



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

BloodLink (Blood Donation Management)

A Software Engineering Project Submitted
By

Semester: Summer 24-25		Section: P	Group Number:	
SL	Student Name	Student ID	Contribution (CO3+CO4)	Individual Marks
1	DIN MUHAMMAD REZWOAN	23-51712-2		
2	SOUMIK DAS DIPON	23-51709-2		
3	MAOUN BILLAH	22-48813-3		
4	SANJANA SAYED NUHA	23-53087-3		

The project will be evaluated for the following Course Outcomes

CO3 (PO-g-1) <i>Select appropriate software engineering models, project management roles and their associated skills for the complex software engineering project and evaluate the sustainability of developed software, taking into consideration the societal and environmental aspects</i>	Total Marks	
Selection of Software Engineering Models: Process model selection and presents sufficient evidence to support argument for the model selection	[5 Marks]	
Role identification and Responsibility Allocation: Well-planned project with proper role identification and responsibility allocation in the project management activities	[5Marks]	
Formatting and Submission: Submission, Defense, Completeness, Spelling, grammar, and Organization of the Project report	[5Marks]	
CO4 (PO-k-1) <i>Apply engineering management principles and economic decision making to develop software engineering project management plan.</i>	Total Marks	
Project WBS and Testcases: Relevant WBS (project task list) and testcases for the proposed project are stated properly.	[5Marks]	
Effort Estimation and Scheduling: Project estimation was described using proper effort estimation or schedules based on available project resources	[5Marks]	
Risk Management: Sufficient and appropriate risks are identified, analyzed, and properly categorized or prioritized.	[5Marks]	

1. PROJECT PROPOSAL

1.1 Background to the Problem

The current blood donation process is critically inefficient, creating a disconnected and unreliable system for a life-saving resource. Potential donors are frequently discouraged by the lack of accessible information on donation locations and eligibility, while hospitals in urgent need of blood must resort to frantic, manual phone calls that cause dangerous treatment delays. This fragmentation also plagues blood banks, which struggle with managing their inventory, leading to the simultaneous problems of critical shortages and the wastage of expired blood units. At its core, the problem is a complete lack of a centralized, real-time communication platform that connects donors, hospitals, and blood banks. This systemic failure creates a high-risk environment where patient care is compromised, directly establishing the need for an integrated technological solution.

1.2 Solution to the Problem and Process Model Selection

To address the inefficiencies and communication gaps in the current blood donation lifecycle, this project proposes the development of the **BloodLink Management System**. This integrated web platform is designed to serve as a central hub connecting all key stakeholders—donors, hospitals, and blood banks—in real time. The primary objective is to boost donor engagement, minimize blood wastage through efficient inventory management, enhance emergency response capabilities, and ultimately save more lives by creating a smarter, more interconnected blood donation ecosystem.

Project Scope and Features

The system's scope encompasses the entire donation process, from initial donor registration to the final delivery of blood units and subsequent data analysis. The platform's creative solution to the problem is delivered through a suite of core features:

- **Smart Matching and Logistics:** The system will utilize algorithms to intelligently match donors with needs based on geographic location, blood type, and the urgency of the request.
- **Predictive Inventory Management:** It will employ forecasting techniques to anticipate potential blood shortages, allowing administrators to optimize stock levels and proactively plan donation drives.
- **Real-Time Tracking and Notifications:** The platform will provide live inventory tracking for hospitals and blood banks and send integrated, automated notifications to keep all stakeholders informed of urgent needs, appointment reminders, and delivery statuses.
- **User-Centric Portals:** Dedicated dashboards will be provided for each user role, offering tools for easy registration and simplified appointment scheduling for donors.

- **Data Analytics and Reporting:** The system will include a robust analytics module to help administrators monitor donation trends, measure campaign effectiveness, and efficiently allocate resources.

User Stories and Task Management

The detailed functional requirements of the system have been broken down into user stories to ensure a user-centric development approach. The project's tasks, progress, and workflow are managed using a Trello board.

- **Project Requirement Document (PRD) with User Stories:** The complete list of user stories is maintained in the project's PRD, available at the following link:
https://olive-koi-fe3.notion.site/Project-Requirement-Documents-PRD-248c4085a9578042a52cf9fdea218ae4?source=copy_link
- **Trello User Story Board:** The project's task board can be viewed here:
<https://trello.com/invite/b/688adde9ff1ea6a061a243fb/ATTI53b0454f1985e12ebc3616a5ed86d97738CCF963/bloodlink>

Process Model Selection and Justification

After a thorough analysis of the project's goals and environment, the **Extreme Programming (XP) model** was selected as the most suitable software development process model.

Analysis of the Project Environment The BloodLink system operates within a dynamic healthcare environment. While core requirements such as donor registration and inventory logging are stable, other features related to analytics, notification systems, and user dashboards are likely to evolve based on stakeholder feedback and potential changes in healthcare policies. This moderately dynamic nature makes an adaptive and iterative model essential for success.

Justification for Selecting the XP Model

The Extreme Programming model is ideally suited for this project for several compelling reasons:

- **Adaptability to Changing Requirements:** XP is a highly flexible, iterative model. Its structure of short development cycles and frequent releases allows the team to continuously incorporate feedback from donors, hospital staff, and other stakeholders. This ensures the final product is precisely aligned with user needs, even if they change during development.
- **Team Support and Communication:** The model is well-suited for our small-to-medium team size as it promotes close collaboration, daily communication, and coordinated task management through practices like pair programming and a collective codebase ownership. This ensures the solution is feasible and that development stays aligned with the business objectives.

- **Risk Management:** XP effectively manages project risks and uncertainties at every stage. The emphasis on short, iterative cycles means that technical and schedule risks are identified early. Continuous testing and integration mitigate quality risks, while frequent stakeholder collaboration ensures that the project does not deviate from user expectations.
- **Support for Timely Delivery:** The model directly supports on-time delivery and adherence to deadlines. By breaking the project into small, functional increments delivered through sprints, the team can make steady, measurable progress. This iterative approach allows for the prioritization of high-value features, ensuring that a viable product is available quickly.

Superiority Over Alternative Models The XP model was chosen over other alternatives due to its balanced and practical approach. Unlike the rigid, sequential **Waterfall model**, XP can easily accommodate the expected changes in a healthcare project. Compared to the **Spiral model**, XP is simpler and more practical to implement for a team of our size. While the **Prototype model** focuses primarily on early UI/UX validation, XP goes further by delivering fully functional and tested software increments throughout the entire development lifecycle, making it the most reliable and efficient choice for this project.

Target Users and Project Contributions

Target Users and Benefits The platform is designed to serve a diverse group of users, each receiving distinct benefits:

- **Blood Donors:** Benefit from a simplified registration and appointment scheduling process, as well as timely notifications and insights into their impact.
- **Hospitals and Blood Banks:** Gain access to real-time inventory tracking, faster fulfillment of urgent requests, and predictive insights for better stock management.
- **Patients and Emergency Services:** Ultimately benefit from quicker access to required blood, which can significantly improve outcomes in critical situations.

Contribution to Scientific Development This project contributes to scientific and public health research by providing a well-documented, data-driven platform for analyzing blood donation patterns, donor behavior, and regional demand trends. By systematically capturing and organizing this information, the system creates a reliable and valuable dataset. This data can be used for future academic studies, policy-making, and optimizing healthcare logistics, demonstrating a practical application of technology to improve life-saving operations.

1.3 Project Role Identification and Responsibilities

For the BloodLink project, the team adopted a structure where each member assumed leadership over specific domains. This approach enabled the assignment of tasks based on individual expertise, covering all project phases from initial ideation and system architecture to detailed documentation and planning.

Nuha – Project Visionary & System Architect

Nuha was responsible for the project's initial concept and its technical foundation. She initiated the project and translated its vision into a concrete architectural design.

- **Key Responsibilities:** Formulating the initial project idea, drafting the project proposal, and designing the entire system architecture.
- **Role Across Project Stages:**
 - **Requirements Gathering:** Initiated the project by conceptualizing the idea and creating the first draft of the **Project Proposal**.
 - **Design:** This was Nuha's primary area of contribution. She was solely responsible for creating all essential system diagrams: the **Use Case Diagram**, **Class Diagram**, **Activity Diagram**, and **Data Flow Diagram (DFD)**.

Maoun – Project Manager

As the Project Manager, Maoun was responsible for finalizing the project's scope and ensuring the team's work was coordinated and aligned with the stated objectives.

- **Key Responsibilities:** Finalizing core project documentation, managing team coordination, and overseeing the project's overall progress.
- **Role Across Project Stages:**
 - **Requirements Gathering:** Assumed responsibility for editing and **finalizing the Project Proposal** from the initial draft, ensuring it was comprehensive and polished.
 - **Management:** Maintained oversight throughout all project phases to ensure all deliverables were integrated smoothly.
 - **Final Decision-Making:** Held the primary responsibility for final decisions regarding the project's scope and deliverables.

Rezwoan – UI/UX Designer, Process Lead & Lead Tester

Rezwoan was responsible for the user-centric and process-oriented aspects of the project. This included writing detailed specifications, designing the user interface, and defining the development and quality assurance workflows.

- **Key Responsibilities:** Authoring detailed requirement documents, designing the user interface, documenting the development workflow, and defining the project's testing strategy.
- **Role Across Project Stages:**
 - **Requirements Gathering:** Wrote the comprehensive **Software Requirements Specification (SRS)** and **Product Requirements Document (PRD)**.

- **Design:** Created all the visual **Figma wireframes** and UI mockups for the application.
- **Implementation:** To ensure a structured development process, he documented the **Git Workflow**, outlining version control practices for the team.
- **Testing:** As the lead tester, he authored the **Testing** documentation, which outlined the plan for verifying the system's functionality.

Soumik – Project Planner & Metrics Analyst

Soumik was responsible for the analytical and quantitative aspects of the project, focusing on planning, estimation, and the measurement of the product's design quality.

- **Key Responsibilities:** Estimating project effort and cost, creating the project schedule, and applying software metrics to evaluate the system's design.
- **Role Across Project Stages:**
 - **Planning:** Led the **Project Estimation and Scheduling** (Part 3), defining timelines and forecasting the required effort.
 - **Quality Analysis:** Applied **Software Product Metrics** (Part 7) to the design. This role involved using formal metrics to analyze the complexity, maintainability, and quality of the system's structure, which is distinct from functional testing.

Distribution and Justification of Responsibilities

The distribution of these roles and responsibilities was based on the specific skills and expertise of each team member.

- **Decision Making:** **Maoun** held final decision-making authority as the Project Manager. Key technical decisions related to system structure were made by **Nuha**.
- **Quality Assurance:** Quality Assurance was a dual responsibility. **Rezwoan** led the process-oriented side by defining the **testing strategy**. **Soumik** handled the data-oriented side by using **software metrics** to analyze the quality of the design itself.
- **Resource Management:** This was managed by **Soumik**, who created the schedule and effort estimates, with oversight from **Maoun**.

2. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) / PRODUCT REQUIREMENTS DOCUMENT (PRD)

This section serves as the technical blueprint for the BloodLink system, detailing its core capabilities and quality standards. It is divided into two key areas: Functional Requirements, which specify *what* the system will do for its users, and Non-Functional Requirements, which define *how well* the system must perform. The comprehensive Project Requirement Document (PRD), which forms the basis for these specifications, is accessible via the link below.

Project Requirement Document (PRD) Link: https://olive-koi-fe3.notion.site/Project-Requirement-Documents-PRD-248c4085a9578042a52cf9fdea218ae4?source=copy_link

2.1 Functional Requirements

The system's functionalities are designed around the specific needs of its users, ensuring that each role is equipped with the necessary tools to perform their tasks effectively.

For the Donor To create an engaging and seamless experience for donors, the system will offer the following capabilities:

- **Authentication & Profile Management:** Donors can create a secure personal profile , log in to their account , reset a forgotten password , and easily update their personal information or profile photo.
- **Appointment Scheduling:** A user-friendly interface allows donors to find nearby donation centers, book a donation appointment , and cancel if their plans change.
- **Information and Impact:** Through a personal dashboard, donors can view their eligibility status for future donations and see the positive impact their contributions have made on the community.
- **Urgent Notifications:** Donors can opt-in to receive critical notifications for urgent blood needs in their area, enabling them to respond quickly in emergencies.

For Blood Bank Staff To empower Blood Bank Staff with efficient control over daily operations, the system provides a suite of management tools:

- **Centralized Dashboard:** Staff can access a comprehensive dashboard for a real-time overview of operations.
- **Inventory and Donor Management:** The system allows staff to manage donation schedules , update the blood inventory in real-time , view donor records , and register new donors on-site during blood drives.
- **Reporting:** Staff can generate detailed inventory reports to aid in analysis and stock management.

For Hospital Staff To ensure hospital personnel can meet urgent patient needs swiftly, the platform offers direct access to critical information and services:

- **Real-Time Blood Availability:** Staff can view the live inventory of connected blood banks to quickly find required blood units.
- **Urgent Request System:** An integrated system enables staff to place urgent digital requests for blood and monitor their status through to completion.
- **Delivery Tracking and History:** The platform provides real-time tracking of a delivery's status and maintains a viewable history of all past requests for record-keeping.

For Delivery Personnel To facilitate transparent and secure transportation, logistics personnel are equipped with the following tools:

- **Task Management:** A clear interface allows personnel to view all assigned deliveries and their details.

- **Live Status Updates:** Drivers can update the delivery status in real-time, keeping the hospital and blood bank informed.
- **Secure Confirmation:** Deliveries are confirmed using a secure OTP, ensuring the blood reaches its intended destination.

For Campaign Organizers To support community engagement, organizers have the functionality to plan and execute successful blood drives:

- **Event and Volunteer Management:** The system provides tools to plan blood drives , manage volunteers , and assign them specific roles for events.
- **Performance Analytics:** Organizers can track event performance to analyze the success of their campaigns and improve future outreach efforts.

For the Administrator To ensure smooth operation and platform integrity, the administrator has top-level oversight and control:

- **User and System Management:** The admin can manage all user accounts , define roles and permissions , monitor system-wide analytics , and view system logs.
- **Oversight and Reporting:** Admins are responsible for approving campaign requests and can generate platform-wide reports on usage and performance.

2.2 Non-Functional Requirements

Beyond specific features, the system's success depends on its quality and reliability. The following attributes define the operational standards BloodLink must meet.

- **Performance:** In a time-sensitive environment, system responsiveness is paramount. BloodLink is designed to be fast and efficient, ensuring that critical functions like urgent requests and inventory searches are completed in under 500 milliseconds. The architecture will be robust enough to handle peak loads without a drop in performance.
- **Reliability:** As a critical healthcare support tool, the BloodLink platform must be consistently available. The system is engineered for 24/7 operation with a 99.9% uptime target, featuring robust backup and recovery protocols to prevent data loss and ensure uninterrupted service.
- **Security:** Protecting the sensitive health and personal data of users is a fundamental requirement. All data will be secured with encryption, both in transit and at rest. Strict role-based access control and strong user authentication will be enforced to guarantee that users can only access information appropriate for their role.
- **Usability:** A positive user experience is essential for adoption and effectiveness. The interface will be designed to be intuitive, clean, and consistent across all devices, reducing the learning curve for donors and minimizing the risk of errors for staff performing critical tasks.
- **Maintainability:** To ensure the long-term viability of the platform, it will be built with clean, well-documented code and a modular architecture. This approach will simplify future updates, bug fixes, and the integration of new features.

- **Scalability:** The system must be prepared for growth. Its architecture will be designed to scale efficiently, allowing it to support an increasing number of users, hospitals, and blood banks without compromising its performance or reliability.

3. PROJECT ESTIMATION AND SCHEDULING

3.1 Effort and Cost Estimation

Scope Identification

BloodLink is a web application for Blood Donation Management System. This application's goal is to create a user friendly system that will simplify the whole process of blood donation. The system helps Blood donors, blood bank staff, hospital staff, delivery personnel and administrators with features like registration, blood inventory, delivery tracking, delivery assignment and management.

Lines of Code Estimation

We are assuming that our project will need 85000 lines of code. Then if a developer writes 50 lines of code a day.

$$85000 \div 50 = 1700 \text{ days}$$

It is about 57 person-months.

COCOMO Model

SLOC = 85000 (approximate)

let's assume this is organic type software project

Coefficient<Effort Factor> = 2.4

P = 1.05

T = 0.38

Effort = PM = Coefficient <Effort Factor> * (SLOC/1000)^P

$$= 2.4 * (85000/1000)^{1.05}$$

$$= 254.74$$

Development time = DM = 2.50 * (PM)^T

$$= 2.50 * (254.74)^{0.38}$$

$$= 20.52$$

Required people = ST = PM / DM

$$= 254.74 / 20.52$$

$$= 12.41 \sim 13$$

- Effort: about 254.74 person-months
- Development Time: about 20.52 months
- Team Size: about 13 people

Three Estimation Result

Lets use SLIM model

LOC = 85000

B = 2.5

P = 800 (typically)

Lets assume we will finish in 20 months

So, Effort , $E = \left(\frac{LOC \times B^{0.33}}{P}\right)^3 \times \frac{1}{T^4}$

$$E = \left(\frac{85000 \times 2.5^{0.33}}{800}\right)^3 \times \frac{1}{20.52^4}$$

E = 18.57 person-months

By first effort estimation we got 57 person-months

By Cocomo model we got 254.74 person-months

By Slim model we got 18.57 person-months

Assumptions

First we assumed that our project is 85000 lines of code. Then in first estimation we thought that a developer can write 50 lines of code in a day. In slim model we assumed productivity factor as 2.5 and productivity parameters 800. And finally project time 20 months.

3.2 Project Scheduling

Task Breakdown :

1. **Requirement Analysis:** Collect functional & non-functional requirements, stakeholder interviews & surveys, define user roles & permissions
2. **System Design:** Design database schema for donors, hospitals, blood units, UI/UX design for donor portal, staff dashboard, design APIs for hospital requests & logistics updates
3. **Implementation:** Coding and building whole system
4. **Testing:** Unit Testing, Integration Testing, System Testing
5. **Deployment & Documentation:** Deploy the system, staff training & documentation
6. **Maintenance & Future Improvements:** Bug fixing & updates, Monitor performance & reliability, Add new features

Task Dependency:

Requirement Analysis has no dependency

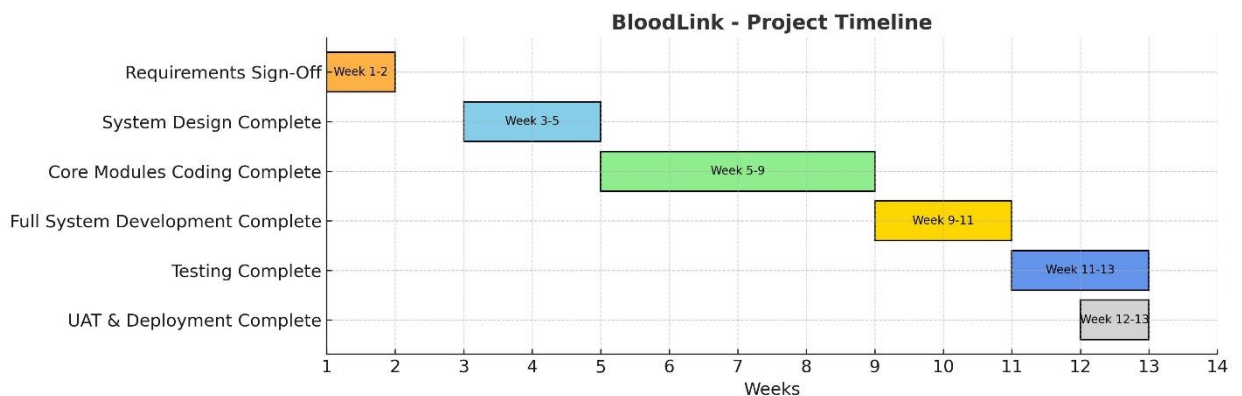
System Design depends on requirements analysis

Implementation depends on System Design
 Testing comes after implementation
 After testing System is deployed

Effort Allocation

Phase	% Effort	Tasks Included
Analysis & Design	40%	Requirement analysis, UI/UX design, database/API design
Coding / Implementation	20%	Development of portals, dashboards, logistics & admin modules
Testing & Deployment	40%	Unit & integration testing, UAT, deployment, training

Gantt chart



Major deliverables

- Requirement Analysis : Requirement documents (User stories & roles)
- Design Complete : Design documents (ER diagram & Wiireframes)
- Core Modules Coding : Donor & Blood Bank Modules
- System Complete : Full System (Integrated system for all users)
- Testing Complete : QA Reports
- Deployment Complete : Live system

Progress Tracking

- **Regular Meetings:** Weekly progress updates with team leads.
- **Milestone Reviews:** After design, coding, testing, and deployment.
- **Schedule Checks:** Compare actual progress vs planned milestones using simple % complete.

Earned Value Analysis (EVA)

Task	Planned Effort	Actual Effort
1	22	24
2	8	8.3
3	22	21.2
4	13	12.6
5	13	14
6	16	16.1
7	7	8.2
8	20	20.3
9	13	11.8
10	22	20
11	11	9.6
12	14	15.9
13	11	10.6
14	19	17.7
15	14	16.6
16	8	8.6
17	21	18.5
18	8	8.5
19	16	14.6
20	15	16.8
21	17	15.3
22	9	9.3
23	10	12.66
24	22	12.5
25	13	12.34
26	18	12.18
27	13	12.02
28	18	11.87
29	16	11.71
30	16	11.55
31	21	11.39
32	8	11.23
33	21	11.07

34	20	10.92
35	16.09	10.76
36	16.14	10.6
37	16.2	10.44
38	16.25	10.28
39	16.31	10.12
40	16.36	9.97
41	16.41	9.81
42	16.47	9.65
43	16.52	9.49
44	16.58	9.33
45	16.63	9.18
46	16.68	
47	16.74	
48	16.79	
49	16.85	
50	16.9	
51	16.95	
52	17.01	
53	17.06	
54	17.12	
55	17.17	
56	17.22	
57	17.28	
58	17.33	
59	17.39	
60	17.44	
61	17.49	
62	17.55	
63	17.6	
64	17.66	
65	17.71	
66	17.76	
67	17.82	
68	17.87	
69	17.92	
70	17.98	
71	18.03	
72	18.09	
73	18.14	
74	18.19	
75	18.25	
76	18.3	
77	18.36	

78	18.41	
79	18.46	
80	18.52	
81	18.57	
82	18.63	
83	18.68	
84	18.73	
85	18.79	
86	18.84	
87	18.9	
88	18.95	
89	19	
90	19.06	
91	19.11	
92	19.17	
93	19.22	
94	19.27	
95	19.33	
96	19.38	
97	19.43	
98	19.49	
99	19.54	
100	19.6	
101	16.93	
102	16.93	
103	16.93	
104	16.93	
105	16.93	
106	16.93	
107	16.93	
108	16.93	
109	16.93	
110	16.93	
111	16.93	
112	16.93	
113	16.93	
114	16.93	
115	16.93	
116	16.93	
117	16.93	
118	16.93	
119	16.93	
120	16.93	
121	16.93	

122	16.93	
123	16.93	
124	16.93	
125	16.93	
126	16.93	
127	16.93	
128	16.93	
129	16.93	
130	16.93	
131	16.93	
132	16.93	
133	16.93	
134	16.93	
135	16.93	
136	16.93	
137	16.93	
138	16.93	
139	16.93	
140	16.93	
141	16.93	
142	16.93	
143	16.93	
144	16.93	
145	16.93	
146	16.93	
147	16.93	
148	16.93	
149	16.93	
150	16.93	
151	16.93	
152	16.93	
153	16.93	
154	16.93	
155	16.93	
156	16.93	
157	16.93	
158	16.93	
159	16.93	
160	16.93	

Metric	Value
BAC (Budget at Completion)	2708.49
ACWP (Actual Cost of Work Performed)	569.57

BCWP (Budgeted Cost of Work Performed)	694.96
BCWS (Budgeted Cost of Work Scheduled)	1128.25
CV (Cost Variance)	125.39
SV (Schedule Variance)	-433.29
CPI (Cost Performance Index)	1.220148533
SPI (Schedule Performance Index)	0.615962774
% Scheduled	0.416560519
% Completed	0.256585773
Total Tasks	160

Calculations :

BAC (Budget at Completion) = 2708.49

If we assume 45 task is done from 70 scheduled task

BCWP (Budgeted Cost of Work Performed) = 694.96

BCWS (Budgeted Cost of Work Scheduled) = 1128.25

ACWP (Actual Cost of Work Performed) = 569.57

Schedule performance Index, SPI = (BCWP/BCWS) = 0.616

Schedule variance, SV = (BCWP – BCWS) = -433.29

% Scheduled = (BCWS/BAC) = 0.4166

% Completed = (BCWP/BAC) = 0.2566

Cost Performance Index, CPI = (BCWP/ACWP) = 1.2201

Cost Variance, CV = BCWP – ACWP = 125.39

As SPI is negative project is behind schedule. Negative SV represents the amount of work done is less than planned. CPI greater than 1 means we are cost efficient and we are on proper budget. Positive CV means we have spent less effort than we planned.

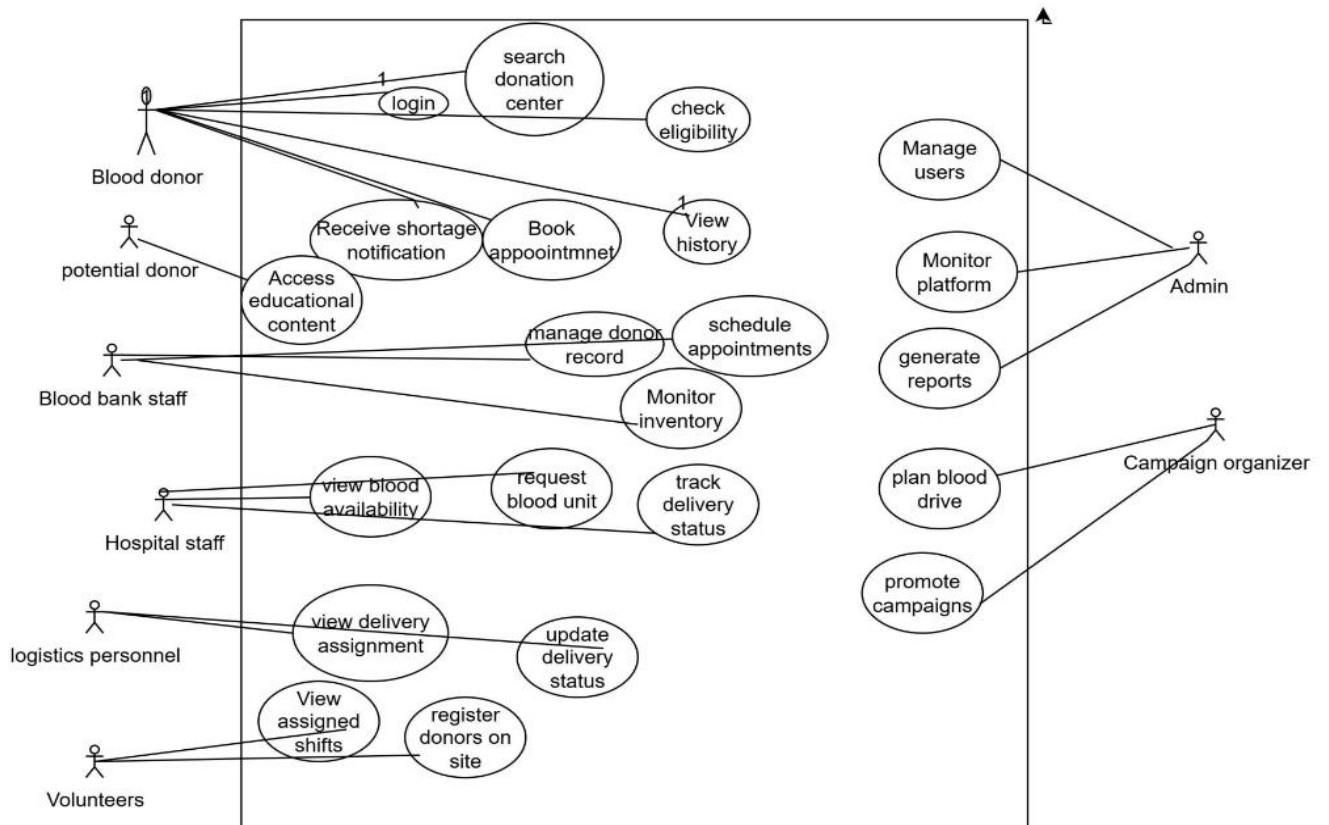
4. SOFTWARE DESIGN

4.1 System Design

This section presents the architectural blueprint of the BloodLink system through a series of Unified Modeling Language (UML) and data-flow diagrams. These models provide a comprehensive visual representation of the system's structure, user interactions, process workflows, and data movement, ensuring a clear and consistent understanding of the software's design.

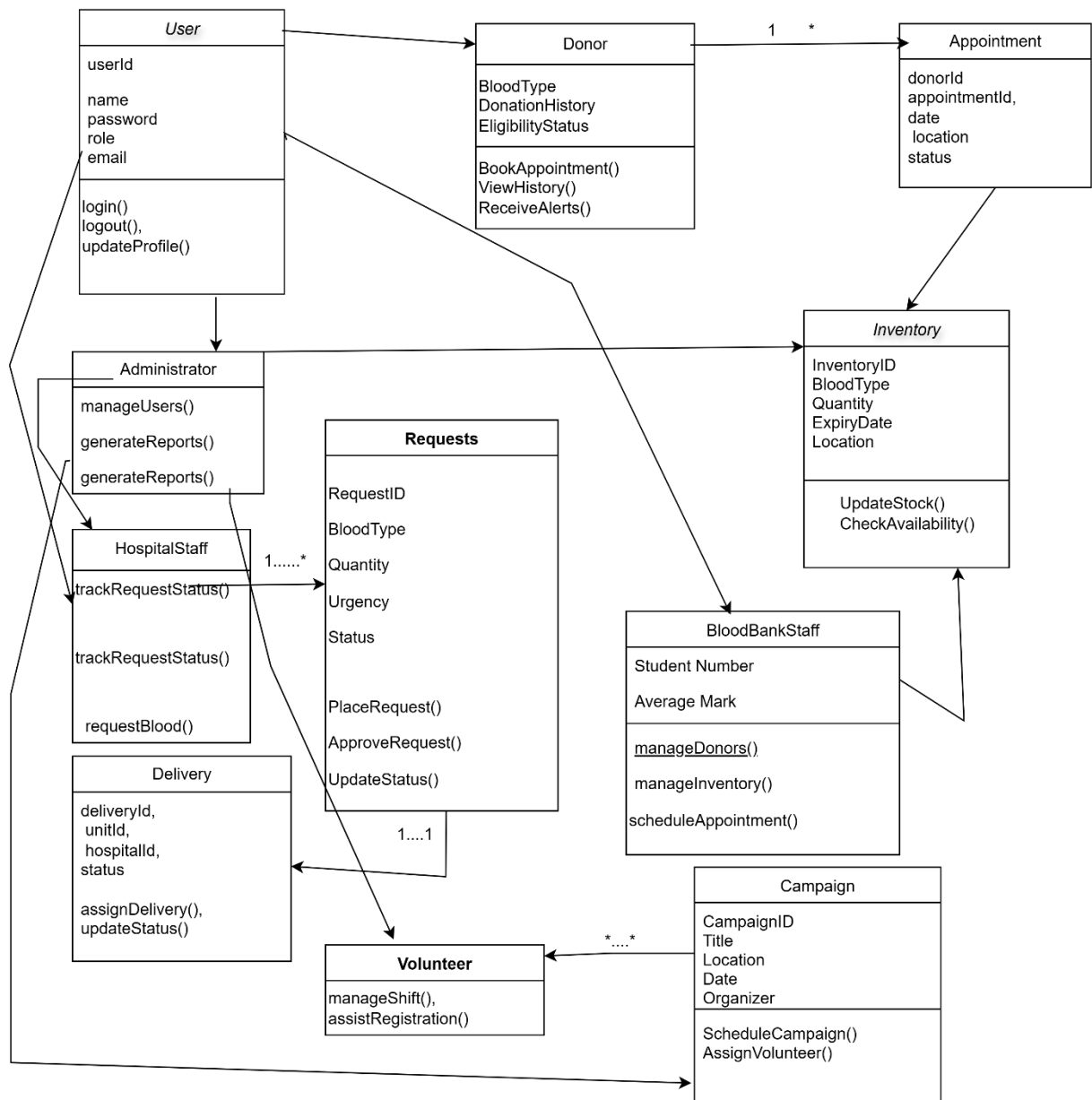
Use Case Diagram

The Use Case diagram identifies the primary actors of the system and illustrates their interactions with the system's core functionalities. It defines the system's boundary and shows how different users, such as the **Donor**, **Hospital Staff**, **Blood Bank Staff**, and **Administrator**, will utilize the platform to achieve their respective goals.



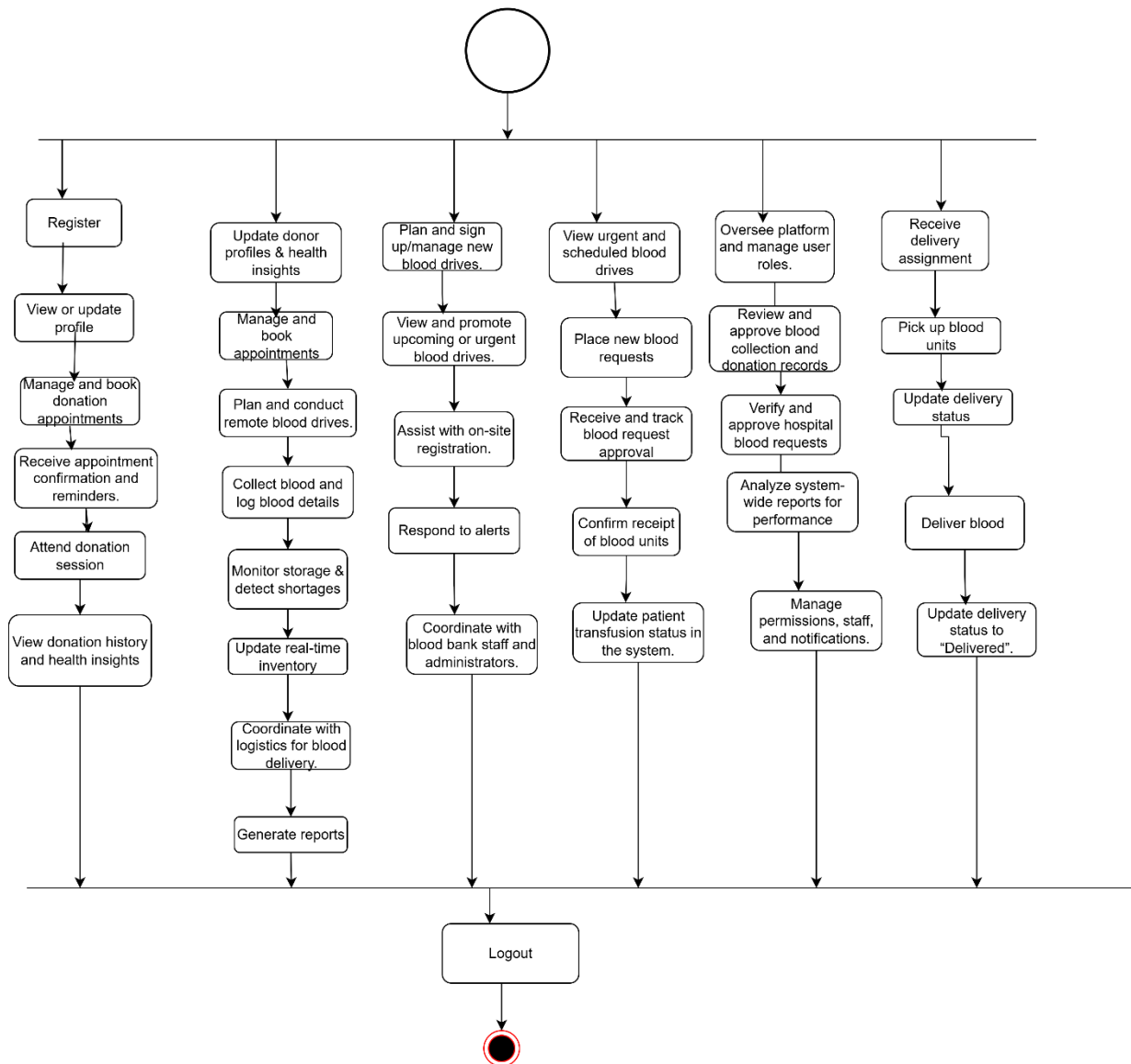
Class Diagram

The Class Diagram below models the static structure of the BloodLink system. It identifies the main classes, such as **User**, **Donor**, **BloodBankStaff**, **Appointment**, and **Inventory**, along with their attributes and operations. The diagram also maps the essential relationships between these classes, including inheritance (e.g., Administrator, HospitalStaff, and Donor are specialized types of User) and associations (e.g., a Donor can have one or more Appointments), providing a clear foundation for the system's object-oriented design.



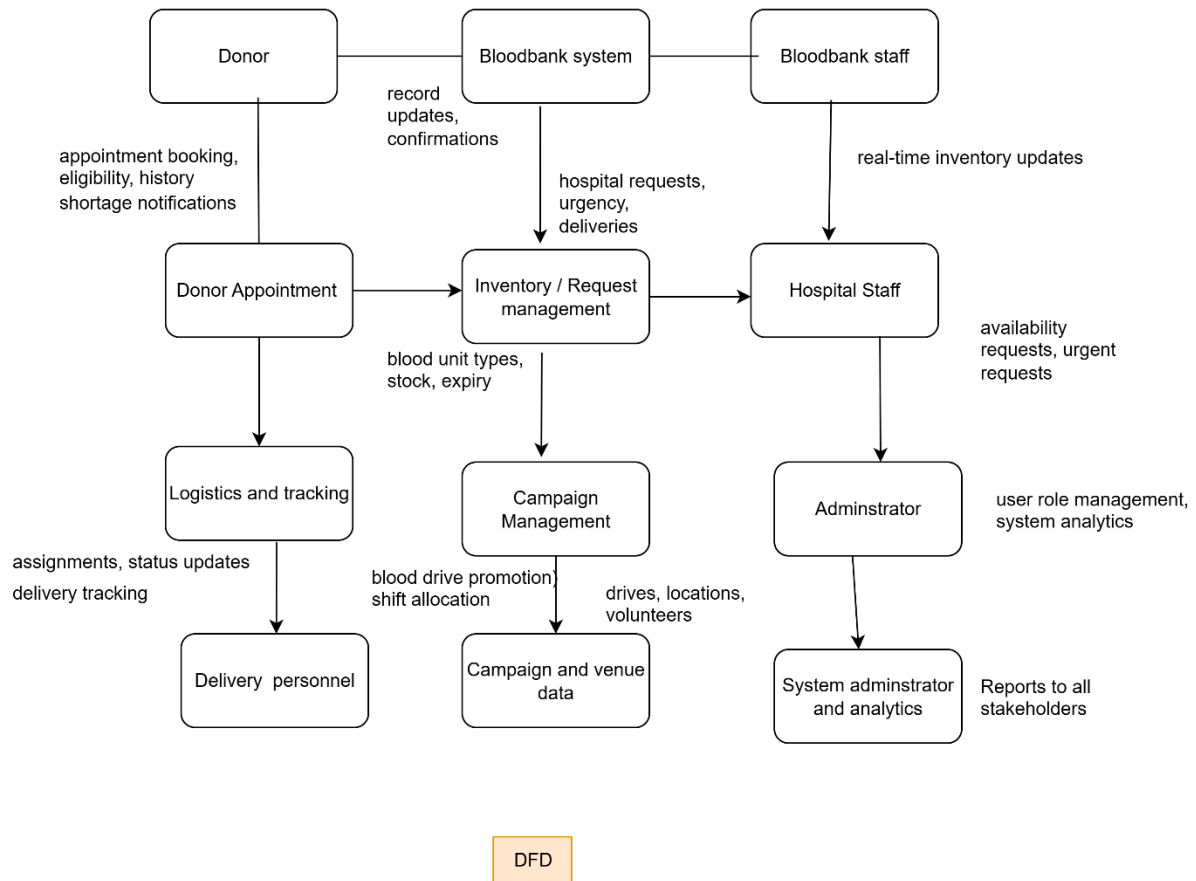
Activity Diagram

This Activity Diagram illustrates the dynamic workflows within the BloodLink system. It visualizes the flow of control from one activity to another, showing the parallel processes available to different user roles after they log in. Each vertical swimlane represents a primary actor—such as a **Donor**, **Blood Bank Staff**, **Campaign Organizer**, **Hospital Staff**, **Administrator**, and **Delivery Personnel**—and details the sequence of actions they can perform within the system before logging out.



Data Flow Diagram (DFD)

The Data Flow Diagram provides a high-level view of how information moves through the BloodLink system. It showcases the interactions between external entities (e.g., **Donor**, **Bloodbank staff**), key processes (e.g., **Inventory / Request management**), and data stores (e.g., **Campaign and venue data**). This diagram effectively maps the path of data from its origin to its destination, highlighting how different components of the system exchange information to function cohesively.



4.2 UI / Wireframe Design

The user interface was designed using Figma. The following images are key screens from the clickable prototype, illustrating the application's user flow and core functionalities.

Figma Prototype Link:

<https://www.figma.com/design/40ppaVqeaA0ij7iD21jJwX/BloodLink?m=auto&t=YtjEWWr25mnmI1Cn-1>

Home Page



Your Donation Can Save a Life Today

[Find a Donation Center](#)

Boosts Donor Engagement

Engage with a community of donors, track your impact, and receive personalized updates.



Minimizes Blood Wastage

Reduce blood wastage through efficient inventory management and real-time demand matching.



Enhances Emergency Response

Improve emergency response times with a streamlined blood donation and distribution process.

Signup Page

Create your account

Full Name

Email Address

Password

Confirm Password

☐ I agree to the Terms & Conditions

[Create Account](#)

Login Page

Welcome back

Email Address

Enter your email

Password

Enter your password

Remember me ☐

[Forgot Password?](#)

Login

Donor Dashboard

 Sophia Carter

 My Profile

 Donation History

 Appointments

 Settings

My Dashboard

Lives Saved

3

Upcoming Appointments

July 2024						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Donation History

Date	Amount	Type	Status
June 15, 2024	\$50	Blood	Completed
May 20, 2024	\$75	Plasma	Completed
April 5, 2024	\$100	Blood	Completed

Next Donation Date

15

Days

08

Hours


30

Minutes


12

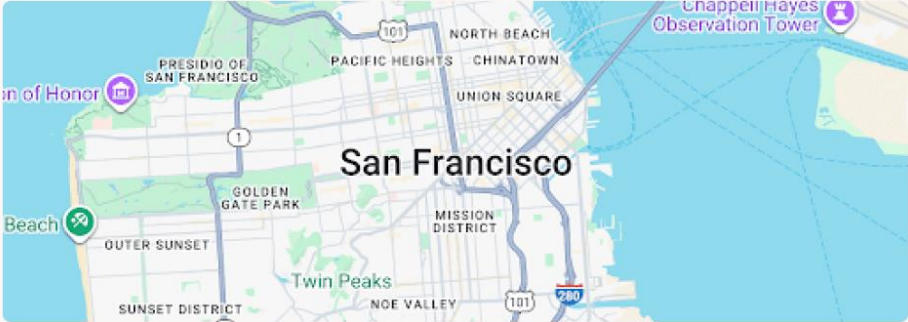
Seconds

Donor - Find & Book Appointment


BloodLink

[Find a drive](#)
[My appointments](#)
[My profile](#)







Nearby Donation Centers

San Francisco Blood Center 123 Main Street, San Francisco, CA	Book Appointment
Community Blood Drive 456 Oak Avenue, San Francisco, CA	Book Appointment
Mobile Blood Drive 789 Pine Lane, San Francisco, CA	Book Appointment

Donor - Donation History & Impact







BloodLink


[Home](#)
[Explore](#)
[My Impact](#)



My Impact

See the difference you're making in the world.

- 
Donation to Education Fund
 July 15, 2023 - Education in Rural Areas
- 
Support for Environmental Conservation
 June 20, 2023 - Rainforest Preservation
- 
Contribution to Healthcare Initiative
 May 5, 2023 - Medical Supplies for Clinics
- 
Donation to Disaster Relief
 April 10, 2023 - Earthquake Relief in South Asia
- 
Support for Animal Welfare
 March 1, 2023 - Animal Shelter Support



Your Impact

See the difference you're making in the world.

Your donations have helped plant 100 trees, educate 50 children, and provide medical care to 200 people.

Hospital Staff Dashboard

Blood Inventory Dashboard

Real-Time Blood Inventory

Type A+

Sufficient

Type B-

Low

Type O+

Critical

Type AB+

Sufficient

Place Urgent Blood Request

Active Blood Requests

Request ID	Blood Type	Status	Estimated Delivery
REQ12345	O+	Pending	15 minutes
REQ67890	A-	In Transit	30 minutes
REQ11223	B+	Delivered	Completed
REQ44556	AB-	Pending	20 minutes
REQ77889	O-	In Transit	45 minutes

Hospital Staff - Track Delivery Status

Delivery Status

- ✓ Request Received
10:00 AM
- ✓ Assigned to Driver
10:15 AM
- ✓ In Transit
10:30 AM
- ✓ Out for Delivery
10:45 AM
- ✓ Delivered
11:00 AM

Driver Information

Ethan Carter
Driver

(555) 123-4567
Phone

11:00 AM
Estimated Arrival

Blood Delivery Tracking

Order #123456

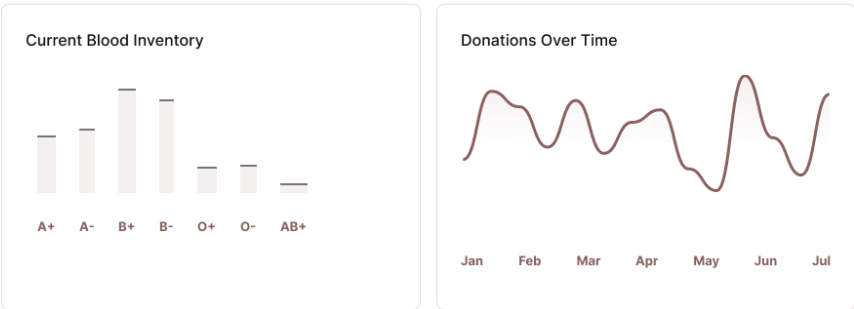


Blood Bank Staff Dashboard

- Dashboard
- Donors
- Inventory
- Appointments
- Reports

Dashboard

Real-Time Data



Upcoming Appointments

Donor Name	Appointment Date	Appointment Time	Blood Type	Status	Actions
Ethan Carter	2024-07-15	10:00 AM	A+	Pending	Approve/Reject
Olivia Bennett	2024-07-16	11:30 AM	B-	Pending	Approve/Reject
Noah Thompson	2024-07-17	09:00 AM	O+	Pending	Approve/Reject
Ava Harper	2024-07-18	02:00 PM	AB+	Pending	Approve/Reject
Liam Foster	2024-07-19	03:30 PM	O-	Pending	Approve/Reject


Notifications



Soumik Das Dipon

New donor registration

Blood Bank Staff - Organize Blood Drives




Dashboard


Donors

Drives

Inventory

Reports





Organize Blood Drives

Plan and manage blood donation events to meet community needs.

Plan a New Drive

Event Name

e.g., Community Blood Drive

Date

Select Date

Time

Select Time

Venue

e.g., Community Center

Donation Goal


e.g., 50 units

Schedule Drive


Upcoming Drives


Event	Date	Venue	Goal	Status	Actions
Community Blood Drive	2024-07-15	Community Center	50 units	Scheduled	Promote View Volunteers
Regional Blood Drive	2024-08-20	Regional Hospital	100 units	Scheduled	Promote View Volunteers
University Blood Drive	2024-09-10	University Campus	75 units	Scheduled	Promote View Volunteers


Blood Bank Staff Page

BloodLink

[Dashboard](#)[Donors](#)[Inventory](#)[Requests](#)[Reports](#)


 Search





Donor Management

Add New Donor

 Search donors by name or ID

Name	Blood Type	Contact Number	Last Donation Date	Eligibility
Ethan Carter	O+	555-123-4567	2023-11-15	Eligible
Olivia Bennett	A-	555-987-6543	2023-10-20	Eligible
Noah Thompson	B+	555-246-8013	2023-09-05	Ineligible
Ava Rodriguez	AB+	555-369-1470	2023-12-01	Eligible
Liam Walker	O-	555-789-0123	2023-11-22	Eligible
Isabella Hayes	A+	555-456-7890	2023-10-10	Eligible
Jackson Reed	B-	555-654-3210	2023-08-15	Ineligible
Sophia Morgan	AB-	555-111-2222	2023-12-10	Eligible
Aiden Cooper	O+	555-333-4444	2023-11-05	Eligible
Mia Foster	A-	555-555-6666	2023-09-20	Ineligible

Admin Dashboard

Admin Dashboard

Overview of platform operations and key metrics

Total Donors

12,345

Total Donations

8,765

Urgent Requests
Fulfilled

456

Total Blood Drives

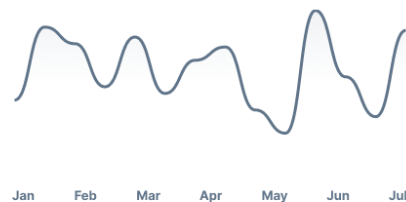
234

System-wide Analytics

Donations Over Time

+15%

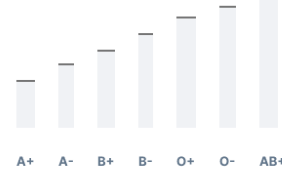
Last 30 Days **+15%**



Donations by Blood Type

2,345

Last 30 Days **+10%**



Manage User Roles and Permissions

User	Role	Permissions	Actions
Emily Carter	Admin	Full Access	Edit
Lucas Bennett	Donor	View Only	Edit
Chloe Foster	Coordinator	Manage Drives	Edit
Owen Harper	Donor	View Only	Edit
Isabella Hayes	Coordinator	Manage Requests	Edit

5. GIT WORKFLOW

For the "BloodLink" project, our team used a structured Git workflow on GitHub to manage development and collaboration.

5.1 Central Repository and Branching Strategy

A central repository was created on GitHub (<https://github.com/Rezwoan/BloodLink>) to host the project. The main branch was used as the primary branch for the stable version of our application. Each team member was assigned a personal developer branch (e.g., ayexha-dev, nusrat-dev, etc.) to work on their tasks independently.

5.2 Development and Collaboration Process

Our workflow consisted of the following steps:

1. Each member cloned the repository and worked within their assigned developer branch.
2. Changes were committed regularly with messages describing the work done.
3. The developer branches were pushed to the remote repository on GitHub.
4. Pull requests were used to merge the developer branches into the main branch. This allowed the project lead to review the code before integrating it into the main codebase.

5.3 Evidence of Collaboration

The commit history on our GitHub repository provides clear evidence of collaboration, with multiple commits, branches, and merges from all group members. A screenshot of the commit history will be added to this section.



5.4 Repository Organization

To maintain a clear project history, all developer branches were kept after being merged. This tracks the project's workflow from initialization to completion.

The project repository can be found at the following link: <https://github.com/Rezwoan/BloodLink>

6. TESTING

We performed a total of 16 manual test cases to check the quality and functionality of the BloodLink system. These tests covered essential features across the application, including user login and registration, donor profile management, blood bank searches, and hospital blood requests.

Out of the 16 tests executed, 10 passed and 6 failed. The failed tests helped us identify critical bugs and areas for improvement. The following tables provide a detailed breakdown of each test case.

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_01			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Soumik Das Dipon	
Module Name: Auth Feature			Test Execution date: 26/09/2025	
Test Title: Verify user login with valid credentials				
Description: Tests if a registered donor can successfully log in to the application				
Precondition: The user must have a registered and active account.				
Dependencies: None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Enter a valid registered email. 3. Enter the correct password. 4. Click "Login".	Email: donor@test.com, Password: Password123	The user should be redirected to their personal dashboard.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_02		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Soumik Das Dipon		
Module Name: Auth Feature		Test Execution date: 26/09/2025		
Test Title: Verify user login with invalid credentials				
Description: Tests if the system prevents access when incorrect login details are provided.				
Precondition: The user attempts to log in.				
Dependencies: None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Enter a valid registered email. 3. Enter an incorrect password. 4. Click "Login".	Email: donor@test.com, Password: WrongPassword	An error message "Invalid email or password" should be displayed.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_03		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Soumik Das Dipon		
Module Name: Auth Feature		Test Execution date: 26/09/2025		
Test Title: Verify new donor profile creation with valid data				
Description: Test the user registration functionality.				
Precondition: The user is not already registered with the provided email.				
Dependencies: None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the registration page. 2. Fill in all required fields. 3. Click "Register".	Name: John Doe, Email: johndoe.new@test.com, Blood Type: O+	User account is created successfully and user is redirected to the login page or dashboard.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_04		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Soumik Das Dipon		
Module Name: Donor Profile		Test Execution date: 26/09/2025		
Test Title: Verify donor can update their profile information				
Description: Tests if a logged-in donor can modify their personal details.				
Precondition: The user is logged in to their account.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to "My Profile". 2. Click "Edit Profile". 3. Change the contact number. 4. Click "Save Changes".	New Contact Number: 01234567891	A success message should be displayed and the new contact number should be visible.	Page reloads, but the contact number remains unchanged. No success or error message is displayed.	Fail

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_05		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Sanjana Sayed Nuha		
Module Name: Auth Feature		Test Execution date: 26/09/2025		
Test Title: Verify the "Forgot Password" functionality				
Description: Tests if the password reset mechanism is working correctly.				
Precondition: The user has an existing account but has forgotten their password.				
Dependencies: Email service integration must be functional.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page. 2. Click "Forgot Password?". 3. Enter the registered email. 4. Click "Send Reset Link".	Email: donor@test.com	User should see a confirmation message and receive a password reset email.	Confirmation message is shown, but no email is received after 10 minutes.	Fail

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_06		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Sanjana Sayed Nuha		
Module Name: Donor Profile		Test Execution date: 26/09/2025		
Test Title: Verify donor can view their donation history				
Description: Tests if past donation records are displayed correctly for a logged-in user.				
Precondition: The user is logged in and has made at least one donation previously.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as a donor. 2. Navigate to the "Donation History" section.	None	A list of past donations should be displayed in chronological order.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_07		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Sanjana Sayed Nuha		
Module Name: Blood Bank Portal		Test Execution date: 26/09/2025		
Test Title: Verify blood bank staff can search for donors by blood type				
Description: Tests the donor search functionality for blood bank staff.				
Precondition: The user is logged in as Blood Bank Staff.				
Dependencies: Multiple donors with different blood types exist in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Blood Bank Staff. 2. Navigate to "Search Donors". 3. Select "A+" from the filter. 4. Click "Search".	Blood Type: A+	Search results should display a list of all registered donors with blood type "A+".	As expected	Pass

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_08			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Sanjana Sayed Nuha	
Module Name: Blood Bank Portal			Test Execution date: 26/09/2025	
Test Title: Verify donor search filter accuracy				
Description: Tests that the search filters are not returning incorrect results.				
Precondition: The user is logged in as Blood Bank Staff.				
Dependencies: Database contains donors of types O+ and O-.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Blood Bank Staff. 2. Navigate to "Search Donors". 3. Select "O+" from the filter and search.	Blood Type: O+	The results should only display donors with blood type "O+".	The search results list contains donors with both "O+" and "O-" blood types. The filter is not specific.	Fail

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_09			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): High			Test Executed by: Maoun Billah	
Module Name: Hospital Portal			Test Execution date: 26/09/2025	
Test Title: Verify hospital staff can create a blood request				
Description: Tests the blood request submission process for hospitals.				
Precondition: The user is logged in as Hospital Staff.				
Dependencies: The system has available blood units.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Hospital Staff. 2. Go to "Request Blood". 3. Select a blood type and enter units. 4. Click "Submit Request".	Blood Type: B-, Units:	Request is submitted successfully and appears in the "Pending Requests" list.	User is redirected to a "500 Internal Server Error" page. The request is not created.	Fail

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_10			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Maoun Billah	
Module Name: Admin Portal			Test Execution date: 26/09/2025	
Test Title: Verify admin can view system logs				
Description: Tests if an administrator can access and view system activity logs.				
Precondition: The user is logged in as an Administrator.				
Dependencies: There is recent user activity on the platform.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an Administrator. 2. Navigate to "System Monitoring" -> "View Logs".	None	A chronological list of system events should be displayed.	The page loads but the log area is empty with a "Failed to load" message.	Fail

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_11			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): Low			Test Executed by: Maoun Billah	
Module Name: Campaign Organizer Portal			Test Execution date: 26/09/2025	
Test Title: Verify campaign organizer can create a new donation event				
Description: Tests the creation of a new blood donation campaign.				
Precondition: The user is logged in as a Campaign Organizer.				
Dependencies: None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as a Campaign Organizer. 2. Go to "Campaigns" and click "Create New Event". 3. Fill in all event details. 4. Click "Save Event".	Event Name: City Center Blood Drive, Date: 30/09/2025	Event is created successfully and appears in the "Upcoming Campaigns" list.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_12		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Maoun Billah		
Module Name: Donor Profile		Test Execution date: 26/09/2025		
Test Title: Verify donor can schedule a new appointment at a campaign				
Description: Tests if a donor can sign up for an upcoming donation event.				
Precondition: The user is logged in as a donor, and an active campaign exists.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as a donor. 2. Navigate to "Upcoming Events". 3. Select an event and time slot. 4. Click "Book Appointment".	None	Appointment is booked successfully and appears in the donor's "My Appointments" section.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_13		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Din Muhammad Rezwoan		
Module Name: Admin Portal		Test Execution date: 26/09/2025		
Test Title: Verify admin can deactivate a user account				
Description: Tests the user management feature for administrators.				
Precondition: The user is logged in as an Administrator.				
Dependencies: At least one active donor account exists.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an Administrator. 2. Go to "User Management". 3. Search for a donor. 4. Click "Deactivate". 5. Confirm action.	User Email: johndoe.new@test.com	User's status changes to "Inactive" and they can no longer log in.	As expected	Pass

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_14			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Din Muhammad Rezwoan	
Module Name: Blood Bank Portal			Test Execution date: 26/09/2025	
Test Title: Verify blood bank staff can update blood unit inventory				
Description: Tests that staff can add new blood units to the system inventory.				
Precondition: The user is logged in as Blood Bank Staff.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Blood Bank Staff. 2. Go to "Inventory Management". 3. Click "Add New Unit" and fill details. 4. Click "Save".	Blood Type: AB+	New blood unit is added to the inventory and visible with "Available" status.	As expected	Pass

Project Name: BloodLink		Test Designed by: Din Muhammad Rezwoan		
Test Case ID: TC_15		Test Designed date: 26/9/2025		
Test Priority (Low, Medium, High): High		Test Executed by: Din Muhammad Rezwoan		
Module Name: Hospital Portal		Test Execution date: 26/09/2025		
Test Title: Verify hospital staff can view the status of their blood requests				
Description: Tests if hospital staff can track their submitted requests.				
Precondition: The user is logged in as Hospital Staff and has submitted at least one request.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Hospital Staff. 2. Navigate to the "Request History" page.	None	A list of all submitted blood requests with their current statuses should be displayed.	As expected	Pass

Project Name: BloodLink			Test Designed by: Din Muhammad Rezwoan	
Test Case ID: TC_16			Test Designed date: 26/9/2025	
Test Priority (Low, Medium, High): Medium			Test Executed by: Din Muhammad Rezwoan	
Module Name: Donor Profile			Test Execution date: 26/09/2025	
Test Title: Verify user cannot update their email to an existing email				
Description: Tests data integrity by preventing duplicate emails during profile updates.				
Precondition: Two distinct user accounts exist (donor@test.com and donor2@test.com). User donor2@test.com is logged in.				
Dependencies: A valid user session.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as donor2@test.com. 2. Go to "Edit Profile". 3. Change email to an existing user's email. 4. Click "Save Changes".	Email: donor@test.com	An error message "Email address is already in use" should be displayed.	A generic "Profile updated" message is shown and the user is logged out. The account becomes inaccessible.	Fail

7. SOFTWARE PRODUCT METRICS

Whitmire's OO Design Metrics

Size

Number of classes: 11

(User, Donor, BloodBankStaff, HospitalStaff, Administrator, Volunteer, Campaign, Appointment, Inventory, Delivery, Requests)

Number of modules (features): 36

Complexity

Classes are interconnected with multiple other classes. For example:

Donor connects with Appointment, User, and Campaign.

BloodBankStaff connects with Inventory, HospitalStaff, and indirectly Donor.

Delivery connects with Inventory, HospitalStaff, and Administrator

Coupling

Classes and their coupled classes:

- User - Donor, BloodBankStaff, HospitalStaff, Administrator, Campaign, Volunteer
- Donor - User, Appointment, Campaign
- BloodBankStaff - User, Donor, HospitalStaff, Inventory
- HospitalStaff - BloodBankStaff, Inventory, Delivery
- Administrator - User, Inventory, Delivery
- Campaign - User, Donor, Appointment
- Volunteer - User, Appointment
- Appointment - Donor, Volunteer, Campaign
- BloodUnit - BloodBankStaff, HospitalStaff, Delivery, Administrator
- Delivery - Inventory, HospitalStaff, Administrator
- Requests – User, HospitalStaff, Administrator

Cohesion (LCOM values approximation)

- User: 3
- Donor: 3
- BloodBankStaff: 3
- HospitalStaff: 3
- Administrator: 3
- Volunteer: 2
- Campaign: 2
- Appointment: 0
- Inventory: 2
- Delivery: 2
- Requests: 3

Sufficiency

- Most classes (Donor, Appointment, Inventory, Delivery) cover their intended responsibilities.
- User acts as a base identity class, ensuring role-based access.
- Administrator and BloodBankStaff have slightly overlapping responsibilities (inventory, reporting).

Completeness

- Domain entities (Inventory, Appointment, Delivery) are reusable.
- Role-based classes (Donor, HospitalStaff, Volunteer) are specific to the system.
- Admin-related tasks are less reusable due to role concentration.

Primitiveness

- Most operations are atomic (e.g., manageDonors(), requestBlood(), assignDelivery()).

Similarity

- User-type classes: Donor, BloodBankStaff, HospitalStaff, Volunteer, Administrator share login/role-related behaviors.
- Domain-entity classes: Inventory, Appointment, Delivery are persistent entities with CRUD-like operations.
- Role-specialization classes: Staff roles and organizer/volunteer extend the system with specialized responsibilities.

Function Based Metrics

External Inputs (EIs):

- User registration/login/logout
- Update profile
- Book appointment (donor)
- Schedule appointment (staff)
- Manage donors (staff)
- Manage inventory (staff)
- Request blood (hospital staff)

- Assign delivery (admin)
- Update delivery status (delivery)
- Plan/promote drive (organizer)
- Volunteer shift management

External Outputs (EOs):

- View donor history
- View eligibility
- View hospital request status
- View appointment list
- Generate reports (admin, staff)
- Track delivery status
- Campaign/drive report
- Notifications/alerts

Internal Logical Files (ILFs):

- User records
- Donor records
- Appointment records
- Blood inventory
- Delivery records
- Campaign records

External Interface Files (EIFs):

- Notification system (alerts/email/SMS)
- Hospital integration

External Inquiries (EQs):

- View appointment status
- Track request status
- View donor records
- View delivery assignment
- View campaign performance

Function Point Calculation:

$$\text{EIs: } 12 \times 4 = 48$$

$$\text{EOs: } 8 \times 5 = 40$$

$$\text{ILFs: } 6 \times 10 = 60$$

$$\text{EIFs: } 2 \times 7 = 14$$

$$\text{EQs: } 5 \times 4 = 20$$

Total Function Points = 182

Object-Oriented and Class Metrics

Donor Class

- Methods: 3
- WMC: 3
- DIT: 1
- NOC: 0
- CBC: 3 (User, Appointment, Campaign)
- LCOM: 3

Appointment Class

- Methods: 0
- WMC: 0
- DIT: 0
- NOC: 0
- CBC: 3 (Donor, Volunteer, Campaign)
- LCOM: 0

BloodBankStaff Class

- Methods: 3
- WMC: 3
- DIT: 1
- NOC: 0
- CBC: 3 (Donor, Inventory, HospitalStaff)
- LCOM: 3

HospitalStaff Class

- Methods: 2
- WMC: 2
- DIT: 1
- NOC: 0
- CBC: 3 (BloodBankStaff, Inventory, Delivery)
- LCOM: 3

Administrator Class

- Methods: 3
- WMC: 2

- DIT: 1
- NOC: 0
- CBC: 3 (User, Inventory, Delivery)
- LCOM: 3

Volunteer Class

- Methods: 2
- WMC: 2
- DIT: 1
- NOC: 0
- CBC: 2 (User, Appointment)
- LCOM: 2

Campaign Class

- Methods: 2
- WMC: 2
- DIT: 1
- NOC: 0
- CBC: 2 (User, Donor, Appointment)
- LCOM: 2

Inventory Class

- Methods: 2
- WMC: 2
- DIT: 0
- NOC: 0
- CBC: 4 (BloodBankStaff, HospitalStaff, Delivery, Admin)
- LCOM: 2

Delivery Class

- Methods: 2
- WMC: 2
- DIT: 0
- NOC: 0
- CBC: 3 (BloodUnit, HospitalStaff, Administrator)
- LCOM: 2

Operation-Oriented Metrics

Total methods/operations: **26**
Average operations per class: **2.36**
Parameters per method: 1.11

Software Maturity Index

Total modules, $MT = 36$
added modules, $Fa = 3$
Changed modules, $Fc = 6$
Deleted modules, $Fd = 1$

$$SMI = \frac{MT - (Fa + Fc + Fd)}{MT}$$

$$SMI = \frac{36 - (3 + 6 + 1)}{36}$$

$$SMI = 0.7222$$

SMI is not 1 but it is little close to it. The closer it is to 1 the better it is.

8. CONCLUSION AND FUTURE WORK

Conclusion

The main goal of the BloodLink project was to fix the disconnected and often slow process of blood donation. We've designed a single, easy-to-use platform that connects everyone involved: donors, blood banks, and hospitals. By carefully planning all the features and the system's structure, we've created a solid blueprint for a tool that will make donating and receiving blood much simpler and faster.

This system is built to encourage more people to donate, ensure less blood goes to waste, and get blood to patients much faster in an emergency. In the end, BloodLink is all about improving the entire process to help save lives.

Future Work

This project is a strong starting point, but there's always room to make it even better. Here are a few ideas for what could come next:

- **A Mobile App:** Build a simple app for phones so donors can book appointments and get urgent alerts on the go.
- **Smarter Predictions:** Use AI to predict where blood shortages might happen, so we can be better prepared.
- **Make Donating More Fun:** Add game-like features like points, badges, and rewards to encourage people to donate more often.

- **Connect with Hospital Systems:** Link our platform directly with official hospital records to make everything run more smoothly.
- **Better Tools for Blood Drives:** Improve the tools for event organizers to help them manage volunteers and blood drives more easily.

By adding these features over time, BloodLink can become an even more essential tool for our community's health.