A

Theme Based Project Report on

**BLOOD DONATING APPLICATION**

Submitted for partial fulfilment of the

requirements for the award of the degree of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

**SHAIK ABUBAKR (2451-22-733-133)**

**GUNDI HARSHA (2451-22-733-143)**

**MOHAMMED NABEEL (2541-22-733-185)**

Under the guidance of

**Ms. B. SARITHA**

Associate Professor

Department of CSE



**MATURI VENKATA SUBBA RAO(MVSR) ENGINEERING COLLEGE**

**(An Autonomous Institution)**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Balapur Mandal, Hyderabad – 501 510

Academic Year: 2023-2024

**MATURI VENKATA SUBBA RAO(MVSR) ENGINEERING COLLEGE**

**(An Autonomous Institution)**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Balapur Mandal, Hyderabad – 501 510

Academic Year: 2023-2024



*This is to certify that the Theme Based project work entitled “****BLOOD DONATING APP****” is a bonafide work carried out by* ***SHAIK ABUBAKR (2451-22-733-133), GUNDI HARSHA (2451-22-733-143), MOHAMMED NABEEL (2451-22-733-185)*** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *from* ***Maturi Venkata Subba Rao(MVSR) Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2023-2024. under our guidance and supervision.*

*The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.*

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Internal Guide** |  | **Head of the Department** |
| B. SARITHA |  | Prof J PrasannaKumar |
| Associate Professor |  | Professor & Head |
| Department of CSE |  | Department of CSE |

**DECLARATION**

This is to certify that the work reported in the present Theme Based project entitled “BLOOD DONATING APPLICATION” is a record of Bonafide work done by us in the Department of Computer Science and Engineering, M.V.S.R. Engineering College, Osmania University. The reports are based on the work done entirely by us and not copied from any other source. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

|  |  |
| --- | --- |
| ***SHAIK ABUBAKR*** | ***GUNDI HARSHA MOHAMMED NABEEL*** |
| ***(2451-22-733-133)*** | ***(2451-22-733-143) (2451-22-733-185)*** |
|  |  |

Dept. of CSE, MVSREC(A)

II

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project guide Ms. B.SARITHA, Associate Professor for her valuable suggestions and interest throughout the course of this project. We are also thankful to our principal Dr. G. Kanaka Durga and Prof. J Prasanna Kumar, Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (CSE). We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed. Finally, we would like to

take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

SHAIK ABUBAKR (2451-22-733-133)

GUNDI HARSHA (2451-22-733-143)

MOHAMMED NABEEL (2451-22-733-185)

##### VISION

To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

##### MISSION

To make the learning process exciting, stimulating and interesting.

To impart adequate fundamental knowledge and soft skills to students. To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students. To develop economically feasible and socially acceptable software.

##### PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

The Bachelor’s program in Computer Science and Engineering is aimed at preparing graduates who will:

PEO-1: Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives.

PEO-2: Practice life-long learning by pursuing professional certifications, higher education or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

##### Program Specific Outcomes (PSOs)

PSO1: Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.

PSO2: Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship.

 **CERTIFICATES**

1. **2451-22-733-133**
2. **2451-22-733-185**





1. **2451-22-733-143**



ABSTRACT

The blood donating application is a mobile application that helps people give and receive blood donations. It lets Users sign up to donate blood or request blood donations. When someone needs blood, they can post a request on the application. Donors can see these requests and choose to help. Once a donor accepts a request, the application shares contact info so they can coordinate. The application also keeps track of donations and lets requestors confirm when a donation is successful. Overall, it makes it easier for people to donate blood and for those in need to get help.

SHAIK ABUBAKR (2451-22-733-133)

GUNDI HARSHA (2451-22-733-143)

MOHAMMED NABEEL (2451-22-733-185)

INDEX

|  |  |
| --- | --- |
| CHAPTER | PAGE NO.’S |
| 1.INTRODUCTION  1.1 PROBLEM STATEMENT  1.2 OBJECTIVE  1.3 MOTIVATION  1.4 SCOPE | 1-3 |
| 2.SYSTEM REQUIREMENT SPECIFICATIONS  2.1 SOFTWARE REQUIREMENTS  2.2 HARDWARE REQUIREMENTS | 4 |
| 3.TOOLS USED | 5-6 |
| 4.SYSYTEM DESIGN  4.1 ARCHITECTURE | 7-8 |
| 5.IMPLEMENTATION  5.1 ENVIRNONMENTAL SETUP  5.2 MODULES AND THEIR DESCRIPTION | 9-11 |
| 6.TESTING AND RESULTS  6.1 SCREENSHOTS | 12-18 |
| 7.CONCLUSION AND FUTURE ENCHANCEMENTS | 19 |
| 8.REFERENCES | 20 |
| 9.APPENDIX | 21-22 |

**LIST OF FIGURES:**

|  |  |  |  |
| --- | --- | --- | --- |
| S NO. | FIGURE NO. | FIGURE NAME | PAGE NO. |
| 1 | 4.1 | SYSTEM ARCHITECTURE FOR BLOOD DONATING APPLICATION | 7 |
| 2 | 4.2 | MODULES SYSTEM FLOW | 8 |
| 3 | 6.1 | SIGNUP PAGE | 12 |
| 4 | 6.1.1 | FIREBASE AUTHENTICATION | 12 |
| 5 | 6.2 | LOGIN PAGE | 12 |
| 6 | 6.3 | UPDATE PROFILE PAGE | 13 |
| 7 | 6.3.1 | USERS DATABASE | 13 |
| 8 | 6.4 | HOME PAGE | 13 |
| 9 | 6.5 | UPDATED HOME PAGE | 14 |
| 10 | 6.6 | REQUEST FOR BLOOD PAGE | 14 |
| 11 | 6.6.1 | REQUSTORS HOME PAGE | 15 |
| 12 | 6.7 | HOME PAGE WITH UPDATED EMERGENCY REQUESTS | 15 |
| 13 | 6.8 | ALL BLOOD REQUEST PAGE | 15 |
| 14 | 6.9 | NOTIFICATION PAGE | 16 |
| 15 | 6.10 | ACCEPTED REQUEST DATABASE | 17 |
| 16 | 6.11 | MY REQUEST PAGE | 16 |
| 17 | 6.12 | HISTORY DATABASE | 17 |
| 18 | 6.13 | DONOR HISTORY | 18 |
| 19 | 6.14 | REQUEST HISTORY | 18 |

**1.INTRODUCTION**

* 1. **PROBLEM STATEMENT**:

The current blood donation system faces numerous challenges including inefficiencies in donor recruitment, limited accessibility to donation centers, and delays in matching donors with recipients. Coordination between blood banks and donors is often fragmented, leading to shortages and mismatches in critical blood supplies. These problems make it harder to get blood to patients who need it quickly, which can put their health at risk.

Implementing a dedicated mobile application for blood donation offers a solution to these challenges. By providing a centralized platform, the application can streamline the donor recruitment process, and facilitate real-time matching between donors and requestors. Such an application has the potential to significantly enhance the efficiency and effectiveness of the blood donation process, ultimately saving lives and improving public health outcomes.

* 1. **OBJECTIVES:**

**Increase Blood Donation Rates:** The application aims to boost blood donation rates by making it easier for eligible donors to find donation opportunities and participate in donation drives.

**Contribute to Health Sector:** Contribute to the overall improvement of public health by addressing blood shortages, reducing response time during emergencies, and creating a sustainable ecosystem for blood donation. Top of Form

**Promoting Blood Donation Awareness:** Raise awareness about the importance of blood donation through educational resources and outreach campaigns within the app.

* 1. **MOTIVATION:**

Our Motivation for Development of this Blood Donating Application are the following;

**Lifesaving Impact:** This blood donation application has the power to save lives. Motivate Users by emphasizing that each donation directly contributes to someone's chance at a healthier, longer life.

**Emergency Preparedness**: Blood donations are crucial during emergencies and natural disasters when the demand for blood often increases dramatically. By donating blood regularly, you can help ensure an adequate supply of blood is available for those in need during critical times.

**Saving Lives:** Empowering users to become heroes by easily donating blood, contributing to a life-saving network that can make a significant impact during emergencies.

**Health Impact:** Showcase the positive health benefits of regular blood donation, motivating users to adopt a habit that not only helps others but also contributes to their own well-being.

**Easy Good Deeds:** Showcase the simplicity of making a positive impact. Motivate users by emphasizing that with just a few clicks on their phones, they can easily engage in a meaningful act of kindness that could make a world of difference to someone in need.

**Making a Big Impact**: Even a little bit of blood can help a lot of people. By giving blood, you're making a big impact and helping your community.

* 1. **SCOPE:**

The scope of the blood donating application aims to create a comprehensive solution that streamlines the blood donation process, enhances communication between donors and requestors, and ultimately contributes to saving lives and improving public health outcomes.

**IN SCOPE:**

**User Registration**: Users can create accounts by providing basic information like username, email, and password.

**User Authentication**: The application will authenticate users using email and password.

**Profile Management**: Users can manage their profiles by adding or editing personal details like name, email, phone number, blood group, date of birth, gender, and location.

**Blood Requests**: Registered users can post requests for blood donations specifying their requirements like patient name, blood type, quantity needed, location, and contact details.

**Donate Blood**: Registered Users can Donate Blood if they are Active for Donation.

**Contact Exchange:** Upon accepting a Blood request, contact information will be exchanged between Donor and Requestor to facilitate the donation process.

**Donation Verification**: Implementing a feature for requestors to verify whether the donor has donated blood, allowing requestors to confirm the successful completion of the donation process.

**OUT OF SCOPE**:

**Medical Consultations:**

* + The application will not provide medical advice or consultations.
  + Users should consult healthcare professionals for medical concerns.

**Blood Testing:**

* + The application will not conduct blood tests or provide medical diagnostics.
  + Users should visit healthcare facilities for blood tests.

**Financial Transactions:**

* + The application will not facilitate financial transactions related to blood donation.
  + Users should not exchange money for blood donations through the app.

**Transportation Services:**

* + The application will not provide transportation services for donors or blood products.
  + Users are responsible for arranging transportation to donation centers.

**ASSUMPTIONS:**

1. **User Responsibility:**
   * Users are responsible for providing accurate and truthful information during registration.
   * Users are expected to follow medical guidelines and eligibility criteria for blood donation**.**
2. **Application Availability:**

• The application will be available for download on Android devices only.

• Users must have access to compatible devices and internet

connectivity to use the app.

**DEPENDENCIES:**

1. **Internet Connection:**
   * Users require a stable internet connection to access the app's features and functionalities.
   * The app's performance may be affected by the quality of the internet connection.
2. **Database Integration:**
   * The application relies on integration with a database to store user information and donation requests.
   * Database maintenance and updates may affect the app's functionality.
3. **Regulatory Compliance:**
   * The application must comply with regulatory requirements related to healthcare and data privacy.
   * Changes in regulations may necessitate updates to the app's features and policies**.**

**Top of Form**

1. **SYSTEM REQUIREMENTS AND SPECIFICATIONS**
   1. **SOFTWARE REQUIREMENT:**

**1.Android Studio:**

Android Studio is the official integrated development environment (IDE) for Android application development. It provides tools for designing, coding, testing, and debugging Android apps.

**2.Java or Kotlin Programming Language:**

Android applications can be developed using either Java or Kotlin programming languages. Both languages are officially supported by Android Studio.

**3.Firebase Database:**

Firebase provides a suite of tools and services for developing mobile applications. For Blood Donating application, Firebase is used for features such as user authentication and real-time database.

**4. Internet Connection:**

A reliable internet connection is required to download dependencies, plugins,

and Firebase configurations.

* 1. **HARDWARE REQUIREMENT:**

1. **Processor**: A fast processor is important for running software tools and compiling code. A multi-core processor is recommended for best performance.

2. **Memory** (RAM): A minimum of 8GB of RAM is recommended for building a

website, but more is better. Having more RAM allows the computer to handle

larger files and run multiple programs at the same time.

3. **Storage**: A solid-state drive (SSD) is recommended for the best performance,

As it is faster than a traditional hard drive. A minimum of 512GB of storage

is recommended.

**3. TOOLS USED:**

**ANDROID STUDIO:**

Android Studio is a comprehensive integrated development environment (IDE) specifically designed for Android application development. It provides a range of features and tools that make application development efficient and straightforward. Here's how Android Studio is helpful in creating an Android app:

1. **Project Setup:**

Android Studio assists in creating and organizing your project structure. It sets up necessary directories, configurations, and files for our Blood Donating Application.

1. **Code Editor:**

The IDE offers a powerful code editor with features like auto-completion, code navigation, and syntax highlighting. It supports both Java and Kotlin, the primary languages for Android development.

1. **Layout Editor:**

Android Studio provides a visual layout editor for designing Blood Donating application’s User Interface.

1. **XML Editor:**

For finer control over UI design, Android Studio allows us to use the XML editor to directly edit layout files. Android Studio highlights errors and provides suggestions while writing code.

1. **Emulator:**

The built-in emulator allows us to test application on various Android devices and screen sizes without needing physical devices. It supports different Android versions and configurations.

1. **Real-time Testing:**

Android Studio supports real-time testing with features like Instant Run. Changes are instantly on the emulator or device without restarting the entire app.

1. **Integrated Debugger:**

Debugging is made easy with the integrated debugger. It allows us to set breakpoints, inspect variables, and step through the code to find and fix issues.

1. **Version Control Integration:**

Android Studio seamlessly integrates with version control systems like Git. Using this we can commit, pull, push, and manage your project's version history directly from the IDE.

1. **Firebase Integration:**

Android Studio has built-in tools for integrating Firebase services into our Blood Donating application. It is easy to add authentication, real-time database, cloud messaging, and more.

By leveraging these features, We can efficiently build, test, and deploy Blood Donating application using Android Studio.

**FIREBASE DATABASE:**

Firebase is a powerful platform provided by Google that offers various services to facilitate the development of mobile applications. In a Blood Donation App, Firebase can be used in several ways:

**Authentication:** Firebase Authentication can be used to implement user authentication, allowing Users to securely create accounts, sign in, and manage their profiles within the application.

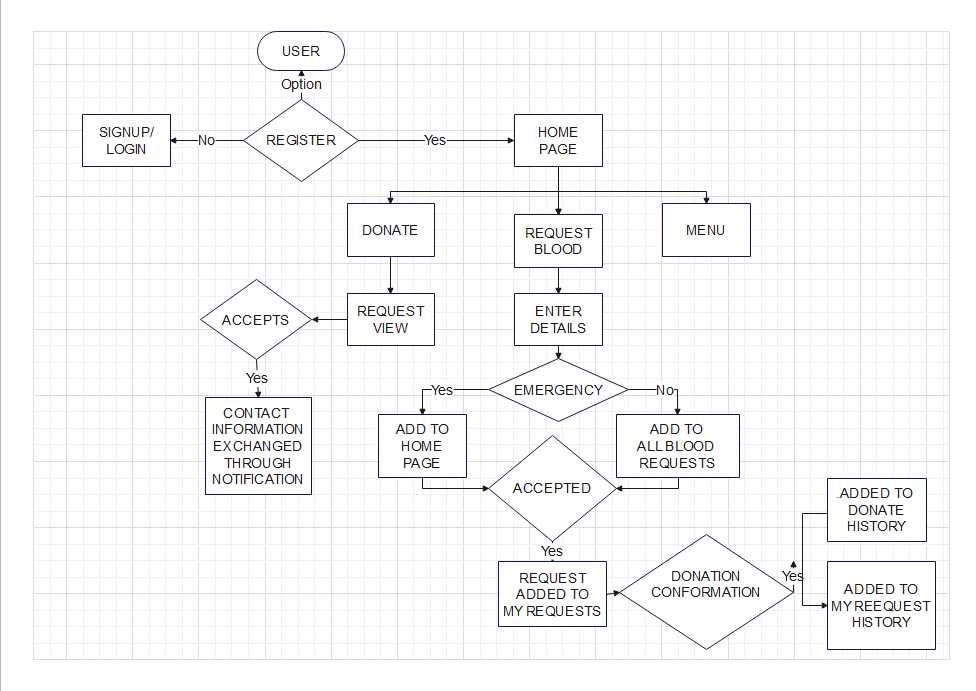
**Real-time Database:** Firebase Realtime Database is a cloud-hosted NoSQL database that allows developers to store and sync data between users in real-time. In the Blood Donation App, Firebase Realtime Database can be used to store donation requests, donor information, Donation History and other relevant data, ensuring that all users have access to the latest information.

**JSON Storage**: Firebase Realtime Database stores data as JSON. This makes understanding the data structure easier and allows you to use a standard data format widely used on the web.

By leveraging these Firebase services, we can enhance the functionality, performance, and user experience of the Blood Donation Application, while also simplifying development tasks and reducing infrastructure overhead.

**4.SYSTEM DESIGN**

**4.1 ARCHITECTURE:**



**Fig-4.1**

**SYSTEM ARCHITECTURE FOR BLOOD DONATING APPLICATION**

Following is the description for the Contents mentioned in the Fig-4.1,

**SIGNUP/LOGIN:** This is the initial Interface where a user can either sign up as a new user or log in if they already have an account.

**REGISTER:** If the user is new, then user is directly Navigated to update Profile to fill all the required details such as Name, contact number, Gender, Blood group, Date of Birth, Address, Email id and status (whether user is Active to Donate Blood or In-Active), after updating profile user is directed to Home Page of the Application.

**DONATE:** This option in the Home page allows users to donate blood. All the Blood requests are displayed in this DONATE page, User can accept the request if and only if user is Active and has same blood group.

**REQUEST BLOOD:** This option allows users to request for blood. They can fill the required details and request will be added to Blood requests page and to Home Page/Emergency requests (if it is Critical), where donors can see these requests and accept it.

**ALL BLOOD REQUESTS:** This option shows all existing blood requests.

**EXCHANGE OF CONTACT NUMBERS:** This is the means by which the application exchanges contact information between Donor and Requestor for proceeding further donation process.

**MY REQUESTS:** After the Request is Accepted by the Active donor, contact information is exchanged as mentioned above and a card view will be added to this My Request page personally in the Requestor’s account.

**DONATION CONFIRMATION**: requestor can confirm the Donation process whether Donation is successful or cancelled.Based on this information is personally added to Donate history of Donor’s Account and Request History of a Requestor’s Account.

**DONATE HISTORY:** This is the user's personal history of blood donations.

**REQUEST HISTORY**: This is the user's personal history of blood requests.

**MENU BUTTON:** The user is can select the main menu where they can access different options such as profile, My Requests, Request History, Donate History, Privacy Policy, Logout, Signup and login.

APPLICATIOBN MODULES SYSTEM FLOW

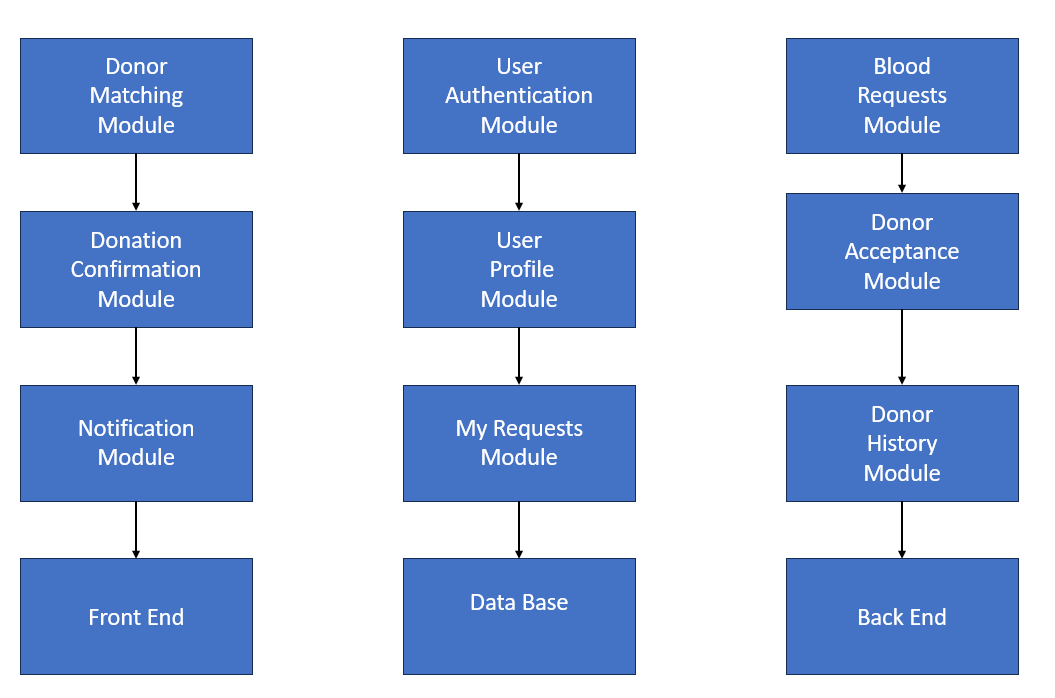


Fig-4.2

**5.IMPLEMENTATION**

Implementing a blood donation application using Android Studio involves several steps, including creating user authentication, managing blood donation requests, managing user profiles, Request History and Donation History. Below is an overview of how you can implement each of these components:

**User Authentication:**

Using Firebase Authentication to allow users to sign up and log in to the app securely.

Implementing registration and login Interfaces in this Blood Donating Application.

Using Firebase Authentication APIs to handle User authentication and manage user accounts.

**User Profiles:**

* Developing comprehensive user profiles to store essential details such as contact information, blood type, gender, date of birth and donation preferences.
* Creating intuitive interfaces for users to easily manage and update their profiles.
* Creating a database (Users) structure in Firebase Realtime Database to store user’s information.

**Request for Blood:**

* Creating a database (Requestors) structure in Firebase Realtime Database to store blood donation requests.
* Implementing Interfaces in application for requestors to create and submit blood donation requests.
* Storing request information such as patient name, blood type, required units, and contact details in the database.

**Blood Requests:**

• Implementing Interface in application for donors to display all the existing Blood

Requests.

• Retrieving Data from Requestors database, passing the required data to the Blood

Request’s Card View and implementing code to display the blood requests in

Blood Requests Page.

**Accepted Requests:**

• Creating a database (Accepted Requests) structure in Firebase Realtime Database to

store Accepted blood requests.

• Relocating the accepted requests data from Requestors database to Accepted Requests

Database.

• Switching Donor’s status as In-Active.

• Passing Requestor contact number, Name to the Donor and vice-versa through

Notification.

• Adding a Card view personally to My Requests page of Requestor’s Account to

Update the donation process where it is successful or cancelled.

**Donation History and Request History:**

• Creating a database (History) structure in Firebase Realtime Database to

store successfully completed donation details.

• Relocating the completed donation details data from Accepted Requests database to

History database.

• Implementing a Donation history and Request history features to track user’s past

contributions.

• Adding Donor name, Patient name, Blood type and Donation Place to donor’s and

requestor’s History after successful Donation completion.

**5.1 ENVIRONMENTAL SETUP**

**Development Environment:**

* Operating System: Windows.
* Integrated Development Environment (IDE): Android Studio.
* JDK (Java Development Kit): Version 8

**Version Control:**

Git: Version control system for managing project source code.

**Frontend Technologies:**

Android SDK: Software Development Kit for building Android applications.

XML: Markup language for designing user interfaces.

**Backend Techniques:**

a. Authentication and Authorization:

* Implementing secure authentication mechanisms to verify the identity of users accessing the application.
* Enforcing authorization rules to control access to sensitive data and functionalities based on user roles and permissions.

b. Data Storage and Management:

* Designing efficient data models and schemas to store user information, donation records, and other relevant data.
* Implementing data validation and integrity checks to maintain data consistency and accuracy.

**Backend Services:**

* Firebase: Backend-as-a-Service platform for database and user authentication,
* Firebase Realtime Database: Store and sync data in real time.
* Firebase Authentication: Securely authenticate users for Signup and login.

**Testing:**

Emulator: Built-in Android emulator in Android Studio for testing application on virtual

Devices.

Physical Device: Testing app on real Android devices for accurate performance evaluation.

**Release Build:**

Generate a signed APK or an application bundle for distribution.

**5.2 MODULES AND THEIR DESCRIPTION**

As in the Fig-4.2, System Flow of the Application is distributed into different modules as below:

1.User Authentication Module:

•Handles user registration and login.

•Manages user account information securely.

2.User Profile Module:

•Allows users to update their profiles.

•Stores donor-related information (for donors).

3.Blood Requests Module:

•Requestor submits a blood donation request.

•The request includes details such as patient name, contact number, blood

type, quantity needed, and location.

•Requests are stored in a database.

4.Donor Matching Module:

•Donor will same Blood group and with Active status can accept blood requests.

5.Donor Acceptance Module:

•Allows donors to accept a blood donation request.

•If accepted, donor information is shared with the requestor.

6.Donation Confirmation Module:

•After the donation, the requestor confirms the donation's success.

•Successful donations are recorded.

7. Request history Module:

•Displays a history of all successful blood donation requests made by the user.

8.History Module:

•Records the history of successful blood donations made by a donor and

requestor.

•Provides donors with a summary of their contribution.

9.Notification Module:

•Send notifications in the notification page of the application to notify users about donation confirmation.

10.Database:

•Stores user profiles, blood donation requests, accepted requests, donor history, and other relevant data.

11.Backend Server:

•Manages the business logic of the application.

•Handles communication between the frontend and the database.

12.Frontend:

•User interfaces for registration, login, profile management, viewing and accepting requests, confirming donations and donation history.

**6.TESTING AND RESULTS**

Considering a Test case where New User is operating our BLOOD DONATING APPLICATION.

**Login and Signup:**

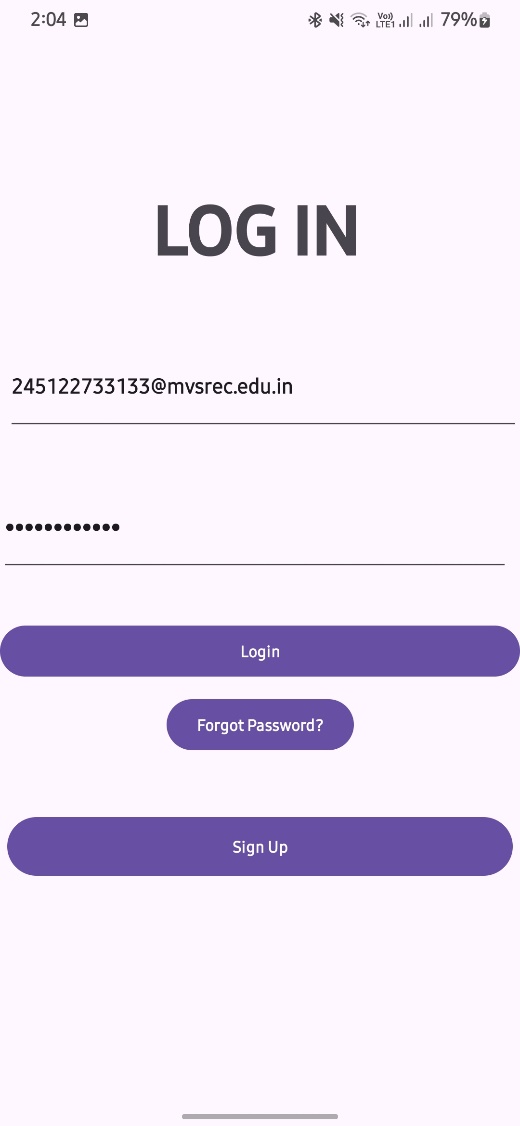


Fig-6.2 LOGIN PAGE

Fig-6.1 SIGNUP PAGE

Fig-6.2

Fig-6.1

When the User opens the Application, if the User is new then User is directed to signup/login page where User should enter Email id, Unique username and set password, After the successful signup Account we be created in Firebase Authentication.

As in the Figure 6.1 after filling the following requirements Account is created in the firebase Project with mentioned Email id as shown in Fig-6.1.1.

Email id: [245122733133@mvsrec.edu.in](mailto:245122733133@mvsrec.edu.in)

Username: abubakr\_133 Password: 245122733133

Fig-6.1.1

Fig-6.1.1 FIREBASE AUTHENTICATION

After Authentication User have to login as shown in Figure 6.2 and User is navigated to Update profile page.

**Updating Profile:**

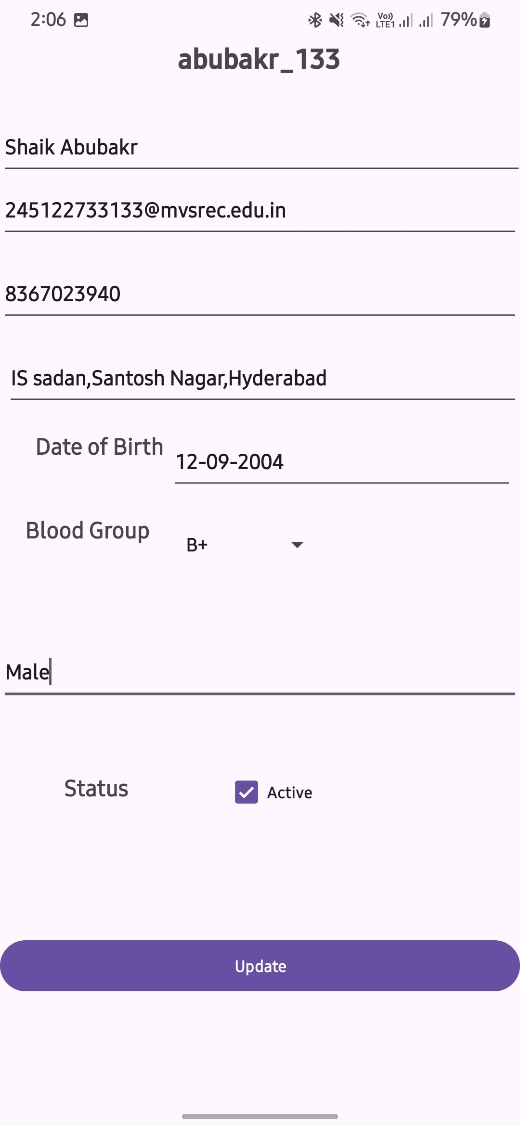
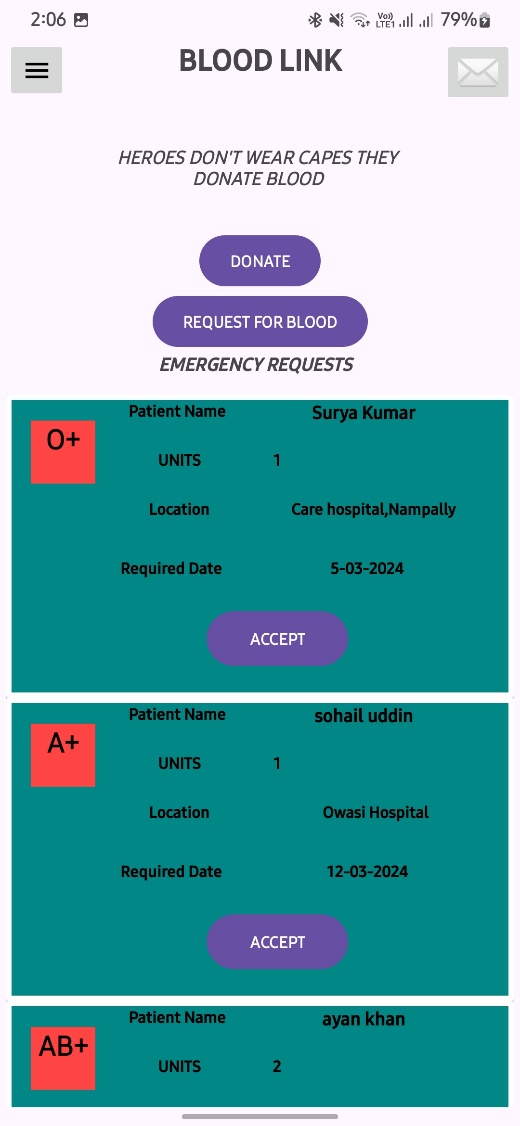


Fig-6.4 HOME PAGE

FIg-6.3 UPDATE PROFILE PAGE

Fig-6.3

Fig-6.4

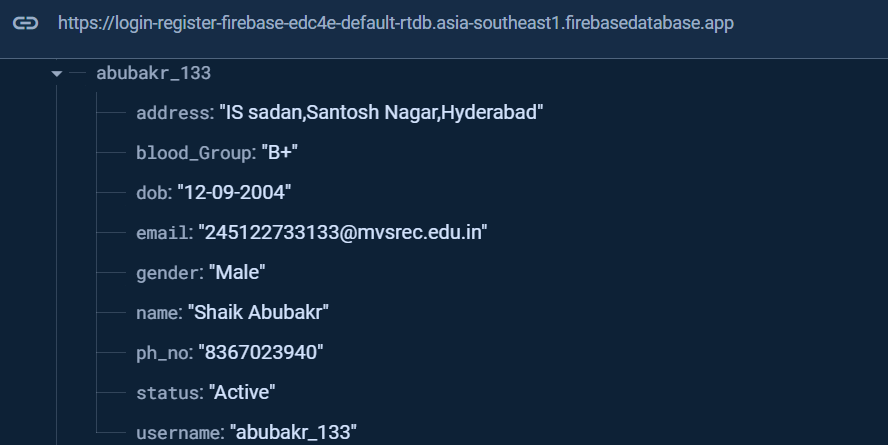
User must update all the Required information mentioned in the Figure 6.3, After updating Profile User is navigated to the Home Page of the Application and profile details will be added to Realtime database as shown in Figure 6.3.1.

Fig-6.3.1

Fig-6.3.1 USERS DATABASE

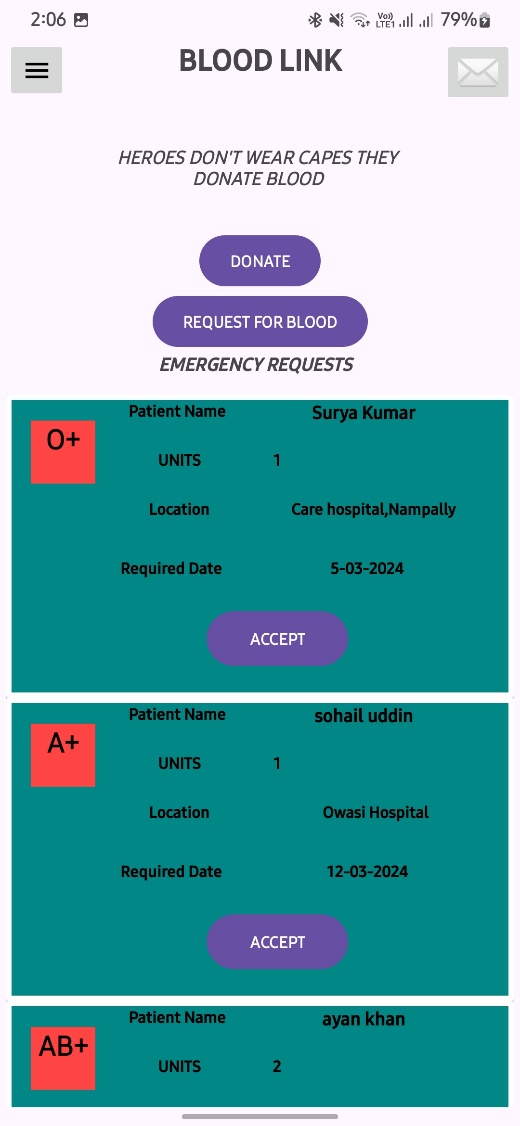
**Home page**:

Fig-6.5

Fig-6.5 UPDATED HOME PAGE

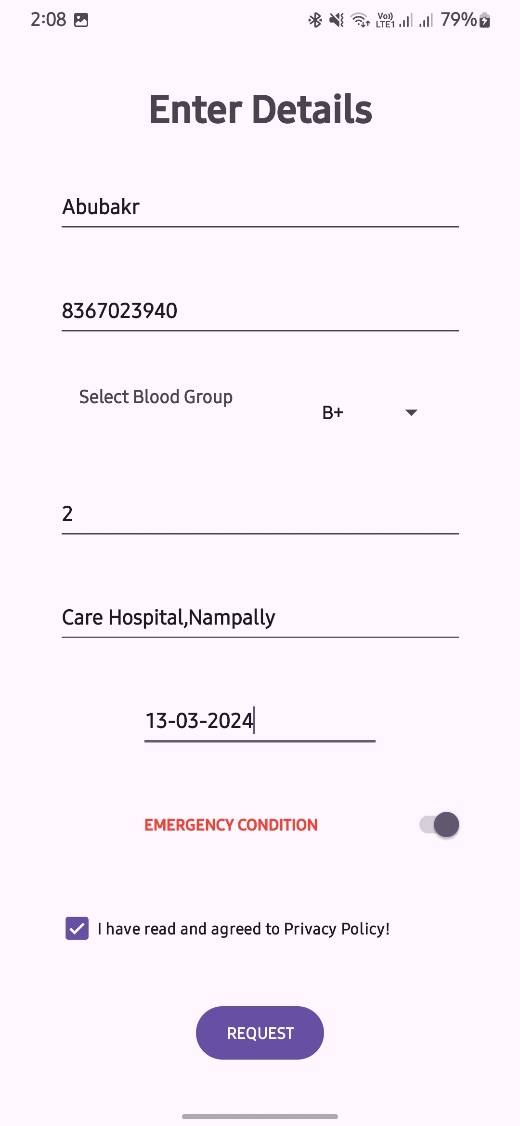


Fig-6.6 REQUEST FOR BLOOD PAGE

Fig-6.6

Now User can donate Blood or make Blood Request as per User’s Requirement.

If the User required blood then User can select the button “REQUEST FOR BLOOD” and he will directed to Fig-6.6 where user must enter all the required details and click on Request button after reading the Privacy policy. After Requesting Blood data will be stored in the database as shown in the Figure 6.6.1



Fig-6.6.1 REQUESTORS DATABASE

**Adding Request to the Blood Request Page:**

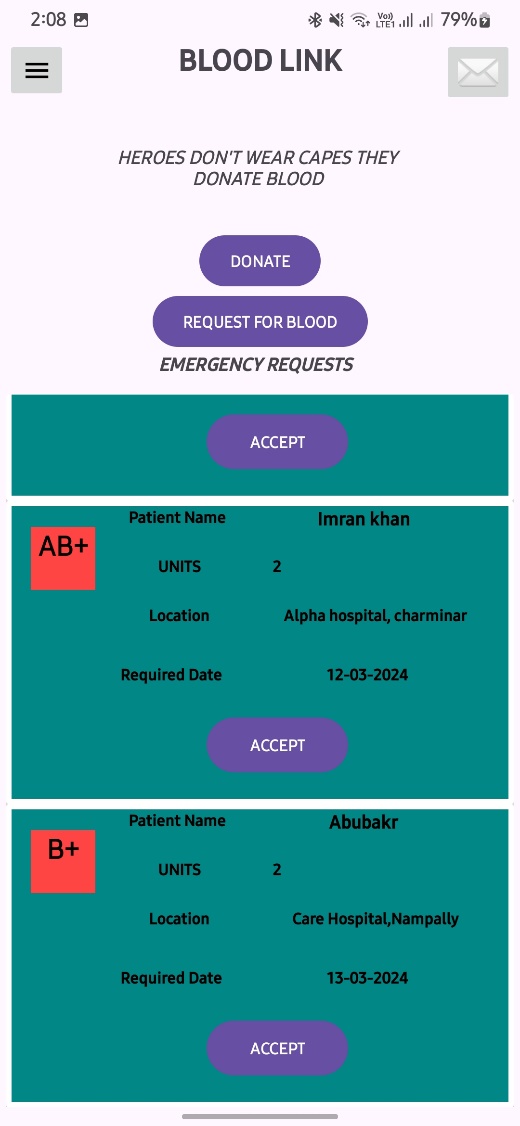
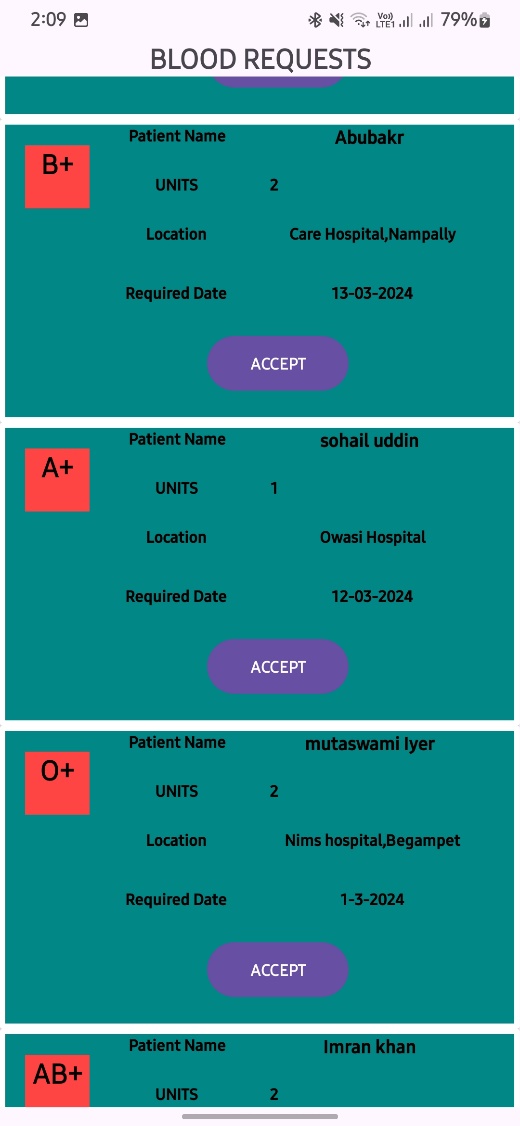


Fig-6.8

Fig-6.8 ALL BLOOD REQUEST PAGE

Fig-6.7

Fig-6.7 HOME PAGE WITH UPDATED REQUESTS

If the requirement is critical, then the Request will be added to Emergency Request of Home page as shown in Fig-6.7, and all the Request if they’re critical or not it will be added to All Blood Request page as shown in Fig-6.8.

**Donor Acceptance:**

If the User want to Donate blood User can Click on “DONATE” Button available in the Home Page and User can see all the Blood Requests available. If the user has the same Blood Group and User status is Active, then user can ACCEPT the Blood Request and contact information will be exchanged as shown in Fig-6.9, user’s status will be reset to Inactive, blood request details will be relocated from Blood Requests Database to Accepted Request Database as shown in the Fig-6.10

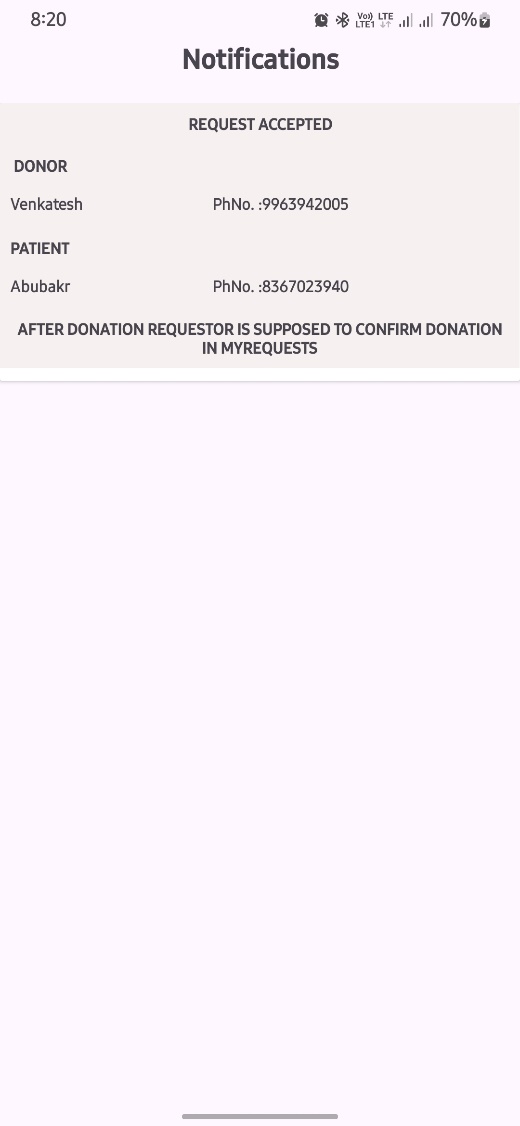
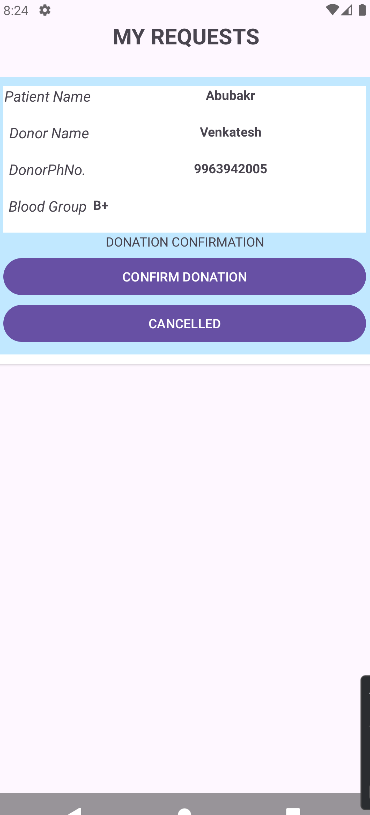


Fig-6.11

Fig-6.11 MY REQUEST PAGE

FIg-6.9 NOTIFICATION PAGE

Fig-6.9

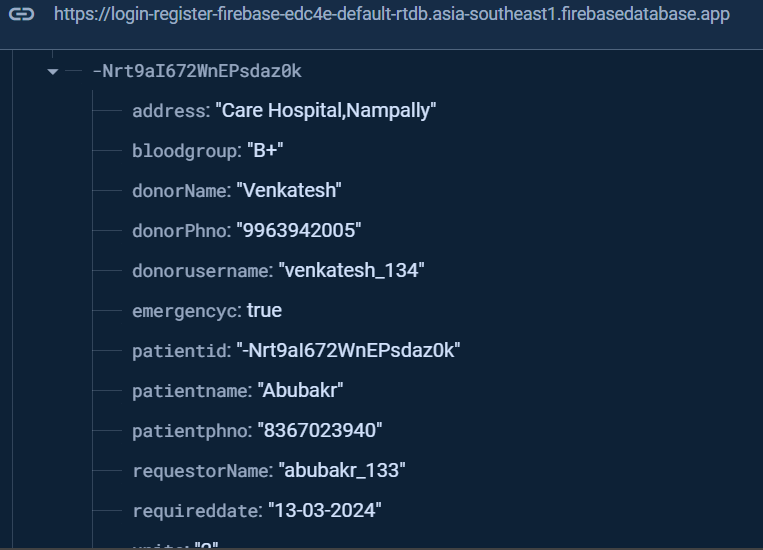
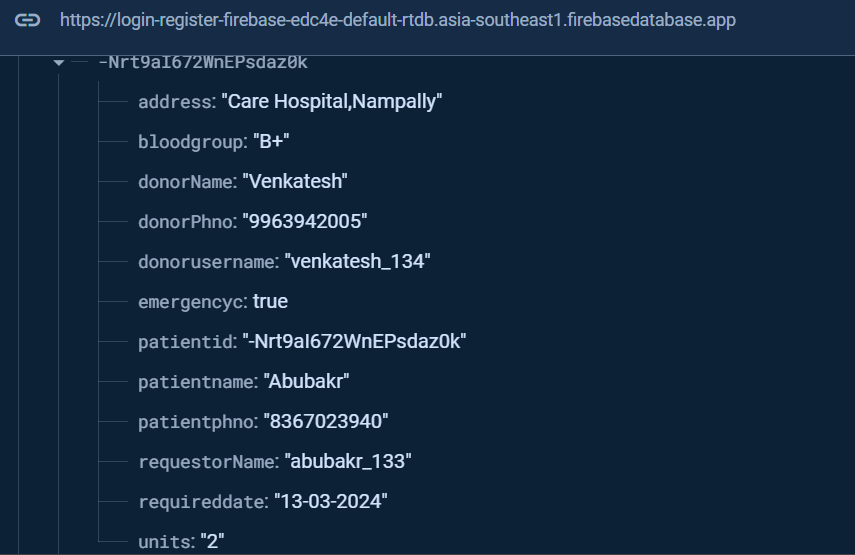
****

Fig-6.12

Fig-6.12 HISTORY DATABASE

Fig-6.10

Fig-6.10 ACCEPTED REQUEST DATABASE

**Donation Confirmation:**

A card view will be added personally to my request page of Requestor account where requestor can confirm the donation as shown in the fig-6.11. After successful donation all the donation details will be relocated from accepted blood request database to History database as shown in the figure-6.12 and this data will be personally added to the Donor’s History of a Donor Account and Request History of a Requestor’s Account as shown in the Fig-6.13 and Fig-6.14 respectively.

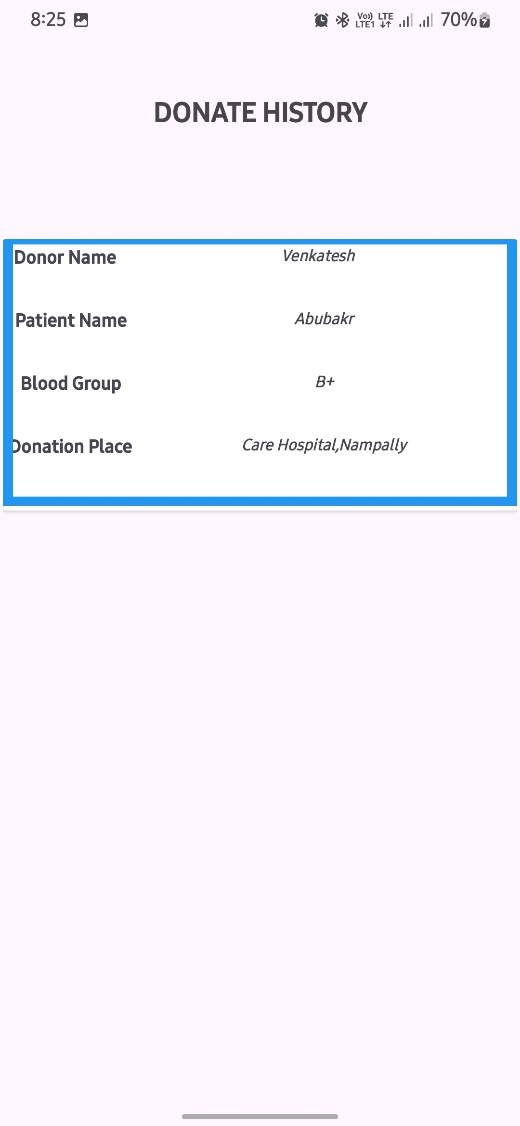
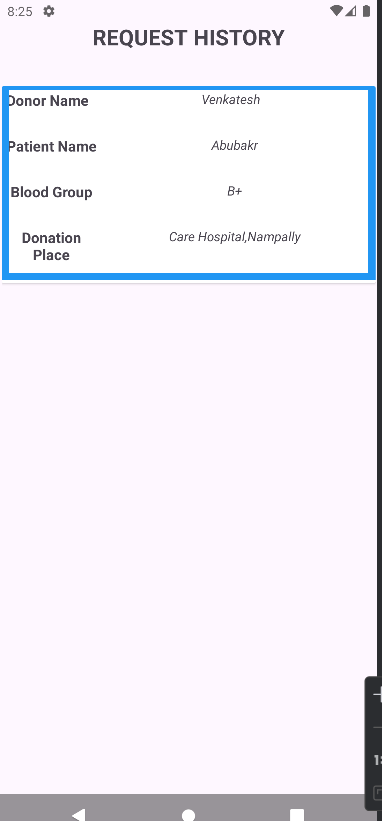


Fig-6.14

Fig-6.14 REQUEST HISTORY

Fig-6.13 DONOR HISTORY

**7.CONCLUSION& FUTURE ENHANCEMENTS**

**7.1 CONCLUSION**

The blood donation application aims to connect donors and requestors efficiently. Users can register, either as donors or requestors, submit and accept blood donation requests, and confirm successful donations. The application prioritizes user security with robust authentication and focuses on a user-friendly interface. Utilizing Android Studio, Kotlin, and relevant libraries, the application ensures seamless development. With thorough testing, a secure backend, and thoughtful UI/UX design, the application aims to make a meaningful impact in facilitating blood donations and saving lives.

**7.2 FUTURE ENHANCEMENT**

Firebase Cloud Messaging (FCM):

FCM allows us to send push notifications to their application users,

Even when the application is not actively in use. Using we can send notification of Emergency Requests to the Active Donor with same Blood Group.

Google Maps API:

location-based features, integrating the Google Maps API for Android. Using this we can Facilitate the donor by giving the route of destination from donor’s location.

Geolocation Services (GPS):

The Location API provides a set of classes and interfaces to access

location-based services, including GPS, network-based location and

other providers. Using this Requestor can Track the Location of Donor in critical case.

SMS API:

To send OTP (One-Time Password) messages for verification/conformation. This will Help us in verify the correct email id and contact number of the User.

Adding Securities, more constraints and Privacy to make application more Efficient and user-friendly.

**8. REFERENCE**

1. <https://play.google.com/store/apps/details?id=com.alen>.
2. <https://developer.android.com>.
3. <http://console.firebase.google.com>.
4. <https://stackoverflow.com>.
5. Android Studio Giraffe Essentials (Java Edition) Book

**9.APPENDIX**

**SAMPLE SOURCE CODE**

Taking Input Information for Blood Request and adding to Data base,

binding.rbButton.setOnClickListener(new View.OnClickListener() {  
 public void onClick(View v) {  
 String patientname=binding.patientnameText.getText().toString();  
 String patientphno=binding.phonenoText.getText().toString();  
 String bloodgroup=binding.brSpinner.getSelectedItem().toString();  
 String units=binding.unitsText.getText().toString();  
 String address=binding.addressText.getText().toString();  
 String requireddate=binding.editTextDate.getText().toString();  
 boolean privacy;  
 boolean emergency;  
 String patientid;  
 requestorname=FirebaseAuth.*getInstance*().getCurrentUser().getDisplayName().toString();  
 DatabaseReference dataref=FirebaseDatabase.*getInstance*().getReference("Requestors");  
 patientid=dataref.push().getKey();//uid  
 if(binding.emergencySwitch.isChecked())  
 {

emergency=true;

}  
 else

{  
 emergency=false;

}  
 if(binding.privacyCheck.isChecked())

{  
 privacy=true;

}  
 else

{  
 privacy=false;

}  
 try{  
 if(patientname.equals("") || address.equals("") || patientphno.equals("") || units.equals("") || bloodgroup.equals("")|| requireddate.equals(""))  
 Toast.*makeText*(requireContext(),"Please Fill All Details",Toast.*LENGTH\_SHORT*).show();  
 else if(privacy!=true)  
 {  
 Toast.*makeText*(requireContext(),"PrivacyPolicy not confirmed!",Toast.*LENGTH\_SHORT*).show();  
 }  
 else  
 {  
 dataref.child(patientid).setValue(new Requestordata(address,bloodgroup,requireddate,patientname,patientphno,units,patientid,requestorname,emergency));  
 //FirebaseAuth.getInstance().getCurrentUser().(new UserProfileChangeRequest.Builder().setDisplayName(username).build());  
 ScheduledExecutorService executer= Executors.*newScheduledThreadPool*(1);  
 Toast.*makeText*(requireContext(),"Request Added!",Toast.*LENGTH\_SHORT*).show();  
 Runnable task=()->{ Navigation.*findNavController*(requireView()).navigate(R.id.*action\_requestbFragment\_to\_HomeFragment*);  
 };  
 ScheduledFuture<?> Future=executer.schedule(task,1, TimeUnit.*SECONDS*);  
 }  
}  
 catch (NullPointerException e){  
 }  
 }

});