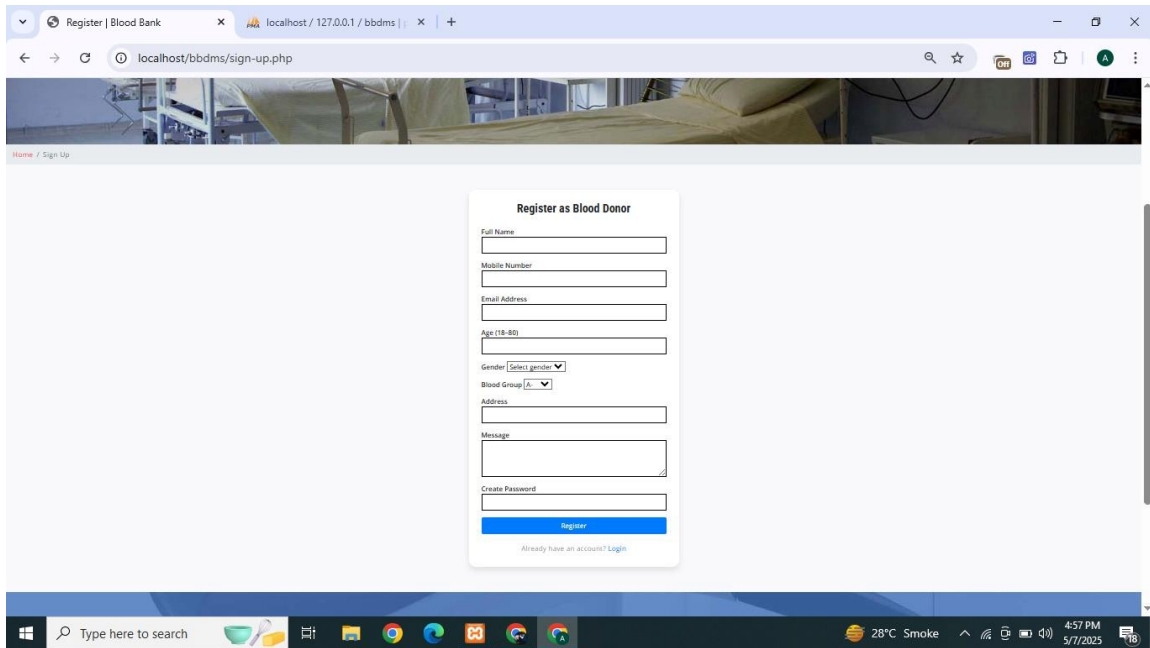


Figure 5.1: Signup Page

5.2.2 Login Page

To authenticate users and direct them to their respective dashboards.

1. Username and password input
2. Role selection (Donor / Hospital / Admin)
3. Forgot password link
4. Error messages for invalid input

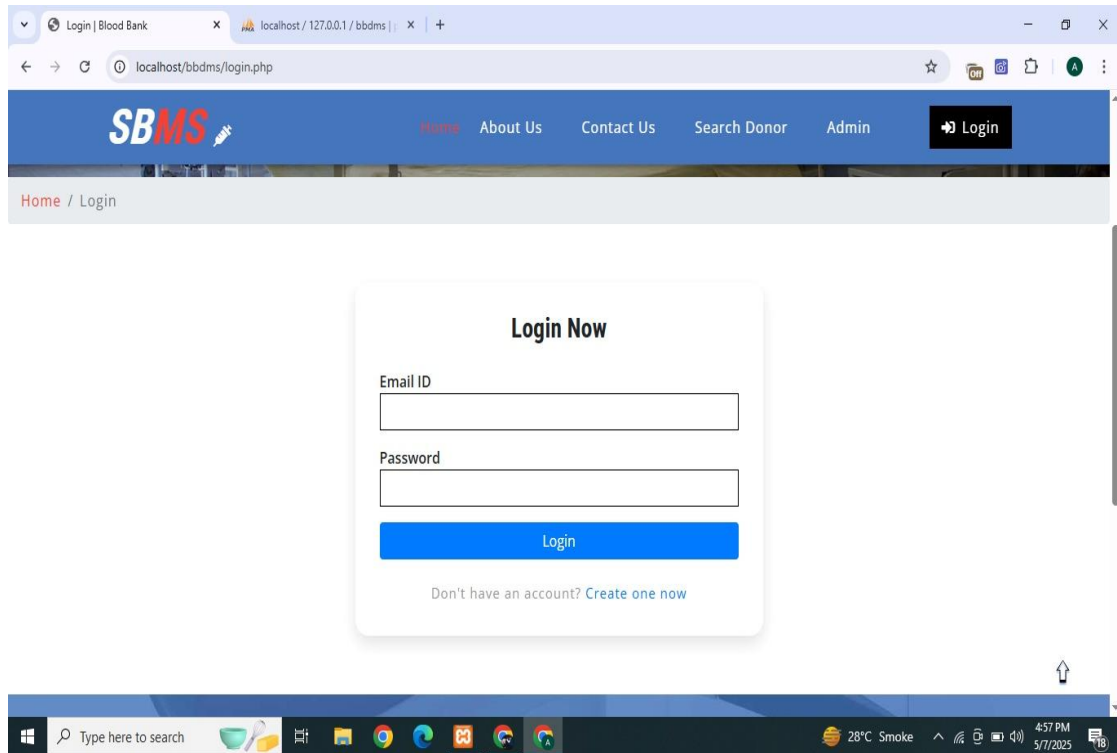


Figure 5.2: Login Page

5.2.3 Donor Dashboard

This personalized dashboard is designed to give you quick and easy access to all your donation-related activities. Track your past donations, check your eligibility for the next donation, receive notifications, and stay informed about blood requests in your area.

1. View personal profile and donation history
2. Receive notifications for blood donation requests
3. Check donation eligibility status
4. Update contact information
5. Logout button for session security

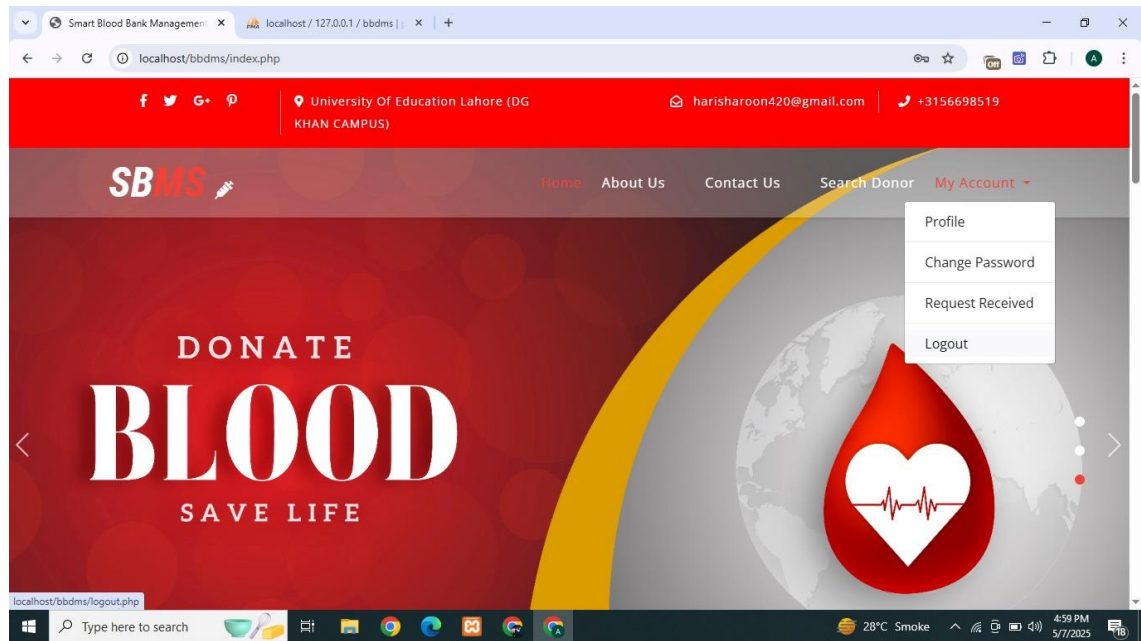


Figure 5.3: Donor Dashboard

5.2.4 Blood Bank Admin Dashboard

This dashboard provides administrators with full control over blood inventory, donor records, hospital requests, and donation scheduling. Monitor real-time data, manage stock levels efficiently, verify donation entries, and ensure timely responses to urgent blood needs. Streamline operations and support life-saving decisions from one centralized system.

1. Add/update blood inventory records
2. Approve or reject hospital blood requests
3. Monitor stock levels and expiry dates
4. View reports and analytics
5. Manage donor registration approvals

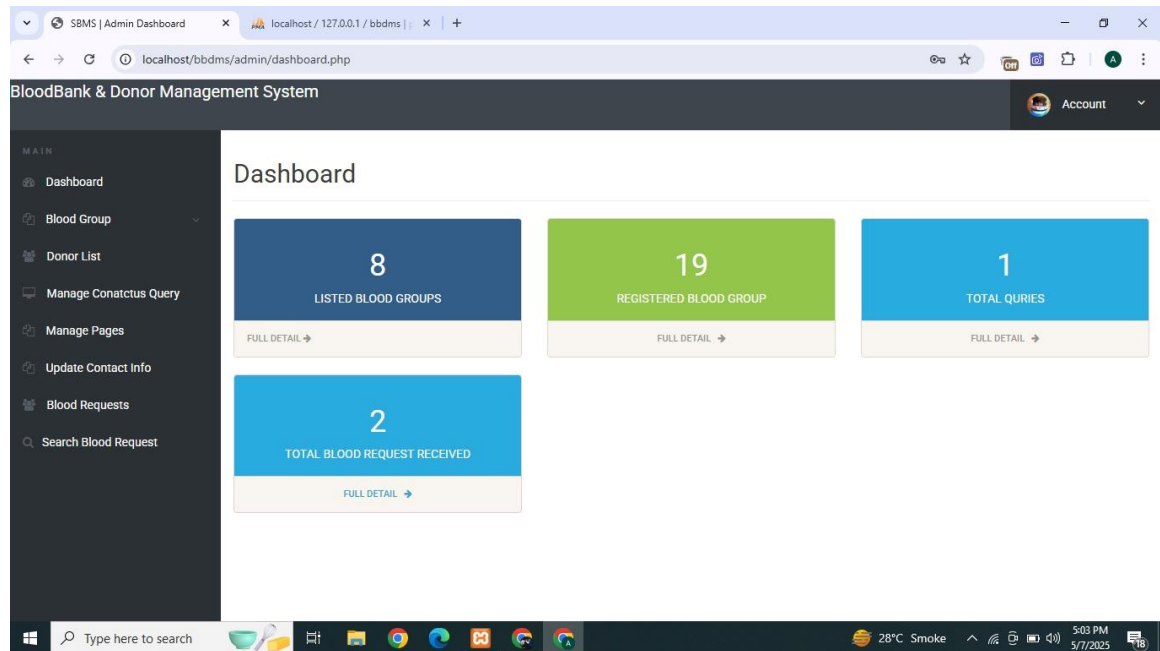


Figure 5.4: Blood Bank Admin Dashboard

5.2.5 System Admin Panel

This secure panel gives system administrators complete access to manage users, roles, permissions, and overall system settings. Monitor platform activity, handle technical configurations, and ensure smooth operation across all modules. Designed for efficiency, control, and full oversight of the system's backend operations.

1. Full system control: users, data, settings
2. Role and access management
3. System logs and usage tracking
4. Backup and recovery tools
5. Manage notifications and alerts

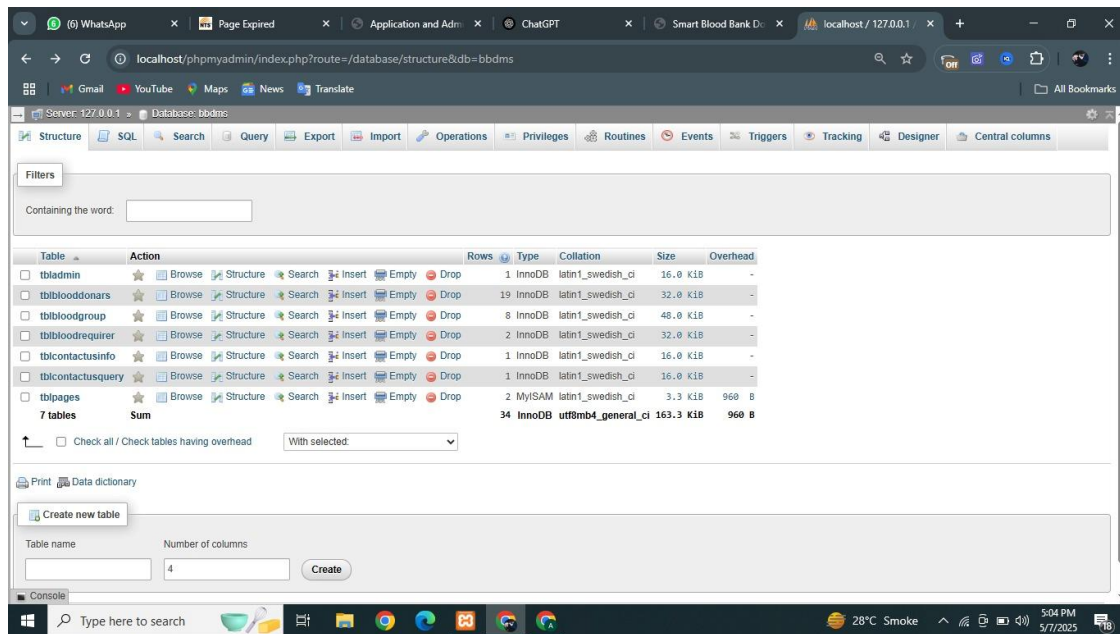


Figure 5.5: System Admin Panel

5.3 Navigation and User Flow

Each user role is given a specific flow, ensuring tasks can be completed with minimal clicks. Navigation menus are kept simple, with clearly labeled icons and tooltips. Buttons and links are placed strategically to guide users through the system's processes (e.g., "Donate Now", "Request Blood", "Update Stock").

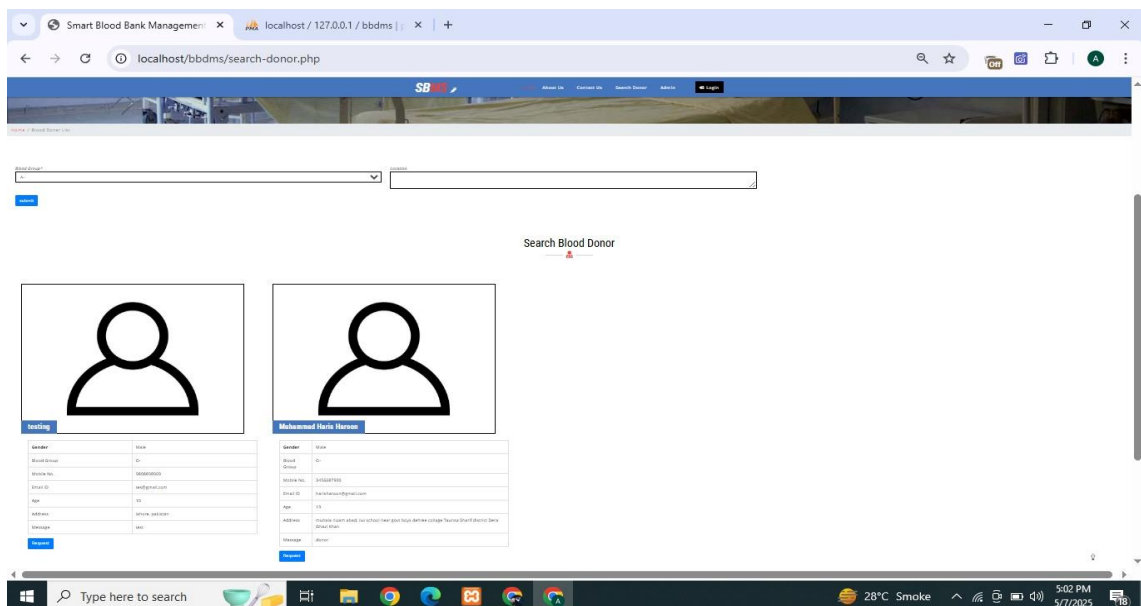


Figure 5.6: Navigation and User Flow

5.4 Interface Tools and Technologies

The GUIs are built using modern web technologies:

- **Frontend:** HTML5, CSS3, JavaScript, JQuery
- **Frameworks:** Bootstrap (for responsive design), possibly React or Vue.js for interactive components
- **Icons & Styling:** Font Awesome, Google Fonts
- **Backend Communication:** PHP with MYSQL for smooth data retrieval

CHAPTER 6

TESTING

6.1 Introduction

Testing is a crucial phase in the software development lifecycle that ensures the system functions as intended, meets user requirements, and is free from critical bugs or errors. The Smart Blood Bank System underwent rigorous testing to verify both its functionality and performance. Two primary testing methods were used: **Black Box Testing**, which evaluates the system based on inputs and expected outputs without looking into internal code structure, and **White Box Testing**, which examines the internal logic and code paths. This chapter outlines the test scenarios, test plan, defined test cases, and the outcomes of both black box and white box testing.

6.2 Test Scenario (Sample Attached)

A test scenario defines what to test, covering a broad functionality or feature. Below is a sample test scenario for the system:

Test Scenario ID	Scenario Description
TS001	Verify that a new donor can successfully register
TS002	Ensure login fails with incorrect credentials
TS003	Check if hospital can request a specific blood type
TS004	Validate admin can approve or reject blood requests

6.3 Test Plan

The test plan outlines the approach, resources, and schedule for the testing activities. It includes:

- **Objectives:** Ensure system accuracy, reliability, and performance

- **Scope:** All core modules—registration, login, blood inventory, request handling, notifications
- **Testing Types:** Black box, white box, regression, functional, and usability testing
- **Tools Used:** Manual testing, browser developer tools, SQL test queries
- **Responsibility:** QA testers and developers
- **Entry Criteria:** Functional modules developed and integrated
- **Exit Criteria:** All test cases passed with zero critical bugs

6.4 Definition of Test Cases

Test cases are defined sets of conditions and inputs to verify a particular function. Each test case includes the input, expected result, and actual result. Well-defined test cases help identify whether the system behaves as expected across different user actions.

6.5 Test Case Specifications

Test Case ID	Functionality	Input	Expected Output	Status
TC001	Donor Registration	Valid donor details	Registration successful, donor profile saved	Passed
TC002	Login	Wrong password	Error message: "Invalid credentials"	Passed
TC003	Blood Request	Request for A+ blood	Request submitted and visible to admin	Passed

6.6 Test Case Results

6.6.1 Black Box Test Cases

Black box testing was used to test the overall functionality without accessing internal code logic. It involved input validation, button actions, and user navigation.

1. **Login Functionality:** Correct and incorrect input tested → Passed

2. **Donor Registration:** Valid/invalid inputs tested → Passed
3. **Blood Search:** Results matched database values → Passed
4. **Request Handling:** Approval/rejection tested → Passed

6.6.2 White Box Test Cases

White box testing focused on the internal structure of the system. Developers tested individual functions, loops, and conditions.

1. **Database Query Validation:** Confirmed correct joins and conditions
2. **Code Path Testing:** All logical paths executed successfully
3. **Session Handling:** Checked for login/session expiry and redirect handling
4. **Form Validation Functions:** Field-level validation tested and secure
5. **Security Checks:** SQL injection and input sanitization tested

All white box test cases passed with expected outcomes and no critical issues were found.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The development of the Smart Blood Bank System marks a significant advancement in streamlining blood donation and transfusion processes. By integrating donors, hospitals, and blood bank administrators into a unified digital platform, the system enhances the efficiency, transparency, and responsiveness of blood supply management. Key functionalities such as real-time inventory tracking, donor registration, blood request handling, and automated notifications ensure that critical blood needs are met promptly. The system's user-friendly interfaces and robust backend infrastructure contribute to its reliability and ease of use. Overall, the Smart Blood Bank System addresses the challenges of traditional blood bank operations by leveraging technology to save lives and improve healthcare outcomes.

7.2 Future Work

While the current implementation provides a solid foundation, several enhancements can be pursued to further improve the system:

- **Integration with National Health Databases:** Connecting the system with national health records can facilitate better tracking of donor eligibility and transfusion histories.
- **Mobile Application Development:** Creating dedicated mobile apps for Android and iOS platforms can increase accessibility and user engagement.
- **Advanced Analytics and Reporting:** Implementing data analytics tools can provide insights into donation patterns, demand forecasting, and inventory optimization.
- **Multi-Language Support:** Adding support for multiple languages can cater to a diverse user base and improve usability across different regions.

- **Enhanced Security Measures:** Incorporating advanced security protocols, such as two-factor authentication and data encryption, can safeguard sensitive user information.

References

- American Psychological Association. (2010). *Publication manual of the American Psychological Association* (6th ed.). Washington, DC: Author.
- Streefkerk, R. (2023, September 4). APA Format (6th ed.) for Academic Papers and Essays [Template]. *Scribbr*. Retrieved April 16, 2025, from <https://www.scribbr.com/apa-style/6th-edition/archived-format/>
- Ahmed, A., & Kumar, R. (2018). *Design and implementation of an automated blood bank system using web-based technologies*. *International Journal of Computer Applications*, 179(18), 10–15.
- Kumar, S., & Sharma, M. (2020). *A smart blood bank system using cloud computing and IoT*. *Journal of Emerging Technologies and Innovative Research*, 7(6), 112–117.
- Red Cross. (n.d.). *Donate blood, platelets or plasma*. American Red Cross. Retrieved April 16, 2025, from <https://www.redcrossblood.org>
- Ministry of Health and Family Welfare. (n.d.). *e-Rakt Kosh – A centralized blood bank management system*. National Health Portal of India. Retrieved April 16, 2025, from <https://www.eraktkosh.in>
- GitHub. (n.d.). *Blood bank management system projects*. GitHub. Retrieved April 16, 2025, from <https://github.com/search?q=blood+bank+management+system>

Appendix

Appendix A: Donor Registration Form

Include a screenshot or detailed description of the donor registration form used in the system.

Appendix B: Blood Request Workflow Diagram

Provide a flowchart illustrating the process of requesting blood, from submission to fulfillment.

Appendix C: Sample Notification Messages

List examples of automated notifications sent to donors and hospital staff.

Formatting Guidelines for Appendices (APA 6th Edition):

- Begin each appendix on a new page.
- Label each appendix with a letter (e.g., Appendix A, Appendix B) centered and bolded at the top of the page.
- On the next line, provide a descriptive title, also centered and bolded.
- The content should be left-aligned, double-spaced, and follow standard APA formatting.
- Refer to each appendix in the main text by its label (e.g., see Appendix A).

For detailed formatting instructions, refer to the APA Style guidelines on appendices:
<https://apastyle.apa.org/style-grammar-guidelines/paper-format/appendices>