འཇིགས་མེད་རྣམ་རྒྱལ་བཟོ་རིག་མཐོ་རིམ་སློབ་གྲྭ།

Royal University of Bhutan

Jigme Namgyal Engineering Collage, Dewathang

ONLINE BLOOD DONATION SYSTEM

Module: **Information System Development and Management**

Module code: **SYD102**

Module Tutor: **Younten Tshering**

Submitted by: **Group 1**

Chogyal Wangdi (Project Manager)

Sonam Jamtsho

Dorji Samdrup

Norbu Wangmo

Contents

List of Figures

List of Tables

1. Chapter 1: Introduction

Blood is a lifeline in healthcare, it is essential for treating injuries, surgeries, chronic illnesses, and emergencies. It cannot be artificially manufactured which makes voluntary donations the only source for patients in need. According to the World Health Organization (WHO), a single donation can save up to three lives, highlighting its irreplaceable role in medical care.

Despite its critical importance, the blood donation system faces significant challenges. Hospitals and blood banks struggle with inefficient donor-recipient matching, outdated manual recordkeeping, and lack of real-time updates on blood availability. These inefficiencies lead to delays, shortages, and even preventable deaths during emergencies.

To address these issues, our project proposes an **Online Blood Donation System (OBDS)**, a digital platform that connects donors, recipients, and blood banks in real time. The system will streamline donor registration, automate blood requests, and provide instant notifications for urgent needs. By integrating a centralized database, our project aims to reduce response times, eliminate manual errors, and ensure timely access to blood.

The expected outcome is a more efficient, transparent, and user-friendly system that increases donor participation, optimizes blood inventory management, and ultimately saves more lives. With this solution, we hope to bridge the gap between supply and demand, making blood donation faster, safer, and more accessible for everyone.

1.1 Current System of blood donation

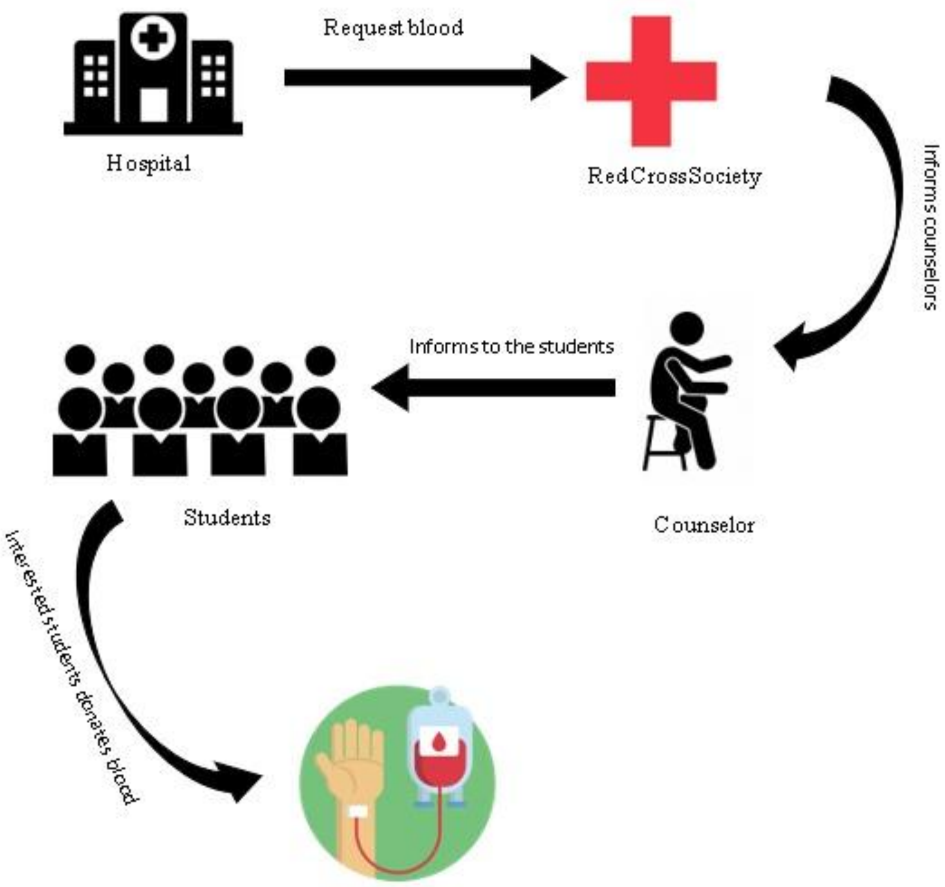


Figure 1.1 Current System of Blood Donation

The existing system's reliance on outdated methods which creates unnecessary bottlenecks, risking lives during emergencies. The need for a digital, real-time, and user-friendly blood donation management system is evident to streamline donor recruitment, real-time tracking, and emergency response. Our proposed **OBDS** aims to resolve these inefficiencies by automating processes and improving communication.

The diagram illustrates the manual processes dominating blood donation systems today:

- Donors must visit physical centers to register or donate.
- Hospitals and blood banks rely on phone calls or paper records to track inventory.
- No centralized database exists to match donors with recipients in real time.

Detailed breakdown of the existing challenges:

1. Manual and Time-Consuming Processes
    - Potential donors must visit physical blood banks or donation camps to register, which is time consuming.
    - Blood banks and hospitals often maintain handwritten logs or basic digital spreadsheets, making it difficult to track donor availability, blood types, and expiration dates efficiently.
    - In urgent cases, hospital staff must make phone calls or send messages to individual donors, wasting precious time during life-or-death situations.

2. Fragmented Communication
    - Direct communication is rare, forcing patients' families to personally seek donors through social media or community appeals.

3. No Real-Time Updates
    - Blood banks struggle to maintain up-to-date records of stock levels, leading to shortages or wastage of expired blood units.
    - There is no unified system to match donors with recipients quickly, resulting in delays when specific blood types are needed.

1.2 Problem Statement

Blood donation is a crucial for healthcare systems worldwide, as it ensures the availability of blood for emergencies, surgeries, and treatments for various medical conditions. However, the process of managing blood donations, connecting donors with recipients, and maintaining an adequate blood supply is often inefficient and disorganized.

Here are some of the challenges faced by hospitals, blood banks and donors:

1. Lack of Real-Time Information: Hospitals and blood banks face difficulties to maintain real-time information about blood availability, leading to delays in emergency situations.
2. Difficulty in Finding Donors: In urgent cases, finding compatible donors quickly can be challenging due to the lack of a centralized system.
3. Inefficient Donor Management: Blood banks often find it difficult to track donor details, donation history, and eligibility for future donations.
4. Manual Processes: Traditional methods of managing blood donation records are time-consuming, error-prone, and inefficient.

1.3 Aim

The aim is to develop Online Blood Donation System that will digitalize the current blood donation process and promote ease of donating blood and also to improve the healthcare system as a whole.

1.4 Objectives

The main objective of developing the Online Blood Donation System is to provide efficient and effective way to support blood donation and save lives

- The primary objective of the Online Blood Donation System is to effectively solves the issues and problems faced by hospitals and blood banks this includes maintaining a real-time information about donors and blood availability during emergencies.
- The Online Blood Donation System reduces the time in finding compatible donors during emergency cases. The Online Blood Donation System aims to maintain a secure digital donor record thereby replacing paper processes with easy online forms.
- The system sends automatic alert messages to nearby eligible donors when needed.
- The system ensures mobile and computer access for all users, protect donor privacy, send donation reminders, help hospitals manage blood supplies better, and ultimately save lives by speeding up emergency responses.

2. Chapter 2: Planning Phase

2.1 Methodology

Our system, **OBDS**, is designed to effectively connect blood donors with those in need. Since this system requires real-time updates, smooth user interaction, and continuous improvement, we chose the **Agile Methodology** to develop this project.

Reasons for Choosing Agile:

1. **Adaptability to Changing Requirements**: Donor details and blood requests may change frequently. Agile allows us to handle these changes smoothly without affecting the system's performance.
2. **Reduced Risk During Testing and Creation**: Since the blood donation system involves sensitive information, any errors can lead to serious consequences. By using Agile, we can test each feature step-by-step thereby reducing risks.
3. **Improved Project Predictability**: Agile helps us to track progress and meet deadlines.
4. **Early Design Problem Discovery**: The Agile model allows us to detect design flaws and security issues in the early phase.

Case study like the **Red Cross Blood Management System** (Abdul-Gafaru et al., 2024), which also follows the Agile methodology, has shown significant improvement in handling donor data and managing blood requests. With Agile, their system easily adapted to changing donor requirements and update blood availability in real-time. Moreover, Agile reduced risks during testing, ensuring that critical errors are identified at an early stage. This approach has improved project predictability and allowed the Red Cross system to deliver a secure and efficient platform for managing blood donations.

Similarly, the **Blood Bank Management System** successfully implemented Agile to improve the efficiency of blood inventory management (Blood Bank Management System, 2023). Through Agile's continuous testing and feedback loop, the system was able to detect design issues early and make necessary improvements. The flexibility of Agile also allowed the Blood Bank system to handle emergency blood requests and donor registration without affecting the system's performance. This has resulted in a more reliable and user-friendly platform.

2.2 Expected outcome

This project is expected to produce a functional online system for blood donations. Patients and health care facilities in need of blood will receive better and quicker services using an online blood donation system.

1. Register
    - Users (donors, hospitals, and admins) can easily sign up through a simple registration process.
    - Ensures a verified database of donors and healthcare facilities.

2. Login
    - Secure access to personalized dashboards for donors, hospitals, and administrators.
    - Allows users to manage their profiles, track donations, and request blood when needed.

3. Admin
    - The admin oversees the entire system, managing user registrations, blood requests, and availability.
    - Ensures smooth operations, verifies donor eligibility, and prevents fraudulent activities.

4. Volunteers for blood donation
    - Individuals willing to donate blood can register and receive updates on nearby donation camps.
    - Volunteers can track their donation history and receive reminders for future eligibility.

5. Hospital registration
    - Hospitals can register to request blood, check donor availability, and manage inventory efficiently.
    - Reduces delays in emergency cases by connecting hospitals directly with suitable donors.

2.3 Project Overview

Project Title: Online Blood Donation System (OBDS)

The project aims to develop a comprehensive Online Blood Donation System to provide an effective and effective platform to support blood donation and save lives. This system will address the challenges faced by donors, hospitals and blood banks in this existing manual blood donating progress. It aims to address the challenges faced by hospitals and blood banks by maintaining real-time information on donors and blood availability, esp. during emergencies. The system reduces the time required to find compatible donors, replaces traditional paper processes, and ensures easy access via mobile and computer. Additionally, it sends automatic alerts to eligible nearby donors, protects donor privacy, sends donation reminders, and helps hospitals manage blood supplies more effectively. Ultimately leading to enhance emergency response and saving more lives.

We plan to use an agile method with an iterative approach because we have a clear idea of what we are going to build and it provides several benefits to our System. With the use of agile methodology our project is expected to automate blood donation process, make the system easier and more engaging for users, and improve how data is managed. Additionally, the project is practical and possible in every way (technically, economically, operationally, legally and contractually, politically and in terms of schedule as it fulfills all the necessary requirements with available resources and support to ensure successful deployment of the system.

2.4 Feasibility Summary

The Online Blood Donation System looks like a good investment, even though it’s a non-profit project. It only needs Nu. 10,000 to build and about Nu. 7,000 each year to keep running. While it doesn’t earn money directly, it helps save around Nu. 12,000 every year by making things faster and reducing paperwork.

Financial checks show that the system is worth the cost. It has a positive return over time, and the money spent will be recovered in around 2 and a half years. After that, it just keeps saving more. On top of that, it brings other useful benefits, like happier donors, faster help in emergencies, and more openness in the process.

On the technical side, there’s nothing complicated about it. The tools and skills needed are already available. The system is very easy to use so it doesn’t need any training and is practical. It also follows the legal rules in Bhutan keeping people’s information safe. Politically, some hospitals might not be ready to share data right away, but with proper training, guidelines, and trying it first in smaller hospitals can get more support and can grow over time.

2.5 Resources Requirement

To make the OBDS work well, we need some important things. First, we need computers or laptops to work on the system, a good internet connection, and storage like hard drives or cloud space to save important data. We will use Python for coding, and tools like Jira to manage the project. To store donor and receiver information, we will use a MySQL database. Some web development tools will also help us build the system more easily. We need frontend and backend developers to write the code, UI designers to create an "easy to use" design, and data analysts to manage the data. A project manager will make sure everything is going as planned and on time. We need approval from the administration and help from the IT team. And also, feedback from donors, medical staff, and blood donation coordinators will help us improve the system and make it better.

2.6 Schedule

Based on our project plan and available resources, we can complete the development of the OBDS on time. Our team has four members, each monitored by the scrum master assigning each member specific tasks with enough time to complete them efficiently. With guidance from our stakeholder, we are confident that all parts of the project can be finished on schedule. We also have a clear plan for the development process, which shows that the project is achievable and will be completed within the planned timeframe.

2.7 Proposed Solution

To solve the problems in the current blood donation process, like manual work and no real-time updates, we plan to create an OBDS. This system will let users check upcoming donation events, sign up to donate, and get important information using their phones or laptops with internet access. The system will make the donation process faster and easier. It will save time and improve how information is managed. Everything will be more organized and clearer, which will encourage more people to donate blood and help the process run smoothly.

2.8 Feasibility Assessment

2.8.1 Economic Feasibility

i. Initial Development Cost: Nu. 10,000 (one-time investment)

ii. Operational Costs: Nu. 7,000

- Hosting in the cloud: Nu. 4,000 per year
- Software updates, maintenance and training: Nu 3000 per year

iii. For benefit we have evaluated decrease in cost of Nu. 1200

Intangible Costs:

i. Time-consuming implementation process.

ii. Requires staff training for system adoption.

iii. Potential temporary productivity loss during transition.

Intangible Benefits:

i. Enhanced donor satisfaction.

ii. Improved communication and transparency.

iii. Faster emergency response capabilities.

iv. Better donor retention through streamlined processes.

v. Increased community engagement.

Benefits

- Increased Revenue: In this project there is no increased revenue as it is non-profit organization.
- Decreased costs: Nu. 12,000 (from administrative efficiency, paper reduction, error reduction, and time savings)
  - This enhanced administrative efficiency leads to decreased operational cost, making overall blood inventory management more cost-efficient and efficient.

Tangible benefits: There is no tangible benefits in this project because it is not profit driven project and it is mainly to serve hospitals and blook band and address issues.

Intangible benefits: The benefits provided by our system are often intangible. Intangible benefits like enhancing donor's satisfaction, ensuring faster emergency response, ensuring donor retention through streamlined processes, etc.

Financial Analysis

Net present Value (NPV): To calculate NPV we have to determine the cash flows for each period of the investment or project, discount them to present value, and subtract the initial investment from the sum of the project discounted cash flows. We have calculated net present value by dividing total annual benefits by discount rate (10%).

Return on Investment (ROI): ROI is the ratio between net income and investment. It is calculated by subtracting the initial cost of the investment from its final value, then dividing this new number by cost of the investment, and finally, multiplying it by 100.

Break -Even Analysis: It is a financial calculation that weighs the costs of new business, service or product against the unit sell price to determine the point at which you will break even. In our project the break-even point is supposed to be between 1 and 2 years and the actual break-even point is 0.99 which is almost 1 year.

Financial analysis cost:

Discount Rate: 10%

Development Cost: Nu. 10,000 (one-time investment)

Operational Cost: Nu. 7,000

Benefits

Increased revenue: Null

Decreased costs: Estimated amount 12000 annually.

Net Present Value (NPV) Calculation:

Year 0: Initial investment = -10,000

Years 1-5:

Annual Benefits: 45492.36

Annual Costs: 36537.21

Annual Net Benefit: 8955.15

Present Value of Net Annual Benefits:

Year 1: 10909.1

Year 2: 9917.36

Year 3: 9015.78

Year 4: 8196.72

Year 5: 7453.42

NPV calculation

NPV = Present Value (benefit) – Present Value (Cost)

NPV = 45492.36 – 36537.21

= 8955.15

Return on Investment (ROI):

ROI = (NPV (benefits) – NPV (cost)/ NPV (Cost)) * 100%

ROI = (45492.36 – 36537.21 / 36537.21) * 100% = 24.51%

Break-Even Analysis:

The system reaches break-even point at approximately 2 years, 4months.

Year	NPV (Benefits)	NPV (Costs)
0	0	1000
1	10909.09	16363.64
2	20826.45	22148.76
3	29842.22	27407.96
4	38038.945	32189.385
5	45492.36	36537.21

Table 2.1 Break-Even Analysis

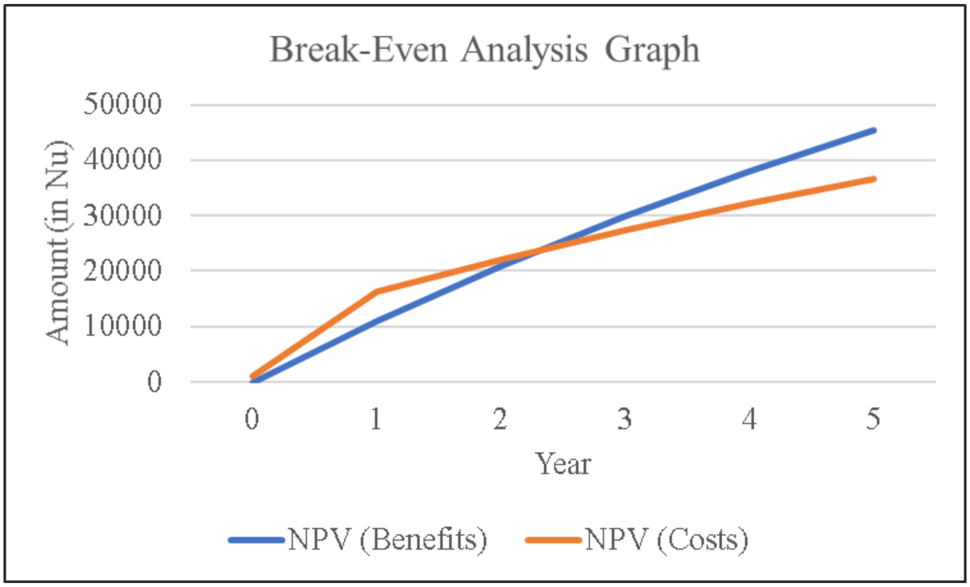
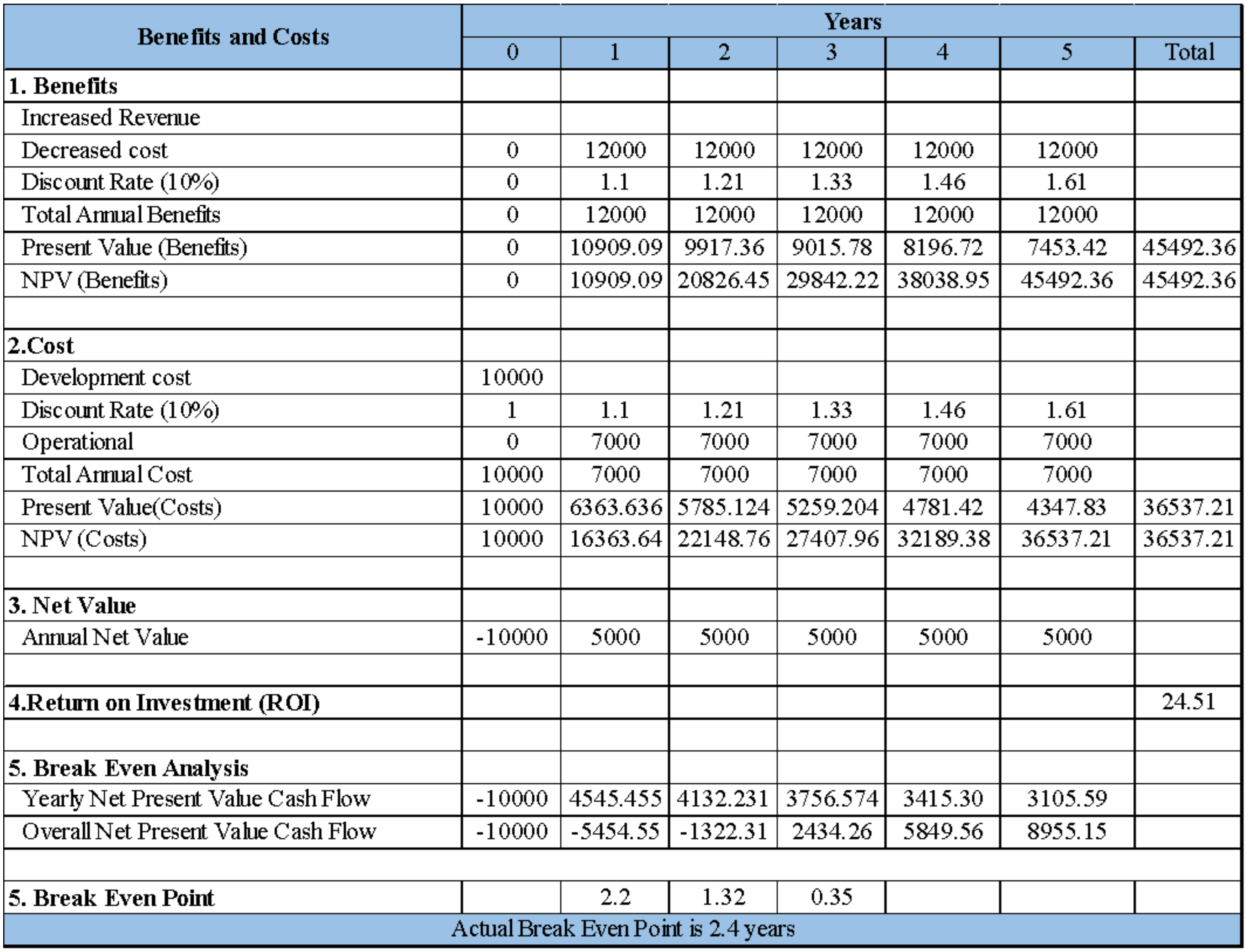


Figure 2.1 Break-Even Analysis Chart



Benefits and Costs	**Years**						
	0	1	2	3	4	5	Total
1. Benefits							
Increased Revenue							
Decreased cost	0	12000	12000	12000	12000	12000	
Discount Rate (10%)	0	1.1	1.21	1.33	1.46	1.61	
Total Annual Benefits	0	12000	12000	12000	12000	12000	
Present Value (Benefits)	0	10909.09	9917.36	9015.78	8196.72	7453.42	45492.36
NPV (Benefits)	0	10909.09	20826.45	29842.22	38038.95	45492.36	45492.36
2.Cost							
Development cost	10000						
Discount Rate (10%)	1	1.1	1.21	1.33	1.46	1.61	
Operational	0	7000	7000	7000	7000	7000	
Total Annual Cost	10000	7000	7000	7000	7000	7000	
Present Value(Costs)	10000	6363.636	5785.124	5259.204	4781.42	4347.83	36537.21
NPV (Costs)	10000	16363.64	22148.76	27407.96	32189.38	36537.21	36537.21
3. Net Value							
Annual Net Value	-10000	5000	5000	5000	5000	5000	
4.Return on Investment (ROI)							24.51
5. Break Even Analysis							
Yearly Net Present Value Cash Flow	-10000	4545.455	4132.231	3756.574	3415.30	3105.59	
Overall Net Present Value Cash Flow	-10000	-5454.55	-1322.31	2434.26	5849.56	8955.15	
5. Break Even Point		2.2	1.32	0.35			
Actual Break Even Point is 2.4 years							

Table 2.2 Cost-benefit Analysis

The implementation of the OBDS clear financial benefits, resulting in total annual benefits of Nu.12,000. Accounting for the time value of money (discount rate of 10%), the present value of these benefits over five years amounts to Nu.45,492.36. Development costs for the system are initially incurred at Nu.10,000, while operational costs amount to Nu.7,000 annually. Considering the discount rate (10%), the present value of these costs over five years totals Nu.36,537.21. After factoring both benefits and costs, the net annual values improve from an initial deficit of Nu.10,000 to consistent positive values of Nu.5,000 in subsequent years. The overall ROI for the project is calculated to be 24.51%, indicating a strong return on investment. The project achieves its break-even point between the second and third years, with the actual break-even point occurring at 2.4 years.

In conclusion, the OBDS is economically feasible, as it generates substantial benefits, a positive net present value, and a favorable ROI, with a reasonable break-even period.

2.8.2 Technical Feasibility

Technical feasibility assesses whether the proposed system can be developed using existing technology, within the available technical resources, and by the development team's current capabilities. It ensures that the system can be successfully implemented without encountering insurmountable technological issues.

Our OBDS is technically possible to build because we have all the tools and technologies we need. We know how to use web development tools and databases, and we have the skills to create the system. Additionally, we are further supported by two stakeholders and a teacher who will provide general guidance throughout the project. On top of that, we have access to two technicians to assist with troubleshooting and resolving any technical issues. With this strong foundation of tools, skills, and support, the system can be successfully developed and deployed within the given timeframe.

2.8.3 Operational Feasibility

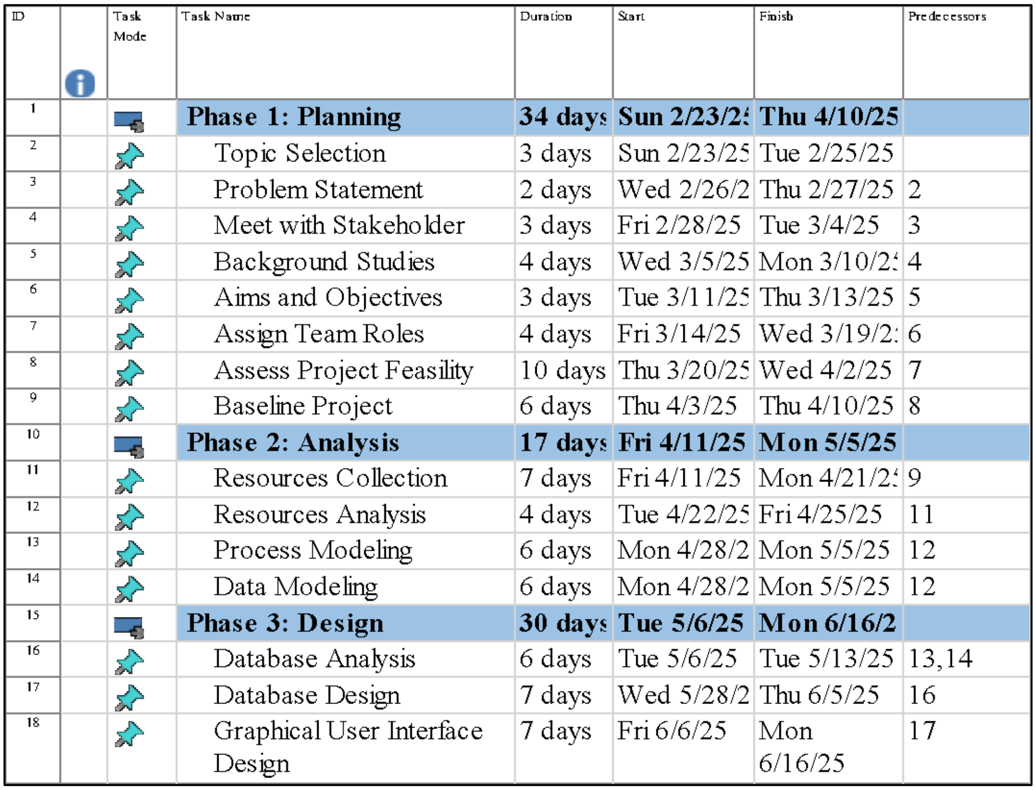
It is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The system working is quite easy to use and learn due to its simple but attractive interface. Users require no special training for operating the system. Data retrieval and the data presenting will be done by the system so the report generation will be automated by the system.

Our Online Blood Donor System is operationally possible because it solves a real problem and is helpful to many people. In emergencies, people often find it hard to get blood donors quickly. Our system makes it easier by letting users find donors based on blood group and location. This saves time and can even save lives. The system is simple to use for donors, hospitals, and admins. We also have support from our stakeholders and teacher to help run the system smoothly. With our team and technicians available for maintenance and support, with this support and a good plan, our system can work well in real situations. And also, we believe the system will work well in real-life use.

2.8.4 Schedule Feasibility

Table 2.3 Schedule Feasibility



ID	ℹ	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1			**Phase 1: Planning**	**34 days**	**Sun 2/23/2**	**Thu 4/10/25**	
2			Topic Selection	3 days	Sun 2/23/25	Tue 2/25/25	
3			Problem Statement	2 days	Wed 2/26/2	Thu 2/27/25	2
4			Meet with Stakeholder	3 days	Fri 2/28/25	Tue 3/4/25	3
5			Background Studies	4 days	Wed 3/5/25	Mon 3/10/2	4
6			Aims and Objectives	3 days	Tue 3/11/25	Thu 3/13/25	5
7			Assign Team Roles	4 days	Fri 3/14/25	Wed 3/19/2	6
8			Assess Project Feasility	10 days	Thu 3/20/25	Wed 4/2/25	7
9			Baseline Project	6 days	Thu 4/3/25	Thu 4/10/25	8
10			**Phase 2: Analysis**	**17 days**	**Fri 4/11/25**	**Mon 5/5/25**	
11			Resources Collection	7 days	Fri 4/11/25	Mon 4/21/2	9
12			Resources Analysis	4 days	Tue 4/22/25	Fri 4/25/25	11
13			Process Modeling	6 days	Mon 4/28/2	Mon 5/5/25	12
14			Data Modeling	6 days	Mon 4/28/2	Mon 5/5/25	12
15			**Phase 3: Design**	**30 days**	**Tue 5/6/25**	**Mon 6/16/2**	
16			Database Analysis	6 days	Tue 5/6/25	Tue 5/13/25	13,14
17			Database Design	7 days	Wed 5/28/2	Thu 6/5/25	16
18			Graphical User Interface Design	7 days	Fri 6/6/25	Mon 6/16/25	17

Gannt Chart

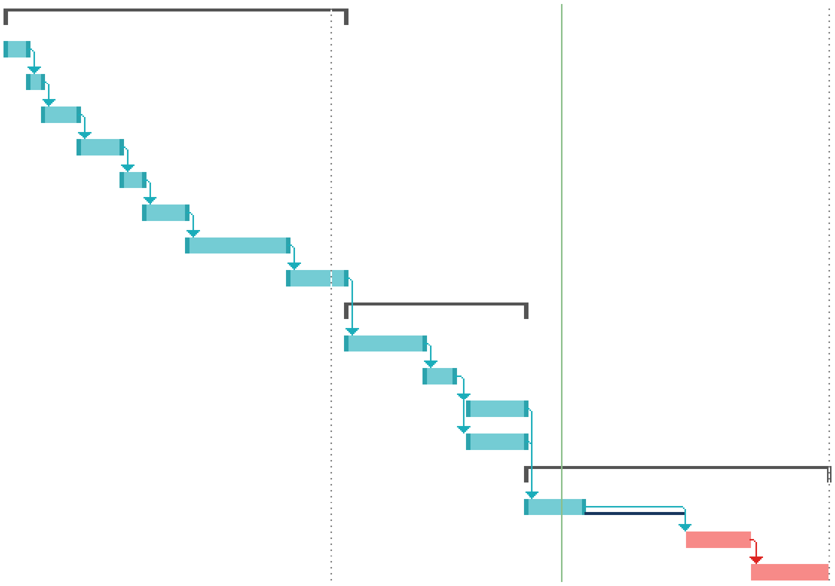
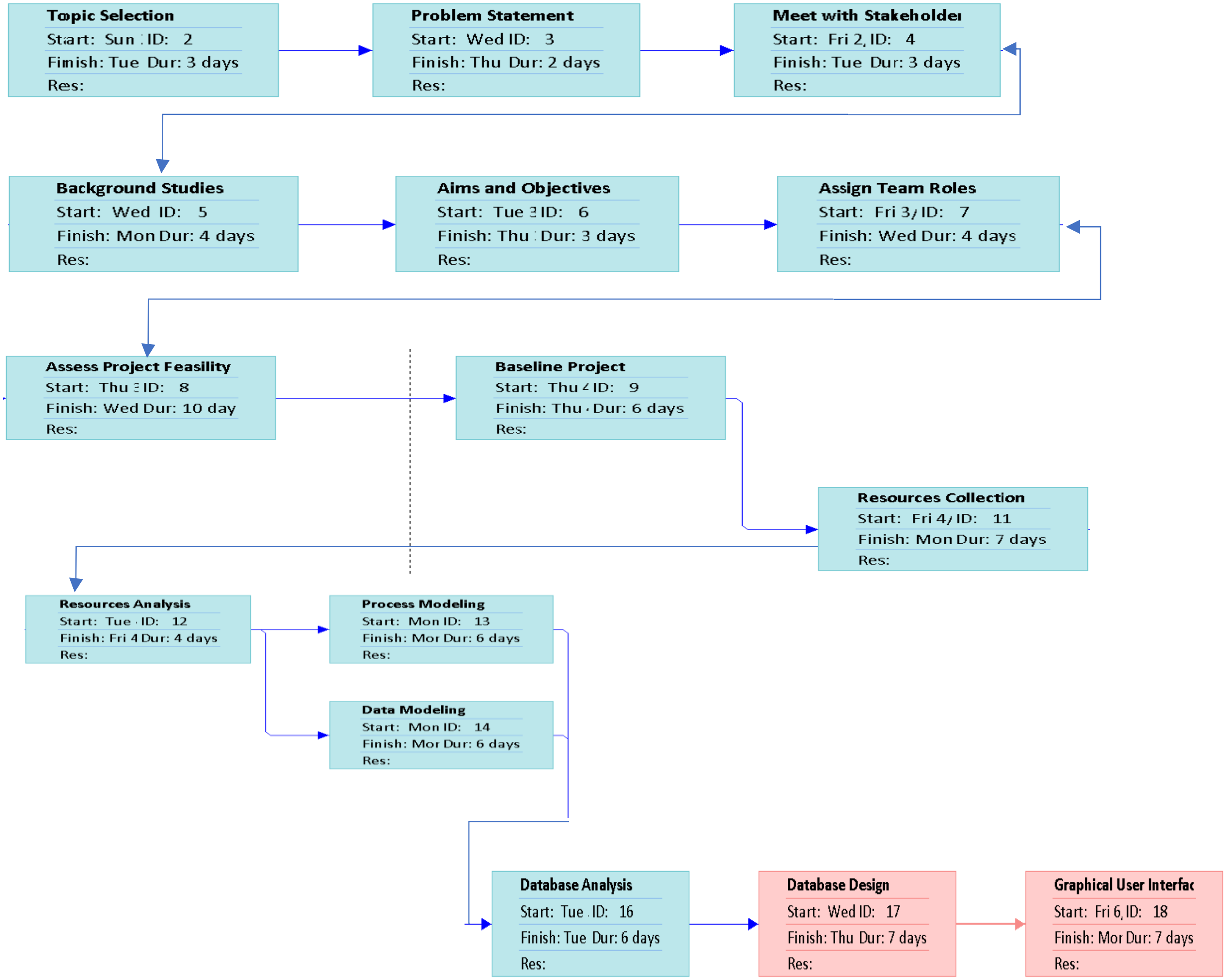


Figure 2.2 Gantt Chart

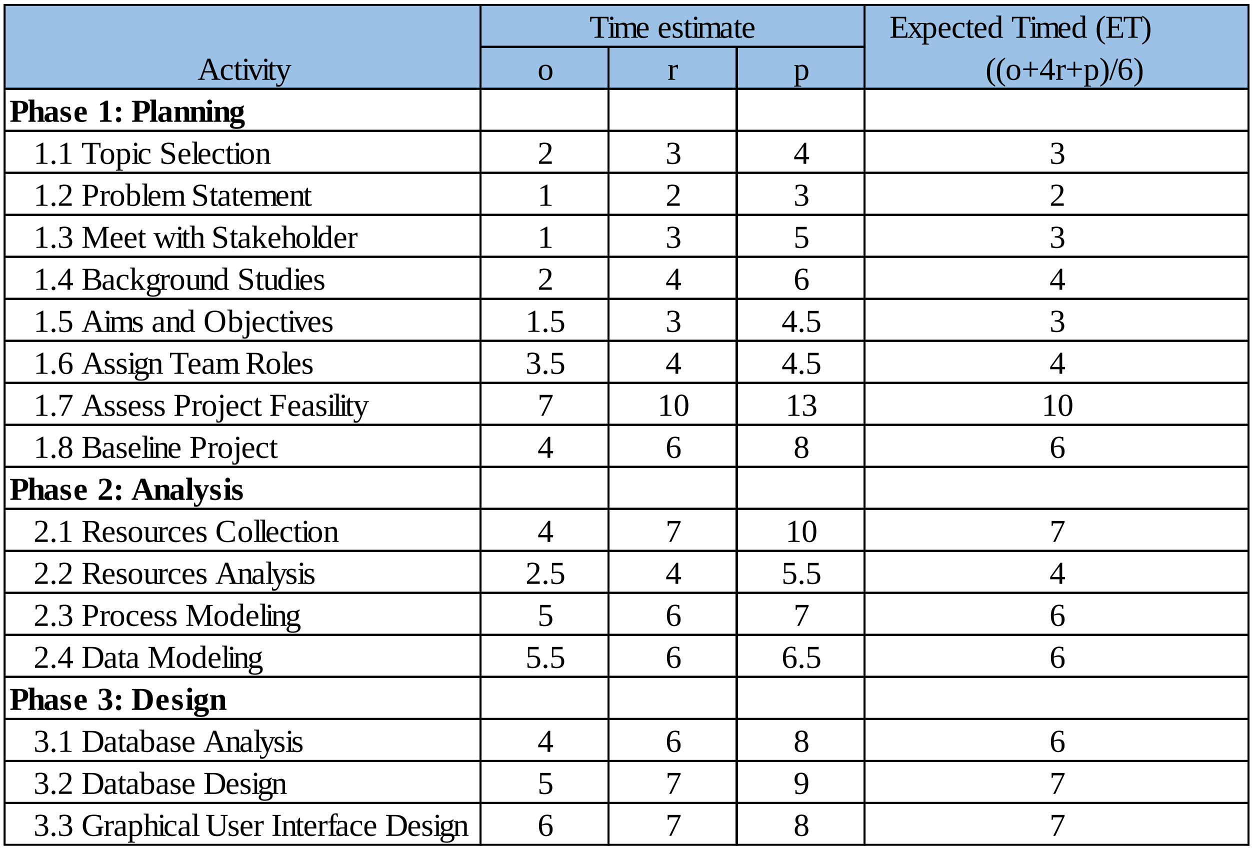
Network Diagram

Figure2.3 Network Diagram



PERT Analysis

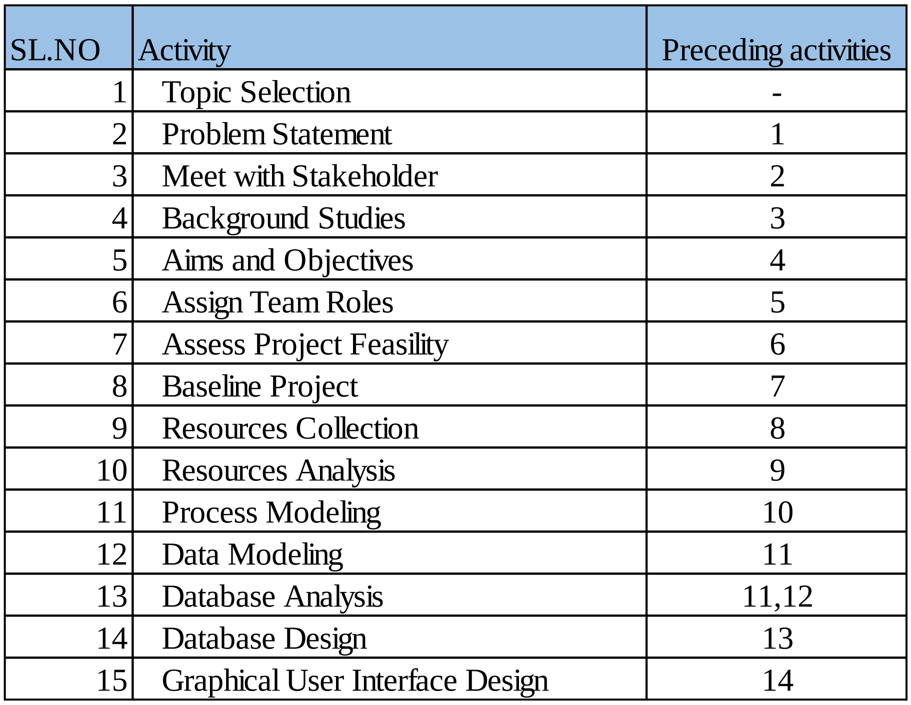
Table 2.4 PERT Analysis



	Time estimate			Expected Timed (ET)
Activity	o	r	p	((o+4r+p)/6)
Phase 1: Planning				
1.1 Topic Selection	2	3	4	3
1.2 Problem Statement	1	2	3	2
1.3 Meet with Stakeholder	1	3	5	3
1.4 Background Studies	2	4	6	4
1.5 Aims and Objectives	1.5	3	4.5	3
1.6 Assign Team Roles	3.5	4	4.5	4
1.7 Assess Project Feasility	7	10	13	10
1.8 Baseline Project	4	6	8	6
Phase 2: Analysis				
2.1 Resources Collection	4	7	10	7
2.2 Resources Analysis	2.5	4	5.5	4
2.3 Process Modeling	5	6	7	6
2.4 Data Modeling	5.5	6	6.5	6
Phase 3: Design				
3.1 Database Analysis	4	6	8	6
3.2 Database Design	5	7	9	7
3.3 Graphical User Interface Design	6	7	8	7

Critical Path Scheduling

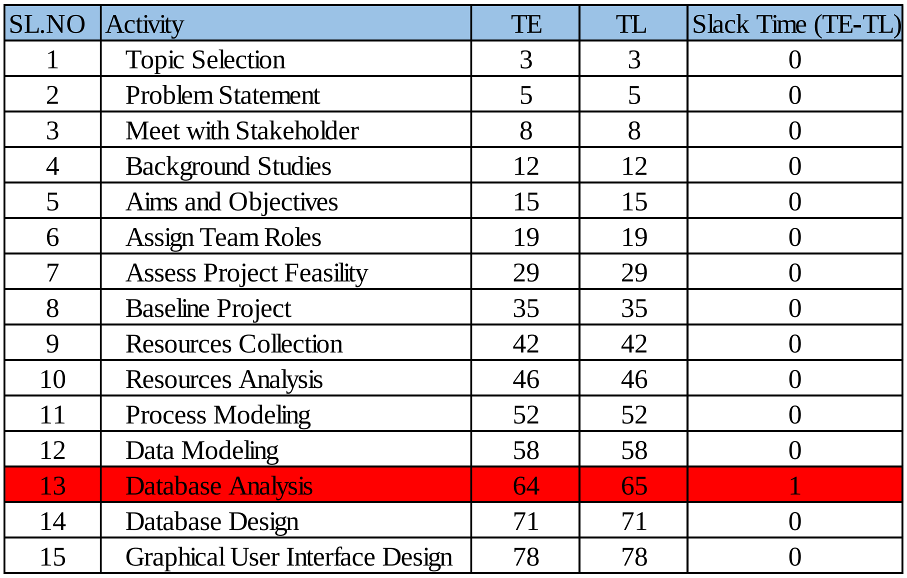
Table 2.5 Critical Path Scheduling



SL.NO	Activity	Preceding activities
1	Topic Selection	-
2	Problem Statement	1
3	Meet with Stakeholder	2
4	Background Studies	3
5	Aims and Objectives	4
6	Assign Team Roles	5
7	Assess Project Feasility	6
8	Baseline Project	7
9	Resources Collection	8
10	Resources Analysis	9
11	Process Modeling	10
12	Data Modeling	11
13	Database Analysis	11,12
14	Database Design	13
15	Graphical User Interface Design	14

Critical Path Calculation

Table 2.6 Critical Path Calculation



SL.NO	Activity	TE	TL	Slack Time (TE-TL)
1	Topic Selection	3	3	0
2	Problem Statement	5	5	0
3	Meet with Stakeholder	8	8	0
4	Background Studies	12	12	0
5	Aims and Objectives	15	15	0
6	Assign Team Roles	19	19	0
7	Assess Project Feasility	29	29	0
8	Baseline Project	35	35	0
9	Resources Collection	42	42	0
10	Resources Analysis	46	46	0
11	Process Modeling	52	52	0
12	Data Modeling	58	58	0
13	Database Analysis	64	65	1
14	Database Design	71	71	0
15	Graphical User Interface Design	78	78	0

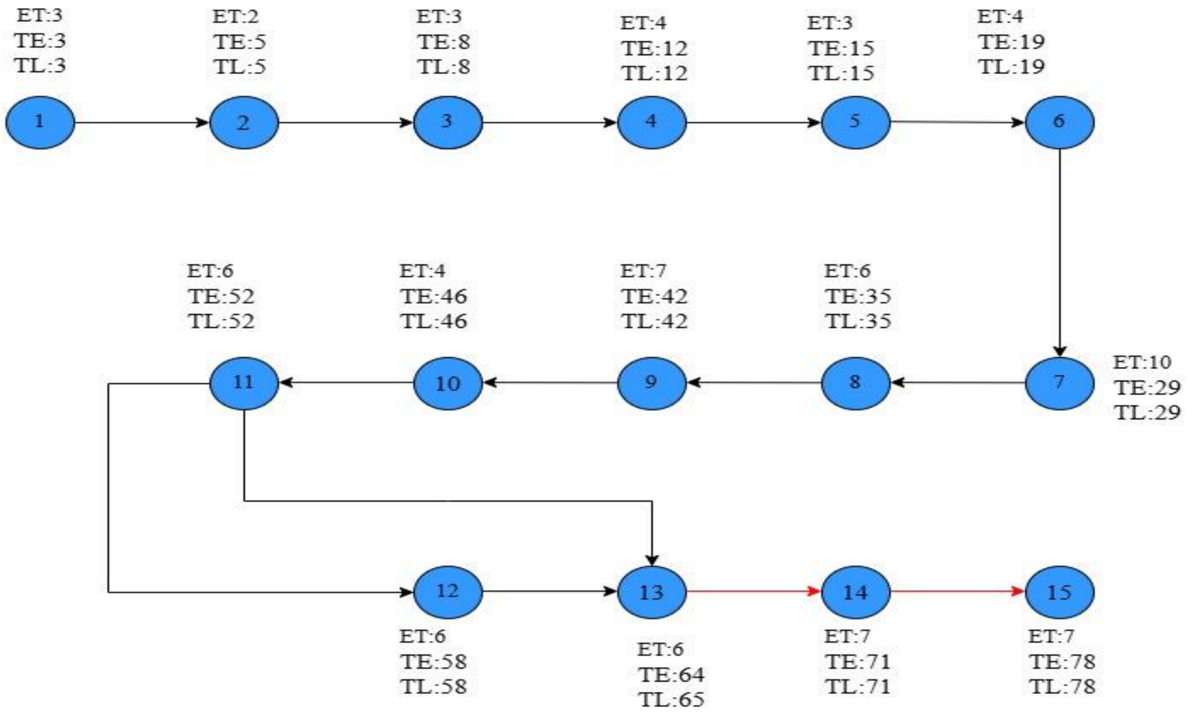


Figure 2.4 Critical Path

2.8.5 Legal and Contractual Feasibility

Legal Feasibility

The online blood donation system involves collecting and managing user data such as names, personal information and location details. To ensure legal feasibility, the system will stick to relevant privacy and some data encryption methods. This system supports voluntary donation of blood, which is in line with Bhutan's National Blood Policy (2014) that promotes non-remunerated, safe donation practices (Ministry of Health, 2014). Thus, this system ensures that it does not violate towards law.

Contractual feasibility

Since the system is being developed as an academic project and does not require formal agreements or contract with external organizations like hospitals and government agencies. However, after the testing and ready to deploy then the collaboration with hospitals (blood bank) plays an important role to enhance system's functionality and trust.

2.8.6 Political Feasibility

Political feasibility depends on how different groups within the healthcare system react to it. Even though the Hospitals and blood banks support it to make the blood management easier, there will be some staff that resist the fearing technology will be replacing their jobs. If it is following the government policies, they will likely approve it, while donors will be willing to keep their data private. The key is balancing these interests by providing training for staff, ensuring strong data protection, and showing how the system benefits everyone.

Changing how blood and donor information is shared can make some hospitals uncomfortable. They might not want to share their data or may worry about losing control. To make the new system work, it should start with the small hospitals that are willing to try it. Once it shows the good results, others may join. If the rules about the data ownership are clear rules about data ownership and the support from the government authorities will reduce worries. The main goal is to show that working together improves the blood donation process and helps save more lives.

2.9 Management Issues

2.9.1 Team configuration and management

1. **Project Manager**

   Responsibilities:

   i. Manage every stage of the project, from planning to implementation.
   ii. Create project budgets, schedules, and plans.
   iii. Work together with stakeholders and team members to make sure project goals are fulfilled.
   iv. Track developments, evaluate risks. and put mitigation plans into action. Maintain frequent communication red cross club to gather feedback.

   Assigned Individual: **Chogyal Wangdi**

   Email Address: 05240129.jnec@rub.edu.bt

2. **Designer (UX/UI Designer):**

Responsibilities:

i. Create simple user interfaces for administrators, hospitals, and donors.
ii. Design layouts, prototypes, and wireframes for online and mobile platforms.
iii. Ensure the design is user-friendly and aligns with the hospitals and blood banks guidelines.
iv. Collaborate with frontend developers to make designs accessible and responsive.
v. To increase usability and engagement, do user research and testing.

Assigned Individual: Dorji Samdrup

Email Address: 05240133.jnec@rub.edu.bt

3. **Data Analyst:**

Responsibilities:

i. Gather, maintain, and verify donor and blood inventory information.
ii. Determine patterns in donations and produce reports to assist in decision-making.
iii. Create dashboards to track donations and blood stock levels in real time.
iv. Assure privacy, security, and obedience to health requirements.
v. Work together to create data-driven solutions with blood banks, hospitals, and backend developers.

Assigned Individual: Norbu Wangmo

Email Address: 05240141.jnec@rub.edu.bt

4. **Backend Developer:**

Responsibilities:

i. Create and manage safe databases, APIs, and server-side logic applications.
ii. Assure effective data exchange between hospitals, blood banks, and users.
iii. Implement authentication, donor database, and inventory management systems.
iv. Assist frontend developers to ensure a smooth connection.
v. Assure errors correction, data backups, and accordance to health data laws.

Assigned Individual: Sonam Jamtsho

Email Address: 05240159.jnec@rub.edu.bt

Supervision plan

To help the effective development of the Online Blood Donation System, the Scrum Master supervises each stage of the project from the initial planning to the final deployment. The team conducts regular daily meetings to review progress, divide each responsibility, and make key decisions as a group. Team members share updates on their individual tasks either through Google Drive or face-to-face discussions and the centralized Google Sheet is used to monitor deadlines and the task completion.

To make better decisions during the important stages, the team talks with the stakeholders to hear their concerns and get feedback for improvements. This helps the team finish all main parts of the system like donor and recipient features, the user interface, and the database on time, and makes sure the system works well and is easy to use.

2.9.2 Communication Plan

Methods and frequency of communication

Establish various communication channels to ensure effective interaction between stakeholders throughout the project lifecycle. These may include:

i. **Email:** For official announcements, updates, reports and document distribution.

ii. **Collaborative Platforms**: Utilizing platforms like WhatsApp, Google Meeting, Google Drive or Email for real time discussing the progress, addressing issues and file sharing.

iii. **Meetings**: Regular team meetings to discuss work progress, address issues, and brainstorm.

iv. **Project Management Tools**: Implementing project management software such as Microsoft Project, BPP, Jira for tracking, Figma for design collaboration and GitHub for code.

3. Chapter 3: Analysis phase

3.1 Requirement Gathering

Based on an interview with a blood bank manager, it was found that the current blood donor registration process is manual and paper-based. This outdated method leads to frequent delays, especially during emergencies, as the blood bank struggles to reach suitable donors on time. When a patient needs blood urgently, staff resort to using social media platforms such as WhatsApp and Messenger to find donors. However, this process is long and inefficient because even after a donor is found, they still need to fill out manual forms, which causes further delays. According to the manager, there have been five cases where patients died due to delays in arranging blood on time.

Blood stocks are preserved for up to 42 days, after which they expire. To avoid wastage, blood is often sent to other hospitals like those in Mongar and Samdrup Jongkhar before it expires. However, manually managing blood stock and identifying rare blood types is a major challenge. The lack of a centralized digital system makes it hard to search for donors with rare blood groups during emergencies.

Currently, all donor records and blood inventory are maintained on paper, which wastes time and resources. Additionally, an old system called the **ePIS (Electronic Patient Information System)** is used only to store donor history. Unfortunately, the same information that exists on paper is manually entered again into ePIS, creating unnecessary duplication of effort and increasing paper use. Instead of managing two separate systems, the proposed Online Blood Donor System will centralize all data into one platform, reducing redundancy and saving valuable resources.

Overall, an online system would significantly improve the efficiency of the blood donation process by enabling real-time donor tracking, faster identification of rare blood types, and better inventory management. It would also enhance donor recruitment and retention through automated communication and engagement features.

3.1.1 Functional Requirement

1. Donors can register personal details (Name, contact information, blood type, medical history)
2. Donors can schedule, reschedule, or cancel blood donation appointments
3. Donors can view their donation history and eligibility status
4. Donors can receive and respond to blood requests from authorized requesters
5. Donors can update their profile information and availability status
6. Requesters can submit blood requests with detailed requirements (blood type, quantity, urgency)
7. Requesters can attach supporting medical documents for verification
8. Requesters can view the status of their requests
9. Requesters can search for available donors matching specific criteria
10. Requesters can communicate with potential donors through a secure messaging system
11. Admin can verify and approve/reject donor eligibility based on health criteria
12. Admin can manage (create, edit, deactivate) accounts for donors, requesters, and staff
13. Admin can view and manage all donation appointments
14. Admin can generate reports on donations, requests, and system usage
15. Admin can verify and approve/reject blood requests from requesters

3.1.2 Non-functional Requirement

1. Performance: the system should be responsive and able to handle multiple simultaneous donations without significant delays.
2. Scalability: it should be able to accommodate a growing number of donors and donations over time without degradation in performance.
3. Reliability: the system should be highly available and reliable, ensuring that donation records are not lost or corrupted.
4. Data backup and recovery: the system should implement regular data backup and have procedures in data recovery in case of system failures or data lose.
5. Usability: the system should be easy to use, even for non-technical people. It should work well on phones and computers.
6. Compliance: the system will follow hospital care policy laws.

3.2 Requirement Structuring

3.2.1 Process Modeling

DFD Diagram

i. Online Blood Donation System, DFD level 1

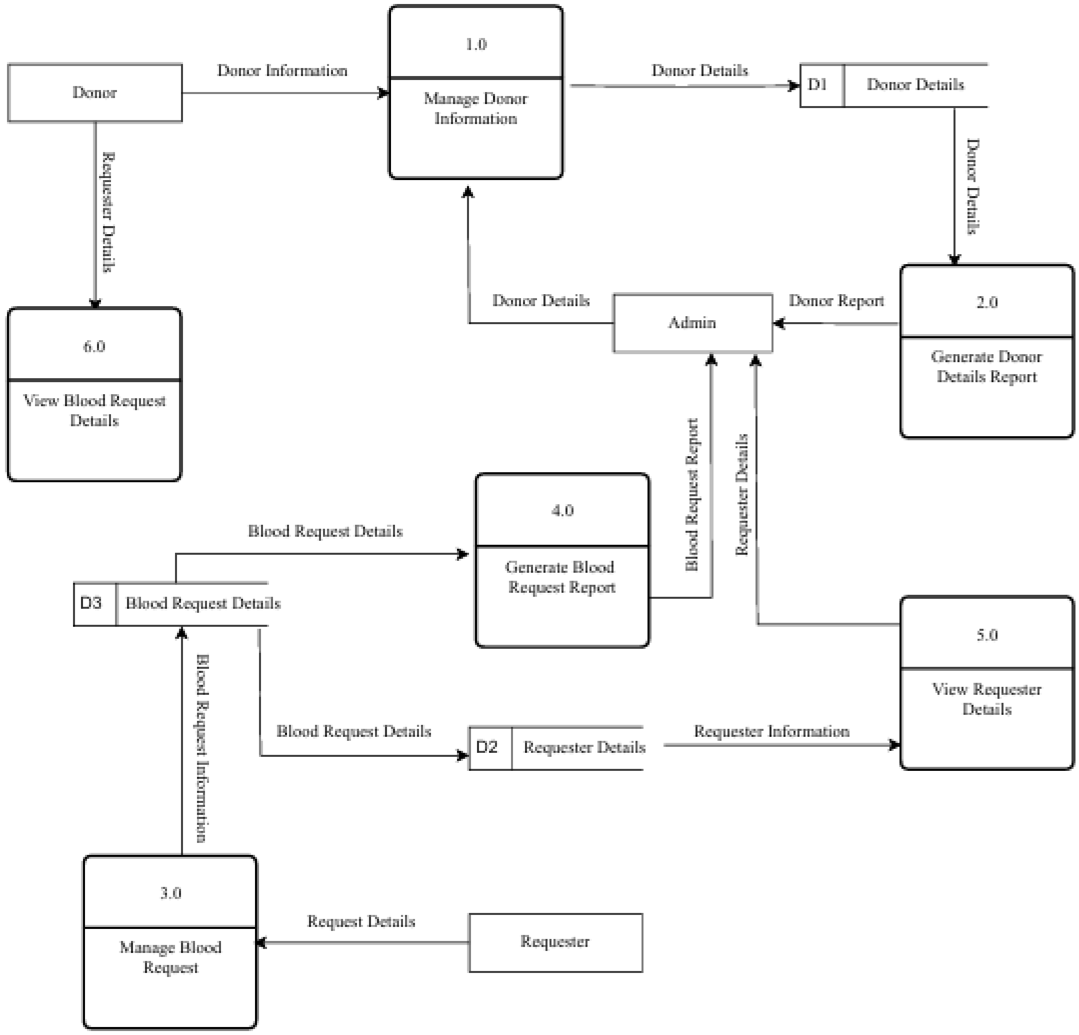


Figure 3.1 Level 1 DFD

ii. DFD Level 1 Process 1.0: Manage Donor Information

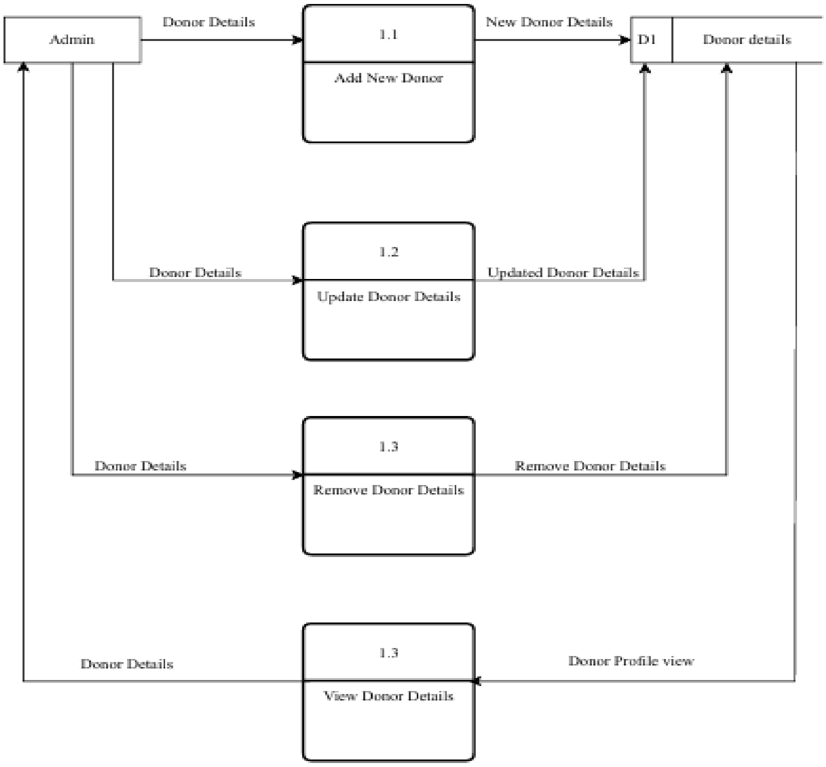


Figure 3.2 DFD Level 1 Process 1.0: Manage Donor Information

iii. DFD Level 1 Process 3.0: Manage Blood Request

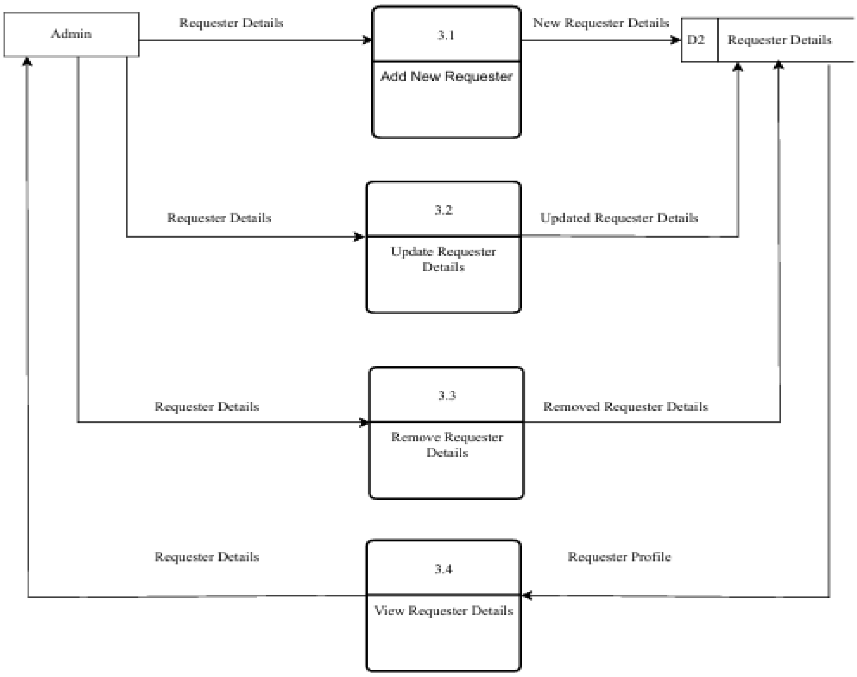


Figure 3.3 DFD Level 1 Process 3.0: Manage Blood Request

iv. DFD Level 1 Process 4.0: Generate Blood Request Report

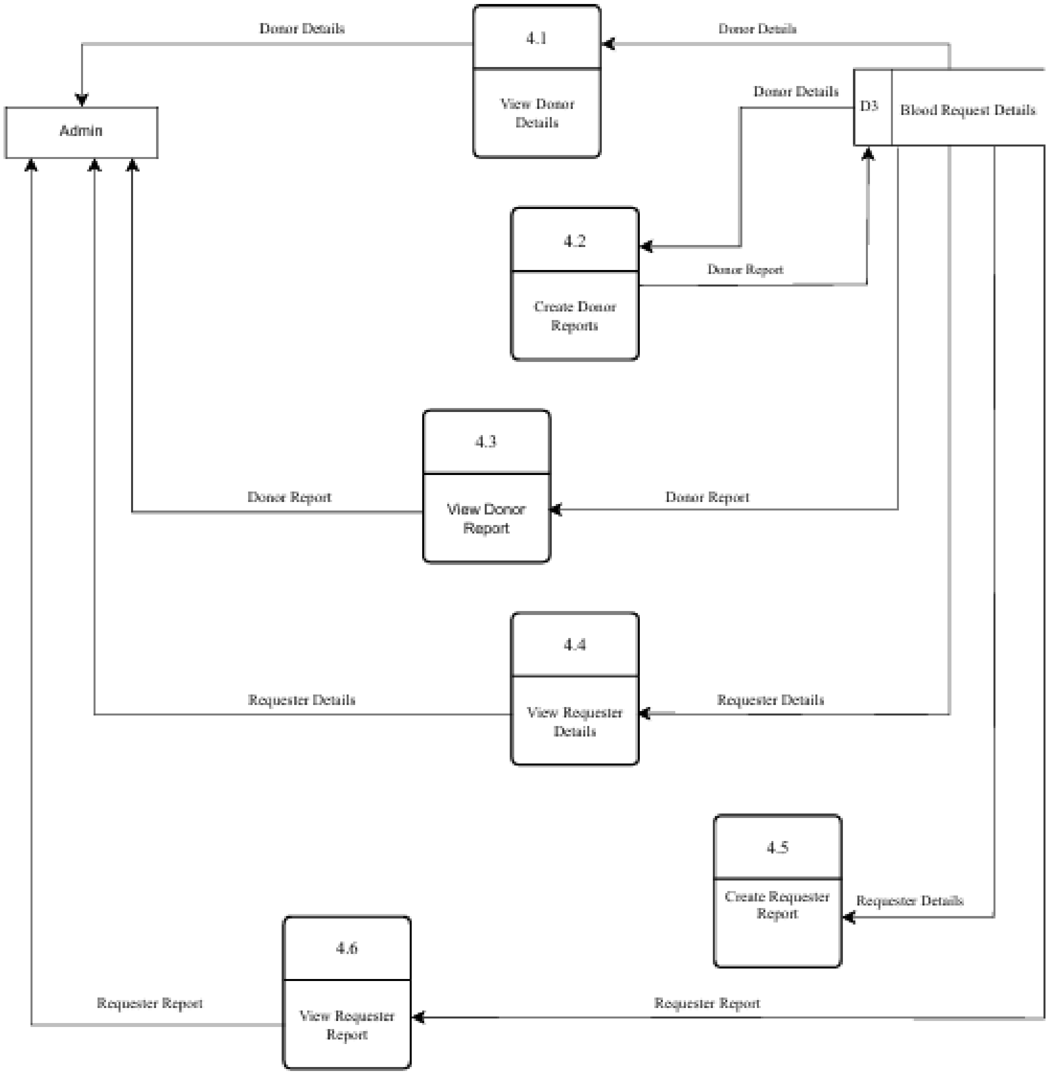


Figure 3.4 DFD Level 1 Process 4.0: Generate Blood Request Report

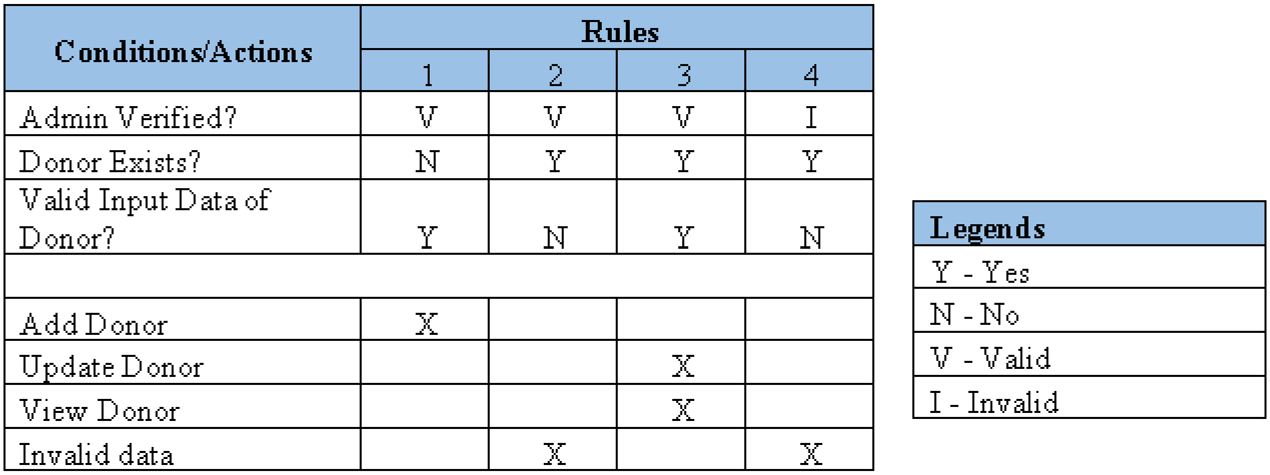
3.2.2 Logical Modeling

i. Structure English for process 3.0: Manage Blood Request

```
IF user_selection= “Add new requester” THEN
    ASK for requester details
    CHECK if blood type is valid
    IF blood type is valid THEN
    SAVE the requester details
    ELSE
    SHOW an error message
    END IF
END IF
IF user_selection= “update requester” THEN
    GET the current requester details
    ASK for updated details
    CHECK if updated details are valid
    IF details are valid THEN
    SAVE the updated requester details
    ELSE
    SHOW an error message
    END IF
END IF
IF user_selection= “view requester” THEN
    GET the requester details
    ASK user to confirm deletion
    IF user to confirms THEN
    DELETE the requester details
    ELSE
    CANCEL the deletion
    END IF
END IF
IF user_selection= “view requester” THEN
    GET the requester details
    SHOW the requester profile
END IF
```

ii. Decision Table for Process 1.0: Manage Donor Information

Table 3.1Decision Table for process 1.0: Manage Donor Information



Conditions/Actions	Rules			
	1	2	3	4
Admin Verified?	V	V	V	I
Donor Exists?	N	Y	Y	Y
Valid Input Data of Donor?	Y	N	Y	N
Add Donor	X			
Update Donor			X	
View Donor			X	
Invalid data		X		X

Legends
Y - Yes
N - No
V - Valid
I - Invalid

iii.Decision Tree for process 4.0: Generate Blood Request Report

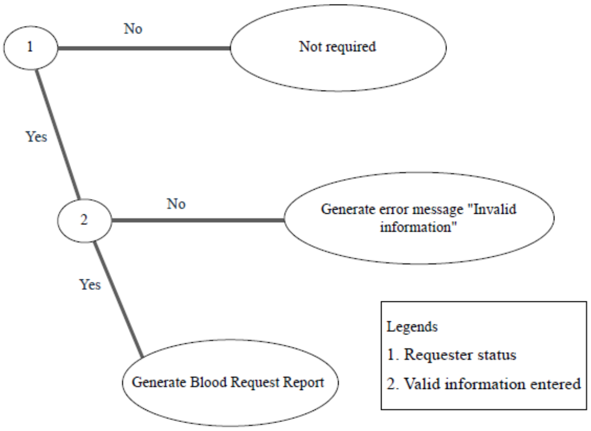

1
No
Not required
Yes
2
No
Generate error message "Invalid information"
Yes
Generate Blood Request Report
Legends
1. Requester status
2. Valid information entered

Figure 3.5 Decision Tree for process 4.0: Generate Blood Request Report

3.2.3 UML-Based Structuring

Use-case Diagram

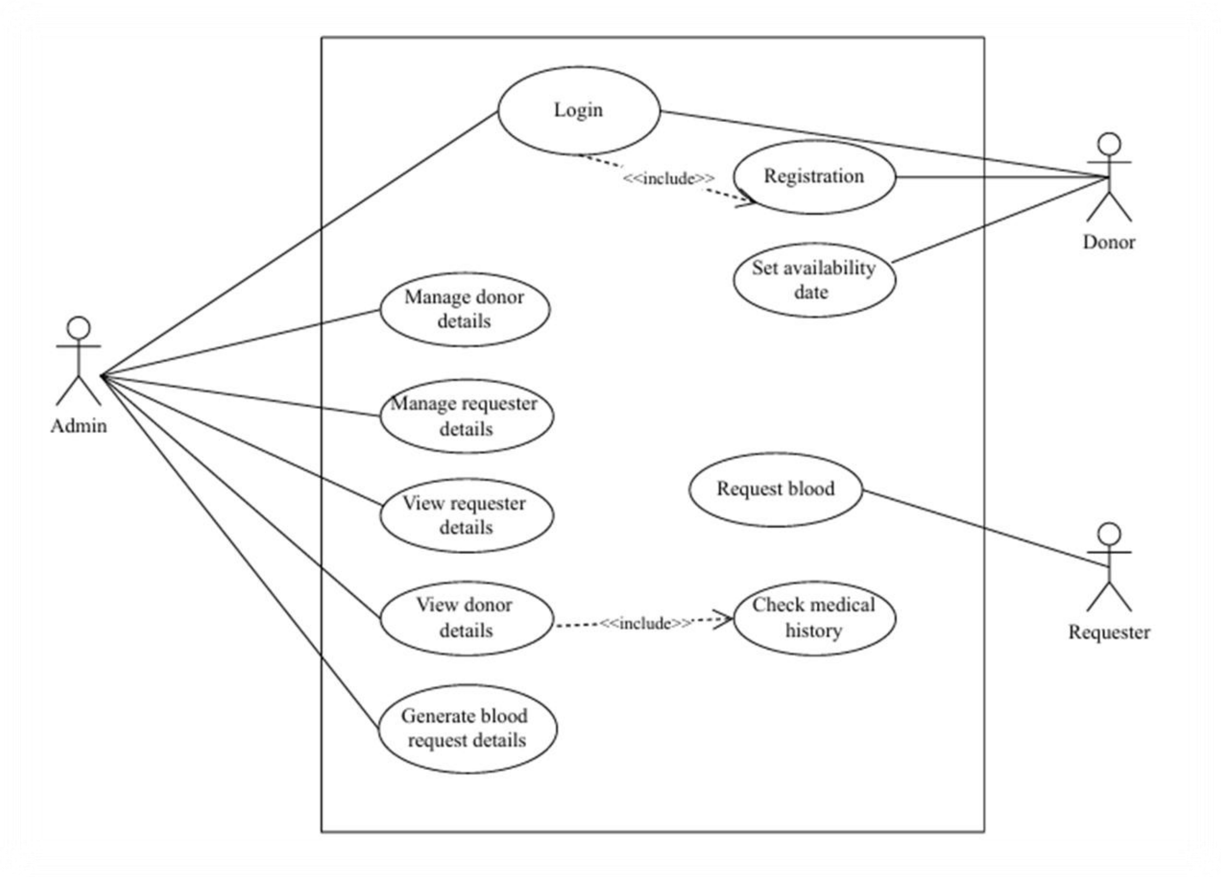


Figure 3.6 Use Case Diagram

Table 3.2 use cases explanation

Use case	Actors	Description	Precondition	Main flow	Alternative flow	Post condition
Registration	Donor, Admin	Allows new users to create an account in the system.	User is not registered.	1. User selects Registration 2. Enters details 3. Submits form	1. Invalid details entered 2. System prompts for correction	User account is created.
Login	Donor, Admin	Authenticates users to access the system.	User has an account.	1. User enters credentials 2. System verifies	1. Wrong credentials 2. System shows error	User logged in.
Set availability date	Donor	Allows donors to set a date when they are available to donate.	User is logged in.	1. Donor selects date 2. Submits availability	1. Invalid date format 2. Prompt to re-enter	Availability is recorded.
Request blood	Requester	Allows a requester to request blood.	User is logged in.	1. User selects blood type 2. System checks medical history 3. Submits request	1. Medical history not approved 2. Request denied	Blood request is recorded.
Manage donor details	Admin	Admin can add, update, or delete donor information.	Admin is logged in.	1. Admin views donor list 2. Modifies details	1. Invalid data 2. System prompts for correction	Donor details updated.
Manage requester details	Admin	Admin can add, update, or delete requester information.	Admin is logged in.	1. Admin views requester list 2. Modifies details	1. Invalid data 2. System prompts for correction	Requester details updated.
View donor details	Admin	Allows admin to view donor information.	Admin is logged in.	1. Admin selects donor list	None	Donor details displayed.
Generate blood request details	Admin	Generates a report of blood request activities.	Admin is logged in.	1. Admin selects report option	1. No data available 2. Error message shown	Blood request report generated.
Check medical history	System (included)	Validates the medical history of requester.	Blood request initiated.	1. System retrieves and checks medical records	1. Records not found 2. Show error	Medical history verified or rejected.

Sequence Diagram

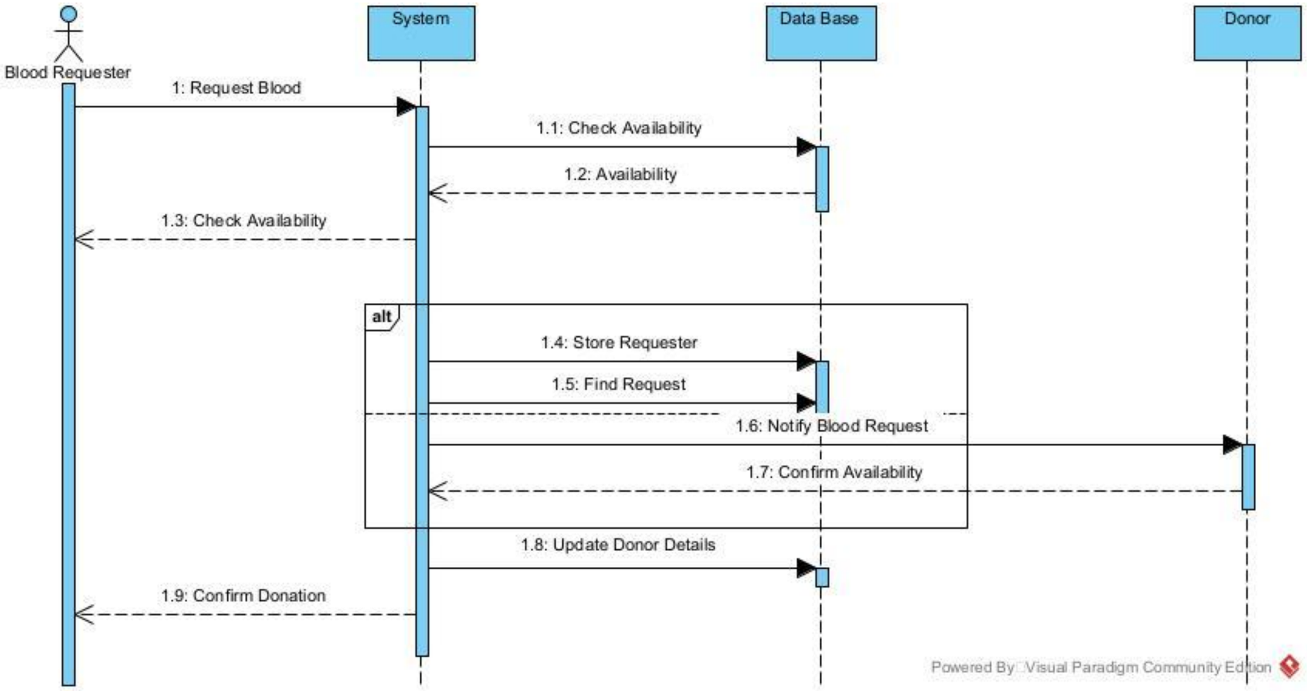


Figure 3.7Sequence Diagram of Blood Requester

This sequence diagram shows the steps involved in requesting blood from a system. It starts with the requester checking blood availability (1.3), which triggers the system to verify availability (1.1) and respond (1.2). The database then stores the requester's details (1.4), finds the request (1.5), notifies about the blood request (1.6), and confirms availability (1.7). Finally, the donation is confirmed (1.9), and the donor's details are updated (1.8). The diagram outlines the flow of interactions between the requester, system, and database to complete the blood request process.

3.3 Data Conceptual Modeling

3.3.1 Entity-Relationship (ER) Diagram

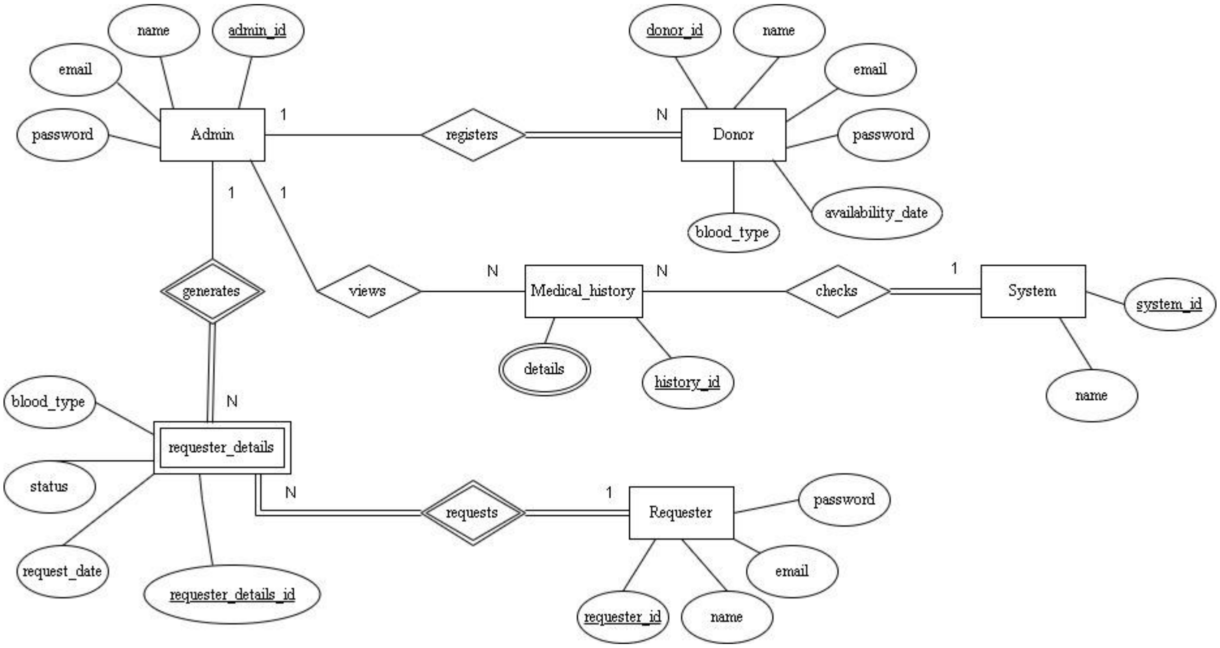


Figure 3.8 ERD Diagram

Data Dictionary

Admin Table

Table 3.3 Admin Table

Field	Type	Description
admin_id (PK)	INT NOT NULL	This field stores as a unique identifier for each user. It is an integer and is required (NOT NULL) to have a value.
name	VARCHAR (50) NOT NULL	This field stores the name with a variable length up to 50 characters. NULL values allowed.
password	VARCHAR (100) NOT NULL	This field stores the hashed password with a variable length string of up to 100 characters and it is required.
email	VARCHAR (100) UNIQUE	This field stores the email address (max 100 characters). This field allows NULL values allowed.

Donor Table

Table 3.4 Donore table

Field	Type	Description
donor_id (PK)	INT NOT NULL	This field stores as a unique identifier for each donor. It is an integer and is required (NOT NULL) to have a value.
password	VARCHAR (100) NOT NULL	This field stores the password with a variable length string of up to 100 characters and it is required.
name	VARCHAR (200) NOT NULL	This field stores the name with a variable length string of up to 200 characters and it is required to have a value.
email	VARCHAR (100) NOT NULL	This field stores the email address (max 100 characters). This field is required.
blood_type	VARCHAR (5) CHECK	This field stores blood type (such as A+, B-,0+, etc).
availability_date	DATE NULL	This field stores the next available date for donation. It is optional (NULL allowed).

Requester Table

Table 3.5 Requester Table

Field	**Type**	**Description**
requester_id (PK)	INT NOT NULL	This field stores as a unique identifier for each user. It is an integer and is required (NOT NULL) to have a value.
password	VARCHAR (100) NOT NULL	This field stores the password with a variable length string of up to 100 characters and it is required.
name	VARCHAR (200) NOT NULL	This field stores the requester's name with a variable length string of up to 200 characters and it is required to have a value.
email	VARCHAR (100) UNIQUE	This field stores the email address (max 100 characters). This field is required.

Requester_details Table

Table 3.6 Rrquester_details table

Field	Type	Description
requester_details id (PK)	INT NOT NULL	This field stores a unique slot identifier for requester details. It is an integer and is required (NOT NULL) to have a value.
request_date	DATE NULL	It stores the date when the blood request was made. (Optional)
blood_type	VARCHAR (5) NOT NULL	This field stores required blood type (e.g., "O-") and it is required.
status	VARCHAR (20) NOT NULL	This field stores request status (e.g., "Pending").
requester_id (FK)	INT NOT NULL	This field is linked to requester and it is required.

Medical_history Table

Table 3.7 Medical_history Table

Field	Type	Description
history_id (PK)	INT NOT NULL	This field stores a unique medical records. It is an integer and is required (NOT NULL) to have a value.
details	TEXT	It stores the health conditions, allergies.
donor_id (FK)	INT	This field links to donor
requester_id (FK)	INT NOT NULL	This field is linked to requester and it is required.

System Table

Table 3.8 System Table

Field	Type	Description
system_id (PK)	INT NOT NULL	This field stores a unique system identifier. It is an integer and is required (NOT NULL) to have a value.
name	VARCHAR (50)	This field stores system name (such as “Donation_Tracker”)

Relationships Constraints

Table 3.9 Relationships Constraints

Relationship	Cardinality	Rules
Admin → Donor	1:N	One admin can register many donors.
Donor → Medical_history	1:1	Each donor must have one medical record.
Requester → Blood_Request	1:N	One requester can make multiple blood requests.
System → Medical_history	1:1	System validates each medical record individually.

System-Enforced Constraints

1. Password Encryption:

   All user passwords (for Admin, Donor, Requester) will be hashed before being stored in the database.

   Passwords must:

   - Be at least 8 characters long.
   - Include a mix of uppercase letters, lowercase letters, numbers, and symbols.
   - Be validated on both the client side and server side to prevent weak entries.

2. Blood Type Validation:

   Blood types will be validated using regular expressions to ensure correct formatting.

   Regex pattern: ^(A|B|AB|O)[+-]$

   This ensures that only valid blood types (e.g., A+, O−, AB−) are accepted and stored.

3. Email Format Validation:

   Email addresses must follow standard formats and be unique within the system.

   Validation rule:

   - Must contain @ and a valid domain name (e.g., user@example.com)

4. Phone Number Validation:

   Phone numbers will:

   - Be numeric.
   - Match standard formats (e.g., 8 digits).

5. Date Validation:

   All date inputs (such as availability date, request date) will be checked for:

   - Proper format (e.g., YYYY-MM-DD).
   - Logical correctness (e.g., future dates only for donor availability).
   - No acceptance of invalid or expired dates for requests.

Normalization

- Relationship between Admin and Donor (1 to many)

Admin table			
Admin_id (PK)	Name	Email	Password

Donor table					
Donor_id (PK)	Name	Email	Availability_date	Password	Blood_type

Donor_id (PK)	Name	Email	Availability_date	Password	Blood_type	Admin_id (FK)

- Relationship between Admin and Medical_history (1 to many)

Admin table			
Admin_id (PK)	Name	Email	Password

Medical_history table	
History_id (PK)	details

Details	
History_id	Medical_condition

History_id (PK)	Admin_id (FK)

- Relationship between Admin and Requester_details (1 to many)

Table 3.10 Admin table

Admin table			
Admin_id (PK)	Name	Email	Password

Table 3.11 Requester_details table

Requester_details table			
Requester_details_id (PK)	Blood_type	Status	Request_date

Requester_details_id (PK)	Blood_type	Status	Request_date	Admin_id (FK)

- Relationship between Requester and Requester_details (1 to many)

Table 3.12 Requester table

Requester			
Requester_id (PK)	Username	Email	Password

Table 3.13 Requester_details table

Requester_details table			
Requester_details_id (PK)	Blood_type	Status	Request_date

Requester_details_id (PK)	Blood_type	Status	Request_date	Requester_id (FK)

- Relationship between Medical_history and system (1 to 1)

Table 3.14Medical_history table

Medical_history table	
History_id (PK)	Details

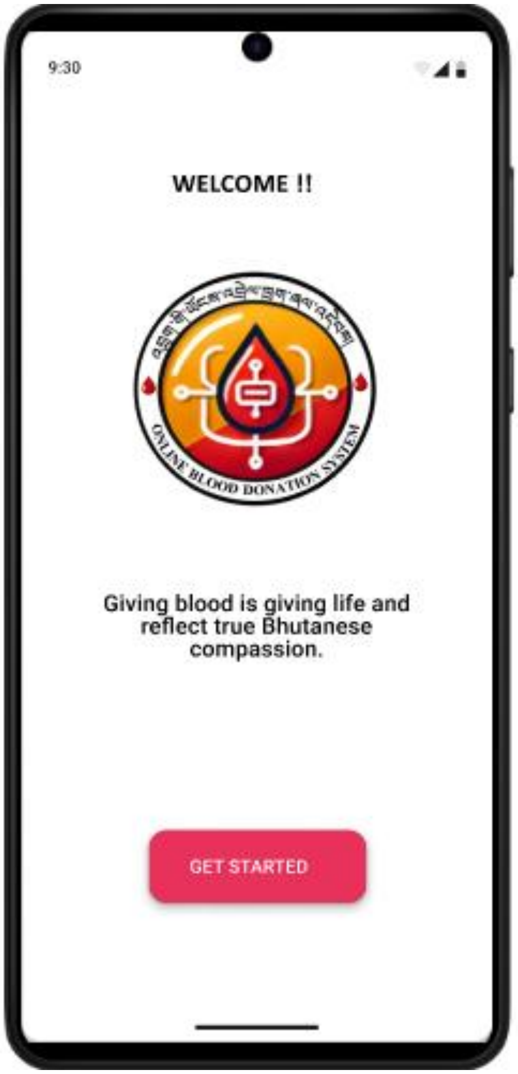
Table 3.15 Medical_history table

System table	
System_id (PK)	Name

System_id (PK)	Name	History_id (FK)

4. Chapter 4: Design Phase

The design phase of the Online Blood Donation System focused on creating a simple, efficient platform to connect blood donors with recipients in need. We prioritized user-friendly interfaces with clear buttons like "Donate" and "Request Blood," ensuring even first-time users can navigate easily. The system accommodates both donors (who can register, view requests, and log donations) and recipients (who can post needs and find nearby donors). Below, each design image is described
in detail, highlighting the key features and elements included.



Giving blood is giving life and reflect true Bhutanese compassion.

Figure 4.2 Welcome Screen

The image shows the welcome screen of a blood donation app. At the top, a motivational message ("Giving blood is giving life") encourages user engagement, while a prominent "GET STARTED" button initiates the donation process.

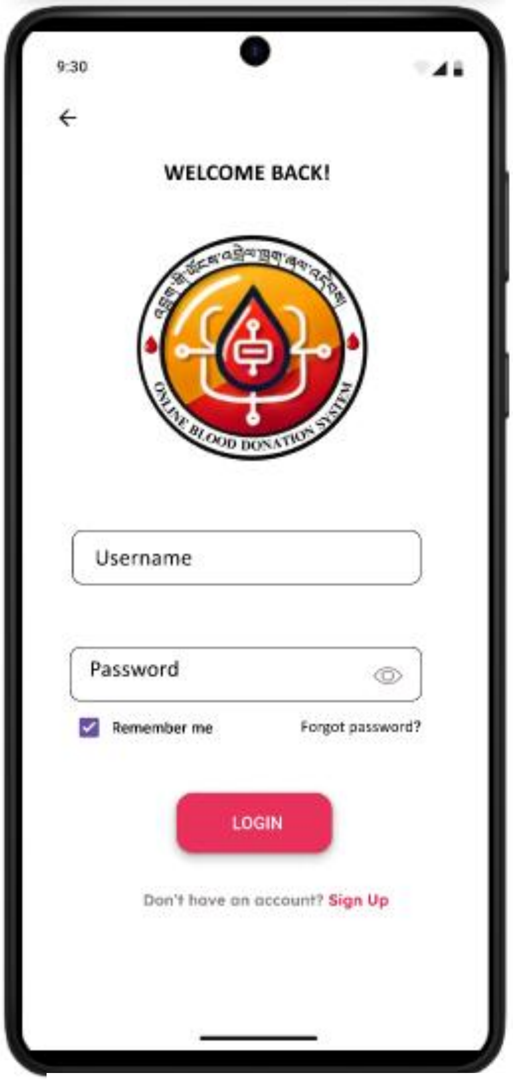


Figure 4.1 sign up page

The image shows the login screen of a blood donation app. The lower section provides a login area for returning users with username/password fields and a "Sign Up" option for new users, creating a clear entry point for both donor registration and account access.

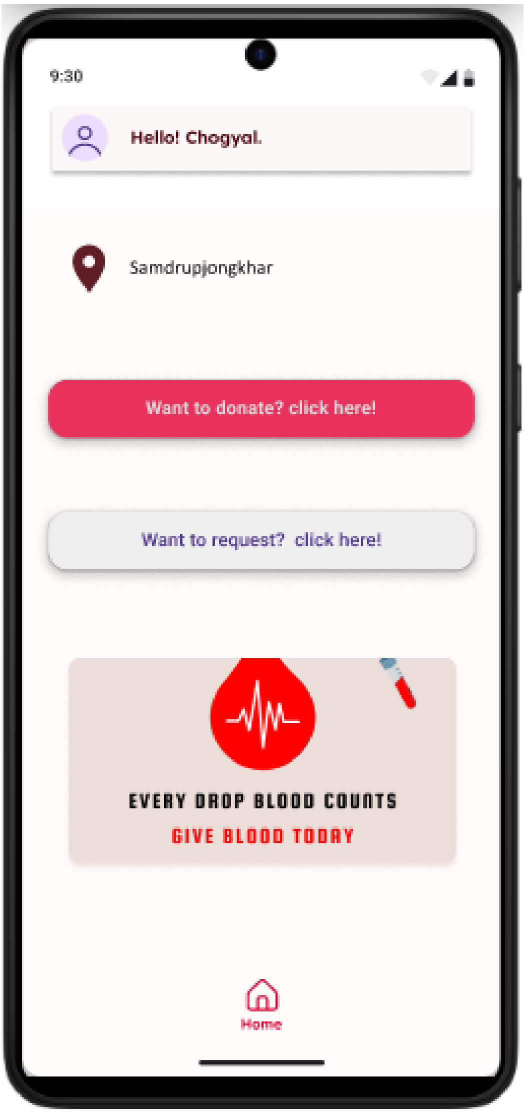


Figure 4.4 Main Dashboard

This interface appears to be the main dashboard of a blood donation application. The screen provides two clear call-to-action buttons: "Want to donate? click here!" for users interested in donating blood and "Want to request? click here!" for those needing blood donations. The bold motivational messages "EVERY DROP BLOOD COUNTS" and "GIVE BLOOD TODAY" reinforce the app's purpose of encouraging life-saving donations.

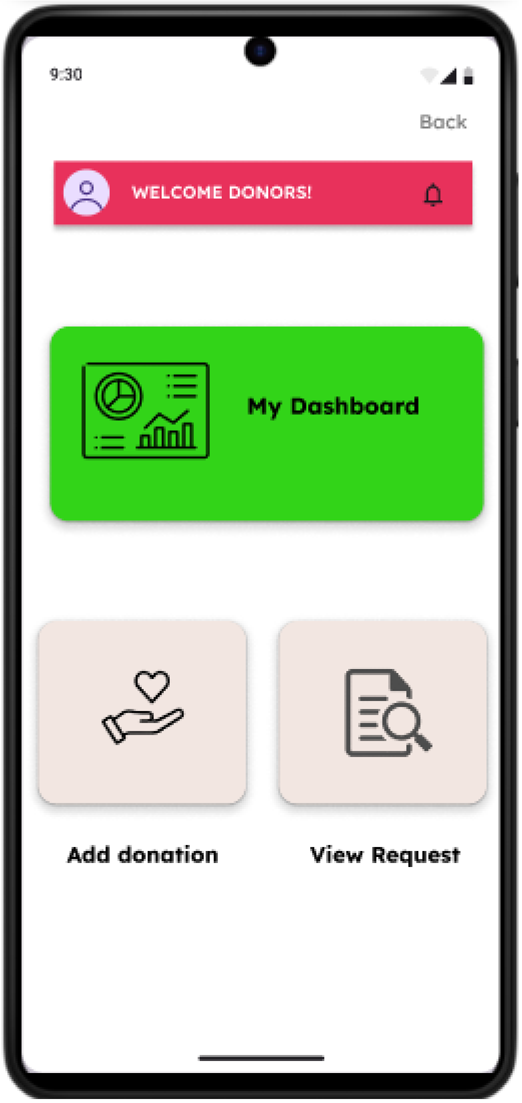
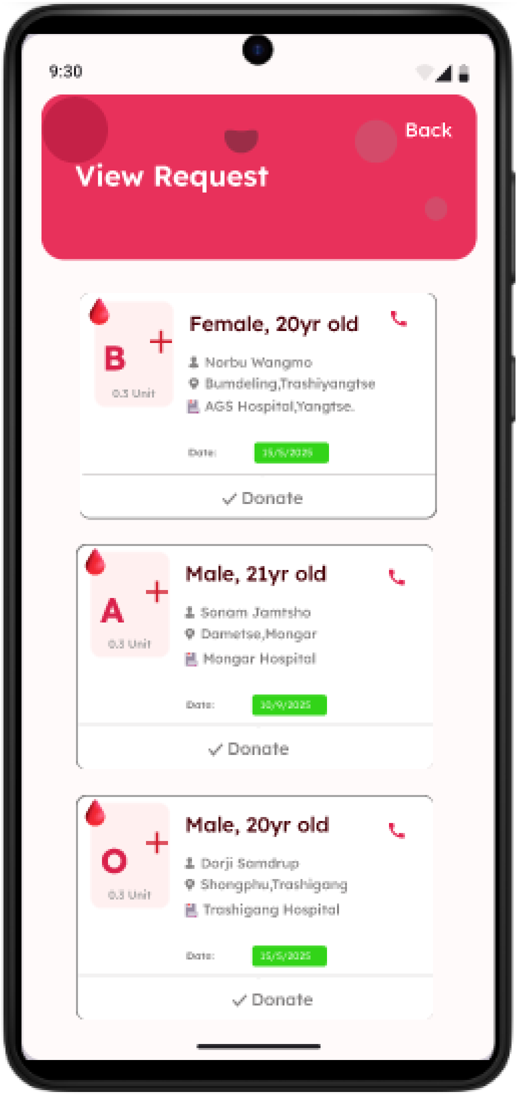


Figure 4.3 Donor Welcome Screen

This interface serves as the donor welcome screen in a blood donation application. It greets users with a prominent "WELCOME DONORS!" header and provides three key options: "NOTIFICATION" to check alerts or requests, "Add donation" for donors to log new donations, and "View Request" to see current blood needs. The inclusion of a "Back" button at the top allows for easy navigation to previous screens. The minimalist design focuses on core donor actions, making it straightforward for users to either contribute blood or respond to urgent requests.



9:30

View Request

Back

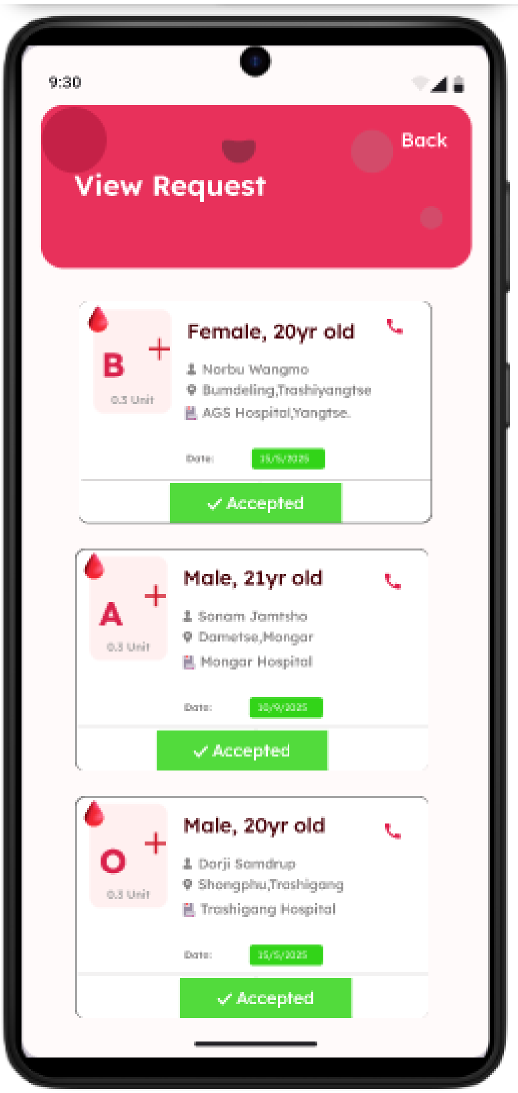
B+ 0.3 Unit — Female, 20yr old — Norbu Wangmo — Bumdeling,Trashiyangtse — AGS Hospital,Yangtse. — Date: 15/5/2025 — ✓ Donate

A+ 0.3 Unit — Male, 21yr old — Sonam Jamtsho — Dametse,Mongar — Mongar Hospital — Date: 10/9/2025 — ✓ Donate

O+ 0.3 Unit — Male, 20yr old — Dorji Samdrup — Shongphu,Trashigang — Trashigang Hospital — Date: 15/5/2025 — ✓ Donate

Figure 4.5 Blood Request Listing Screen

This interface displays active blood donation requests with key details: patient's blood type (B+), gender/age (Female, 20yr), name (Norbu Wangmo), location (Bumdeling), hospital (AGS Hospital), and request date (15/6/2025). Each entry has a "Donate" button for quick response, organized in a clean vertical list for easy scanning.



9:30

View Request

Back

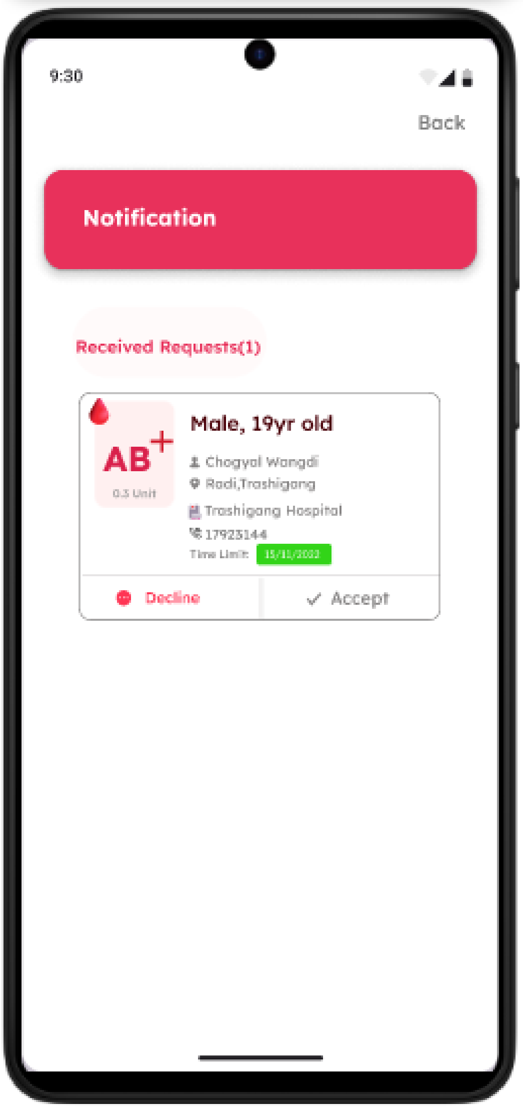
B+ 0.3 Unit — Female, 20yr old — Norbu Wangmo — Bumdeling,Trashiyangtse — AGS Hospital,Yangtse. — Date: 15/5/2025 — ✓ Accepted

A+ 0.3 Unit — Male, 21yr old — Sonam Jamtsho — Dametse,Mongar — Mongar Hospital — Date: 10/9/2025 — ✓ Accepted

O+ 0.3 Unit — Male, 20yr old — Dorji Samdrup — Shongphu,Trashigang — Trashigang Hospital — Date: 15/5/2025 — ✓ Accepted

Figure 4.6 Accepted Requests Screen

Shows previously accepted donations, formatted with: blood type (A+), patient info (Sonam Jamtsho, 21yr), location/hospital details, and acceptance date (25/01/2022). The "Accepted" tags and consistent numbering (1-3) create a structured record of fulfilled requests.



AB+
0.3 Unit
Male, 19yr old
Chogyal Wongdi
Radi,Trashigang
Trashigang Hospital
17923144
Time Limit: 15/11/2023

Figure 4.8 New Request Notification

Highlights an urgent blood need (AB+ male, 19yr) with critical details: requester name (Chogyal Wongdi), contact (17923144), location (Redi), hospital (Trashigang), and expiration date (18/18/2023).

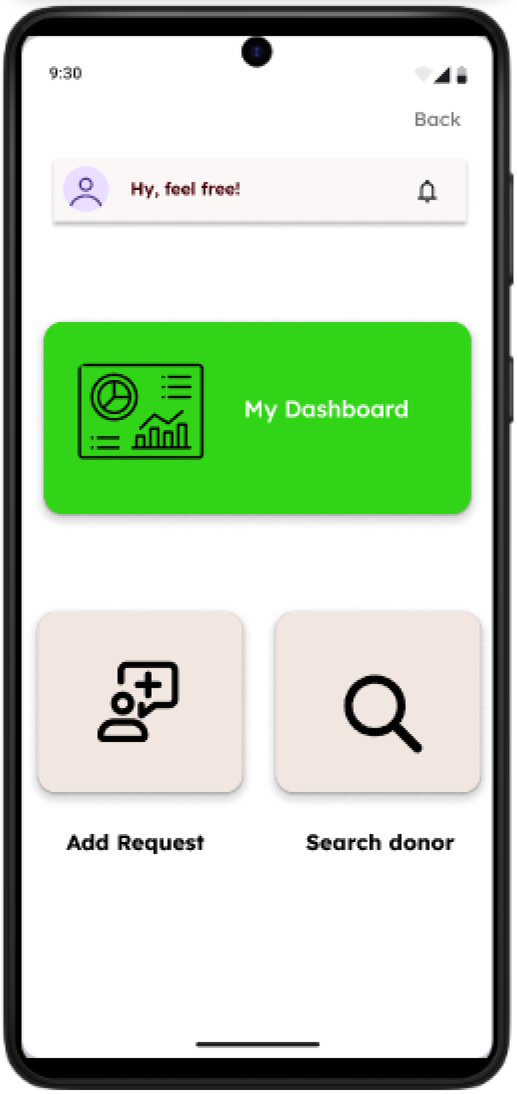


Figure 4.7 Requester Screen

This interface displays the requester screen where two clearly labeled buttons – "Add Request" and "Search donor", providing immediate access to the app's core functions.

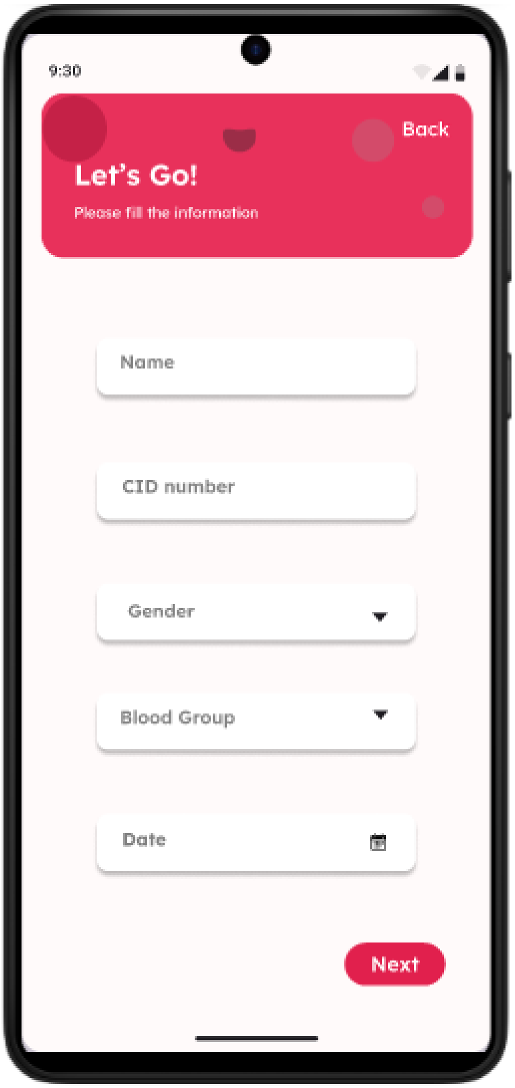


Figure 4.10 Blood Request Form Interface

The blood request form appears when users click "Add Request", guiding them through a simple two-step process to submit their needs. The first screen collects personal details like name, CID number, gender, blood group, and date, with a "Next" button to proceed.

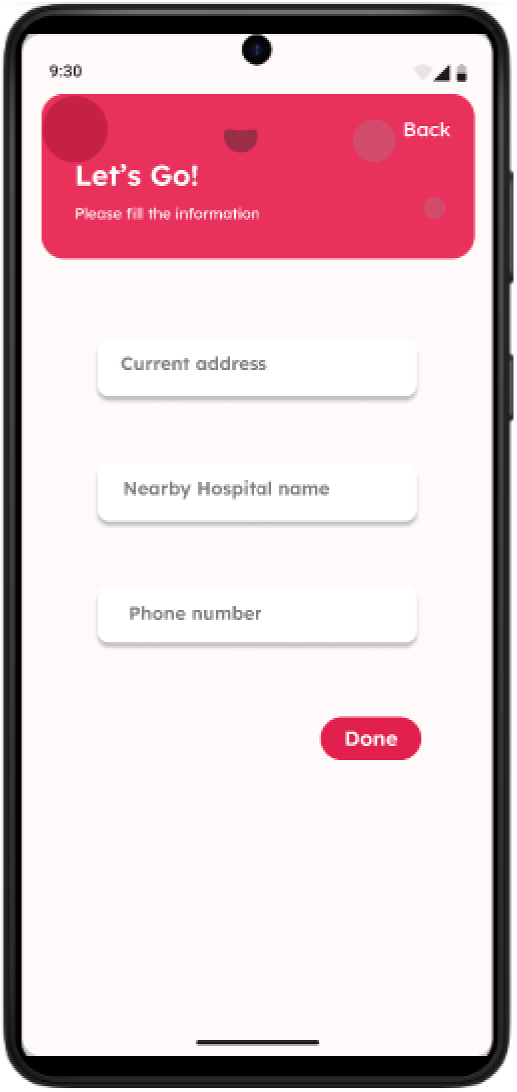
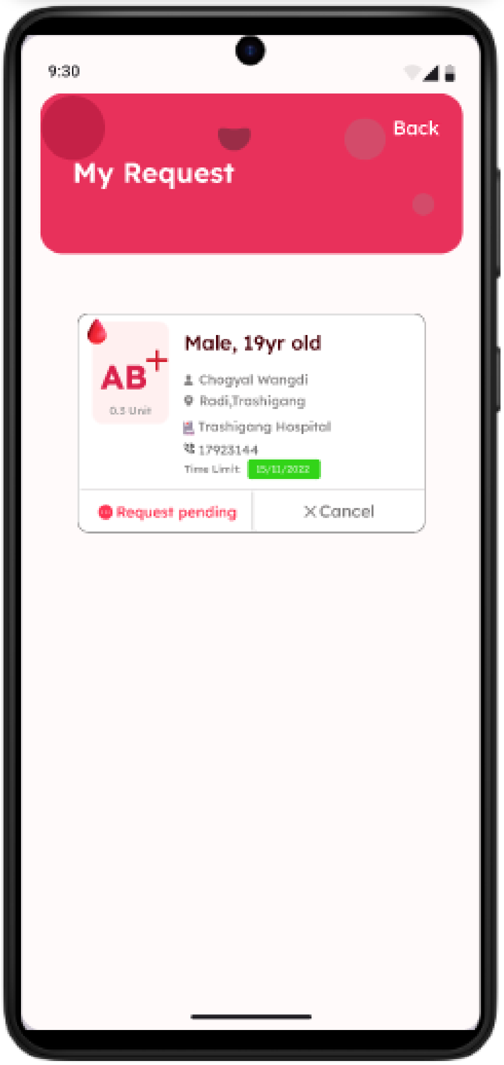


Figure 4.9 Blood Request Form Interface

The second screen requester's location and contact information, including current address, nearby hospital name, and phone number, finalized with a "Done" button.



9:30

Back

My Request

Male, 19yr old

AB+

0.5 Unit

Chogyal Wangdi

Radi,Trashigang

Trashigang Hospital

17923144

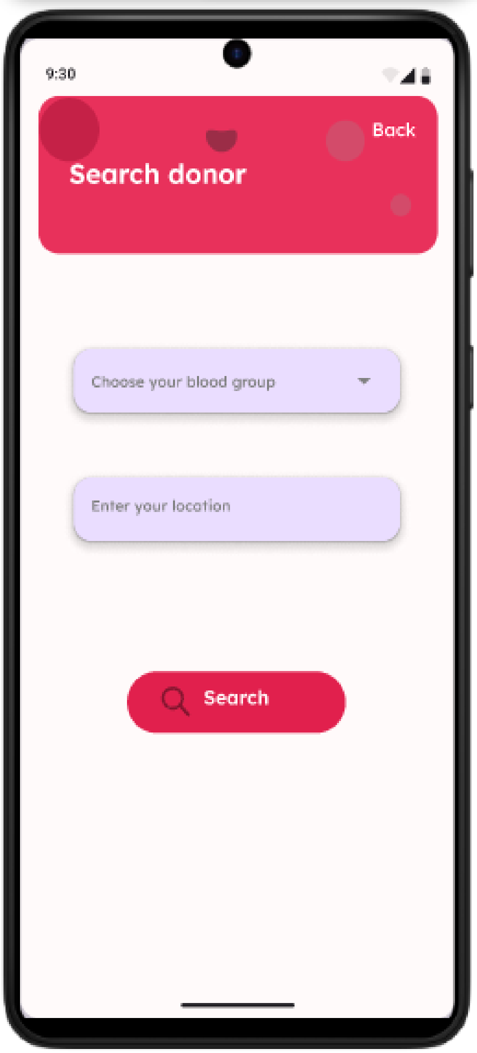
Time Limit: 15/11/2022

Request pending

× Cancel

Figure 4.12 Pending Blood Request

This screen shows a pending blood request in the donation app, displaying all critical details for potential donors. At the bottom, the "Request pending" status is shown alongside a "Cancel" option and a "Back" button for navigation.



9:30

Back

Search donor

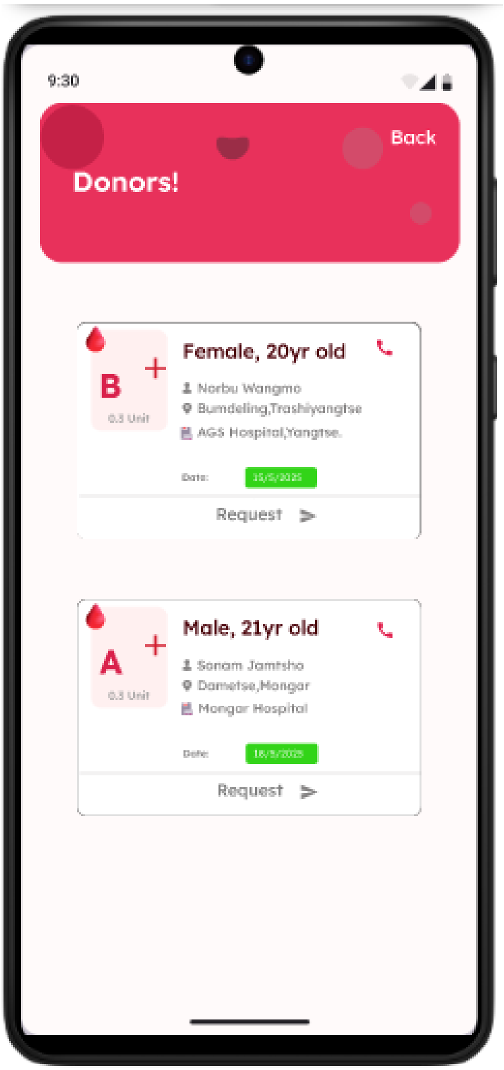
Choose your blood group

Enter your location

Search

Figure 4.11 Donor Search Interface

This screen offers a clean donor search feature with just two inputs: blood type selection and location entry, plus a prominent search button. The back button (top-left) and timestamp maintain app consistency while the minimalist design ensures quick, emergency-ready searching.



9:30

Back

Donors!

B+
0.3 Unit

Female, 20yr old

Norbu Wangmo
Bumdeling,Trashiyangtse
AGS Hospital,Yangtse.

Date: 15/5/2025

Request

A+
0.3 Unit

Male, 21yr old

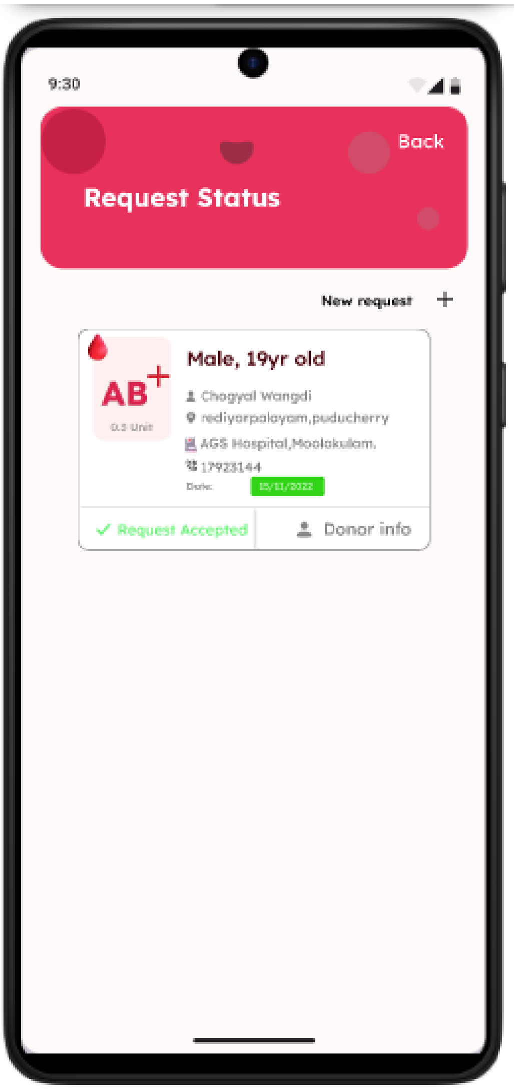
Sonam Jamtsho
Dametse,Mongar
Mongar Hospital

Date: 16/5/2025

Request

Figure 4.14 List of Eligible Blood Donors

This screen shows available blood donors in a clear list format. Each entry includes the donor's name, age/gender, location/hospital details, and last donation date. A prominent "Request" button appears below each profile for quick contact initiation, while the standard "Back" button ensures consistent navigation. The minimalist design prioritizes essential information, enabling recipients to efficiently find and connect with matching donors.



9:30

Back

Request Status

New request +

AB+
0.3 Unit

Male, 19yr old

Chogyal Wangdi
rediyarpalayam,puducherry
AGS Hospital,Moolakulam.
17923144

Date: 15/11/2022

✓ Request Accepted

Donor info

Figure 4.13 Status of a Blood Request

This interface displays the status of a blood donation request, split into two clear sections. The top shows a new request with complete details including location, hospital, and contact number. The bottom section confirms the request has been accepted with a placeholder for donor information to be populated.

Admin Panel

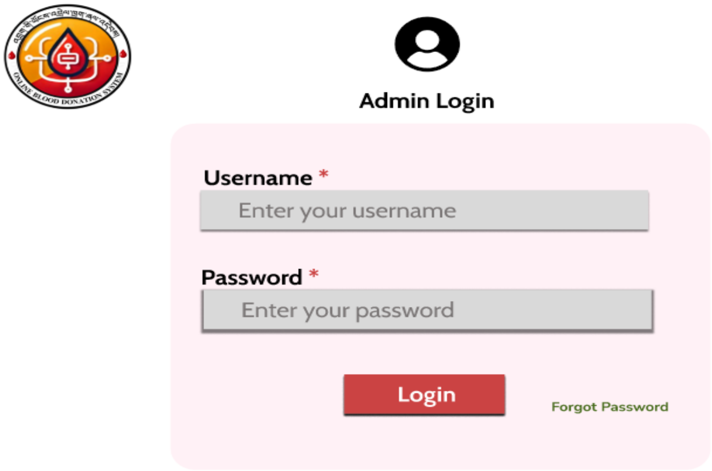
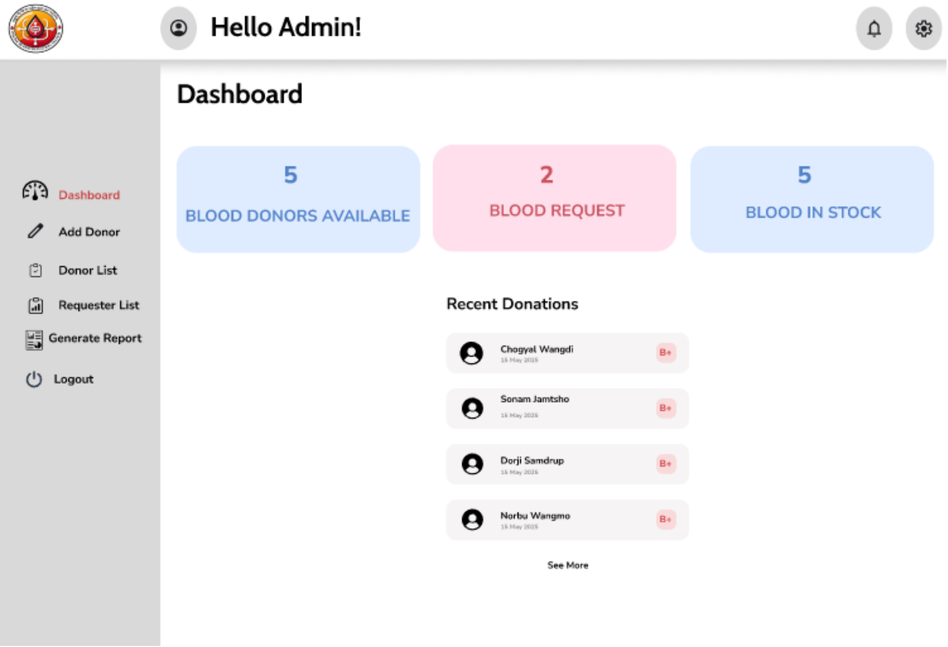


Figure 4.16 Admin Login Screen

This admin login screen features secure username/password fields (marked as required) with placeholder hints, a prominent Login button, and Forgot Password option.



5 BLOOD DONORS AVAILABLE

2 BLOOD REQUEST

5 BLOOD IN STOCK

Recent Donations

Chogyal Wangdi 15 May 2025	B+
Sonam Jamtsho 15 May 2025	B+
Dorji Samdrup 15 May 2025	B+
Norbu Wangmo 15 May 2025	B+

See More

Figure 4.15 Admin Dashboard

This admin dashboard provides a comprehensive overview of the blood donation system, featuring quick-access tabs for managing donors, requesters, and reports. Key metrics displaying available donors, pending requests, and blood units in stock, while a "Recent Donations" section track. The clean layout combines actionable controls (Add Donor, Generate Report) with real-time system data for efficient management.

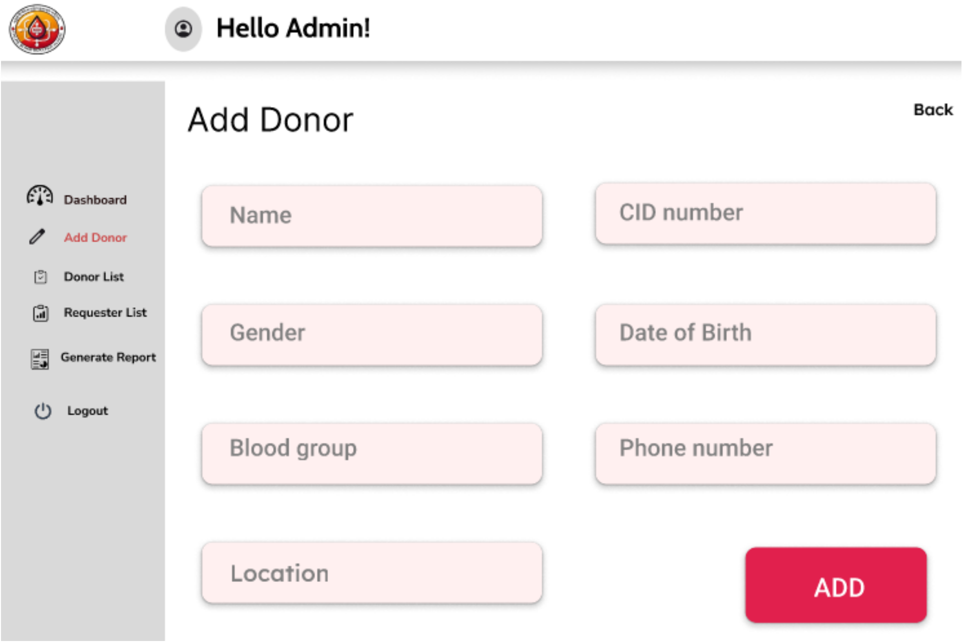


Figure 4.18 Donor Registration Interface for Administrators

This is the donor registration interface for administrators in the blood donation system.This form collects essential donor details: name, CID number, gender, date of birth, blood group, phone number, and location. A prominent "ADD" button submits the information.

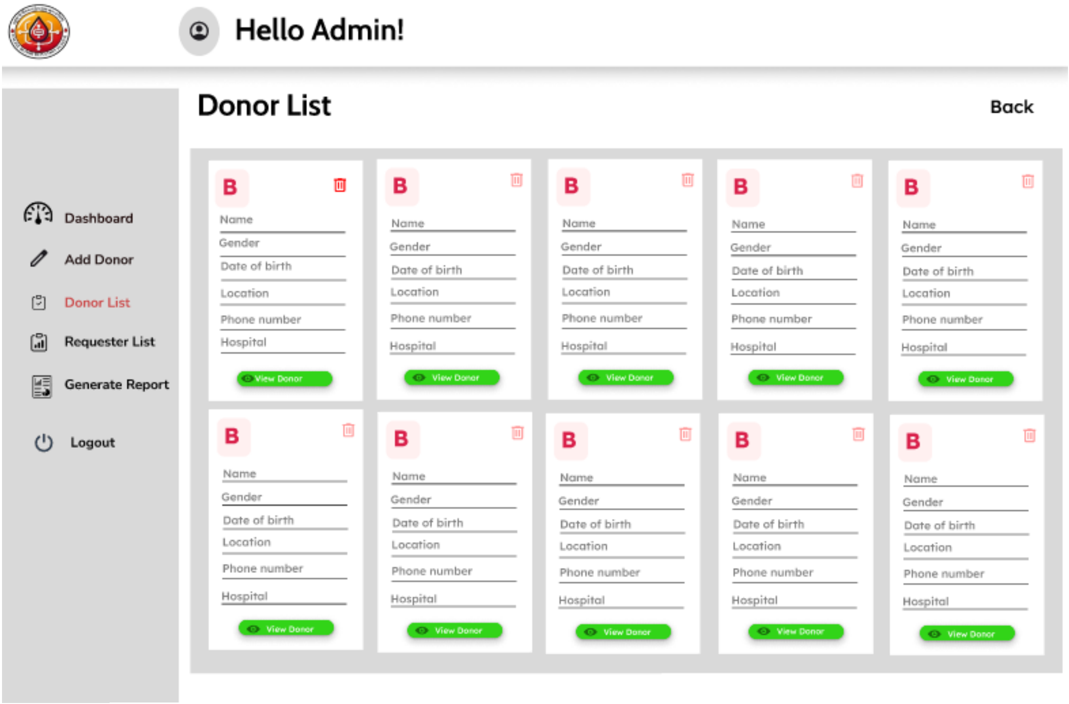


Figure 4.17 Donor List

The Donor List screen allows the admin to view individual donor details by clicking the "View Requester" button, while the Donor List screen lets the admin view requester details via the "View Requester" button. Additionally, it includes functionalities to "Add Requester" and navigate back to the previous page.

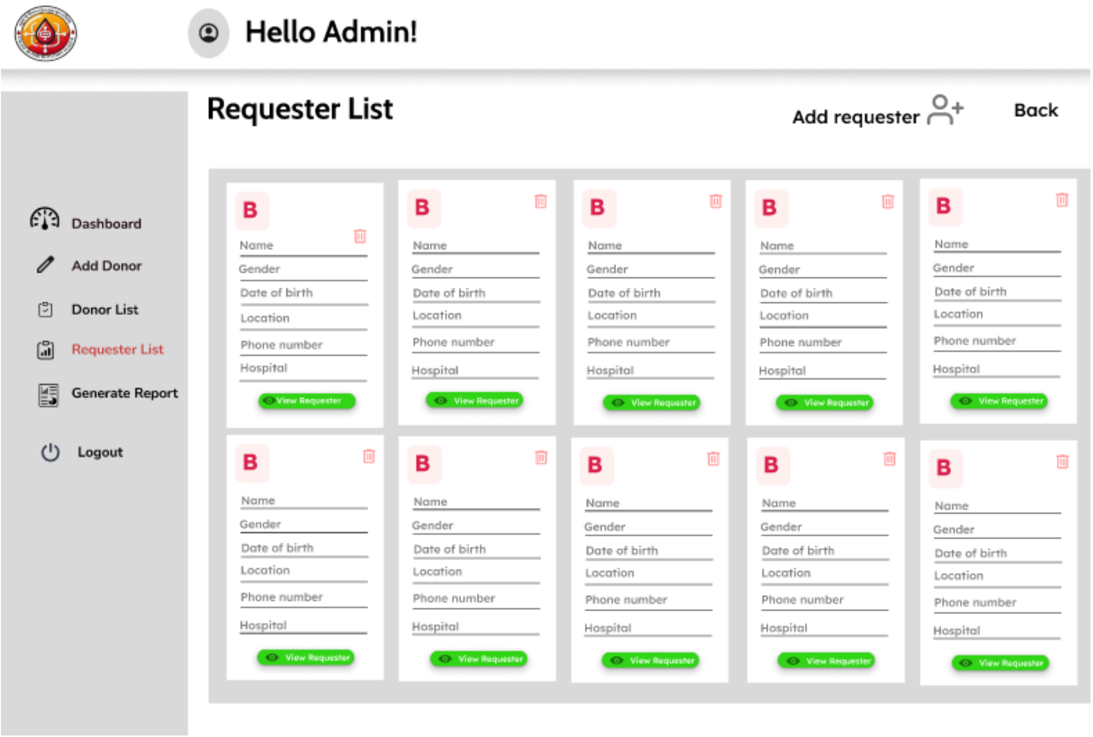


Figure 4.20 Requester List

The Requester List screen allows the admin to view individual donor details by clicking the "View Donor" button, while the Donor List screen lets the admin view requester details via the "View Requester" button. Additionally, it includes functionalities to "Add Requester" and navigate back to the previous page.

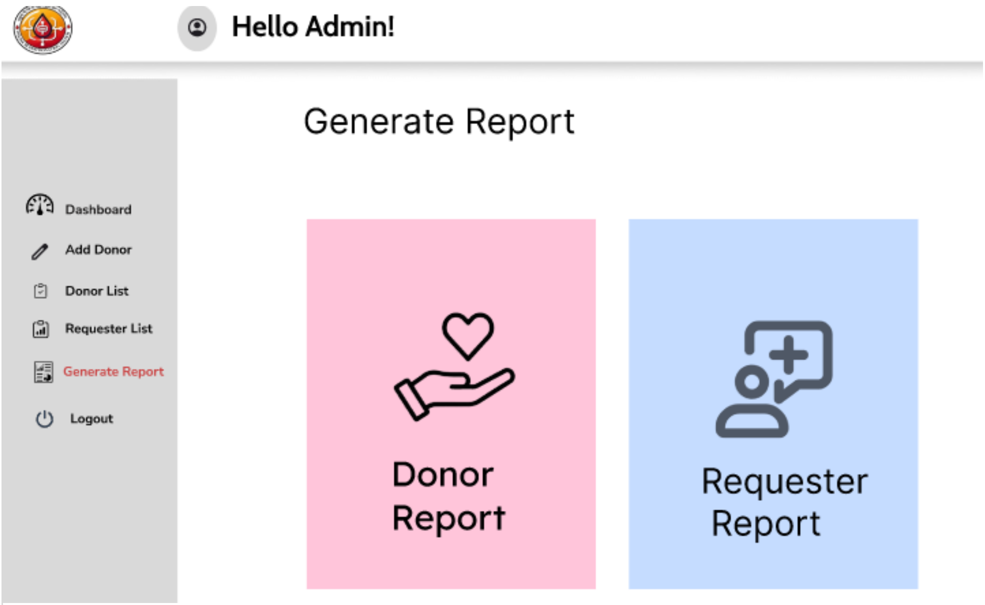


Figure 4.19 Report Generation Interface

This report generation interface allows administrators to produce system reports with a streamlined workflow, report selection with two clear options: Donor Report and Requester Report.

5. Chapter 5: Implementation and Maintenance

5.1 Implementation

System implementation is the phase where the design and planning of the OBDS are transformed into a working solution. It includes coding, testing, installing, and documenting the system to ensure that it performs as expected in a real-world environment. The goal is to deliver a functional, user-friendly platform that efficiently manages blood donations and enhances coordination between donors, hospitals, and administrators.

5.1.1 Coding, verification, and validation

Coding involves writing the actual source code based on the system design. In this project, the team will build the OBDS, which will allow donors, hospitals, and blood requesters to interact seamlessly. Developers will create modules for donor registration, blood request submission, appointment scheduling, and admin control. For example, donors will be able to register and update their availability, while hospitals will submit blood requests and track inventory. Functional components such as buttons for "Register as Donor", "Request Blood", and "Check Availability" will be programmed. The backend logic will be implemented using technologies like Python, while the frontend will use HTML/CSS, JavaScript, or a framework such as React. MySQL database will be integrated to store user records, donation logs, and blood stock data.

Verification will be conducted by developers and testers to ensure every part of the system functions correctly. Unit testing will be used to confirm that individual features such as "Send Alert to Donor" work as expected. Integration testing will ensure that connected modules—such as donor profiles working with the blood request system—function together smoothly. The team will also carry out code reviews where members will evaluate each other's code for quality and consistency.

Validation will be carried out with real users such as hospital staff, blood bank managers, and volunteer donors to ensure the system solves the actual problems identified. Test scenarios will include: "Can a requester easily submit a blood request?", "Can a donor update their availability date?", and "Does the admin receive real-time notifications?". User Acceptance Testing (UAT) will be done to collect feedback on usability and functionality. Any suggestions or requests for improvement will be noted and incorporated to make the system more effective and user-friendly.

5.1.2 Installation and documentation

Installation will involve deploying the OBDS on a web server, configuring the domain and hosting services, and setting up a secure, stable production environment. The system database will be initialized, application files will be uploaded, and dependencies will be installed. The initial setup will include the creation of test users such as sample donors, requesters, and admin profiles.

The **documentation** will serve as a guide to ensure smooth operation, ease of use, and future maintenance of the system. A **User Manual** will be developed to help donors, hospital staff, and administrators understand how to register, log in, request or donate blood, manage their profiles, and receive notifications. This manual will include step-by-step instructions with screenshots and examples to make it user-friendly and accessible to both technical and non-technical users.

In addition to the user guide, **Technical Documentation** will be written for developers and IT support personnel. This will include detailed information about the system architecture, database design (including table structures and relationships), API specifications, data flow diagrams, and logic behind key modules such as donor verification, blood request management, and admin functions. This documentation will make it easier for future developers to understand and extend the system if needed.

A **Maintenance Guide** will also be prepared to assist with the upkeep of the system after deployment. It will provide instructions for performing regular system backups, applying security updates, troubleshooting common issues, and updating the database or source code. This guide will help ensure that the OBDS remains reliable, secure, and responsive to evolving requirements. All documentation will be reviewed and updated regularly based on user feedback and system changes, ensuring it stays relevant and effective over time.

5.1.3 Testing

Testing plays a crucial role in ensuring that software functions correctly and fulfills its intended purpose. For the **Online Blood Donation System**, it is essential to verify that users—such as donors, recipients, and healthcare staff—can interact with the platform effectively and without encountering technical glitches. One widely used approach is **black-box testing**, where the focus is on examining the system's behaviour from a user’s perspective, without access to the internal code. For instance, testers might enter various login credentials to confirm that valid users can sign in successfully, while incorrect information triggers appropriate error messages. Other core functions, like scheduling a donation or searching for available blood types, can also be tested this way to ensure they operate smoothly. This approach helps validate the system’s interface and overall functionality.

Another technique is **white-box testing**, which involves inspecting the internal workings of the system. Developers assess the logic within the code—such as conditions, loops, and functions—to verify accuracy and security. For example, they might review the part of the code responsible for determining donor eligibility based on health criteria. This method helps identify hidden issues that could disrupt system processes. Combining both black-box and white-box testing ensures that the blood donation platform is both dependable and secure, enhancing user confidence and the quality of service.

In developing the **Online Blood Donation System**, various testing methods are implemented to evaluate the system’s performance across different areas:

- **Unit Testing** targets individual elements of the system to confirm they perform their specific tasks correctly. For example, we can verify if the “Request” function adds the request correctly to the donors dashboard accurately.

- **Integration Testing** ensures that different system modules communicate and operate together as expected. For example, once a donor registers and chooses a date, the system should successfully update both the user interface and backend database.
- **Functional Testing** is conducted to verify whether the platform's features meet the desired functional specifications. Key tasks—like donor sign-up, blood search, or request processing—are tested to confirm they deliver the expected results.
- **System Testing** evaluates the platform in its entirety, simulating real-life scenarios. A tester could perform a full walkthrough—from account creation to making a donation request and contacting support—to ensure that all processes flow as intended.
- **Stress Testing** is used to determine how the system handles peak usage. For instance, if there's an emergency requiring many people to log in and make blood requests at the same time, the platform should be able to manage the load without failure.

Among these methods, **Functional Testing** and **System Testing** are especially effective and practical for the blood donation platform, as they replicate real user journeys and help identify major usability and operational issues.

For example, during Functional Testing, a tester might check whether a donor can successfully register, input their blood type and location, and receive confirmation after submitting the form. Meanwhile, in System Testing, the entire workflow could be tested—such as a user creating an account, searching for nearby donation centers, booking an appointment, and receiving an email reminder—to ensure all components work together seamlessly and without errors.

- **Performance Testing** evaluates how the platform performs under various levels of demand. For instance, checking how quickly the system responds when multiple users are posting requests or browsing for blood types helps ensure smooth functionality during high-traffic times.
- **Usability Testing** centres around how easily users can navigate and interact with the platform. This includes assessing whether donors can intuitively register, find donation locations, or edit their profiles without confusion.
- **Acceptance Testing** is used to verify that the finished system satisfies the original goals and requirements. Before going live, all major functions—such as account creation, blood inventory search, and appointment scheduling—are tested to ensure they are complete and operational.
- **Regression Testing** ensures that updates to the platform do not interfere with features that previously worked correctly. For example, after adding a new donation history section, we re-test login and request features to confirm they are still functional.
- **Beta Testing** involves releasing the system to a limited number of real users, such as hospital staff or regular blood donors, to gather hands-on feedback. This testing phase helps uncover real-world issues that might not be evident in controlled testing environments.

During the early development stages of the **Online Blood Donation System**, **Usability Testing** and **Beta Testing** are especially valuable. They focus on feedback from actual users, which helps refine the interface and improve user satisfaction. Simple tools like **Google Forms** can be used to gather responses efficiently. This method is cost-effective, widely accessible, and offers easy-to-analyze results, making it ideal for enhancing the system based on user input.

5.2 Maintenance

After successful deployment, the system will require ongoing maintenance to ensure long-term performance, security, and reliability.

- **Corrective Maintenance** will be performed to fix any bugs or system errors identified by users after deployment (e.g., login failures, missing data).
- **Adaptive Maintenance** will be carried out to make the system compatible with new hardware, software environments, or regulatory changes.
- **Perfective Maintenance** will involve enhancing system features based on user feedback. For example, if users suggest a blood type filter or notification customization, those updates will be implemented.
- **Preventive Maintenance** will include routine system audits, regular database backups, and optimization to prevent future failures or data loss.

Through continuous monitoring and updates, the system will remain robust, secure, and responsive to the evolving needs of its users.

Cost of Maintenance:

Maintaining the Blood Donor System will need some money and time every year. We may need to pay for fixing errors, updating the system, and making sure the system stays safe and fast. For example, fixing bugs might cost around Nu. 2,000–3,000 a year, while adding new features or updating software could cost Nu. 3,000–5,000. We also need to pay for hosting the website and renewing the domain name, which might cost around Nu. 1,500–2,000. So, the total maintenance cost for one year is about Nu. 8,000 to Nu. 12,500. This helps the system stay active and useful, especially during emergencies when someone urgently needs blood.

Managing Maintenance:

Managing maintenance means planning how to take care of the system regularly. In our Online Blood Donor System, we can manage it by keeping the donor information updated, fixing problems on time, and listening to user feedback. For example, if a user says the "Request Blood" button is not working, we should fix it immediately. We can also add features like a rating system where people rate donors or hospitals based on their experience. This helps other users trust the system. Regular updates and checking for bugs or errors will keep the system safe, fast, and helpful for everyone who needs or donates blood.

In conclusion, maintenance is an important part of our Online Blood Donor System. It helps the system work without problems, keeps users happy, and builds trust. By fixing errors, updating features, and planning for future issues, we make sure the system is always ready to help people in need of blood. With proper care and regular maintenance, our system can save lives and serve the community for a long time.

6. Chapter 6: Conclusion

The development of the Online Blood Donation System has been a valuable and eye-opening experience. Through each phase—planning, analysis, and design—we gained a better understanding of the real-life problems faced in blood donation, such as the lack of real-time information, difficulty in finding donors quickly, and the manual handling of blood requests.

During the planning phase, we clearly defined the objectives and scope of the project, ensuring that the system would address the key issues faced by both donors and hospitals. We identified the primary stakeholders and set realistic goals to make the platform user-friendly, efficient, and accessible to the public. Proper time and resource planning allowed us to approach each stage with clarity and confidence.

In the analysis phase, we gathered valuable information through interviews and research, particularly from a blood bank manager, to understand the real-life challenges in blood donation and management. We analyzed user requirements, categorized them into functional and non-functional aspects, and identified key features such as donor registration, blood request handling, and availability tracking. This phase was crucial in shaping a system that aligns closely with user needs and expectations.

Moving into the design phase, we translated our understanding into a structured system blueprint. We created data flow diagrams (DFDs), entity-relationship diagrams (ERDs), and designed the user interface for a smooth experience. The focus was on ensuring that the system is intuitive, secure, and scalable. Proper system architecture and database design were emphasized to support future development and real-time functionality.

In conclusion, the completion of the planning, analysis, and design phases has laid a strong foundation for the Online Blood Donation System. Each phase contributed significantly to transforming the initial idea into a well-defined solution. With a clear roadmap ahead, we are now ready to proceed confidently to the implementation phase.

7. Reference

Abdul-Gafaru, T., Laar, D. S., & Kubuga , K. K. (2024). *Design and Implementation of an Online Blood Donation System*. Asian Journal of Research in Computer Science, 17(4), 44–61. https://doi.org/10.9734/ajrcos/2024/v17i4429

Blood Bank Management System. (2023). *Blood bank management system*. SciSpace.

https://www.researchgate.net/publication/370602395_Blood_Bank_Management_System

Gupta, A. (n.d.). *Online blood donation management system report.* Scribd.

https://www.scribd.com/doc/304887882/Online-Blood-Donation-Management-System-Report

Paily, C. (n.d.). *Blood bank management system*. Scribd. https://www.scribd.com/document/394157548/Blood-Bank-Management-System

8. Appendix

Appendix I: Admin Manual

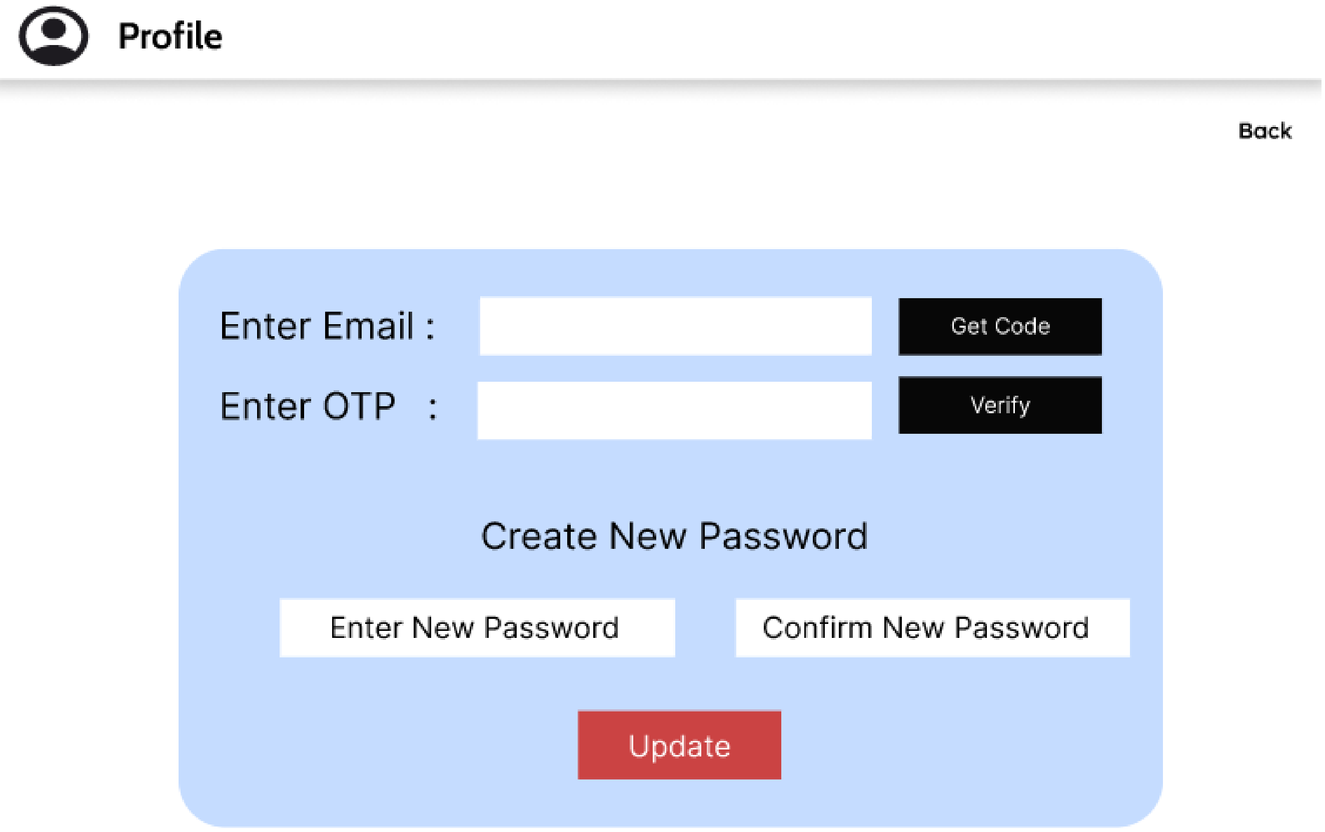
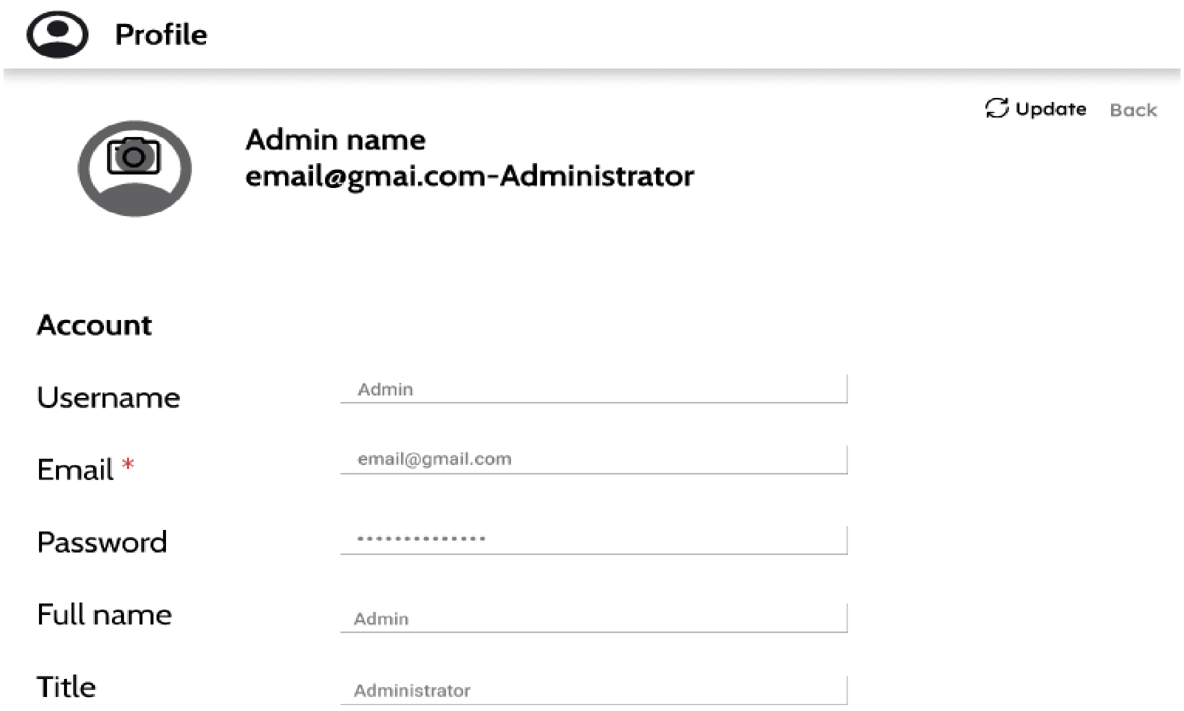


Figure 8.1 Password Reset Interface

This password reset interface features a three-step secure process: email entry for OTP delivery, code verification, and new password creation with confirmation. The clean, linear flow combines security (multi-factor authentication) with user-friendly design through clear labels and minimal fields



Admin name
email@gmai.com-Administrator

Account

Username: Admin

Email *: email@gmail.com

Password: ••••••••••••••

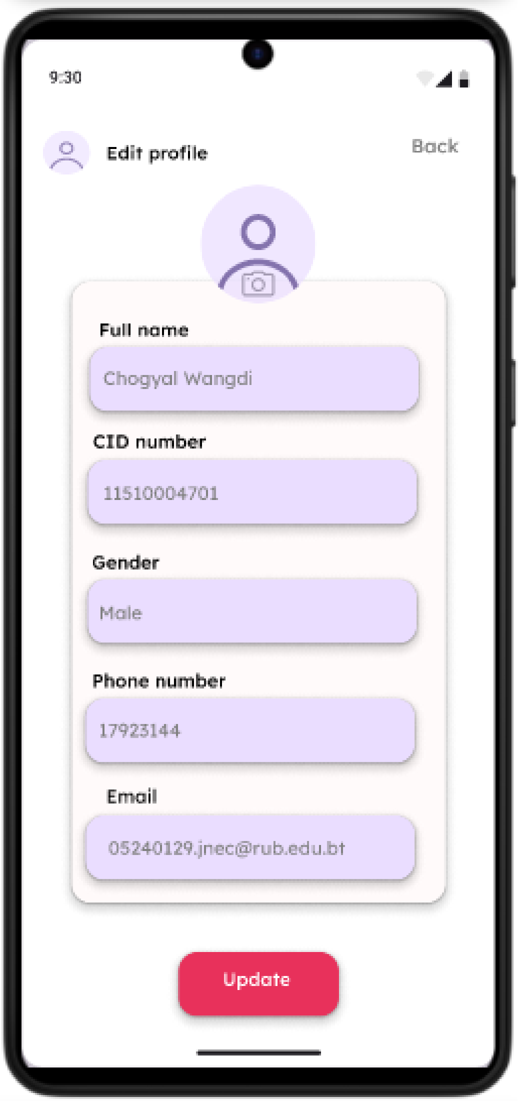
Full name: Admin

Title: Administrator

Figure 8.2 Admin Profile Interface

This admin profile interface displays account details (username: "Admin", email: "email@gmail.com", role: "Administrator") in a structured layout. It features three clear sections: account info (with required email field), password management (hidden for security), and personal details (name/title). The first design uses dividers for visual clarity, while the second opts for space-saving nested bullets - both maintaining secure, professional credential management.

Appendix II: User Manual



Full name

Chogyal Wangdi

CID number

11510004701

Gender

Male

Phone number

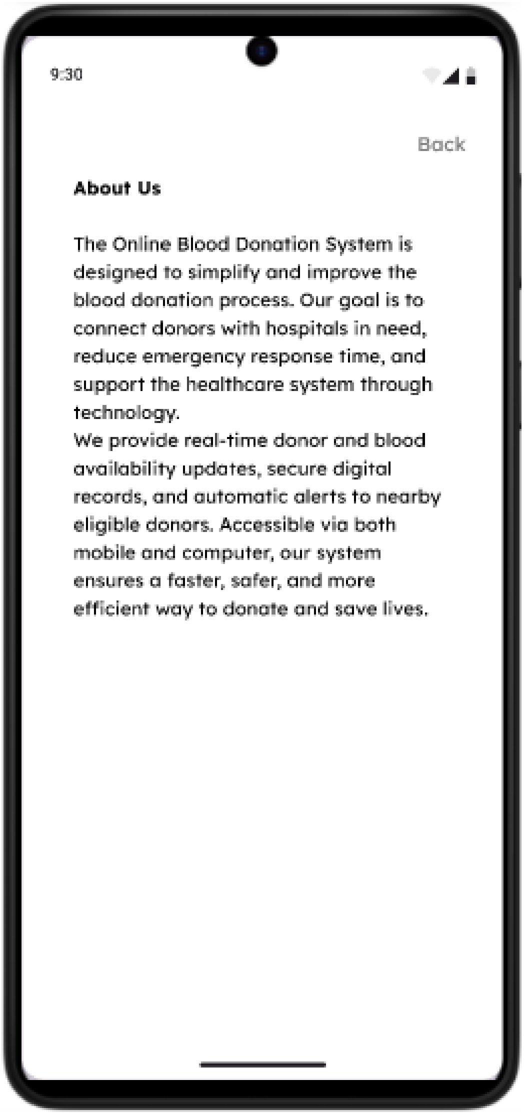
17923144

Email

05240129.jnec@rub.edu.bt

Figure 8.4 Profile Editor

This profile editor lets users update their details (name, CID, gender, phone, email) with an "Update" button to save changes, enabling easy modifications.

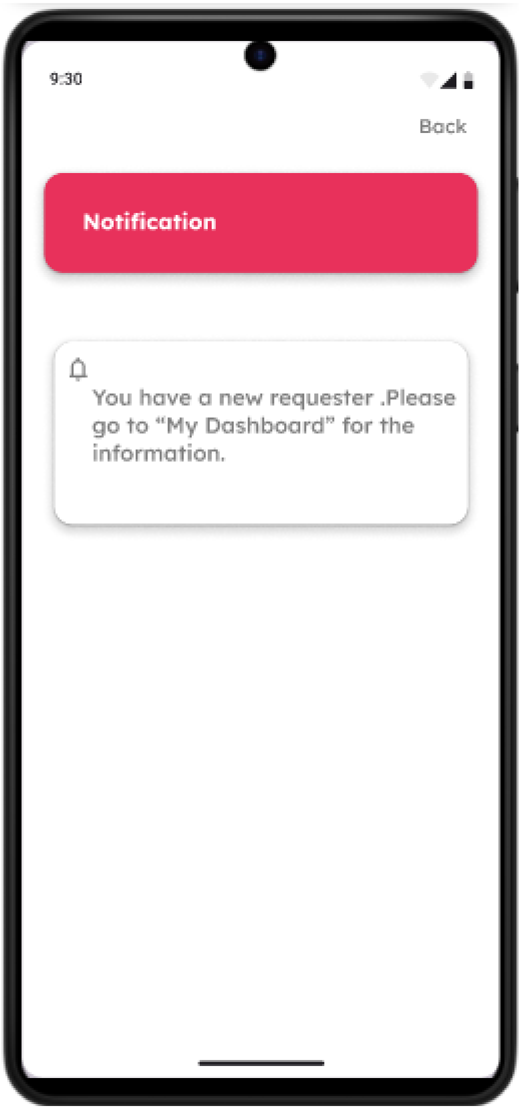


About Us

The Online Blood Donation System is designed to simplify and improve the blood donation process. Our goal is to connect donors with hospitals in need, reduce emergency response time, and support the healthcare system through technology.
We provide real-time donor and blood availability updates, secure digital records, and automatic alerts to nearby eligible donors. Accessible via both mobile and computer, our system ensures a faster, safer, and more efficient way to donate and save lives.

Figure 8.3 About Us Screen

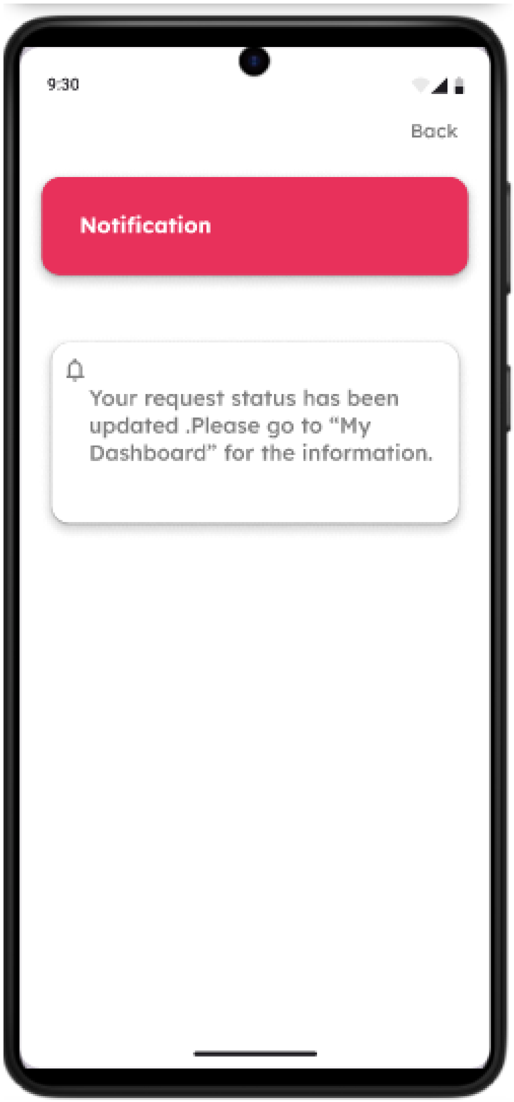
This "About Us" screen explains the system's mission to digitally connect blood donors with hospitals, featuring real-time tracking and emergency alerts. The text-focused design clearly communicates its lifesaving purpose, with a "Back" button for navigation.



You have a new requester .Please go to “My Dashboard” for the information.

Figure 8.6 Donor Notification Alert

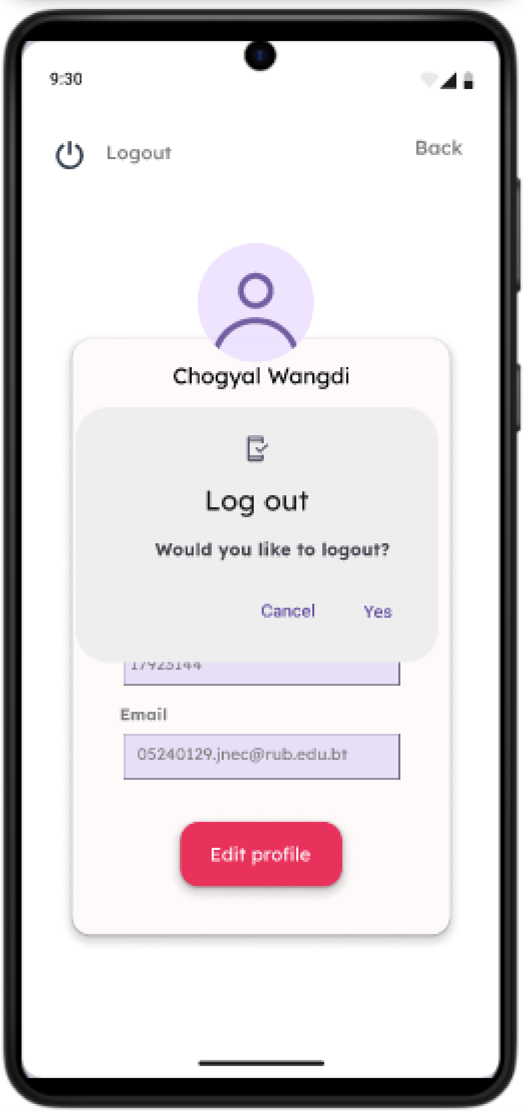
This notification alert informs donors about a new blood request, prompting them to check "My Dashboard" for details.



Your request status has been updated .Please go to “My Dashboard” for the information.

Figure 8.5 Requester Notification Alert

This notification alert informs requesters about updated blood request status, prompting them to check "My Dashboard" for details.



Chogyal Wangdi

Log out

Would you like to logout?

17923144

Email

05240129.jnec@rub.edu.bt

Figure 8.7 Logout Confirmation Screen

This logout confirmation screen verifies identity (showing user Chogyal Wangdi and academic email) before signing out, with Cancel/Yes options. It maintains app consistency with Back button,

Appendix III: Requirement Determination

Our team has selected interviews as the key method for gathering requirements for the **Online Blood Donation System**. Since this system involves multiple stakeholders—such as donors, hospital staff, blood bank administrators, and recipients—interviews allow us to collect detailed, personalized insights while maintaining a human-centric approach. Given the sensitive nature of blood donation (privacy concerns, trust, and urgency), face-to-face or structured virtual interviews help build rapport and encourage honest feedback. Additionally, interviews provide flexibility to explore unexpected concerns and clarify complex workflows in real time.

To ensure comprehensive requirement gathering, we will ask targeted questions, such as:

Q1. Currently, how do donors register to give blood? What challenges do they face in this process?

Q2. When a patient needs blood urgently, how do you find donors? What slows this down?

Q3. How do you keep track of blood stock levels? What problems arise?

Q4. What difficulties arise when searching for rare blood types manually?

Q5. How are donor records and blood inventory currently tracked? What problems does this create?

Q6. How does manual record-keeping impact blood wastage and shortages?

Q7. Based on these challenges, how would an online system improve the blood donation process?

Q8. How does the lack of an online system affect donor recruitment and retention?

By conducting interviews, we can identify critical needs, validate assumptions, and prioritize features that align with real-world usage. This method ensures that the system is user-friendly, secure, and effective in addressing gaps in the current blood donation ecosystem.

Appendix IV: Statement of Work

Online Blood Donation System

Project Charter

Project Name: Online Blood Donation System
Project Manager: Chogyal Wangdi
Users: Donors, Requester and Hospital

Project Start/End (Projected)
Beginning phase of project: Second Semester (first year)
Final phase of project: First Semester (second year)

Project Overview:
The project aims to develop a comprehensive Online Blood Donation System to provide an effective and effective platform to support blood donation and save lives.

Objectives:

- Support blood donation and save lives.
- Solve blood bank and hospital issues with real-time donor info.
- Reduce donor search time in emergencies.
- Replace paper records with secure digital forms.
- Ensure mobile and computer access for all users.
- Protect donor privacy and send donation reminders.
- Help hospitals manage blood supplies.
- Speed up emergency response

Key Assumptions:

- Users (donors and hospitals) have stable internet access for system usage.
- Donors register with accurate and up-to-date personal and health information.
- Hospitals and blood banks regularly update blood inventory in the system.
- The system will be accessible on both mobile and desktop platforms.

Stakeholders and Responsibilities

Stakeholder	Role	Responsibility	Signature
Younten Tshering	Module tutor	Project progress coordinator	
Karma Jigme Wangchuk	IT support specialist	Support for team members	
Chogyal Wangdi	Scrum master, Project Manager	Plan, Lead, Delivery	
Norbu Wangmo	Data Analyst	Collect, Analyze, Interpret	
Sonam Jamtsho	Back-end developer	Built, Integrate, Maintain	
Dorji Samdrup	Designer, Front-end developer	Design UI	