

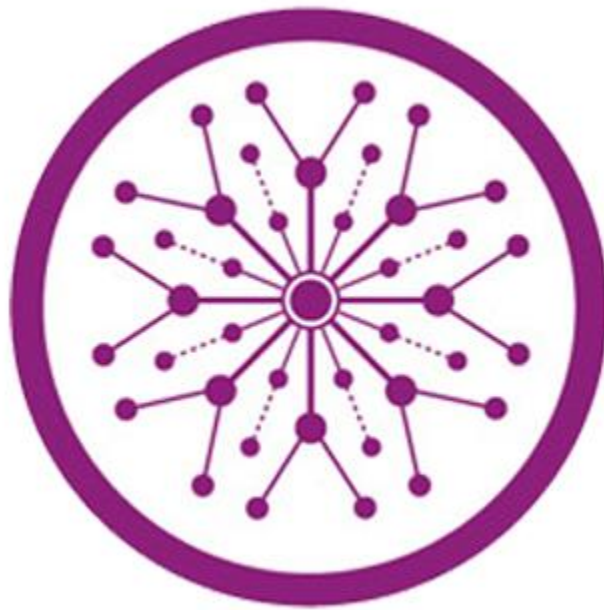
Blood Donation App

Final Year Project

Session 2020-2023

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Fall 2023

Type (Nature of project)	☒ Development ☐ Research ☐ R&D			
Area of specialization				
FYP ID	FYP-BCSM-S23-016			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-S20-036	Hamza Khalid	BCSM-S20-036@superior.edu.pk	
(ii)	BCSM-S20-037	Muhammad Hammad Khan	BCSM-S20-037@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I Hamza Khalid Son of Khalid Mehmood, group leader of FYP under registration no BCSM-S20-036 at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: **29/11/20203** Name of Group Leader: Hamza Khalid Signature: _____

Name of Supervisor: Mr. Rizwan

Designation: Lecturer

Signature: _____

HoD: Dr. Irfan Ud Din

Signature: _____

Project Report

Blood Source

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Hamza Khalid	1.0	02-02-2023	Research	
Hammad Khan	1.1	07-02-2023	Market Analysis	
Hamza Khalid	1.2	18-02-2023	Documentation	
Hammad Khan	2.0	26-02-2023	UI Design	
Hamza Khalid	3.0	15-03-2024	Chapter 1	
Hammad Khan	4.0	25-03-2023	Chapter 2	
Hamza Khalid	5.0	10-04-2023	Chapter 3	
Hamza Khalid	6.0	20-04-2023	Chapter 4	
Hammad Khan	7.0	15-05-2023	Chapter 5	
Hamza Khalid	8.0	16-09-2023	Chapter 6	
Hammad Khan	9.0	1-10-2023	Chapter 7	
Hamza Khalid	8.0	15-10-2023	Presentation	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

"With profound gratitude to the life-saving heroes who give the gift of life, this Blood Donation Mobile App stands as a beacon of hope. Dedicated to every donor, your selfless contribution strengthens the bonds of humanity. May this app serve as a seamless bridge between those in need and the noble hearts ready to share the gift of vitality. Together, we unite to make a lasting impact, one drop at a time. Thank you for embodying the spirit of compassion and saving lives through the power of blood donation."

Acknowledgements

I would like to express my sincere gratitude to my supervisor for their invaluable guidance and unwavering support throughout the development of the blood donation mobile app. Their expertise, insightful feedback, and commitment to excellence played a pivotal role in shaping the project. I am truly thankful for their mentorship, which has not only enriched my technical skills but also inspired a sense of purpose in creating a platform that can potentially save lives through efficient blood donation.

Executive Summary

This application will provide 24/7, support to the blood requesters. The requesters can post anytime from anywhere. Having an online automated system can eliminate the drawbacks and hassles that come with unforeseen circumstances as the process of helping can never come to a standstill. So, our motive is to provide a system for donor and requester's in-need to reach each other without facing the many constraints that lie around us. The requester sees all the donors and selects the nearest donor. Our app recommends nearest donors from his location. Requester directly calls the donor and share details. After accepting the blood, the requester and donor both can give ratings to each other. After donating the blood, profile of donor will automatically lock for 3 months. He can donate blood again but after three months.

Table of Contents

Dedication	5
Acknowledgements.....	6
Executive Summary.....	7
Table of Contents	8
Chapter 1.....	13
Introduction	13
1.1. Background.....	15
1.2. Motivations and Challenges.....	15
1.3. Goals and Objectives.....	17
1.4. Literature Review/Existing Solutions	18
1.5. Gap Analysis	19
1.6. Proposed Solution	21
1.7. Project Plan	22
1.7.1. Work Breakdown Structure	22
1.7.3. Gantt Chart	23
1.8. Report Outline.....	23
Chapter 2.....	25
Software Requirement Specifications	25
2.1. Introduction.....	26
2.1.1. Purpose.....	27
2.1.2. Document Conventions	27
2.1.3. Intended Audience and Reading Suggestions	28
2.1.4. Product Scope.....	29
2.1.5. References	29
2.2. Overall Description.....	29
2.2.1. Product Perspective.....	29
2.2.2. Product Functions.....	29
2.2.3. User Classes and Characteristics	29
2.2.4. Operating Environment.....	30
2.2.5. Design and Implementation Constraints.....	31
2.2.6. User Documentation	31
2.2.7. Assumptions and Dependencies	32
2.3. External Interface Requirements	33
2.3.1. User Interfaces.....	33
2.3.2. Hardware Interfaces	33
2.3.3. Software Interfaces	33
2.3.4. Communications Interfaces.....	34
2.4. System Features	34
2.4.1. System Feature 1	34
2.4.1.1. Description and Priority	34
2.4.1.2. Stimulus/Response Sequences	34

2.4.1.3.	Functional Requirements.....	35
2.4.2.	System Feature 2	36
2.4.2.1.	Description and Priority	36
2.4.2.2.	Stimulus/Response Sequences	36
2.4.2.3.	Functional Requirements.....	37
2.4.3.	System Feature 3 (and so on)	38
2.5.	Other Nonfunctional Requirements.....	45
2.5.1.	Performance Requirements	45
2.5.2.	Safety Requirements	47
2.5.3.	Security Requirements	48
2.5.4.	Software Quality Attributes.....	48
2.5.5.	Business Rules.....	49
2.6.	Other Requirements.....	51
Chapter 3.....		54
Use Case Analysis.....		54
3.1.	Use Case Model.....	55
3.2.	Use Case Descriptions	56
Chapter 4.....		57
System Design		57
4.1.	Architecture Diagram	59
4.3.	Entity Relationship Diagram with data dictionary	60
4.4.	Class Diagram	61
4.5.	Sequence / Collaboration Diagram	62
4.6.	Operation contracts	62
4.7.	Activity Diagram	63
4.8.	State Transition Diagram.....	65
4.9.	Component Diagram	65
4.10.	Deployment Diagram.....	66
4.11.	Data Flow diagram [only if structured approach is used - Level 0 and 1]	67
Chapter 5.....		68
Implementation		68
5.2.	Components, Libraries, Web Services and stubs	69
5.3.	Deployment Environment.....	69
5.4.	Tools and Techniques.....	70
5.5.	Best Practices / Coding Standards.....	71
Chapter 6.....		73
Testing and Evaluation.....		73
6.1.	Use Case Testing.....	74
6.2.	Equivalence partitioning	75
6.3.	Boundary value analysis.....	75
6.4.	Data flow testing	76
6.5.	Unit testing.....	76
6.6.	Integration testing.....	76
6.7.	Performance testing.....	77

6.8. Stress Testing.....	78
Chapter 7.....	79
Summary, Conclusion and Future Enhancements.....	79
7.1. Project Summary.....	80
7.2. Achievements and Improvements	80
7.3. Critical Review	80
7.4. Lessons Learnt	81
7.5. Future Enhancements/Recommendations	82
Appendices.....	84
Appendix A: User Manual	85
Appendix B: Administrator Manual	85
Appendix C: Information / Promotional Material.....	86
Reference and Bibliography.....	90

List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

Chapter 1

Introduction

Chapter 1: Introduction

The project is design to help the procurer to arrange the blood by just requesting the people those want to donate the blood in the way of Allah. This project is a good source for procurer and donor. In the past, one of my friends was need of blood, he talked with the blood bank, which was situated in the hospital, but unfortunately, the blood was not present over there. He called to different people asking for blood. But most of them was sending the list of blood donor. As it's difficult work for the needy person to call randomly to everyone to ask for blood. Anyhow, he called to some people but they said, presently I gave the blood and now you can contact with other one. So, our project will help the procurer to get the blood without facing much problems. Procurer have to install this app and after registration, he can request for blood. He can call the donors that are nearest to his location.

This project requires internet access and thus there is a disadvantage of internet failure. The main aim of developing this application is to reduce the time to a great extent that is spent in searching for the right donor and the availability of blood required. Thus, this application provides the required information in no time and helps in quicker decision-making. There are many modules included in the development of this project and some of them are explained below:

User login:

This allows only the registered donor and procurer users to login in order to use this application.

User:

There are three types of users, which are given below:

Admin: He can view the list of procurers and the blood donor and update, delete, create, and read all the data.

Procurer:

Procurer will request for the blood and directly calls the donor and shared details.

Donor: This module helps the donor to insert all the necessary details that is personal information and medical information plus the blood group, which together helps to select a donor.

1.1. Background

This project intends to create a blood donation app in order to overcome these issues and eliminate the distance between blood donors and recipients. The software uses mobile technology and connection to develop a user-friendly platform that links people looking to donate blood with those who require blood transfusions. The app aims to streamline the blood donation process and enhance all aspects of blood transfusion services by offering a practical and effective medium for coordination and communication.

The goal of the blood donation app is to engage users in the process of saving lives through blood donation. The app intends to increase awareness of blood donation, promote regular blood donations, and encourage frequent blood transfusions by leveraging the extensive use of smartphones and the growing popularity of mobile applications.

1.2. Motivations and Challenges

Motivations

The urgent need to address blood shortages and enhance the effectiveness of the blood donation procedure is what spurred developers to create a blood donation app. Blood banks sometimes struggle to keep an appropriate supply of blood on hand, which can have serious ramifications for patients who are in need. By developing a smartphone app, we hope to eliminate the gap that now exists between blood donors and recipients, providing a simple and convenient way for people to support this vital cause. The main goal of the app is to improve accessibility and convenience by making it simple for potential donors to register, locate nearby blood drives, and receive notifications when their blood type is in high demand. By offering personalized profiles, the app also seeks to improve donor engagement and retention.

Challenges

Trust and Security: Building trust among potential donors and ensuring the security of their personal information is crucial. People need to feel confident that their data is protected and will not be misused.

User Adoption: Encouraging individuals to download and use the app can be a challenge. You'll need to focus on user-friendly design, intuitive navigation, and effective marketing strategies to increase adoption rates.

Verification and Eligibility: Verifying the eligibility of donors is essential for ensuring the safety of the blood supply. Implementing a robust verification process that checks criteria such as age and health conditions.

Geographic Coverage: Blood donation needs to be accessible across different locations. Expanding the app's reach to cover a wide range of regions can be challenging, especially in areas with limited technological infrastructure or low smartphone penetration.

Volunteer Retention: Encouraging donors to return for repeat donations can be difficult. Implementing features such as reminders, rewards, and personalized updates can help in retaining volunteers and sustaining a steady blood supply.

Scalability and Reliability: As your app gains popularity, it should be able to handle increased traffic and maintain a high level of performance. Scalability and reliability are vital to handle a growing user base without compromising the user experience.

Legal and Regulatory Compliance: Blood donation apps need to adhere to various legal and regulatory requirements, such as data privacy laws and healthcare regulations. Ensuring compliance can be a complex process that may require collaboration with legal experts.

Public Awareness and Education: Promoting the app and raising awareness about the importance of blood donation can be challenging. Developing effective marketing campaigns and educational materials can help overcome this challenge.

Maintenance and Updates: Once the app is launched, it requires ongoing maintenance and updates to fix bugs, add new features, and adapt to changing technologies and user needs. Ensuring a dedicated team is available for continuous development and support is essential.

1.3. Goals and Objectives

Goals

Raise Blood Donation Rates:

The main objective of the blood donation app is to raise blood donation rates by enhancing the procedure for potential donors by making it more convenient, approachable, and entertaining. The software seeks to increase the number of donors, promote regular giving, and ultimately aid in maintaining a sufficient and sustainable blood supply.

Bridging the Gap between Donors and Recipients:

By offering a seamless platform for connecting donor availability with recipient needs, the app aims to bridge the gap between blood donors and recipients. The app seeks to ensure that given blood reaches the appropriate people at the correct time, especially during emergencies or urgent situations, by improving effective communication and coordination.

Enhance Donor Engagement and Retention:

Improving donor engagement and retention is a key objective of the app. The app will have elements that encourage donor engagement, like custom profiles, thanks for donations, and the option to follow the results of their donations. The app attempts to motivate contributors to continue participating and giving frequently by encouraging a sense of commitment and appreciation.

Optimize Resource Allocation:

The goal of the blood donation app is to allocate donated blood and associated resources as efficiently as possible. The app will offer blood banks and healthcare organizations useful insights by gathering and analyzing data on donor behavior, demographics, and donation trends. Strategic planning, targeted outreach initiatives, and resource allocation can all benefit from these insights, guaranteeing effective use of donated blood and reducing wastage.

Objectives

User-Friendly and Accessible Mobile App:

A user-friendly and accessible mobile application that potential blood donors can quickly install and use will be the main emphasis of the app development process. To ensure inclusivity and

simplicity of use, the app will priorities intuitive design, linguistic support, and compatibility with various devices and operating systems.

Implement Real-Time Communication and messages:

To provide donors with timely messages and updates, the app will have real-time communication features. It will enable blood banks to issue urgent alerts during crises or during periods of high demand for a particular blood type. The goal is to make sure that potential contributors get information quickly so they can meet the needs of recipients.

Personalize Donor Profiles and Recognition:

The app will give donors personalized profiles so they can keep track of their contributions' history, see how they've affected others, and receive badges or other awards in appreciation. Enhancing donor participation, producing a sense of accomplishment, and strengthening the bond between donors and the cause are the goals.

Data Collection and Analysis for Insights:

The app will gather and examine donor information, such as demographics, frequency of donations, and geographic distribution. Gaining useful insights into donor behavior, preferences, and trends is the goal. In order to best serve the unique requirements of various regions and communities, these insights will be used to inform targeted outreach initiatives, enhance donor engagement tactics, and optimize resource allocation.

1.4. Literature Review/Existing Solutions

Existing Blood Donation Apps:

Various blood donation apps have been created and put into use all across the world. Examples include the Blood Donor App by the blood donation Pakistan and blood connect. These apps have functions including registering donors, making appointments, sending reminders, and tracking prior donations.

Accessibility and Convenience:

Studies have shown how beneficial blood donation applications are for potential donors in terms of accessibility and convenience. According to research, applications can raise awareness,

make it easier to access information about blood drives, and give real-time alerts on critical needs and blood shortages. Increased involvement and donor engagement may result from this accessibility.

Engagement and Retention of Donors:

Keeping a steady supply of blood depends on donor retention and engagement. According to published research, efficient mobile app use can improve donor engagement and promote repeat donations. It has been discovered that features like personalized profiles, recognition, and gamification components have a beneficial impact on donor commitment and motivation.

Communication and Coordination:

Effective communication and coordination between blood banks, donors, and receivers are crucial for the smooth operation of the blood donation process. By offering real-time notifications, emergency alerts, and updates on donation campaigns, blood donation applications can enhance communication. Through improved coordination made possible by these apps, blood banks are better able to match recipient needs with available donors.

Challenges and Limitations:

There are various difficulties and restrictions with the current blood donation apps. Low adoption rates among potential donors, privacy issues with disclosing private and medical data, and a lack of connectivity with current blood bank systems are a few of them. The user experience might also be hampered by technical concerns, such as software crashes or compatibility issues.

1.5. Gap Analysis

Accessibility Gap:

The existing blood donation procedure could not be easily accessible, which would reduce the number of potential donors. People who are restricted by geography, lack of understanding, or time constraints could be left out by physical donation centers and limited outreach initiatives. This gap can be filled by the blood donation app, which offers a user-friendly platform that makes it simple for people to sign up as donors, locate blood drives nearby, and get reminders

about donation opportunities. The software improves accessibility and broadens the pool of possible donors by doing this.

Communication Deficit:

Good communication is essential to the blood donation process, particularly in cases of emergency or pressing need. It's possible that the current routes of communication among blood banks, donors, and receivers are ineffective or out of date, leading to delays or missed opportunities. By providing real-time communication capabilities, the blood donation app can close this gap by enabling blood banks to send urgent notifications to potential donors based on their location and blood type. This function makes it easier to quickly mobilize donors, speeding up reaction times and ultimately saving lives.

Gap in Donor involvement:

Keeping a steady supply of blood depends on donor involvement. However, it's possible that long-term donor retention and motivation won't be as successful with the present approaches to donor engagement. By offering donors personalized profiles, tracking their donation histories, and honoring their contributions with badges or awards, the blood donation app can close this gap. The app can also remind qualified donors to donate frequently by sending them reminders. The app attempts to promote a sense of commitment and raise donor retention by improving donor interaction.

Data Underutilization Gap:

Data are frequently underutilized or fragmented during the blood donation procedure. The insights derived from donor behavior, preferences, and donation patterns may be limited by existing systems' inability to accurately record and analyses donor data. By gathering and examining information on donor demographics, donation frequency, and geographic distribution, the blood donation app can close this gap. Blood banks and healthcare organizations can improve the blood donation process by utilizing data analytics to acquire insightful information for focused outreach efforts, strategic planning, and resource allocation.

1.6. Proposed Solution

User-Friendly Interface and Accessibility:

The blood donation app will have a user-friendly layout that guarantees a seamless and simple user experience. It will also be accessible. In order to serve a wide range of users, it will be compatible with a variety of mobile devices and operating systems. To accommodate various language preferences and provide inclusivity and accessibility for all potential donors, multilingual support will be implemented.

Donor Registration and Profile Management:

Potential donors can establish personalized profiles and register as donors using the app, which will make the procedure quick and easy. Donors will have the option of entering their contact details, medical background, and blood type. Additionally, they can customize their choices for receiving updates such as reminders for appointments and donations. The software will manage and securely store donor profiles, assuring privacy and data security.

Real-Time Matching of Donors and Recipients:

The suggested solution will include a real-time matching algorithm that matches donors and recipients according to blood type, location, and urgency. The app will notify eligible donors nearby when a need for a certain blood type emerges, facilitating quick responses. The matching algorithm will give priority to closeness and compatibility, ensuring quick and accurate matching of donors and recipients.

Donor Recognition and Rewards:

The app will have elements that acknowledge and thank contributors for their donations in order to promote donor engagement and retention. For their contributions, donors will be given badges, certificates, or virtual gifts of appreciation. Additionally, the app might have a points-based system where donors can accrue points for each donation and exchange them for benefits or incentives.

1.7. Project Plan

Phase 1: Initiation and Requirements Analysis (2 weeks)

Phase 2: Design and Technology Selection (4 weeks)

Phase 3: Development (8 weeks)

Phase 4: Testing and Deployment (6 weeks)

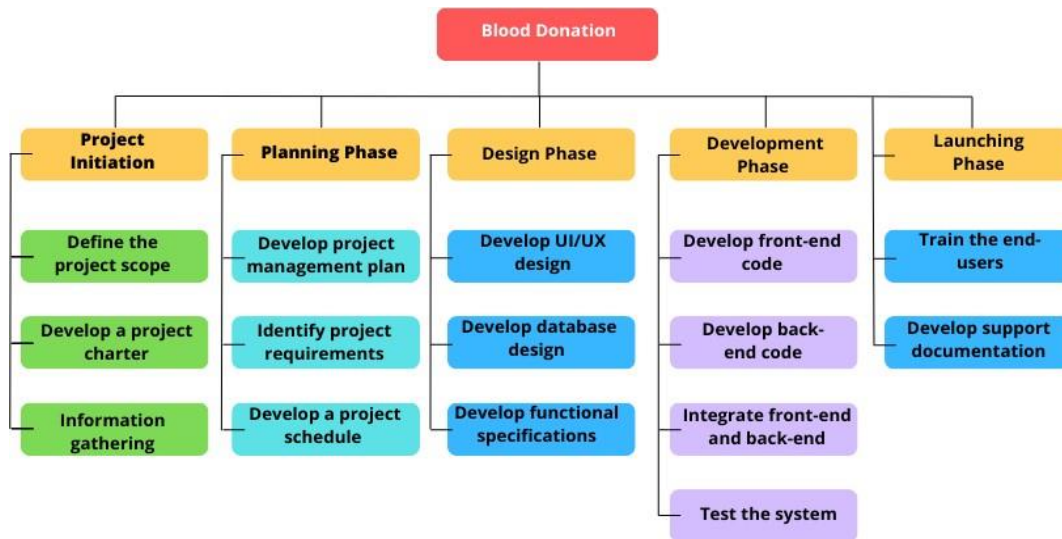
Phase 5: Marketing and Awareness (4 weeks)

Phase 6: User Training (2 weeks)

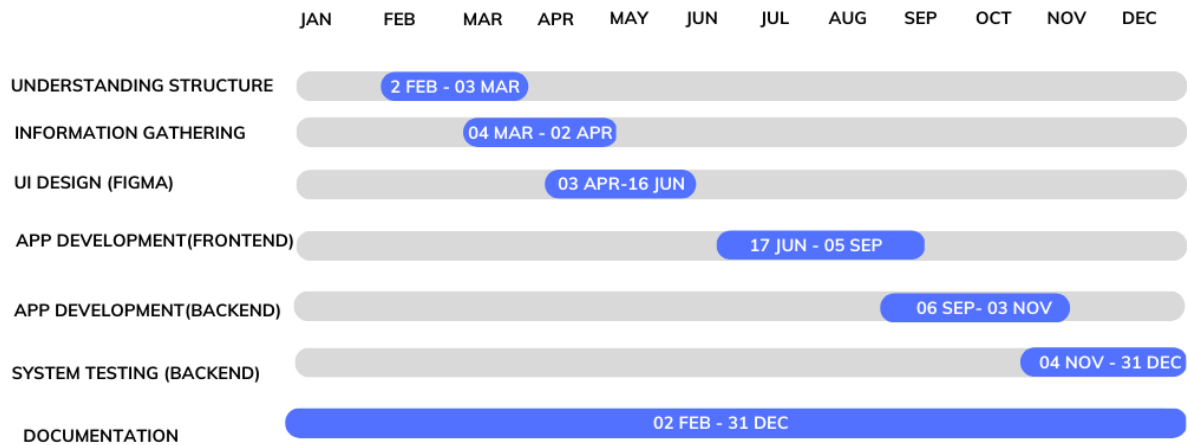
Phase 7: Maintenance and Updates (Ongoing)

Phase 8: Evaluation and Future Enhancements (Ongoing)

1.7.1. Work Breakdown Structure



1.7.2. Gantt Chart



1.8. Report Outline

I. Introduction

- A. Background and significance of blood donation
- B. Purpose and objectives of the blood donation app
- C. Overview of the app's features and functionalities

II. User Interface and Experience

- A. Overview of the app's user interface design
- B. User experience flow within the app
- C. Accessibility considerations for a diverse user base

III. Donor Management System

- A. Donor registration process
- B. Donor eligibility assessment and screening procedures
- C. Donor profile management and updates

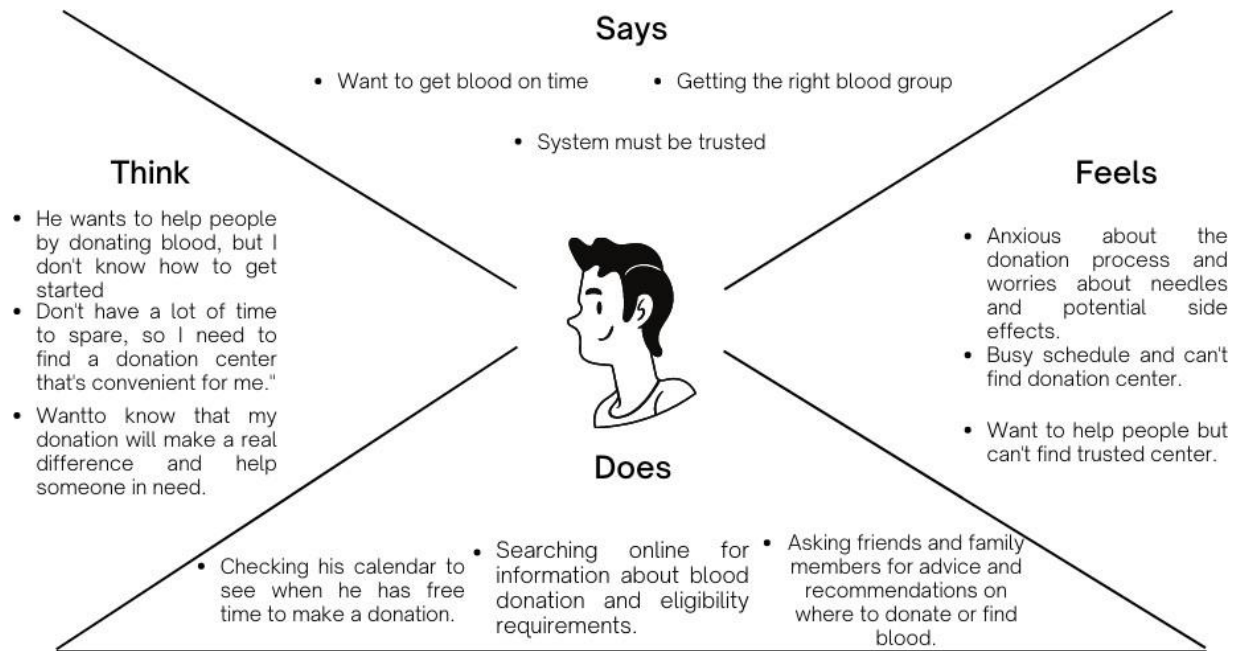
III. Blood Request and Matching System

- A. Find donor by nearest location
- B. Search donor by blood group
- C. Contact with Donor

IV. Rewards

- A. Recognition and rewards for active donors
- B. Gamified challenges to encourage regular donations
- C. Incentive programs to increase donor participation

1.9. Empathy Map



Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

A thorough summary of the specifications for creating a blood donation app may be found in the Software Requirement Specifications (SRS) document. This document defines the system restrictions, user expectations, and functional and non-functional requirements necessary for the app to be implemented successfully.

2.1.1. Purpose

The purpose of this document is to clearly define the scope, features, and specifications of the blood donation app. It serves as a reference for the development team, stakeholders, and testers to ensure that all necessary functionalities are included and implemented accurately.

2.1.2. Document Conventions

Priority Inheritance: Priorities assigned to higher-level requirements are assumed to be inherited by detailed requirements unless explicitly specified otherwise.

Detailed requirements statements are not required to have their own priority unless there is a specific need to deviate from the inherited priority.

Section and Subsection Organization: The SRS is organized into sections and subsections to provide a clear structure and facilitate easy navigation.

Each section and subsection are labeled with a heading or title to indicate its content.

Requirement Numbering: Each requirement statement is assigned a unique identifier or number to enable easy reference and traceability. The numbering system may follow a hierarchical structure, such as using section numbers or a combination of section and requirement numbers.

Fonts and Formatting:

- Standard fonts and formatting conventions are used throughout the SRS for readability and consistency.
- Typically, a standard font (such as Times New Roman, Arial, or Calibri) with a readable size is used for the text.

- Headings, titles, or section labels may be presented in a bold or larger font size to distinguish them from regular text.
- Bullet points or numbering may be used for lists or subsists to enhance clarity and organization.
- Special Significance or Highlighting:
 - There may be specific keywords, terms, or phrases that are highlighted or emphasized to draw attention to important concepts, requirements, or constraints.
- Special formatting, such as italics or underlining, may be used to indicate specific emphasis or special significance.

2.1.3. Intended Audience and Reading Suggestions

Developers: Developers involved in the implementation of the blood donation app would primarily focus on the detailed functional and non-functional requirements, system architecture, and technical specifications. They would be interested in sections that provide information about the app's features, data structures, algorithms, integration points, and technology stack.

Project Managers: Project managers would require a holistic understanding of the project scope, objectives, constraints, and timelines.

They would be interested in sections related to project overview, business goals, high-level requirements, milestones, resource allocation, and project management approach.

Marketing Staff: Marketing staff would be interested in understanding the key features, target audience, and value proposition of the blood donation app.

They would focus on sections that describe the app's benefits, user personas, competitive analysis, and marketing strategies.

Users: Users of the blood donation app, such as donors, recipients, or healthcare professionals, would be interested in sections that explain how the app works, its user interface, and the benefits it offers.

They would also be interested in sections related to user requirements, usability, privacy, and support.

Testers: Testers would primarily focus on sections that define the test cases, test scenarios, and acceptance criteria for validating the app's functionality and performance.

They would be interested in sections related to system testing, integration testing, user acceptance testing, and test data requirements.

Documentation Writers: Documentation writers would refer to the entire SRS to gather information and understand the context, requirements, and specifications of the blood donation app. They would focus on all sections of the document to extract relevant details and create user manuals, system documentation, or other technical documents.>

2.1.4. Product Scope

The goal of the blood donation app is to effectively and conveniently match blood donors with receivers. Finding blood donors and managing a database of potential donors will all be made easier by the app. It will offer a smooth user experience and improve the process of giving blood in general.

2.1.5. References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2.2. Overall Description

2.2.1. Product Perspective

2.2.2. Product Functions

- User Registration and Profile Management
- Blood Donation Search
- Communication between Donors and Recipients
- Real-Time Blood Availability Updates
- History of recent called donors

2.2.3. User Classes and Characteristics

Donors: Characteristics: Individuals who are willing to donate blood.

Pertinent Characteristics: Good health, meeting eligibility criteria (e.g., age, weight), compatible blood type.

Role: Actively search and participate in blood donation opportunities, provide necessary information, and coordinate with recipients or blood banks.

Recipients: Characteristics: Individuals who require blood for medical purposes.

Pertinent Characteristics: Specific blood type requirements, urgency levels for blood transfusion.

Role: Submit blood requests, provide essential medical information, and communicate with potential donors or blood banks.

App Administrators: Characteristics: Individuals or teams responsible for managing and maintaining the blood donation app.

Pertinent Characteristics: Technical expertise, administrative privileges.

Role: Manage user accounts, oversee system operations, handle technical issues, ensure data security and privacy, monitor app usage and performance.

Regulatory Authorities:

Characteristics: Government agencies or bodies responsible for overseeing and regulating blood donation practices.

Pertinent Characteristics: Legal and regulatory knowledge, policy enforcement.

Role: Establish and enforce regulations, audit blood donation activities, ensure compliance with safety and quality standards.

2.2.4. Operating Environment

Hardware:

Servers: The blood donation app may require dedicated servers or cloud-based infrastructure to host and manage the application, database, and other related services.

User Devices: The app should be accessible and compatible with a range of user devices such as smartphones and tablets.

Software: Operating System: The app should be designed to run on popular operating systems such as Android.

Development Frameworks: The app may be built using specific programming languages, frameworks, and libraries for the development of frontend, backend, and database components.

Database Management System: A reliable and scalable database system is needed to store and manage user data, blood donation records, and other relevant information.

Third-Party Integrations: The app may integrate with external services for features like push notifications, SMS gateways, or payment gateways.

Network: Internet Connectivity: The blood donation app requires a stable internet connection for users to access and interact with the app's features.

APIs and Web Services: The app may rely on APIs and web services to communicate with external systems such as blood banks, hospitals, or notification services.

Security: The app should implement secure communication protocols (e.g., HTTPS) to protect user data and ensure the privacy and confidentiality of sensitive information.

Data Centers: If the app uses dedicated servers, they may be housed in secure and reliable data centers to ensure high availability and data redundancy.

Monitoring and Maintenance: Ongoing monitoring and maintenance processes should be in place to address any performance issues, apply software updates, and ensure smooth operation of the app.

Data Protection: The app should comply with relevant data protection laws and regulations to ensure the security and privacy of user data.

Medical Regulations: Depending on the jurisdiction, the app may need to adhere to specific medical regulations and quality standards related to blood donation and transfusion practices.

2.2.5. Design and Implementation Constraints

2.2.6. User Documentation

User Manual: A comprehensive document that provides an overview of the blood donation app, its purpose, and the steps required to use its various features. It covers topics such as registration, finding donation opportunities, requesting blood, managing user profiles, and engaging in communication with other users.

Tutorials: Interactive guides or walkthroughs that demonstrate how to perform specific tasks within the blood donation app. These tutorials can include videos, animations, or interactive

simulations to provide users with visual and practical guidance. Tutorials may cover topics such as account setup, blood donation process, searching for recipients, and managing notifications.

Frequently Asked Questions (FAQ): A document or section that addresses common questions and provides concise answers to assist users in resolving queries or concerns. The FAQ section may cover topics related to eligibility criteria for blood donation, privacy settings, security measures, and guidelines for interacting with other users.

Release Notes: Documentation that outlines the changes, enhancements, and bug fixes introduced in each version or update of the blood donation app. Release notes inform users about new features, improvements, and any known issues or limitations. They can provide insights into the app's evolution and keep users informed about updates.

Glossary: A list of terms, acronyms, and definitions specific to the blood donation app. The glossary helps users familiarize themselves with the terminology used in the app, providing clarity and understanding of technical or medical terms.

2.2.7. Assumptions and Dependencies

Assumptions:

User Availability: It is assumed that there will be a sufficient number of potential blood donors and recipients who will actively engage with the app and participate in the blood donation process.

Internet Connectivity: It is assumed that users of the app will have reliable internet connectivity to access and use the app's features effectively.

User Device Compatibility: The app assumes that users will have compatible devices such as smartphones, tablets, or computers to access the app. It is assumed that users will have devices with up-to-date operating systems and browsers.

Regulatory Compliance: The app assumes compliance with relevant regulations and guidelines related to blood donation and privacy laws in the target regions of operation.

Dependencies:

External APIs and Services: The app may depend on integration with external services such as blood bank databases, payment gateways, SMS gateways, or push notification services to provide seamless functionality.

Database Management System: The app may depend on a specific database management

system for storing and retrieving user data, donation records, and other relevant information.

Security Measures: The app may depend on security measures such as encryption protocols, secure communication channels, and authentication mechanisms to protect user data and maintain privacy.

Ongoing Support and Maintenance: The app may depend on ongoing technical support and maintenance to address issues, perform updates, and ensure the app's smooth operation.

2.3. External Interface Requirements

2.3.1. User Interfaces

Donor Interface: With the use of this interface, donors will be able to set up and manage their profiles, look for chances to give, book appointments, view their gift history, and get alerts about their contributions.

Acceptor Interface: Through this interface, receivers will be able to create and manage their profiles, look for possible blood donors, request donations, and get alerts when donors become available.

Interface for Admin: Admin will allow to view or modify all profiles.

2.3.2. Hardware Interfaces

The blood donation app will be created to run on portable devices like smartphones and tablets, taking advantage of their hardware capabilities, such as:

GPS: The app may use GPS technology to pinpoint the user's location in order to provide location-based services, such as locating blood drives or donation centers close by.

2.3.3. Software Interfaces

The blood donation app will communicate with a variety of computer programmers and online resources, such as:

Operating Systems: To ensure seamless functionality across many platforms, the app will be compatible with the most recent iterations of well-known mobile operating systems, such as Android.

Backend Services: To store and retrieve user data, donor records, the app will interface with backend servers and databases.

The software will interface with notification systems to deliver updates, reminders, and alerts in real-time to users' devices.

Authentication Services: For safe user authentication and data protection, the app uses login and password.

2.3.4. Communications Interfaces

The blood donation app will require communication interfaces to facilitate real-time communication between donor and acceptor. These interfaces may include:

Internet Access:

In order to sync data, send notifications, and access contact, the app will need internet access.

2.4. System Features

User Registration and profile management:

2.4.1. System Feature 1

2.4.1.1. Description and Priority

User Registration and profile management:

Users can create accounts and manage their profiles with personal information.

Priority: High

Benefit: 9

Penalty: 2

Cost: 5

Risk: 3

Risk: 4

2.4.1.2. Stimulus/Response Sequences

User Registration and Profile Management:

Stimulus: User selects the registration option.

Response: App displays a registration form for the user to enter personal information.

Stimulus: User submits the registration form.

Response: App validates the information, creates a user account, and redirects the user to their profile page.

2.4.1.3. Functional Requirements

Registration: The app should provide a user-friendly registration process to allow users to create new accounts. Users should be able to enter their personal information, such as name, email address, phone number, and password, during registration. The app should validate the entered information to ensure its accuracy and completeness. The registration process should include appropriate error handling and provide clear error messages for any validation failures. Once registered, users should receive a confirmation email or SMS to verify their account.

Login: Users should be able to log into the app using their registered email address or phone number and password. The app should authenticate user credentials and provide appropriate error messages for login failures. Secure mechanisms, such as encryption and secure hashing, should be employed to protect user passwords during login.

Profile Creation and Editing: Users should have the ability to create and edit their profiles after registration. The app should allow users to provide additional information, such as blood type, age, weight, and medical history, to complete their profiles. Validation checks should be implemented to ensure the accuracy and validity of the entered profile information.

Users should have the option to upload a profile picture or identification documents if required.

Profile Viewing: Users should be able to view their own profiles and verify the information stored. The app should display the user's profile information in a clear and organized manner.

Users should have the ability to review and update their profile information at any time.

Privacy Settings:

The app should provide privacy settings that allow users to control the visibility of their profile information. Users should have the option to choose what information is visible to other app users or recipients. Privacy settings should include options for hiding specific details or making the profile entirely private.

Account Recovery: The app should include a mechanism for users to recover their accounts in case they forget their password. Users should be able to initiate the account recovery process

by providing their registered email address or phone number. The app should send a password reset link or verification code to the user's registered email address or phone number. Users should be guided through the password reset process and allowed to set a new password for their account.

Account Deactivation: Users should have the option to deactivate or delete their accounts if they no longer wish to use the app. The app should provide a straightforward process for account deactivation, including confirmation prompts to prevent accidental account deletion. Deactivated accounts should be securely archived and not accessible by the user or other app users.

User Data Protection:

The app should implement appropriate security measures to protect user data, including profile information, from unauthorized access or data breaches. User data should be stored securely using encryption and access controls. The app should comply with applicable data protection regulations and guidelines to ensure the privacy and security of user information.

2.4.2. System Feature 2

Donor Search and Matching:

2.4.2.1. Description and Priority

Donor Search and Matching: Description: Donors can search for donation opportunities based on location, date, and blood type. The app matches donors with recipients based on compatibility criteria.

Priority: High

Benefit: 8

Penalty: 3

Cost: 6

2.4.2.2. Stimulus/Response Sequences

Donor Search and Matching:

Stimulus: Donor selects the search option.

Response: App displays search filters and options for location, date, and blood type.

Stimulus: Donor selects search criteria and submits the search request.

Response: App retrieves matching donors based on the specified criteria and displays the search results.

2.4.2.3. Functional

RequirementsSearch Filters:

The app should provide search filters for donors based on criteria such as location, date, and blood type.

Users should be able to select one or multiple filters to refine their search.

Matching Algorithm: The app should employ a matching algorithm that compares donor and recipient profiles based on compatibility criteria such as blood type, urgency, and other relevant factors. The algorithm should prioritize donors who are most suitable for the specific recipient's needs.

Search Results: The app should display search results in a clear and organized manner, presenting relevant information about potential donors. Each search result should include details such as the donor's name, blood type, location, availability, and any additional information provided by the donor.

Donor Profiles: The app should allow donors to create and manage their profiles, including personal information, blood type, availability for donation, and any relevant medical history. Donor profiles should be visible to recipients and healthcare professionals for evaluation and matching purposes.

Availability Status: Donors should have the ability to set their availability status for donation, indicating whether they are currently available or unavailable for donation. The app should update the availability status in real-time based on the donor's preferences and previous donations.

Notification of Matching Opportunities: Donors should receive notifications when there is a matching opportunity with a recipient in need of their blood type. The app should send timely notifications to inform donors about potential matches, allowing them to respond promptly.

Donor Response and Confirmation: Donors should have the ability to review and evaluate the matching opportunity before confirming their availability for donation. The app should provide

a response mechanism for donors to accept or decline the matching opportunity. Upon confirmation, the app should update the donor's availability status and notify the recipient or healthcare professional.

Communication Between Donors and Recipients:

The app should facilitate communication between donors and recipients to exchange necessary information, clarify any queries, and coordinate donation logistics.

In-app messaging or chat features should be provided for secure and convenient communication.

Verification and Authentication:

The app should implement a verification and authentication process to ensure the legitimacy and credibility of both donors and recipients. Donors' identities and eligibility criteria should be verified by healthcare professionals before finalizing the matching process.

Reporting and Monitoring:

The app should provide reporting and monitoring capabilities to track the effectiveness of the donor search and matching feature. Metrics such as successful matches, response rates, and overall satisfaction should be monitored to improve the matching process and enhance user experience.

2.4.3. System Feature 3

2.4.3.1 Description and Priority

Donation History:

Description: Donors can view their donation history, including dates, locations, and quantities of previous donations. Recipients can track the blood they received.

Priority: Medium

Benefit: 7

Penalty: 2

Cost: 4

Risk: 2

2.4.3.2 Stimulus/Response Sequences

Donation History:

Stimulus: Donor navigates to the donation history section.

Response: App retrieves the donor's donation records and displays them in chronological order.

Stimulus: Donor selects a specific donation record for more details.

Response: App presents detailed information about the selected donation, including date, location, and quantity donated.

2.4.3.2 Functional requirements

View Donation History:

The donor should be able to view their donation history, including the dates, locations, and quantities of their previous donations. The donation history should be displayed in a chronological order with the most recent donations appearing first.

Filter and Sort Donations: The donor should be able to filter and sort their donation history based on specific criteria, such as donation date, location, or donation quantity. The app should provide options to filter and sort the donation history to help donors find specific donations or analyze their donation patterns.

Detailed Donation Information: The app should provide detailed information about each donation, including the donation date, location, quantity donated, and any relevant notes or comments associated with the donation.

Donation Status Tracking: The app should display the status of each donation, indicating whether the donation was successfully completed, pending verification, or rejected for any reason. The donor should be able to track the progress of their donations and know if any further action is required.

Edit and Update Donation Information: The donor should have the ability to edit and update the information associated with a specific donation, such as correcting the donation date, location, or adding additional details if necessary. The app should allow donors to make changes to their donation history while ensuring data accuracy and integrity.

Donation Certificates or Receipts: The app should provide the option for donors to generate and download donation certificates or receipts for their contributions. The certificates or

receipts should include essential details such as the donor's name, donation date, location, and any necessary legal or tax-related information.

Donation Statistics and Analytics: The app may provide statistical analysis and insights into the donor's donation history, such as the total number of donations, cumulative blood volume donated, average donation frequency, and other relevant metrics. The app should present the donation history in a visually appealing and informative manner to help donors track their impact and understand their contributions.

Export or Share Donation History: The donor should be able to export or share their donation history, either as a file or through various communication channels (email, messaging apps, etc.). The exported donation history should be in a format that is easily readable and can be shared with healthcare professionals, employers, or for personal records

2.4.4 System Feature 4

2.4.4.1 Description and Priority

Eligibility Screening: Description: The app screens potential donors for eligibility based on criteria such as age, weight, medical history, and recent travel.

Priority: High

Benefit: 8

Penalty: 4

Cost: 6

Risk: 3

2.4.4.2 Stimulus/Response Sequences

Eligibility Screening:

Stimulus: Donor initiates the eligibility screening process.

Response: App presents a series of questions and prompts related to age, weight, medical history, and recent travel.

Stimulus: Donor provides answers to the eligibility questions.

Response: App evaluates the responses and determines if the donor meets the eligibility criteria, providing a notification of eligibility or ineligibility.

Functional Requirements:

User Input: The app should prompt the user to provide relevant information required for eligibility screening, such as age, weight, medical history, and recent travel.

Validation and Verification: The app should validate and verify the user's input to ensure accuracy and completeness of the provided information. The app should check if the entered age and weight meet the minimum and maximum requirements for blood donation. The app should verify the medical history and travel details against eligibility criteria and guidelines.

Eligibility Criteria Evaluation: The app should evaluate the user's input against predetermined eligibility criteria to determine if the user meets the requirements for blood donation. The app should consider factors such as age, weight, medical conditions, medications, recent surgeries, chronic illnesses, travel to high-risk areas, and potential exposure to infectious diseases.

Notification of Eligibility or Ineligibility: Based on the evaluation results, the app should notify the user of their eligibility or ineligibility for blood donation. If the user is eligible, the app should provide information on the next steps for scheduling a donation appointment. If the user is ineligible, the app should provide reasons for the ineligibility and may suggest alternative ways to contribute to the cause.

Accessibility Considerations: The app should ensure that the eligibility screening process is accessible to users with disabilities, providing appropriate accommodations such as support for screen readers, alternative input methods, or text-to-speech capabilities.

Security and Privacy: The app should handle the user's personal and medical information securely, following privacy regulations and guidelines. The app should encrypt and protect sensitive user data during transmission and storage.

Integration with Donor Profile: The eligibility screening process should integrate with the user's donor profile, updating the eligibility status and notifying the user of any changes or updates.

2.4.5 System Feature 5

2.4.5.1 Description and Priority

Donation Locator:

Description: Users can find nearby donation from his location.

Priority: Medium

Benefit: 7

Penalty: 2

Cost: 5

Risk: 2

2.4.5.2 Stimulus/Response Sequences

Donation Locator:

Stimulus: User selects the location.

Response: App accesses location services and displays nearest donors.

Stimulus: User selects a specific location or center.

Response: App provides additional details, such as address, contact information, and directions to the selected donor.

Functional:

User Location Input: The app should allow users to input their current location or enable automatic detection of their location using GPS.

Search by Location: Users should be able to search for donation centers, blood drives, or hospitals based on a specific location or within a specified radius.

Filter by Donation Type: Users should have the option to filter donation centers based on the type of donation they are interested in, such as whole blood, platelets, or plasma.

Display Donation Centers: The app should display a list of donation centers, blood drives, or hospitals based on the user's location or search criteria. Each listing should include essential information, such as the name, address, contact details, and distance from the user's location.

Detailed Information: Users should be able to view detailed information about each donation center, including operating hours, available donation types, and any special requirements or restrictions.

Contact Information: Users should be able to access contact information, such as phone numbers or email addresses, to directly communicate with donation centers or seek additional information.

Integration with External Systems: The donation locator should integrate with external systems or APIs to retrieve and update accurate and up-to-date information about donation centers, blood drives, or hospitals.

Accessibility Considerations: The app should consider accessibility features, such as support for screen readers, ensuring text legibility, and providing alternative text for images on the donation locator interface.

2.4.6 System Feature 6

2.4.6.1 Description and Priority

2.4.7 System Feature 7

2.4.7.1 Description and Priority

Data Security and Privacy:

Description: The app ensures secure storage and transmission of user data, adhering to data privacy regulations and guidelines.

Priority: High

Benefit: 9

Penalty: 3

Cost: 7

Risk: 4

2.4.7.2 Stimulus/Response Sequences

Data Security and Privacy:

Stimulus: User enters personal information during registration or profile update.

Response: App encrypts and securely stores the user's information in the database.

Stimulus: App sends or receives data over the network.

Functional Requirements:

User Authentication: The app must require users to authenticate themselves before accessing their profiles or performing any sensitive actions. Users should be able to create strong and unique passwords, and the app should enforce password complexity requirements. The app may provide additional authentication mechanisms such as biometric authentication (fingerprint, face recognition) for enhanced security.

Secure Storage of User Data: User data, including personal information and donation history, should be securely stored in a protected and encrypted database. The app should utilize appropriate encryption algorithms and protocols to safeguard sensitive data from unauthorized access or breaches.

Adequate measures should be taken to prevent data loss, including regular backups and redundancy strategies.

Secure Communication: All communication between the app and its backend servers, as well as third-party services, should be encrypted using secure protocols such as HTTPS. The app should avoid transmitting sensitive user information over unsecured channels, including public Wi-Fi networks.

Role-Based Access Control: The app should implement role-based access control to ensure that only authorized users, such as healthcare professionals, can access sensitive functionalities or view certain user information. Access privileges should be assigned based on user roles and responsibilities, and regular audits should be conducted to review and update access permissions.

Data Minimization: The app should collect and store only the necessary user data required for its intended purpose. Users should have control over their data and be able to review and modify their information within the app. The app should adhere to relevant data protection laws and regulations, such as GDPR or HIPAA, to ensure compliance with data privacy requirements.

Consent and Privacy Settings: The app should obtain explicit user consent before collecting, processing, or sharing their personal information. Users should have the ability to review and modify their privacy settings, including controlling the visibility of their profiles or donation history to other users.

Security Incident Handling: The app should have a documented process for handling security incidents, such as data breaches or unauthorized access attempts. Appropriate measures should be in place to detect, report, and respond to security incidents in a timely manner. Users should be promptly notified in case of any security incidents that may impact their data privacy.

Regular Security Updates and Maintenance: The app should undergo regular security assessments, vulnerability scans, and penetration testing to identify and address potential security vulnerabilities. Security patches and updates should be promptly applied to the app's infrastructure, server software, and libraries to mitigate known security risks.

Data Deletion and Retention:

The app should provide users with the option to delete their accounts and associated data permanently. Clear data retention policies should be defined, specifying the duration for which user data is stored and outlining the procedures for data deletion after the retention period expires.

Compliance with Data Protection Regulations: The app should comply with relevant data protection regulations and guidelines, such as GDPR, HIPAA, or local privacy laws applicable to the jurisdiction in which it operates. User consent mechanisms, data access requests, and other necessary procedures should be implemented to ensure compliance with data protection requirements.

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

Response Time: The app should have quick response times to user actions, such as navigating between screens, searching for donors, and scheduling appointments.

Response times should be within an acceptable range, typically measured in milliseconds or seconds, to ensure a smooth and efficient user experience.

Loading Time: The app should load quickly when launched, ensuring that users can access its features and functionalities without significant delays. Loading times should be optimized to minimize user frustration and improve overall app usability.

Scalability: The app should be designed to handle a growing number of users, donor records, and appointment schedules without significant degradation in performance. The system should scale horizontally or vertically to accommodate increased demand, ensuring consistent performance even during peak usage periods.

Network Efficiency: The app should efficiently utilize network resources to minimize data usage and provide a seamless experience for users, particularly in areas with limited or slow internet connectivity. Network requests and data transfers should be optimized to reduce latency and minimize the consumption of user's mobile data.

Reliability and Availability: The app should be reliable and available to users at all times, ensuring minimal downtime or disruptions. System failures, crashes, or data loss should be minimized, and appropriate backup and recovery mechanisms should be in place to ensure data integrity and system availability.

Battery Efficiency: The app should be optimized to consume minimal battery power to prolong device usage and minimize the impact on the user's device battery life. Battery-efficient coding practices, such as minimizing background processes, optimizing network requests, and reducing CPU usage, should be implemented.

Memory Management: The app should efficiently manage memory resources to prevent excessive memory usage, crashes, or slowdowns. Memory leaks and excessive memory usage should be minimized to ensure smooth app performance, especially on devices with limited RAM.

Security Performance: The app should prioritize strong security measures to protect user data and ensure secure transmission and storage of sensitive information.

Encryption algorithms, secure communication protocols, and appropriate security measures should be implemented to maintain data confidentiality and integrity.

Compatibility: The app should be compatible with a wide range of mobile devices, operating systems (e.g., Android), and screen sizes/resolutions, ensuring consistent performance across different platforms. Compatibility testing should be conducted to identify and address any performance issues specific to different device configurations.

2.5.2. Safety Requirements

User Authentication and Authorization: The app should have secure user authentication mechanisms to ensure that only authorized users can access sensitive data and perform critical actions. User passwords should be securely stored using strong encryption techniques. Two-factor authentication can be implemented to provide an additional layer of security.

Data Encryption and Protection: User data, including personal and medical information, should be encrypted both during transmission and at rest to prevent unauthorized access. Adequate measures should be taken to protect against data breaches, such as implementing robust security protocols and regular security audits.

Privacy and Confidentiality: The app should comply with relevant data privacy regulations to protect user privacy and confidentiality. User consent should be obtained for collecting, storing, and sharing their personal and medical information. Anonymization techniques can be employed to ensure the privacy of donors and recipients while still maintaining necessary records.

Error Handling and Data Validation: The app should implement proper error handling mechanisms to prevent data corruption, loss, or unintended actions. Data input should be validated to prevent malicious or incorrect data from compromising the system's integrity or affecting user safety.

Secure Communication: All communication between the app and backend servers should be encrypted using secure protocols, such as SSL/TLS, to protect sensitive data during transit. Secure APIs should be used to prevent unauthorized access or tampering of data exchanged between different system components.

Adverse Event Reporting: The app should provide a mechanism for users to report any adverse events or complications related to blood donation. Proper procedures should be in place to handle and address reported events promptly, ensuring user safety and well-being.

User Safety Education: The app should provide educational resources and guidelines on blood donation safety, including eligibility criteria, pre-donation instructions, and post-donation care. Safety warnings and precautions should be clearly communicated to users to minimize risks associated with blood donation.

Continuous Monitoring and Updates: The app should undergo regular monitoring and maintenance to identify and address potential security vulnerabilities or safety risks.

Regular updates and patches should be applied to address any discovered vulnerabilities or enhance the app's safety features.

Compliance with Regulatory Standards: The app should comply with relevant regulatory standards and guidelines specific to blood donation and healthcare, such as those provided by health authorities and blood donation organizations.

2.5.3. Security Requirements

Authentication and Access Control: The app should enforce secure user authentication mechanisms, such as password-based authentication, biometric authentication, or two-factor authentication, to ensure that only authorized users can access the app. User roles and access privileges should be defined to restrict access to sensitive features and data based on user roles (donors, recipients, healthcare professionals, administrators).

Data Encryption and Protection: User data, including personal information, medical history, and donation records, should be encrypted during transmission and storage to prevent unauthorized access or data breaches. Adequate measures, such as encryption algorithms, secure protocols (e.g., HTTPS), and secure storage practices, should be implemented to protect sensitive data.

Secure Communication: The app should use secure communication channels (e.g., Transport Layer Security) to ensure that data transmitted between the app and servers or other endpoints remains confidential and tamper-proof. Encryption should be applied to messages and notifications exchanged between users to protect the privacy of their conversations.

Secure Storage and Database Management: User data and sensitive information should be stored in a secure manner, adhering to best practices for secure database management.

Measures such as access controls, data encryption, and regular backups should be implemented to protect against data loss, unauthorized access, or data corruption.

Secure Third-Party Integrations: If the app integrates with third-party services or APIs (e.g., authentication services, notification services), proper security measures should be implemented to ensure the integrity and security of data exchanged with these services. Only

trusted and reputable third-party services should be used, and appropriate security protocols and data protection measures should be followed.

User Privacy and Consent: The app should obtain explicit user consent for collecting, storing, and using their personal information, in compliance with relevant privacy regulations (e.g., GDPR). Privacy settings and options should be provided to users, allowing them to control the visibility and sharing of their personal information within the app.

Secure Error Handling: The app should implement secure error handling mechanisms to prevent the exposure of sensitive information or system vulnerabilities through error messages or debug logs. Error messages presented to users should be informative without revealing sensitive details that could be exploited by attackers.

Regular Security Assessments and Updates: Regular security assessments, vulnerability testing, and code reviews should be conducted to identify and address potential security flaws or vulnerabilities in the app. Security updates and patches should be promptly applied to the app and its underlying software components to protect against known security vulnerabilities.

Compliance with Regulatory Standards: The app should comply with relevant data protection and privacy regulations, such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or any other applicable regional or industry-specific standards.

Secure Disposal of Data: The app should ensure secure deletion or disposal of user data from the system when it is no longer needed or requested by the user, following proper data disposal practices to prevent unauthorized access or recovery of discarded data.

2.5.4. Business Rules

Donor Eligibility:

- Donors must meet the minimum age and weight requirements to be eligible for blood donation.
- Donors must meet the health criteria and pass the eligibility screening questionnaire to ensure their safety and the quality of donated blood.
- Donors must adhere to the specified interval between donations to allow sufficient time for their bodies to recover.

Recipient Verification:

- Recipients must provide valid and verified medical documentation or prescriptions from authorized healthcare professionals to request blood donations.
- Recipients must provide accurate and up-to-date information regarding their medical condition and blood requirements.

Consent and Privacy:

- Users must provide their consent for data collection, storage, and usage according to privacy regulations and guidelines.
- User information, including personal and medical data, must be securely stored and accessed only by authorized personnel.

Donation Center Guidelines:

- Donation centers must adhere to strict protocols and guidelines for blood collection, handling, and storage to ensure the safety and quality of donated blood.
- Donation centers must maintain accurate records of donors, donations, and recipient information for traceability and transparency.
-

Communication and Etiquette:

- Users, including donors, recipients, and healthcare professionals, must engage in respectful and professional communication within the app's messaging features.
- Users should refrain from sharing inappropriate, offensive, or misleading content during interactions.

Reporting and Compliance:

- Donors, recipients, and healthcare professionals must report any issues, incidents, or concerns related to the blood donation process promptly.
- The app must comply with relevant laws, regulations, and guidelines pertaining to blood donation, privacy, data protection, and medical ethics.

Donor Health Monitoring:

- Donors should report any changes in their health status, medical conditions, or recent travel that may affect their eligibility for blood donation.

- Donation centers or healthcare professionals may request additional medical information or conduct physical examinations to ensure donor safety.

Donation Records and Tracking:

- Accurate records of donations, including dates, quantities, and donor information, must be maintained for tracking purposes, recipient verification, and donor history.

2.6. Other Requirements

Database Requirements:

- The app should have a secure and scalable database system to store user profiles, donation records, appointment details, and other relevant data.
- The database should support efficient data retrieval, search, and filtering functionalities to provide a seamless user experience.
- Data backup and recovery mechanisms should be implemented to prevent data loss in case of system failures or emergencies.

Internationalization Requirements:

- The app should support multiple languages to cater to a diverse user base.
- The user interface, including menus, labels, and messages, should be translatable and localized for different languages and cultural norms.
- Date and time formats, currency symbols, and other regional preferences should be configurable based on the user's location.

Legal and Regulatory Requirements:

- The app must comply with applicable laws and regulations related to blood donation, healthcare, data protection, and privacy in the target regions.
- User consent for data collection, storage, and usage should be obtained in accordance with privacy laws and regulations.
- Any necessary certifications or approvals from regulatory bodies should be obtained to ensure the app's compliance with industry standards.

Reuse Objectives:

- The project should identify and document components, modules, or code segments that can be reused in future development efforts or similar projects.

- Reusability guidelines and best practices should be established to maximize code reuse and minimize redundant development efforts.

Security Requirements:

- The app should implement appropriate security measures to protect user data, prevent unauthorized access, and secure data transmission over the network.
- User authentication mechanisms, such as passwords or biometric authentication, should be implemented to ensure only authorized users can access sensitive information.
- Encryption techniques should be used for storing sensitive data and during data transmission to prevent unauthorized access or data breaches.

Performance and Scalability Requirements:

- The app should be designed to handle a significant number of users concurrently without significant performance degradation.
- Response times for critical functionalities, such as searching for donors or scheduling appointments, should be optimized for a seamless user experience.
- The system should be scalable to accommodate potential increases in user base and transaction volumes without compromising performance.

Error Handling and Logging:

- The app should have robust error handling mechanisms to handle exceptions, unexpected scenarios, and edge cases gracefully.
- Error messages should be informative and user-friendly, providing clear instructions or guidance on how to proceed.
- System logs should be maintained to capture errors, warnings, and relevant information for troubleshooting and auditing purposes.

Usability and Accessibility:

- The app should have an intuitive and user-friendly interface, allowing users to navigate and perform tasks with ease.
- Accessibility features, such as support for screen readers, adjustable font sizes, and high contrast options, should be implemented to accommodate users with disabilities.

Integration Requirements:

- The app may require integration with external systems, such as healthcare databases, blood bank systems, or notification services, to enhance functionality and exchange relevant information securely.

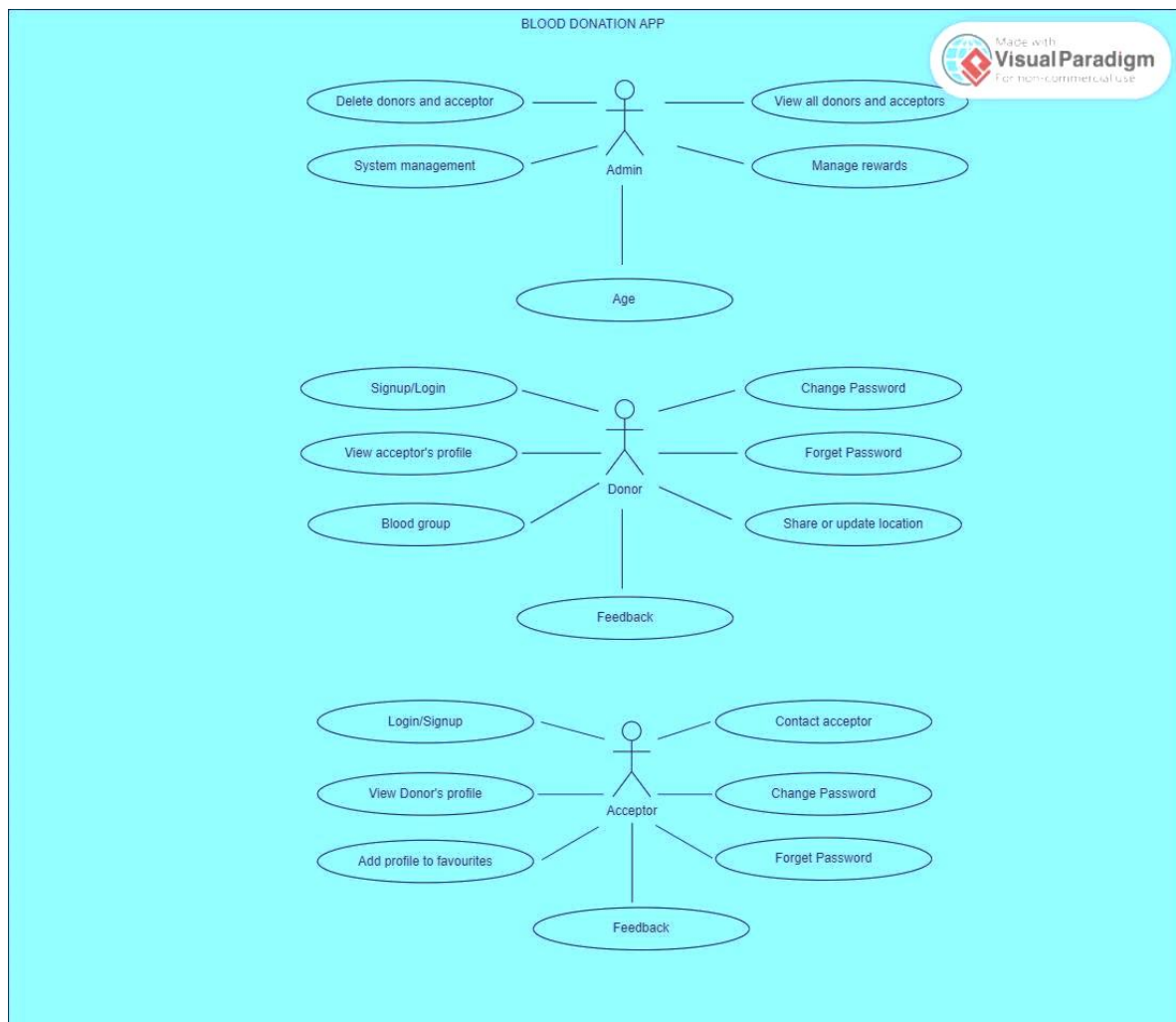
Chapter 3

Use Case Analysis

Chapter 3: System Analysis

Gathering stakeholder needs, examining use cases, creating the system architecture and user interface, connecting with existing systems, and testing all form part of the system analysis for the blood donation app. In order to create a comprehensive and user-friendly software that satisfies their demands, it seeks to comprehend the requirements and expectations of donors. The analysis stage lays the groundwork for successful implementation and aligns the app with the expected interactions and features within the ecosystem for blood donation.

3.1. Use Case Model



3.2. Use Case Descriptions

Donor:

In this module donor has Signup. User must have to register in app. After signup, user login to app then donor will get notification if acceptor wants to contact donor. After donating the blood donor get points. After getting 100 points donors will get reward from app.

Acceptor:

In this module acceptor has Signup. User must have to register in app. After signup, user login to app then donor see the lists of all available donors nearest to his location. Donor can contact any donor easily; they can share location.

Admin:

In this module, user has signup. After login admin view all the donor and acceptors. Admin can modify there profile.

Chapter 4

System Design

Chapter 4: System Design

User Interface: A user-friendly interface allowing users to easily navigate and access features such as donor/recipient registration, search for blood donation opportunities, and communication tools.

Database Management: A database to store and manage user profiles, donation records, and other relevant information securely. **Integration with External Systems:** Integration with blood bank databases, payment gateways, and SMS/email notification services to facilitate seamless coordination between donors and recipients.

Security and Privacy: Implementation of robust security measures, such as encryption protocols and user authentication, to protect sensitive user data and maintain privacy.

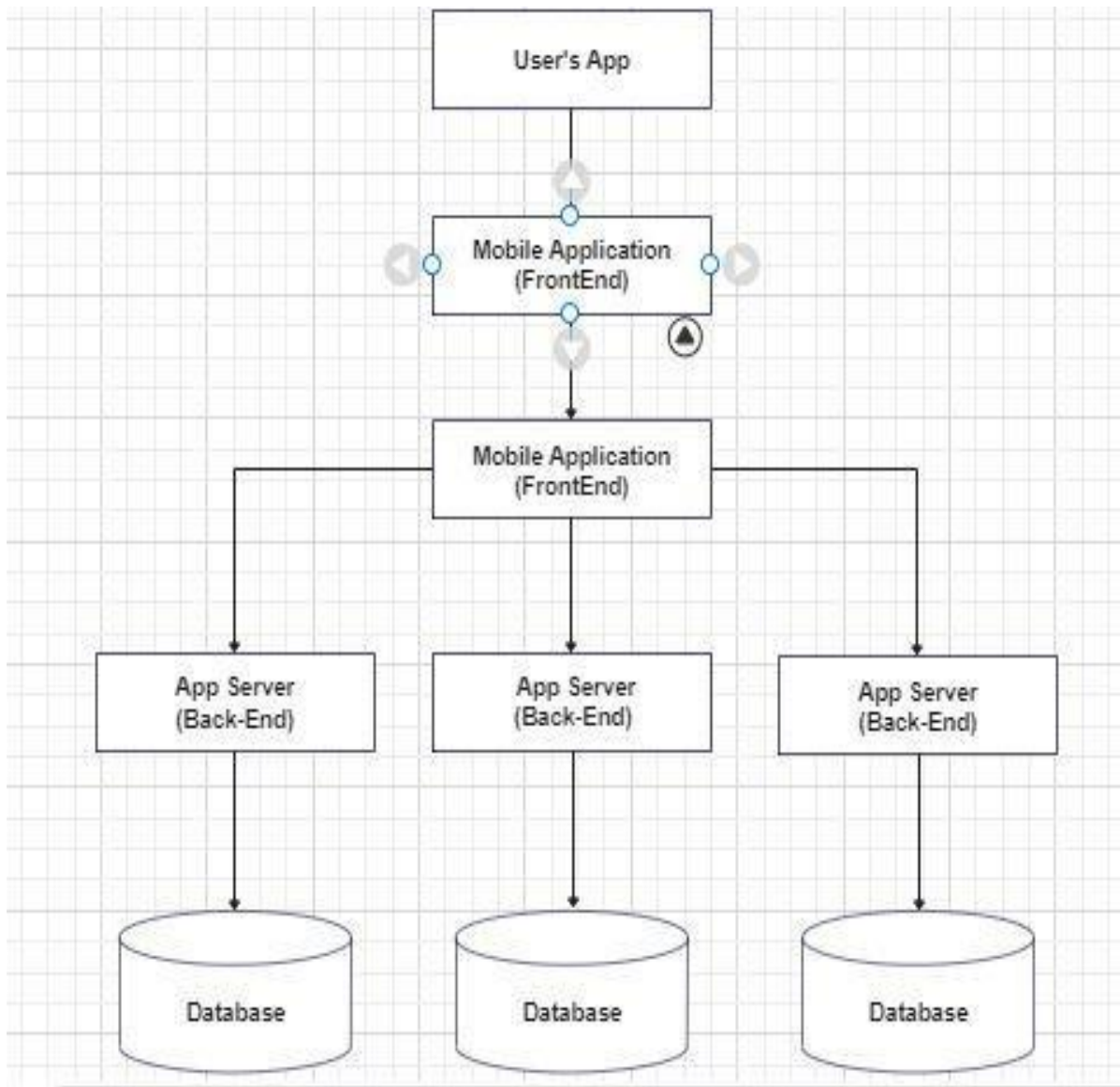
Communication Channels: Integration of messaging or notification features to enable real-time communication between donors and recipients or facilitate important updates.

Reporting and Analytics: Incorporation of reporting and analytics capabilities to track donation statistics, identify trends, and generate insights for improving the app's effectiveness.

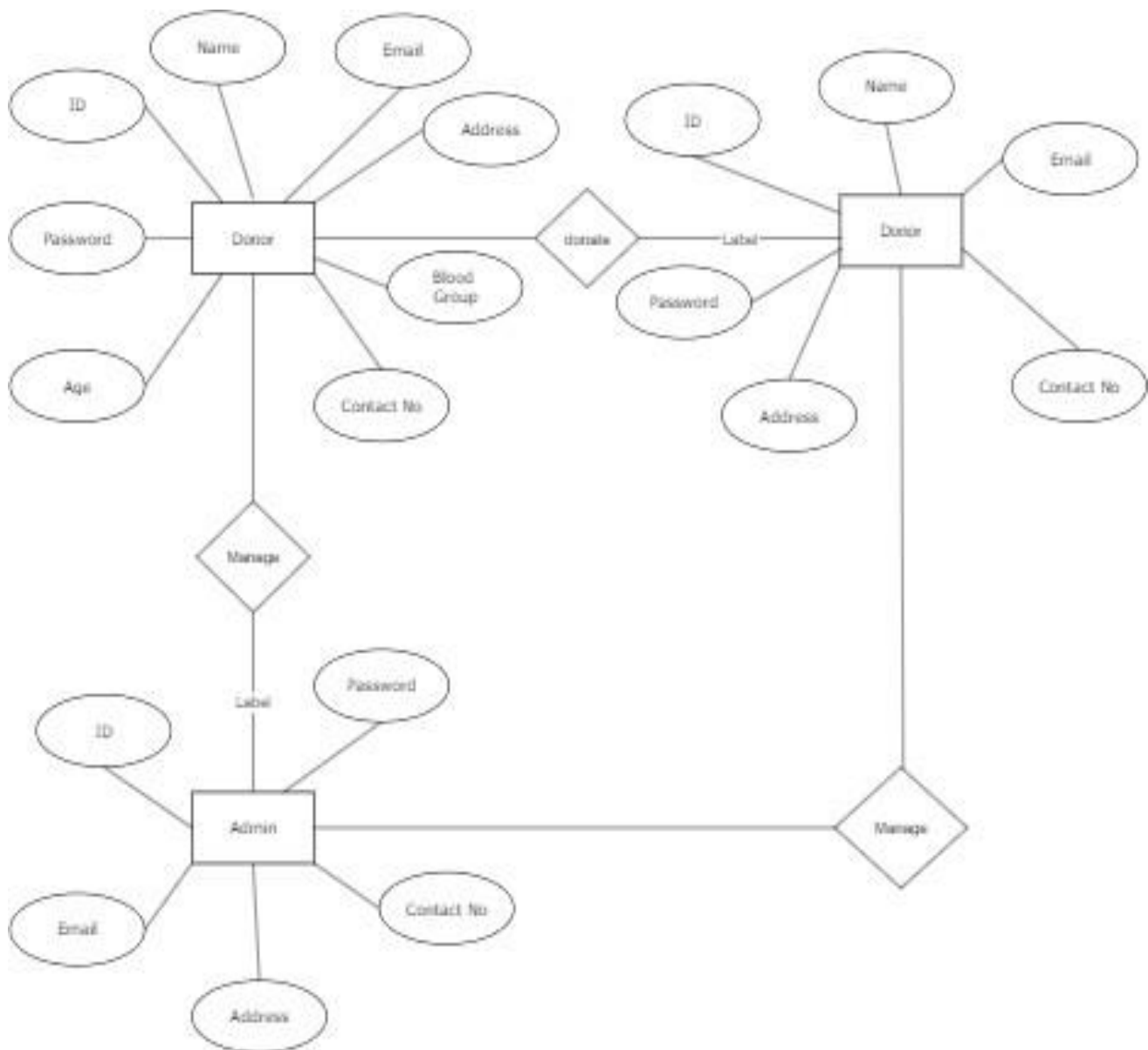
Scalability and Performance: Designing the system to handle a large number of users, accommodate future growth, and ensure optimal performance under varying loads.

Platform Compatibility: Ensuring compatibility with different platforms (e.g., web, mobile) and devices to provide a seamless experience for users across various devices and operating systems

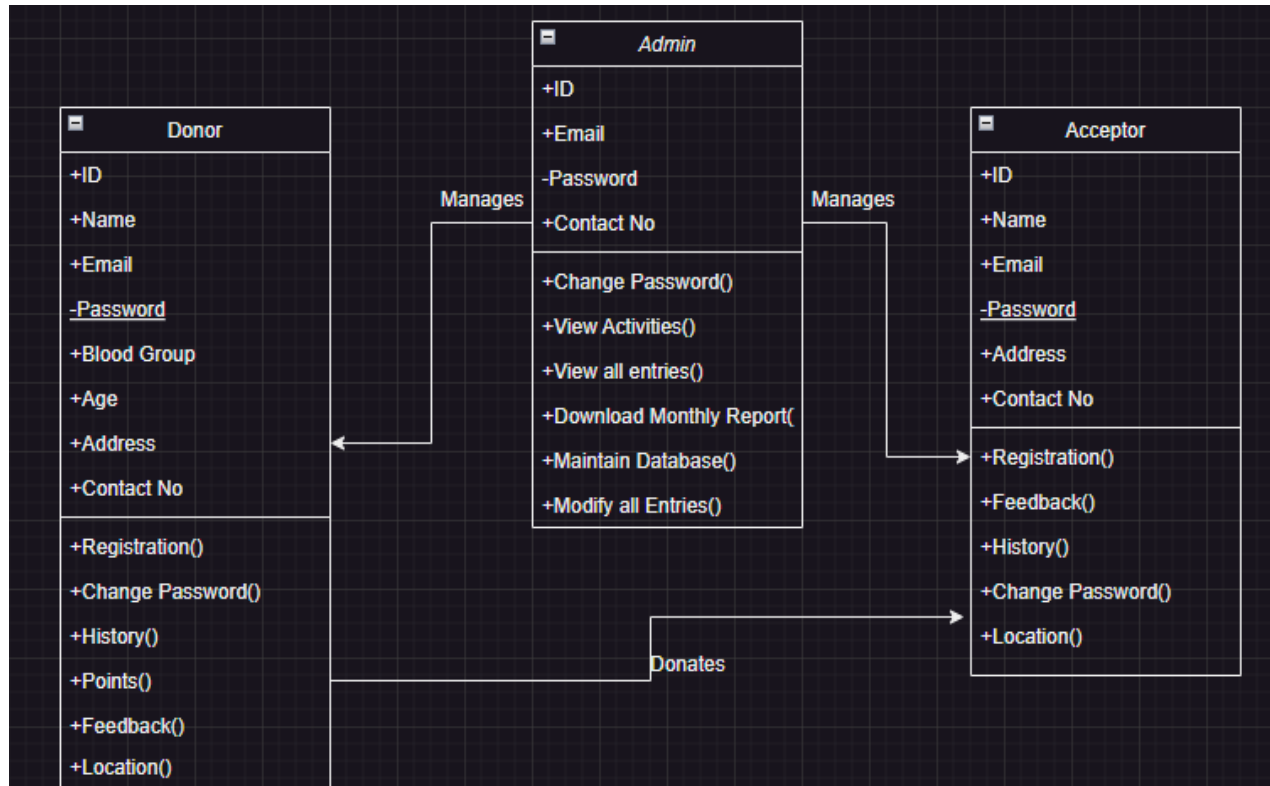
4.1. Architecture Diagram



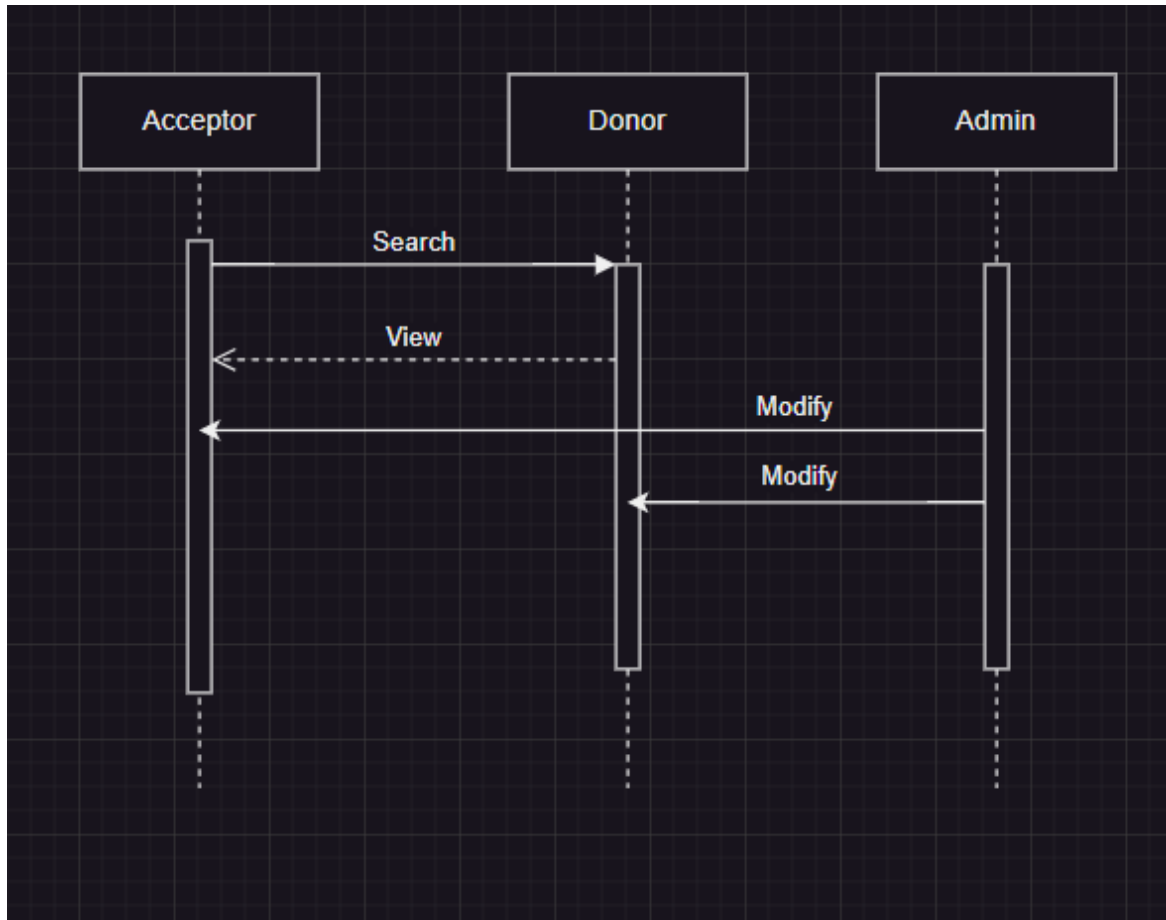
4.2. Entity Relationship Diagram with data dictionary



4.3. Class Diagram



4.4. Sequence / Collaboration Diagram



4.5. Operation contracts

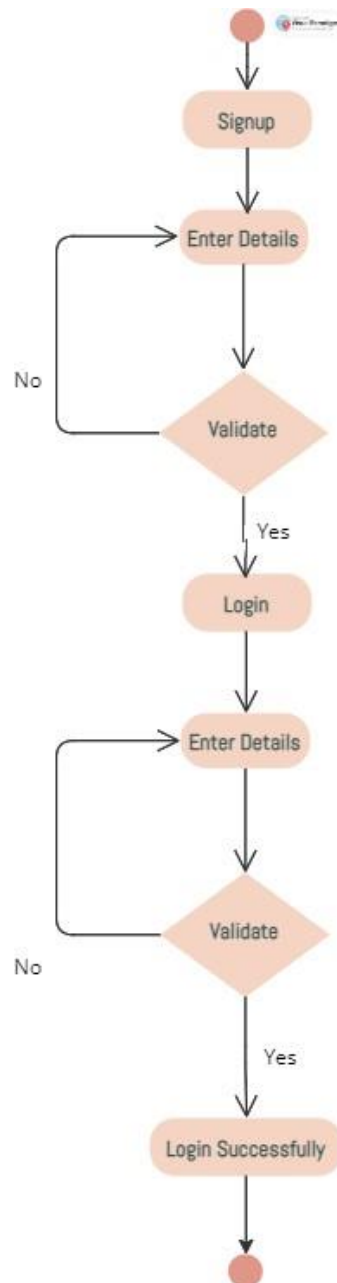
To ensure the smooth operation of a blood donation app, it is essential to have a robust contact system in place. Here are some key operational contacts that should be considered:

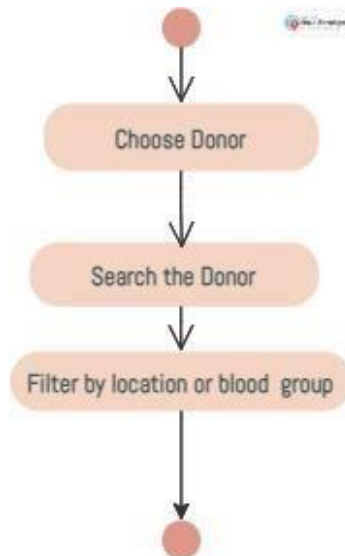
- Technical Support Team:
- Email: support@blooddonationapp.com
- Phone: +42 123456789
- Availability: 24/7

The technical support team assists users with any technical issues they may encounter while using the app, such as login problems, app crashes, or general troubleshooting. They respond promptly to user inquiries and provide guidance on navigating the app's features.

4.6. Activity Diagram

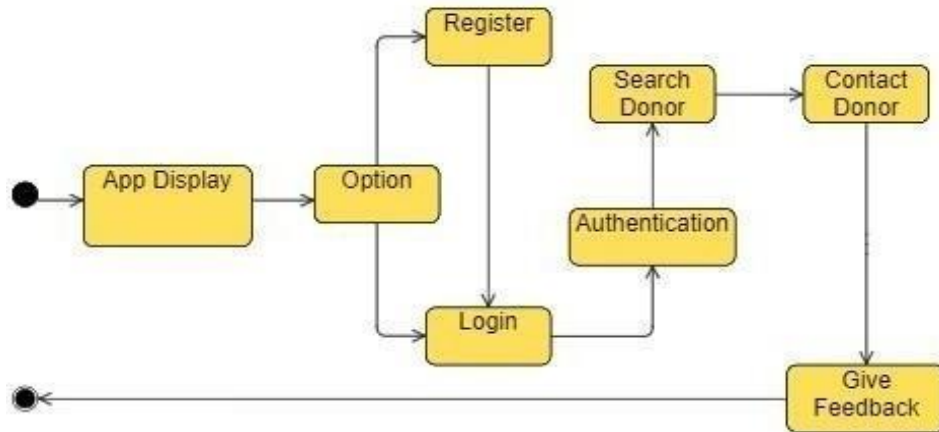
Signup / Login:



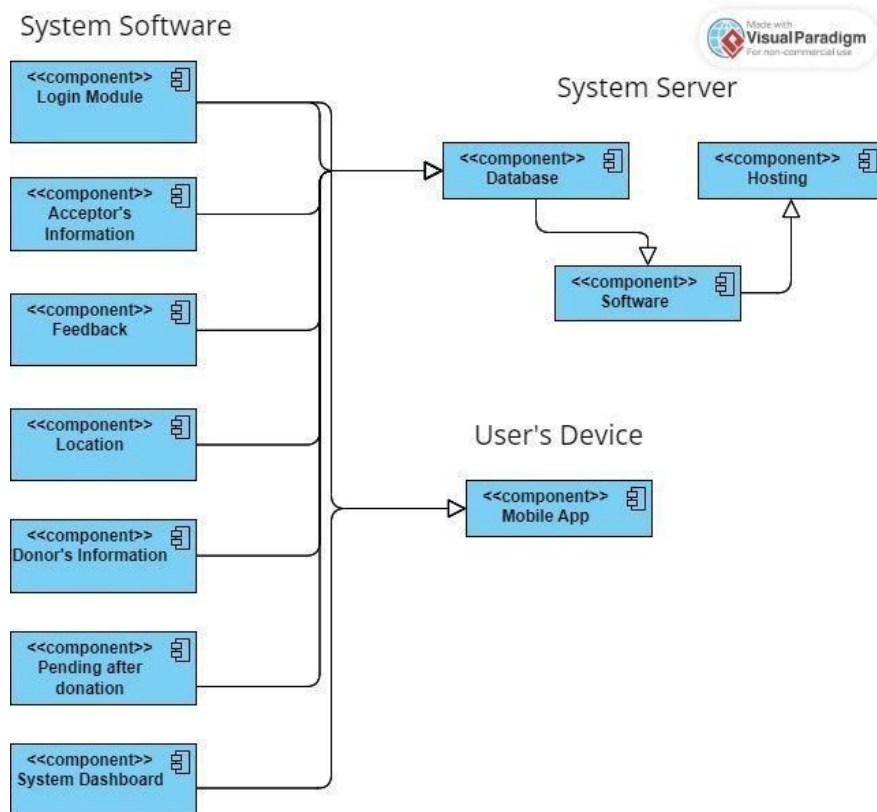
Acceptor:**Admin:**

S

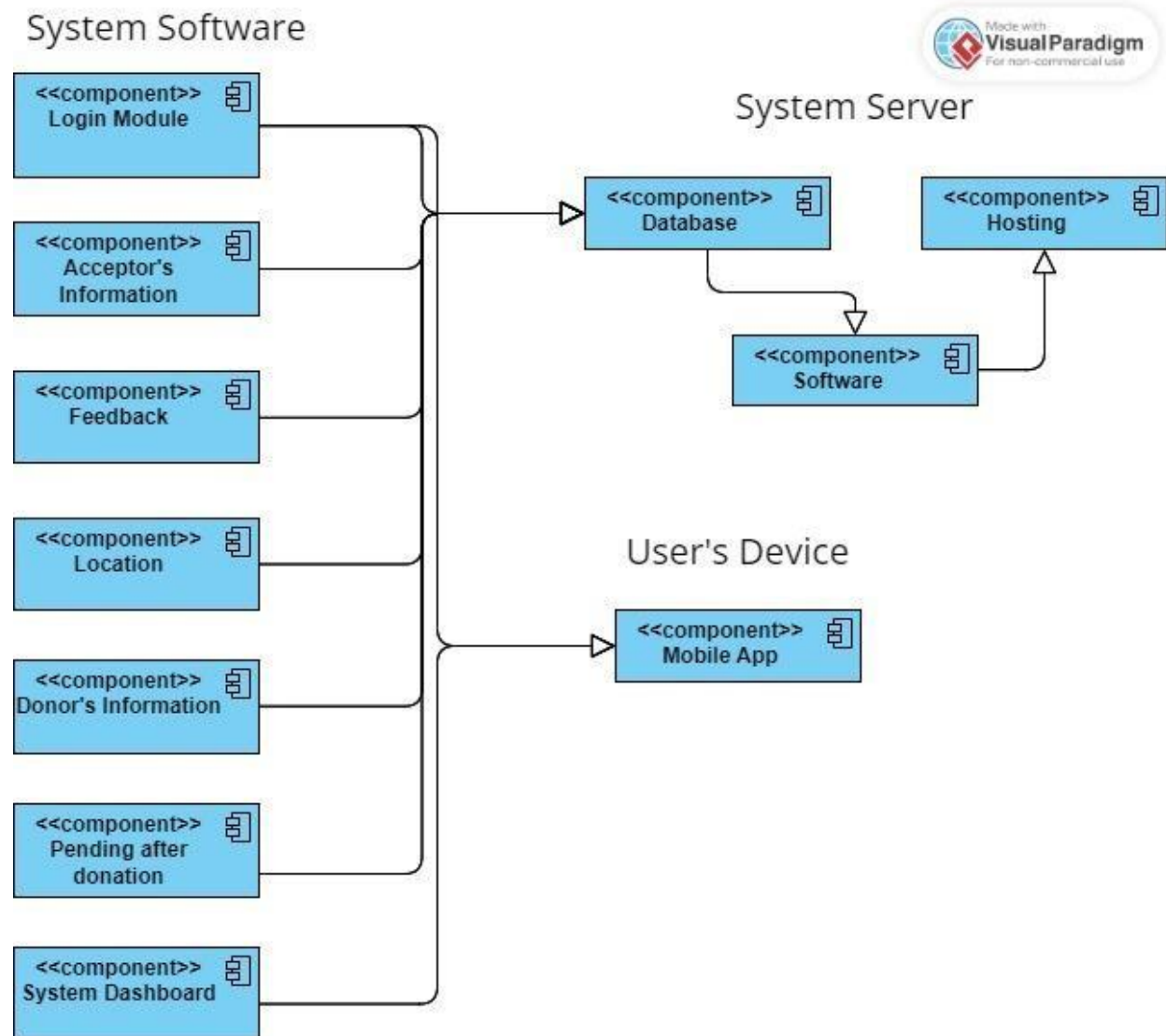
4.7. State Transition Diagram



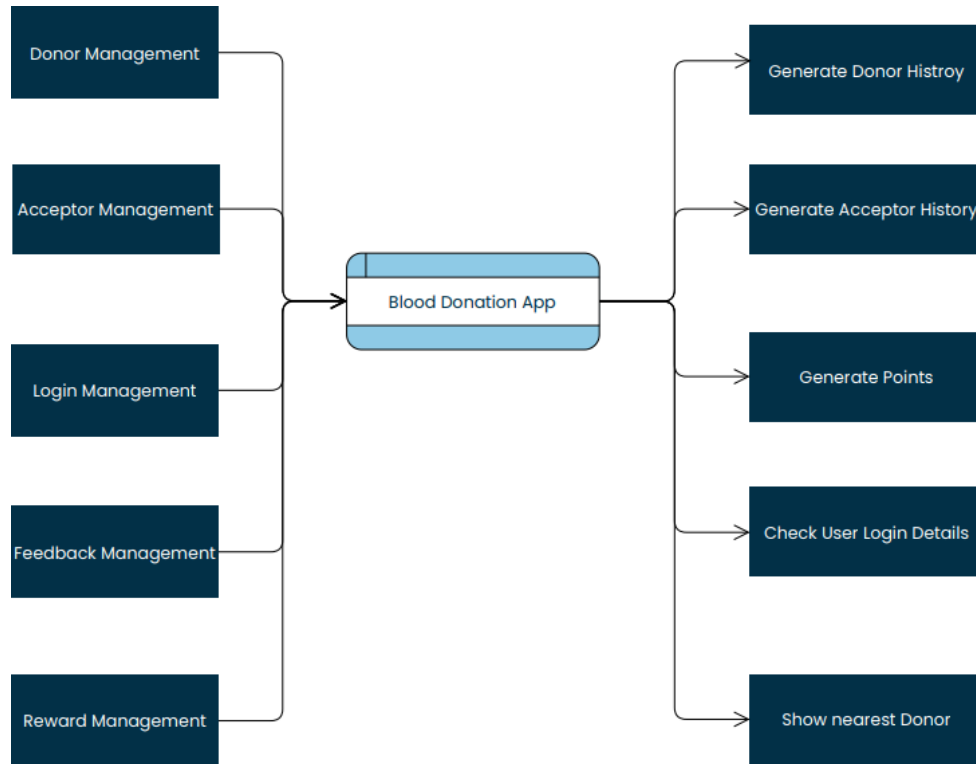
4.8. Component Diagram



4.9. Deployment Diagram



4.10. Data Flow diagram



Chapter 5

Implementation

Chapter 5: Implementation

The implementation of the blood donation app involved the development of user-friendly interfaces for donor and recipient registration, integration with blood bank databases, and secure communication channels. Robust security measures were implemented to protect user data, while reporting and analytics capabilities were incorporated to track donation statistics and generate insights for improvement. The app was designed to be compatible with multiple platforms and devices for a seamless user experience.

5.1. Components, Libraries, Web Services and stubs

Components:

User Interface (UI): Design and develop the user interface components, such as registration forms, notifications, and donation progress screens.

Database: Set up a database to store user information, donation history, and other relevant data.

Authentication: Implement authentication components to secure user accounts and protect sensitive information.

API Integration: Connect with external APIs to access services like geolocation, blood bank data, or SMS gateways.

Analytics: Integrate analytics components to track app usage, user behavior, and performance metrics.

Libraries/Frameworks:

Front-end Frameworks: Utilize frameworks like React building dynamic and responsive user interfaces.

Back-end Frameworks: Firebase

5.2. Deployment Environment

Web Server:

Apache HTTP Server to handle incoming HTTP requests and serve the application to users.

Configure the web server to handle SSL/TLS certificates for secure communication over HTTPS.

Application Server:

Set up an application server to host the backend code and handle the app's business logic and data processing.

Common choices include Apache Tomcat, Node.js, or a framework-specific server like Django's built-in development server.

Database Server:

Database is MongoDB.

Configure the database server and optimize it for performance and security.

Operating System: Android**Scalability and High Availability:**

Design your deployment environment to be scalable and highly available to handle increased traffic and ensure uninterrupted service.

Utilize load balancers, auto-scaling groups, and database replication techniques to distribute load and maintain high availability.

5.3. Tools and Techniques

Integrated Development Environment (IDE):

IDEs like Visual Studio Code, IntelliJ IDEA, or Xcode provide a comprehensive development environment with features like code editing, debugging, and project management.

Programming Languages: React (Android)**Database Management Systems (DBMS):**

MongoDB utilize database systems to store and manage user data, appointments, and other relevant information.

User Interface (UI) Design Tools:

Figma assist in designing and prototyping the user interface for a visually appealing and intuitive app.

Testing Frameworks:

Use testing frameworks like Jest, Mocha, for unit testing, integration testing, and UI testing to ensure the app's functionality and quality.

5.4. Best Practices / Coding Standards

Consistent and Readable Code:

Follow a consistent coding style throughout the project to enhance code readability and maintainability.

Use meaningful and descriptive variable and function names.

Indent code consistently and use proper spacing to improve readability.

Modular and Maintainable Code:

Organize code into modules or components with well-defined responsibilities and boundaries.

Follow the principles of modularity and separation of concerns to promote reusability and ease of maintenance.

Commenting and Documentation:

Include clear and concise comments in the code to explain complex logic or clarify the purpose of functions and classes.

Document public APIs and functions using tools like JSDoc or Doxygen to generate API documentation.

Error Handling and Logging:

Implement appropriate error handling mechanisms to gracefully handle exceptions and errors.

Use proper logging techniques to capture and log relevant information for debugging and troubleshooting purposes.

Input Validation and Sanitization:

Validate and sanitize user inputs to prevent security vulnerabilities like SQL injection or cross-site scripting (XSS) attacks.

Implement server-side validation to ensure data integrity and validate user-submitted data before processing or storing it.

Security:

Follow secure coding practices to protect sensitive user data and prevent common security vulnerabilities.

Implement secure authentication mechanisms, such as password hashing and token-based authentication.

Regularly update dependencies and libraries to address security vulnerabilities.

Testing:

Implement automated testing practices, including unit tests, integration tests, and end-to-end tests, to ensure the correctness of your code.

Write testable code by following the principles of test-driven development (TDD) or behavior-driven development (BDD).

Code Reviews:

Conduct code reviews regularly to ensure code quality, catch bugs, and enforce best practices.

Involve team members in the code review process to share knowledge and identify areas for improvement.

Version Control and Collaboration:

Use a version control system (e.g., Git) to manage source code, branches, and collaboration among team members.

Follow branching strategies like Gitflow or trunk-based development to manage feature development, bug fixes, and releases effectively.

Performance Optimization:

Optimize the app's performance by analyzing and improving critical sections of code.

Implement caching mechanisms, minimize network requests, and optimize database queries for better performance.

Accessibility:

Follow accessibility guidelines (such as WCAG 2.0 or 2.1) to ensure that the app is accessible to users with disabilities.

Use proper semantic HTML, provide alternative text for images, and ensure keyboard navigation support.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Usability studies with prospective donors are a good way to gauge how well an app for blood donation functions in terms of its user interface and navigation. Utilize surveys and feedback forms to find out how satisfied users are with the whole experience. Monitor the effectiveness of the app in matching blood donors with recipients and blood donation events. Analyze the app's responsiveness and dependability in situations that occur in real time. Examine user engagement data to see how the app affects promoting and enabling blood donations, such as frequency of usage and successful donation transactions. For the software to be refined and operate better, feedback loops and iterative testing are crucial.

6.1. Use Case Testing

User Registration:

Scenario: A new user wants to register on the app.

Test Steps:

1. Open the app and locate the registration option.
2. Enter valid information (name, contact details, blood type, etc.).
3. Verify that the registration is successful.
4. Check for any confirmation emails or messages.

Location:

Scenario: User can find the donor according to their location.

Test Steps:

1. After login and click on search tab.
2. Search the location.

Lock Profile:

Scenario: User can lock his profile manually after donating his blood.

Test Steps:

1. After donating the blood through our app or from outside just go to profile.
2. Click on the Lock profile.

Last Donated:

Scenario: User can set date of last time donated blood.

Test Steps:

1. After login and click on profile tab.
2. Set the date manually.

6.2. Equivalence partitioning**User Registration:****Valid Equivalence Classes:**

Valid Name (e.g., "Abdullah Anwer")

Valid Email Address (e.g., "abdullahanwer@gmail.com")

Valid Phone Number (e.g., "+92 3013663636")

Valid Blood Type (e.g., "A+", "B-", etc.)

Invalid Equivalence Classes:

Invalid Email Address (e.g., "invalid email")

Invalid Phone Number (e.g., "abcde")

Missing Name or Email

6.3. Boundary value analysis

User Age:

Boundary Values: 17 (just below the minimum age), 18 (minimum age), 60 (maximum age), 61 (just above the maximum age).

Blood Type:

Boundary Values: A, B, AB, O, Rh+/- (test each blood type to ensure compatibility).

Location for Donation:

Boundary Values: Test with a location in the urban, suburban, and rural areas.

Device Types:

Boundary Values: Test on the minimum and maximum supported versions of Android and iOS.

6.4. Data flow testing

User Registration:

Input: User-provided registration data (name, email, phone, blood type).

Processes: Data validation, storage in the user database.

Output: Confirmation message, user profile creation.

Testing Scenarios:

Test data validation for each input field.

Verify that the data is stored correctly in the user database.

Check if the confirmation message is displayed.

Integration with Wearable Devices:

6.5. Unit testing

User Registration:

Test: Verify that user registration functions correctly, storing valid user data in the database and providing appropriate error messages for invalid inputs.

Donor Profile Update:

Test: Confirm that updating donor profiles accurately modifies user information, validating health data, and ensuring changes reflect in the database.

Community Interaction:

Test: Confirm that community engagement like viewing phone number or calling the number.

6.6. Integration testing

User Registration and Authentication:

Test the integration between user registration and authentication modules to ensure a smooth user onboarding process.

Donor Eligibility and User Profile:

Verify the integration between the donor eligibility check and user profile modules to ensure accurate eligibility information is displayed.

Blood Donation Process and Database:

Verify the integration between the blood donation process and the database to ensure accurate recording and retrieval of donor information.

6.7. Performance testing

Load Testing:

Simulate a realistic number of concurrent users to assess how the app performs under normal conditions.

Determine the app's capacity by gradually increasing the load until performance metrics start to degrade.

Response Time Testing:

Measure the time it takes for the app to respond to user interactions under various load conditions.

Identify bottlenecks and optimize components that contribute to longer response times.

Scalability Testing:

Evaluate the app's ability to scale horizontally or vertically to accommodate an increased number of users.

Assess how well the app utilizes additional resources as the load grows.

Network Latency Testing:

Simulate different network conditions (3G, 4G, Wi-Fi) to ensure the app performs well across various connectivity scenarios.

Measure the impact of network latency on data retrieval and app responsiveness.

Reliability and Stability Testing:

Evaluate the app's stability over prolonged periods of usage to identify memory leaks or performance degradation over time.

Assess how the app recovers from unexpected failures or interruptions.

Database Performance:

Test the efficiency of database queries and transactions to ensure smooth data retrieval and storage.

Optimize database performance to handle increased load.

6.8. Stress Testing

High Concurrent User Load:

Simulate a higher-than-usual number of concurrent users to assess how the app handles increased traffic.

Gradually increase the load until the app's performance starts to degrade.

Rapid User Interactions:

Simulate scenarios where users perform rapid interactions, such as scheduling appointments or checking eligibility, to assess the app's responsiveness under intense user activity.

Sudden Traffic Spikes:

Introduce sudden spikes in user traffic to evaluate how the app scales and whether it can handle unexpected increases in demand.

Data Volume Stress:

Test the app's performance with an unusually high volume of data, such as a large number of registered users, appointments, or emergency requests.

Long Duration Testing:

Run stress tests for an extended duration to assess the app's stability and performance over time, identifying any memory leaks or degradation that may occur with prolonged usage.

Peak Load Handling:

Determine the maximum load the app can handle without crashing or exhibiting severe performance degradation.

Assess how the app recovers once the stress conditions are reduced.

Network Interruptions:

Introduce intermittent network disruptions to evaluate how the app handles connectivity issues and recovers when the network is restored.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

The Blood Donation App is a transformative mobile application designed to streamline and enhance the blood donation process, connecting donors and recipients in need. The app prioritizes user-centric design, offering a simple and secure registration process and personalized donor profiles. The achievements of the app include increased donor engagement, a reliable supply of blood during emergencies, and positive contributions to healthcare efficiency. The user can find the donor via a location and contact him.

7.2. Achievements and Improvements

Increased Donor Engagement:

The app successfully increased donor engagement through community-building features, resulting in a more committed and active user base.

Community Impact:

The app played a significant role in creating a positive impact on the community by fostering a culture of regular blood donation and raising awareness about its importance.

Educational Outreach:

Successfully educated users about the importance of regular blood donation, health and wellness, contributing to a more informed and health-conscious donor community.

7.3. Critical Review

User Interface (UI) and User Experience (UX):

Evaluate the app's design for simplicity, clarity, and ease of navigation.

Consider the user flow from registration to blood donation, ensuring a seamless and user-friendly experience.

Functionality:

Assess the core functionalities, including user registration, eligibility checks and calling or viewing phone number.

Verify that the app performs these functions accurately and efficiently.

Database Integrity:

Evaluate the app's data management, ensuring accurate storage and retrieval of donor information.

Check for data consistency and integrity across different modules.

Security Measures:

Assess the app's security protocols, especially concerning user data and sensitive information.

Ensure that the app complies with relevant privacy and data protection regulations.

Performance:

Test the app's performance under various conditions, including normal usage, high traffic, and stress scenarios.

Check for responsiveness, speed, and resource utilization.

Accessibility:

Evaluate the app's accessibility features to ensure inclusivity for users with disabilities.

Consider factors such as text size, color contrast, and voiceover capabilities.

Updates and Maintenance:

Review the app's update history to ensure that it is actively maintained.

Check for a responsive development team that addresses bugs and introduces new features as needed.

7.4. Lessons Learnt

User-Centric Design:

Lesson: Prioritize user experience and design simplicity to encourage user engagement.

Application: Ensure that the app's interface is intuitive, user-friendly, and accessible to a diverse user base.

Data Security and Privacy:

Lesson: Protecting user data is a top priority to build trust and comply with privacy regulations.

Application: Implement robust security measures to safeguard user information and adhere to data protection standards.

Performance Optimization:

Lesson: Conduct thorough performance testing to identify and address potential bottlenecks or issues under different conditions.

Application: Regularly monitor and optimize the app's performance to ensure responsiveness and reliability.

Accessibility and Inclusivity:

Lesson: Prioritize accessibility features to ensure that the app is inclusive and usable by individuals with diverse abilities.

Application: Regularly assess and enhance accessibility features, considering factors such as readability, navigation, and compatibility with assistive technologies.

Education and Awareness:

Lesson: Education and awareness campaigns are vital for increasing blood donation participation.

Application: Integrate features that provide educational content, share success stories, and raise awareness about the importance of blood donation.

7.5. Future Enhancements/Recommendations

User Interface (UI) and User Experience (UX) Improvements:

Ensure a clean, intuitive, and user-friendly design to encourage more people to use the app.

Implement a simple and quick registration process for donors.

Provide a clear and easily navigable interface for finding donation centers, scheduling appointments, and accessing information.

Real-time Notifications:

Enable push notifications to remind donors of upcoming donation appointments, provide updates on blood supply needs, and send thank-you messages after donations.

Personalized User Profiles:

Allow users to create profiles with their donation history, blood type, and any relevant health information. This can help donors track their contributions and receive targeted donation requests.

Social Media Integration:

Enable users to share their donation status and achievements on social media platforms to encourage others to donate and raise awareness about blood donation needs.

Community Engagement Features:

Create a community or forum within the app where donors can share their experiences, support each other, and discuss blood donation-related topics.

Emergency Alerts and Updates:

Develop a feature that sends emergency alerts to nearby donors in the event of a shortage or urgent need for a specific blood type.

Integration with Health Records:

Explore partnerships or integrations with health record systems to allow donors to securely share relevant health information with donation centers.

Appendices

Appendix A: User Manual

The appendix of the blood donation app user manual contains additional resources and reference materials to enhance user understanding. It may include FAQs, troubleshooting guides, glossaries, or detailed technical information. Users can refer to this section for in-depth insights, addressing common queries and providing supplementary information for a comprehensive app experience.

Appendix A: Appendix Title **Donate Blood, Save Lives**

.

A.1. Empowering communities through a Blood Source App

A.1.1. Bridging Gaps in Blood Donation Accessibility

A.1.1.1 User Registration and Profiles

Appendix B: Administrator Manual

The Administrator Manual for Blood Source provides comprehensive guidance for managing and overseeing the blood donation app. It covers user management and real-time monitoring of blood inventory. The manual outlines procedures for handling ensuring data security and optimizing app performance. Administrators will find step-by-step instructions for efficiently coordinating donation drives, reviewing analytics, and maintaining a seamless experience for both donors and recipients. This manual is an essential resource for empowering administrators to uphold the app's mission of saving lives through effective blood donation management.

B.1. System Overview for Administrators

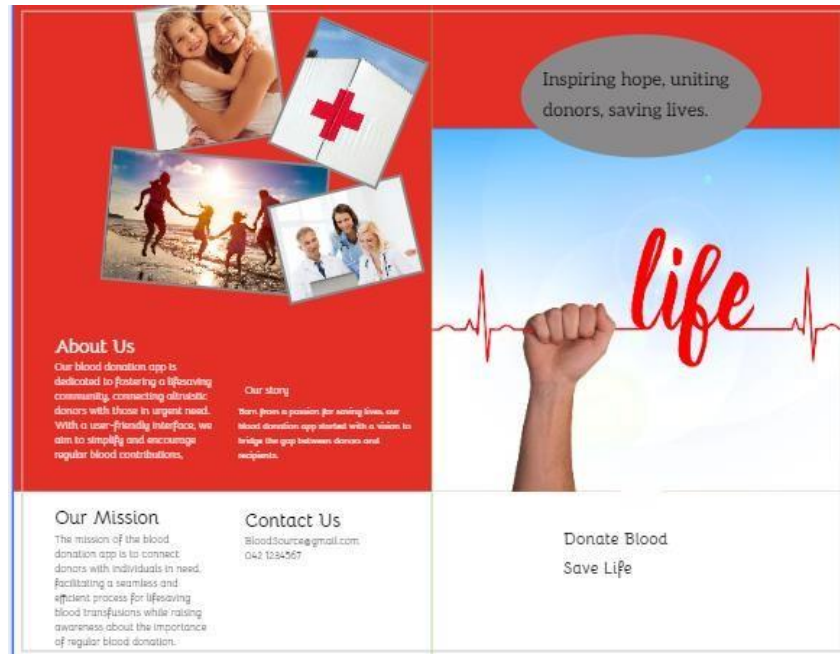
B.1.1. Blood Inventory control

B.1.1.1 Resolving User Issues and Support

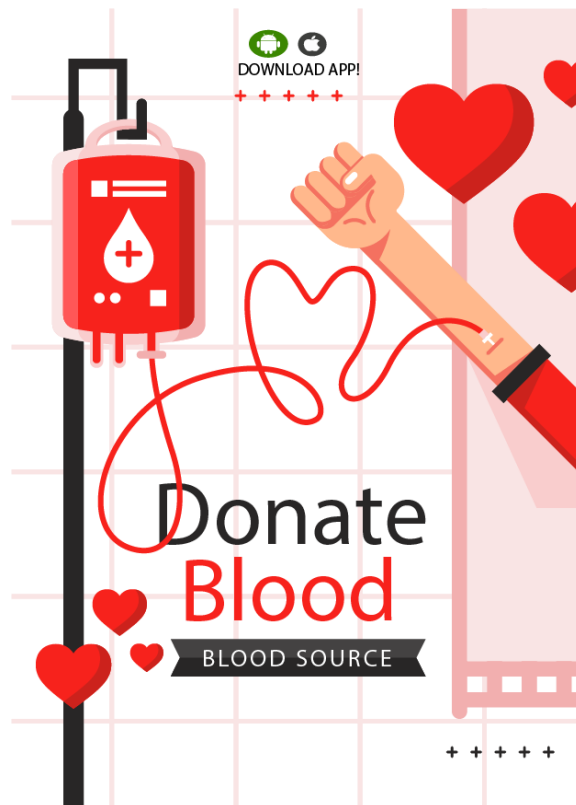
Appendix C: Information / Promotional Material

Introducing BloodSource – The app that connects heroes with hearts! Save lives effortlessly by donating blood through our user-friendly platform. Instantly find nearby blood drives, track your donations, and join a community dedicated to making a difference. With real-time alerts, personalized profiles. Download BloodSource today and become a part of a lifesaving movement. Your donation matters – join us in creating a healthier, happier world!"

C.1. Broacher



C.2. Flyer



C.3. Standee



C.4. Banner



Reference and Bibliography

Reference and Bibliography

- [1] Ouhbi, S., Fernández-Alemán, J. L., Toval, A., Idri, A., & Pozo, J. R. (2015). Free blood donation mobile applications. *Journal of medical systems*, 39, 1-20.
- [2] Yuan, S., Chang, S., Uyeno, K., Almquist, G., & Wang, S. (2016). Blood donation mobile applications: are donors ready?. *Transfusion*, 56(3), 614-621.
- [3] Ouhbi, S., Fernández-Alemán, J. L., Pozo, J. R., Bajta, M. E., Toval, A., & Idri, A. (2015). Compliance of blood donation apps with mobile OS usability guidelines. *Journal of medical systems*, 39, 1-21.
- [4] Li, L., Valero, M., Keyser, R., Ukuku, A. M., & Zheng, D. (2023). Mobile applications for encouraging blood donation: A systematic review and case study. *Digital Health*, 9, 20552076231203603.
- [5] Mostafa, A. M., Youssef, A. E., & Alshorbagy, G. (2014). A framework for a smart social blood donation system based on mobile cloud computing. *arXiv preprint arXiv:1412.7276*.
- [6] Nabil, M., Ihab, R., El Masry, H., Said, S., & Youssef, S. (2020). A web-based blood donation and medical monitoring system integrating cloud services and mobile application. In *Journal of Physics: Conference Series* (Vol. 1447, No. 1, p. 012001). IOP Publishing.
- [7] Potgieter, A., & Rensleigh, C. (2022). Blood donor app usage behaviour and perceptions: Considerations for a blood donation app. *South African journal of information management*, 24(1), 1-7.
- [8] Meiappane, A., Logavignesh, K., Prasanna, R., & Sakthivel, T. (2019, March). D'WORLD: Blood donation app using Android. In *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)* (pp. 1-5). IEEE.
- [9] Sundarde, C., Jain, S., & Shaikh, E. (2015). Advancement of Blood Donation Application.
- [10] Batis, A. A., & Albarrak, A. (2021). Preferences and features of a blood donation smartphone app: a multicenter mixed-methods study in Riyadh, Saudi Arabia. *Computer Methods and Programs in Biomedicine Update*, 1, 100005.

- [11] Müller, H. M., & Reuter-Oppermann, M. (2022, May). A design science approach to blood donation apps. In *International Conference on Design Science Research in Information Systems and Technology* (pp. 221-232). Cham: Springer International Publishing.
- [12] Diba, S. N. (2018). *Blood donation application with implementation of machine learning* (Doctoral dissertation, BRAC University).
- [13] Saad, A. S. A., & Christensen, L. R. (2019). Improving and supporting blood donation practices in khartoum, sudan blood banks through android mobile app and web application system. *International Journal of Computer Science Trends and Technology*, 7(2), 83-92.
- [14] Pande, S., Mate, S., Mawal, P., Jambulkar, A., & More, N. S. (2018). E-blood bank application using cloud computing. *International Research Journal of Engineering and Technology (IRJET)*, 5(2).
- [15] Sajith, T., Santhosh, D., Senthalan, S., & Vasanth, V. (2021). Blut: Blood Donation Application.
- [16] Mandale, M., Jagtap, P., Mhaske, P., Vidhate, S., & Patil, S. S. (2017). Implementation of blood donation application using android smartphone. *Int. J. Adv. Res. Ideas Innovations Technol*, 3(6).
- [17] Muhammad, G., Asif, H., Abbas, F., Memon, I., & Fazal, H. (2020). An ERP Based Blood Donation Management System for Hospital and Donor. *Sukkur IBA Journal of Emerging Technologies*, 3(1), 44-54.
- [18] Abu-Naser, S. S., Zaqout, I. S., & Abumughessib, R. K. (2016). Design and development of mobile blood donor tracker.
- [19] Saitong, P. (2022). Designing, Developing, and Efficiency Evaluation of a Smartphone Application for Blood Donation. *Ingénierie des Systèmes d'Information*, 27(4).