RK.UNIVERSITY

BLOOD BANK MANAGEMENT SYSTEM

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

B.TECH. (COMPUTER ENGINEERING)

TO

RK UNIVERSITY, RAJKOT

SUBMITTED BY

Name of Student	Enrollment No.
Harendra Chaudhary	21SOECE11607
Pradip Tharu	21SOECE11643

UNDER THE GUIDANCE OF

Internal Guide
Prof. Alpana Kumari
Assistant Professor,
RK University
Rajkot

September 2024

SCHOOL OF ENGINEERING
RK.UNIVERSITY

SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

RK.UNIVERSITY

CERTIFICATE

This is to certify that the work which is being presented in the Project Report entitled **“Blood Bank Management System”,** in partial fulfillment of the requirement for the award of the degree of **B.Tech. (Computer Engineering)** and submitted to the School of Engineering, RK University, is an authentic record of my/our own work carried out during a period from **June 2024 to November 2024.**

The matter presented in this Project Report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Student (S)

Harendra Chaudhary(21SOECE11607)

Pradip Tharu(21SOECE11643)

This is to certify that the above statement made by the student(s) is correct to the best of my knowledge.

Internal Guide
Prof. Alpana Kumari
Assistant Professor
CE/IT Department
RK University, Rajkot

Head of Department
Dr. Chetan Shingadiya
CE/IT Department
School Of Engineering
RK University, Rajkot

SCHOOL OF ENGINEERING
RK.UNIVERSITY

SCHOOL OF ENGINEERING, RK UNIVERSITY, RAJKOT

DECLARATION

I/We hereby certify that I/We am/are the sole author(s) of this project work and that neither any part of this project work nor the whole of the project work has been submitted for a degree to any other University or Institution. I/We certify that, to the best of my/our knowledge, my/our project work does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my/our project document, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I/We declare that this is a true copy of my/our project work, including any final revisions, as approved by my/our project review committee.

Signature of Student (S)

Harendra Chaudhary(21SOECE11607),
Date: 30/09/2024
Place: Rajkot, Gujarat

Pradip Tharu(21SOECE11643)
Date: 30/09/2024
Place: Rajkot, Gujarat

Acknowledgement

We would like to express our sincere gratitude to all those who have contributed to the completion of this project and the successful culmination of my final year in computer engineering. First and foremost, we extend our heartfelt thanks to **Dr. Chetan Shingadiya**, Head of the Computer Engineering Department at RK University, for providing the necessary facilities and unwavering support throughout this journey.

We are deeply grateful to our project guide, **Prof. Alpana Kumari**, mentorship, and expertise have been invaluable in shaping this project and our understanding of the subject matter.

Lastly, we extend our thanks to all those who have directly or indirectly contributed to this endeavor. Your support has been instrumental in the successful completion of this project.

Thank you all for your contributions and support.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Project Summary: The Blood Bank Management System (BBMS) is a web-based application designed to streamline the processes of blood donation, storage, and distribution. By enhancing the efficiency and accessibility of blood bank operations, BBMS aims to improve the management of blood resources and ultimately save lives.

1.2 Purpose: The primary goal of Blood Bank Management System(BBMS) is to provide a comprehensive platform that connects donors, blood banks, and recipients. Specific Goals and objectives include:

- Facilitating online registration for blood donors.
- Managing inventory and tracking blood stocks.
- Automating the scheduling of donation drives and appointments.

1.3 Scope:

Scope of Blood Bank Management System:

What the system Can Do:

- Allow users to register as donors.
- Enable blood banks to manage donor information and blood inventory.
- Provide an interface for hospitals to request blood.
- Generate reports on blood donation trends and inventory status.

What it Can't Do:

- Directly conduct medical screenings for donors.
- Handle financial transactions related to blood donations.
- Operate without an internet connection.

1.4 Technology and Literature

Review of Past Work/System

The Blood Bank Management System (BBMS) is built using **ASP.NET Core MVC**, a robust and scalable framework ideal for developing modern web applications.
It allows for the creation of dynamic and responsive user interfaces, which are crucial for managing the diverse needs of blood bank operations.

Technologies Used:

- Frontend Development: The frontend is developed using HTML, CSS, and
- JavaScript to create a user-friendly and responsive interface for seamless user
- interaction.
- Backend Framework: ASP.NET, a widely used web application framework, serves
- as the backbone of **Blood Bank System**, providing the necessary tools for server-side
- Programming Language: C# is the primary programming language for the backend
- development, ensuring efficient code execution and compatibility with ASP.NET.
- Database: MySQL, a powerful open-source relational database management system,
- is employed to store user data, appointment information, and profiles.

Literature Review of Past Work/System:

- Digital Solutions: Automation in blood banking addresses inefficiencies in manual processes, leading to improved data accuracy and operational efficiency.
- User-Centric Design: Systems that incorporate user feedback have higher adoption rates.
- Advanced Technology Integration: Technologies like cloud computing enable real-time data access and better collaboration among healthcare providers, noted in the Journal of Health Informatics.
- Data Security: Protecting sensitive health data is essential. The Journal of Healthcare Information Management emphasizes the need for encryption and adherence to regulations like HIPAA.
- Case Studies: Successful automated systems have shown significant efficiency gains, with one case study in the Transfusion Medicine Reviews reporting a 30% increase in operational efficiency.

2.0 PROJECT MANAGEMENT

2.1 Project Planning and scheduling

2.1.1 Project Development Approach:

The project development approach for the Blood Bank Management System website is based on an Agile methodology. Agile is chosen for its iterative and collaborative nature, which is particularly well-suited to dynamic web development projects. This approach allows for flexibility in responding to changing requirements and continuous userfeedback, ensuring that the final product meets the evolving needs of students and Company.

Justification:

- **Iterative Development:** Agile allows for incremental development and frequent releases, enabling early userengagement and feedback.
- **User-Centric:** The nature of the Blood Bank Management System website, which involves user interactions and preferences, benefitsfrom a process that emphasizes user involvement throughout development.
- **Adaptability:** In the fast-paced world of web development, Agile provides the flexibility to adapt to changing technologiesand user demands.

2.1.2 Project Plan:

Milestones:

- Project Initiation: Define project scope, objectives, and keystakeholders.
- System Design: Develop the system architecture, database structure, and user interface wireframes.
- Backend Development: Implement the backend logic and database integration.
- Frontend Development: Create the user interface and frontendfunctionalities.
- User Testing: Conduct usability tests and gather user feedback.
- Integration and Testing: Ensure seamless communication between frontend and backend components.
- Security Implementation: Enhance data security measures.
- Deployment: Deploy the website on a production server.

Deliverables:

- Functional website with appointment scheduling, user profiles, and tracking features.

- User documentation and support resources.
- Testing reports.
- Deployment and server setup documentation.

Roles and Responsibilities:

- Project Manager: Oversee the entire project, manage timelines, and coordinate team efforts.
- Backend Developer: Responsible for server-side logic, database integration, and email functionality.
- Frontend Developer: Develop the user interface and client-sidefunctionalities.
- UI/UX Designer: Design user-friendly interfaces andwireframes.
- Security Specialist: Implement security measures and dataprotection protocols.
- Technical Writer: Prepare user documentation and trainingmaterials.

Dependencies:

- Frontend development depends on system design andwireframes.
- Email integration is dependent on both backend and frontenddevelopment.
- User testing depends on the completion of frontend and backend development phases.
- Security implementation is intertwined with the developmentphases.

2.1.3 Schedule Representation: A Gantt chart or similar scheduling tool can be utilized to represent the timeline of the project phases and milestones.

2.2 Risk Management

2.2.1 Risk Identification:

- **Data Security Breach:** The potential for unauthorized access to user data, including personal information and appointment details, poses a significantrisk. This could compromise user privacy and trust.
- **Technical Compatibility Issues:** Incompatibility with certain browsers or devices could lead to a suboptimal user experience, limiting the website's accessibility.
- **User Adoption:** Ensuring that both students and Company actively use the platform may be a challenge. If the user adoption rate is low, the project's success could be impacted.
- **Email Notifications:** Dependence on email notifications may lead to issuessuch as emails going to spam folders or not being delivered at all, affecting the reliability of the appointment confirmation process.
- **Data Loss:** Inadequate data backup and recovery mechanisms may result indata loss in the event of server failures or other technical issues.
- **User Feedback Handling:** Managing and implementing user feedback effectively may prove challenging, potentially leading to the prioritization of less critical features or changes.

2.2.2 Risk Analysis:

- **Data Security Risks:** High probability; serious impact on reputation and legal compliance.
- **Technical Risks:** Medium probability; moderate impact on project timelines.
- **Regulatory Risks:** Low probability; serious impact if regulations are not followed.

2.2.3 Risk Planning:

- Data Security Breach: Plan includes robust encryption, regular security audits,and legal compliance with data protection regulations. A dedicated security team will be responsible for monitoring and responding to security threats.
- Technical Compatibility Issues: Regular testing on various platforms and browsers will be conducted during development. Compatibility issues will be addressed promptly with updates and patches.
- for early adopters, and targeted marketing campaigns.

3.0 SYSTEM REQUIREMENTS STUDY

3.1 User Characteristics:

The Blood Bank Management System website caters to three primary types of users within the context ofan educational institution:

- **Donors:** Individuals willing to donate blood, seeking an easy registration process.
- **Blood Bank Staff:** Responsible for managing blood inventory and donor information.
- **Hospitals:** Users needing access to blood availability and requesting blood supplies.

3.2 Hardware and Software Requirements:

Minimum Hardware Requirements:

- A computer, laptop, or mobile device with internet connectivity.
- Modern web browsers (e.g., Chrome, Firefox, Safari) that support HTML5 and JavaScript.

Minimum Software Requirements:

- Operating System: Windows, macOS, Linux, or mobile OS (for mobileaccess).
- Web server (e.g., IIS, Apache) for hosting the ASP.NET-based application.
- MySQL database management system.
- Email server for sending appointment notifications.
- Necessary server-side dependencies for ASP.NET and C#.

3.3 Constraints:

- Regulatory Policies: Must comply with health and safety regulations regarding blood donation and handling.
- Hardware Limitations: Requires sufficient server capacity to handle user load.
- Interfaces to Other Applications: May need to integrate with hospital management systems.

4 SYSTEM ANALYSIS

4.0 Study of Current System

The current blood bank management systems predominantly rely on manual processes for donor registration, inventory tracking, and blood distribution. This often leads to inefficiencies, errors, and difficulties in real-time data access. Existing systems lack integrated interfaces, resulting in disconnected operations between blood banks and hospitals.

4.1 Problem and Weaknesses of Current System

- Inefficiency: Manual record-keeping is time-consuming and prone to errors.
- Limited Accessibility: Information is not readily available to stakeholders, leading to delays in blood requests and management.
- Poor Inventory Management: Difficulty in tracking blood stocks can lead to shortages or wastage.
- Lack of Real-Time Updates: Hospitals may not receive timely information about blood availability.
- Data Security Concerns: Existing systems may not comply with stringent health regulations regarding data privacy.

4.2 Requirements of New System (Proposed System)

4.2.1 Functional Requirements:

- User Registration: Allow donors to register online.
- Inventory Management: Track blood stocks and manage donor information.
- Blood Request System: Enable hospitals to request blood and view availability.
- Automated Notifications: Send updates to users regarding blood drives and availability.
- Reporting: Generate reports on donation trends and inventory status.

4.2.2 Non-Functional Requirements:

- Security: Implement robust data encryption, access controls, and regular security audits.
- Performance: Ensure fast response times, especially during peakusage.
- Usability: Interface must be intuitive and user-friendly for diverse users.
- Reliability: High availability, especially during peak donation

drives.

- Compliance: Adhere to health regulations and standards.
- Cost-Effectiveness: Develop and maintain the system within budget constraints.

4.3 Feasibility Study

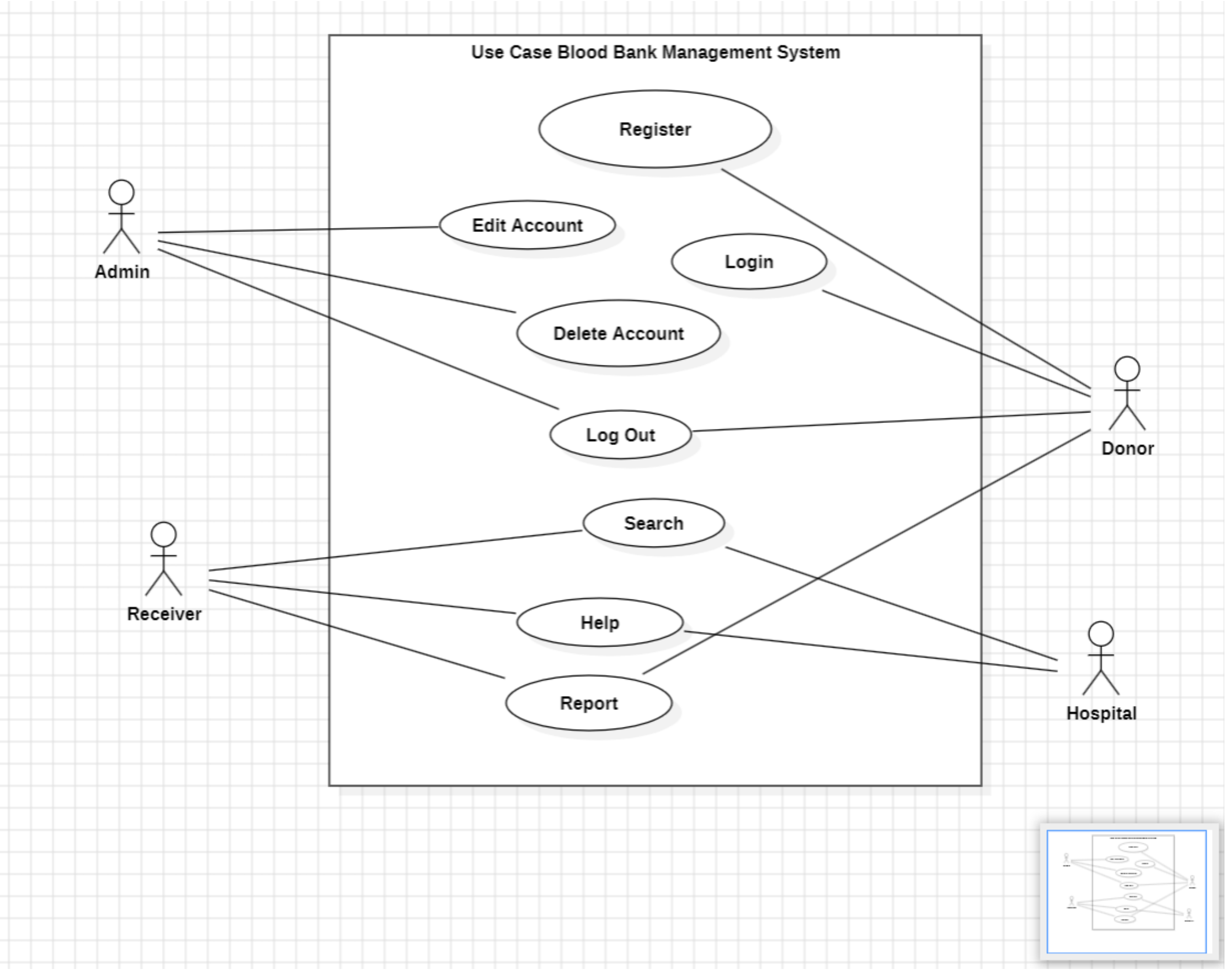
- Contribution to Objectives: The system directly supports the goal of improving blood resource management and operational efficiency.
- Technology Implementation: The proposed technologies (Node.js, MongoDB, etc.) are currently available and suitable for development within the expected budget and timeline.
- Integration Potential: The system can be designed to interface with existing hospital management systems, facilitating better collaboration.

4.4 Requirements Validation:

To ensure the requirements accurately define the system desired by stakeholders, feedback sessions with potential users (donors, blood bank staff, and hospitals) will be conducted. Prototypes will be developed to gather input and confirm that the proposed functionalities meet user needs.

4.5 Functions of System

4.6.1 **Use Cases**, event trace or scenario ***(Use Case Diagram)***



4.7.1 Data Dictionary

Donor: ID, Name, Contact Information, Blood Type, Donation History.

Blood Inventory: ID, Blood Type, Quantity, Expiry Date.

Hospital: ID, Name, Contact Information, Blood Requests.

1.Donor

The entity is Donor. The attributes and their data types are as follows:

- **Unique identifier:** This is an integer that uniquely identifies the donor**.**
- **Name:** This is a string that represents the name of the donor.
- **Age:** This is an integer that indicates the age of the donor.
- **Blood type:** This is a string that specifies the blood type of the donor.
- **Date of donation:** This is a string that shows the date when the donation occurred.

Blood Inventory

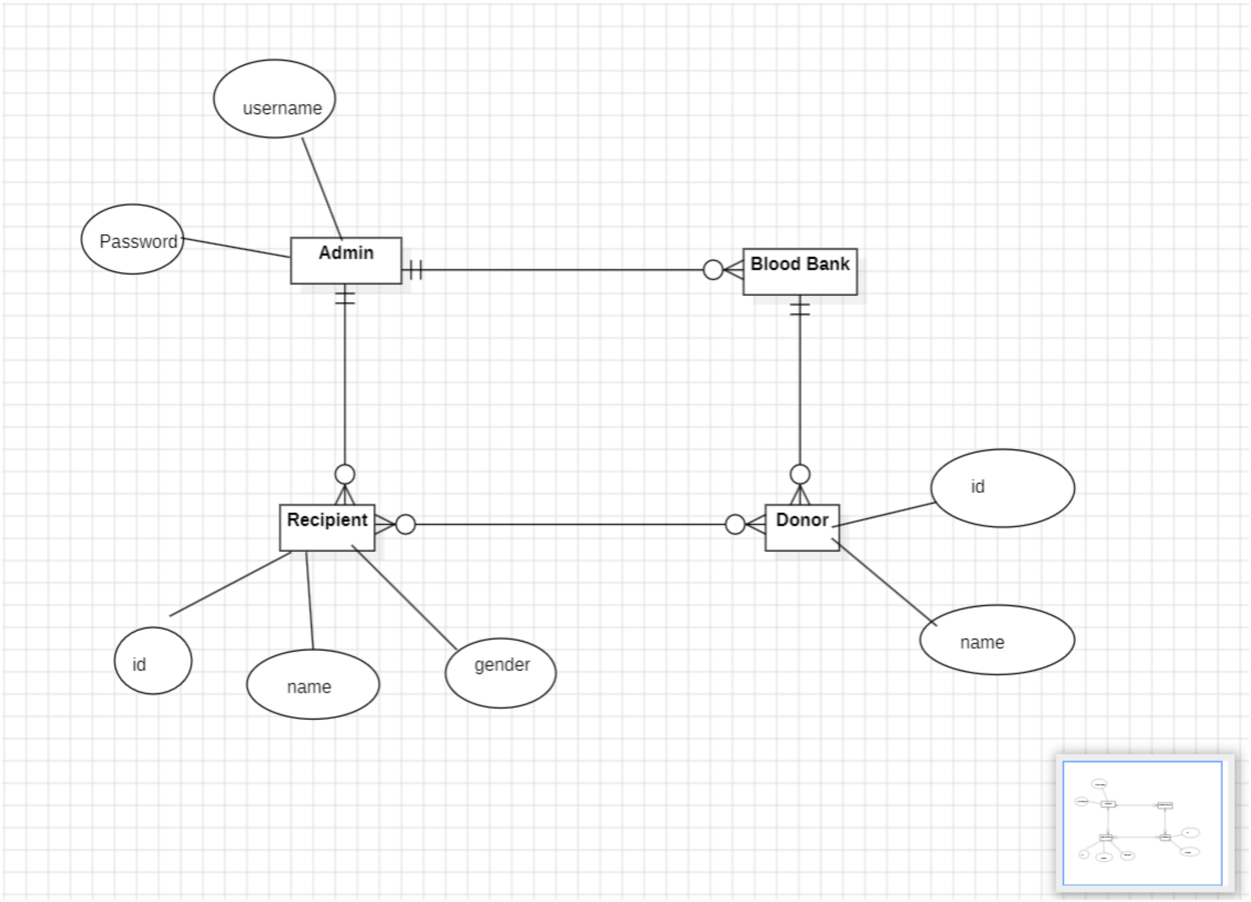
- **BloodID**: Unique identifier for each blood unit (Integer)
- **BloodType**: Type of blood (String)

- **Quantity**: Amount of blood available (Integer)
- **ExpiryDate**: Date when the blood unit expires (Date)
- **DonationDate**: Date when the blood was donated (Date)

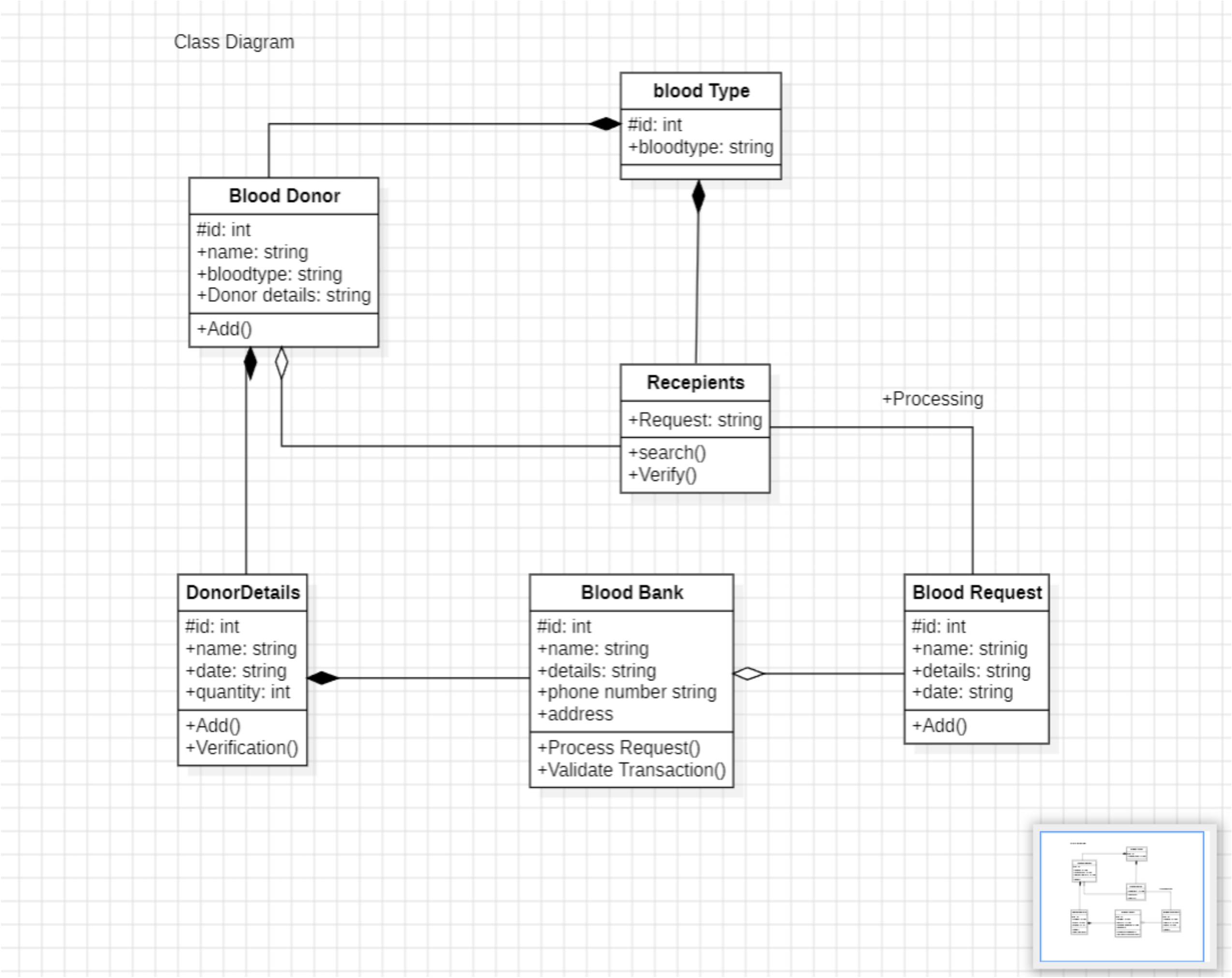
Hospital

- **HospitalID**: Unique identifier for each hospital (Integer)
- **Name**: Name of the hospital (String)
- **ContactInfo**: Contact details of the hospital (String)
- **Address**: Physical address of the hospital (String)

4.7.2 ER diagrams



4.7.3 Class Diagram



4.7.4 System Activity

The system activity for the Blood Bank Management System website encompasses a dynamic view ofthe system's functionalities and how it responds to various user interactions. This section outlines key system activities and processes:

User Registration and Authentication:

- Users can register by providing necessary information.
- The system verifies user credentials during login.
- Failed login attempts trigger security measures.

Appointment Management:

- Users schedule appointments by selecting Company and preferred time slots.
- The system checks Company availability and updates the appointment status.
- Users can modify or cancel appointments.
- Notifications are sent for appointment confirmations, changes, and cancellations.

Profile Management:

- Users update their profiles, including personal details and preferences.
- Changes are reflected in the user's profile.

Communication:

- Users send messages to Company or other users.
- Notifications are sent for incoming messages.
- Users can view and reply to messages within the system.

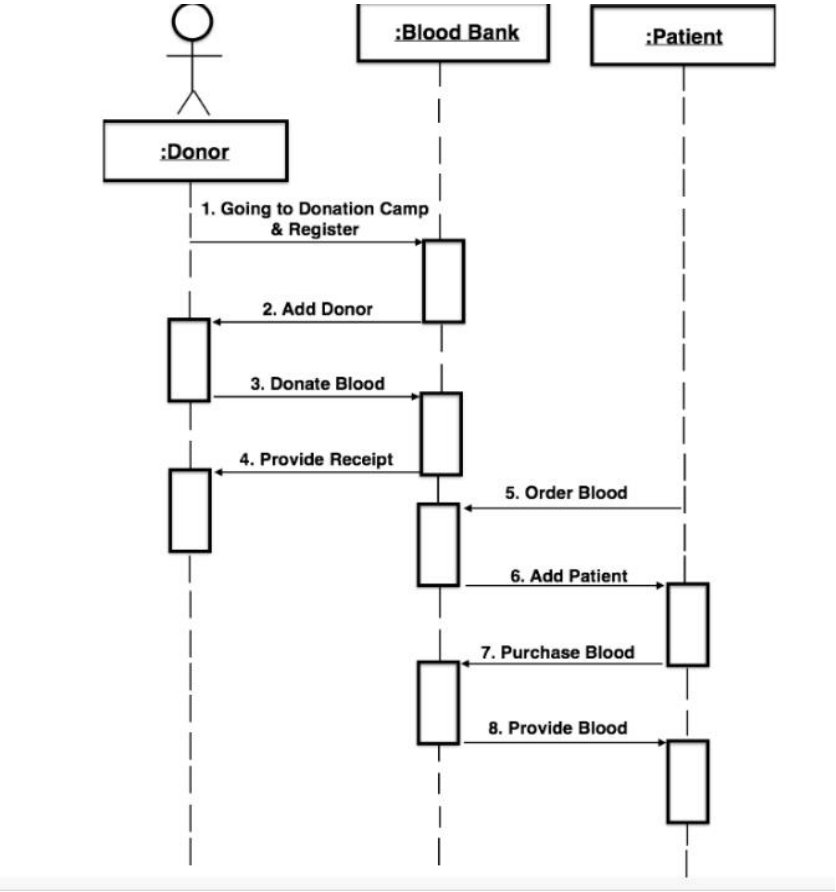
Security Measures:

- The system implements security protocols to protect user data.
- Encryption is used for sensitive information.
- Access controls and user authentication are in place.

Notification Delivery:

- The system ensures timely delivery of notifications via email.
- Email server integration is used to send notifications.

4.7.5 Object interaction Diagram

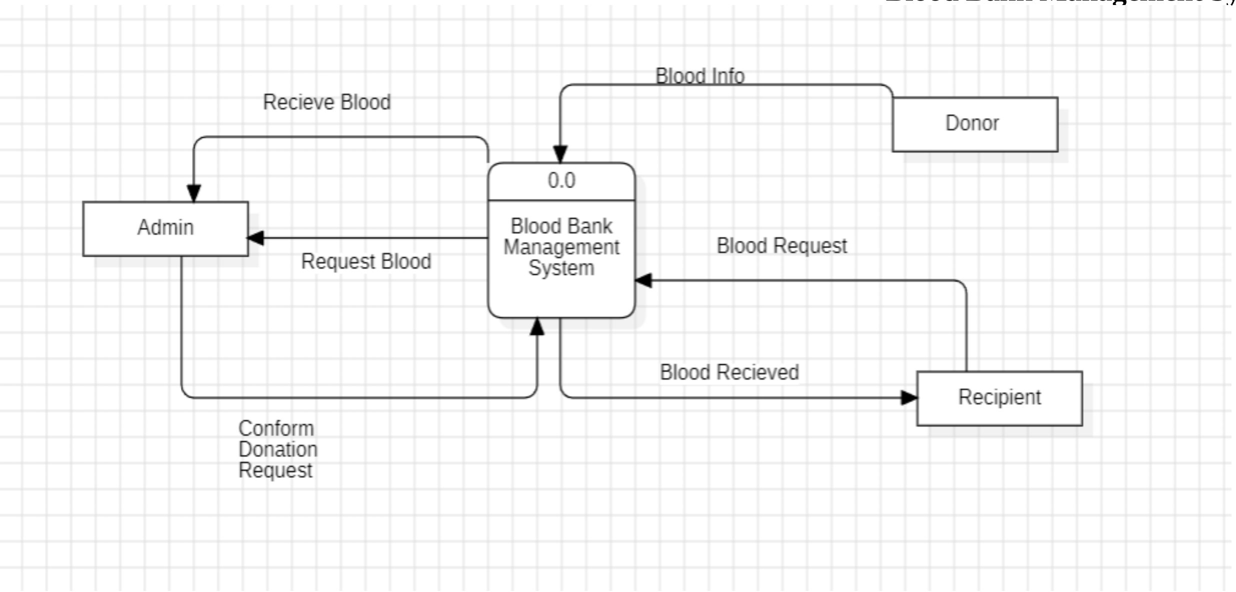


4.6 Functional and Behavioral Modeling

4.8.1 Data Flow Diagram (0 and 1 level)

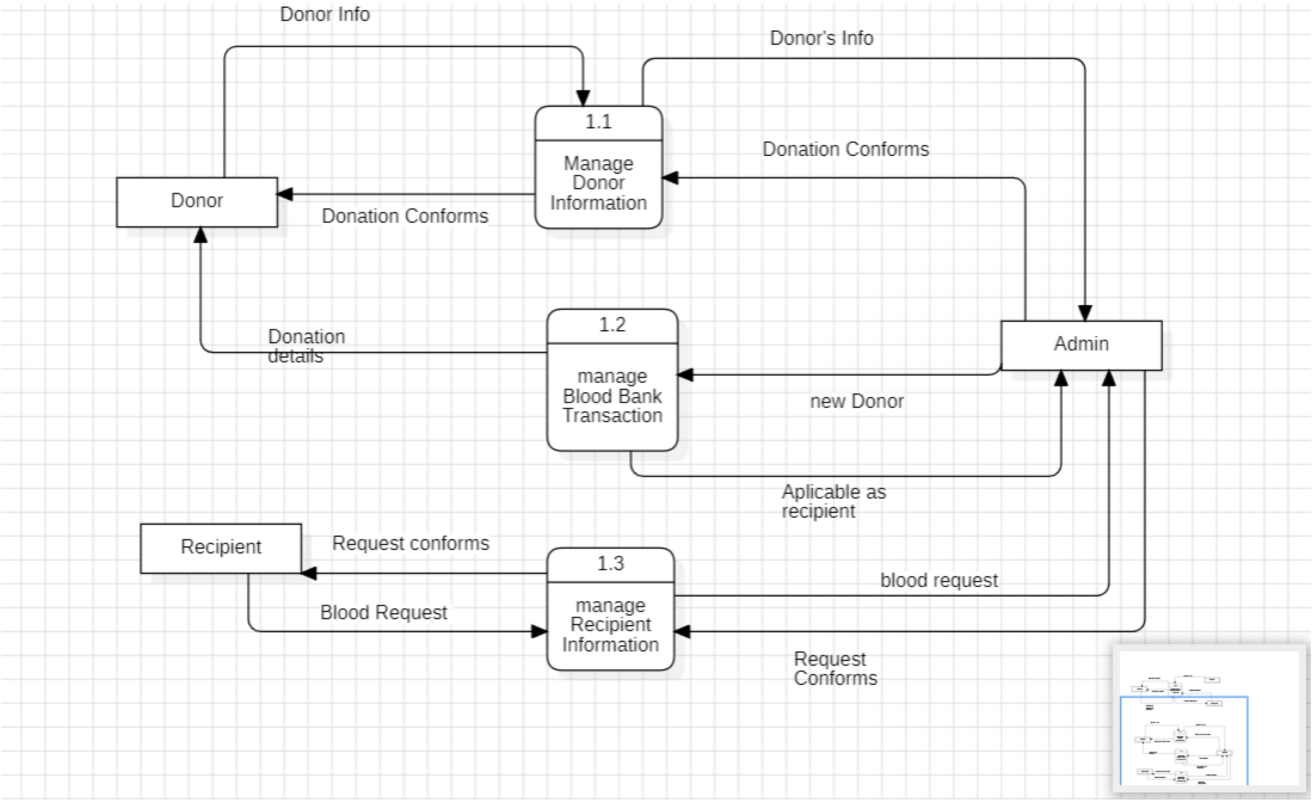
0- Level DFD (Context Diagram):

The context diagram provides an overview of the entire system, showingexternal entities, processes, data stores, and data flows.



0-Level DFD (Appointment Scheduling Process):

This diagram dives deeper into the appointment scheduling process within theBlood Bank Management System.



4.8.2 Process Specification and Decision Table

This process allows students to request appointments with Company and allows Company to accept or reject appointment requests.

Register Donor

- Input: Donor Information (Name, Age, Blood Type, Contact Info)
- Process: Validate information, Store in database
- Output: Confirmation of registration

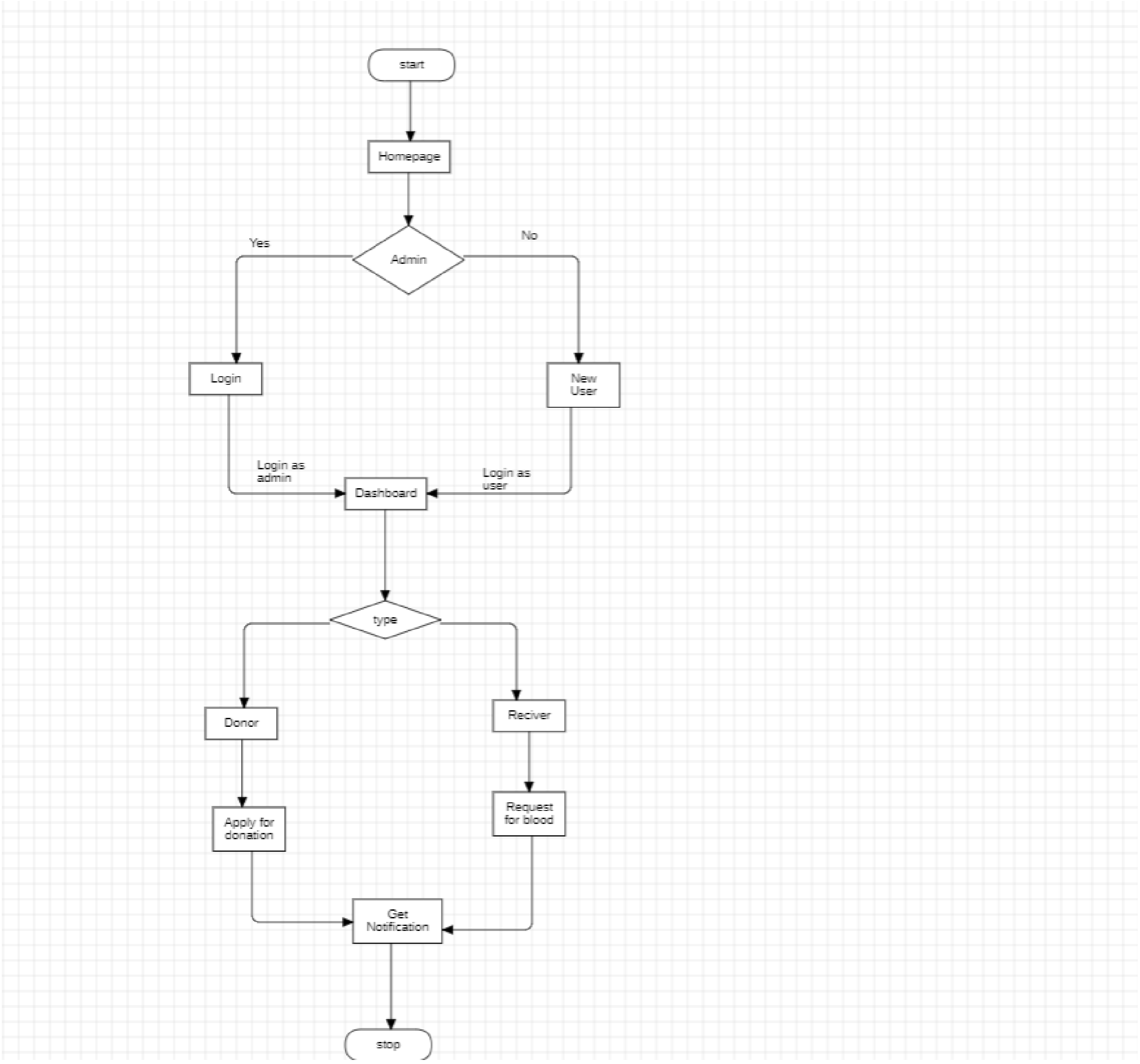
Process Blood Request

- Input: Blood Request Details (Blood Type, Quantity)
- Decision:
    - If Blood is Available → Fulfill Request
    - Else → Notify Hospital of Unavailability
- Output: Fulfillment Status

Decision Table:

- **Condition:** Request Valid, Blood Available: Yes, Hospital Registered: Yes → Action: Fulfill Request
- **Condition:** Request Valid, Blood Available: No, Hospital Registered: Yes → Action: Notify Hospital Unavailable
- **Condition:** Request Invalid, Blood Available: Any, Hospital Registered: Yes → Action: Reject Request
- **Condition:** No Request, Blood Available: Any, Hospital Registered: Any → Action: No Action

4.8.3 Control flow diagram



4.7 Main Modules of New System

- User Management: Handles donor registration and authentication.
- Inventory Management: Manages blood stocks, including tracking and reporting.
- Request Management: Processes blood requests from hospitals.
- Notification System: Sends alerts and updates to users.
- Reporting Module: Generates reports on inventory and donation trends.

4.8 Selection of Hardware and Software and Justification

The selection of hardware and software for the Blood Bank Management System website is a critical decision that influences system performance, scalability, and reliability. Here arethe hardware and software choices, along with justifications:

Hardware Selection:

- Web Servers: High-performance web servers are chosen to ensure fast response times, even during peak usage. Load balancers are employed to distribute traffic efficiently.
- Database Server: A robust database server, such as MySQL, is utilized to store and manage user data and appointment records. It provides data integrity and scalability.
- Security Hardware: Firewalls, intrusion detection systems, and data encryption hardware are implemented to safeguard user data and maintain the security of the system.

Software Selection:

- Programming Languages and Frameworks: ASP.NET and C# are chosen for web development due to their robustness, security features, and compatibility with the Windows environment.
- Database Management System: MySQL is selected as the database management system for its reliability, scalability, and compatibility with ASP.NET.
- Web Development Tools: Visual Studio and Visual Studio Code are used for web application development, providing a powerful development environment.
- Content Delivery Network (CDN): CDNs are utilized to optimize the delivery of static content, enhancing website performance.

5.0 SYSTEM DESIGN

5.1 Database Design/Data Structure Design

5.1.1 Mapping objects/classes to tables (if non-OO languages)

Though ASP.NET MVC supports Object-Oriented Programming, database tables are mapped to objects in the code using Entity Framework or ADO.NET. Each class represents a table in the database, and properties of the class correspond to columns in the table.

5.1.1 Mapping objects/classes to tables (if non-OO languages)

Though ASP.NET MVC supports Object-Oriented Programming, database tables are mapped to objects in the code using Entity Framework or ADO.NET. Each class represents a table in the database, and properties of the class correspond to columns in the table.

5.1.2 Tables and Relationship

The tables and relationships represent how data is organized. In the Blood Bank system, there will be several tables like:

- Donors: Stores donor information.
- Blood-Stock: Maintains current blood availability.
- Hospitals: Stores details of hospitals requesting blood.
- Requests: Keeps track of blood requests.
- The relationships could include:

A one-to-many relationship between Donors and Blood-Stock (one donor can donate multiple blood units).

A many-to-one relationship between Requests and Blood-Stock (multiple requests can be made for the same blood type).

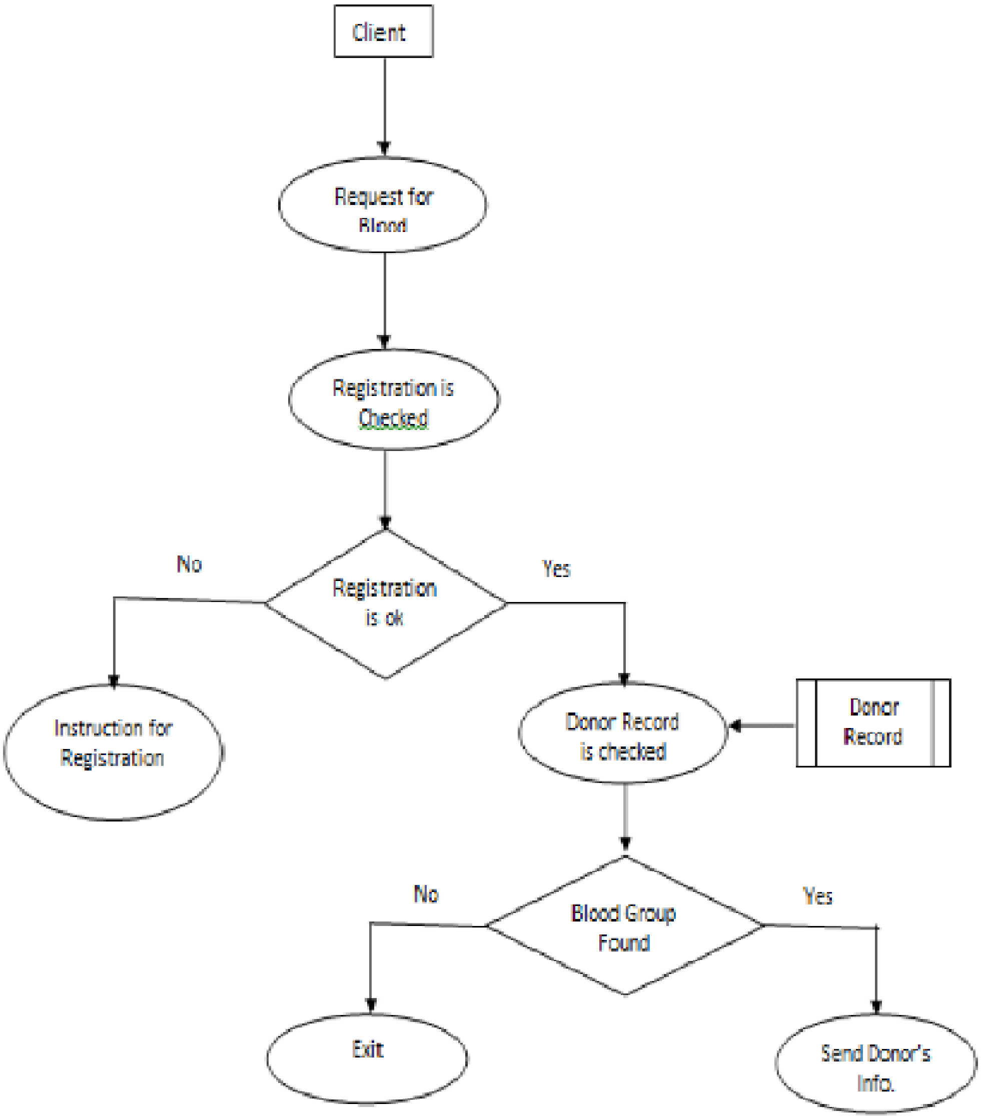
5.1.3 Logical Description OF Data

Each entity in the database has a logical role:

- Donor: A person donating blood, linked to a specific blood type.
- Blood-Stock: Represents the available blood inventory.
- Requests: Track requests from hospitals or patients for a specific blood type.

5.2 System Procedural Design

5.2.1 Flow chart or activity design



5.3 Input/Output and Interface Design

5.3.1 Samples of Forms and Interface

Login Page:

Provides a secure login interface for existing users to access theiraccounts.

User Registration Form:

Allows new users to register them.

5.3.2 Access Control and Security

User Roles and Permissions:

- Admin: Full access to all features (manage donors, blood stock, requests).
- Donors: Can view personal data and donation history only.

Authentication and Authorization:

- ASP.NET Identity handles user login and role-based access.
- HTTPS secures all communication, and sensitive data (passwords) is encrypted.

Security:

- Client- and Server-Side Validation ensures form inputs are correct and safe from injection attacks.
- Audit Logging tracks critical actions for security audits and traceability.

5.4 System Architecture Design

(Transformation of DFD into structural chart/Hierarchical Charts which shows control hierarchy of modules or sub-systems).

6.0 IMPLEMENTATION PLANNING AND DETAILS

6.1 Implementation Environment (Single vs Multiuser)

- **Single vs Multiuser:**

  The Blood Bank Management System will be a multi-user application where different users (Admin, Hospital, Donor) can log in simultaneously and perform their respective tasks.

- **GUI vs Non-GUI:**

  The system will have a Graphical User Interface (GUI). It will be user-friendly, intuitive, and responsive, built using ASP.NET MVC along with front-end technologies like Bootstrap, JavaScript, and jQuery.

6.2 Program/Modules Specification

The system will consist of the following major modules:

1. **Admin Module:**

- Manage users (Hospitals, Donors).
- View and manage blood stock.
- Manage blood donation drives.

2. **Donor Module:**

- Register and log in as a blood donor.
- View blood donation history.
- Schedule blood donation.

3.Blood Stock Management Module:

- View current stock of different blood types.
- Add or update blood units when blood is donated or used.

6.3 Security Features

The system will have the following security features:

- Authentication and Authorization: Implemented via ASP.NET Identity, ensuring that only authenticated users can access specific modules based on roles (Admin, Donor, Hospital).
- Data Encryption: Sensitive information (passwords, donor details) will be encrypted using ASP.NET Core Identity’s hashing mechanisms.
- Role-Based Access Control (RBAC): Users will have different access rights

depending on their role (Admin, Hospital, Donor).

- SQL Injection Protection: Use of parameterized queries and Entity Framework to prevent SQL injection.
- Input Validation: Server-side validation to avoid malicious input (e.g., cross-site scripting).

6.4 Coding Standards

- Naming Conventions
- Classes will follow PascalCase (e.g., BloodBankController, DonorService).
- Variables and method names will use camelCase (e.g., bloodType, getDonorDetails()).
- Code Comments: Proper comments will be provided for clarity and maintainability.
- Error Handling: Proper try-catch blocks for exception handling and logging errors.
- Separation of Concerns: Following MVC architecture principles, keeping Models, Views, and Controllers well-separated.
- Reusable Code: Utility classes and functions will be used for commonly required operations (e.g., validation, logging)

6.5 Sample Coding

7.0 TESTING

7.1 Testing Plan

The testing plan involves multiple stages of testing:

- Unit Testing: Each individual component of the system (e.g., models, controllers, and services) will be tested to ensure they work correctly in isolation.
- Integration Testing: Ensures that different modules (such as Donor Registration, Blood Request, and Blood Stock Management) interact with each other smoothly.
- System Testing: The entire system will be tested as a whole to ensure that it behaves correctly under real-world conditions.
- User Acceptance Testing (UAT): Final testing performed by actual users (Admin, Hospitals, and Donors) to confirm that the system meets the functional requirements.
- Performance Testing: This will evaluate how the system performs under load, ensuring that it can handle multiple users accessing the system simultaneously.

7.2 Testing Strategy

The testing strategy focuses on ensuring both functional and non-functional aspects of the system are validated:

- Manual Testing: All major functionalities (such as user registration, blood donation scheduling, and blood stock updates) will be manually tested to ensure the user interface and user experience work as intended.
- Automated Testing: Unit tests will be automated using tools like xUnit to verify the logic and flow of individual modules (e.g., blood stock management, user authentication).
- Black Box Testing: Focuses on testing the system's output against the expected results without worrying about the internal implementation details.
- White Box Testing: For critical components, the internal code logic will be reviewed and tested to ensure proper control flow and error handling.

7.3 Testing Methods

1.Functional Testing:

- Ensures that all features of the system are working according to the functional specifications. This includes registration, login, blood requests, and stock management.

2. **Non-Functional Testing:**
   - Performance Testing: Verifies how well the system performs under load, including response time and scalability.
3. **Security Testing:**
   - Ensures user data (donors, hospitals) is protected using appropriate authentication, authorization, and encryption mechanisms.
4. **Regression Testing:**
   - Conducted after adding new features or fixing bugs to ensure no previously functioning modules have been broken by the changes.
5. **User Acceptance Testing (UAT):**
   - Conducted with stakeholders (such as actual hospital and donor users) to ensure the system is user-friendly and meets the end-users' needs.

7.4 Test Cases (Purpose, Required output, Expected Result)

Test Case 1: User Login Validation

- Purpose: To validate that users can successfully log in with valid credentials.
- Scenario: The user enters correct login credentials on the login page.

Test Case 2: Invalid Login Attempt

- Purpose: To ensure that the system rejects incorrect login attempts.
- Scenario: The user enters invalid credentials (either username or password is incorrect).

Test Case 3: Blood Request by Hospital

- Purpose: To verify the functionality of requesting blood by hospitals.
- Scenario: A hospital user attempts to request blood units of a specific type.

Test Case 4: Donor Registration Validation

- Purpose: To ensure the donor registration form performs proper input validation.
- Scenario: The donor enters invalid details (e.g., wrong email format or incorrect phone number).

Test Case 5: Blood Donation Scheduling

- Purpose: To confirm that donors can successfully schedule blood donations.
- Scenario: A registered donor schedules a future date for blood donation.

Test Case 6: Blood Stock Update After Donation

- Purpose: To verify that blood stock is updated after a donation is made.
- Scenario: Admin updates the system with a new blood donation.

Test Case 7: Role-Based Access Control

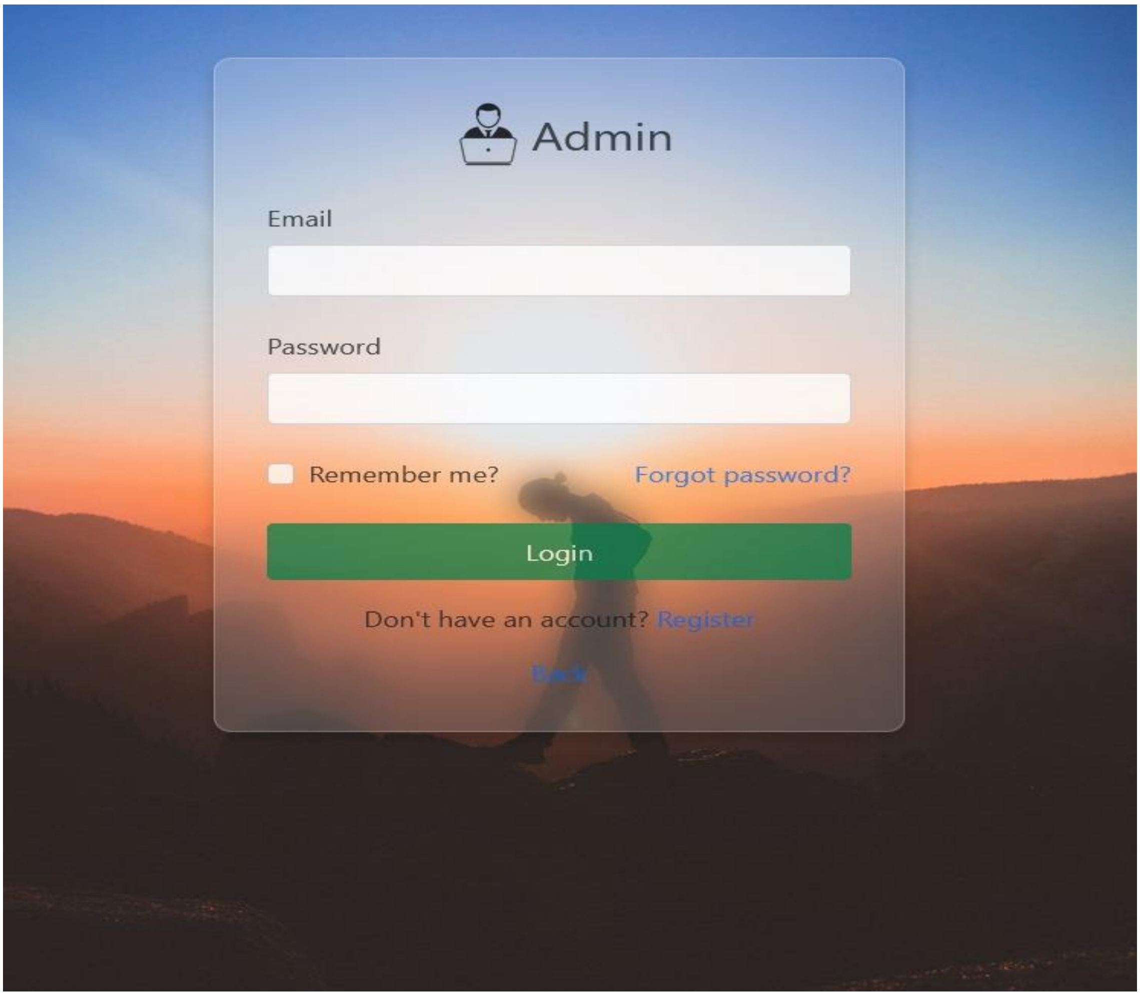
- Purpose: To ensure that unauthorized users cannot access restricted areas.
- Scenario: A hospital user attempts to access an admin-only section.

Test Case 8: Blood Stock Availability Display

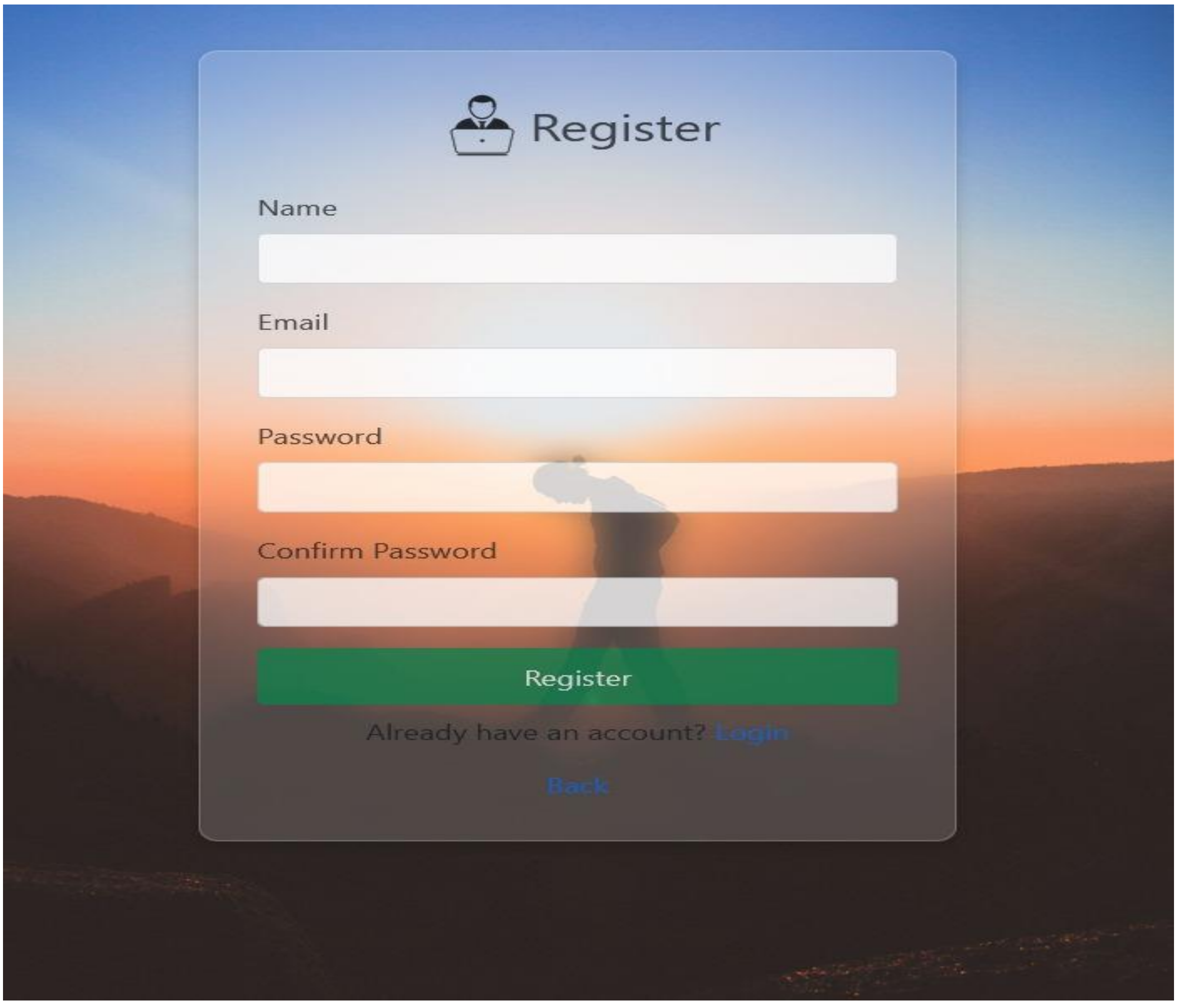
- Purpose: To validate that the blood stock availability is displayed correctly for donors.
- Scenario: A donor checks the current blood stock levels.

8.0 SCREEN SHOTS AND USER MANUAL:

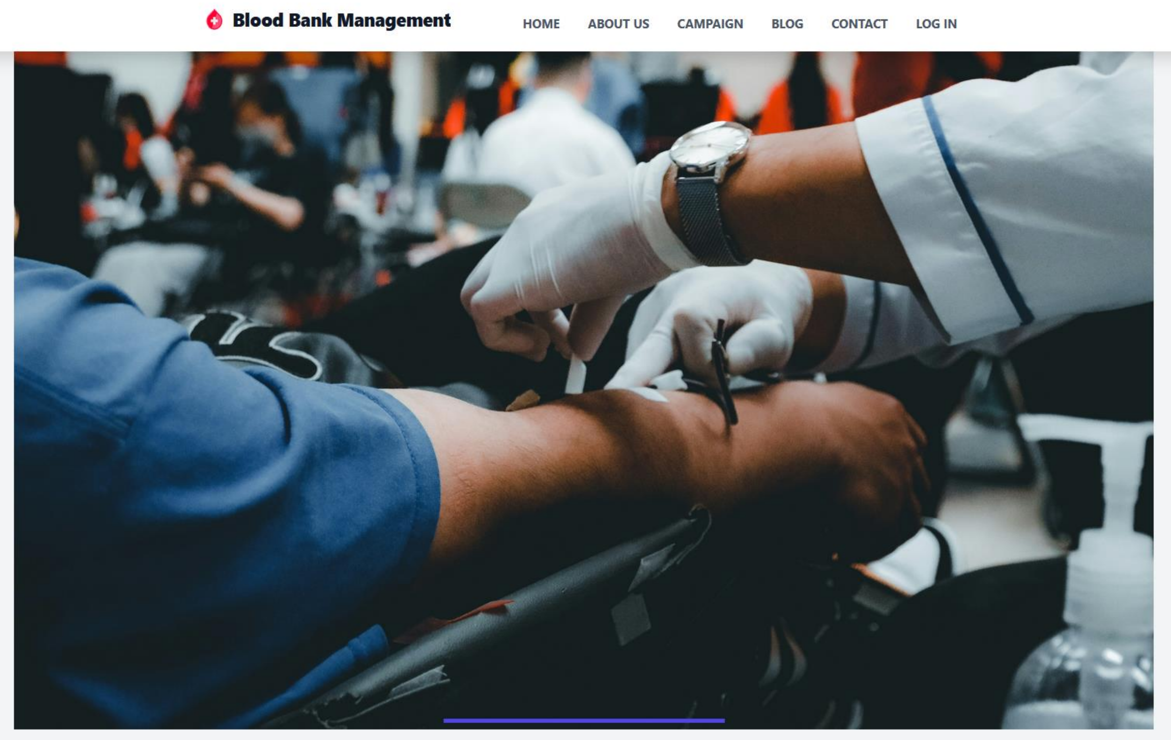
Login Page

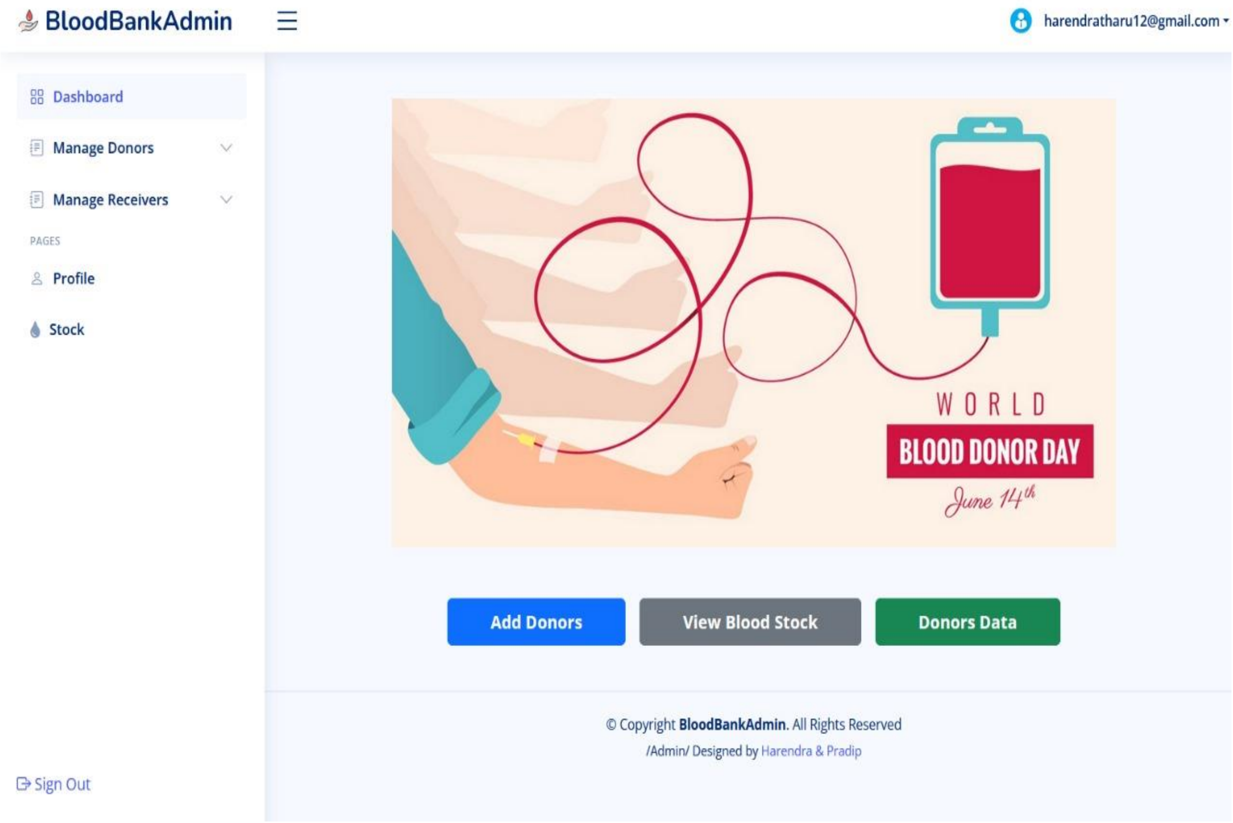


Register Page

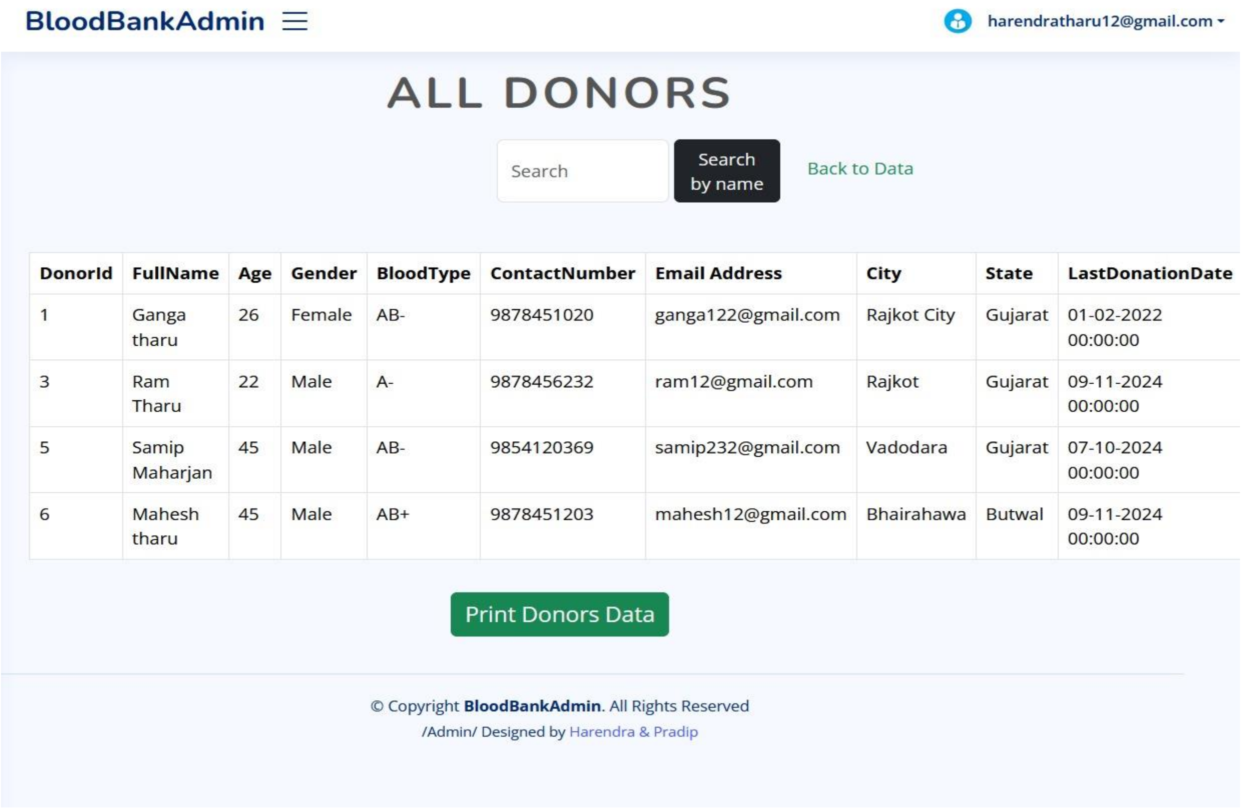


Home Page:





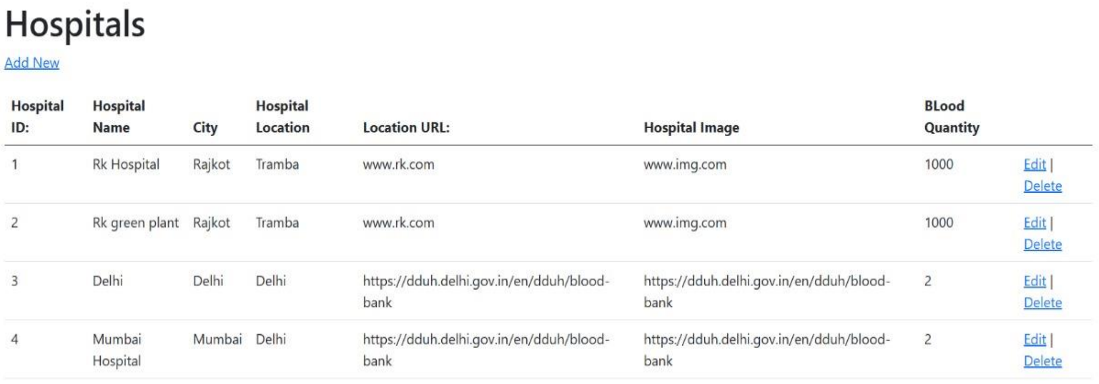
© Copyright **BloodBankAdmin**. All Rights Reserved
/Admin/ Designed by Harendra & Pradip



ALL DONORS

DonorId	FullName	Age	Gender	BloodType	ContactNumber	Email Address	City	State	LastDonationDate
1	Ganga tharu	26	Female	AB-	9878451020	ganga122@gmail.com	Rajkot City	Gujarat	01-02-2022 00:00:00
3	Ram Tharu	22	Male	A-	9878456232	ram12@gmail.com	Rajkot	Gujarat	09-11-2024 00:00:00
5	Samip Maharjan	45	Male	AB-	9854120369	samip232@gmail.com	Vadodara	Gujarat	07-10-2024 00:00:00
6	Mahesh tharu	45	Male	AB+	9878451203	mahesh12@gmail.com	Bhairahawa	Butwal	09-11-2024 00:00:00

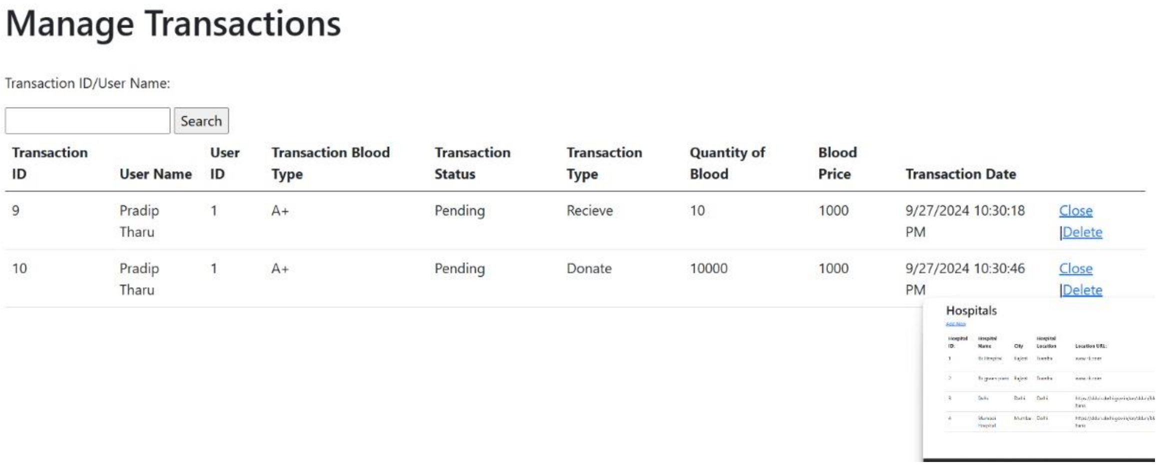
© Copyright **BloodBankAdmin**. All Rights Reserved
/Admin/ Designed by Harendra & Pradip



Hospitals

Add New

Hospital ID:	Hospital Name	City	Hospital Location	Location URL:	Hospital Image	BLood Quantity	
1	Rk Hospital	Rajkot	Tramba	www.rk.com	www.img.com	1000	Edit | Delete
2	Rk green plant	Rajkot	Tramba	www.rk.com	www.img.com	1000	Edit | Delete
3	Delhi	Delhi	Delhi	https://dduh.delhi.gov.in/en/dduh/blood-bank	https://dduh.delhi.gov.in/en/dduh/blood-bank	2	Edit | Delete
4	Mumbai Hospital	Mumbai	Delhi	https://dduh.delhi.gov.in/en/dduh/blood-bank	https://dduh.delhi.gov.in/en/dduh/blood-bank	2	Edit | Delete

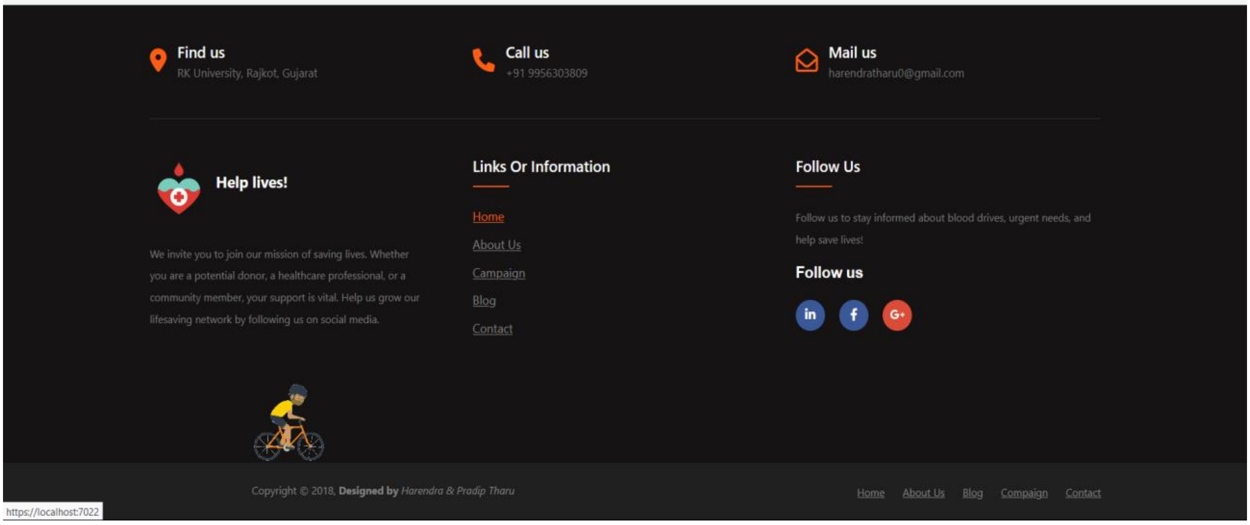


Manage Transactions

Transaction ID/User Name:

Search

Transaction ID	User Name	User ID	Transaction Blood Type	Transaction Status	Transaction Type	Quantity of Blood	Blood Price	Transaction Date	
9	Pradip Tharu	1	A+	Pending	Recieve	10	1000	9/27/2024 10:30:18 PM	Close |Delete
10	Pradip Tharu	1	A+	Pending	Donate	10000	1000	9/27/2024 10:30:46 PM	Close |Delete



Find us
RK University, Rajkot, Gujarat

Call us
+91 9956303809

Mail us
harendratharu0@gmail.com

Help lives!

We invite you to join our mission of saving lives. Whether you are a potential donor, a healthcare professional, or a community member, your support is vital. Help us grow our lifesaving network by following us on social media.

Links Or Information

- Home
- About Us
- Campaign
- Blog
- Contact

Follow Us

Follow us to stay informed about blood drives, urgent needs, and help save lives!

Follow us

Copyright © 2018, **Designed by** *Harendra & Pradip Tharu*

Home About Us Blog Compaign Contact

https://localhost:7022

9.0 Limitation and Future Enhancement

9.1 Limitations

- **Limited User Roles:**
  The system currently supports only three user roles: Admin, Hospitals, and Donors. Expanding the system to accommodate additional stakeholders, such as Blood Bank Staff or Volunteers, could enhance operations.
- **Manual Blood Stock Update:**
  Blood stock levels are updated manually by the admin after donations, which increases the risk of human error. Automating this process with real-time updates from donation centers would be more efficient.
- **No Integration with External Systems:**
  The system does not currently integrate with external health systems or hospital management software. This limits its potential for real-time updates and centralized access to broader health-related information.
- **Lack of Geolocation Features:**
  The system does not have geolocation capabilities for finding nearby donors or blood banks. This could be crucial for emergency situations where proximity matters.

9.2 Future Enhancements

- **Real-Time Blood Stock Tracking:**
  Implementing real-time blood stock tracking or integrating with hospital systems would allow for more accurate and up-to-date information on blood availability.
- **Geolocation and Donor Proximity Alerts:**
  Adding a geolocation feature that allows hospitals to locate the nearest available donors during emergencies.
- **Automated Blood Donation Scheduling:**
  The system could automatically suggest available time slots based on a donor’s location, previous donations, and blood type, making the donation process more efficient for both donors and blood banks.
- **Mobile Application Development:**
  Developing a dedicated mobile app for donors and hospitals would increase usability and accessibility, enabling users to schedule donations, check blood availability, and receive alerts while on the move.

10 CONCLUSION

10.1 Conclusion

The development of the Blood Bank Management System using ASP.NET MVC aimed to streamline the processes involved in blood donation, blood stock management, and blood requests. The system provides a centralized platform for admin, hospitals, and donors, improving the efficiency of blood bank operations by facilitating easier communication and management.

Through extensive testing, the system’s functionality has been validated for core features such as user authentication, blood requests, and stock updates. While the system is functional and meets the initial requirements, several limitations have been identified, including the lack of real-time blood stock tracking, manual updates, and limited user roles.

11. REFERENCES

- **Bootstrap. (n.d.). *Introduction*. Retrieved from Bootstrap Documentation.**
- **Bootstrap: JavaScript · Bootstrap v5.3**
- **Website: e-RaktKosh:Centralized Blood Bank Management System**
- **Youtube:** Step-by-step ASP.NET MVC Tutorial for Beginners | Mosh