DONORS

MANAGEMENT SYSTEM

SCOPE OF WORK DOCUMENT

Group Members

Waleed Ghani (0123-0082)

Areeba Waheed (0123-0031)

TABLE OF CONTENT

ACKNOWLEDGMENT

First and foremost, I am thankful to Almighty Allah for giving me the strength, knowledge, and perseverance to complete this project successfully.

I would like to express my sincere gratitude to **Ms. Alisha Farman**, my respected teacher at **Iqra University, Airport Campus**, for her continuous guidance, support, and encouragement throughout the completion of this project. Her valuable insights and feedback have been instrumental in shaping this work. I am also thankful to my university for providing me with the resources and learning environment that enabled me to complete this project successfully. Finally, I would like to acknowledge the efforts of my team members and all those who directly or indirectly contributed to the development of this project.

INTRODUCTION

Blood donation drives are a crucial lifeline for patients suffering from conditions such as thalassemia, who require regular and timely blood transfusions to survive. However, these drives often face significant challenges, including inefficient donor record-keeping, difficulty in identifying eligible donors, and delays in connecting the right donor with the right patient at the right time. Traditional methods of managing donor data are fragmented, time-consuming, and prone to errors, which can ultimately put patients' lives at risk.

To overcome these challenges, the Donor Management System (DMS) has been developed as a modern, web-based platform that streamlines and digitizes the entire process within universities. The system consolidates student donor records into a secure database, storing personal information, medical and donation history, and automatically calculating donor eligibility based on health parameters and medical guidelines.

What sets DMS apart is its AI-powered recommendation engine. This intelligent feature analyzes stored data and patient requirements to identify the most suitable donors, prioritize them by urgency and medical suitability, and even predict future donation dates using historical health trends.

By bridging the gap between patients' urgent needs and donors' availability, the DMS not only improves the efficiency of blood donation drives but also enhances donor participation. Through personalized eligibility updates, educational resources, and reminders, it nurtures a culture of regular, voluntary blood donation within the university community, ultimately making a lasting impact on healthcare accessibility and patient well-being.

1.1. Problem Statement:

In this project, the university organizes weekly blood donation drives for thalassemia patients, but many students avoid participation by falsely claiming prior donations or medical ineligibility (such as low BMI). This causes inefficiencies, delays in finding suitable donors, and potential shortages of required blood types, wasting volunteer time in verifying eligibility and making it harder to respond to urgent patient needs.

1.2. Project Objective:

The main objective of the Donor Management System is to centralize donor data, verify eligibility, and enable AI-powered donor matching to improve the efficiency and effectiveness of blood donation drives within universities.

1.2.1. Proposed Solution:

The proposed solution is to build a secure, web-based platform that stores student donor records, verifies medical eligibility, and uses AI to recommend the best matching donors for urgent blood requests.

1.3. Scope:

1.3.1. Requirement Gathering:

1.3.1.1. Functional Requirements:

1. **Authentication System:**
    - **Admin Login:** Secure login for university staff and blood drive organizers.
    - **Student Registration & Login:** Students can create accounts by providing their email, student ID, and personal details.
    - **Role-Based Access Control:**
        i. **Admin:** Full access to manage donors, approve eligibility, and run AI matching.
        ii. **Student:** Can view their eligibility status, donation history, and receive notifications.
2. **User Profile Management:**
    - Students can update their personal details, BMI, and other health parameters.
    - Admins can verify and approve student-submitted data.

3. **Eligibility Tracking:**
    - Automatic calculation of eligibility based on last donation date, BMI, and medical rules.
    - Visual indicators (Eligible / Not Eligible / Eligible Soon).
4. **AI-Powered Donor Matching**
    - Recommend best-matching eligible donors for specific patient blood type needs.
    - Rank donors by suitability, urgency, and donation history.
5. **Search & Filter:**
    - Search donors by name, student ID, department, blood type, and eligibility.
    - Filters for quick access to matching donors for urgent cases.
6. **Donation History Tracking:**
    - Maintain a complete log of each student's past donations.
7. **Reports & Analytics:**
    - Generate reports for blood type availability, donation trends, and eligibility statistics.

1.3.1.2. Non-Functional Requirements:

2. **Empathy Factor:**
   The system should send motivational and empathetic notifications (e.g., thanking donors, sharing impact stories of how their donation helped) to encourage long-term engagement and voluntary donations.

3. **Green Computing:**
   The system should optimize server usage and resource consumption to reduce its carbon footprint, ensuring eco-friendly digital operations.

4. **Personalization:**
   The interface should adapt based on user roles and preferences (e.g., administrators see detailed analytics, while student donors receive simplified dashboards with donation reminders).

5. **Gamification:**
   The platform should include gamified elements such as achievement badges, donation streaks, or leaderboards to promote healthy competition and encourage frequent donations.

6. **Knowledge Enhancement:**
   The system should provide donors with educational content (e.g., health tips, importance of maintaining hemoglobin levels) to improve awareness and well-being.

1.3.1.3. Technical Requirements:

1.3.1.4. Hardware Requirements:

Category	Requirement	Details
Server Infrastructure	Application Servers	High-performance servers for handling donor record requests and running the application backend.
	Database Servers	Dedicated servers to manage and store donor data, medical records, and eligibility information.
	Backup Servers	Servers for automated backups and disaster recovery to ensure data safety.
Networking	Firewall	For securing donor data and protecting against unauthorized access.
	Load Balancer	To distribute traffic efficiently across servers, ensuring smooth performance during peak usage.
User Access Devices	Development Machines	High-performance computers for developers working on backend, frontend, and AI modules.
	Testing Devices	Smartphones, tablets, and desktops for testing cross-platform compatibility of the system.

Table 2.0

1.3.1.5. Software Requirements:

Category	Requirement	Details
Frontend Technologies	Web Framework	React.js or Angular for building an interactive and user-friendly interface.
	HTML/CSS	For structuring and styling the donor management web application.
	JavaScript	For dynamic features such as dashboards, search filters, and donor notifications.
Backend Technologies	Programming Language	PHP for backend development.
	Framework	Laravel framework for server-side logic, API development, and database handling.
	APIs	RESTful APIs for secure and efficient communication between frontend and backend.
Database Management	Database System	MySQL or PostgreSQL for managing donor records, eligibility status, and donation history.
	ORM Tool	Eloquent ORM (Laravel's built-in ORM) for efficient database interaction.
Artificial Intelligence	Recommendation Engine	AI-based module to identify eligible donors, predict next donation dates, and match patients with suitable donors.
	Libraries/Tools	Tensor Flow/Keras for prediction models; Pandas & Scikit-learn for data analysis.

Table 2.1

1.3.2. Budget:

The estimated budget for the Donor Management System project is divided into the following categories:

Category	Percentage	Amount	Description
Development	37%	185,000	Covers coding, integration of backend and frontend modules
Website	15%	75,000	Includes designing the responsive user interface, user experience optimization, and front-end development.
AI Integration	15%	75,000	Implementation of AI-powered donor matching system using Tensor Flow, along with training/testing of AI models.
Infrastructure	10%	50,000	Server hosting, cloud storage, domain registration, and hardware resources for smooth system operation.
Testing & QA	8%	40,000	Quality assurance, bug fixing, system testing (unit, integration, and user acceptance testing).
Marketing & Launch	10%	50,000	Awareness campaigns, promotional materials, launch events, and communication to increase system adoption.
Miscellaneous Expenses	5%	25,000	Covers unexpected costs such as maintenance, small-scale updates, and resource adjustments.

Table 2.2

1.3.3. Team Member Roles & Tasks:

1.3.3.1. Areeba Waheed (0123-0031): Project Manager

Responsibilities:

- Oversees project progress, manages timelines, coordinates with stakeholders, and ensures the project meets its objectives.

1.3.3.2. Waleed Ghani (0123-0082): UI Designer

Responsibilities:

- Designs wireframes and mock-ups, creates an intuitive user interface, and ensures the system is user-friendly.

1.3.3.3. Waleed Ghani (0123-0082): Developer

Responsibilities:

- Implements the backend and frontend functionalities, integrates the database, and develops the AI matching engine.

1.3.3.4. Areeba Waheed (0123-0031): QA Tester

Responsibilities:

- Conducts functional and non-functional testing, identifies bugs, ensures data security, and validates system reliability.

Methodology:

This project will follow the Agile Scrum methodology for iterative and incremental development. The approach divides the development into sprints, enabling continuous feedback, adaptation to changes, and faster delivery of functional components.

2.1. Main Features Table

Feature	Description
Authentication	Provides a secure login mechanism for both admins and students, ensuring that only authorized users can access the system.
Role-Based Access Control	Assigns specific permissions based on user roles. Admins can manage donor records, run analytics, and oversee system operations, while students can update personal profiles, track their donation history, and receive notifications.
Student Registration & Profile Management	Allows students to create and manage personal profiles containing details such as contact information, department, blood type, BMI, and health indicators.

Eligibility Tracking	Automatically evaluates donor eligibility by analyzing health data (e.g., BMI, hemoglobin levels) and last donation dates, ensuring compliance with medical guidelines.
AI Donor Matching	Uses artificial intelligence to recommend the most suitable donors for patients based on blood type, health parameters, and availability, while prioritizing urgent cases.
Donation History Tracking	Records and displays each student's past donations, including date, blood type, and health metrics, providing transparency and continuity for donors and administrators.
Reports & Analytics	Generates detailed reports and insights on donation trends, donor availability, and overall contribution rates, helping administrators make informed decisions.
Notifications & Alerts	Sends automated reminders to donors about upcoming eligibility, urgent blood requests, and general awareness campaigns to boost participation.
Data Security & Encryption	Ensures that sensitive donor and patient information is protected through encryption, authentication, and secure storage protocols.
Responsive UI/UX Design	Delivers a clean, user-friendly interface accessible across devices (desktop, tablet, and mobile), ensuring ease of use for both admins and students.
Cloud Hosting & Scalability	Utilizes cloud infrastructure to host the system, providing high availability, reliability, and the ability to scale as the number of donors increases.

Table 2.3

2.2. Why Use Scrum?

Scrum is used because it ensures iterative development, regular feedback, adaptability to changes, and faster delivery of working features.

2.2.1. Scrum Artifacts:

Scrum **artifacts** are the key documents or deliverables in a Scrum project that give **transparency** and help track progress. They basically answer *"What are we building? What are we working on now? What have we completed*?"

- Product Backlog
- Sprint Backlog

2.2.1.1. Product Backlog:

The Product Backlog is a prioritized list of all features, enhancements, and requirements for the Donor Management System. It serves as the single source of work for the development team and evolves throughout the project as new needs are identified. Each item is described briefly with its feature, purpose, and priority to guide sprint planning.

Features	Description	Priority
User Authentication	Secure login for Admins and Students	High
Role-Based Access Control	Different permissions for Admin and Student users	High
Student Registration	Students can create and manage personal profiles	High
Donor Eligibility Tracking	System auto-checks BMI, health, and last donation date	High
AI-Powered Donor Matching	Suggests best donors using AI model	High
Search & Filter Donors	Quick search by name, blood type, or eligibility	Medium
Donation History Tracking	Logs and displays past donation details	Medium
Reports & Analytics	Generates insights on donation trends, availability, and stats	Medium
Notifications & Alerts	Sends reminders, eligibility updates, and urgent requests	Medium
Data Security & Encryption	Ensures donor data privacy with encryption	High
Responsive UI/UX Design	User-friendly and mobile-friendly interface	High
Cloud Hosting Setup	Infrastructure setup for scalability and availability	Medium

Table 2.4

2.2.1.2. Sprint Backlog:

The **Sprint Backlog** is a subset of the product backlog that contains the features and tasks selected for development in a specific sprint. It provides a detailed view of **which features will be developed, their estimated duration, required effort, and current status**. The sprint backlog helps track progress and ensures the team meets its sprint goals within the defined timeline.

Sprint No	Features	Duration	EST Hrs.	Status
Sprint 1	• Setup project environment (IDE, framework, DB, version control) • Authentication UI (Login, Register forms with validations) • Role-based login (Student/Admin)	2 Weeks	25 hrs.	To Do
Meeting and Retrospective				
Sprint 2	• Donor Profile CRUD (Create, Read, Update, Delete) • Eligibility Check (BMI, Last Donation Date, Hemoglobin) • Show status: Eligible / Not Eligible / Eligible Soon	2 Weeks	30 hrs.	To Do
Meeting and Retrospective				
Sprint 3	• Search & Filter donors (by Name, ID, Blood Type) • Filter donors by eligibility status (Eligible, Not Eligible, Urgent Need) • Optimize listing for admin	1 Weeks	15 hrs.	To Do
Meeting and Retrospective				
Sprint 4	• AI Donor Matching module • Input form (Gender, Blood Type, Urgency, Required Date) • AI suggests best matching donors (ranked list) • Admin can view donor details from match results	2 Weeks	35 hrs.	To Do
Meeting and Retrospective				
Sprint 5	• Reports & Analytics (graphs, donation trends, eligibility stats) • Admin Dashboard (Total Donors, Eligible Donors, Urgent Requests, Upcoming Eligibility)	2 Weeks	25 hrs.	To Do

	• Notifications system (Admin alerts for new/updated profiles)			
Meeting and Retrospective				
Sprint 6	• Final Testing (Unit, Integration, Security) • Deployment on cloud server • User Acceptance Testing (UAT) • Documentation finalization	2 Weeks	20 hrs.	To Do

Table 2.5

The sprint plan covers **6 sprints with an estimated total duration of around 2 months and 3 weeks**, ensuring systematic development and timely delivery of the Donor Management System.

2.3. Design Process

2.3.1. Use Case Diagram:

A **Use Case Diagram** is a UML diagram that illustrates the functional requirements of a system by showing how different users (actors) interact with various system processes (use cases). It provides a high-level view of the system's functionality and the relationships between actors and use cases. As shown in Fig.2.1.

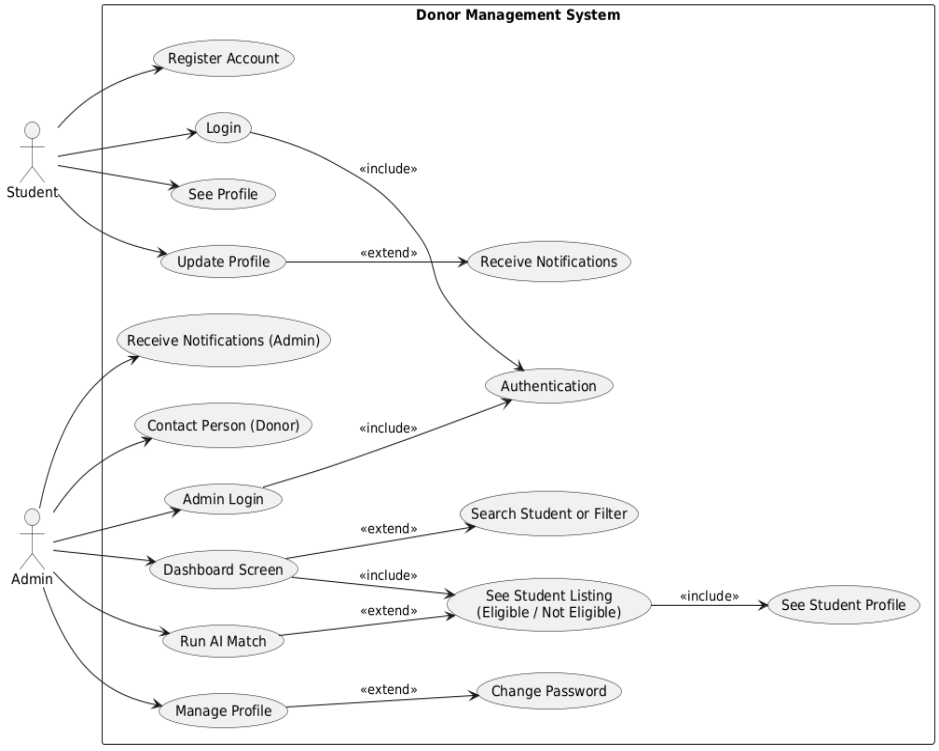


Fig.2.1use case diagram that elaborates relation of admin and student with different cases.

2.3.1.1. Student Use Cases and Relations:

1. **View Donation History:**
    a. The student can view their past donation records.
    b. Helps them track their donation activities.
2. **Receive Notifications:**
    a. Students receive alerts about updates (eligibility status, donor matches, admin approvals, etc.).
    b. This *extends* the **Update Personal Details** use case, since notifications may be triggered when personal details are updated.
3. **Update Health Details:**
    a. Students can update their medical or health records.
    b. This information is important for eligibility checks and donor matching.
4. **Login:**
    a. Students must log in to access the system.
    b. This use case *includes* **Authentication** (login always requires authentication).
5. **Register Account:**
    a. Students can create a new account.
    b. This use case *includes* **Provide Personal Details** (registration always requires personal details).
6. **Track Eligibility:**
    a. Students can track whether they are eligible to donate, based on their health details and admin approval.
    b. This use case *includes* **Donation History** (eligibility checking may rely on past donations).
7. **AI Donor Matching:**
    a. Students can use the system's AI to find suitable donor matches.
    b. This use case *includes* **Donation History** (donor matching also considers past donation history).

2.3.1.2. Admin Use Cases and Relations:

1. **Manage Donors Profiles:**
    a. Admins can manage donor accounts and details.
    b. This use case *extends* **Donation History** (managing profiles often requires reviewing history).
2. **Verify & Approve Data:**
    a. Admins review and approve health records or personal data submitted by students/donors.
    b. This use case *includes* **Search & Filter Donors** (to verify data, admins may need to search/filter donor information).

3. **Generate Reports & Analytics:**
    a. Admins can generate reports and perform analysis (e.g., donation trends, eligibility statistics).
    b. This relies on **Search & Filter Donors** to gather relevant data.

2.3.2. Sequence Diagram:

A **Sequence Diagram** is a UML diagram that shows how objects or components in a system interact with each other over time. It focuses on the order of messages exchanged between actors and the system, representing the flow of control step by step. This sequence shows the process of our application , As shown in the figure 2.2.

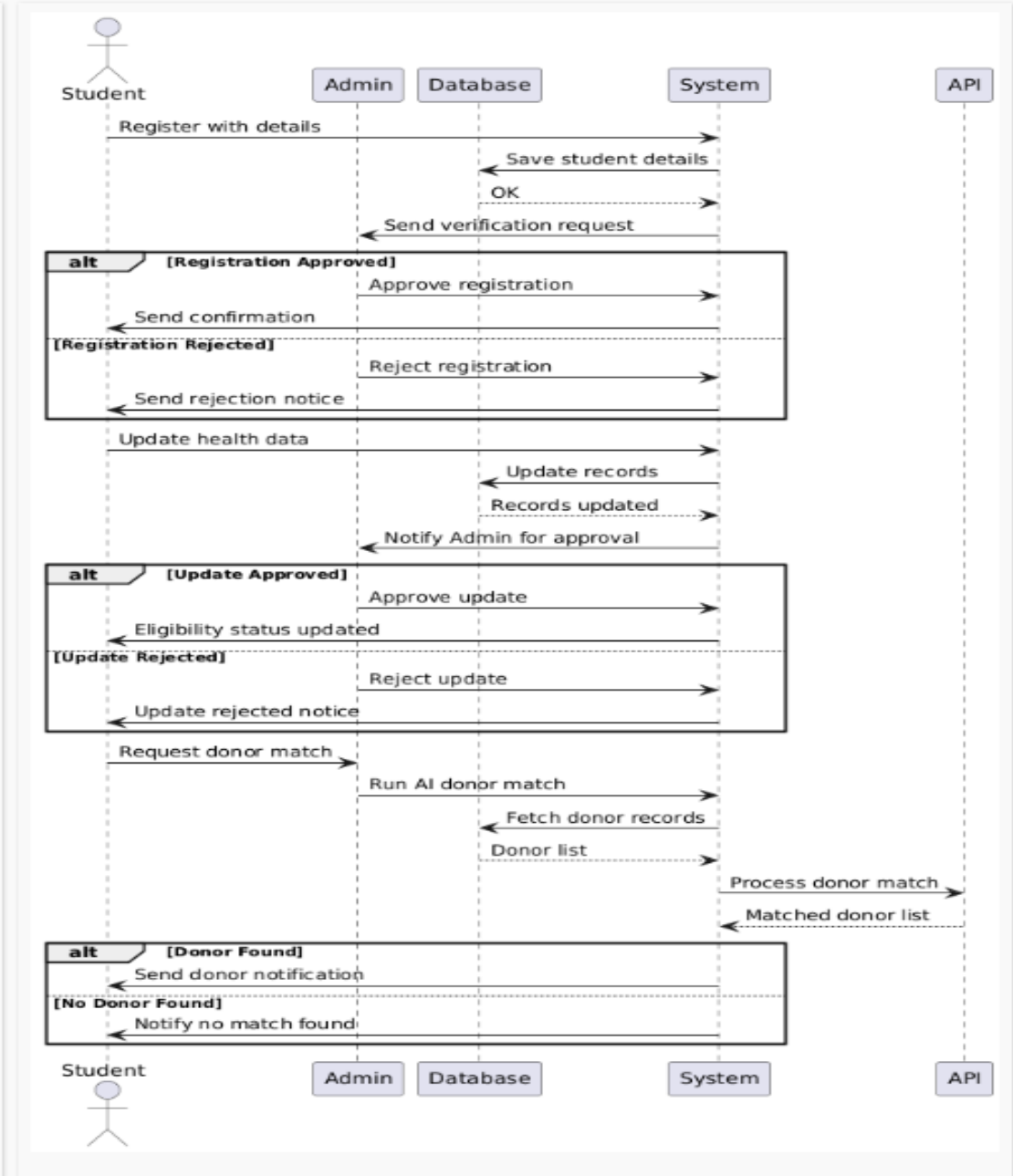
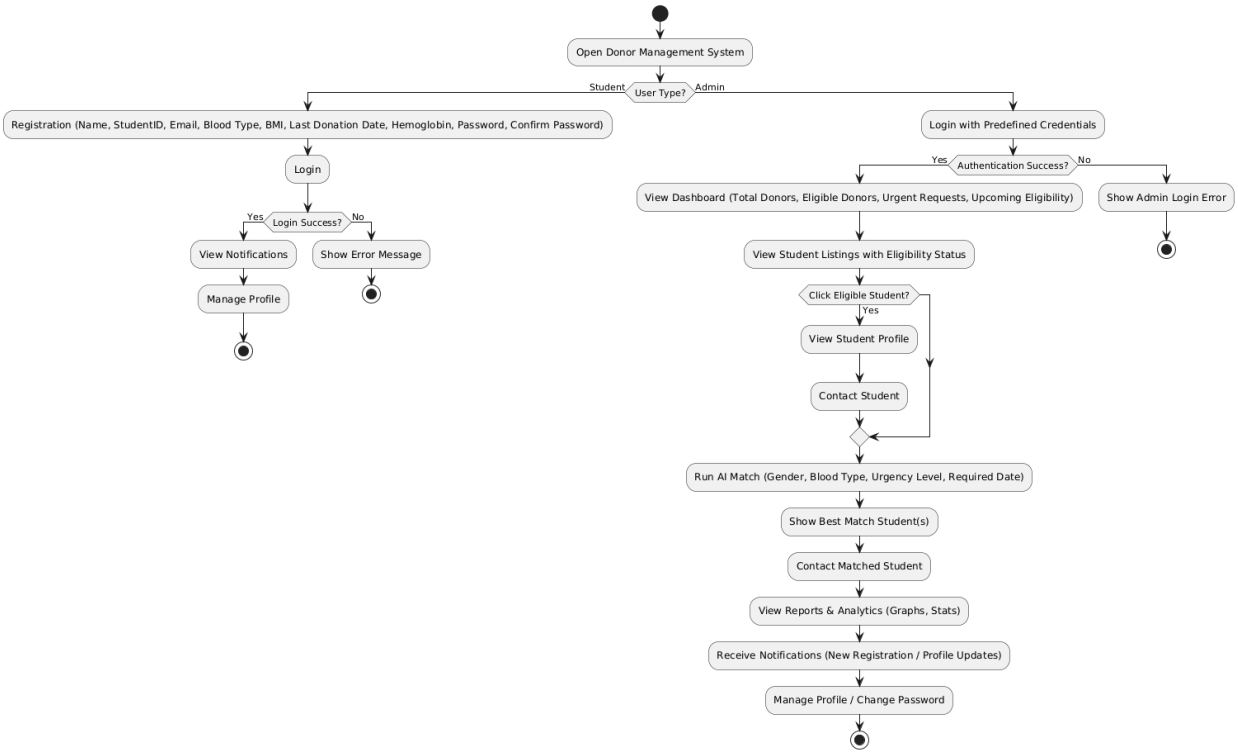


Fig: 2.2 visually represents how objects interact in a system.

2.3.3. Activity Diagram:

An **Activity Diagram** is a UML behavioral diagram that illustrates the dynamic aspects of a system by showing the flow of control or data from one activity to another. It represents the step-by-step execution of processes, decisions, and parallel activities, much like a flowchart. They can also describe the steps in diagram fig 2.3.



2.4. Prototype:

2.4.1. Low-Fidelity Prototype:

- Basic wireframes for Login Page, Registration Page, and Dashboard
- Simple layout sketches for Donor List and Profile pages
- Navigation flow diagram showing movement between modules

2.4.2. High-Fidelity Prototype:

The high-fidelity prototype will include:

- **Login Page:** Allows admins and students to securely sign in to the system.

- **Student Registration Page:** Enables new students to create an account and submit their personal and health details.
- **Admin Dashboard:** Displays key donor statistics, urgent requests, and quick access to core features.
- **Donor List Page:** Shows a searchable and filterable list of all donors with eligibility and AI recommendations.
- **Donor Profile Page:** Provides detailed personal, health, and donation history information for a specific donor.
- **AI Donor Matching Page:** Generates a ranked list of best-matching eligible donors for specific patient needs.
- **Student Dashboard:** Allows students to view their eligibility status, donation history, and received donation requests. Also students can update their details.

2.4.3. Screen 01 Landing Page:

- **Purpose:** The public-facing home page of the system where new visitors arrive.
- **Functionality:**
    - Introduce the Blood Donation Management System.
    - Explain the process and benefits of blood donation.
    - Encourage visitors to register or log in.
- **Elements:**
    - Hero section (image + headline encouraging donation).
    - "How it Works" section.
    - Testimonials from users/donors.
    - Footer with contact information, links, and social media.

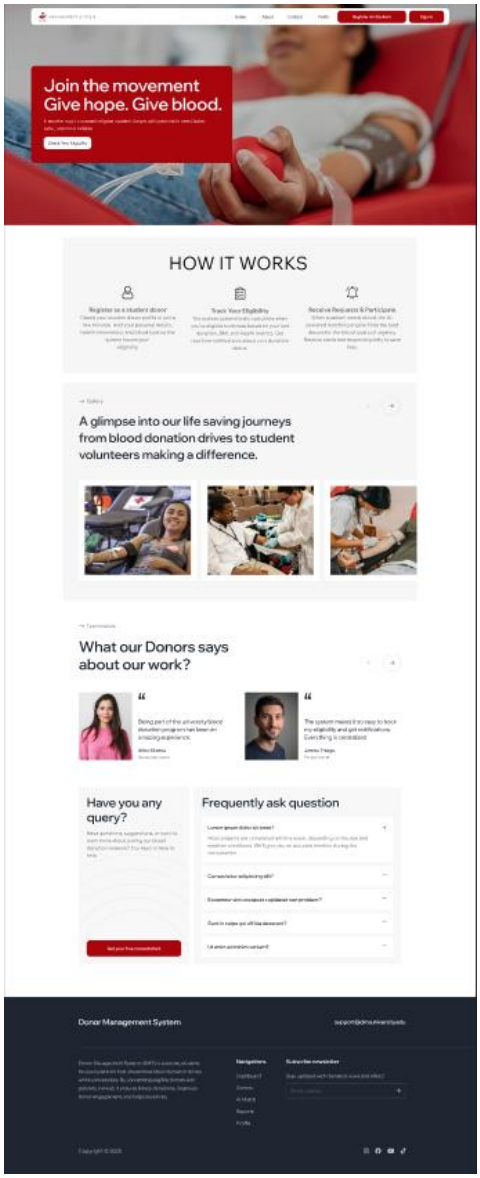


Figure 2.0

2.4.4. Screen 02 Login Page:

- **Purpose:** Entry point for registered users to access the system.
- **Functionality:**
  - Authenticate users using email/username and password.
- **Elements:**
  - Email/Username input.
  - Password input.
  - Login button.
  - Forgot password link.

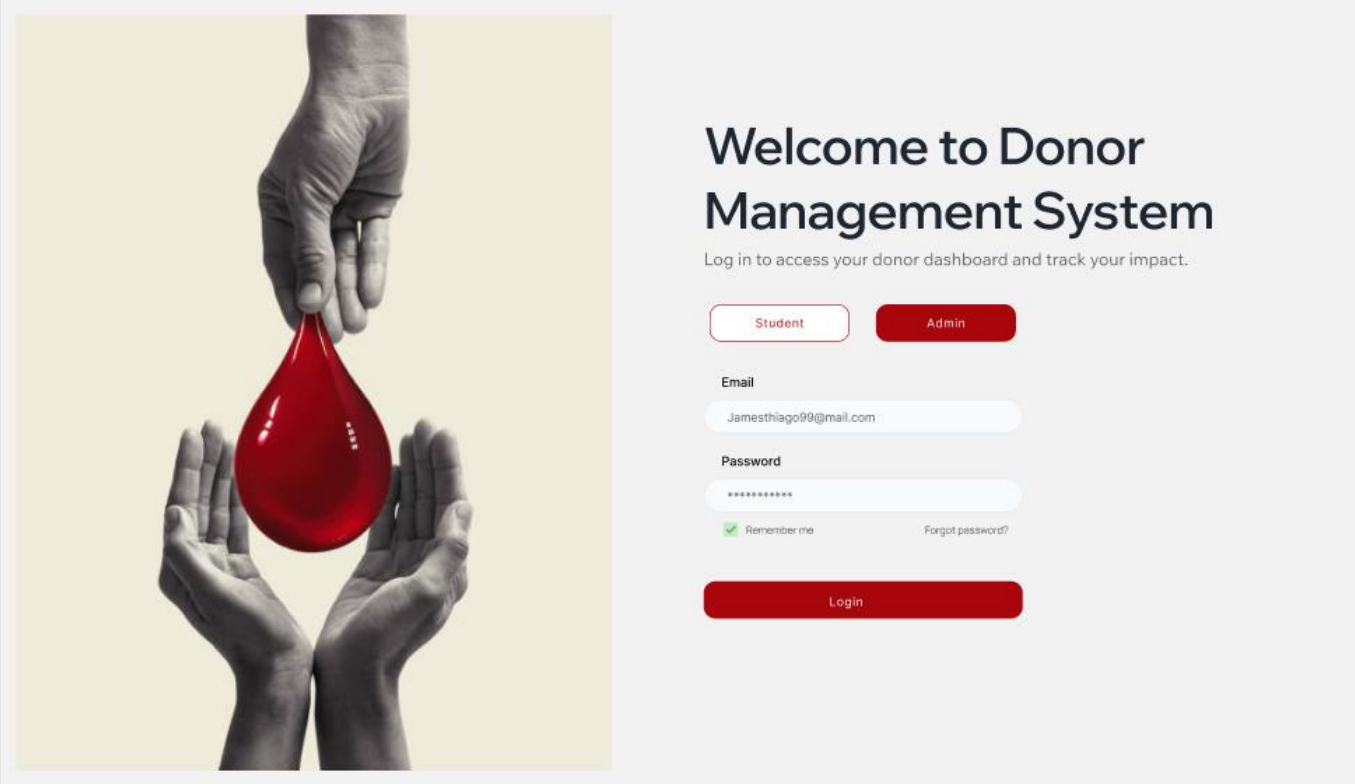

Welcome to Donor Management System
Log in to access your donor dashboard and track your impact.
Student
Admin
Email
Jamesthiago99@mail.com
Password
Remember me
Forgot password?
Login

Figure 2.1

2.4.5. Screen 03 Registration Page:

- **Purpose:** Allow new users (students) to create an account.
- **Functionality:**
  - Collect personal details and register the user in the system.
- **Elements:**
  - Input fields (Name, Email, Phone, Blood Group, Password).
  - Register/Submit button.
  - Already have an account? → Login link.

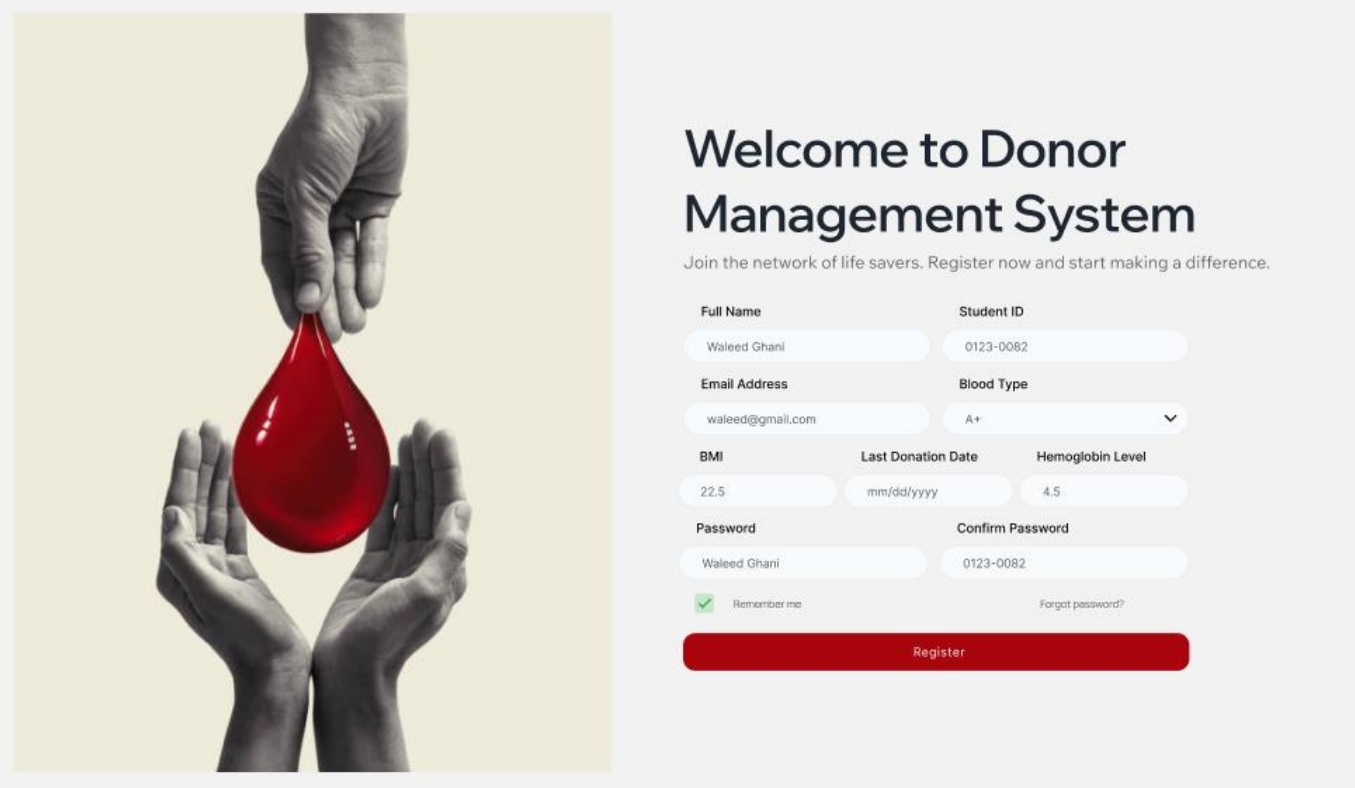


Figure 2.2

2.4.6. Screen 04 Dashboard (Overview / Stats):

- **Purpose:** The main control panel after login.
- **Functionality:**
  - Show an overview of important data (e.g., total donors, available blood, recent activity).
- **Elements:**
  - Sidebar navigation (links to Dashboard, Profile, Logout, etc.).
  - Top navigation bar (user info/avatar).
  - Summary cards showing key statistics.

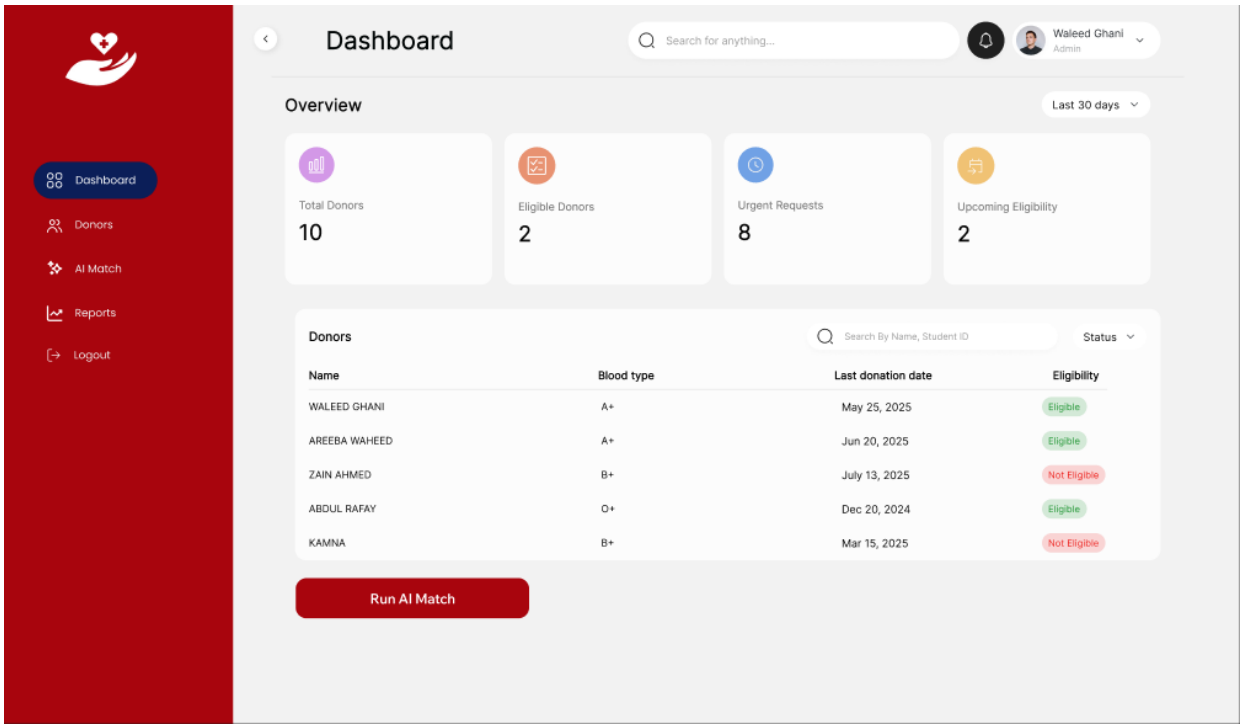


Figure 2.3

2.4.7. Screen 05 List of All Eligible Donors:

Dashboard (Table / List View):

- **Purpose:** To display a list of donors, donations, or requests in a structured way.
- **Functionality:**
  - Users can view, search, edit, or delete records.
- **Elements:**
  - Table with columns (Name, Blood Group, Contact, Status).
  - Edit/Delete action buttons for each record.
  - Pagination or filters for navigation.

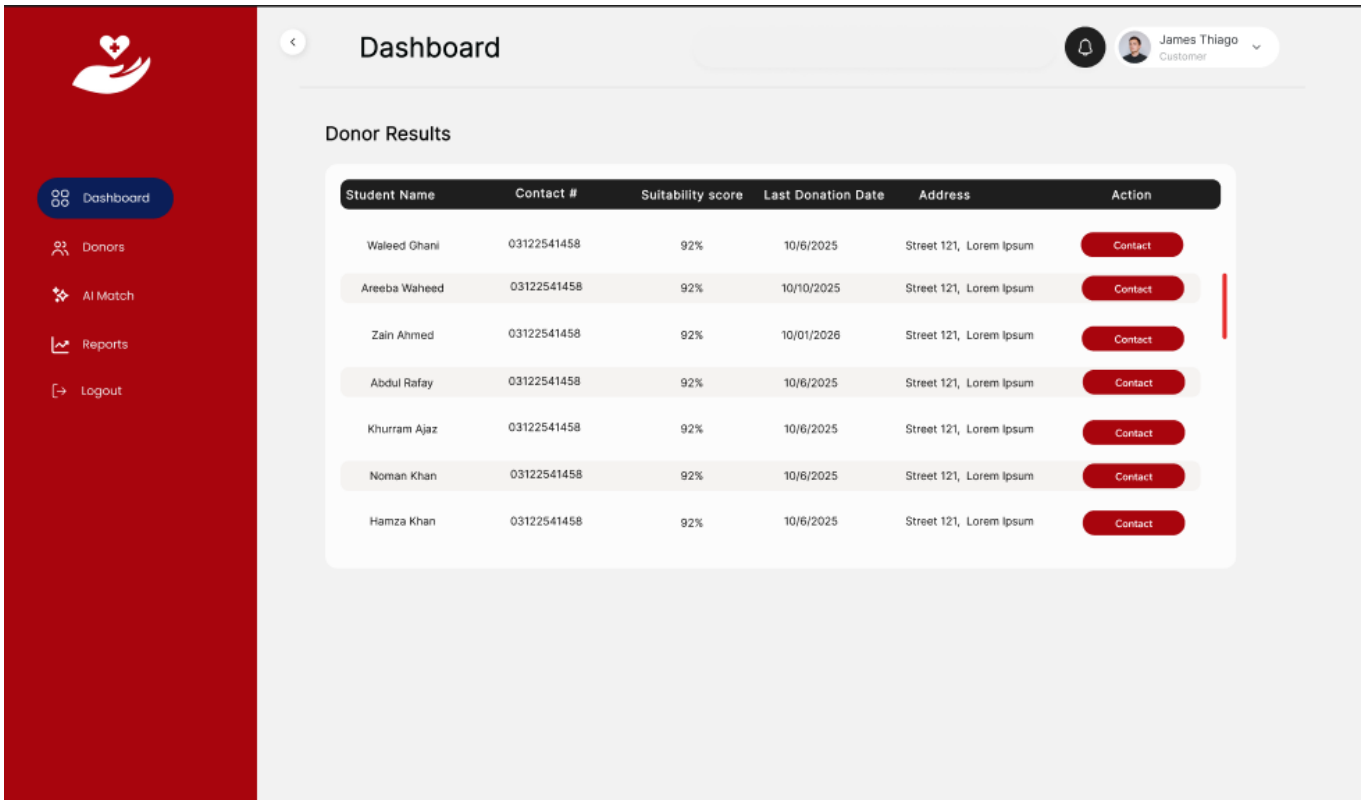


Figure 2.4

2.4.8. Screen 06 Donor Profile

- **Purpose:** To show the full details of a particular donor.
- **Functionality:**
  - View donor information and availability.
  - Admin/user can manage donor details.
- **Elements:**
  - Profile picture.
  - Basic details (Name, Email, Phone, Address).
  - Blood group and donation availability.

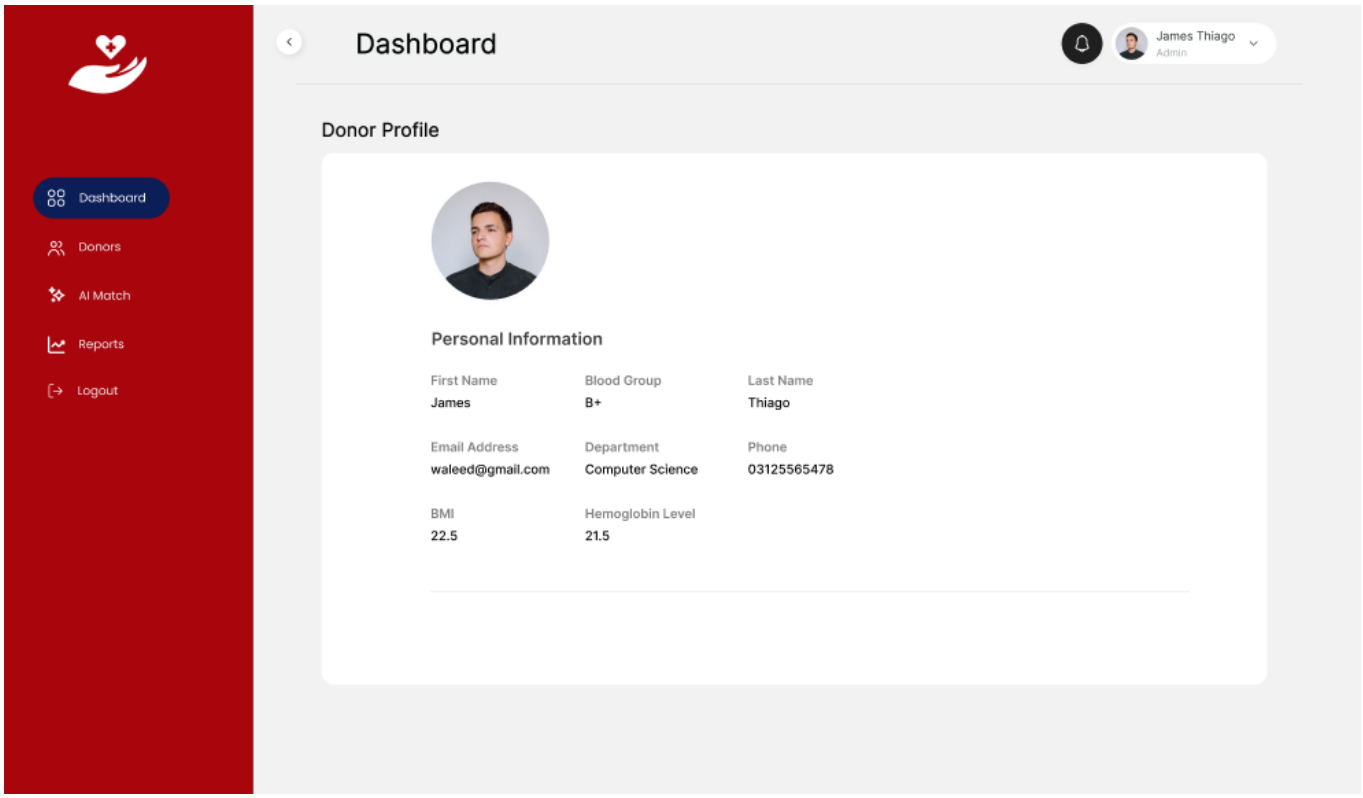


Figure 2.5

2.4.9. Screen 07 Edit My Profile

- **Purpose:** Allows a logged-in user to update their personal details.
- **Functionality:**
  - Edit fields like name, email, phone number, and address.
- **Elements:**
  - Pre-filled input fields with existing data.
  - Save Changes button.

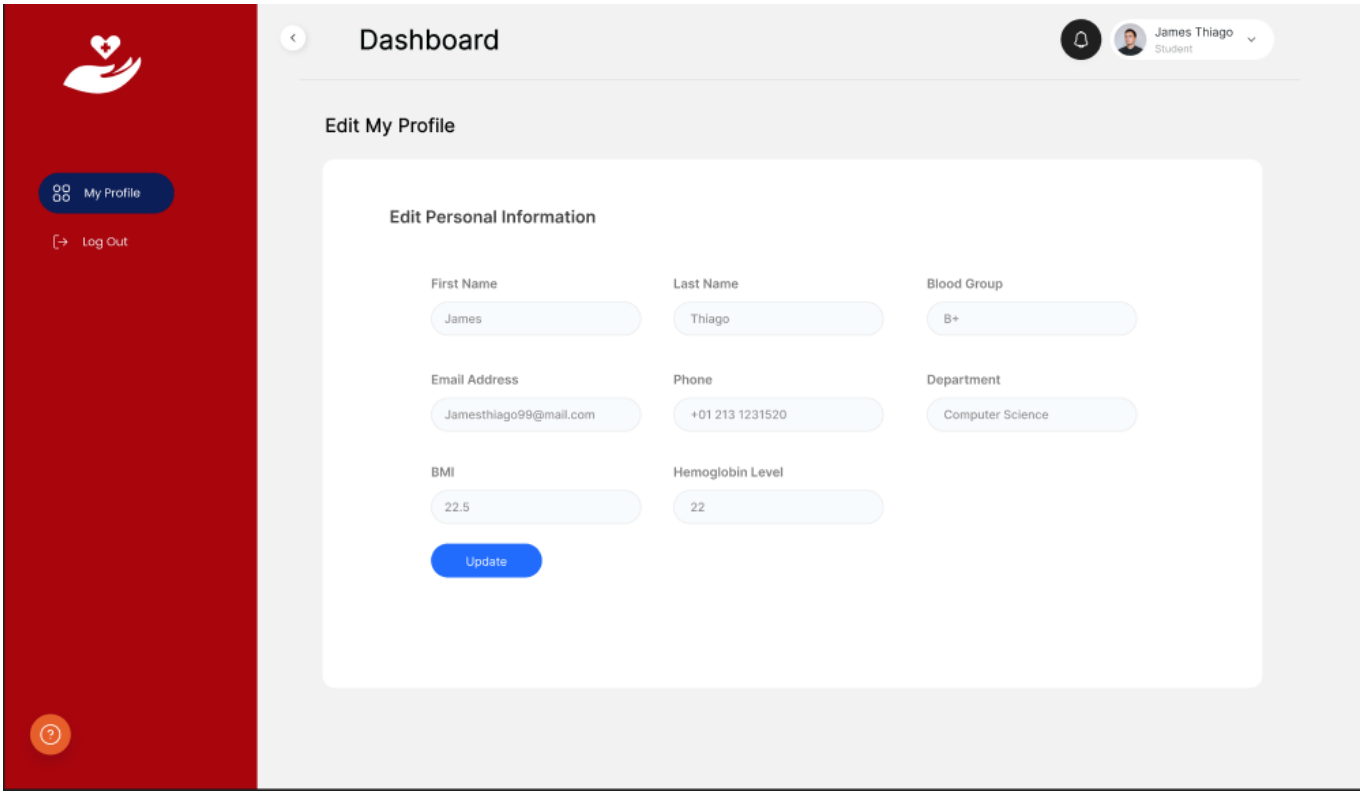


Figure 2.6

2.4.10. Screen 08 Change My Password

- **Purpose:** Provides a secure way for users to update their password.
- **Functionality:**
  - Verify old password.
  - Set a new password.
- **Elements:**
  - Fields for Current Password, New Password, Confirm Password.
  - Save/Update Password button.

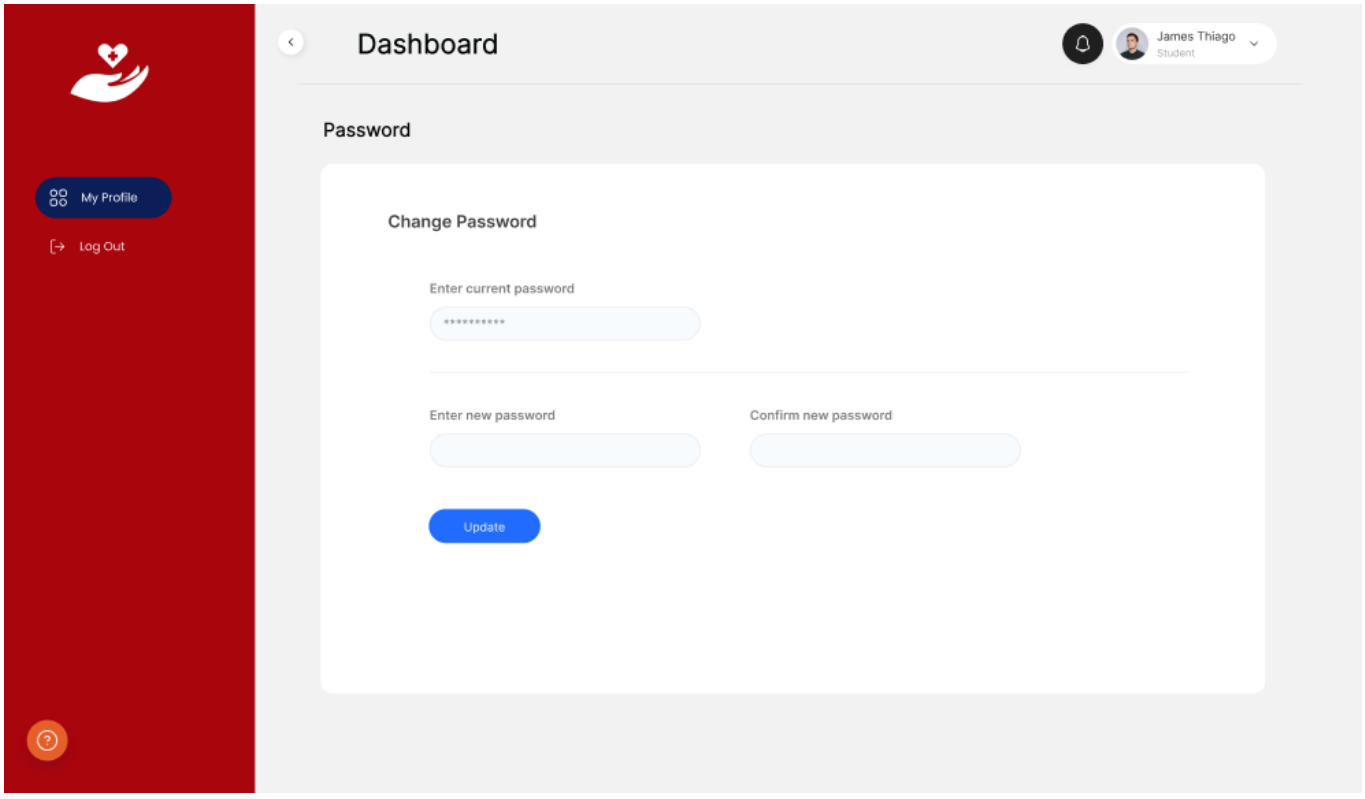


Figure 2.7

2.4.11. Screen 09 My Profile (View Mode):

- **Purpose:** To display the personal details of the logged-in user.
- **Functionality:**
  - User can see their profile details and donation-related information.
  - Option to go to "Edit My Profile."
- **Elements:**
  - Profile information (Name, Email, Phone, Blood Group).
  - Edit Profile button.

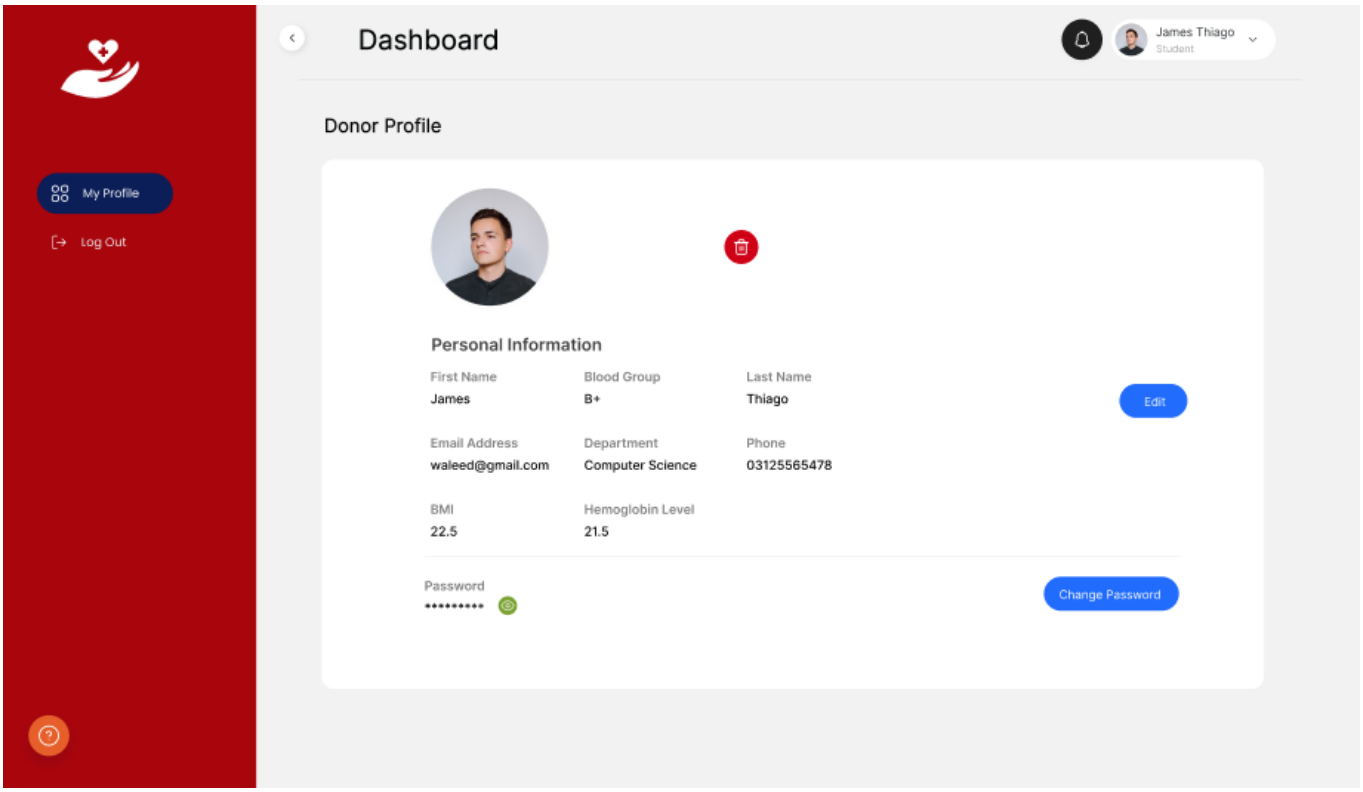


Figure 2.8

2.4.12. Screen 10 Run Ai Match:

- **Purpose:** To Run the AI matching mechanism.
- **Functionality:**
  - Admin can run AI match mechanism by filling the required fields.
- **Elements:**
  - Gender
  - Blood Type
  - Urgency Level
  - Required Date

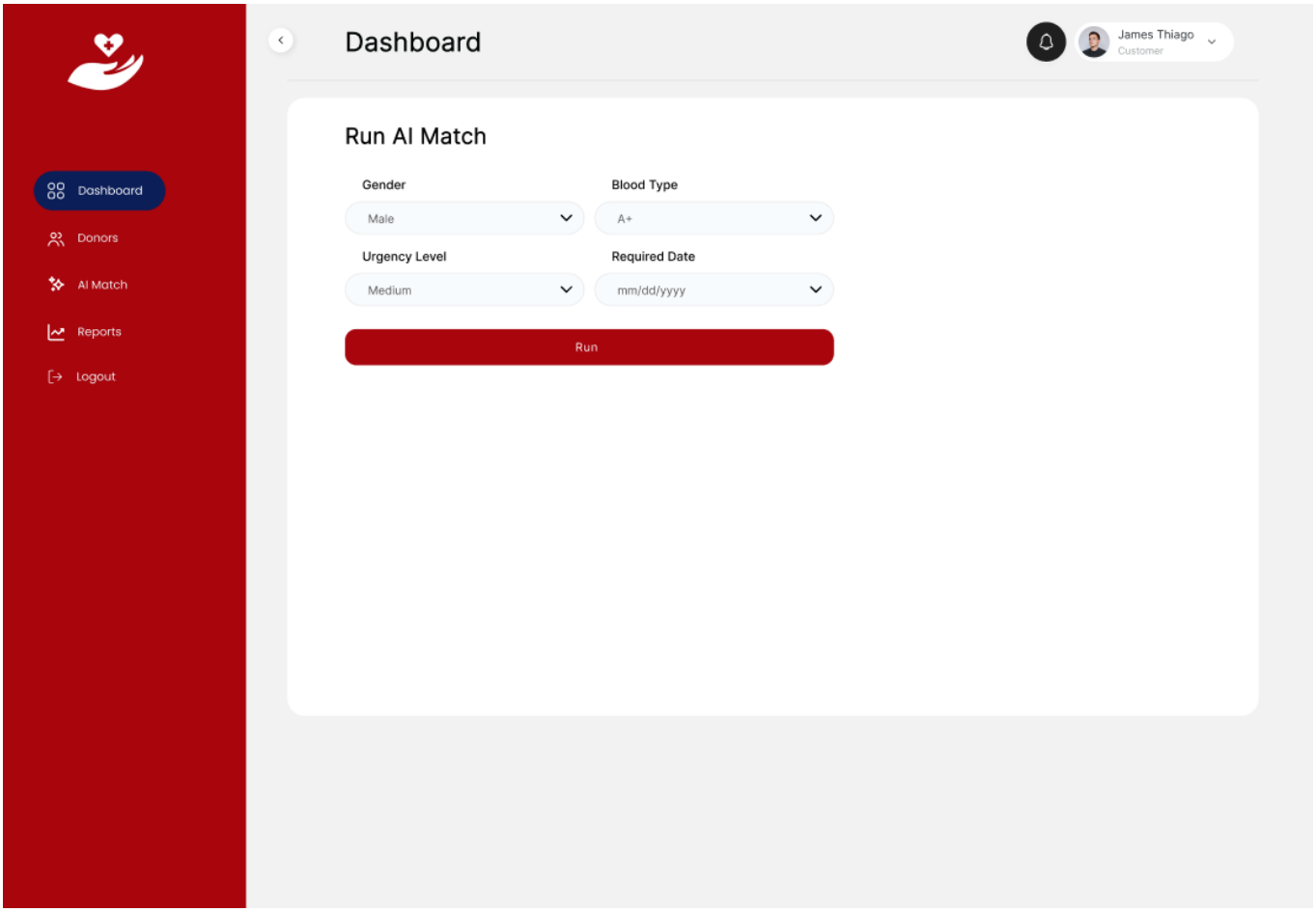


Figure 2.9

3. Risk Identification & Mitigation:

3.1. Risk Register:

Risk ID	Risk Description	Response Plan	Impact	Likelihood
R1	AI model produces inaccurate donor matches	Retrain model with real datasets; validate results	High	Medium
R2	Server downtime or infrastructure failure	Use cloud hosting with backup & monitoring	High	Low
R3	Sensitive donor data breach	Implement encryption, strong authentication, and audits	High	Medium
R4	Low adoption by students/admins	Conduct awareness campaigns and provide training	Medium	Medium
R5	Project delays due to small team size	Prioritize features, apply strict sprint planning	Medium	High
R6	Incomplete or incorrect donor information entered	Add validation checks, compulsory fields, and periodic audits	Medium	Medium
R7	Difficulty integrating system with university infrastructure	Coordinate with IT department, use standardized APIs	Medium	Low
R8	Legal or ethical issues with medical data handling	Follow data protection laws, seek approval from ethics committee	High	Low
R9	Lack of technical expertise in advanced features (e.g., AI model)	Take online training, consult experts, focus on MVP first	Medium	Medium
R10	Budget/resource limitations for system maintenance	Optimize resource use, seek university support/funding	Medium	Medium

Table 2.6

3.2. SWOT Analysis:

Table 2.7

Strengths	Weaknesses

AI-powered donor matching improves accuracy and speed in finding suitable donors.	Limited team size (only 2 members) may slow down development.
Centralized database ensures all donor records are secure and easily accessible.	High dependency on technology (AI, cloud), which may face integration challenges.
User-friendly web application with responsive design.	Initial setup and infrastructure costs may be high.
Supports thalassemia patients by ensuring consistent blood availability.	Limited testing resources compared to large-scale projects.

Opportunities	Threats
Potential to scale for multiple universities and hospitals.	Data privacy risks and potential security breaches.
Can integrate with NGOs or blood banks for wider adoption.	Resistance from students/admins in adopting a new system.
Raises awareness about regular blood donation and community impact.	Competition from other donor management platforms.
Future integration with mobile apps for broader accessibility.	AI model may face ethical or accuracy concerns.

Table 2.8

4. Conclusion

The **Donor Management System (DMS)** successfully addresses the challenges of managing blood donation drives by providing a secure, centralized, and AI-powered platform. Through features such as donor eligibility tracking, intelligent donor matching, and detailed reports and analytics, the system ensures the timely availability of blood for thalassemia patients and other critical cases.

By implementing modern technologies, including web-based interfaces and AI integration, the project enhances efficiency, reduces manual errors, and encourages a culture of regular blood donation among students. Despite limitations such as a small development team, the project demonstrates the potential to scale for wider use across multiple universities and healthcare organizations.

In conclusion, the Donor Management System not only improves the operational process of blood donation drives but also contributes to saving lives, making it a valuable step toward social good and technological innovation.