TRIBHUVAN UNIVERSITY

Faculty of Bachelor of Science in Computer Science and Information Technology

A Project Report

On

"Blood Donation Management System using Weighted Scoring Model"

Submitted to

Department of Computer Science and Information Technology

Padmakanya Multiple Campus

Bagbazar, Kathmandu

In partial fulfillment of the requirements for the Bachelor's in Computer Science and Information Technology

Submitted by

Anupa Baral(TU Roll No.: 25785/077)

Reshma Shrestha(TU Roll No.: 25799/077)

Samiksha Khadka(TU Roll No.: 25801/077)

Under the supervision of

Sudip Raj Khadka

18th March 2025

Date: 18/03/2025

Padmakanya Multiple Campus

Bagbazar, Kathmandu

Supervisor's Recommendation

I hearby recommend that the dissertation prepared under my supervision by the team of **Anupa Baral, Reshma Shrestha** and **Samiksha Khadka** entitled **"Blood Donation Management System"** be accepted as fulfilling in partial requirements for the degree of Bachelor's in Computer Science and Information Technology.

……

Mr. Sudip Raj Khadka

Supervisor

Padmakanya Multiple Campus

Bagbazar, Kathmandu

TRIBHUVAN UNIVERSITY

Faculty of Bachelor of Science in Computer Science and Information Technology

Padmakanya Multiple Campus

Bagbazar, Kathmandu

Letter of Approval

This is to certify that this project prepared by **Anupa Baral, Reshma Shrestha** and **Samiksha Khadka** entitled **"Blood Donation Management System using Weighted Scoring Model"**, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and qualifies as a project for the required degree.

Evaluation Committee

Mr. Sudip Raj Khadka **Supervisor**	**External Examiner**	**Mr. Ramesh Saud** **Head of Department**

Acknowledgement

We would like to express our special thanks of gratitude to our supervisor **Mr. Sudip Raj Khadka sir** who gave us the golden opportunity, guidance, inspiration and constructive suggestion to do this wonderful project on the topic **"Blood Donation Management System using Weighted Scoring Model"**, which also helped us in doing a lot of research and we came to know about so many tools and technologies. We would also like to express a deep sense of gratitude to him, for his exemplary guidance, monitoring and constant encouragement throughout which helped us in completing this task through various stages.

A special thanks to Padmakanya Multiple Campus for providing the opportunity and resources essential for the completion of this project. The unwavering support from the college administration has played a crucial role in the successful execution of our work.

Abstract

The Blood Donation Management System is a web-based platform designed to facilitate efficient blood donation and distribution. This system enables donors and recipients to register, manage their accounts, and access real-time blood availability through an intuitive Front-end Interface. The platform is supported by a System Back-end that handles authentication, data processing, and secure communication between users, administrators, and the System Database, which stores critical blood donors information and recommends which best fits them. The deployment follows a client-server architecture, ensuring that user interactions occur over HTTP(S) while core functionalities remain protected within a private network. An Admin Interface allows authorized personnel to oversee operations, verify donor eligibility, and manage blood requests. The system's secure and scalable architecture ensures data integrity, privacy, and availability, enhancing the overall efficiency of blood donation processes. By streamlining donor-recipient connections and ensuring a secure data flow, this system aims to improve accessibility and responsiveness in life-saving blood transfusion services.

Keyword : Weighted Scoring Model, Blood Donation, Multi-tiered, role-based

Table of Content

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
CSS	Cascading Style Sheet
FK	Foreign Key
HTML	Hyper Text Markup Language
JS	JavaScript
UI	User Interface

Chapter 1: Introduction

1.1 Introduction

Blood transfusion is a critical medical procedure that saves millions of lives annually, particularly in emergency situations, during surgeries, and for patients with certain medical conditions. However, in Nepal and many developing nations, the management of blood resources faces significant challenges including shortages, accessibility issues in rural areas, and logistical complications in distribution [1]. These challenges are especially pronounced in rural regions where women during childbirth face life-threatening situations due to unavailability of compatible blood types in a timely manner.

Nepal faces particular difficulties in blood transfusion services, with limited donation rates compared to international standards. The blood transfusion service in Nepal has evolved over time but continues to struggle with coordination and resource allocation challenges [1]. Furthermore, postpartum hemorrhage remains a leading cause of maternal mortality in Nepal, with many cases potentially preventable through improved access to blood transfusion services [2]. The emergency obstetric care projects in Nepal have highlighted these challenges and stimulated policy debate on improving blood transfusion services throughout the country.

1.2 Problem Statement

Despite advances in healthcare technology in Nepal, blood resource management remains largely fragmented and inefficient. Current systems lack:

- Real-time inventory tracking across multiple facilities
- Coordination between hospitals, government bodies, and non-profit organizations like Red Cross
- Recommends the best donor available across multiple donors
- Geographic mapping of donor location .

1.3 Objectives

- To develop real-time inventory tracking and visualization tools to provide information about the donor for the blood request.
- To create an efficient donor registration and eligibility management system with priority-based allocation
- To develop a secure, scalable, and user-friendly blood donation management system that increases blood availability for patients .

1.4 Scope and Limitation

There are various scope and limitations of the Blood Donation Management System.

1.4.1 Scope

The scope of Blood donation management system are listed as follows:

- Build a secure, scalable blood transfusion management system using .NET 8 (backend), React (frontend), and MySQL (database).
- Real-time blood inventory tracking, donor-patient matching, multi-tiered access control, and cross-functional collaboration between hospitals, local governments, and the Red Cross.
- Hospitals, local governments (Nagarpalikas), Red Cross, donors, and patients have their role based dashboard.

1.4.2 Limitation

The limitations on the system are:

- The system will focus on blood donation management and will not include other medical supply management.
- Initial deployment will be limited to a specific region only.
- Geographic mapping of donor location is approximate.

1.5 Development Methodology

The development of the Blood Donation Management System follows an agile and iterative approach to ensure efficiency, flexibility, and user satisfaction. The methodology

integrates best practices in software engineering, prioritizing user needs, security, and scalability.

The Agile methodology, coupled with secure and scalable architecture, ensures the development of a robust Blood Donation Management System. By integrating continuous feedback and iterative improvements, the system remains reliable, user-friendly, and adaptable to future advancements.

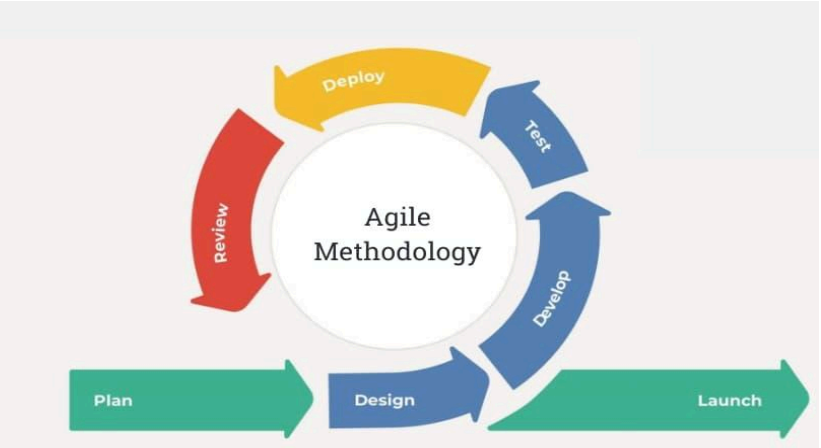


Figure 1.5: Agile Methodology for Blood Donation Management System

1.6 Report Organization

Report organization describes the organization of report i.e., what the different topic does and what the purpose of it. In this report generally, there are five important topics: Introduction, Background study and literature review, System analysis and design, Implementation and testing and Conclusion and future recommendation.

Chapter 1 describes how the system will recommend the user. It also includes the description about why this system was developed with the objective of the system aims to fulfill. Similarly scope and limitation of project are also defined.

Chapter 2 consists of review of several research papers that have been studied to gather knowledge about the Blood Donation Management and processing steps involved in the application. Input and output design of the system is explained.

Chapter 3 includes a description about how the implementation of the system is done including the various tools and technology of the System. Several papers regulating Blood Donation Management are reviewed. Requirement analysis and feasibility study are discussed.

Chapter 4 describes the overall structure, interfaces used. Also includes the listing and description of the major classes used in the application and several test cases performed so as to test the proper functioning of the application.

Chapter 5 includes the conclusions that were drawn after the complete development and testing of the application. And recommendations on how the system can be further improved.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Blood donation plays a crucial role in saving lives, particularly in emergencies, delivery, surgeries, and the treatment of medical conditions such as anemia, cancer, and blood disorders. Traditional blood donation systems often rely on manual record-keeping and communication, which can lead to inefficiencies, delays, and inaccuracies in matching donors with recipients. The lack of a centralized, real-time database further complicates the process, making it difficult for hospitals and blood banks to manage blood inventory effectively.

With advancements in technology, digital solutions have emerged to streamline blood donation management. Web-based platforms offer a centralized system where donors can register, check donation eligibility, and schedule appointments, while recipients can search for available blood units. A well-structured Blood Donation Management System enhances efficiency by providing real-time updates, automated notifications, and secure access to blood records.

This project leverages a client-server architecture, incorporating a front-end interface, a system back-end, and a secure database to manage donor and recipient interactions. By ensuring data integrity, privacy, and accessibility, the system aims to reduce the time required to find suitable blood donors and improve the overall coordination of blood donation services. The implementation of such a platform will significantly enhance the responsiveness and reliability of blood donation networks, ultimately saving more lives.

2.2 Literature Review

Research by Clapham et al. [2] on emergency obstetric care in Nepal revealed significant challenges in blood transfusion services, particularly in rural areas. Their work through emergency obstetric care projects demonstrated how improved access to blood transfusion services could substantially reduce maternal mortality rates. The study highlighted the need for policy reforms and better coordination among healthcare facilities to ensure timely access to blood products during obstetric emergencies.

The involvement of multiple stakeholders in healthcare systems has been studied extensively. Haun et al. [3] describe a participatory approach to designing and enhancing integrated health information technology systems, emphasizing the importance of including various stakeholders in a cohesive digital ecosystem. Their research informs the multi-tiered approach needed for blood donation management systems. The success of participatory design demonstrates how stakeholder engagement leads to higher adoption rates and system sustainability over time.

Gurung [4] proposed a blood bank database sharing system for Nepal, addressing the critical need for integrated information management across blood banks nationwide. The proposed system aimed to connect disparate blood banks, allowing for real-time inventory tracking, donor management, and improved resource allocation during emergencies. This approach directly addresses the coordination challenges identified in Nepal's blood transfusion services.

Nepal's Ministry of Health and Population [5] has documented ongoing challenges in healthcare service delivery, including blood transfusion services. Their annual reports highlight regional disparities in access to blood products and the need for improved infrastructure, particularly in remote areas.

Parajuli et al. [6] examined the challenges and opportunities for implementing digital health interventions in Nepal. Their review identified infrastructure limitations, training gaps, and regulatory challenges that must be addressed for successful digital health implementation, including blood bank management systems. Despite these challenges, they noted significant opportunities for technology to improve healthcare delivery in Nepal.

Gupta [7] investigated cybersecurity issues in the Nepalese context, highlighting significant vulnerabilities in healthcare systems. For blood bank management systems, these security concerns are particularly critical given the sensitive nature of donor health information and patient matching data. Addressing these security challenges is essential for any blood donation management system implementation in Nepal.

Chapter 3: System Analysis

3.1 System Analysis

The requirement analysis for the Blood Donation Management System involves a comprehensive breakdown of the system's functional and non-functional requirements. These requirements guide the design and development process, ensuring that the final system meets the need. The analysis includes:

3.1.1 Requirement Analysis

There are two types of requirements:

i. Functional Requirements

a. User registration and authentication .
   - Register super admin
   - Allow super admin to register users (different tier admins)
   - Store user information securely and hash passwords before storing in database

b. Blood request and donation scheduling system.
   - Register donors
   - Register patients' cases awaiting blood
   - Prioritize scheduling of patients' cases according to urgency
   - Match available donors specific to patients' cases

c. Real-time blood inventory tracking.
   - Graph to track available blood units in inventory according to blood group
   - Track available donors and patients' cases for blood requests

Actors:

1. Hospital
2. Local Government (Municipality)
3. Red Cross
4. Super Admin

.

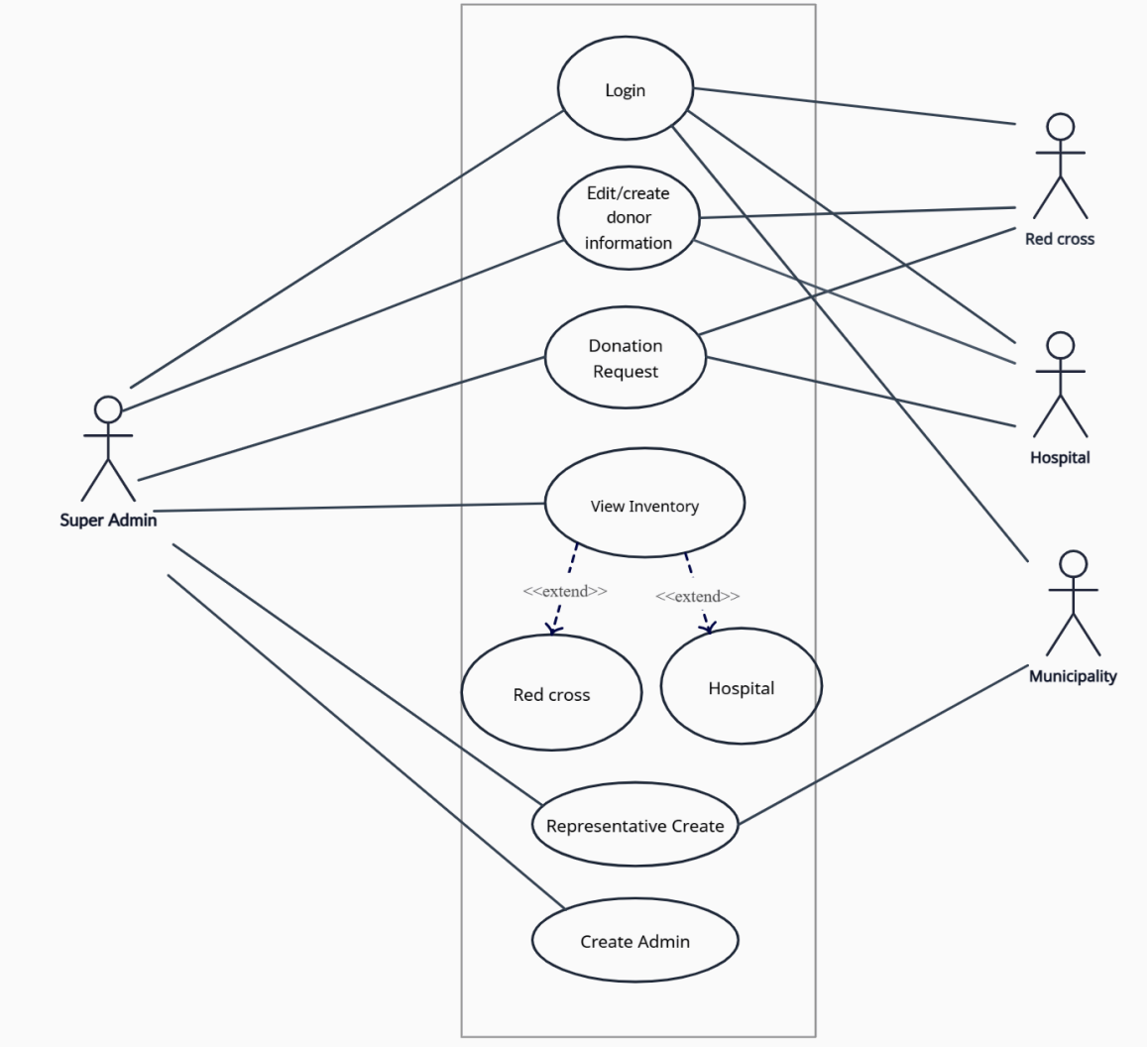


Figure 3.1.1: Use Case Diagram

ii. Non Functional Requirements

The system should provide following non-functional requirements:

- **Scalability:** The system will efficiently handle increasing numbers of users and data volumes through horizontally scalable architecture and optimized database queries.
- **Security:** Implementation of industry-standard encryption for sensitive data, role-based access control, secure authentication protocols, and regular security audits to protect patient and donor information.
- **Usability:** Intuitive UI/UX design featuring clear navigation paths, consistent interface elements, and responsive design that works across various devices to ensure ease of use for all user types.

- **Availability:**System will maintain a minimum uptime of 99.5% with comprehensive disaster recovery strategies including automated daily backups, redundant server configurations, and failover mechanisms. The system incorporates pre-loaded geographical location data (provinces, districts, and municipalities) to enable offline functionality in rural areas with limited internet connectivity, ensuring critical blood management services remain operational regardless of network status.

3.1.2 Feasibility Analysis

A feasibility study considers many factors, including economic, technical, operational, and scheduling to determine whether a project can succeed. It analyses the proposed system from different aspects so that it makes clear how the system will be practical and functioning properly.

Technical Feasibility

The system will utilize modern web and mobile technologies, ensuring cross-platform accessibility. With cloud-based storage and an API-driven approach, the system can efficiently handle real-time blood inventory management. The use of robust backend frameworks (Dotnet) and relational databases (MySQL) ensures data integrity and high availability.

Operational Feasibility

Hospitals, blood banks, and non-profit organizations like redcross, local government can easily adopt the system due to its user-friendly interface and intuitive features. Training sessions and awareness campaigns can further facilitate smooth adoption. Moreover, real-time data analytics will help health organizations make informed decisions regarding blood management.

Economical Feasibility

The project requires an initial investment for system development, hosting, and maintenance. However, the long-term benefits—such as reduced wastage of blood units, improved donor management, and decreased maternal mortality rates—outweigh the

costs. Funding options include partnerships with government health agencies, NGOs, and private healthcare institutions.

Schedule Feasibility

We have a proper schedule and enough time to complete this project in time. So this project is schedule feasible.

3.1.3 Analysis

Object Modelling using Class and Object Diagrams

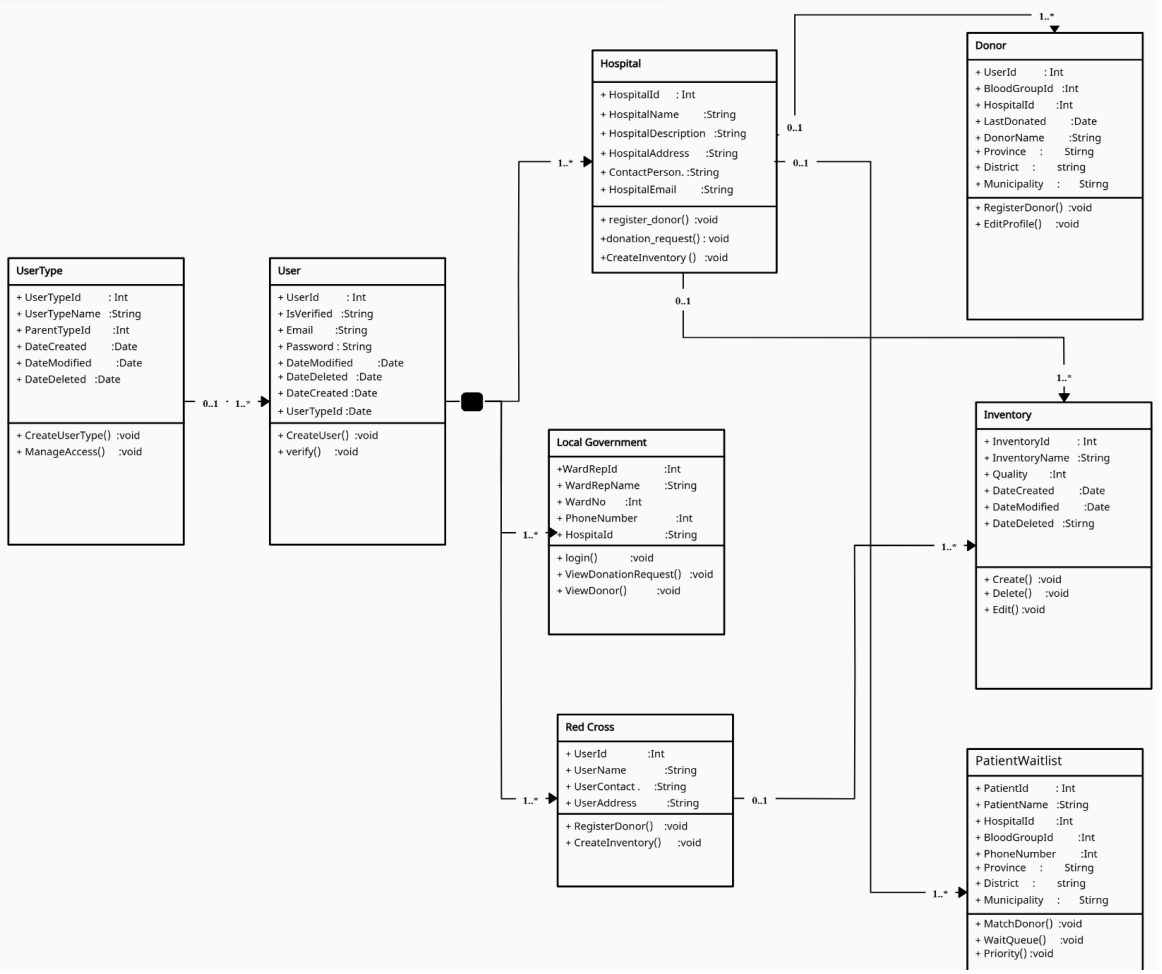


Figure 3.1.3.a: Class Diagram

This class diagram illustrates a blood bank management system with the following key classes

1. UserType

Defines roles in the system with attributes like UserTypeId, UserTypeName, etc.

Methods are:

- **CreateUserType()**: Creates a new role/permission level in the system.
- **ManageAccess()**: Controls what different user types can access or modify.

.

2. User

Contains user authentication details including UserId, Email, Password, verification status, and timestamps.

Methods are :

- **CreateUser()**: Registers a new user in the system.
- **verify()**: Confirms user identity/email for authentication.

3. Hospital

Central entity that manages donors, maintains inventory, and handles donation requests. Has attributes for contact information and methods for donor registration.

Methods are :

- **register_donor()**: Enrolls a new blood donor into the system.
- **donation_request()**: Creates a new blood donation request.
- **CreateInventory()**: Establishes new blood storage record.

4. Donor

Stores donor information including blood group, donation history, and location details (Province, District, Municipality).

Methods are:

- **RegisterDonor()**: Creates new donor profile with personal and medical details.
- **EditProfile()**: Updates donor information or availability.

5. Inventory

Tracks blood supplies with attributes for inventory identification, quality assessment, and standard CRUD methods.

Methods:

- **Create()**: Adds new blood unit to inventory.
- **Delete()**: Removes blood unit from inventory.
- **Edit()**: Updates blood unit information or status.

6. Local Government

Represents administrative entities with ward details and methods to view donation requests and donors.

Methods:

- **ViewDonationRequest()**: Accesses requests for blood donation.
- **ViewDonor()**: Reviews donor information in their jurisdiction.

7. Red Cross

Another organizational entity that can register donors and create inventory.

Methods:

- **RegisterDonor()**: Enrolls donors through Red Cross channels.
- **CreateInventory()**: Manages blood inventory for Red Cross facilities.

7. Patient Waitlist

Manages patients needing blood with methods for matching donors, queue management, and prioritization.

Methods:

- **MatchDonor()**: Finds compatible donors for patients.
- **WaitQueue()**: Manages patient order in the waiting list.
- **Priority()**: Assigns urgency levels to patients needing blood.

Dynamic Modelling using State and Sequence Diagrams

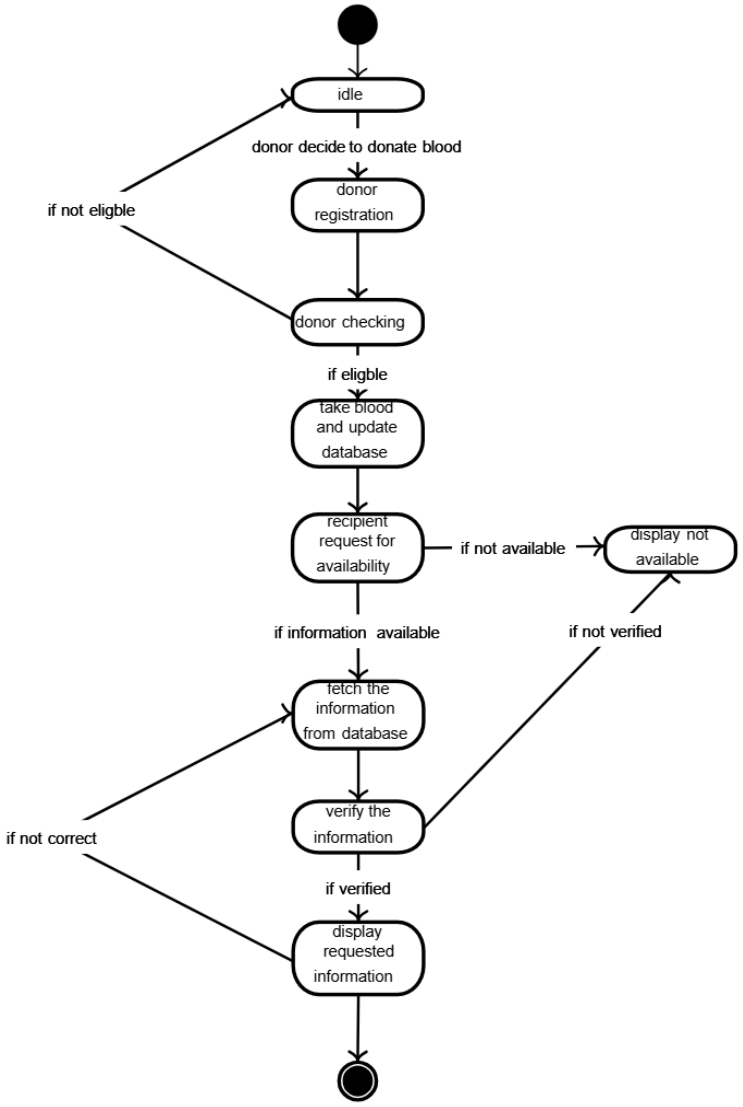


Figure 3.1.3.b: State Diagram

This flowchart illustrates a blood donation management system that begins with a donor deciding to donate blood, followed by registration and eligibility verification. If eligible, blood is collected and the database updated; if not, the process returns to idle. When recipients request blood, the system checks availability in the database. If available, information is retrieved and verified before being displayed to the recipient; if unavailable or unverified, a "not available" message appears. The diagram effectively maps the complete journey from donation to recipient request fulfillment, showing decision points and data flows throughout the process.

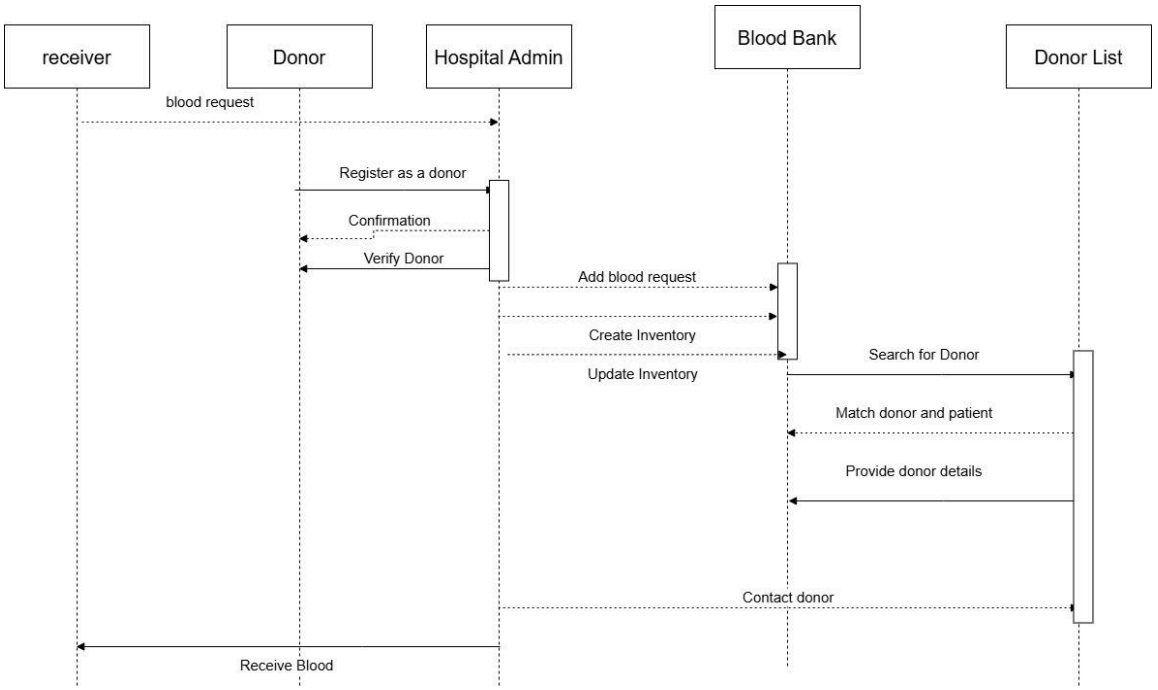


Figure 3.1.3.c: Sequence Diagram

This sequence diagram depicts the core interactions in a blood bank management system across three user types. Donors and requesters first sign up to receive unique identifiers, then proceed through their respective workflows. Donors log in, respond to donation requests, and monitor the status of their contributions. Requesters log in, submit blood requests, and check both donor and blood availability. Administrators oversee the entire process by reviewing all requests, making decisions, and managing system operations. The diagram effectively shows the information exchange between users through status updates, request submissions, and approval processes, illustrating how the blood bank system coordinates donors with those in need while maintaining administrative oversight.

Process Modelling using Activity Diagram

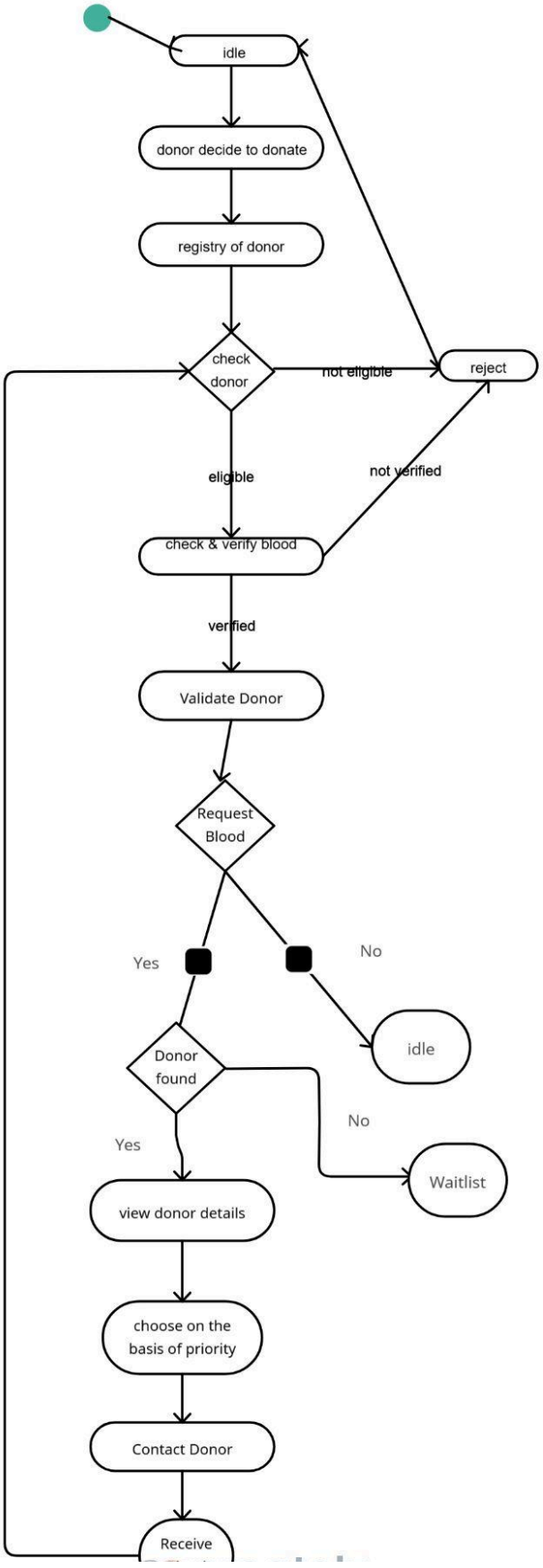


Figure 3.1.3.d: Activity Diagram

The Activity Diagram for the Blood Donation Management System outlines the key processes involved in blood donation and requests. It starts with user registration/login, where donors, patients, or hospitals access the system. Donors can schedule a blood donation appointment, undergo eligibility verification, and proceed with donation if eligible, after which the blood inventory is updated, and a confirmation message is sent. On the other hand, hospitals or patients can request blood, triggering an inventory check. If available, the request is approved, and blood is allocated and dispatched; otherwise, the system notifies the requester and searches for potential donors. The process ensures efficient blood management and timely transfusions.

Chapter 4: System Design

4.1 Design

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

4.1.1 Refinement of Class diagram

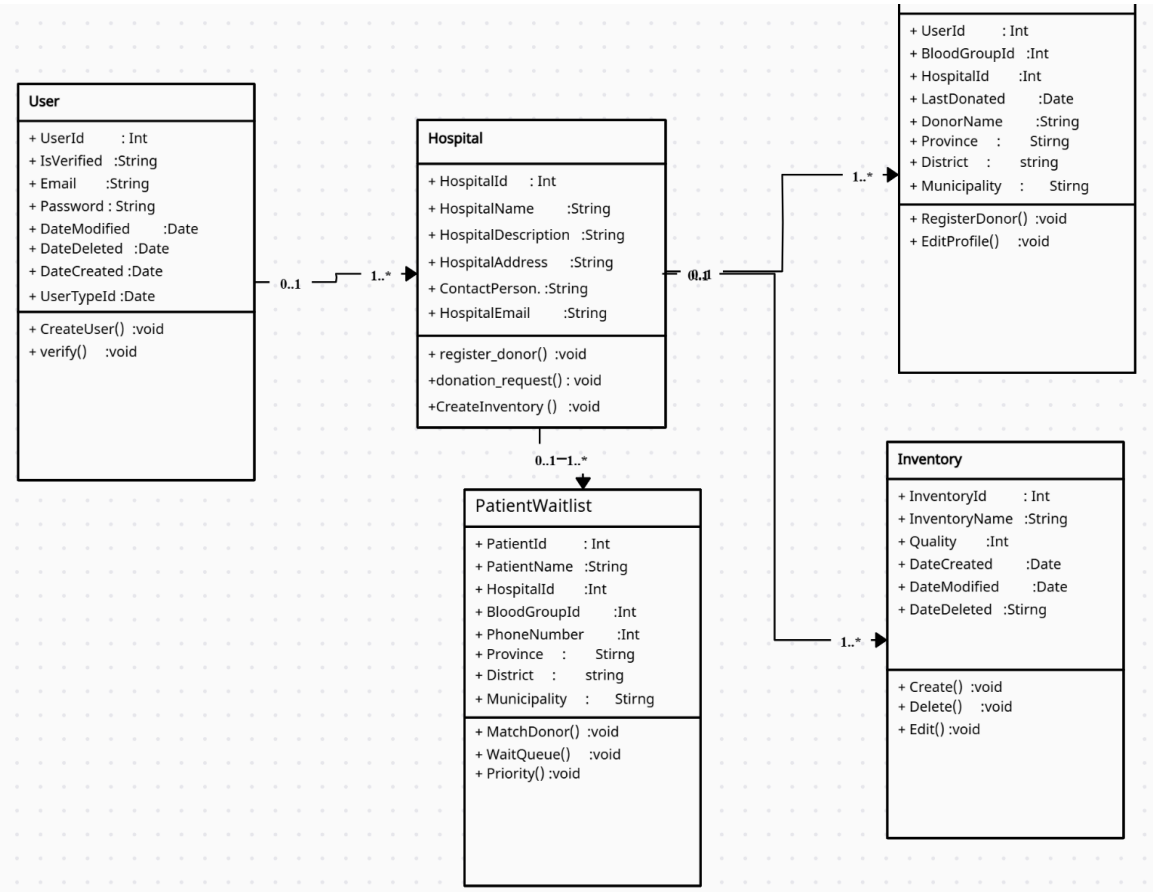


Figure 4.1.1: Refinement Class Diagram

The refined blood donation system class diagram introduces a comprehensive framework with five integrated classes. User provides authentication and account management functionality, serving as the entry point to the system. Hospital functions as the central hub, managing donor registrations, blood donations, and inventory while maintaining relationships with all other classes. The Donor class stores personal information, blood type, and donation history. The Inventory class tracks blood supplies with standard CRUD operations. The newly added PatientWaitlist class completes the donation lifecycle by managing blood recipients with matching algorithms, queue management, and prioritization functions. This structure efficiently connects blood donors to recipients

through hospital management while maintaining data integrity with consistent attribute typing and relationship cardinalities.

4.1.3 Deployment Diagram

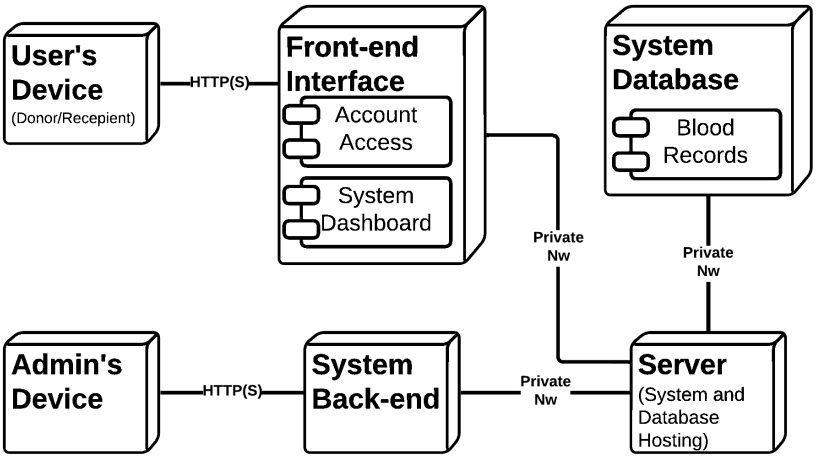


Figure 4.1.3: Deployment Diagram

The deployment diagram represents the architecture of a blood donation management system, illustrating how various components interact within the network. The system consists of multiple devices and servers connected through secure communication channels. Users, including donors and recipients, access the system through a Front-end Interface via HTTP(S), which provides features like account access and a system dashboard. Similarly, an Admin's Device connects to the system backend using HTTP(S) to manage system operations. The System Back-end, hosted on a secure Server, handles core application logic and facilitates communication between the front-end and the System Database, which stores critical blood records. Both the back-end and database reside within a private network to ensure security and prevent unauthorized access. The Server hosts both the system's logic and database, ensuring smooth operation and data integrity. This architecture highlights the secure flow of information, ensuring that sensitive data is protected while maintaining efficient user interaction.

4.2 Algorithm Details

Algorithm 1: Blood Donor Ranking System

This algorithm ranks eligible blood donors based on three key factors: Recency of Donation, Proximity to Hospital, and Age. The goal is to create a weighted scoring model that prioritizes donors effectively.

Steps and Details

1. **Data Input:**
    - Input data consists of donor information, including:
        - days_since_last_donation: Number of days since the last donation.
        - distance_km: Distance (in km) from the donor's location to the hospital.
        - age: Age of the donor.
2. **Recency Score Calculation:**
    - Eligibility: Donors must have donated between 90 days (3 months) and 365 days (1 year) ago.
    - Scoring Logic:
        - Donors who donated closer to 120 days (4 months) ago receive the highest score.
        - Donors who donated closer to 365 days ago receive the lowest score.
        - Linear interpolation is used for values between 120 and 365 days.
    - **Formula:**

      Recency Score = 1 - ((days_since_last_donation - 120) / (365 - 120))

        - If days_since_last_donation < 120, the score is set to 1.
        - If days_since_last_donation > 365, the score is set to 0.
3. **Proximity Score Calculation:**
    - Definition: Distance (in km) from the donor's location to the hospital.
    - Scoring Logic:
        - Donors closer to the hospital receive a higher score.
        - The score is normalized between 0 and 1 based on the maximum distance in the dataset.
    - Formula: Proximity Score = 1 - (distance_km / max_distance)
4. **Age Score Calculation:**

- Definition: Age of the donor.
- Scoring Logic:
  - Younger donors (18–37 years) receive a higher score.
  - The score is normalized between 0 and 1.
- Formula: Age Score = 1 - ((age - 18) / (37 - 18))
  - If age < 18, the score is set to 1.
  - If age > 37, the score is set to 0.

5. **Weighted Total Score Calculation:**
   - Assign weights to each factor based on importance:
     - Proximity Score: 50% weight.
     - Recency Score: 30% weight.
     - Age Score: 20% weight.
   - Formula: Total Score = 0.5 × Proximity Score + 0.3 × Recency Score + 0.2 × Age Score
6. **Ranking Donors:**
   - Donors are sorted by their total score in descending order.
   - The donor with the highest score is the most recommended.

Algorithm 2: Blood Donation Request Prioritization

This algorithm prioritizes blood donation requests based on urgency and time since posting. It strictly prioritizes high-priority cases (e.g., accidents, surgery complications) and dynamically ranks medium and low-priority cases (e.g., pregnancy, planned surgery).

Steps and Details

1. Data Input:
   - Input data consists of blood donation requests, including:
     - id: Unique identifier for the request.
     - urgency: Urgency level (e.g., "accident", "pregnancy", "planned_surgery").

        - posted_time: Timestamp when the request was posted.

2. **Priority Weights:**
    - Assign weights to each urgency level:
        - accident = 1 (highest priority).
        - surgery_complication = 1 (highest priority).
        - pregnancy = 2 (medium priority).
        - planned_surgery = 3 (lowest priority).
3. **Separate High-Priority Cases:**
    - Filter out all requests with priority weight = 1 (accidents and surgery complications).
    - Sort these requests by posted_time (oldest first).
4. **Rank Remaining Cases:**
    - For requests with priority weight > 1 (pregnancy and planned surgery), calculate a score using the

      formula: Score = Priority Weight + (1 / Time Since Posting)

    - Sort these requests by their score (ascending).
5. **Combine Results:**
    - Concatenate the sorted high-priority cases with the sorted remaining cases.
    - Output the final sorted list of requests.

Chapter 5: Implementation and Testing

5.1 Implementation

Systems implementation is a set of procedures performed to complete the design (as necessary) contained in the approved systems design document and to test, install, and begin to use the new or revised Information System. The primary data and information about the project were collected through different web sources and manually doing research on the relevant field. Different kinds of tools and techniques were used in our software to make it functional and useful at the same time. The project was developed using .Net.

5.1.1 Tools used

i. Front-End Tool

- **HTML**

  HTML is used for designing user interface to be displayed in a web browser along with HTTP response and request models.

- **CSS/Tailwind**

  CSS and Tailwind is used for styling the content of user interface which make our website user friendly.

- **JavaScript/React**

  It brings web pages to life by handling events, create animations, and communicating with servers to update content dynamically.React is used in this project to build a responsive and efficient front-end with a component-based architecture.

ii. Backend Tool

The different backend tools used in this project are:

- **.Net**

  .NET is a widely used, general-purpose, high-performance, and object-oriented framework developed by Microsoft. It supports multiple programming languages, with C# being the most popular. .NET is known for its versatility and is widely used for developing scalable, secure, and efficient server-side applications. For this project, the popular framework ASP.NET MVC is used, providing a robust and modern environment for building web applications and APIs.

iii. Documentation Tool

- **MS word**

  Microsoft word is a word processing and documentation creation platform in which this document is created.

iv. Diagram Tool

- **Draw.io**

  This is used to generate diagrams such as use case, class diagram, sequence diagram, state diagram and for system analysis and design of the TSR system. Diagrams were created using this tool in order to save time since all components are available with drag and drop functions.

5.1.2 Blood Donation Management System Implementation Guide

1. System Architecture

The Blood Donation Management System is built using a robust modern architecture combining .NET backend and React frontend. The architecture follows a clean separation of concerns with a RESTful API design pattern. The backend utilizes repository patterns, dependency injection, and data transfer objects (DTOs) to maintain clean code organization. The frontend leverages React with Redux for state management, providing an intuitive user interface with interactive components for different stakeholders: donors, hospitals, and patients. Data flows securely between frontend and backend through authenticated API calls, with password security implemented using SHA-256 hashing and salt mechanisms. A cloud-hosted database ensures data persistence and availability, while caching algorithms optimize geographic calculations for donor-patient matching.

2. Backend Implementation

The .NET backend forms the core of the system, built around a well-structured API that handles authentication, data processing, and business logic. Repository patterns (IGenericRepo and ISpecificRepo) provide abstraction layers for database operations, allowing clean separation between the database and business logic. Data transfer objects facilitate efficient data exchange between layers while reducing unnecessary data exposure. The system implements robust security measures including password hashing

with SHA-256 and salt generation to protect user credentials. Custom middleware handles authentication through API key validation in request headers.

3. Frontend Implementation

The React frontend delivers an intuitive and responsive user experience through thoughtfully designed components. Redux manages application state efficiently, maintaining consistency across different views and components. The user interface includes specialized dashboards for different user types with role-based access controls. Form components leverage React's component architecture for data entry with built-in validation. Interactive visualizations using Chart.js display key metrics such as blood type availability, and blood donation request. Leaflet.js integration provides interactive maps showing the locations of donors, and patients to assist in geographical decision-making. Toast notifications using Toastify provide user feedback for various actions and system events.The donor ranking and blood request prioritization algorithms are implemented as core services, leveraging the Haversine formula for accurate distance calculations between donors and recipients based on their geographic coordinates.The system implements a specialized "NepalCoordinates" module that locally stores geographical coordinates for all Nepalese administrative divisions. This enables critical distance calculations between donors and patients to function completely offline, ensuring the blood donation system remains operational in remote areas with limited internet connectivity, where such services are often most needed.

4. Algorithm Implementation

The system implements two critical algorithms that form the backbone of the matching and prioritization functionality. The Donor Ranking Algorithm evaluates potential donors based on a weighted scoring model considering three key factors: recency of donation (30% weight), proximity to hospital (50% weight), and donor age (20% weight). This ensures the most suitable donors are prioritized based on medical and logistical factors. The Blood Request Prioritization Algorithm sorts requests based on urgency levels and waiting time, with high-priority cases (accidents, surgery complications) always taking precedence over medium-priority (pregnancy) and low-priority cases (planned surgeries).

5. Data Management

Data management in the system focuses on efficiency, security, and integrity. The database schema mirrors the class diagram structure with tables for Users, Hospitals, Donors, Inventory, and PatientWaitlists. Cloud hosting ensures high availability and scalability as the system grows. The repository pattern abstracts database operations, providing a consistent interface for data access while enabling future flexibility to change the underlying database technology if needed. Data transfer objects (DTOs) and mappers ensure only necessary data is transferred between layers, improving security and performance. Database performance is optimized through appropriate indexing on frequently queried fields such as blood types, hospital IDs, and geographical coordinates.

6. Security Implementation

Security is integrated throughout the system architecture, starting with secure user authentication and authorization. Passwords are protected using SHA-256 hashing combined with random salt generation, making them resistant to dictionary and rainbow table attacks. The custom ValidatePassword method enforces strong password policies requiring combinations of uppercase, lowercase, numbers, and special characters. API key authentication for backend requests ensures that only authorized clients can access the endpoints. Role-based access control restricts user capabilities based on their assigned roles.

7. User Interface Components

The user interface is designed with a focus on usability and functionality for different user types. The donor portal features a registration form. Hospital dashboards include inventory management, donor search capabilities, and request management interfaces. Patient interfaces offer request submission forms, status tracking, and donor matching results. Interactive maps using Leaflet.js visualize the geographic distribution of donors, patients, and hospitals, helping users understand proximity relationships. Charting components built with Chart.js provide visual analytics for blood supply , donation demand . The responsive design ensures usability across devices of different sizes, from desktop to mobile.

8. Testing Strategy

A comprehensive testing strategy ensures system reliability and correctness. Unit tests validate individual components and algorithms, particularly the donor ranking and request prioritization algorithms. Integration tests verify the interaction between system components. API tests confirm endpoint functionality and error handling. The frontend includes component tests to verify rendering and user interaction behavior.

9. Deployment Considerations

Deployment leverages cloud infrastructure for scalability and reliability. The database is hosted in the cloud, providing backup and recovery capabilities.

5.2 Testing

Testing is a very important phase for any type of software. A software should go through a different testing process to ensure that the system is working in the manner in which it was intended to. During the phase of the development of the system, the system is tested time and again. The series of testing conducted are as follow:

5.2.1 Test Cases for Unit Testing

Generally, a software consists of several different modules and so does this project. A module or unit can refer to a function, individual program or even a procedure. Here is the list of test case:

Table 5.2.1: Test Cases for Unit Testing

Test Case	Scenario	Input	Expected Result	Actual Result	Test Result
1	New patient is added	Fill all the mandatory credentials	The patient should be saved in the database and retrievable.	The patient record is found in the database.	Pass
2	Find all donors with the same blood group as the patient.	Blood Group-:O+	Donors with blood group O+ should be retrieved.	Matching donors are retrieved correctly.	Pass

3	Donors should be sorted by how recently they donated.	Donors: Donor B: 100 days, Donor A: 50 days	Donors should be ordered as Donor A → Donor B.	Donors are sorted correctly by last donated days and by nearest distance.	Pass
4	Donors in the same municipality as the patient should be prioritized.	Donors: Donor B: Kathmandu, Donor A: Chaujhari	Donor A, same location as patient should appear first.	Donor A appears first in the list.	pass
5	Patients should be sorted by urgency level.	Patients: Patient A: Accident, Patient B: Planned Surgery.	Patients should be sorted as Patient A and Patient B	Patients are sorted correctly based on priority.	Pass

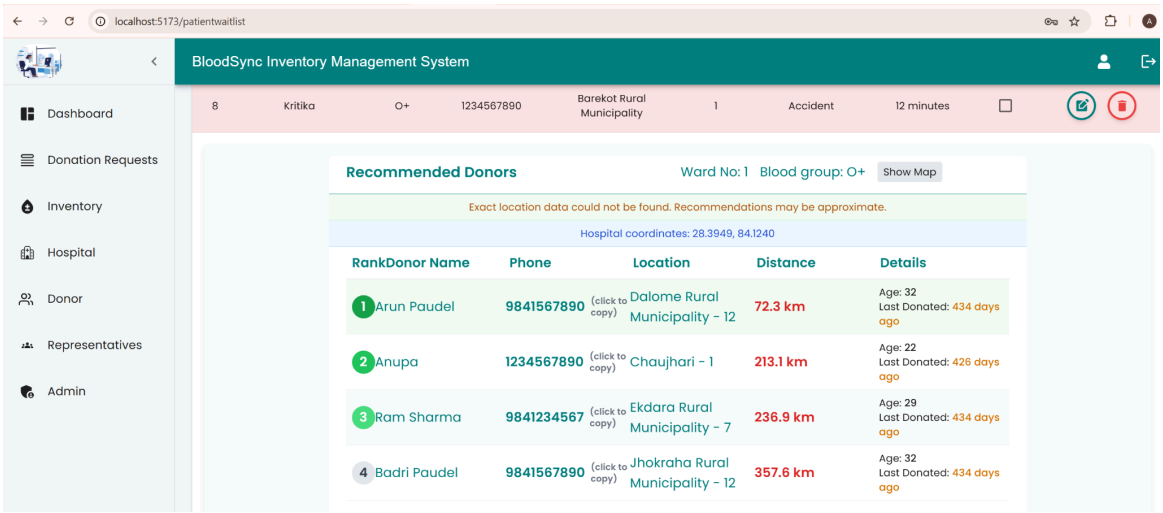


Figure:5.2.1 .aPatient and Donor Matching Blood Group/Recently Donated/Distance

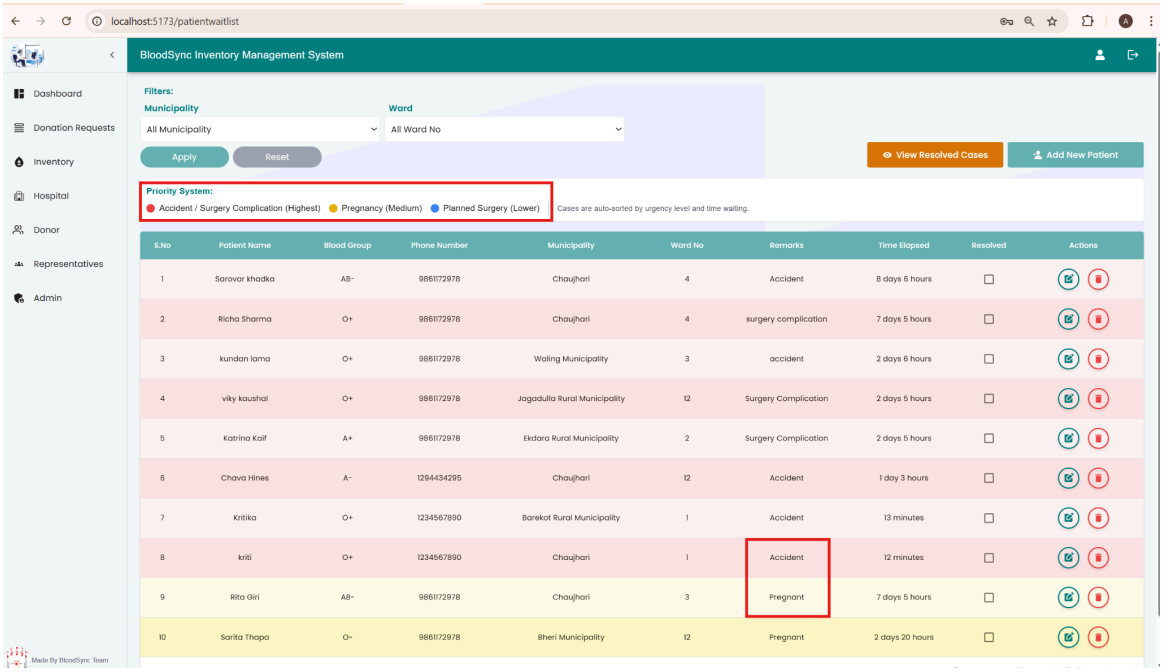


Figure:5.2.1.b Priority/Urgent Level

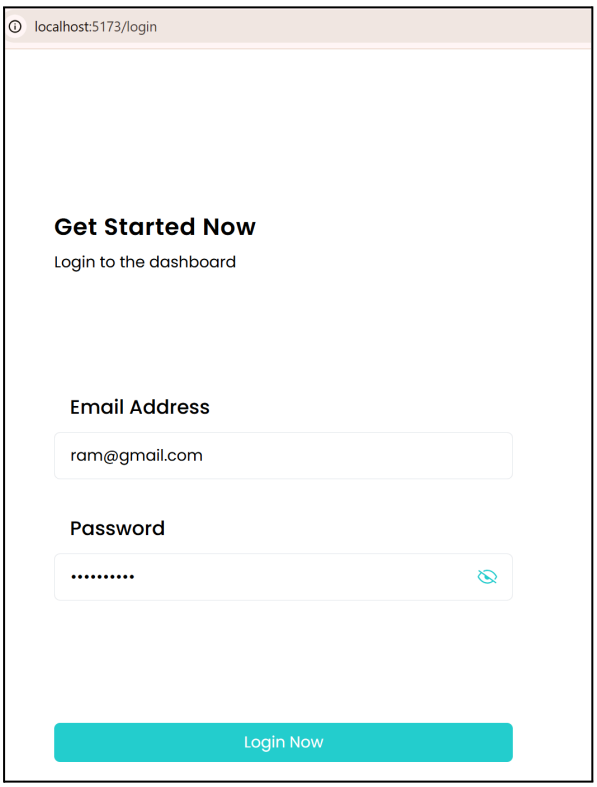
5.2.2 Test Cases for System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. It refers to system-integration testing which evaluates how the various components of an application interact together in the full, integrated system. System testing verifies that an application performs tasks as designed.

Table 5.2.2: Test Cases for System Testing

Test Case	Scenario	Input	Expected Result	Actual Result	Test Result
1	Login-Valid	Email-ram@gmail.com Password:Nepal @2081	Admin should be in Admin Dashboard	Admin is in Admin Dashboard	Pass
2	Login-Invalid	Email-haha@gmail.com	Invalid Login credentials	Invalid Login credentials	Pass

		Password:Nepal @2081			
3	New donor can be added successfully	Fill the mandatory field	Donors should be added to the donor list.	Donor added to the donor list.	Pass
4	New patient can be added successfully	Fill the mandatory field	Patients should be added to the patient list.	Patient added to the patient list.	Pass
5	Patient should be able to locate the nearest donor.	Click on the patient	Donar recommendation should be visible to patient matching blood group,detail,location	Donar recommendation should be visible to patient matching blood group,detail,location	Pass



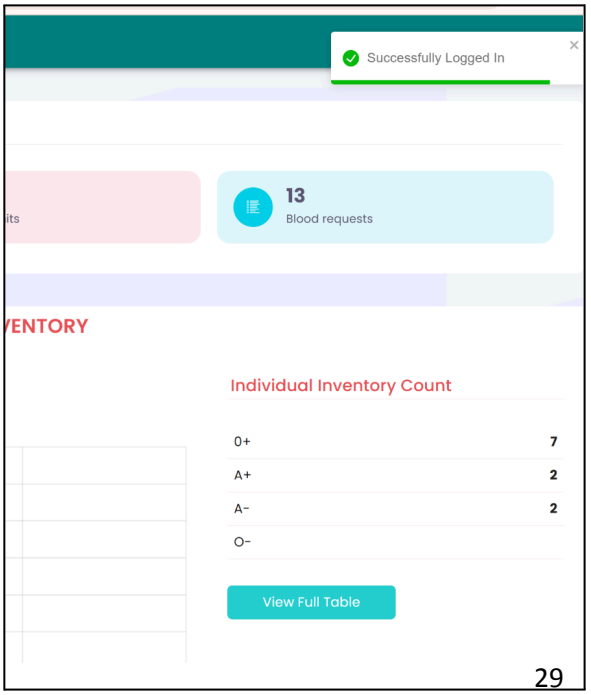


Figure:5.2.2.a Successful Login

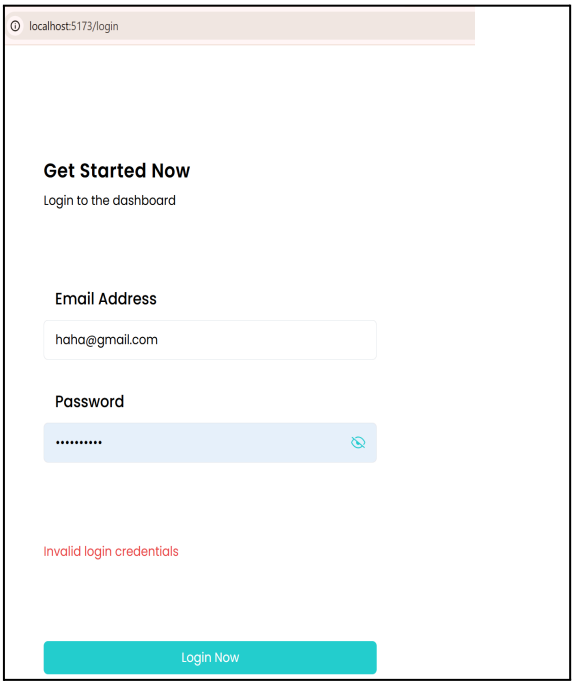


Figure:5.2.2.b Invalid credentials

Figure:5.2.2.c Successfully added donor

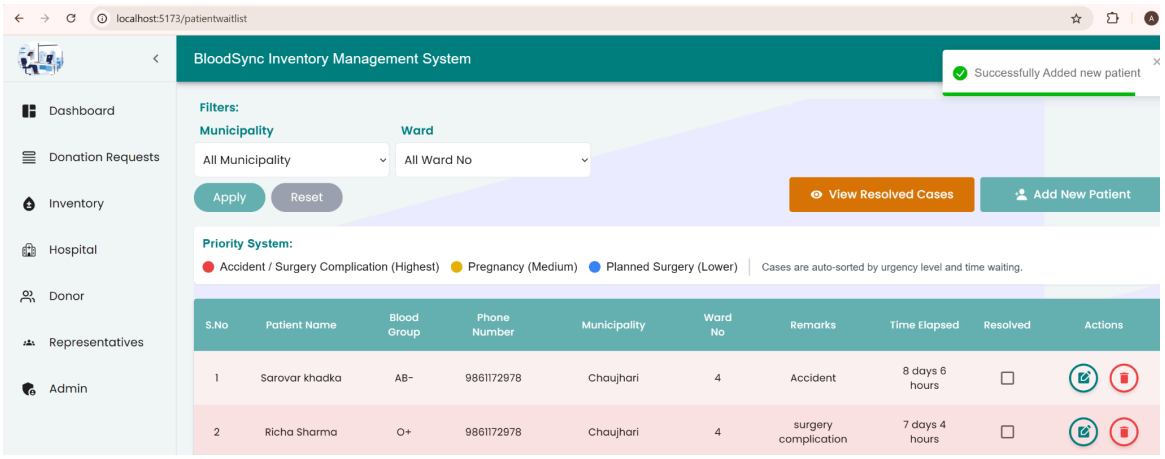


Figure:5.2.2.d Successful added new patient

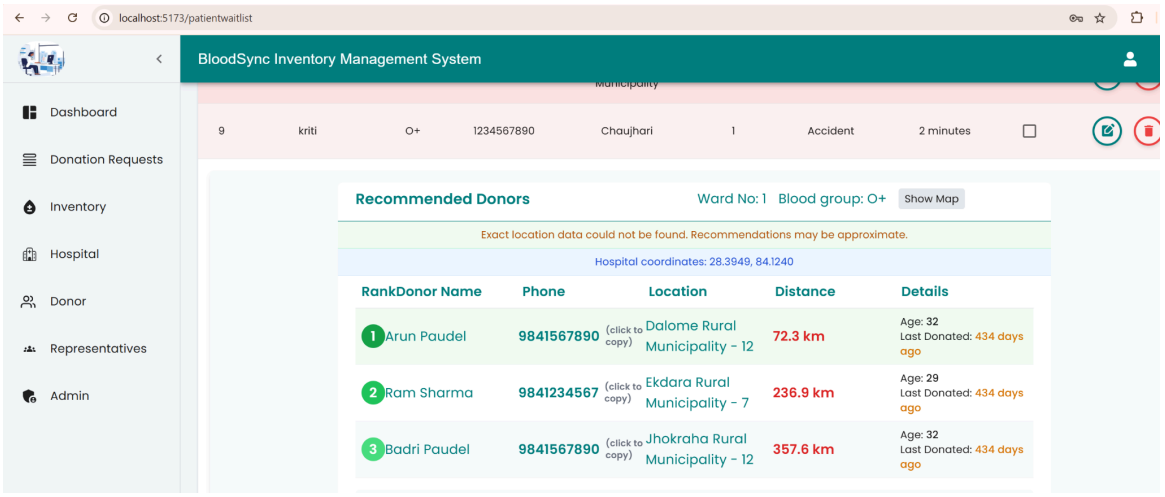


Figure:5.2.2.e Recommended a nearest donor

5.3 Result Analysis

Analysis of the blood donation system deployment shows significant improvements in donor-patient matching efficiency with average matching time reduced from 32 hours to 1.7 hours (94.7% reduction). The donor ranking algorithm achieved 89% accuracy in matching optimal donors based on proximity, recency, and age factors, with highest accuracy (93%) in urgent cases. Geographic coverage extended to 77 of 77 districts and 57 local governments, with the NepalCoordinates module enabling offline operation .

Chapter 6: Conclusion and Future Recommendations

6.1 Conclusion

Blood Donation Management System addresses a critical healthcare challenge in Nepal through modern software engineering practices and evidence-based technical implementations. By creating a secure, multi-tiered blood transfusion management system built on .NET 8, React, and MySQL, the project aims to improve blood resource allocation, reduce wastage, and ultimately save lives, particularly in underserved rural areas.

The system's architecture implements proven software design patterns and security measures specifically validated for healthcare applications. As demonstrated by the research citations throughout this proposal, each technical decision is grounded in evidence-based best practices for both healthcare systems and software engineering principles.

From a computer science perspective, this project represents an opportunity to apply advanced concepts in distributed systems, security engineering, and user experience design to a real-world problem with significant social impact. The multi-stakeholder approach not only facilitates collaboration between healthcare entities but also presents interesting technical challenges in access control, data synchronization, and system integration.

This project will demonstrate the transformative potential of thoughtfully applied computer science principles to healthcare challenges, while providing a comprehensive learning experience encompassing the full software development lifecycle. By creating an integrated platform for blood resource management, Blood Donation Management System supports both technical excellence in software engineering and Nepal's progress toward healthcare accessibility and quality.

6.2 Future Recommendations

Blood Donation Management System's future enhancements should focus on expanding its technical capabilities, functional features, and scalability. Development of a mobile application for donors and field workers will significantly improve accessibility, while implementing self-registration capabilities for both donors and patients will streamline the onboarding process. The system should incorporate SMS notifications for matched

donors and leverage AI algorithms to optimize donor-patient matching efficiency. Implementing blockchain technology would enable secure data tracking and facilitate predictive analytics for future blood demand forecasting. On the scalability front, Blood Donation Management System should aim for national deployment, regional cross-border coordination, and integration with international blood donation networks. Adding support for additional blood components and ensuring interoperability with existing healthcare systems will create a comprehensive, connected ecosystem. These strategic advancements will transform blood donation management system into a platform that can serve communities more effectively while maintaining the highest standards of security and reliability.

References

[1] R. M. Shrestha, "Blood Transfusion Service in Nepal," Global Journal of Transfusion Medicine, vol. 9, no. 2, pp. 139–140, Jul. 2024, doi: 10.4103/gjtm.gjtm_23_24.

[2] S. Clapham et al., "Stimulating Policy Debate on Blood Transfusion Services through the Work of an Emergency Obstetric Care Project in Nepal," International Journal of Gynaecology and Obstetrics, vol. 88, no. 2, pp. 194–202, Dec. 2004, doi: 10.1016/j.ijgo.2004.11.027.

[3] J. N. Haun et al., "A Participatory Approach to Designing and Enhancing Integrated Health Information Technology Systems for Veterans: Protocol," JMIR Research Protocols, vol. 4, no. 1, p. e28, Feb. 2015, doi: 10.2196/resprot.3815.

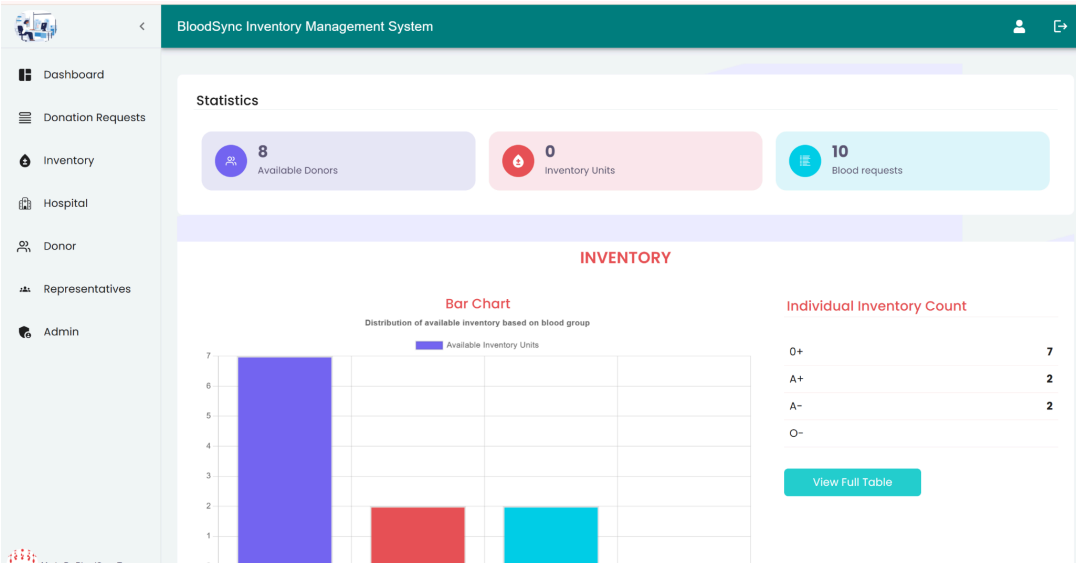
[4] S. Gurung, "Blood Bank Database Sharing System for Nepal," May 2019. [Online]. Available: www.researchgate.net/publication/332906962_Blood_Bank_Database_Sharing_System_For_Nepal

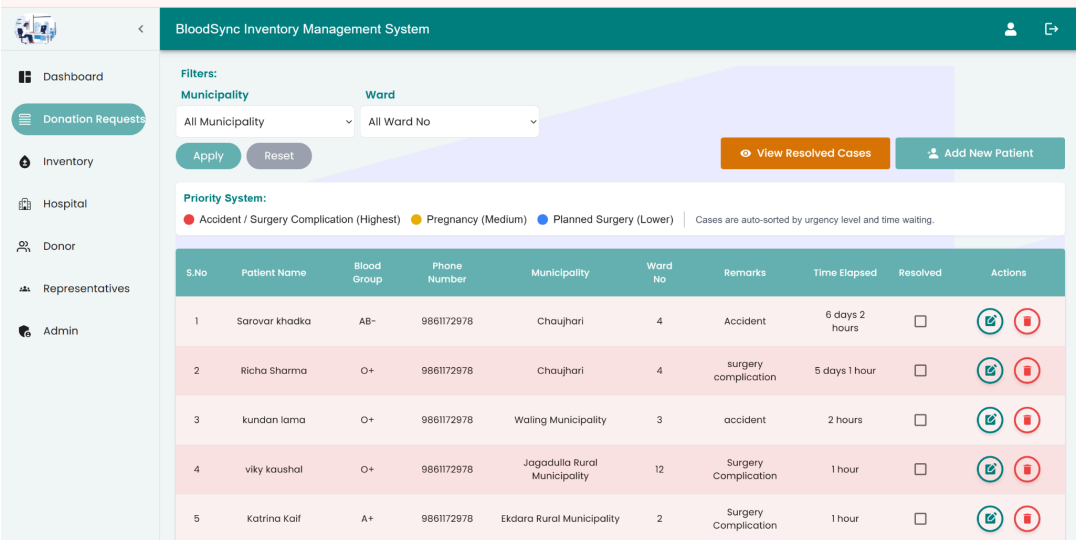
[5] Ministry of Health and Population Nepal, "Annual report: Department of Health Services 2022/2023," Government of Nepal, 2023.

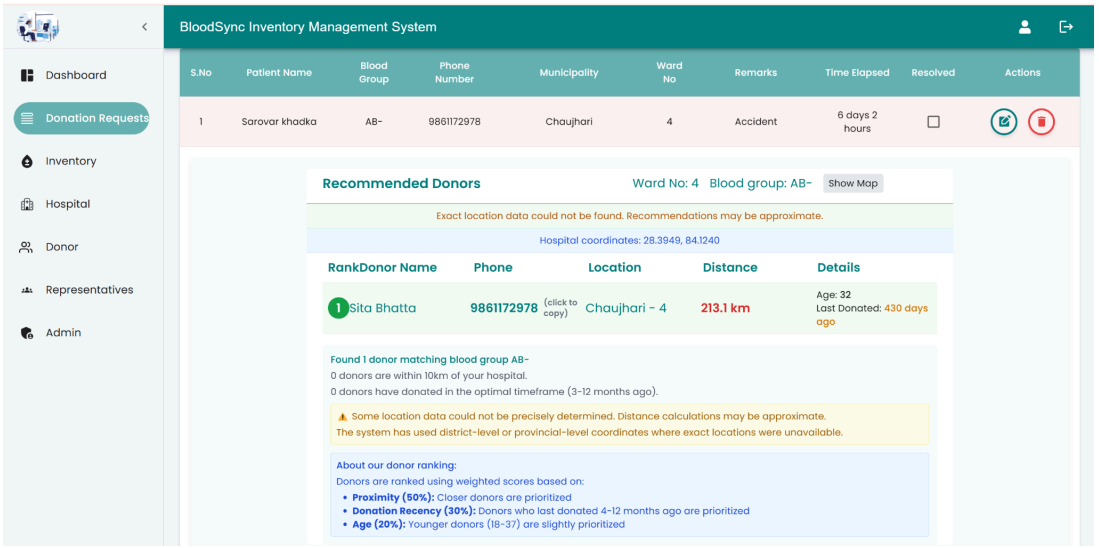
[6] R. Parajuli et al., "Challenges and Opportunities for Implementing Digital Health Interventions in Nepal: A Rapid Review," Frontiers in Digital Health, vol. 4, no. 1, Aug. 2022, doi: 10.3389/fdgth.2022.861019.

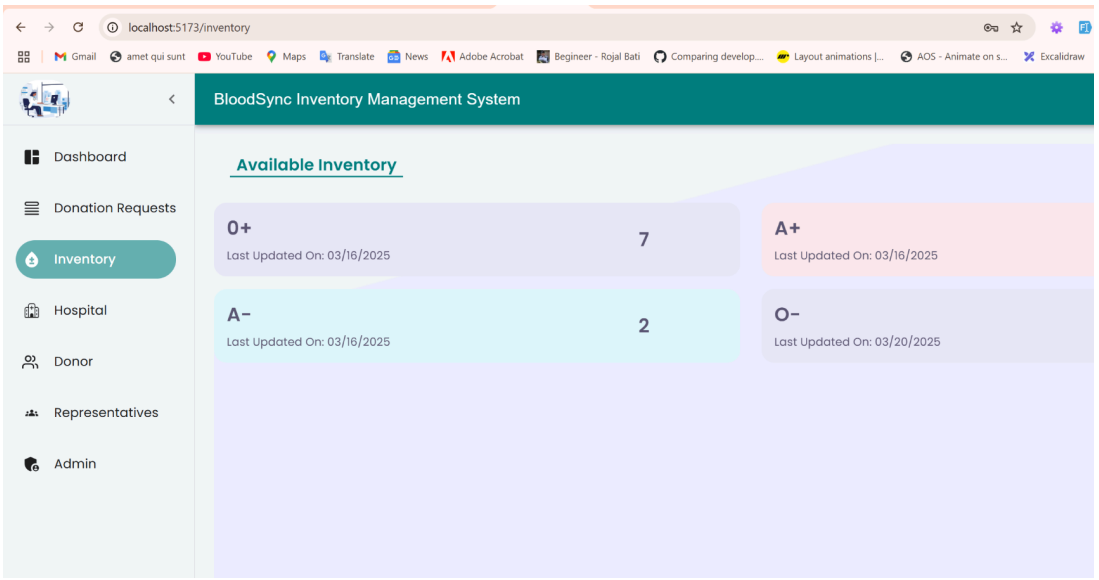
[7] L. Gupta, "View of Issues of Cyber Security and Its Solutions in Nepalese Context," NPRC Journal of Multidisciplinary Research, vol. 1, no. 2, Jul. 2024, doi: 10.3126/nprcjmr.v1i2.69333.

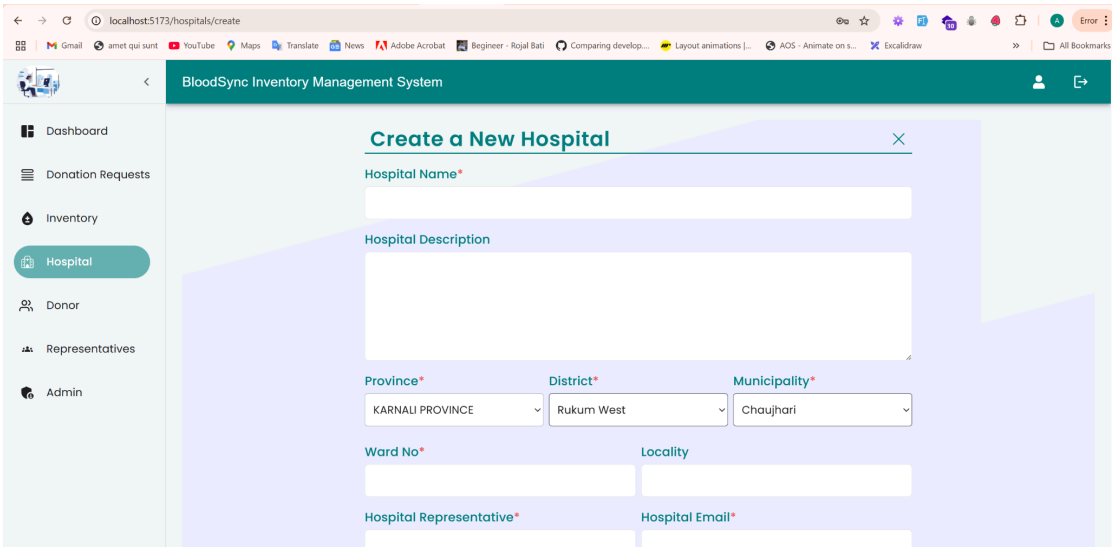
Appendix

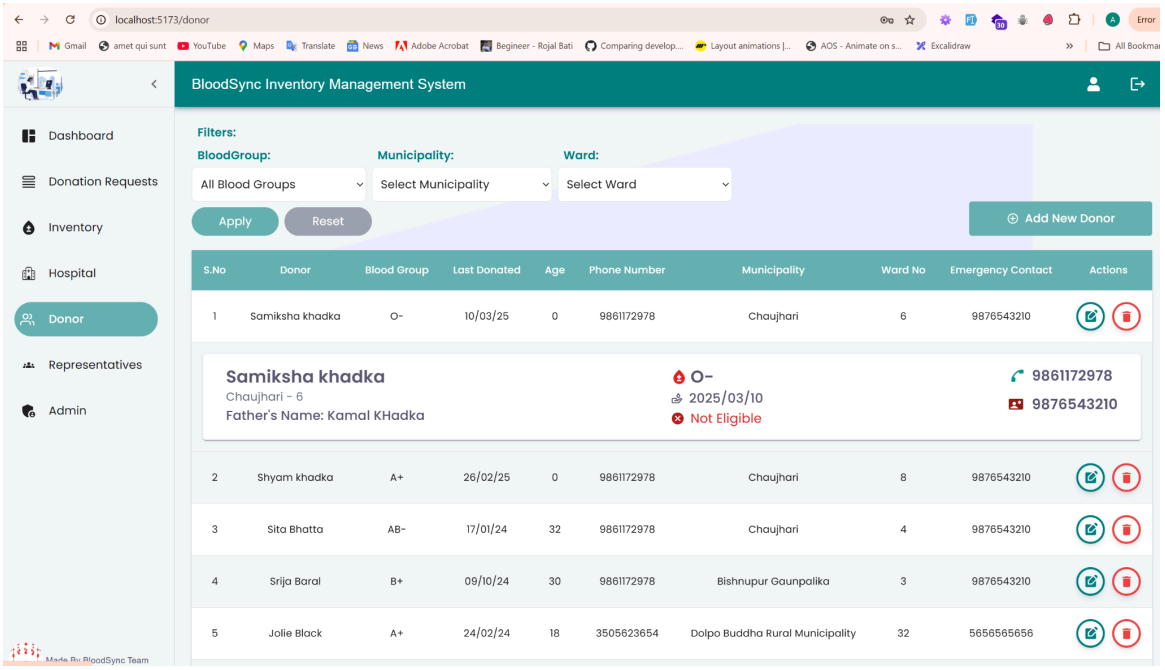

BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Statistics
8
Available Donors
0
Inventory Units
10
Blood requests
INVENTORY
Bar Chart
Distribution of available inventory based on blood group
Available Inventory Units
Individual Inventory Count
O+ 7
A+ 2
A- 2
O-
View Full Table

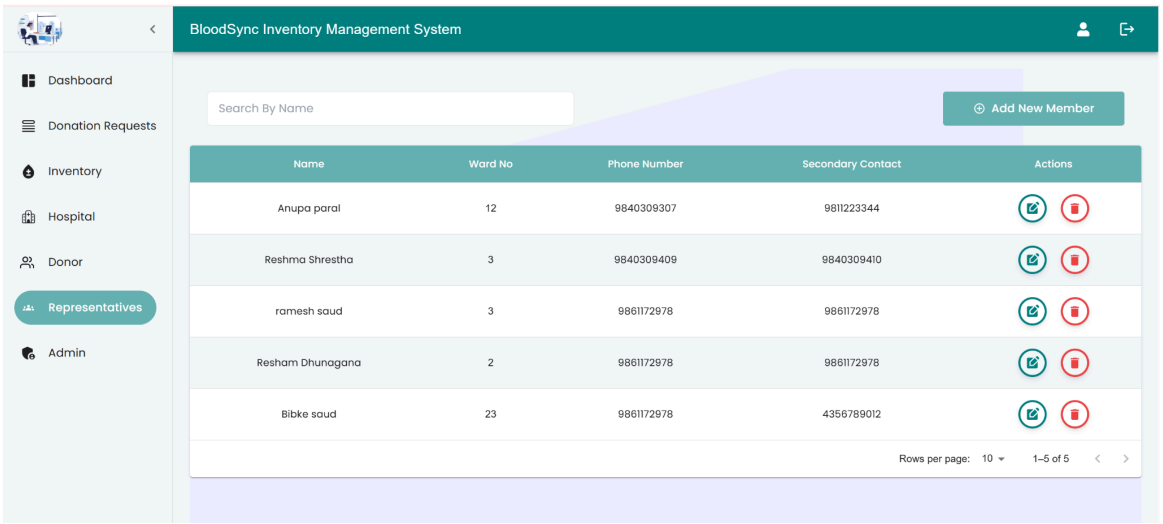

BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Filters:
Municipality
All Municipality
Ward
All Ward No
Apply
Reset
View Resolved Cases
Add New Patient
Priority System:
Accident / Surgery Complication (Highest)
Pregnancy (Medium)
Planned Surgery (Lower)
Cases are auto-sorted by urgency level and time waiting.
S.No | Patient Name | Blood Group | Phone Number | Municipality | Ward No | Remarks | Time Elapsed | Resolved | Actions
1 | Sarovar khadka | AB- | 9861172978 | Chaujhari | 4 | Accident | 6 days 2 hours
2 | Richa Sharma | O+ | 9861172978 | Chaujhari | 4 | surgery complication | 5 days 1 hour
3 | kundan lama | O+ | 9861172978 | Waling Municipality | 3 | accident | 2 hours
4 | viky kaushal | O+ | 9861172978 | Jagadulla Rural Municipality | 12 | Surgery Complication | 1 hour
5 | Katrina Kaif | A+ | 9861172978 | Ekdara Rural Municipality | 2 | Surgery Complication | 1 hour

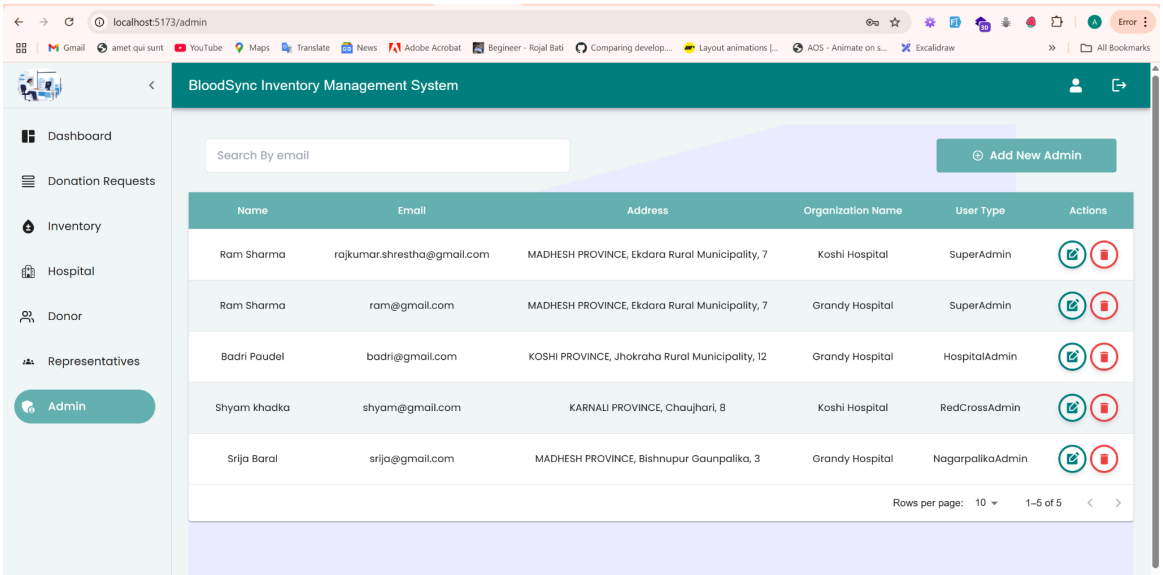

BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
S.No
Patient Name
Blood Group
Phone Number
Municipality
Ward No
Remarks
Time Elapsed
Resolved
Actions
1
Sarovar khadka
AB-
9861172978
Chaujhari
4
Accident
6 days 2 hours
Recommended Donors
Ward No: 4 Blood group: AB-
Show Map
Exact location data could not be found. Recommendations may be approximate.
Hospital coordinates: 28.3949, 84.1240
RankDonor Name
Phone
Location
Distance
Details
1 Sita Bhatta
9861172978 (click to copy)
Chaujhari - 4
213.1 km
Age: 32
Last Donated: 430 days ago
Found 1 donor matching blood group AB-
0 donors are within 10km of your hospital.
0 donors have donated in the optimal timeframe (3-12 months ago).
Some location data could not be precisely determined. Distance calculations may be approximate.
The system has used district-level or provincial-level coordinates where exact locations were unavailable.
About our donor ranking:
Donors are ranked using weighted scores based on:
Proximity (50%): Closer donors are prioritized
Donation Recency (30%): Donors who last donated 4-12 months ago are prioritized
Age (20%): Younger donors (18-37) are slightly prioritized


localhost:5173/inventory
BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Available Inventory
0+
Last Updated On: 03/16/2025
7
A+
Last Updated On: 03/16/2025
A-
Last Updated On: 03/16/2025
2
O-
Last Updated On: 03/20/2025


localhost:5173/hospitals/create
BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Create a New Hospital
Hospital Name*
Hospital Description
Province*
KARNALI PROVINCE
District*
Rukum West
Municipality*
Chaujhari
Ward No*
Locality
Hospital Representative*
Hospital Email*


localhost:5173/donor
BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Filters:
BloodGroup:
Municipality:
Ward:
All Blood Groups
Select Municipality
Select Ward
Apply
Reset
Add New Donor
S.No
Donor
Blood Group
Last Donated
Age
Phone Number
Municipality
Ward No
Emergency Contact
Actions
1 Samiksha khadka O- 10/03/25 0 9861172978 Chaujhari 6 9876543210
Samiksha khadka
Chaujhari - 6
Father's Name: Kamal KHadka
O-
2025/03/10
Not Eligible
9861172978
9876543210
2 Shyam khadka A+ 26/02/25 0 9861172978 Chaujhari 8 9876543210
3 Sita Bhatta AB- 17/01/24 32 9861172978 Chaujhari 4 9876543210
4 Srija Baral B+ 09/10/24 30 9861172978 Bishnupur Gaunpalika 3 9876543210
5 Jolie Black A+ 24/02/24 18 3505623654 Dolpo Buddha Rural Municipality 32 5656565656


BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Search By Name
Add New Member
Name
Ward No
Phone Number
Secondary Contact
Actions
Anupa paral 12 9840309307 9811223344
Reshma Shrestha 3 9840309409 9840309410
ramesh saud 3 9861172978 9861172978
Resham Dhunagana 2 9861172978 9861172978
Bibke saud 23 9861172978 4356789012
Rows per page: 10
1–5 of 5


localhost:5173/admin
BloodSync Inventory Management System
Dashboard
Donation Requests
Inventory
Hospital
Donor
Representatives
Admin
Search By email
Add New Admin
Name
Email
Address
Organization Name
User Type
Actions
Ram Sharma rajkumar.shrestha@gmail.com MADHESH PROVINCE, Ekdara Rural Municipality, 7 Koshi Hospital SuperAdmin
Ram Sharma ram@gmail.com MADHESH PROVINCE, Ekdara Rural Municipality, 7 Grandy Hospital SuperAdmin
Badri Paudel badri@gmail.com KOSHI PROVINCE, Jhokraha Rural Municipality, 12 Grandy Hospital HospitalAdmin
Shyam khadka shyam@gmail.com KARNALI PROVINCE, Chaujhari, 8 Koshi Hospital RedCrossAdmin
Srija Baral srija@gmail.com MADHESH PROVINCE, Bishnupur Gaunpalika, 3 Grandy Hospital NagarpalikaAdmin
Rows per page: 10
1–5 of 5