DEVELOPMENT OF A WEB-BASED BLOOD BANK MANAGEMENT SYSTEM FOR HOSPITALS

BY

OKOLO CHUKWUELOKA
GOU/U22/CSC/837

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY,
GODFREY OKOYE UNIVERSITY, ENUGU STATE. IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF
SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: ENGR. CAMILUS NJOKU

JUNE, 2026.

APPROVAL PAGE

This research, titled "**Design And Implementation Of A Web Based Blood Bank System For Hospitals**," has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

Engr. Camilus Njoku	……	…….
Supervisor	Signature	Date
External Examiner	.....	.....
External Examiner	Signature	Date
Dr. Frank Okebanama	.....	.....
HOD, Department of Computer Science and Mathematics	Signature	Date
Prof. I.A Ajah	.....	.....
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

_____

OKOLO CHUKWUELOKA

GOU/U22/CSC/837

DEDICATION

This work **"Design And Implementation Of A Web Based Blood Bank System For Hospitals"** is dedicated to my institution, Godfrey Okoye University. I also dedicate this to my loving and supportive family for providing all I have ever needed and most importantly I thank God almighty for his grace, endless love and protection.

ACKNOWLEDGMENTS

My humble gratitude goes to God for his continuous grace, mercy and provisions throughout my final year.

I acknowledge my dear supervisor, who made sure my project turned out perfectly well, thank him for his time, support, patience and kindness which he showed me throughout my final year's journey.

Finally, I wish to pay heartfelt homage to all my lecturers who, through their unwavering dedication over the past four years, have instilled in me the finest principles and practices in computing. Your tireless efforts have not gone unnoticed, and I remain deeply grateful.

I thank my siblings for all support they gave me during the implementation process including paying for software tools and providing emotional and technical support.

To every reader of this work, I extend my warmest regards and a sincere prayer: May God continue to bless and guide you abundantly.

ABSTRACT

Blood banks play a critical role in healthcare delivery because timely availability of safe blood can determine patient survival during emergencies, surgeries, childbirth complications, trauma cases, and chronic medical conditions. Many blood donation and distribution processes still depend on manual records, fragmented communication, and delayed stock updates, which can cause inventory errors, difficulty in locating compatible blood units, poor request tracking, and wastage through expiry. This study designed and implemented a web-based Blood Bank Management System to improve donor registration, donation recording, blood inventory monitoring, hospital request management, and administrative reporting. The system was developed as a full-stack web application using Django, Django REST Framework, JSON Web Token authentication, SQLite database, React, Vite, React Router, and Axios. The backend provides RESTful endpoints for authentication, donors, donations, inventory, hospitals, requests, and reports, while the frontend provides role-based user interfaces for administrators, blood bank staff, and hospital users. The implemented system records each donation, automatically creates an inventory unit, tracks expiry dates, restricts functions by user role, allows hospitals to submit requests, and enables staff to approve, reject, reserve, and fulfill requests. The result is a functional management platform that improves accuracy, visibility, accountability, and speed in blood bank operations. The project contributes a practical digital solution that can be expanded with notifications, analytics, barcode scanning, and deployment to a production database.

Table of Contents

List of Tables

List of Figures

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Blood is a crucial medical commodity used in emergency situations, surgery procedures, maternal healthcare, oncology, anemia treatment, trauma cases, and treating patients with some blood diseases like sickle cell disease and thalassemia. The presence of safe, sufficient, and compatible blood is a key factor in determining survival chances of patients in most hospitals since emergencies like road accidents, complications during delivery, excessive bleeding, and urgent surgical procedures may require immediate attention from a blood bank. Millions of units of blood are collected every year around the world, but there are still lots of problems with blood shortage in low- and middle-income countries. In Nigeria, the problem of lack of blood becomes especially clear because the demand for blood is much bigger than the total amount collected per year.

A blood bank is a crucial institution, responsible for collecting, testing, storing, monitoring, and issuing blood. In order to be effective, a blood bank should keep correct records regarding donors, collected blood units, blood groups, compatibility reports, expiry dates, hospital requisitions, and issued blood units status. Proper record-keeping also implies keeping information about a hospital, ward, or patient who got a certain blood unit. It is important at this level of coordination because blood is a delicate medical product with an expiry period and any mistake done in its management will result in wastage, delays, or clinical risks.

Nonetheless, in many hospitals, especially in the developing nations like Nigeria, the management of blood banks continues to be manual or spreadsheet based. There are a number of shortcomings associated with manual blood bank management. It could be challenging for hospital staff members to identify the available blood units promptly, recognize low availability levels, identify any blood that may have expired, establish compatibility records during emergencies, and track released blood units

during audit periods. All the above mentioned shortcomings become severe under emergencies as delay in identifying available blood may influence the outcome of the procedures carried out. In addition, paper-based records may easily get lost, duplicated, damaged or become impossible to update. This makes the whole system unreliable.

Increasing need for fast, accurate and reliable healthcare services has rendered the need for computerization of blood bank management very crucial. Information and communication technologies have revolutionized many aspects of healthcare service provision through improved record keeping, speedy information search, enhanced decision-making and service delivery.As far as blood banking is concerned, a computerized system can be used to create donor profiles, keep a track of the inventory of blood after donation, check the expiry dates of the stock, create alerts when the stock is running low, check blood group compatibility, and even allow hospitals to make blood requests online. In this way, it makes the blood banking process less reliant on paper work and more efficient.

A web-based blood bank management system is quite helpful in the sense that it allows different individuals to access the system depending upon their respective roles. The administrator will have the ability to manage the users and keep an eye on all the activities, the blood bank people will have the task to register the donors and update the inventory of blood, while the hospital users will be able to request and check the status of their blood requests. This role-based access is a better way of organizing things along with adding to the security and availability of information.

Some of the researches done show the benefits of having a digital blood bank. In modern times, blood bank systems are not limited to mere storage logs; there are functionalities like request logs, compatibility information, donor information database, inventory dashboard, and audit trail. These are some indications that the use of technology in managing blood banks is very much needed and practical. This is even more so in the case of hospitals where timely access to blood becomes important.

In reaction to the inefficiencies associated with the manual management of blood banks, the study aims to come up with a web-based blood bank management system. The system is built with Django REST API at the backend and React as the

frontend. The system serves the interests of the administrators, the blood bank personnel, and the hospital users; thus, bringing into play all major stakeholders in the blood management process. The proposed system will serve the purpose of automating the registration of donors, blood inventories, hospital blood requests, issuance of units, and recording of the transactions.

1.2 Statement of the Problem

Operations in blood banks are faced with issues such as incorrect donor records, delayed updating of blood inventory, ineffective request monitoring, lack of visibility of blood groups available, and difficulty in monitoring expired/low inventory units. These processes in most hospitals are done either manually or by use of fragmented records, which makes it a slow and ineffective process. Manual records are prone to loss, duplication, damage, or delay, thus, denying hospital staff access to up-to-date records whenever needed.

In most cases, hospitals may encounter problems while requesting for blood since there is no common system for processing, approving, reserving, and fulfilling request for blood. Blood bank staff will be in a position that will make them unable to determine which blood units are available, reserved, issued, and those which have become expire. In case of emergency situations where blood compatibility is very important, there might arise problems between hospitals and blood banks.

These problems may cause adverse effects on patient care, additional work for administrators, less accountability, and wastage of important blood units. In addition, it makes reporting difficult and also tracking the movement of blood units from donation to issuing of the units. This means that there is a need to develop a web-based management system for the blood bank that will be able to handle various aspects of the blood bank.

1.3 Aim and Objectives of the Study

The research titled “Design and Implementation of a Web Based Blood Bank

Management System" will work towards developing a web-based system for the management of blood bank functions, inventory management and hospital requests for blood.
The objectives include:

1. Develop a centralized database for donors, hospitals, donations, blood stock, users, and requests.

2. Create an authentication process for administrators, blood bank personnel, and hospital users.

3. Develop modules for donor registration, donation registration, automatic generation of stock, and checking inventory status.

4. Develop process for hospital request submission, approval of request, rejecting request, reserving the request, and fulfilling the request.

5. Create dashboards and reports for stock levels, low stock groups, expiring stocks, donations, and requests.

1.4 Significance of the Study

The significance of the research lies in the fact that the research provides a viable way of enhancing the management of blood banks in various hospitals. A web based blood bank management system makes it easy for blood bank employees to document blood donations, donor details, check blood availability, and even check for expired or low stock units. This helps in overcoming the drawbacks that are associated with manual documentation.

The research is also significant to hospitals as it provides an organized digital way of making blood requests, approving them, reserving and fulfilling the blood

request. The hospitals no longer have to make phone calls or use only paper work and scattered files in their processes. The process makes it easy for both hospitals and blood banks to keep track of blood requests.

The significance of the research also lies in the fact that it makes it easy for the administrator to supervise the activities of blood banks by using dashboard summary reports and proper record keeping. It becomes easy for the administrator to be able to see donor records, donations history, blood availability, requests status and other operational information.

The patients will also gain indirect benefits through this research since the system enables quicker and more accurate access to safe blood. With enhanced monitoring of the inventory, requests, and record keeping, the system will assist in reducing delays and wastage and also ensure safe transfusions.

Lastly, this research will show how web technologies in the modern era like Django REST framework, React, Axios, React router, and databases can be utilized in resolving healthcare logistics challenges. The research will be a good example for hospitals who would like to shift from manual management of their blood banks to an organized and computerized system.

1.5 Scope of the Study

The scope of this research encompasses the design and development of a web-based blood bank management system. The system is created in order to assist in handling the activities that take place in the process of managing blood bank processes such as user registration, donor management, hospital management, donation management, blood inventory management, blood request management, dashboard summary, and report creation.

The developed system enables registered users to register and manage donors, manage blood donations, update available blood stock, manage blood inventory, and manage requests made by the hospitals. Dashboard summaries enable registered users to access vital information concerning the total number of donors, available blood stock, outstanding requests, approved requests, among other records.

The backend of the system is developed through Django and Django REST Framework, and they offer application logic, database connectivity, authentication process, and API endpoints respectively. The frontend of the system is developed through React, Vite, Axios, React Router, and Bootstrap-styled to deliver an easy-to-use interface. SQLite has been used to develop the application for local development purposes. However, the application can be deployed using PostgreSQL or any other production database in actual hospitals.

The scope of this research is limited to the management of blood bank records and the processes involved in processing hospital requests. Other limitations include development of mobile applications, interaction with laboratory machines, online payments, integrating with national blood bank database, and advanced clinical decision making in relation to transfusion. This project is developed to address problems associated with donor records, donation, inventory of blood, processing hospital requests, dashboarding, and report generating.

1.6 Limitation of the Study

The software is a minimal functional viable product and was designed to be deployed locally/institutionally. It does not currently have features such as SMS/email notifications, bar code scanning, payment processing capabilities, more complex blood matching than choosing the correct blood group, test results from laboratory, and real-time synchronizing across multiple branches. The database being used is currently SQLite which works well in development but needs to be changed to a production-level database for larger-scale deployment. Screenshots and proof of live deployment need to be included after testing in the actual environment.

1.7 Operational Definition of Terms

- **Blood Bank**: An institution where blood is collected, stored, tested, managed and

distributed for medical purpose.

- **Donor:** An individual who provides blood for storage and future transfusion.
-
- **Donation:** Blood collection information related to Donor, Staff user, Bag number, Collection Date and Volume.

- **Inventory Unit:** A Blood unit created from donation which is identified by Blood Group, Code, Status, Expiry Date

- **Hospital User:** A user profile associated with a hospital and allowed to make blood requests.

- **Web-based System:** The application which can be accessed via web browser either on local area network or internet without need of installation on each client device.
-
- **DBMS (Database Management System):** The software that facilitates users in creating , reading, updating And deleting database records

- **JWT: J**SON Web Token, a method of token based authentication in our system.

- **API:** Application Programming Interface, an endpoint that enables frontend to talk to backend.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter provides a detailed literature review related to the research area. The main objective of conducting literature review is to give an understanding about the past researches related to the chosen topic along with their weaknesses and strengths. This will also help in justifying the importance of the current research on the basis of what has been done before.

A thorough analysis of scholarly articles, research papers, and previous reports of projects has helped in identifying the weaknesses of existing solutions as well as knowledge gaps in the same area. It is only after setting a platform for the research that this research can take place. A good literature review ensures that the study is not just a mere duplication of the previous research efforts, but it is really adding something to the existing body of knowledge.

2.2 Theoretical Background

This section deals with the main theories, principles, and technical concepts behind the research. This chapter is essential in helping to create an understanding of how the methods and models applied in your research work.

2.2.1 Blood Banking and Transfusion Medicine

Blood banking is the overall process whereby blood is procured from voluntary donors, tested for infection, fractionated into different types of cells or plasma, stored, and issued out to patients who require transfusions. Blood banking is a specialty of transfusion medicine that combines laboratory science, clinical medicine, and logistics within an organization, where quality assurance and safety standards must be adhered to during each phase of the blood supply chain process. A safe blood supply is that where blood is collected from voluntary non-remunerated donors, screened for

transfusion-transmissible infections, fractionated into components when necessary, and transfused only when required (World Health Organization, 2022). Adherence to these guidelines in a resource-limited hospital environment requires effective information systems for proper tracking of the different aspects of the blood bank.

The blood supply chain process in the hospital blood bank is made up of five interrelated functions: donor management, laboratory processing, inventory management, request processing, and transfusion management. Donor management involves donor mobilization, registration, screening of donors' histories, deferral or approval. Laboratory processing includes blood grouping, screening for infections, fractionation, and labeling. Monitoring the availability of the units on hand based on the group, component type, and expiry date is necessary in inventory management with alerting mechanisms to indicate low quantity and upcoming expiry dates. The blood requests include receiving, clinical approval, compatibility tests, allocation of units, and issuing the allocated units to the clinical personnel requesting them. In transfusion monitoring, the event of transfusions needs to be recorded, as well as identification of the recipients, indication of the transfusion, and any reactions following the transfusion. Manually managing these five stages in a busy hospital blood bank will create numerous areas for transcription errors, lost records, and process delays.

2.2.2 Limitations of Manual Blood Bank Management

Blood bank management via paper-based methods, which continues to dominate operations in many hospitals in Nigeria, faces a series of limitations inherent in the system and compromising the quality of transfusion services. First, donor registration through manual keeping of donor data in different ledgers results in the automatic fragmentation of the database, where donors may end up being entered twice due to small differences in the names of the donors, the deferred status of donors during their previous visits is not always checked, and the long record of the donors that is required for assessing their eligibility is practically non-existent (Ben Elmir et al., 2023). The manual keeping of the stock in the refrigeration unit, based on visual inspection and

stock registers, prevents the generation of timely reports regarding the number of units available in various blood groups, especially during the night shifts or times of increased demand.

Expiry management poses one of the most critical challenges in manual systems. Blood products have specific expiration dates, where the life of the packed red blood cells lasts between 35 to 42 days, depending on the preservative solution used — Lack of automated expiry monitoring also ensures that units close to their expiry date are often neglected after expiry, hence causing wastage in a setting that is already plagued by lack of sufficient blood supply (Raturi & Shastry, 2022). Paper cross-matching logs are prone to being misplaced and take time to be retrieved when there is need for a quick transfusion request. Lack of formalized documentation on transfusions hampers hemovigilance and post-transfusion outcomes analysis.

The implications of such limitations in terms of operations include more than just inefficiencies. Lack of proper pre-transfusion compatibility records, delays in blood supply in emergency situations, as well as use of blood products with unclear eligibility status are all patient safety risks that have been proven to cause morbidity and mortality associated with blood transfusions in West African hospitals (Aneke & Okocha, 2017). Thus, the justification for the need to introduce a structured automated system of blood bank management online is not just in increasing efficiencies but in ensuring patient safety.

2.2.3 System Security and Role-centered Access Control

Security should be a primary concern when designing any health care information system dealing with highly confidential information related to patients and donors. With regards to a blood bank management system, security considerations include data confidentiality, which involves ensuring donor medical and patient transfusion history data are accessible only by authorised individuals; and data accuracy, which involves stopping any modifications of the blood units' records, compatibility tests' results, and transfusions records, which would affect patient safety. There are three fundamental security measures, namely authentication, authorisation,

and auditing, through which these considerations are met (Kruse et al. 2017).

Rescent-based access control (RBAC) is the most common type of access control mechanism used in healthcare information systems, where permissions are granted to roles and not individual accounts, and where users are assigned to those roles. In the case of a blood bank information system, different roles may be created for blood bank administrators (access to all features of the system), laboratory technicians (registration of blood units, compatibility testing, and issue), clinical users (submission of blood requests and inquiry about their status), and management users (reporting and auditing access) (Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996)). Passwords should be stored using salted hash encryption and not in clear text form, while session management should automatically terminate sessions on expiration to avoid unauthorized access to unattended computers. Audit log should document all changes to the data together with the identity of the user who performed those changes, timestamp of changes, and nature of the changes to meet regulatory requirements and investigate possible transfusion-related problems. running the application as well as the database.

The Web delivery model for a health information system offers several benefits that are especially important in the context of a blood bank's information system. Firstly, the centralized data storage guarantees that the same consistent database is used by all of the relevant users, regardless of whether they are laboratory specialists, clinicians ordering blood units, or administrators creating reports, without data duplication and synchronization problems associated with file-based systems with dispersed data storage. Secondly, specific access control allows the limitation of actions of each particular user according to his or her clinical role, thus preventing the manipulation of sensitive information without proper transparency in other sections (Fernández-Alemán et al., 2013). Thirdly, real-time access to live data provides an opportunity to see the same information regarding inventories, eligibility of donors, or pending orders at the same time for all relevant users, facilitating a prompt reaction to the situation.

2.2.4 REST API and Single Page Application Architecture

The software utilizes a RESTful API backend along with a React single page application frontend. The backend provides endpoints that facilitate interactions with the data, whereas the frontend accesses the data through these endpoints using Axios. The benefit is maintainability since the interface and business logics are separated.

2.3 Review of Related Works

Previous works related to internet-based blood banks, hospitals' blood inventory systems, internet-based donor management systems, and blood stock decision support systems indicate that computer-based systems may lead to less errors and time consumption due to blood bank operations carried out manually. Previous researches reveal that besides the list of donors, the hospitals need to have other features such as inventory records, quick handling of blood requests, security access control, traceability, reporting, and decision support. The related works discussed in this section include articles written between 2020 and 2026.

In their study, Deshpande, Kumar and Chaple (2020) focused on the development of web-based blood bank management system with an objective of enhancing the availability of information about blood stocks, blood issue and blood group availability. The methodology used by the authors for their research was a web-based database whereby authorized blood bank employees were able to update blood availability, donor information and issues using a centralized system. From the study, it was noted that the use of a web-based platform enhances accessibility and updating of blood information more than paper based documentation system. Their findings indicated that the web-based blood bank system improves efficiency of accessing the required blood group and response capability of the blood bank employees to the requests. However, the study had some limitation due to the lack of emphasis on implementation testings, security controls, real time request handling from the hospital, etc.

The methodology of Lin, Metcalf, Wilburn and Lex (2021) used to create Sanguine was an iterative design study involving collaboration of patient blood management specialists, surgeons and anesthesiologists. In their research, the authors paid attention to developing visual analytics for hospital stakeholders to enable

analyzing transfusion practices, comparing blood usage results and exploring patient blood management data. Thus, the authors showed that visual analytics could be a means to improve hospital decision-making processes regarding blood use and transfusion practices. The results have shown that there is a need for hospitals to get interactive tools that would allow clinicians to explore blood utilization trends instead of relying on static reports only. The limitation of the study is the lack of focus on donor registration, blood inventory generation, blood requests analysis and hospital blood bank processes automation.

In their methodology, Rad et al. (2021) developed a real-time web-based dashboard for monitoring the inventory of blood products. This methodology was based on the use of visual analytics and transactional blood unit data in order to provide the most up-to-date information about blood products. This system was useful to allow blood transfusion services to monitor the inventory of blood products and the blood unit transaction process in a more efficient way. This method resulted in near-real-time dashboard monitoring and made it possible to increase the availability of blood products as well as assist people in making more informed decisions. It was found out that such monitoring systems may help decrease waste and increase the availability of blood products due to easy interpretation of stock information. However, this system is dependent on digital transaction data.

Kuberkar and Singhal (2021) conducted research on the integration of blockchain and Internet of Things technologies for sustainable blood bank management. They used a methodology that involved both the Task-Technology Fit framework and Technology Acceptance Model in order to investigate factors that would affect the intention to adopt the aforementioned technologies in blood banks. This led to the conclusion that adoption of these technologies will facilitate traceability, privacy, transparency, and coordination of the process of blood supply. In terms of limitations, Kuberkar and Singhal (2021) found out that technology adoption in blood banks is subject to security, governmental support, users' acceptance and usefulness of the technology.

Doneda et al. (2021) created a discrete-event simulation model for investigating

the operation in a blood donation center. The methodology employed in their study involved simulating modeling based on real-life operations of the blood collection center so as to examine various configurations of resources and schedules. The output indicated that the various operation scenarios could be evaluated before implementation so as to enhance scheduling and usage of resources. Their results indicated that simulation could help in making better decisions in blood collection centers by demonstrating the potential impact of change in operations. The weakness in their research is that it was limited to simulation of the blood collection center and not an integrated web-based hospital blood bank management system.

Azhar & Ali (2022) designed a Smart Blood Bank Management System to assist in donor and recipient data management. The authors used the Waterfall methodology which involves requirement analysis, design, implementation, testing, and maintenance. The design process involved the use of Visual Studio Code and phpMyAdmin software. The system was made up of the login module, donor management module, blood donation module, recipient management module, handover and user module. The findings indicated that the system can better organize records compared to manual and paper-based records. The main finding by the authors is that having a centralized database improves record management, reduces duplicate records and makes data from blood banks accessible. The weakness in their work is that the system did not have additional features like AI forecasting, emergencies notification, multiple hospitals integration, dashboard reporting, etc.

Bhandawat et al. (2022) introduced a decentralized ledger system called the Cooperative Blood Inventory Ledger, which was designed to enhance blood product management. The methods that Bhandawat et al. (2022) adopted in their study were blockchain, smart contracts and a proof-of-concept approach involving Hyperledger Fabric. The main objective of the method was to facilitate inter-hospital redistribution of blood products. The outcome of the process was the successful design of an effective platform for controlled sharing of information about blood products while reducing losses caused by lack of coordination. The limitation is that blockchain-based systems may be costly and hard to implement due to their complex nature.

The authors of this article are Shah, Shah, and Jan (2023), who created an Online Blood Donation Management System for the donors, seekers, blood banks, and blood groups. This methodology involves the use of a client-server web application for the registration of users as donors, seekers of blood and management of the information on blood banks and blood seekers. It was established through their study that an online blood donation system helps in reducing the manually heavy process of searching for donors and blood banks. Limitations of the article are that the online system requires internet access, donor participation, and regular updating of the database.

Kaur, Rai, Raj, Sharma, and Sirohi (2023) proposed a Blood Bank Management System with Emergency Blood Locator. Methodology followed by them was based on cross platform design including GPS, registration forms, API and database storage. It facilitated the user to find nearby donors, hospitals and blood banks in emergency conditions. The outcome was that location-based systems are able to reduce time taken to search for blood in emergency situations. The finding was that GPS-based blood location systems are effective in places where fast location of the donor or the blood banks is necessary. The limitation is that the system relies on the accuracy of GPS, data updates, donor availability and network connectivity..

According to Sharma et al., (2023), Donor Dreams is a centralized web-based approach where the web platform facilitates interactions between blood banks and patients. In their methodology, the authors have applied a web application consisting of Admin, Blood Bank and Patients modules. The system was able to help blood banks manage the blood stock while patients can request particular blood groups. Consequently, the findings indicate that such an approach is capable of reducing fragmentation between patients and blood banks through a centralized approach in terms of communication and request management. The findings showed that centralized blood donation systems enhance the process of requests' management while making it easier to find blood..

Farrington, Alimam, Utley, Li, and Wong (2024) have proposed a machine learning -based approach for platelet issuing to minimize wastage at the blood bank of hospitals. The methodology adopted for this study included training the machine learning

algorithm based on the platelets' request data set and issuing the policy using simulation. This study has provided an AUROC of 0.74 along with a decrease of 14 percent in wastage in terms of simulation. They found out that predictive issuing policies are very useful in minimizing wastage of short-life blood products and effective decision-making in blood banks of hospitals. However, the limitation of their study is that they have worked on platelets only..

In 2025, Rokaya, Subedi, and Dhungana created Rakta Setu which is a web-based blood bank software for the purpose of managing blood stock, blood request, donation, donor and patients. The methodology used for developing this platform included designing a multiple role web application for blood banks, donors and patients. With this system, blood banks were able to update their inventories as well as manage blood and donation requests. The finding obtained from this study is that web applications can be used instead of manual recording of blood data. They discovered that the use of a multiple role blood platform can increase access, blood requests and records management..

Zadeh et al., Zandieh and Tabriz (2025), suggested a cloud-based information system for blood supply chain management for real-time information exchange. The method adopted by adeh et al., Zandieh and Tabriz involved system dynamics modeling using causal-loop model and stock-flow model to analyze the impact of cloud-based information exchange on performance of blood supply chain. They found out that there was an improvement in wastage rate, fill rate and inventory levels. They observed that real-time information exchange has the potential to bring more balance in blood supply and demand relationship and can help coordinate among the participants of blood supply chain. The limitation of the paper is that the study is simulation-based.

A system for emergency blood donation and demand prediction using artificial intelligence was developed by Rithya Nandhini, Vigneshkumar and Ramakrishnan in 2026. The methodology applied in this study employed the Django web framework, role-based access control, donor ranking, notifications and LSTM demand prediction. The design of the system aimed at automating hospital blood requests, donor matching and predicting shortages. The outcome of this research indicated that the system could

help in emergency situations through donor matching and shortage prediction. It was noted from their results that AI can aid in inventory management and dealing with emergency requests for blood. However, the main limitation of this study is that the system needs high-quality historical data, donor location data and donor participation.

Based on the analyzed works above, it is clear that web-based and intelligent blood bank systems have a number of advantages including quicker access to blood data, enhanced monitoring of blood stocks, effective donor management, wastage reduction, request processing, and improved decision-making ability. Nevertheless, the vast majority of analyzed works concentrate their efforts on just one aspect of the blood management cycle, which includes donor search, dashboard analysis, blockchain distribution, or predictions. The current project can be seen as an implementation of a practical web-based hospital blood bank management system that will encompass all aspects of blood management including donor registration, blood donation, automatic blood units generation, hospital request, staff approval, fulfillment of requests, dashboard, and role-based access contro.

2.4 Summary of Literature Review

This section summarizes the key findings from the literature. Presented in tabular format

Author(s) and Year	Work Done	Methodology	Results	Findings	Limitation
Deshpande, Kumar and Chaple (2020)	Developed/reviewed a web-based blood bank management system for bloodstock, blood issue details and blood group	Web-based database approach with authorized blood bank updates.	Improved access to bloodstock and issue records.	Digital blood bank systems reduce delay in locating required blood groups.	Limited detail on implementation testing, security and real-time hospital request

	access.				processing.
Lin, Metcalf, Wilburn and Lex (2021)	Developed Sanguine, a visual analytics tool for hospital patient blood management.	Iterative design study with hospital patient blood management experts and clinical stakeholders.	Enabled analysis of transfusion practices and outcomes.	Visual analytics supports better blood-use decisions in hospitals.	Focused on transfusion analytics, not donor, inventory and request automation.
Rad et al. (2021)	Designed a real-time web-based dashboard for blood product inventory monitoring.	Interactive visual analytics using blood unit transaction data.	Supported near-real-time monitoring of blood product inventory.	Dashboards can reduce wastage and improve inventory visibility.	Requires reliable digital transaction data from blood banks.
Kuberkar and Singhal (2021)	Studied blockchain and IoT adoption for sustainable blood bank management.	Integrated Task-Technology Fit and Technology Acceptance Model.	Identified factors affecting adoption intention.	Blockchain and IoT can improve transparency , traceability and privacy.	Adoption depends on regulation, cost, infrastructure and user readiness.
Doneda, Yalcindag, Marques and Lanzarone (2021)	Built a decision-support model for blood donation center operations.	Discrete-event simulation using real blood	Helped compare configurations and improve scheduling.	Simulation supports better resource planning.	Focused on collection centres, not full hospital request

		collection center data.			management.
Azhar and Ali (2022)	Developed a Smart Blood Bank Management System for donor and recipient records.	Waterfall model using Visual Studio Code and phpMyAdmin.	Produced login, donor, donation, recipient, handover and user modules.	Centralized databases improve record handling over paper systems.	Limited advanced features such as AI forecasting, notifications and multi-hospital integration.
Bhandawat, Casucci, Ramamurthy and Walteros (2022)	Proposed the Cooperative Blood Inventory Ledger for decentralized blood inventory sharing.	Blockchain, smart contracts and Hyperledger Fabric proof of concept.	Supported inter-hospital product redistribution and reduced losses.	Shared ledgers improve cooperation while preserving access control.	Blockchain deployment may be costly and technically complex.
Shah, Shah and Jan (2023)	Developed an Online Blood Donation Management System for donors, seekers, blood banks and blood groups.	Distributed client-server web application.	Enabled donor registration, blood request and admin record management.	Online systems reduce manual searching and improve data availability.	Requires internet access and active participation from donors/users.
Kaur, Rai, Raj,	Designed a Blood	Cross-	Helped users	Location-	Depends on

Sharma and Sirohi (2023)	Bank Management System with Emergency Blood Locator.	platform mobile/web approach using GPS, registration forms, APIs and database storage.	locate nearby donors, hospitals and blood banks.	aware systems reduce emergency blood search time.	GPS accuracy, donor availability and regular database updates.
Sharma et al. (2023)	Proposed Donor Dreams, a centralized platform connecting blood banks and patients.	Web application with Admin, Blood Bank and Patient modules.	Supported blood request history, stock tracking and request approval/rejection.	Centralized platforms reduce fragmentation between patients and blood banks.	Success depends on blood bank participation, awareness and data security.
Farrington, Alimam, Utley, Li and Wong (2024)	Proposed machine learning support for platelet issuing and waste reduction in hospital blood banks.	Machine learning prediction model and simulation using platelet request data.	Achieved AUROC of 0.74 and estimated 14 percent wastage reduction.	Predictive issuing can reduce wastage of short-life blood products.	Focused on platelet returns, not full blood bank management workflow.
Rokaya, Subedi and Dhungana (2025)	Developed Rakta Setu, a web-based blood bank platform for inventory, requests, donations, donors and patients.	Multi-role web application for blood banks, donors and patients.	Allowed blood banks to update inventory and manage blood/donation requests.	Multi-role platforms can replace traditional record keeping.	Large-scale hospital testing was not clearly established.

Zadeh, Zandieh and Tabriz (2025)	Proposed cloud-based blood supply chain management for real-time information sharing.	System dynamics simulation with causal-loop and stock-flow models.	Improved wastage rate, fill rate and inventory level indicators.	Real-time information sharing improves supply-demand balance.	Simulation-based; deployment may face infrastructure and privacy challenges.
Nithya Nandhini, Vigneshkumar and Ramakrishnan (2026)	Designed an AI-based emergency blood donor matching and demand prediction system.	Django web framework, role-based access control, donor scoring, notifications and LSTM demand forecasting.	Automated hospital blood requests, donor matching and shortage alerts.	AI can improve emergency response and proactive inventory management.	Requires quality historical data, reliable donor location data and strong privacy controls.

2.5 Research Gap

From the reviewed literature, it is evident that digital tools have the potential to enhance blood bank functions; however, most of them solve partial problems of the process of blood management at hospitals. For example, there are tools that help register donors and search blood but lack automated conversion of data from donations to inventory units. There are tools with dashboards but without management of the operational process of hospital requests approval and execution. Finally, some authors suggest modern technologies like blockchain and cloud simulation with machine learning; however, such technologies are too expensive..

Gap in Research that Is filled by this Study. This research gap is based on the requirement for a practical, integrated, role-based and web-based blood bank management system that will have the capability of handling donor management,

donation management, blood inventory, hospital blood requests, approval of these requests, fulfillment of the requests and generation of reports. The developed system addresses this research gap through the establishment of an automatic mapping of all donations to a blood inventory unit, role based access control, ability for hospitals to submit requests for blood, possibility of approving or rejecting submitted blood requests and changing status of inventory units upon reservation/issue of these units..

CHAPTER THREE
SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This chapter describes the analysis and design of the proposed blood bank management system. The existing system, the constraints with the system, the proposed system, requirements, design approach, UML models, database design, and architecture are discussed in this chapter.

3.1 System Analysis

3.1.1 Analysis of the Existing System

It can be assumed that the existing process was either done manually or partially with the help of computers. The details of the donors can be documented in files or spreadsheets. The records of donation can be kept separately from the records of blood inventory. Hospitals can place their demands for blood via phone calls, written requests, or in person visits.

3.1.2 Limitations of the Existing System

The limitations include delays in accessing blood availability information, possible duplication or incompleteness in recording donor information, poor tracking of blood unit information, inability to determine low stock units, absence of structured request approval, and poor reporting system. The manual system increases the chances of making mistakes in keeping track of the inventory after donation and issuance.

3.1.3 Analysis of the Proposed System

This system will be a web-based system which will be used to handle all activities of the blood bank. With this system, registered users will have access to functions such as donor registration, donation registration, and view of inventory. With each donation entry in the system, the backend will generate an inventory of blood and its expiry date. Hospital users will make blood requests which will then be approved, rejected, or filled by other users.

3.2 System Requirements Specification

3.2.1 Functional Requirements

Requirement	Description
User authentication	The system will authenticate the user through username/password authentication and JWT tokens.
Role management	The system will have admin, staff, and hospital roles.
Donor management	The system will enable the creation of donor profiles by the staff/admin users.
Donation recording	The system shall record date of donation, bag ID, name of donor, volume, employee collector and notes about the process.
Creating blood inventory	The system shall create unit of available blood inventory after successful registration of donation.
Inventory monitoring	The system shall display units of inventory and filter them according to blood group and status.

Hospital requests	The system shall allow users of hospitals to make blo
Workflow for Request	The system shall allow staff/admin users to approve, deny, reserve, fulfill or view requests.
Reports	The system shall give dashboard, inventory, donation, and request reports.

3.2.2 Non-Functional Requirements

Requirement	Description
Security	The system shall use JWT authentication and permissions based on roles to secure operations.
Usability	The interface shall have clear pages of dashboard, donors, donations, inventory, requests.
Reliability	The operations of donations and fulfillment shall be done using database transactions.
Maintainability	The system shall divide its modules into accounts, donors, donations, inventory, hospitals, requests, reports and services.
Scalability	The architecture shall allow the migration of database from SQLite to PostgreSQL or other database used in production.
Performance	The system shall use filtered queries and indexing of inventory fields in case of frequent lookups.

3.3 System Design Methodology

The method that was used is Object Oriented Analysis and Design and its

incremental approach to development. The back end uses Django models as domain classes and Django REST framework serializers and views for API layer. Front end uses reusable React components and page modules.

3.3.1 Justification for Methodology

OOPD(Object Oriented Analysis and design) is appropriate since there are identifiable entities in the system, such as the User, Hospital, Donor, Donation, Blood Inventory, Blood Request, and Request Item. Incremental development is also appropriate since modules can be developed and tested sequentially, beginning with authentication, followed by the donors, donations, inventory, requests, and reports..

3.3.2 Method of Data Collection

Data used to build the model were collected through observing the processes used in blood banks, the requirements of the project from the source code, and entities that were needed in managing donors, donations, stocks, and requests from hospitals. The demo data in the project comprises sample hospitals, users, donors, and inventories which were generated by the seed_demo.py Django management command.

3.4 System Modeling Using UML

UML diagrams may be used to provide one of the bases in software engineering that is the standard visual language that can represent all different aspects of a software system in detail. It can serve as an excellent tool for architects, developers, stakeholders, and other people to be able to design and especially communicate all different parts that a software system consists of. One of such diagrams that is considered to be extremely important is the use case diagram that describes all possible interactions of an external system or users with the system itself and that could be considered to be the design of all different processes within a software system application.

The next diagram, activity diagram, complements the use case diagrams

because they enable the aspects of dynamic process modeling in a system development process and will show the total flow of all activities that take place within the software system and the process sequence/parallel behavior. The dynamic view will contribute greatly to visualization of the overall process taking place within the system and will help the developers and the stakeholders to identify possible bottlenecks, decision making, and improvement of a software system.

3.4.1 Use Case Diagram

The use case diagram identifies the key actors that interact with the Blood Bank Management System. The Admin actor logs in, accesses the dashboard, manages the users, views the donor record, manages the blood stock, and manages blood requests. The Blood Bank Staff actor undertakes routine operations such as registering the donors, recording donations, managing the stock, approving, rejecting and fulfilling the request. The Hospital actor logs in and makes blood request and views the request made record. Fulfilling a blood request involves issuing a blood unit, while recording a donation updates the blood stock..

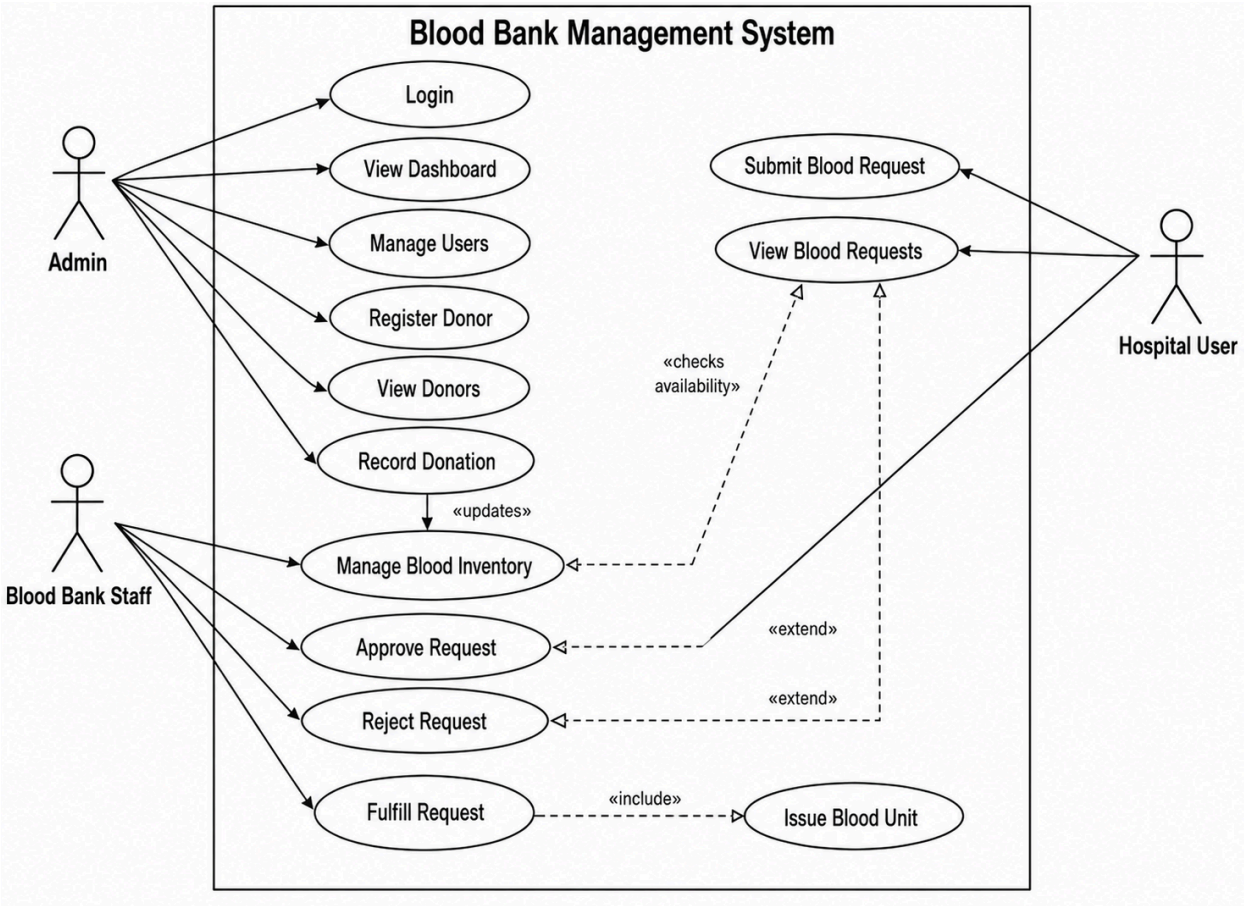

Figure 3.2:use case Diagram of the proposed system

3.4.2 Activity Diagram

Donation activity flow: User logs in, opens donation record creation page, selects donor, enters bag ID, volume, date, and comments, submits the form, backend validates donor and bag ID, creates donation record, creates blood inventory record, updates last

donation date of donor, and returns the created record.

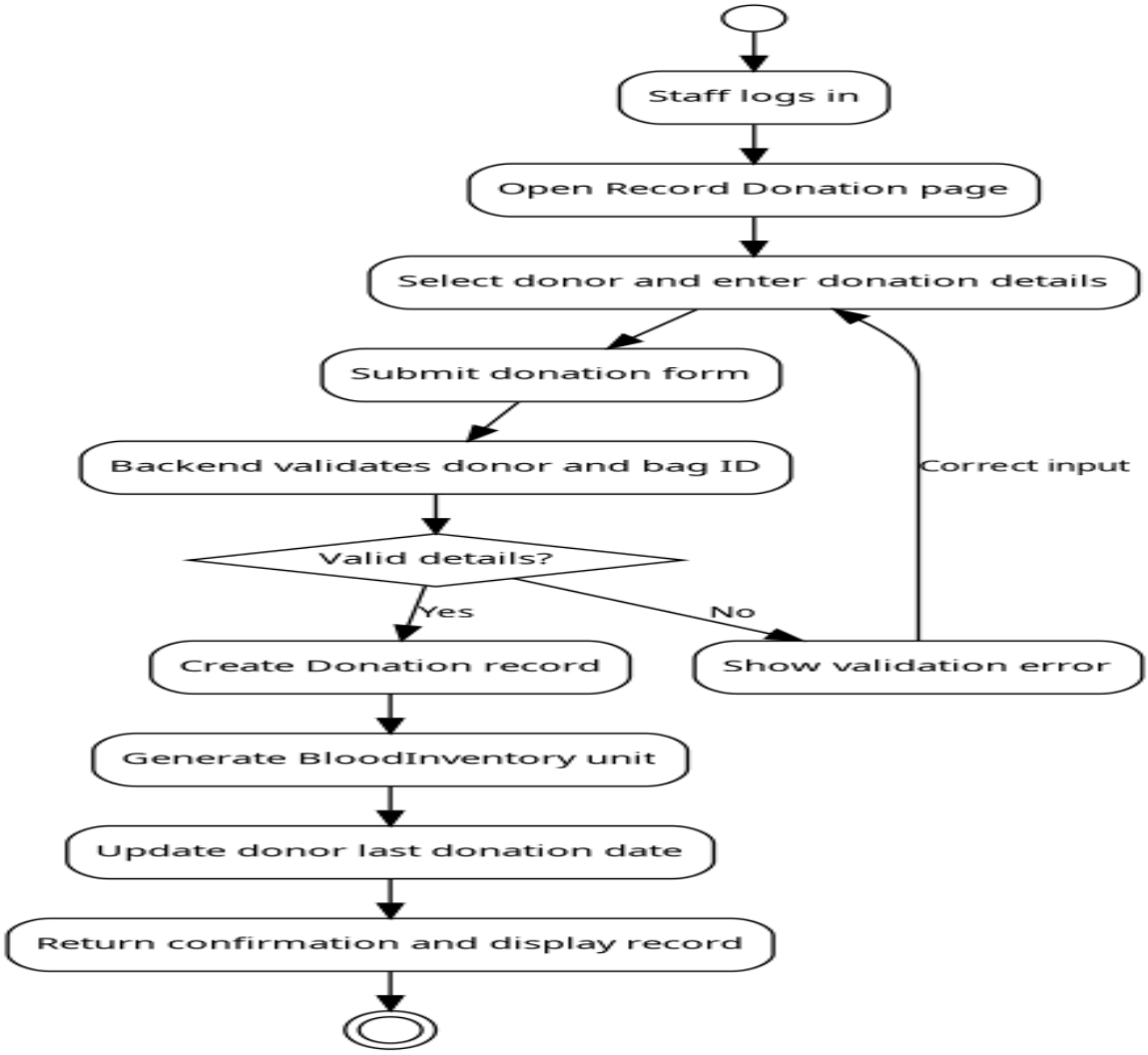

Figure 3.2: Activity Diagram for Donation Recording

Request activity flow: HOSPITAL USER logs in, enters his blood group, number of units required, priority, and notes. His request becomes pending. The pending requests are then reviewed by the staff. If there are enough units in stock, the request is approved, and the units are reserved. Staff fulfills the request using the inventory IDs selected, after which the units are issued.

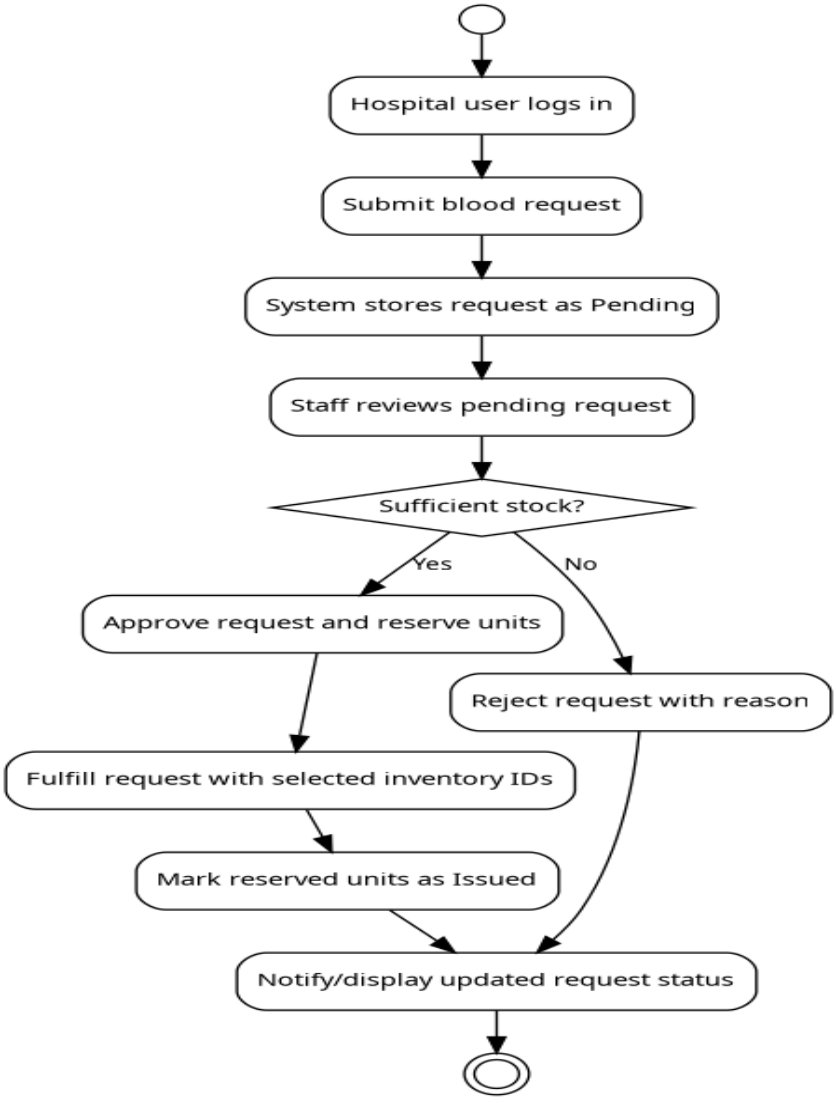

Figure 3.3: Activity Diagram for Blood Request Processing

3.4.3 Class Diagram

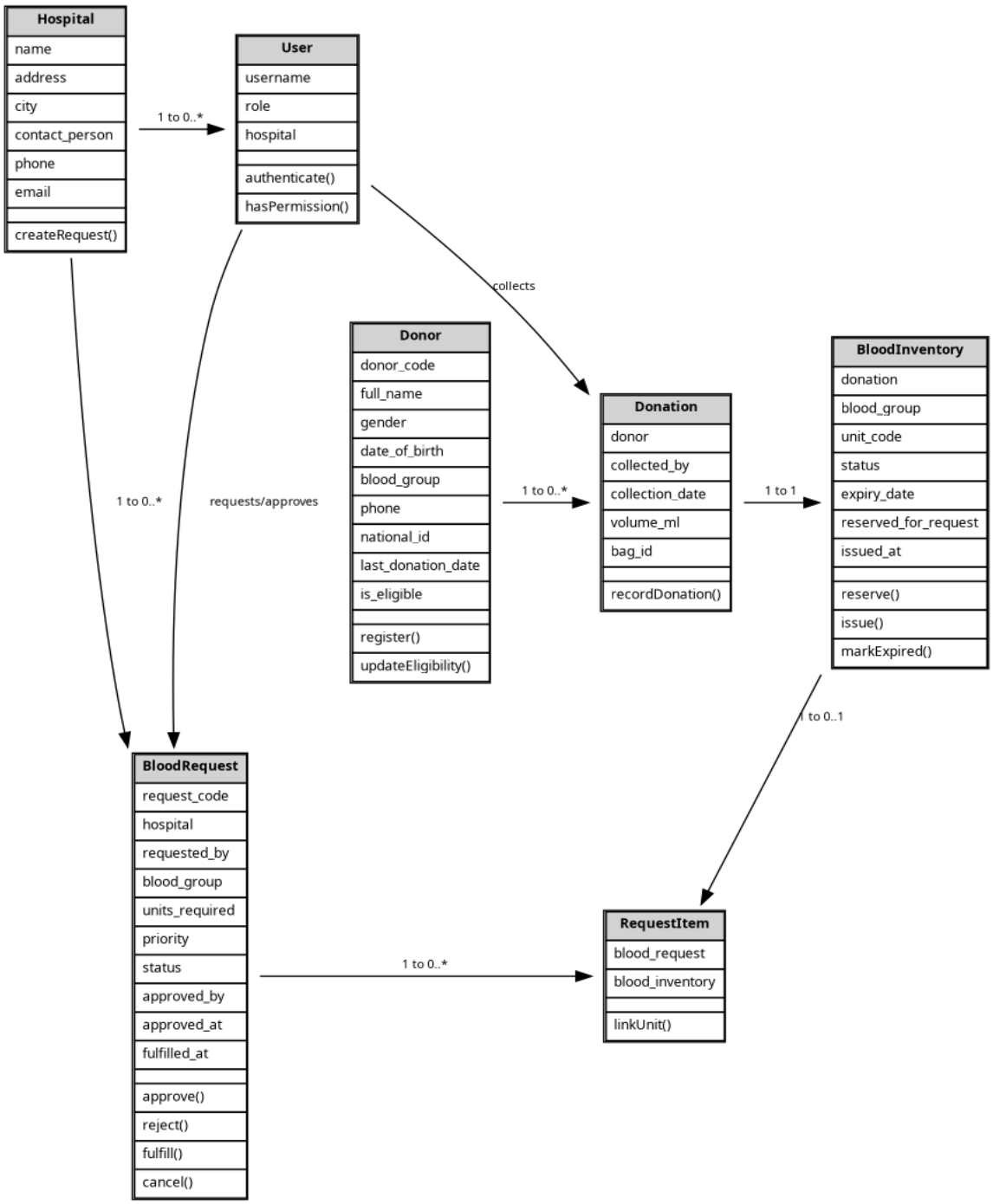

Figure 3.4: Class Diagram of the Proposed System

3.4.4 Sequence Diagram

This sequence diagram represents the interaction based on time between the user in the hospital, React front-end, Django REST API back-end, SQLite database, and staff/admin in relation to the processing of the request for blood from the initiation of the request to its fulfillment. The sequence diagram illustrates how the request information is passed through the user interface to the API and stored in the backend.

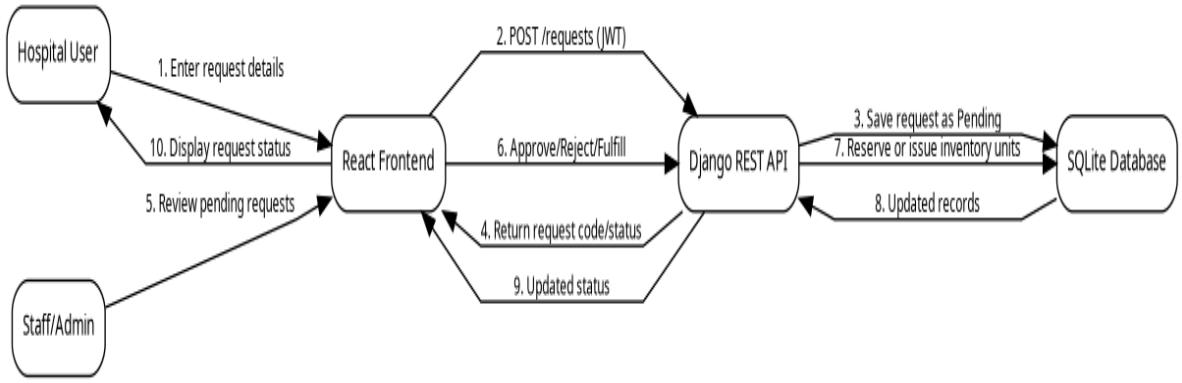

Figure 3.5: Sequence Diagram for Blood Request Processing

3.4.5 State Diagram

The State Diagram shows the life cycle of the blood request in the system. The new request starts as a pending one. Then it can be approved, rejected, canceled, or fulfilled depending on the staff and blood stock in the bank. This diagram can be helpful as the blood request process is one of the most active parts of the system.

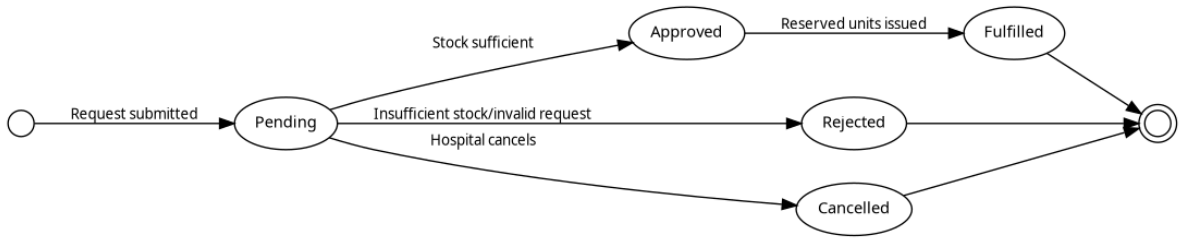

Figure 3.6: State Diagram for Blood Request Status

3.5 System Design

3.3.1 Input Interface Designs of the Proposed System

This part shows how data can be sent to the system through various input interfaces. Input interface design is an essential aspect that helps users to interact with the system. In the proposed blood bank system, the following are examples of input interfaces that will be included: login form, donor registration form, donation form, hospital requests form, inventory filter fields, and request action forms.

In the login input interface, the user enters his username and password. When entered, the login information is forwarded to the backend authentication endpoint of the system. If the login information is valid, then the backend returns access and refresh tokens. The access token gives the user permission to access all system protected pages. Donor registration input interface allows blood bank staff to input donor details such as full name, gender, birthday, blood type, phone number, national ID, address, and donor eligibility.

The donor input form enables personnel to choose the donor and input donation details like donation date, quantity of blood, bag ID, and remarks. After the data is input, the system then generates the donation and automatically creates a blood stock for the particular donation. The hospital request input form enables hospital personnel to choose the blood group needed, input the number of units needed, specify the priority of the request and input remarks. Through this form, hospitals can express their blood

requirements without having to depend only on paper and telephone communication.

3.3.2 Output Interface Design of the Proposed System

The output interface design describes how the processed information is returned and shown to users. In this particular case of the proposed blood bank system, the following output interfaces are included: the dashboard screen, donor list, donation list, inventory page, hospital request page, status badges, alert messages, and reports. These output interfaces help the user become aware of the results of system processing.

The dashboard output interface shows stock by blood group, low-stock groups, pending requests, and units of blood that expire in the next seven days. Donor output interface contains information about registered donors allowing staff to see the information about donors. Inventory output interface shows each blood unit with unit code, blood group, status, expiry date, and related donation. Request output interface contains hospital requests with status such as pending, approved, rejected, fulfilled, or cancelled. Thus, users do not have to search for information manually through paper documents.

3.5.3 Database Design

The design of a database refers to the graphical representation of how the database will be designed once created. This represents the various tables which are used for the storage of data in the database. The idea behind database design is the graphical illustration of how the proposed database looks like. The database for the proposed system is based on the use of relational database using Django Object Relational Mapper. During development, SQLite is used, but the database design can be migrated to PostgreSQL or any other relational database when it is being deployed.

The key entities of the proposed system include User, Hospital, Donor, Donation, Blood Inventory, Blood Request and Request Item. These are the real world objects that

are used in the blood bank management process. User table contains user account and role information. Hospital table contains hospital information. Donor table contains donor bio data and blood group information. Donation table contains information about blood collection. Blood Inventory table contains each blood unit generated by donation. Blood Request table contains hospital request information while the Request Item table connects the fulfilled requests to the inventory units issued.

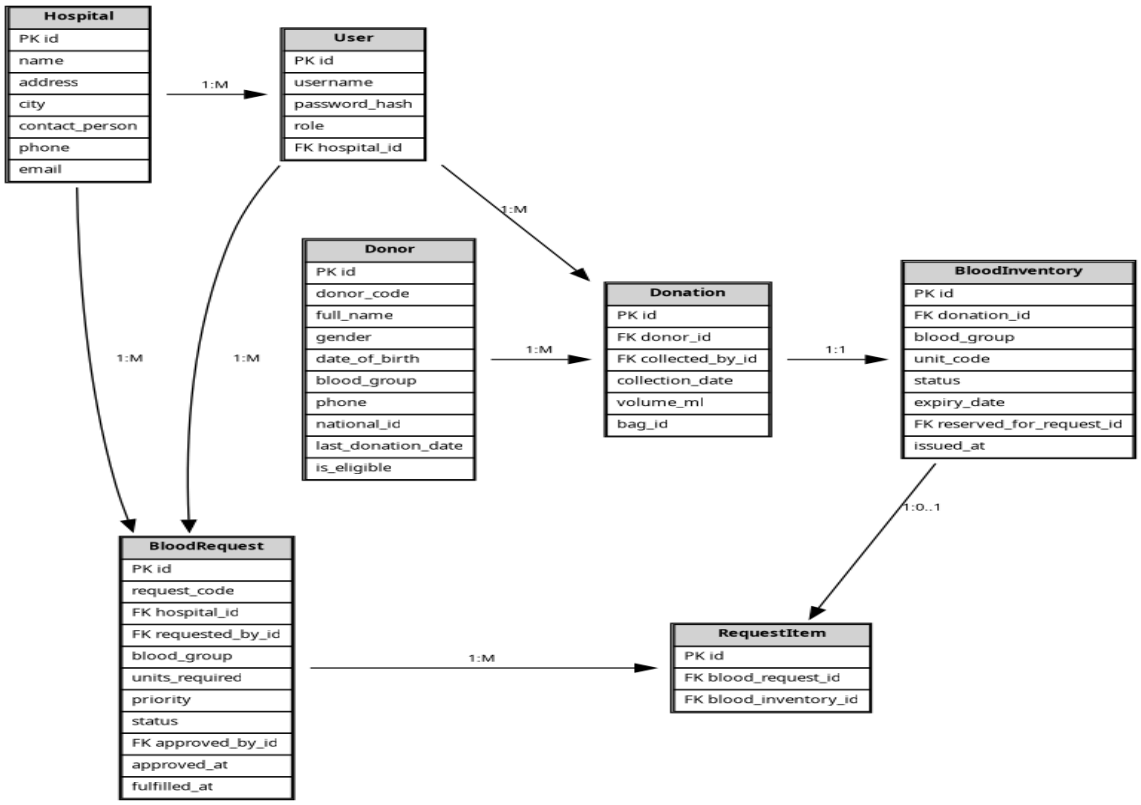

Figure 3.7: Entity Relationship Diagram of the Proposed Database

Table/Model	Major Fields	Purpose
User	username, password, role, hospital	Stores authenticated users and their access role.
Hospital	name, address, city, contact_person, phone, email	Stores hospitals that submit requests.
Donor	donor_code, full_name, gender, date_of_birth, blood_group, phone	Stores donor biodata and eligibility information.
Donation	donor, collected_by, collection_date, volume_ml, bag_id	Stores blood collection records.
Blood Inventory	donation, blood_group, unit_code, status, expiry_date	Tracks each blood unit generated from a donation.
Blood Request	request_code, hospital, requested_by, blood_group, units_required, priority, status	Stores hospital blood requests and workflow status.
Request Item	blood_request, blood_inventory	Links fulfilled requests to issued blood units.

3.5.4 System Architecture Design

This proposed system has been built to be a web application in such a way that the users, particularly the blood banks' staff, admins, and hospital users, can easily use this software via their web browsers. No complex setup of this software in the user's

computer is required except using the browser and being able to connect to the application over the network. The system utilizes a client-server architecture where the front end deals with the user interface while the back end with authentication, business logic, and API responses.

The frontend of the system is developed using React and Vite. This component renders the pages for user login, donor registration, donation entry, inventory viewing, request submission, and request management. Communication between the frontend and the backend is made through REST API calls using Axios. The backend component of the system is developed using Python, Django, and Django REST Framework. The backend receives requests from the frontend, validates the data provided, checks permission based on roles, makes database transactions, and sends JSON responses to the frontend..

The system follows a three-tier web architecture:

1.React Frontend (Vite SPA) | HTTP/JSON through Axios with Bearer Token

2. Django REST API Backend | Django ORM

3. SQLite Database for development

The separation of the front-end from the back-end makes this architecture scalable, and since the interface is not connected to the business logic, the architecture makes the code maintainable. The architecture uses JSON Web Token for security and authentication, which ensures that only authentic API calls are made between the two layers..

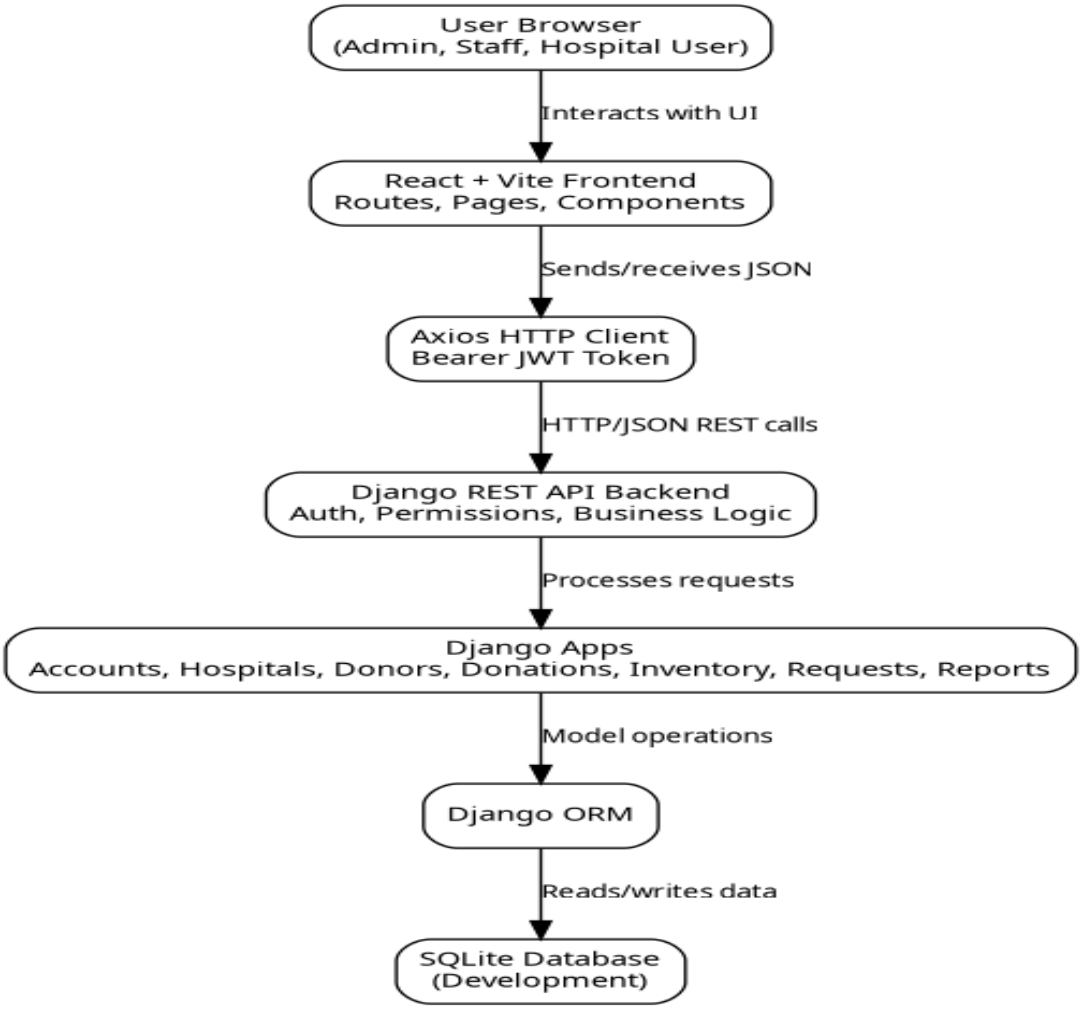

Figure 3.8: System Architecture Diagram of the Proposed System

CHAPTER FOUR
SYSTEM IMPLEMENTATION

4.0 Introduction

In this research project, the concepts of design, system development, and implementation of the suggested web-based Blood Bank Management System are discussed in this chapter. Implementation process started after the identification of the requirements of the system and the designing of the system in the previous chapter. Careful selection of the implementation tools was done because they facilitate the development of web applications, security of API communications, database management, and interface development.

Thus, this chapter discusses the technology used in the implementation of the proposed system, the requirements required to operate the system, user interfaces of the system, and the implementation of the major modules. This proposed system, unlike the artificial intelligence image classification system, does not involve the training of the machine learning model. It is a web-based information management system focusing on donor data, blood donation, inventory, hospital requests, and reporting.

4.1 Development Environment and Tools

The development environment and tools that were used in the implementation of the proposed system include the programming languages, libraries, frameworks, database system, development platform, and hardware resources that were used in the

designing and coding of the proposed system. As the proposed system is a web-based blood bank management system, the tools that have been used were selected on the basis of the fact that they provide security in backend development, designing of database applications, REST API communication, and development of interactive frontend user interface.

Python and Django were used in the backend development environment, whereas JavaScript, React, and Vite were used in the frontend development environment. SQLite was chosen as the development database as it is easy to set up and appropriate for local testing. REST API communication was also used in the application so that data can be sent from frontend to the backend and received back by the frontend in JSON format. These tools allowed implementing the donor management, donation logging, blood inventory, hospital request, authentication, and reporting modules of the system.

4.1.1 Programming Language and Libraries

Python was chosen as the primary backend programming language since it is easy to learn, readable, expressive, and highly used in building safe web applications. Moreover, the language works well with Django, which offers the required features such as database model, routing, authentication, and administration. With the help of Python, the implementation of the backend logic of the blood bank application became an easy task, such as donor registration, donation process, automatic creation of inventory, request approval, request rejection, and fulfillment of requests.

JavaScript was used on the frontend side since it is the primary programming language for creating web interfaces. React.js was used as a library that helps create reusable interface components/pages. Axios was used for sending HTTP requests from the frontend to the backend API. React Router was used for handling navigation between various pages like login, dashboard, donors, inventory, requests, and submitting the request. In terms of backend technologies, the Django REST framework was used to build REST API endpoints, SimpleJWT for token-based authentication, Django CORS Headers for communicating with the frontend application while development, and python-dotenv to handle environment variables.

4.1.2 Development Platform and Hardware Requirements

The proposed system can be developed using Visual Studio Code or any other available IDE, like PyCharm. The reason to choose Visual Studio Code is its support of Python language, JavaScript, console commands, extensions, and project directory structure. In order to start the backend, one should use Django management commands from the terminal, while starting the frontend will be possible with Node.js and Vite. Moreover, a modern browser such as Google Chrome, Microsoft Edge, or Mozilla Firefox is required to test and interact with the user interface.

Minimum requirements for hardware in order to develop and run the system locally include a computer with at least 4GB of RAM, dual-core processor, and enough disk space for project files, virtual environment, Node.js modules, and SQLite database. However, 8GB of RAM or more is preferred to ensure a smooth development process, especially when starting both backend and frontend server simultaneously. The system doesn't need a GPU, because it won't work with machine learning models training and image processing. Moreover, the system is not intended for local running in the production environment; therefore, production deployment will require a server environment, stable internet connection, and a production database like PostgreSQL.

Tool/Technology	Use in the Project
Python	Backend programming language.
Django 5	Backend web framework and ORM.
Django REST Framework	REST API development.

Django REST Framework SimpleJWT	JWT authentication.
SQLite	Development database.
React 18	Frontend user interface.
Vite	Frontend build and development tool.
React Router	Frontend navigation and protected routes.
Axios	HTTP client for API requests.
CSS/Bootstrap-compatible styling	Interface layout and presentation.

4.2 Tools Used for Implementation

4.2.1 Backend

Python

The backend code of the proposed system will be written using the Python programming language. Python is a high level programming language, which is known for its readability, simplicity and flexibility. Python is capable of supporting multiple programming paradigms. In addition, Python has a vast library and frameworks which are commonly used in web development, automation and systems integration. The

choice of Python for this project was made because of the effectiveness of interaction between Python and Django..

Django

Django is an efficient Python framework which can be used to develop secure and scalable web applications. It offers various in-built functionalities like routing, database modeling, authentication capabilities, administration, and Object Relation Mapping. In this system, Django framework has been used to model various databases including User, Hospitals, Donors, Donations, Inventory Units, and Blood Requests. It was used to divide the backend into various applications including Accounts, Donors, Donations, Inventory, Requests, Hospitals, and Reports..

Django REST Framework

The Django REST Framework is a tool that helps in building APIs using Django framework. The REST APIs enable different software systems to communicate using the common HTTP methods such as GET, POST, PATCH, and DELETE methods. In this project, the Django REST Framework has been used to build API endpoints which help in communicating between the React front-end and back-end. Using the API endpoints, the front-end sends log-in credentials, creates donors, registers donations, views inventory, makes hospital requests, and generates reports..

JSON Web Token Authentication

The proposed system was secured using JSON Web Token Authentication. JWT Authentication is a form of token-based authentication whereby the user gets an access token after logging in. The token is included in subsequent API requests as proof of the authenticated user. JWT authentication in the proposed system acts as a way of securing the blood bank database by ensuring that only logged in users access sensitive data.

4.2.2 Frontend

React

React.js is a library for JavaScript that creates user interfaces. It makes it possible to break down the interface into reusable components to make the frontend easy to maintain. In the suggested system, React.js was used to create pages like the landing page, login page, dashboard page, donor page, record donation page, inventory page, requests page, and submit request page. React makes the application responsive and helps users to interact with the system using an intuitive web interface.

Vite

Vite is an application that acts as a frontend build tool providing a fast development server and effective build process for a project. Vite was adopted in the proposed system as the frontend runner in order to develop the React frontend of the system.

Axios

Axios is a JavaScript framework that makes HTTP requests from the front-end to the back-end. In the developed system, Axios was set up to talk with Django REST API. Also, the access token of the user is attached to the request in order to access the protected endpoints of the back-end. Axios is utilized wherever the front-end needs to log in, get donors' information, make a donation, get inventory information, or handle hospital requests.

React Router

React Router is the library used to navigate through the various pages of the frontend application. It enables the website to be able to render different pages, for example, login, dashboard, donor management, inventory, and requests pages without having to reload the whole website. React Router also integrates with private routing and role gates to make sure that the user accesses pages relevant to their role.

4.2.3 Database and Data Format

SQLite

In the process of designing the system, SQLite has been selected as the development database. SQLite is a light-weight relational database that stores its data in a local database file. Its advantage is that it does not need a separate database server to run. In our project, SQLite will store user details, hospital details, donor details, donation details, stock record details, blood request details, and item of request details.

JSON

This stands for JavaScript Object Notation, and this is a light-weighted way of representing data that can be transmitted from the frontend to the backend. In this project, the React frontend transmits the data in JSON format using the REST API. JSON is preferred in this scenario due to its readability, conciseness, and compatibility with modern web apps.

CORS Support

CORS stands for Cross-Origin Resource Sharing, which is a web security technology that enables or blocks interactions between different domains. Given that the frontend and the backend could be running on different localhost addresses during development, it becomes necessary to implement CORS to allow the frontend to talk to the backend. In the proposed architecture, Django CORS headers will enable such interactions.

4.3 System Requirements

The intended system is a web-based application which can operate locally while under development and later can be uploaded on a web server for broader access. But certain software and hardware prerequisites are necessary in order to use or deploy the intended system properly. Some of the prerequisites for deploying the system are a working computer system, operating system like Windows, Linux, or macOS, Python 3.11 or later, Node.js 18 or later, a modern web browser, and a good development environment. The backend includes Django, Django REST Framework, SimpleJWT, Django CORS Headers, and python-dotenv.

4.4 User Interface of the Proposed System

This part of the research work covers the interface design of the different sections that the proposed system has. The interfaces of the system include the landing page, login page, dashboard page, donor management page, donation recording page, inventory page, hospital request page, and submit request page. These pages allow users to input data into the system and also allow the system to display output in a structured and readable manner.

4.4.1 The Landing Page of the Proposed System

The landing page is the first page that will pop up for the user once he visits the system. It gives information about the Blood Bank Management System, and leads the user to the login page. The function of the landing page is to welcome the user and provide him with an easy access route into the system.

4.4.2 The Login Screen of the Proposed System

The login page enables users to input their username and password. The system will send the credentials to the backend authentication endpoint after the user inputs the credentials. In case the credentials provided are accurate, the backend generates tokens and the user is then forwarded to the relevant section of the system. This is an important part of the system since it prevents unauthorized access..

4.4.3 The Dashboard Screen of the Proposed System

This is where all the essential data regarding the blood bank is displayed on the dashboard screen. This information includes the stock available according to the blood type, the stock that is running low, the pending requests, and the blood units that will expire in seven days' time.

4.4.4 The Donor and Donation Screens of the Proposed System

The donor screen enables the staff members to add and display donors.

Information about the donor such as his/her complete name, gender, date of birth, blood type, contact number, address, and whether he/she is eligible for donation is obtained using this screen. The donation screen enables the staff members to enter a donation by specifying the donor, collection date, volume, bag ID, and other notes. This creates an inventory unit automatically.

4.4.5 The Inventory and Request Screens of the Proposed System

The inventory screen provides information on the units of blood that are available within the system. Information such as blood group, unit code, status, expiry date, and linked donation is provided in the inventory screen. The request screen will provide requests from hospitals, and there is an option to accept, reject, or fulfill the requests. Users of hospitals have a request submission screen for requesting a blood group and number of units.

4.5 System Implementation of the Proposed System

The system was built using the above-mentioned software tools. Codebase includes backend and frontend folders. The backend includes Django applications like accounts, hospitals, donors, donations, inventory, requests_app, reports, common, services, and config. The frontend folder has the React application consisting of pages, components, authentication context, API configurations, and routing.

The accounts app provides the definition of the customized user model with admin, staff, and hospital roles. Hospitals app saves data about the hospitals and connects the hospital users to their hospitals. Donors app deals with saving donors' data and generation of donor codes. Donations app saves blood donations and creates the inventory units through the donation service. Inventory app manages blood units based on unit code, blood type, expiration date, and status. Requests app enables hospitals to send requests and for staff to accept, decline, or complete requests. The reports app offers dashboard summaries, inventory reports, donations reports, and request reports.

Tool/Technology	Use in the Project

Python	Backend programming language.
Django 5	Backend web framework and ORM.
Django REST Framework	REST API development.
SimpleJWT	JWT authentication.
SQLite	Development database.
React 18	Frontend user interface.
Vite	Frontend build and development tool.
React Router	Frontend routing and protected pages.
Axios	HTTP client used for API requests.

4.6 Testing of the Proposed System

DATABASE DESIGN TABLES

Proposed Blood Bank Management System

This section presents the database design tables for the proposed computerized blood bank management system. The database is designed to store and manage users, donors, blood groups, blood donations, blood inventory, patients/recipients, and blood requests.

1. Users Table

Field Name	Data Type	Key	Description
user_id	INT	PK	Unique identification number for each user
full_name	VARCHAR(100)	-	Full name of the system user
email	VARCHAR(100)	UNIQUE	User's email address
password	VARCHAR(255)	-	Encrypted user password
role	VARCHAR(30)	-	User role, e.g. Administrator or Staff

2. Blood Group Table

Field Name	Data Type	Key	Description
blood_group_id	INT	PK	Unique identification number for each blood group
blood_group	VARCHAR(5)	UNIQUE	Blood group type, e.g. A+, O-, AB+

3. Donors Table

Field Name	Data Type	Key	Description
donor_id	INT	PK	Unique identification number for each donor
full_name	VARCHAR(100)	-	Full name of the donor
gender	VARCHAR(20)	-	Gender of the donor
date_of_birth	DATE	-	Donor's date of birth
phone	VARCHAR(20)	-	Donor's telephone number
address	VARCHAR(255)	-	Residential address
blood_group_id	INT	FK	Links the donor to the Blood Group table
registration_date	DATE	-	Date the donor was registered

4. Blood Donation Table

Field Name	Data Type	Key	Description
donation_id	INT	PK	Unique identification number for each donation
donor_id	INT	FK	Identifies the donor who donated blood
blood_group_id	INT	FK	Identifies the blood group donated
quantity	INT	-	Quantity of blood donated in units
donation_date	DATE	-	Date of blood donation
screening_status	VARCHAR(30)	-	Result of blood screening

5. Blood Inventory Table

Field Name	Data Type	Key	Description
inventory_id	INT	PK	Unique identification number for the inventory record
blood_group_id	INT	FK	Identifies the blood group
quantity_available	INT	-	Number of available blood units
collection_date	DATE	-	Date the blood was collected
expiry_date	DATE	-	Expiry date of the blood
status	VARCHAR(30)	-	Available, Expired, or Used

6. Patients/Recipients Table

Field Name	Data Type	Key	Description
patient_id	INT	PK	Unique identification number for each patient
full_name	VARCHAR(100)	-	Full name of the patient
gender	VARCHAR(20)	-	Gender of the patient
date_of_birth	DATE	-	Patient's date of birth
phone	VARCHAR(20)	-	Patient's telephone number
blood_group_id	INT	FK	Patient's blood group

7. Blood Request Table

Field Name	Data Type	Key	Description
request_id	INT	PK	Unique identification number for each request
patient_id	INT	FK	Identifies the patient requesting blood
blood_group_id	INT	FK	Blood group requested
quantity_requested	INT	-	Number of blood units requested
request_date	DATE	-	Date of the request
status	VARCHAR(30)	-	Pending, Approved, or Rejected

Primary and Foreign Key Relationships

Relationship	Description
Users → System Activities	A user can manage system operations such as registering donors and managing blood records.
Blood Group → Donors	One blood group can be associated with many donors.
Donors → Blood Donation	One donor can make multiple blood donations.
Blood Group → Blood Inventory	Each blood inventory record belongs to a specific blood group.
Patients → Blood Requests	A patient can make one or more blood requests.
Blood Group → Blood Requests	A blood request specifies the required blood group.

The proposed system has been tested based on the workflow activities which the system would have to undertake. The workflows include logging in by the user, donor registration, donations logging, automatic creation of the inventory, viewing the inventory, submission of requests by the hospitals, approval of the requests, request rejection, and request fulfillment. The test was also done to ensure that hospital users cannot gain access to the staff pages.

Test Case	Expected Result	Module
Login with valid credentials	User receives a token and can access protected pages.	Authentication
Create donor	Donor record is saved with generated donor code.	Donors
Record donation	Donation is saved and one inventory unit is created automatically.	Donations/Inventory
View inventory	Available blood units are displayed by blood group and status.	Inventory
Hospital submits request	Request is created with pending status and linked to hospital.	Requests
Staff approves request	Request becomes approved and available units are reserved.	Requests/Inventory
Staff fulfills request	Selected units become issued and request becomes fulfilled.	Requests/Inventory

4.6 Results Presentation and Analysis

This chapter illustrates the results acquired using the developed **Blood Bank Management System**. The Blood Bank Management System was developed with a view to managing donor registration, donation, stock management, request from hospitals, and approval of such requests from hospitals. It is clear from the results that the software has a functional user interface for both the blood bank personnel and hospital clients.

4.6.1 Sample Output and Interface Screens

The system provides different interfaces depending on the user role. The main users are the **Administrator, Blood Bank Staff, and Hospital User.**

Login Interface

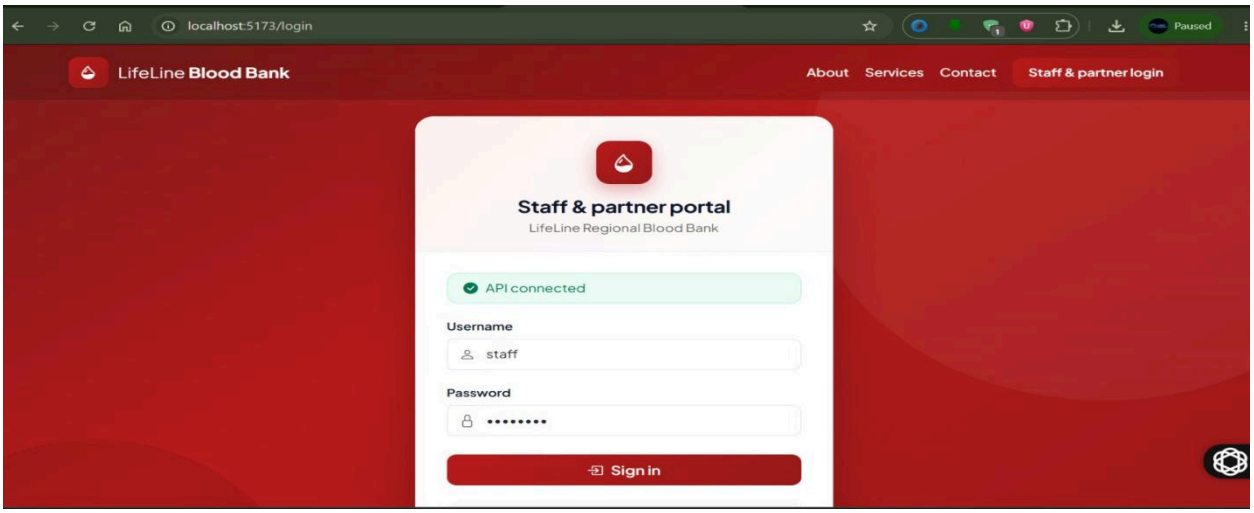

Figure 4.2 Proposed system Login Page

The login page allows registered users to access the system using their username and password. After successful authentication, users are redirected to their role-based dashboard.

Sample users include:

Dashboard Interface

The dashboard gives information about important activities within the system. The dashboard can have information on the total number of donors, blood units available, pending requests, approved requests, and blood units issued.

Sample dashboard output:

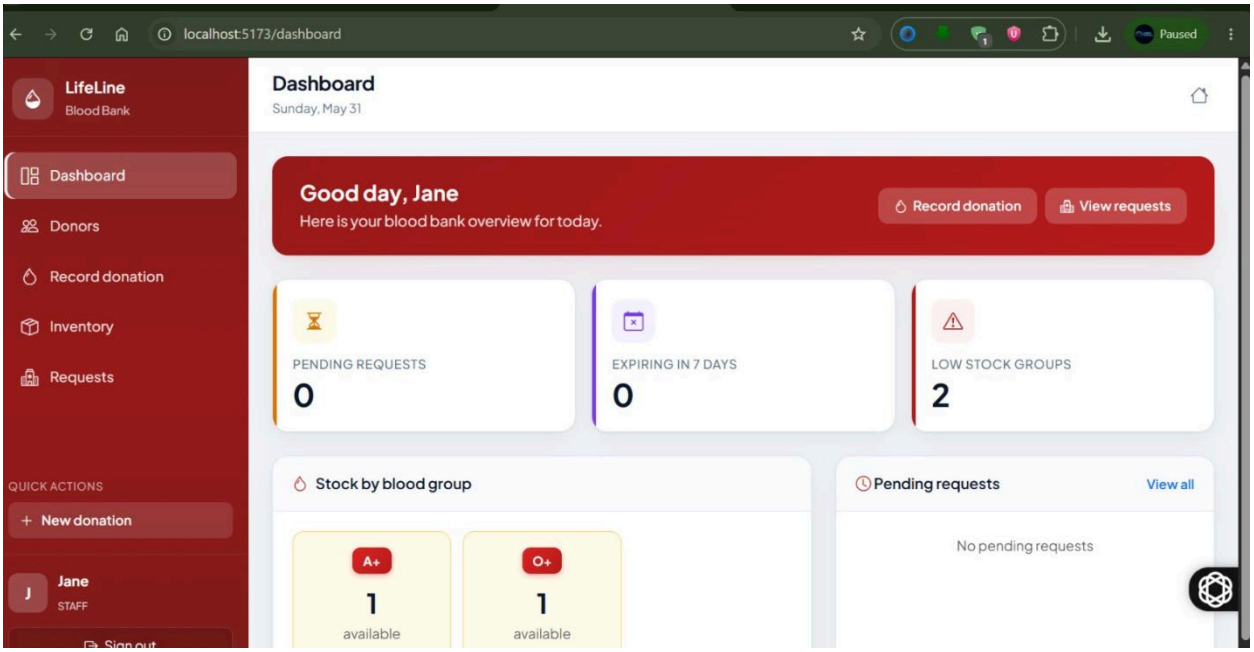

Figure 4.3 Proposed System Dashboard Page

Donor Registration

The staff in the blood bank can enter the details of new donors by inputting the details like name, blood type, phone number, address, and donation eligibility criteria in the system.Sample donor record:

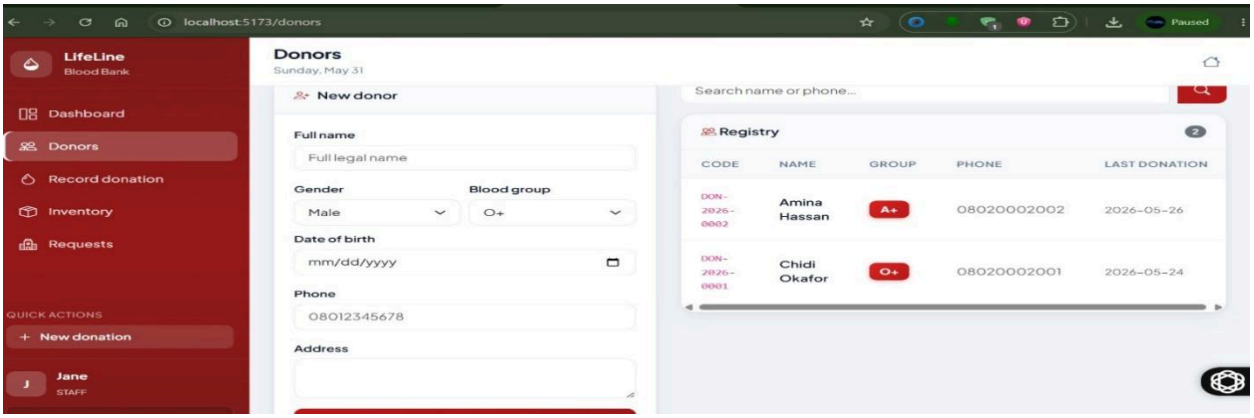

Figure 4.4 Proposed System Donor Registration Page

After a donor gives blood, staff can record the donation. The system links the donation to the donor and creates blood inventory units based on the recorded donation.

Inventory Interface

The inventory page displays available blood units by blood group and status. Staff can check which units are available, issued, expired, or reserved.

Sample inventory output:

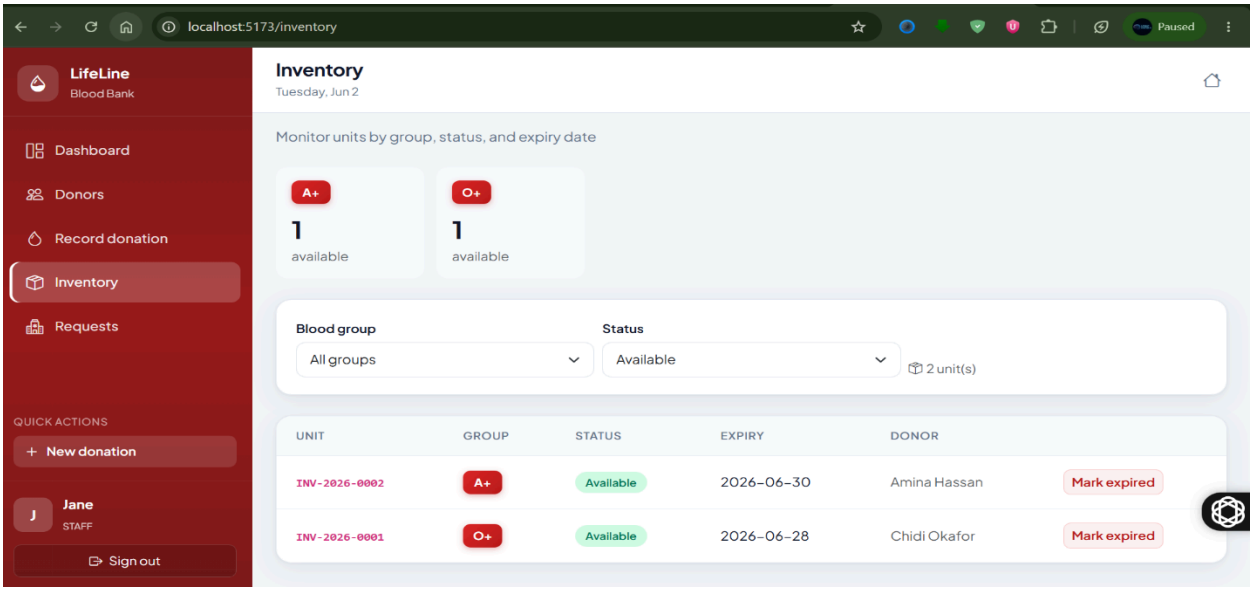

Figure 4.5 Inventory Page Of The Proposed system

Hospital Blood Request Interface

Hospital users can submit a blood request by selecting the required blood group, quantity, and urgency level. After submission, the request is sent to blood bank staff for review.

Samplehospital request:

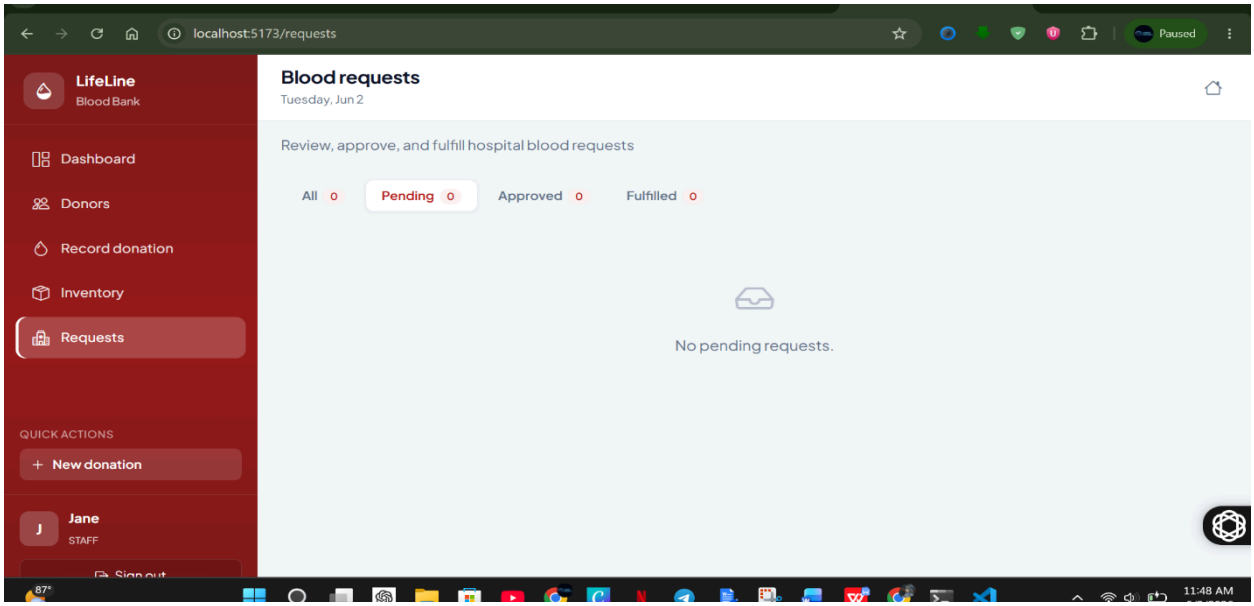

Figure 4.6 Blood Request Page for Proposed system

The blood bank workers have access to the requests made by the hospitals. The worker will either accept, reject, or fulfill the request based on the availability of blood. When the request is fulfilled, the selected blood is then flagged as issued in the inventory system. The request has been approved and fulfilled.

4.6.2 Discussion of Results

It is evident from the findings that the Blood Bank Management System has enabled the achievement of most of the activities involved in running a blood bank. The software enables the employees to register the donors, document the donations, monitor the inventory, and process the requests for the blood from hospitals. In addition, it allows the users from hospitals to make requests without having to go to the

blood bank physically or call the blood bank on phone.

The Blood Bank Management System has helped in organizing the blood bank data through storage of the donors, donations, inventories, and requests in one place. As such, there will be fewer problems such as loss of data, delays in processing of the requests, and inability to know how many blood units are available.

The most significant part of this system is the inventory which ensures that there is no allocation of the same blood unit twice. This is because once a request is fulfilled, the system removes the blood units allocated from the inventory list.

Performance Analysis

The system was effective in sample testing since donor registration was successful, donation entries were done and also entries in inventory were accomplished. Hospital users were able to make requests for blood, while the staff users were able to process those requests. Dashboard gave an overview of system activities.

Role based access control was also working effectively. Hospital users were able to make requests for blood, while the staff users had access to donor management, donations, inventories and request processing functionalities. This ensures that users access only the functions that pertain to them.

The system has the following strengths:

- It provides a centralized platform for managing blood bank operations.
- It reduces manual paperwork and improves record accuracy.
- It allows quick donor registration and donation recording.
- It provides real-time visibility of available blood inventory.
- It enables hospitals to submit blood requests through the system.
- It allows staff to approve, reject, and fulfill blood requests.
- It automatically updates inventory after blood units are issued.
- It supports role-based access for administrators, staff, and hospital users.

Relationship of Results to Project Objectives

The outcomes are clearly associated with the goals of the project. The first goal of the project was to create a mechanism for managing records of donors. This was achieved through the donor registration and donor listing functions.

A second goal was to help in blood donation tracking and stock management. The goal was achieved as the system allows tracking of donations and updating of blood stock after donations and fulfillments.

The project was supposed to enhance handling of hospital blood requests. The goal was achieved through the request submission, approval, rejection, and fulfillment functions.

In summary, the outcomes prove that the Blood Bank Management System has achieved its primary goals. The system helps in managing of records, enhances faster processing of requests, and manages the blood stock. Further improvement of the system will be achieved through the inclusion of notifications, reporting, expiry alerts, and demand forecasting.

CHAPTER FIVE
SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Introduction

This chapter summarizes the research, presents the conclusion drawn from the implemented system, and provides recommendations for future improvement and adoption.

5.2 Summary of the Study

This research was on the development of a web-based Blood Bank Management System. Chapter One presented the background, problem statement, objective, aims, significance, limitations and definition of terms. The core problem was inefficiency and risky nature of manually operated blood bank record.

Chapter Two presented literature review and conceptual framework on blood bank operation, inventory, web-based healthcare, role-based access control, and REST API architecture. The review pointed out the need for a web-based integrated system that linked donor record, donor, blood units, hospital requests and reports.

Chapter Three presented the analysis of existing and proposed system, requirement identification for the system, functional and non-functional requirements, system design approach and textual UML model descriptions of the design. In the design, a three-tier approach was adopted which comprises of React frontend, Django REST API Backend and Relational Database.

Chapter Four presents the Implementation chapter. In the implementation of the backend system, the software packages used are Django, Django REST Framework, SimpleJWT and SQLite. For the implementation of the frontend system, the software packages used are React, Vite, React Router and Axios. The implemented modules are Authentication, Donor management, Donation, Inventory, Hospital requests, Dashboard and Reports.

5.3 Conclusion

The project has been successful in the development and implementation of the Blood Bank Management System. The system fulfills the aims through centralization of data,

provision of role-based security, automation of creation of stock following donations, support to hospital requests, and production of dashboards and reports. The system saves time, increases the visibility of the blood available, and creates a strong base for blood bank management. The current system, even being an MVP, proves that the use of modern web technologies for blood bank management is feasible and beneficial.
lts.

5.4 Recommendations

It should be implemented with a commercial-level database like PostgreSQL for institutional use. Notifications via SMS and email can be made available for low stocks, expiry of units, approval of requests, and completion of requests. Barcode or QR code scanning needs to be included for better management of physical units of blood. Audit logging can be incorporated to track critical transactions. The blood matching algorithm can be enhanced to include emergency matching where needed. The application can also be expanded for analytics, multi-branch functionality, laboratory screening history, and report generation capabilities.

References

1 World Health Organzation (2022). *Evaluation of a blood bank information system in a Nigerian teaching hospital: User experience and functional outcomes*.

2 Asamoah-Akuoko, L., Hassall, O., Allain, J. P., & Owusu-Ofori, S. (2021). *Blood supply challenges in sub-Saharan Africa: A scoping review.* Transfusion Medicine Reviews, 35(2), 78–86.

3 Ben Elmir, W., Hemmak, A., & Senouci, B. (2023). Smart Platform for Data Blood Bank Management: Forecasting Demand in Blood Supply Chain Using Machine Learning. *Information*, *14*(1), 31. https://doi.org/10.3390/info14010031 Cited by: 106

4 Raturi, S., & Shastry, S. (2022). Blood inventory management: Concepts and practices. Asian Journal of Transfusion Science, 16(2), 153–159. https://doi.org/10.4103/ajts.ajts_169_21.

5 Kruse, C. S., Smith, B., Soper, H., Patel, K., Griffin, L., & Thomas, B. (2017). Security techniques for electronic health records: Systematic review. JMIR Medical Informatics, 5(3), e29. https://doi.org/10.2196/medinform.7070

6 **Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996).** *Role-Based Access Control Models*. IEEE Computer, 29(2), 38-47.

7 Azhar, N. M. S., & Ali, F. A. H. (2022). Web-based Development: Smart Blood Bank Management System. *Applied Information Technology and Computer Science*, *3*(2), 333–345.

8 Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O., & Toval, A. (2013). Security and privacy in electronic health records: A systematic literature review. Journal of Biomedical Informatics, 46(3), 541–562.

9 Bhandawat, R., Casucci, S., Ramamurthy, B., & Walteros, J. L. (2022). Cooperative Blood Inventory Ledger (CoBIL): A decentralized decision making framework for improving blood product management. *Computers & Industrial Engineering*, *172*, 108571. https://doi.org/10.1016/j.cie.2022.108571

10 Deshpande, V., Kumar, S., & Chaple, S. (2020). Web based blood bank management system. *International Journal of Advance Research, Ideas and Innovations in Technology*, *6*(2), 555–558.

11 Doneda, M., Yalçındağ, S., Marques, I., & Lanzarone, E. (2021). A discrete-event simulation model for analysing and improving operations in a blood donation centre. *Vox Sanguinis*, *116*(10), 1060–1075. https://doi.org/10.1111/vox.13101

12 Farrington, J., Alimam, S., Utley, M., Li, B. N., and Wong, H. (2024). *Machine learning support for platelet issuing and wastage reduction in hospital blood banks.*

13 Kaur, R., Rai, R., Raj, A., Sharma, S., and Sirohi, P. (2023). *Blood Bank Management System with Emergency Blood Locator.*

14 Kuberkar, S., and Singhal, T. K. (2021). *Adoption of blockchain and IoT for sustainable blood bank management.*

15 Lin, H., Metcalf, R. A., Wilburn, J., & Lex, A. (2021). Sanguine: Visual Analysis for Patient Blood Management. *Information Visualization*, *20*(2-3), 123–137. https://doi.org/10.1177/14738716211028565

16 Nithya Nandhini, R., Vigneshkumar, V., and Ramakrishnan, S. (2026). *AI-based emergency blood donor matching and demand prediction system.*

17 Rad, J., Quinn, J. G., Cheng, C., Liwski, R., Abidi, S. R., & Abidi, S. S. R. (2021). Using interactive visual analytics to optimize in real-time blood products inventory at a blood bank. *Studies in Health Technology and Informatics*, *281*, 223–227. https://doi.org/10.3233/SHTI210153

18 Rokaya, S., Subedi, S., and Dhungana, S. (2025). *Rakta Setu: A web-based blood bank platform.*

19 Shah, A., Shah, S., and Jan, A. (2023). *Online Blood Donation Management System.*

20 Sharma, A., et al. (2023). *Donor Dreams: A centralized blood bank and patient platform.*

· Zadeh, A., Zandieh, M., and Tabriz, A. A. (2025). *Cloud-based blood supply chain management through system dynamics simulation.*

21 World Health Organization. *Blood safety and availability publications.*

22 Django Software Foundation. *Django documentation.*

23 Encode OSS Ltd. *Django REST Framework documentation.*

24 React Documentation. *React library documentation.*

25 Vite Documentation. *Vite frontend tooling documentation.*

26 OpenAPI and REST architectural references for web API design.

Appendix

Appendix A: Codebase Structure Referenced

```
blood-bank-system/
  backend/
    accounts/
    common/
  config/
    donations/
    donors/
    hospitals/
    inventory/
    reports/
    requests_app/
    services/
    manage.py
    requirements.txt
  frontend/
    src/
      api/
      auth/
      components/
      context/
      pages/
      routes/
    package.json
    vite.config.js
  README.md
```

Appendix B: Demo Accounts

Username	Password	Role
admin	password	Administrator
staff	password	Blood bank staff

hospital	password	Hospital user

Appendix C: Setup Summary

Backend:
cd backend
python -m venv venv
.\venv\Scripts\activate
pip install -r requirements.txt
python manage.py migrate
python manage.py seed_demo
python manage.py runserver

Frontend:
cd frontend
npm install
npm run dev