CATHOLIC UNIVERSITY INSTITUTE OF BUEA

(CUIB)

The Growth Mindset Entrepreneurial University

BLOOD DONATION MOBILE APPLICATION WITH PROXIMITY CHECK

A SENIOR YEAR PROJECT REPORT

By

NGAKO TCHAWE ANDY MANUELA

19SI-004362

SUBMITTED TO THE SCHOOL OF INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc.) DEGREE IN SOFTWARE ENGINEERING

SUPERVISOR

Dr. MOKOM FELICITAS

JULY, 2023

DECLARATION

I declare that this project on "**BLOOD DONATION MOBILE APPLICATION WITH PROXIMITY CHECK**" is an original work done by me under the supervision of "**Dr. MOKOM FELICITAS"** of the School of Information Technology at the Catholic University Institute of Buea.

CERTIFICATION

This is to certify that, this project titled **"BLOOD DONATION MOBILE APPLICATION WITH PROXIMITY CHECK"** was carried out by **NGAKO TCHAWE ANDY MANUELA,** registration number **19SI-004362**, an undergraduate senior year student in the School of Information Technology of the Department of Software Engineering, Catholic University Institute of Buea.

…… Date……

NGAKO TCHAWE ANDY MANUELA

Student

This Certification is confirmed by:

……. Date……

Dr. MOKOM FELICITAS

Supervisor

DEDICATION

I dedicate this project to the Almighty God, my lovely mother who has always guided me and my father who has always stood beside me.

ACKNOWLEDGEMENTS

I am overwhelmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

Above all to the great almighty **GOD**, father of heaven and earth, the author of knowledge and wisdom, through which everything happens. I thank the Lord once more for his support, strength, guidance and endless love.

I wish to thank the **Catholic University Institute of Buea, CUIB,** for the educational and moral growth. I wish to express my special thanks of gratitude to my Dean and supervisor, **Dr. Mokom Felicitas** as well as, all the SIT Faculties that helped me gain knowledge in this IT field.

To my parents, **Mr. NGAKO JEAN** and **Mme. NGAKO BEATRICE** for their endless support, kind and understanding, for their financial and moral assistance.

To all relatives, friends and others who in one way or another shared their support, either morally, financially and physically, thank you.

ABSTRACT

Blood transfusions are essential for many medical procedures, but current supplies of blood are insufficient to meet the needs of all patients. With blood banks running low and donations continuing to drop-off across the country, many people lose their lives, time and even money to find compatible blood donors for urgent transfusions. This research study aims at developing a mobile application that puts the power to save lives in the palm of our hands, making blood donation easier than ever. It notifies users about those who need blood donors and propose specific donors according to the proximity to the one requesting. This app was developed following the iterative waterfall model that allows for correction of errors committed during a phase and using technologies such as, React Native and Express js for both client and server side respectively and Dialogflow for chatbot development. This research project application was developed with functionalities that permits users to search for donors, contact donors and request for blood donors. This research project serve as an intermediary between donors and patients or receivers and its implications could be used to promote blood donation over the country.

Keyword: Blood Donation, Blood Transfusion, React Native, Express js.

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF TABLES

TABLE OF ABBREVIATION

Abbreviation	Meaning
Android	An open source platform designed for mobile devices
API	Application Programming Interface
CPU	Central Processing Unit
HTTP	Hypertext Transfer Protocol
JS	JavaScript
JSON	JavaScript Object Notation
MySQL	Open source relational database management system
RESTFUL	Representational State Transfer technology
SDLC	Software Development Life Cycle
SQL	Structured Query Language
SRS	Software Requirement Specification
UML	Unified Modeling Language
URL	Uniform Resource Locators
WHO	World Health Organisation
XAMPP	Cross Platform Apache MySQL PHP and PERL
XML	Extensible Markup Language

CHAPTER ONE
INTRODUCTION

1.1. Background of study

This project is based on building a Blood Donation mobile application that will serve as an intermediate between donors and patients. A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile apps are sometimes categorized according to whether they are web-based or native apps, which are created specifically for a given platform. In recent years, the potential for mobile and communications technology to drive blood donation has begun to be recognized. Usually, when a person requires a particular type of blood and that type is not available in the hospital, the persons send messages to family members, relatives, and friends if they can donate. As such, it is time-consuming and laborious, and may risk the life of the person who needs blood urgently. Since blood cannot be stored for an indefinite amount of time, a constant source of blood donation is required. According to the World Health Organization, the sustainable approach toward ensuring prompt and effective access to the sources of safe blood and blood products is to develop a nationwide organized blood transfusion program focused on voluntary non remunerated blood donors. *(Blood Safety and Availability, 2023*). Recruiting voluntary blood donors remains one of the significant challenges for any blood transfusion service.

Blood transfusions are essential for many medical procedures, but current supplies of blood are insufficient to meet the needs of all patients. Blood donation has enabled effective management of some pathologies, allowed safe surgeries, and reduced maternal mortality in complicated deliveries. Unfortunately, the life-saving practice is currently witnessing a downward trend. The need for full blood banks in hospitals remains high, however collecting the life-saving liquid is rather challenging. Notwithstanding, blood donation has been affected by many factors over the years. Sensitization has become difficult because of social distancing constraints. The absence of the right information also affects blood

donation. Cameroonians are not well educated on blood donation and why it is important. Security problems have made blood collection more difficult and there are no regular blood donors anymore. These reasons account for the fact that, I decided to work on this research topic.

1.2. Problem Statement

With blood banks running low and donations continuing to drop-off across the country, many people lose their lives, time and even money to find compatible blood donors for urgent transfusions. Blood donation should be an everyday plebiscite in Cameroon to the benefit of present and upcoming generations, for every blood donation is a gift of life.

This application puts the power to save lives in the palm of our hands, making blood donation easier than ever. It notifies users about those who need blood according to the proximity to the one requesting, so they can donate to those who are in urgent need of blood thereby saving lives.

1.3. Research Questions

- ✓ How can the blood donation procedure in Cameroon be simplified?
- ✓ What is the main issue in the current blood donation system?
- ✓ Can a user gets patients' requests for blood directly without logging into the application?
- ✓ Is it possible to find donors that are closest to the patient's location?
- ✓ What tools do I need to carry out these tasks?
- ✓ What methodology will be suitable for the development of this project?
- ✓ What is the development process of most mobile apps?

1.4. Objectives of Study

The main objective of this study is to ease the blood donation process.

- ✓ This project is aimed at developing an application that will enable patients to post blood requests and receive potential donors' feedbacks according to their proximity and proposed blood group.
- ✓ This application should be able to notify those potential donors about urgent blood request without them being necessarily logged in into the system.
- ✓ These users should be able to view their donation history, as well as modify their user information.
- ✓ A chatbot that will enable users to get adequate information about blood donation conditions to be respected and campaigns available.

1.5. Scope and Limitations of Study

This project allows users to register into the application, donate blood or request for blood donors within Cameroon. This research study does not cover the actual blood collection activity, and actual blood transfusion operation. Blood donors and patients or recipients of blood donation are our system users.

1.6. Significance of Study

Considering the rate at which patients lose their lives, time, energy and money due to unavailability of blood donors at a particular time and place, this application will help them in reducing and even eradicating the stress of going through such a process by acting as an intermediary between the patients and the potential donors.

Android-based blood donation application is a solution to connect the requester and donor at any time and from any place. The Blood donation mobile application is an android-based blood donation utility that maintains the information of blood donor volunteers. In

instances of an emergency, the request can transmit a request to all eligible donors for donation. It is also a voluntary blood donation as a requesting applicant that is the superior attribute of our submission. The requester can transmit the request to the registered users along with an emergency sign for the blood needed, and a notification will be transmitted to all voluntary donors of blood. When a volunteer confirms the donation of blood, it is recognised as a donor. This project is meant to offer data on the requested blood. The software program assists the requester to deliver the request at some degree within the maintained blood donation network and updates the recipient who's inclined to donate the requested blood at the same time.

CHAPTER 2
LITERATURE REVIEW

This chapter analyzes the existing literature review on blood donation systems. Literature review provides an overview of current knowledge, allowing the identification of relevant theories, methods, and gaps in the existing research that can later be applied to a paper, thesis, or dissertation topic. Blood Donation systems help to provide a direct link between the donor and the recipient in order to facilitate the blood donation process.

2.1.Definition of key words

- ✓ ***Blood Transfusion:*** is a routine medical procedure in which donated blood is provided to you through a narrow tube placed within a vein in your arm. *(Mayo Clinic Staff, 2022*).
- ✓ ***Blood Donation:*** occurs when a person voluntarily has blood drawn and used for transfusion and/or made into biopharmaceutical medications.
- ✓ ***Blood Donation System:*** a system used to collect data about donors interested in donating blood or individuals who require it. It enables individuals who want to donate the blood to help the needy. *(Shreya et al, 2020).*
- ✓ ***Blood:*** Blood is a specialized body fluid. It has four main components: plasma, red blood cells, white blood cells, and platelets.
- ✓ ***Mobile Application:*** a software application developed specifically for use on small, wireless computing devices, such as smartphones.
- ✓ ***Framework:*** A framework, or software framework, is a platform that provides a foundation for developing software applications. It uses shared resources such as libraries, image files, and reference documents and puts them together in one package. That package can be modified to suit the specific needs of the project. *(O'Grady, 2022)*.

2.2.Historical Evolution of Blood Donation Systems and Related Works

2.2.1. Historical Evolution

Blood transfusion is a critical element of health care. It contributes to saving lives in both ordinary and emergency situations. Furthermore, it dramatically improves the anticipation and excellent lifestyles of patients. The delivery of blood is important to meet the stress of older populations. *(Voluntary blood donation: foundation of a safe and sufficient blood supply, 2010).* Blood donation in health care is a complicated procedure that takes days or months to select a donor that has the same blood type as the recipient. Making a blood donation at a vital location today means being welcomed into a clean, modern facility, screened thoroughly to ensure your safety and blood recipients' safety, made comfortable in a donor chair, and hooked up to sterile blood donation equipment, but like many discoveries that occurred centuries before modern medicine was established, the early history of blood transfusion is really frustrating.

The first blood transfusion on record was performed in ***1665*** between dogs. Two years later, successful transfusions from lambs to humans were reported in Europe, but that method eventually was banned because of transfusion reactions. In ***1818***, British obstetrician ***James Blundell*** figured out how to provide his female patient with blood from her husband using a syringe. He performed 10 transfusions in his career, half of which helped his patients. The discovery of three different human blood groups A, B and C (later changed to O) by Austrian physician ***Karl Landsteiner*** in ***1900*** was a huge leap forward for blood transfusion medicine and won him the Nobel Prize. His colleagues discovered group AB a couple of years later. This led to numerous discoveries that paved the way for modern blood transfusion, including: cross matching blood types to improve safety, type O as the universal donor and type AB as the universal recipient, how to indirectly transfuse blood (instead of person to person) and various ways to preserve blood. *(Highlights of Transfusion Medicine History, 2023).*

It is an African American physician ***Charles Drew***, however, who is credited as the father of the modern blood bank. Early blood banks did not have standardized ways of collecting, testing, preserving and handling blood.

Blood transfusions were first reported in Africa (sub-Saharan and excluding South Africa) in the early 1920s, and organized transfusion practices were established before the Second World War. Blood transfusion grew rapidly after ***1945***, along with the construction of new hospitals and expanded health services in Africa. The growth of blood banks continued, as did innovations in the field, in leaps and bounds through the ***1940s*** and ***50s*** and beyond. Notable accomplishments include the invention of the blood bag instead of glass bottles to collect and store blood, and apheresis procedures for collecting different blood components. *(Schneider, 2013).*

In today's world a number of online blood bank databases, blood donation management system, blood donation online web and mobile applications are available to make the donation experience more convenient.

2.2.2. Related Works

- ✓ **American Red Cross Blood Donor Application:**

This mobile application was developed by The ***US American Red Cross*** in an effort to combat the barriers that potential and existing donors were facing. Users of this application can find their nearest blood drive, arrange, reschedule and cancel appointments with the click of a button, earn badges and share their achievements with friends and family. Geo-targeted blood alerts let donors know if their blood type is needed within their community. Donors can track their blood all the way to the hospital it is being used in and be alerted when their blood is used to save a life. However, there is no direct communication between the donor and that clinic in need of a specific blood type (*Blood Donor App, 2023*). As a result, this app is more beneficial for donors but not for clinics to find needed blood type directly and promptly.

✓ **BLOODR: blood donor and requester mobile application**

This research project was done by ***Vamsi Krishna Tatikonda*** and ***Hosam El-Ocla*** submitted to the Department of Computer Science, Lakehead University, Thunder Bay, Canada. BLOODR application provides a reliable platform to connect local blood donors with patients. BLOODR creates a communication channel through authenticated clinics whenever a patient needs blood donation. Technologies used include: Ruby programming language (simply known as Ruby) along with JavaScript and PostgreSQL for database (*Tatikonda VK & El-Ocla H, 2017*). This research supports the fact that, it is difficult to find suitable donor within a limited group of people in a given time. In addition, there is no guarantee that blood banks will have compatible blood group in stock.

✓ **Lifeblood Donor Application**

This application was developed by the ***Australian Red Cross.*** With this mobile application, one can register as a blood donor with Lifeblood, book and manage donations, then sync with your calendar so you remember, find your nearest Lifeblood donor centre with opening hours and directions, watch your donation tally grow, check when you can give life again with the donation countdown, see whether you're eligible to donate and find the answers to frequently asked questions and finally monitor your health stats (like your blood pressure). (*Lifeblood*, 2021)

2.3.Conclusion

Despite the fact that, more and more blood donation systems are been built none of these systems offers a reliable and practicable solution to help improve the blood donation process in Africa and in Cameroonian in particular. This is a major drawback particularly in cases where there is an urgent need of blood. This work aims at creating a blood donation mobile application that fits the needs of patients needing urgent blood donors in Cameroon's hospitals and reduce the rate of mortality.

CHAPTER 3

RESEARCH METHODOLOGY

Research methodology is a way of explaining how a researcher intends to carry out their research. It is a logical and systematic plan to resolve a research problem. A methodology details a researcher's approach to the research to ensure reliable, valid results that address their aims and objectives. It encompasses what data they're going to collect and where from, as well as how it's being collected and analyzed. The Software Development Life Cycle (SDLC) provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use. This section describes the chosen methodology as well as the tools and techniques used to achieve the various objectives of this blood donation system.

3.1. Iterative Waterfall Model

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version, if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration. The Iterative Model allows the accessing of earlier phases, in which the variations are made respectively. In this model, the overall project is divided into various phases, each executed using the traditional Waterfall method. Dividing larger projects into smaller identifiable phase's results in a smaller scope of work for each phase, and the end deliverable of each phase can be reviewed and improved, if necessary, before moving to the next phase.

The feedback paths allow for correction of the errors committed during a phase, when these are detected in a later phase. If during testing a design error is identified, then the feedback path allows the design to be reworked and the changes to be reflected in the design documents. However, it is important to note that there is no feedback path to the feasibility

stage. This means that the feasibility study errors cannot be corrected. Due to these reasons, I decided to implement my solution using this model.

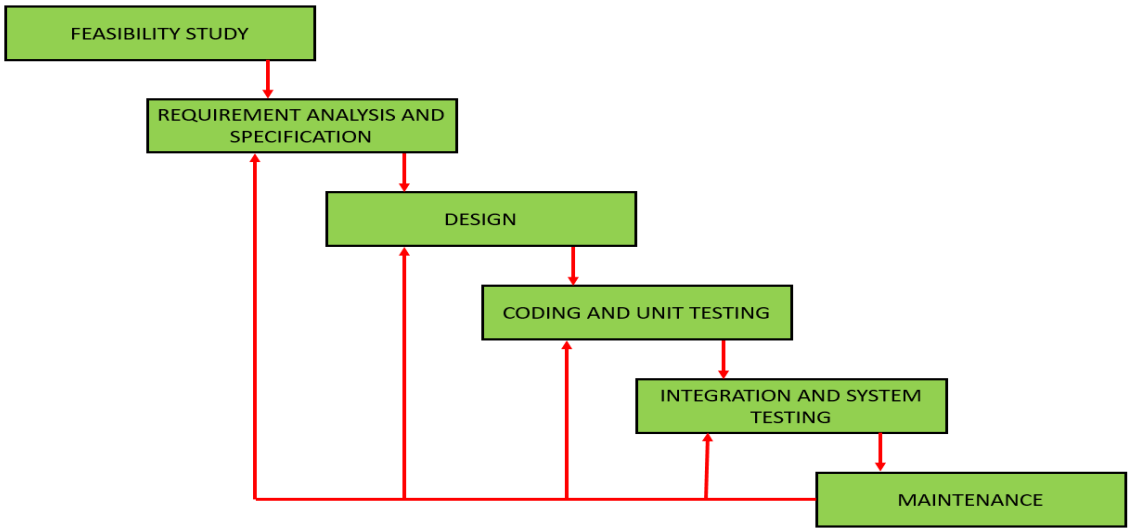

Figure 3.1: Iterative Waterfall Model

- ***Requirement analysis and specification*:** In this phase, requirements are gathered from the customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.
- ***Design:*** In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram or useCase diagram.
- ***Coding and Unit testing*:** In the implementation or coding phase, the requirements are written in the coding language and transformed into computer programmes which are called Software.

- ***Integration and Testing*:** After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.
- ***Deployment and Maintenance:*** After completing all the phases, software is deployed to its work environment. After deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.

3.2. Software Requirement Specification (SRS)

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It is the complete specification and description of requirements of a software that needs to be fulfilled for successful development of the software system. These requirements can be functional as well as non-functional.

3.2.1. Functional Requirements

A functional requirement is a statement of how a system must behave. It defines what the system should do in order to meet the user's needs or expectations. It defines the functionality of a system or one of its subsystems. The functional requirements of the system in this study are as follows:

F1- User Registration

Input: personal details like name, phone number, address, email and blood group.

Output: login to get access to the application.

F2 - User Login

Input: email and password.

Output: access home screen of the application.

F3 - Request Blood Donors

Input: define blood group, location, required date and contact.

Output: await donors' response to the request or contact proposed donors based on proximity check**.**

F4 - Accept blood request

Input: View blood request.

Output: Respond to various requesters' requests through phone call or text message.

F5 - Search for Blood donors

Input: blood type.

Output: View donors search results in a list.

F6 - keep track of blood donations made.

Output: View donation history.

F7 - Send users Notifications

Input: Blood donor request

Output: users with corresponding blood group receives request notifications

F10 - Modify Personal Information

Input: user enters new personal information.

Output: updates user's information.

F11 - View donation Campaigns and conditions to donate blood

Input: user request for information to chatbot.

Output: Get adequate response from automated chatbot.

3.2.2. Non-functional Requirements

Non-functional requirements are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality. These are basically the requirements that outline how well the system will operate. The non-functional requirements of this system are as follows:

- ***Availability:*** The application should be functional 24hours a day and 7 days a week.
- ***Safety:*** The System must be secured with proper email, and passwords for authentication to avoid unauthorized access into the system.
- ***Maintainability:*** The software should frequently be maintained and updated to ensure that all functional requirements work perfectly in accordance with the research project.
- ***Usability:*** This system must have a good looking user friendly interface in such a way that the user finds it easy and interesting to manipulate.
- ***Extensibility:*** Requirements for application extensibility in case there is a need to add new functional requirements.

- ***Capability*** The System must fulfil on storage requirements, today and in the future. The System must be scale up for increasing volume demands.

- ***Portability:*** The software should be created in such a way that it can easily be transported from one device to another without hindering the system from working correctly.

3.3. Project Plan

A project plan is a series of formal documents that define the execution and control stages of a project. A project plan sometimes called a work plan is a blueprint of the goals, objectives, and tasks needed to be accomplished for a specific project. Project planning helps you organise your time according to the work that needs to be carried out, to ensure you remain efficient for the long-term. Project planning does however require a certain attention to detail, in advance of and during the works being carried out. This project plan addresses the scope, cost and schedule baselines of this research project.

Table 3.1: Hardware Components

Hardware Components	**Specifications**
Laptop	Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.20 GHz, Installed RAM 8.00 GB, 64-bit operating system, x64-based processor
Wi-Fi-Box	Orange Easy Box

Table 3.2: Software Components

Software Components	**Types/Details**

Operating system	Windows 11 Pro
Backend	Laravel Framework
Frontend	React Native
Database	XAMPP, MYSQL
Design	StarUML

3.3.1. Cost Analyses and Estimation

Cost estimation of a research project is the process of forecasting the financial and other resources needed to complete a project within a defined scope. The cost analysis of my research project can be classified into three categories: existing, free, and paid. Items or components that are existing and free will be identified by a positive sign meanwhile those that are not existing or free, will be annotated by a negative sign. If an item is paid for, the amount will be annotated else, it will have a negative sign. This can be represented as follows:

Table 3.3: Cost Analyses

Items/Components	Existing	Free	Paid
Laptop	+	–	–
Wi-Fi Box	+	–	20,000FCFA/Monthly
Laravel framework	+	+	–
React Native	+	+	–
Other softwares	+	+	–
Laptop Maintenance	+	–	17,000FCFA

3.3.2. Project Schedule

Project Scheduling involves the listing of activities, deliverables, and milestones within a project. A schedule usually includes a planned start and finish date, duration, and resources assigned to each activity. It is the process of defining a project in detail and creating a plan for when and how it will be completed. Building a project schedule, including a timeline with clear deadlines for each task, gives me the visibility into the work I am expected to complete and keeps me involved on track.

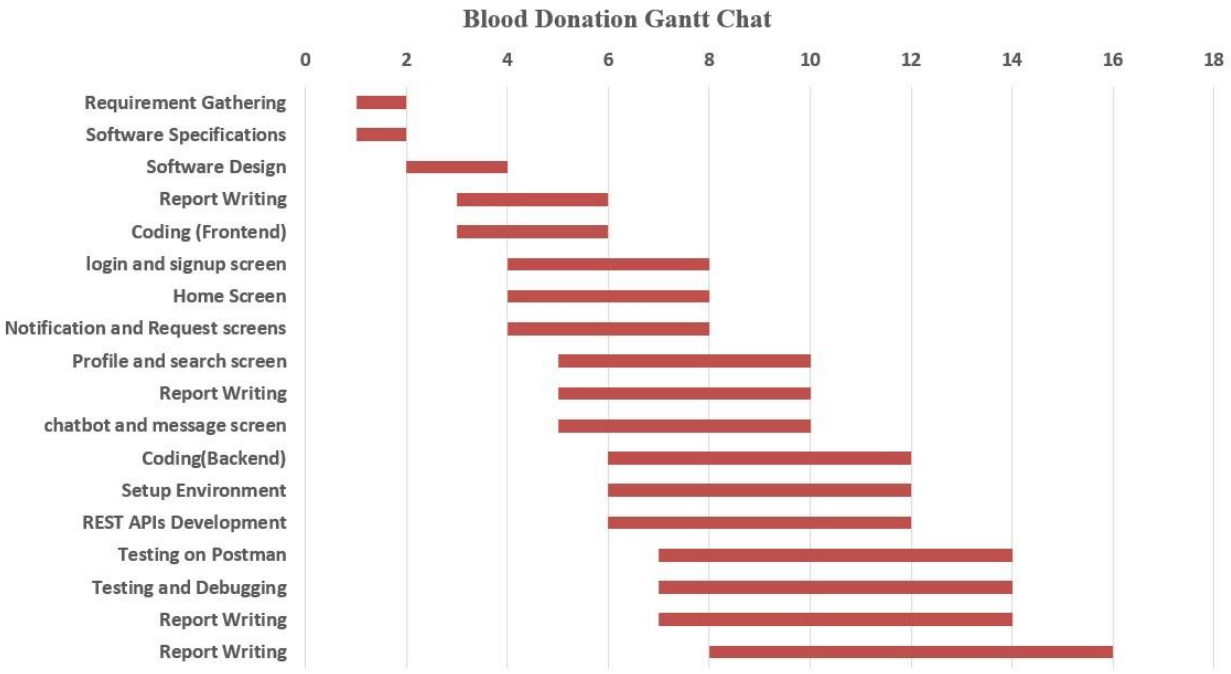

Figure 3.1: Gantt chart

3.4. Use Case Models

A use-case model is a model of how different types of users interact with the system to solve a problem. As such, it describes the goals of the users, the interactions between the users and the system, and the required behavior of the system in satisfying these goals. Use-case diagrams model the behavior of a system and help to capture the requirements of

the system. It describes the high-level functions and scope of a system. *Figure 3.3* identifies the interactions between my blood donation app and its actors.

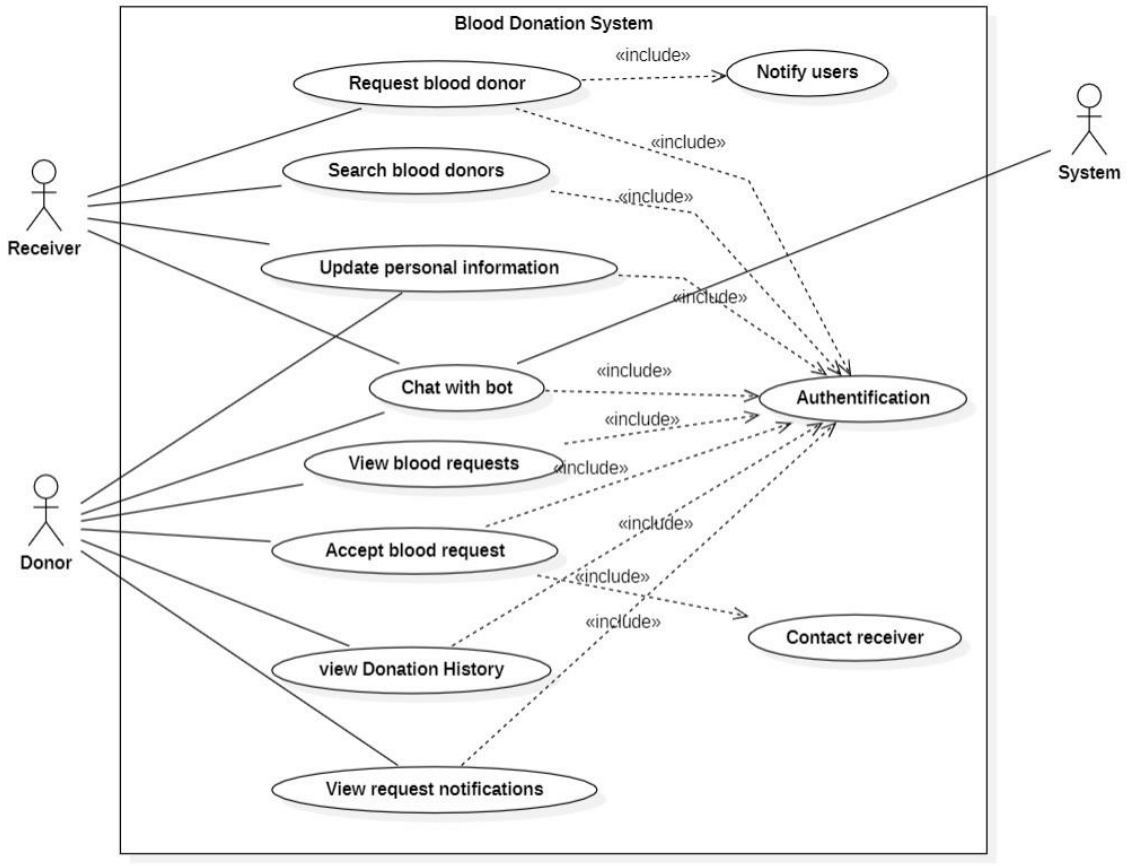

Figure 2.3: Use Case Diagram

a) Authentication

Description: This use case permits users to access the system.

Precondition: Users should have been registered with an email and password.

Post condition: Users can now log in into the system.

Actor: Users

b) Request blood donors

Description: This useCase enables users to post blood donors request for a specific blood group and locality and view proposed donors based on a proximity check.

Pre-Condition: user should be logged into the system and select the request tab.

Post condition: users can fill the blood request form, post their blood request and view proposed donors.

Actor: Users (Requester)

Includes: Authentication, notifies users.

c) Accept blood requests

Description: This useCase enables donors to accept blood request for a specific blood group.

Pre-Condition: user should be logged into the system.

Post condition: user can contact requester through a call or a text message.

Actor: Users (Donor)

Includes: Authentication, contact requester.

d) View blood donors requests

Description: This useCase enables users to view blood donors' requests.

Pre-Condition: user should be logged into the system.

Post condition: user can view requests and contact requesters.

Actor: Users

Includes: Authentication.

e) View donation history

Description: This useCase enables users to view blood donation history.

Pre-Condition: user should be logged into the system and select profile tab.

Post condition: user can view and delete donation history.

Actor: Users (Donor)

Includes: Authentication.

f) Update personal information

Description: This useCase enables users to update it personal information.

Pre-Condition: user should be logged into the system and select the profile tab.

Post condition: user can edit and update his or her personal details.

Actor: Users

Includes: Authentication.

g) Receive request notification

Description: This useCase enables users to receive notifications about urgent blood donors' request.

Pre-Condition: user should be logged into the system or not.

Post condition: user can respond to request.

Actor: Users

Includes: Authentication.

h) Chat with bot

Description: This useCase enables users to get information about conditions to donate and campaigns available in a specific area.

Pre-Condition: user should be logged into the system and select the chat tab.

Post condition: user can get appropriate respond from the bot.

Actor: Users

Includes: Authentication.

3.5. System Design

System design is the process of defining the architecture, interfaces, and data for a system that satisfies specific requirements. The need for system design arises from the fact that complex systems require careful planning and organization in order to function effectively and efficiently. System design is important in ensuring that different components of a system work together seamlessly. By defining clear interfaces and protocols, system designers can help ensure that components can communicate with each other and share data as needed.

3.5.1. Tools and Technologies used

i. **React Native:**

React Native is an open-source user interface software framework created by Meta Platforms, Inc. React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It uses an informative programming concept and the JavaScript library to develop native mobile apps. It provides a smooth and responsive user interface and significantly reduces loading time. Developing apps with React Native is cheaper and faster than native apps while still providing high-quality functionality.

Advantages of using React Native

- ✦ It supports live and hot reloading, meaning developers can see the changes in their code instantly without recompiling the app. This feature enhances the productivity and efficiency of the developers, as they can create the best and most powerful mobile applications with a front-end solution that React Native offers.
- ✦ It allows developers to write code only once and use it to power both their iOS and Android apps with minor platform-specific changes. This means that it is cost-effective and time-saving for app development.
- ✦ It is based on JavaScript, which is a popular and widely used programming language in the world. This is another reason behind its success.
- ✦ It gives a native look and feel to the apps, as it uses native components that render natively on each platform.

ii. **Node js**

It is an open source runtime environment built on Chrome's V8 JavaScript engine. It uses an event-driven and non-blocking I/O model that makes Node.js lightweight, efficient and extremely fast for building applications, APIs and executing web development services.

iii. **Express**

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. It allows to set up middlewares to respond

to HTTP Requests and defines a routing table which is used to perform different actions based on HTTP Method and URL. With the help of express, one can easily create REST APIs.

iv. **REST API**

REST (Representational state transfer) is a popular architecture that is used to create web services while an API (Application Programming Interface) is a code that allows two software programs to communicate with each other. REST API is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server.

v. **Postman**

Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses. While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API. In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

vi. **Android Emulator**

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device. The emulator offers these advantages:

- *Flexibility:* In addition to being able to simulate a variety of devices and Android API levels, the emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.
- *High fidelity:* The emulator provides almost all the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.
- *Speed:* Testing your app on the emulator is in some ways faster and easier than doing so on a physical device. For example, you can transfer data faster to the emulator than to a device connected over USB.

vii. **XAMPP**

XAMPP is an abbreviation where X stands for ***Cross-Platform***, A stands for ***Apache***, M stands for ***MYSQL***, and the Ps stand for ***PHP*** and ***Perl***, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, Maria DB, PHP, and Perl. XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and Maria DB is the most vividly used database developed by MySQL.

viii. **Dialogflow**

Dialogflow (formerly called API.AI) is Google-owned chatbot development framework. It is used for Natural Language Processing using machine learning. It is a natural language understanding platform used to design and integrate a conversational user interface into

mobile apps, web applications, devices, bots, interactive voice response systems and related uses.

ix. **StarUML**

StarUML is a software engineering tool for system modeling using the Unified Modeling Language, as well as Systems Modeling Language, and classical modeling notations.

x. **Visual Studio Code**

Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python, Java, C++, JavaScript, and more.

3.5.2. Class Diagram

Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. It is a type of static structure diagram that describes the structure of a system by showing the systems: classes, their attributes, operations (or methods), and the relationships among objects. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. *Figure 3.4* shows the class diagram of this system.

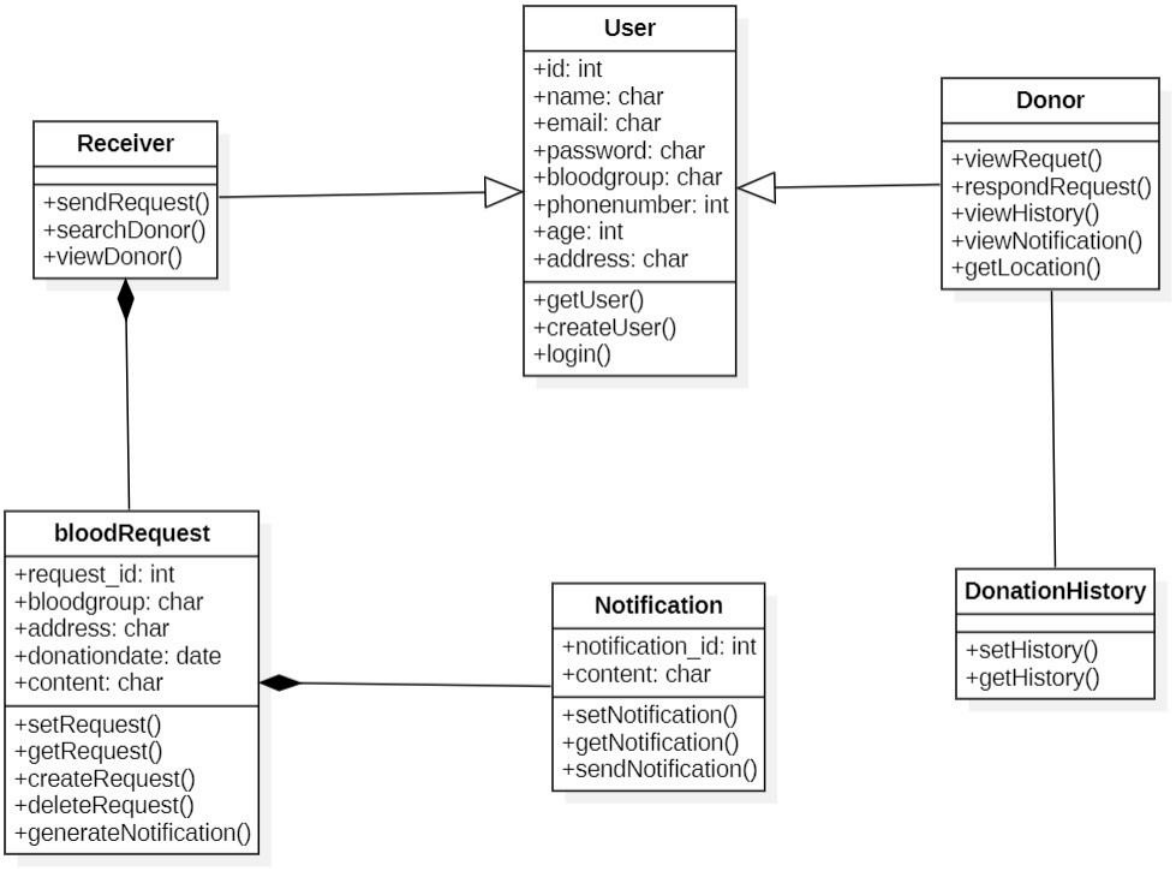

Figure 3.3: Class Diagram

3.5.3. Database Design

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. The main objectives of database design are to produce logical and physical designs models of the proposed database system. The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically. Meanwhile, the physical data design model involves translating the logical database design of the database onto physical

media using hardware resources and software systems such as database management systems. *Figure 3.5* shows the relational database design of the system

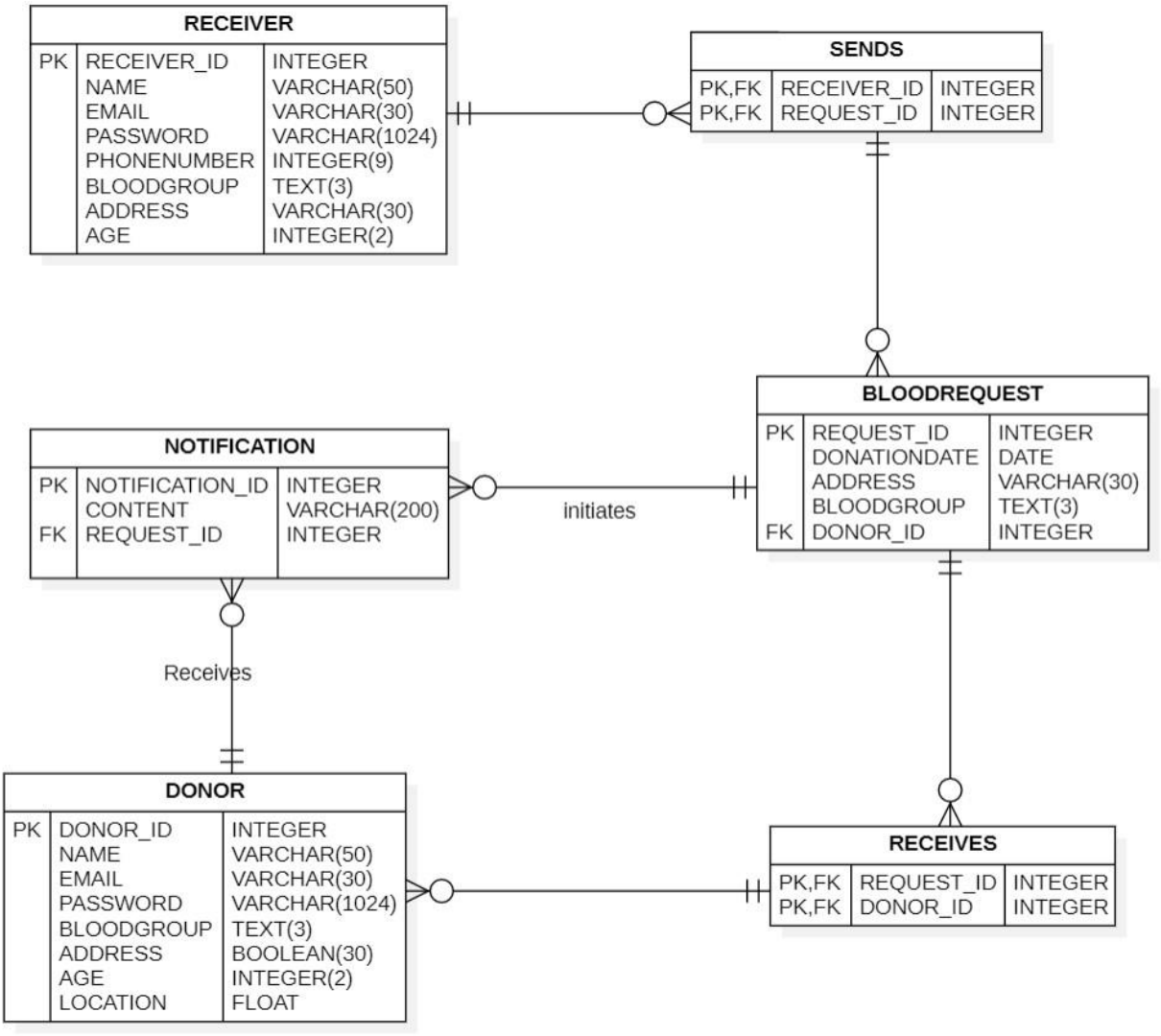

Figure 3.4: Relational Database Design

3.5.4. Activity Design

An activity diagram is a type of UML diagram that is used to model the behavior of a system or process in a clear and structured way. It is often used to show the flow of control in a system and refer to the steps involved in the execution of a use case. Activity diagrams

can be used to model business requirements, create a high-level view of a system's functionalities, analyze use cases, and for various other purposes. They show the individual steps in an activity and the order in which they are presented, and can also show the flow of data between activities.

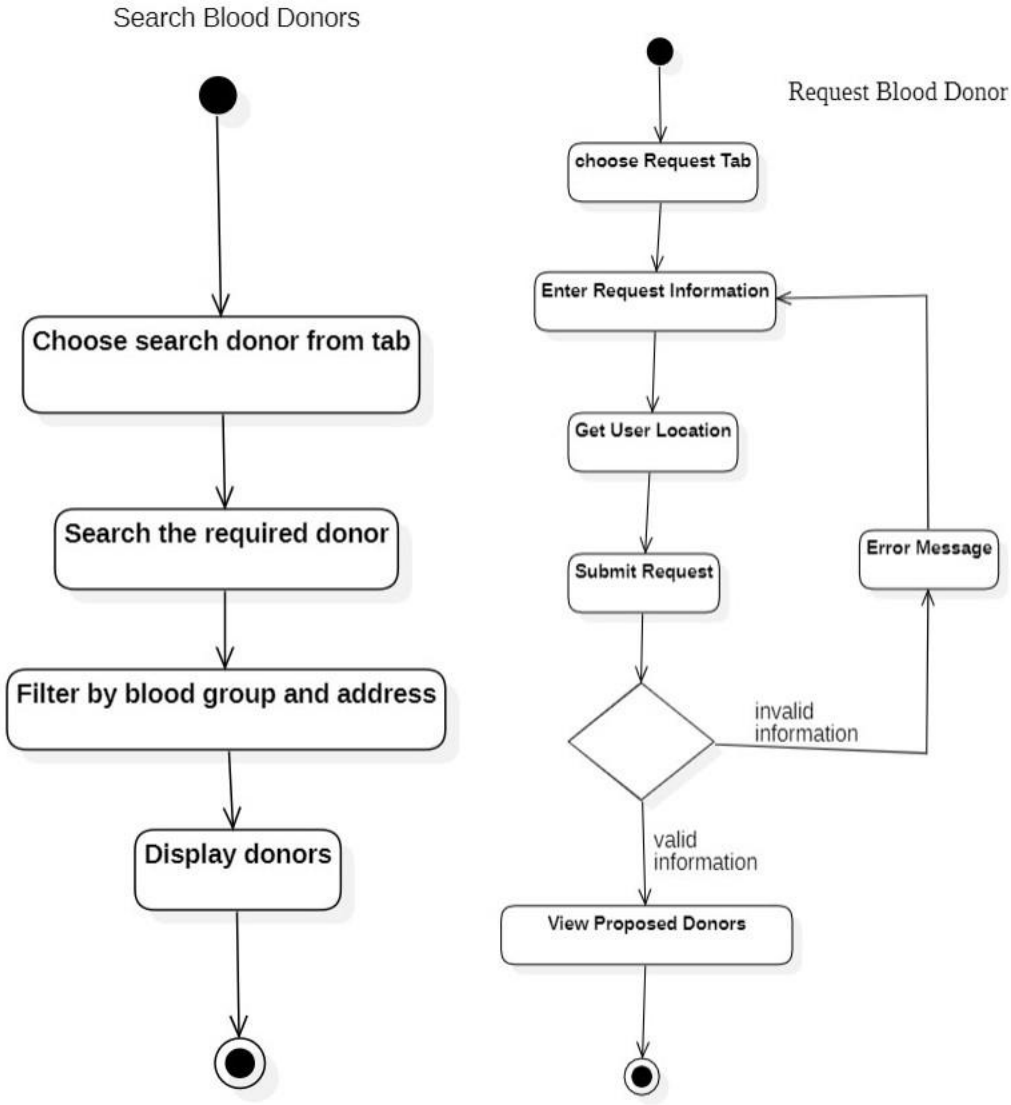

Figure 3.5: search donors and request donors activities diagram

CHAPTER 4

IMPLEMENTATION AND RESULTS

This chapter shows a clear description of the implementation of the above project plan and design, as well as the results obtained. It provides various code snippets of the work done and screenshots of its outcome.

1. BACK END

The back-end, also called the server-side, consists of the server which provides data on request, the application that channels it, and the database which organizes the information.

1.1.REST APIs

RESTful APIs were used to create interaction between my database and my mobile application as well as sending and receiving HTTP requests and responses.

- ***Model:*** In order to connect the Express js APIs to tour MySQL database, a package named ***mysql*** was to be downloaded inside the project directory and configured such all the authentication details of the database should be passed in the database configuration file which can be called and included in other file. This can be shown in *figure 4.1* below:

Figure 4.1: database configuration

An express application was created which returns middleware that only parses json and only looks at requests where the Content-Type header matches the type option. Essentially, whenever a request hits my backend, my Express app will execute the functions I passed to ***app.use()*** in order that is, my users and requests functions to get the various routes and the corresponding controller. *Figure 4.2* shows how my app.js file was configured.

Figure 4.2: creating express application

- ***Service:*** The services contains the database queries, the various input data and returning objects or throwing errors. The *figure 4.3* shows the service to get a specific receiver's blood request from the database. This service return a callback which returns an error if there is, or the query result.

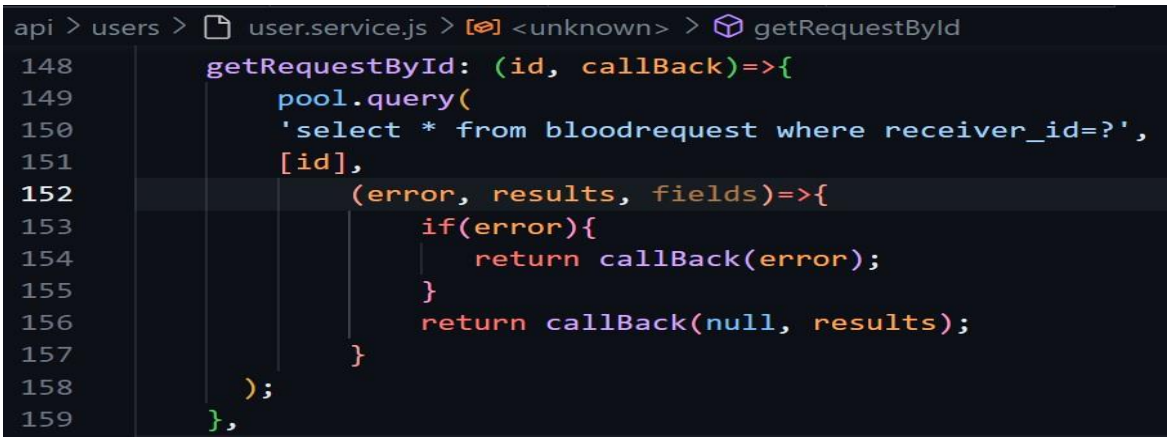

Figure 4.3: API service get a specific request

- ***Controller:*** The controllers handles all the logic behind validating request parameters, query, Sending Responses with appropriate content type. It acts as a middleman between route handlers and services. It controls the flow of data from the client to the service layer. The controller in *figure 4.4* uses the ***getRequestById*** shown above and returns either the error message in the console or the results of the query in the json format.

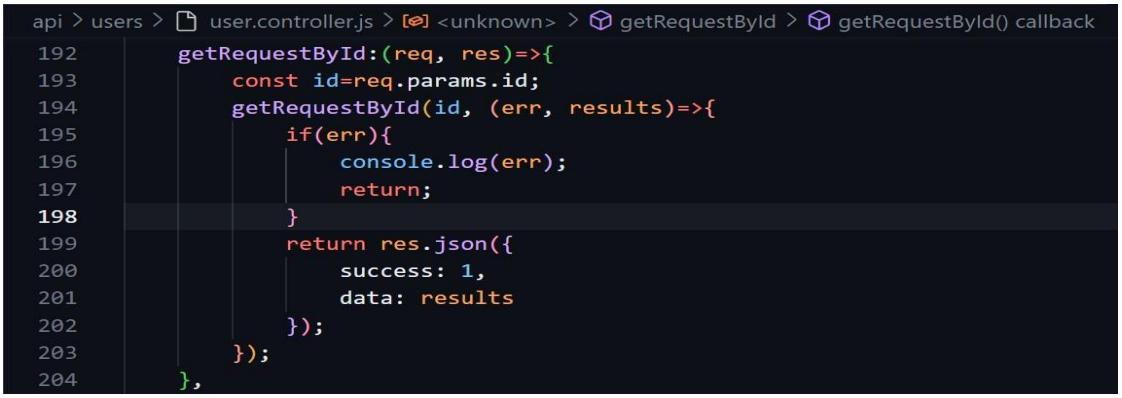

Figure 4.4: Controller to get specific donor request

- ***Route:*** These are the API routes maps to the Controllers**.** It is responsible for passing the request object and related data to the respective controller. *Figure 4.5* shows the

various endpoints developed and tested on Postman to check whether the API returns the appropriate result.

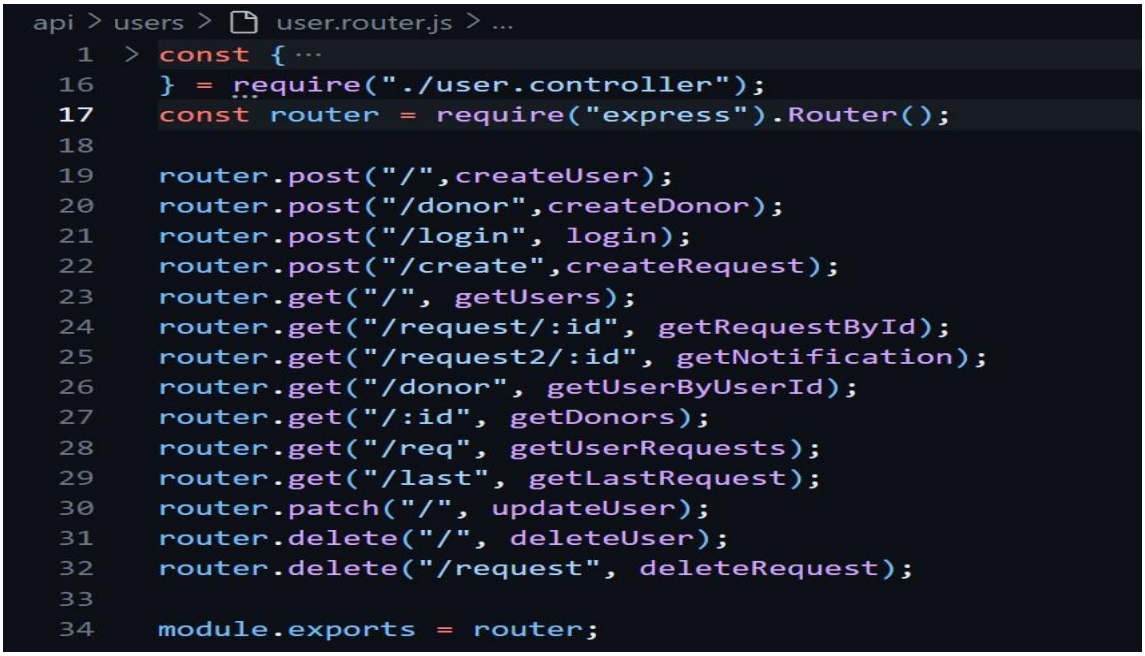

Figure 4.5: APIs various routes or endpoint

2. Client Side

2.1.API Consumption

To consume my APIs in my React Native application, I used two methods that are available, the built-in ***Fetch API*** and the third-party library ***Axios***. I define the API endpoint or URL and any necessary parameters needed. The Fetch API is then used to make a request to the API endpoint which parse the response data, which is typically in JSON format and update the app state with the retrieved data. *Figure 4.6* get the blood request made by a specific user. It uses the ***id*** of the user connected to get it specific record. This ***id*** is been stored in an ***asynchronous storage*** that helps to store user's information in the mobile phone and can be accessed from any screen in the application.

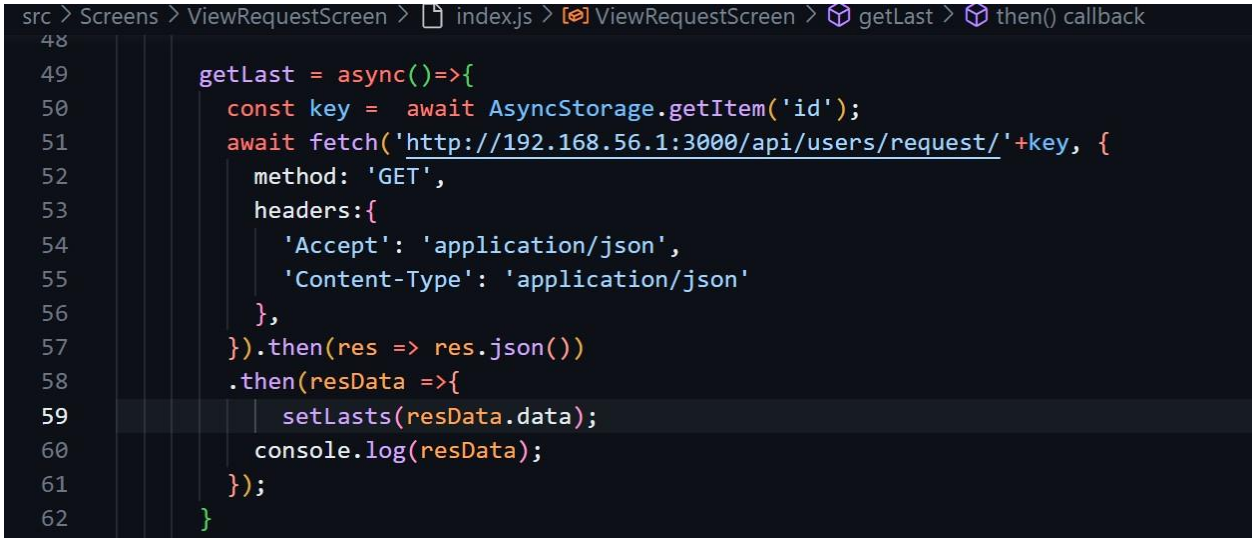

Figure 4.6: API Consumption using Fetch Method

2.2.Chat Bot

In order to create a chat bot that will be able to give user conditions to donate and available donation campaigns using Dialogflow, I needed to create an account on the Dialogflow platform, create an agent called ***BloodDonation***, which is a conversational interface that handles the flow of the conversation with the end-user. I defined my intents, which are the actions that the agent can perform in response to user input. I defined my entities, which are the parameters that the agent can extract from user input and I trained the agent by providing sample phrases and responses for each intent. The agent is then tested by sending sample phrases and verifying that my agent responds correctly. The chatbot was the Integrate the agent with your application or platform using the provided integration options.

```
componentDidMount() {
    Dialogflow_V2.setConfiguration(
      dialogflowConfig.client_email,
      dialogflowConfig.private_key,
      Dialogflow_V2.LANG_ENGLISH_US,
      dialogflowConfig.project_id
    );
  }
onSend(messages = []) {
    this.setState(previousState => ({
      messages: GiftedChat.append(previousState.messages, messages)
    }));
let message = messages[0].text;
    Dialogflow_V2.requestQuery(
      message,
      result => this.handleGoogleResponse(result),
      error => console.log(error)
    );
  }
handleGoogleResponse(result) {
    let text = result.queryResult.fulfillmentMessages[0].text.text[0];
    this.sendBotResponse(text);
  }
```

Figure 4.7: Dialogflow configuration and response handler

From *figure 4.7,* the ***ComponentDidMount*** function setup the configuration to our Dialogflow agent. All the required information such as the client email and private key are provided and stored in an ***env*** file saved in the project root directory. The ***onSend*** function, which will be invoked whenever the send button is clicked by the user is used for sending the message to the bot. The response from the Dialogflow will then be stored in the result variable which is then passed to ***handleGoogleResponse*** function and the ***sendBotResponse*** function will append the text response into chat window.

2.3.Proximity Check

A proximity check was done each time a user sends a blood request. This is done to enable users find potential donors of the specific blood group and their corresponding distance from the request location. In order to implement this functionality, the react native dependency ***Geolocation*** and ***Geolib*** was installed. The ***Geolocation*** library helped us to get users actual location. This is done by adding in our ***android.xml file***,

the Permission necessary for our mobile phone to access the user permission, as shown in *figure 4.8*.

Figure 6.8: Add Location Permission in android.xml file

Later, a ***getLocation*** function was implemented. This function helps us to get the exact current position of the user using the ***getCurrentPosition*** function, but before it is been done, the user must have validated the permission to access his or her defined location. The state Location is then updated and from it, we can access both the position's latitude and longitude.

Figure 4.9: getLocation function

The location gotten from figure 4.9, is then passed through the navigation prop as arguments to be accessed in the proposed donors screen. The function

getPreciseDistance () offered by the ***Geolib*** dependency permits us to get the precise distance between two coordinate point in metres or kilometres. The actual request location is then compared to other donors location and results displayed.

```
{request.map((item,index)=>{

var pdis = getPreciseDistance(
  {latitude: latitude1, longitude: longitude1},
  {latitude: item.latitude, longitude: item.longitude},
);
```

Figure 4.10: Function that gets distance between coordinate

3. RESULT

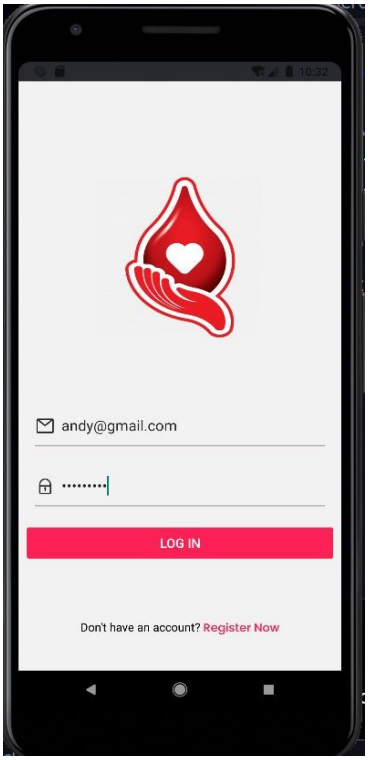

Figure 4.11: Login Screen

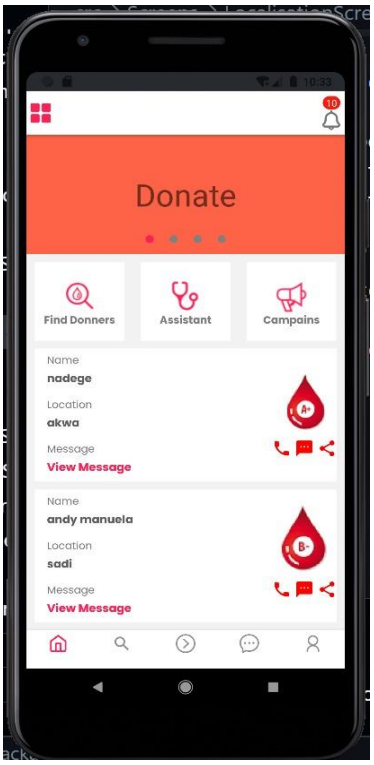

Figure 4.12: Home Screen

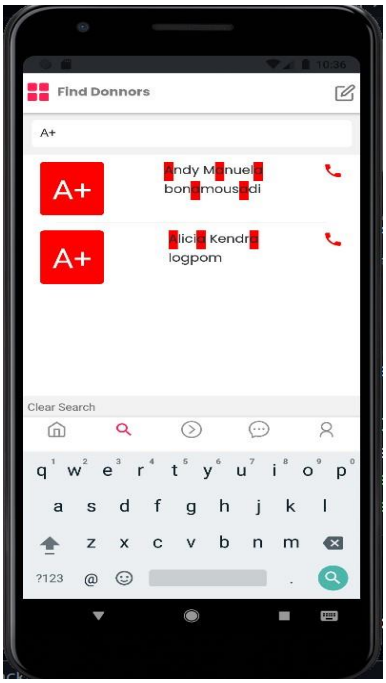

Figure 4.13: Search Donor Screen

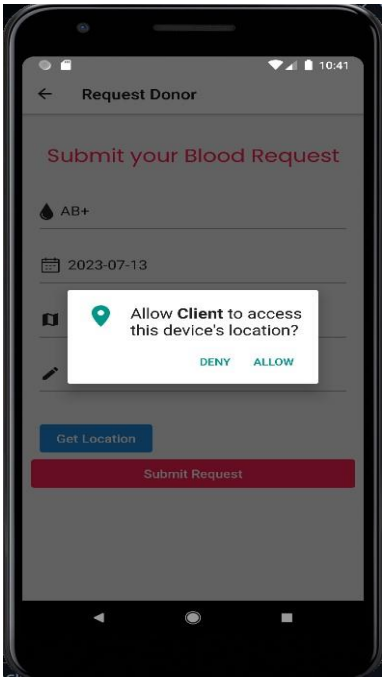

Figure 4.14: Request Donor Screen

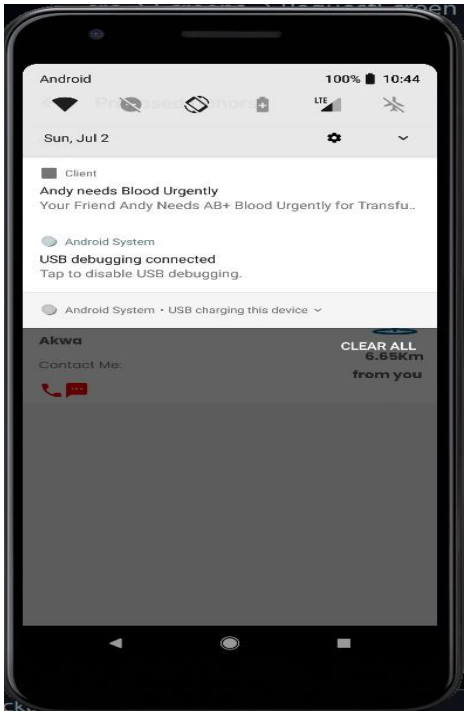

Figure 4.15: App notifies users

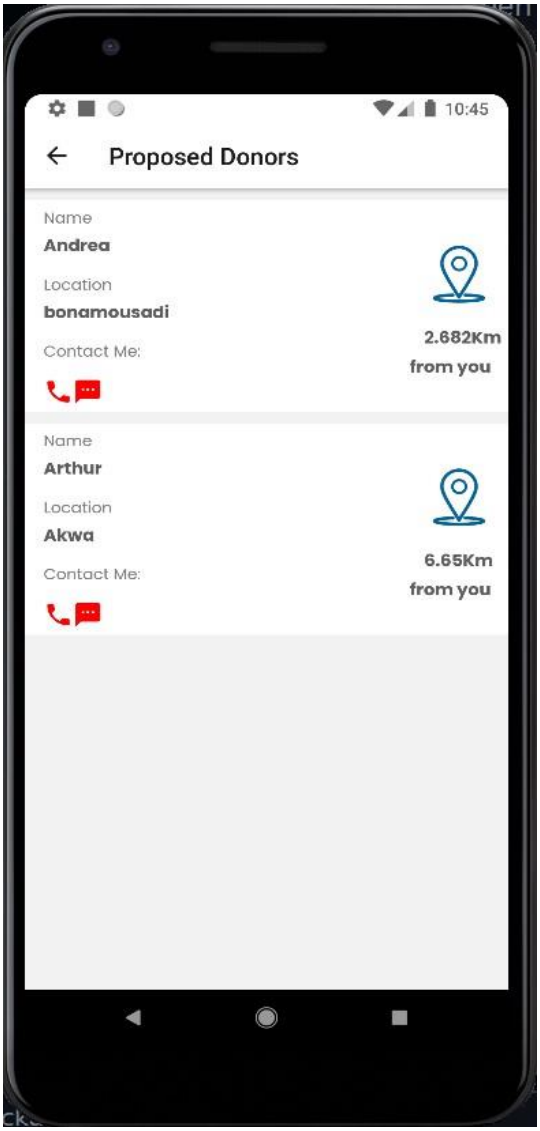

Figure 4.16: Proposed Donors Screen

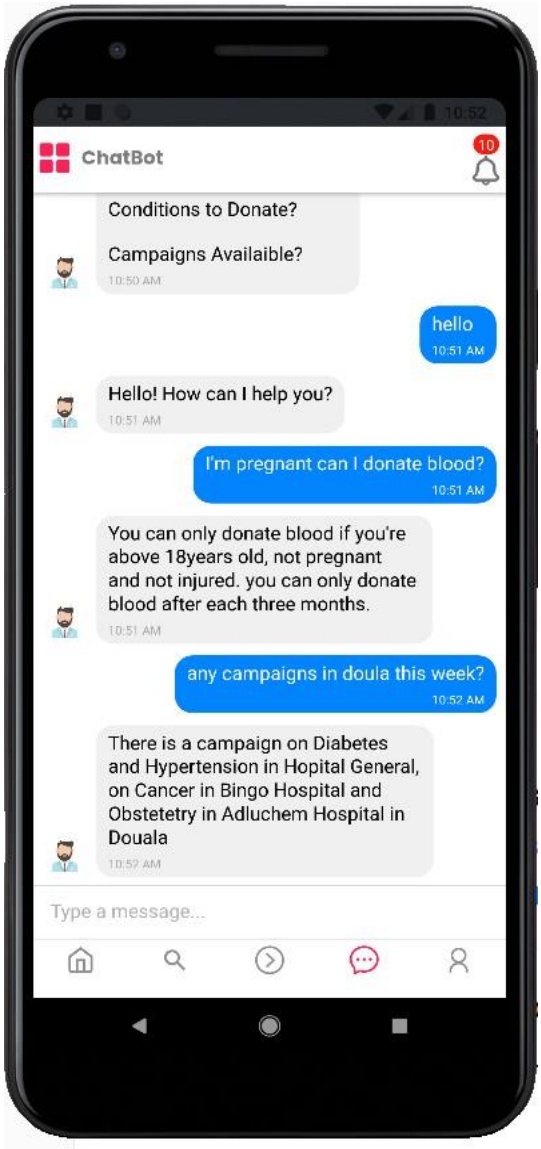

Figure 4.17: Chat Bot

CHAPTER 5

CONCLUSION AND RECOMMENDATION

This chapter summarises the research findings of the blood donation android application, evaluates if the initial objectives were finally met, as well as providing the recommendations and future work.

5.1. Summary of Research Findings

This research project was aimed at building a blood donation mobile application that will serve as an intermediate between blood donors and recipients. The application developed provide functionalities for users such as posting, responding and viewing blood request, searching blood donors as well as getting necessary information on conditions to donate blood and necessary donation campaigns.

5.2. Lesson Learnt

Throughout this research study, I was able to be learn and be comfortable with mobile application development using React Native. This study improved my knowledge of JavaScript and API development using Express js as well as integrating Dialogflow's conversational chat bot in an application.

5.3. Recommendation / Future Works

This blood donation android application can be improved in the following ways:

- Adding a gaming feature in to the mobile application which will enhance and motivates donors' experiences.
- Earn achievement badges each time a donation is made.
- Use react native health kit to get users' biometric and heath date.

5.4. Conclusion

This application reduces time consumption to a greater extent used for searching for blood donors of the required blood group at a particular location. Thus this blood donation application provides the required results in less time and also helps in saving lives. Blood donation should be an everyday plebiscite in Cameroon to the benefit of present and upcoming generations, for every blood donation is a gift of life.

REFERENCES

Abderrahman, B.H., Saleh, & M.Y.N. (2014). *Investigating knowledge and attitudes of blood donors and barriers concerning blood donation.* Procedia-Soc.

Alruwaili, & Muharib. (2017). *The Benefits of Donating Blood.*

Blood safety and availaibility. (2023, June 2). Retrieved from www.who.int/: https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability

Brislin, a., Mayan, J., Canessane, R., & Hamlin, M. (2017). *Blood donation and life saver app.*

Highlits of Transfusion Medicine History. (2023). Retrieved from www.aabb.org: https://www.aabb.org/news-resources/resources/transfusion-medicine/highlights-of-transfusion-medicine-history#:~:text=1665%20The%20first%20recorded%20successful,transfusions%20from%20lambs%20to%20humans.

O'Grady, B. (2022). *Code Institute*. Retrieved from https://codeinstitute.net/: https://codeinstitute.net/nl/blog/what-is-a-framework/

Roshni, M., Sonali, K., Pooja, A., & Shreya, P. N. (2020). *BLOOD DONATION MANAGEMENT SYSTEM.* International Research Journal of Engineering and Technology (IRJET).

Schneider, W. H. (2013, september). *History of blood transfusion in sub-saharan Africa*. Retrieved from www.pubmed.ncbi.nlm.nih.gov: https://pubmed.ncbi.nlm.nih.gov/22981696/

Shah, Mahmoud, Muhrad, A., Jah, & Saheed, U. (2023). *Online Blood Donation Management System.*

Staff, M. C. (2022, April 27). *www.mayoclinic.org*. Retrieved from Blood Transfusion-MayoClinic: https://www.mayoclinic.org/tests-procedures/blood-transfusion/about/pac-20385168#:~:text=A%20blood%20transfusion%20is%20a,a%20vein%20in%20your%20arm.

Towards 100% Voluntary Blood Donation: A Global Framework for Action. (2010). In W. H. Organisation, *Towards 100% Voluntary Blood Donation: A Global*

Framework for Action (pp. 2, Voluntary blood donation: foundation of a safe and sufficient blood supply). National Library of Medicine.