CONNECT-WITH-BLOOD

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

NAME OF THE STUDENT

SHAIK MOHAMMED YASAR

Registration number

12111190

CSE 226

ANDROID APP DEPLOYMENT

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month NovemberYear (2024)

TABLE OF CONTENTS (16 Bold)

Sl.No CONTENTS PAGE NO.

1. Introduction

In today's fast-paced world, access to healthcare resources plays a pivotal role in saving lives. Among these resources, the availability of blood during emergencies is one of the most critical factors. Whether it's an accident, a surgery, or a medical condition requiring a transfusion, the timely availability of blood can mean the difference between life and death. However, finding donors quickly and efficiently has always been a challenge. To address this problem, the **LifeBlood** app was developed as a modern and reliable solution for connecting blood donors with recipients.

The **LifeBlood** app is an innovative Android application designed to streamline the process of blood donation and matching donors with those in need. It combines location-based services, a robust SQL database, and user-friendly features to ensure that users can find the right donors at the right time. The app is aimed at solving several key problems:

1. **Lack of Access:** In many regions, individuals often struggle to find blood donors in their area during emergencies.
2. **Time Sensitivity:** The process of reaching out to friends, family, or social networks for donors can be time-consuming, which may delay critical treatment.
3. **Data Accuracy:** Existing methods often rely on outdated donor lists, which may include unavailable or inactive donors.

By addressing these challenges, **LifeBlood** aims to save lives and make the blood donation process more efficient and reliable.

Purpose of the Application

The core purpose of the LifeBlood app is to create a platform where donors and recipients can seamlessly interact. It allows donors to register their details, making their information accessible to recipients searching for a matching blood type in a specific location. The app serves as a bridge between those willing to help and those in need, fostering a sense of community and mutual support.

In addition to its primary function of donor-recipient matching, the app also emphasizes the importance of user experience. Through features such as session management, a personalized welcome screen, and visually appealing layouts, LifeBlood ensures that users find the app intuitive and engaging.

Problem Statement

Finding blood donors during emergencies often involves multiple hurdles, including lack of communication, unavailability of donors, and delays caused by manual processes. Traditional approaches, such as reaching out through phone calls or relying on blood banks, are not always efficient or successful. This inefficiency can result in lost opportunities to save lives.

LifeBlood directly addresses these issues by:

- Offering a centralized database of registered donors.
- Using real-time search functionality to find donors based on location and blood type.
- Providing an organized and user-friendly interface for both donors and recipients.

Key Features of LifeBlood

The LifeBlood app boasts several key features that differentiate it from other blood donation platforms:

1. **Session Management:**
    - The app remembers user sessions, ensuring a seamless experience for returning users.
    - If the user is already logged in, the app greets them with a personalized message, such as "Welcome back, [Name]."
    - Users who are not logged in are redirected to the login or registration page.
2. **Donor Registration:**
    - Individuals can register as donors by providing essential details like name, age, gender, blood group, contact information, and location.
    - The information is securely stored in a **SQLite database**, ensuring accuracy and reliability.
3. **Recipient Search Functionality:**
    - Recipients can search for donors by specifying their location and required blood group.
    - The app uses efficient SQL queries to fetch matching donor information from the database.
    - Results are displayed in a visually appealing **RecyclerView** with **CardView** layouts.
4. **Splash Screen Animation:**
    - The app opens with an engaging animation, symbolizing the importance of blood donation.
    - During this animation, the app checks for an active user session in the background.
5. **Modern Design:**
    - The app features a clean and intuitive user interface, designed to provide a smooth and enjoyable experience.
    - A visually appealing blood group picker makes donor registration easier.

Technology Stack

The development of the LifeBlood app is based on modern Android development practices, including:

- **Android Studio**: The primary IDE for app development.
- **SQLite Database**: For persistent data storage and efficient query handling.
- **RecyclerView and CardView**: To display donor search results in an organized and visually appealing manner.
- **Location Services**: To fetch the user's current location for accurate donor-recipient matching.
- **Jetpack Libraries**: For enhanced functionality and adherence to modern development standards.

Workflow of the Application

The LifeBlood app follows a well-structured workflow to ensure efficiency and ease of use:

1. **Splash Screen:** When the app is launched, users are greeted with an animated splash screen. Simultaneously, the app checks for an active user session using SQL queries.
2. **Session Validation:**
    - If a session is active, the app retrieves the user's name and displays the **Main Activity** with a personalized welcome message.
    - If no session is active, the app redirects the user to the **Login Activity**.
3. **Main Activity:**
    - The Main Activity contains two buttons: **Save Life** and **Receive Life**.
    - For active sessions, the **Save Life** button is disabled, as the user is already registered as a donor.
    - For inactive sessions, clicking **Save Life** redirects to the **Donate Activity** for donor registration.
4. **Donate Activity:** Users can fill out a form with their details, which are saved in the **donors** table of the SQLite database. The session is then activated, and the user is redirected to the Main Activity.
5. **Receiver Activity:**
    - Users can search for donors by entering their location and required blood group.
    - The app fetches matching results from the database and displays them in a RecyclerView.

Significance of LifeBlood

The LifeBlood app is more than just a tool—it's a lifesaving platform that promotes the spirit of giving and community. It reduces the time and effort required to find blood donors, making it an invaluable resource during emergencies. By encouraging more individuals to register as donors, the app also contributes to the creation of a larger and more accessible donor database.

In a broader sense, LifeBlood embodies the idea of using technology for social good. It demonstrates how simple yet powerful technological solutions can have a profound impact on people's lives. Through its user-friendly design and efficient functionality, the app aims to inspire more people to participate in blood donation, ultimately saving countless lives.

2. Modules or Activity Explanation & Screenshots

The **LifeBlood** application consists of various interconnected modules and activities, each designed to perform a specific task within the app's workflow. Below is a detailed explanation of each activity and module:

i) Splash Activity

Purpose:
The Splash Activity is the starting point of the app, displaying an engaging animation (a blood drop falling into a rectangular box) for 10 seconds. During this time, the app checks for an active user session in the database.

Key Features:

- SQL query validation to check if the user is already logged in.
- If the user is logged in, their name is fetched from the session database, and they are redirected to the **Main Activity** with a personalized welcome message.
- If no active session is found, the user is redirected to the **Login Activity**.

Backend Logic:

The app queries the **session** table in the SQLite database to determine whether an active session exists.If the session is active:

```
if (databaseHelper.isUserLoggedIn()) {

        navigateToMainActivity()

    } else {

        navigateToLoginActivity()

    }
```

"SELECT $COLUMN_IS_LOGGED_IN FROM $TABLE_SESSION "

If no session is active, then log in.

ii) Login Activity

Purpose:
The Login Activity allows users to log into the app using their credentials. If the user logs in successfully, their session is activated and stored in the **session** table.

Key Features:

- Secure login system to authenticate users.
- Session activation after successful login.
- Navigation to the **Main Activity** upon successful login.

Backend Logic:

```
if (databaseHelper.validateLogin(username, password)) {

        Toast.makeText(this, "Login Successful",
        Toast.LENGTH_SHORT).show()



            val intent = Intent(this,MainActivity ::class.java)

            startActivity(intent)

            finish()



        }
```

"SELECT * FROM $TABLE_DONORS WHERE ($COLUMN_EMAIL = ? OR $COLUMN_PHONE = ?) AND $COLUMN_Password = ?",

On successful login, the user's name is saved in the session database:

```
getDonorNameByIdentifier(email)

        if (donorName != null) {

            setUserLoggedIn(donorName)

        }
```

iii) Main Activity

Purpose:
The Main Activity serves as the home screen of the app. It contains two primary options:

1. **Save Life** – For registering as a blood donor.
2. **Receive Life** – For finding blood donors based on location and blood type.

Key Features:

1. Personalized greeting: If a session is active, the user is greeted with a message like *"Welcome back, [Name]"*.
2. Two buttons: **Save Life** and **Receive Life**.

```
donateBloodCard.setOnClickListener {

    if (databaseHelper.isUserLoggedIn()) {

        startActivity(Intent(requireContext(),
                User_already_Loginned::class.java))

    } else {

        startActivity(Intent(requireContext(), DonateActivity::class.java))

    }

}
```

If the user is already registered as a donor (active session), the **Save Life** button is disabled.

iv) Donate Activity

Purpose:
The Donate Activity allows users to register as blood donors by filling out a form with their details.

Key Features:

- Input fields include:
  - Name
  - Age
  - Gender
  - Phone Number
  - Email
  - Blood Group (with a modern picker interface)
  - Location (fetched automatically or entered manually)
- All details are saved in the **donors** table of the SQLite database.
- The user's session is activated after successful registration, and they are redirected to the **Main Activity**.

Backend Logic:

Donor information is saved in the database

```
        val donor = Donor(name,password, age.toString(), gender, phone, email, city,
bloodGroup)

            val result = dbHelper.addDonor(donor)

            return if (result != -1L) {

                dbHelper.setUserLoggedIn(name)

                Toast.makeText(this, "Donor information saved successfully!",
Toast.LENGTH_SHORT).show()

                true

            } else {

                Toast.makeText(this, "Failed to save donor information",
Toast.LENGTH_SHORT).show()

                false

            }
```

The **Receive Life** button is always active.

v) Receive Activity

Purpose:
The Receive Activity helps recipients find blood donors by allowing them to search based on location and blood type.

Key Features:

- Input fields for:
  - Location
  - Blood Group
- Matching donor information is fetched from the **donors** table and displayed in a **RecyclerView** using **CardView**.
- Results are dynamically displayed in real-time.

Backend Logic:

```
private fun findData() {

    val city = tvCity.text.toString()

    val bloodGroup = getSelectedBloodType()



    if (city.isNotBlank() && bloodGroup.isNotBlank()) {

        val donors = dbHelper.getDonorsByCityAndBloodGroup(city, bloodGroup)

        if (donors.isNotEmpty()) {

            val donorDetails = ArrayList(donors.map { donor ->

                mapOf(

                    "name" to donor.name,

                    "age" to donor.age,

                    "gender" to donor.gender,

                    "phone" to donor.phone,

                    "email" to donor.email,

                    "city" to donor.city,

                    "bloodGroup" to donor.bloodGroup

                )

            })



            val intent = Intent(this, sample_show::class.java)

            intent.putExtra("donorData", donorDetails)

            startActivity(intent)

        } else {

            Toast.makeText(this, "No donors found for $bloodGroup in $city.", Toast.LENGTH_SHORT).show()

        }

    } else {

        Toast.makeText(this, "Please enter both city and blood group.", Toast.LENGTH_SHORT).show()

    }

}
```

SQL Logic:

```
 fun getDonorsByCityAndBloodGroup(city: String, bloodGroup: String): List<Donor> {

       val donors = mutableListOf<Donor>()

       val db = readableDatabase

       val cursor = db.rawQuery( "SELECT * FROM $TABLE_DONORS WHERE $COLUMN_CITY = ?
AND $COLUMN_BLOOD_GROUP = ?", arrayOf(city, bloodGroup))



       if (cursor.moveToFirst()) {

          do {

             val donor = Donor(




                cursor.getString(cursor.getColumnIndexOrThrow("name")),

                cursor.getString(cursor.getColumnIndexOrThrow("password")),

                cursor.getString(cursor.getColumnIndexOrThrow("age")),

                cursor.getString(cursor.getColumnIndexOrThrow("gender")),

                cursor.getString(cursor.getColumnIndexOrThrow("phone")),

                cursor.getString(cursor.getColumnIndexOrThrow("email")),



                cursor.getString(cursor.getColumnIndexOrThrow("city")),

                cursor.getString(cursor.getColumnIndexOrThrow("blood_group"))

             )

             donors.add(donor)

          } while (cursor.moveToNext())

       }

       cursor.close()

       db.close()

       return donors

   }
```

The return data stored and saved by the help of intent and display with the help of the adapter

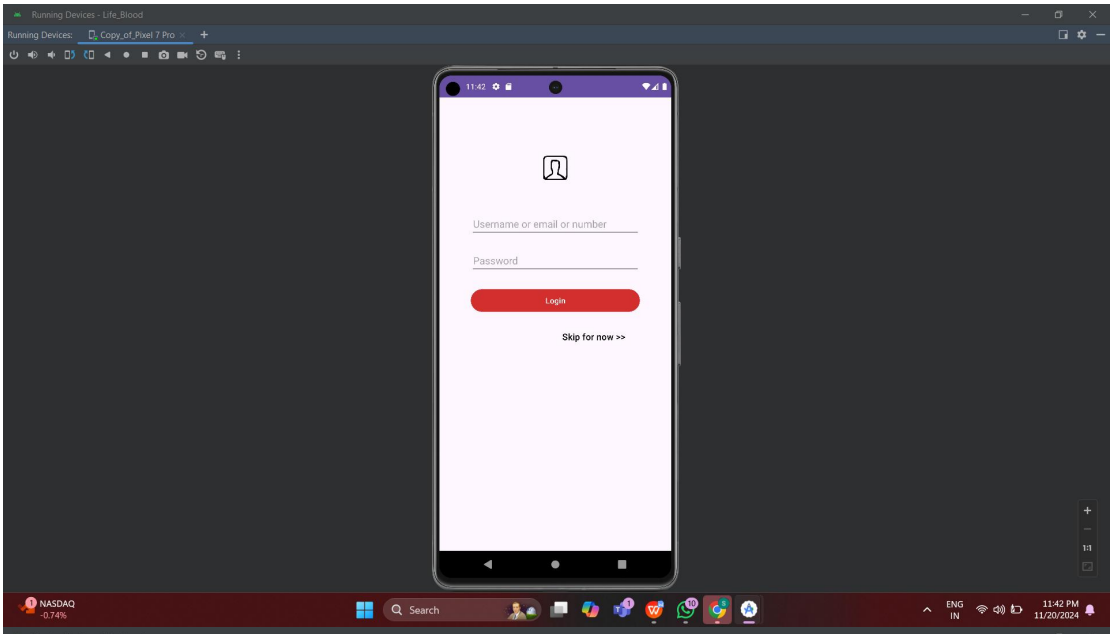
iv) Adapter Activity

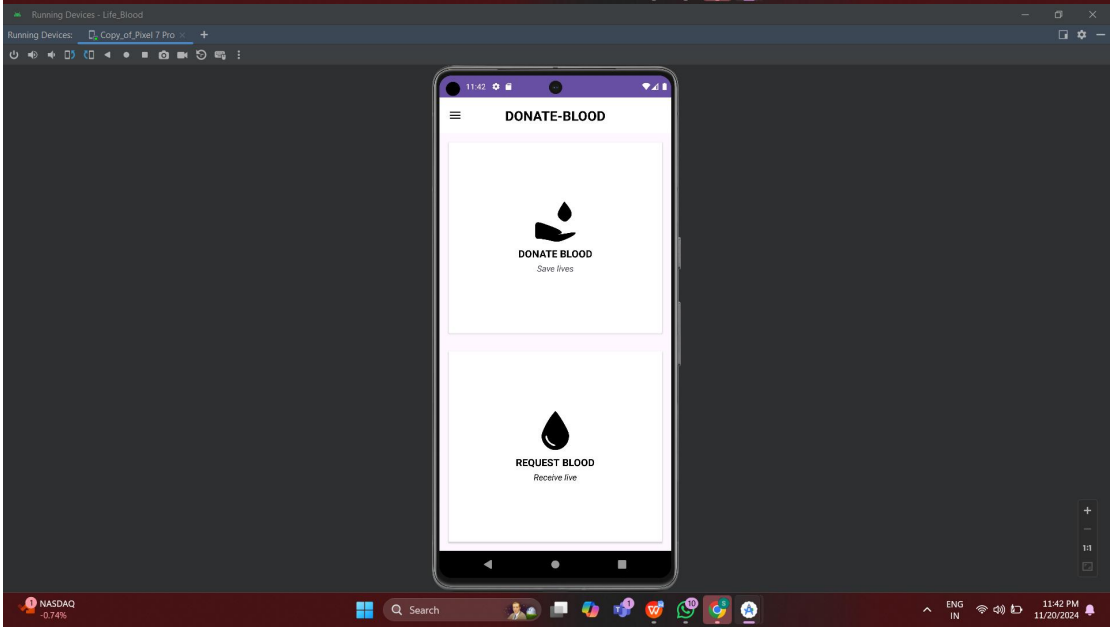
Purpose:

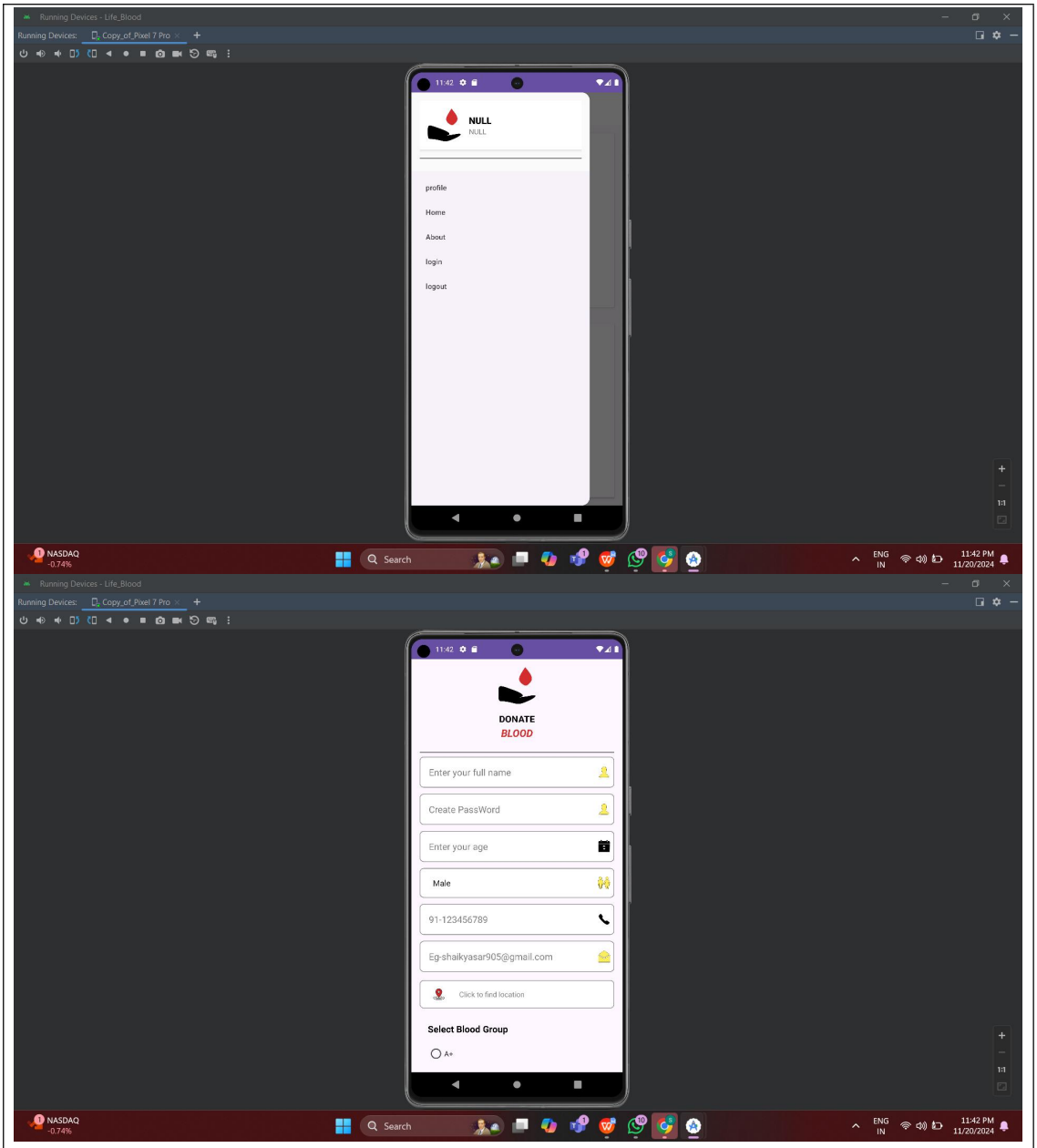
To display the result with the help of recycler view.

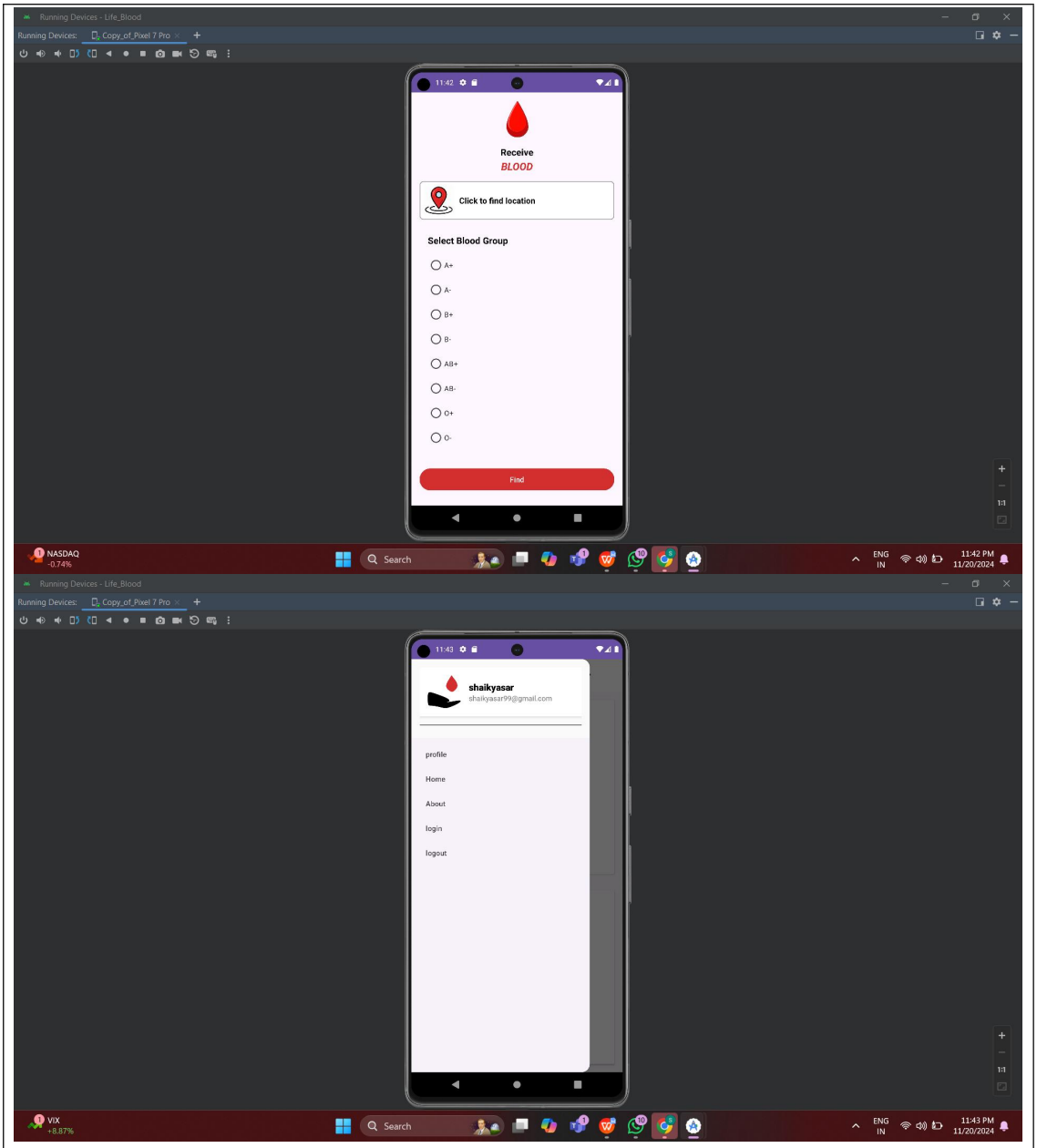
```
holder.nameTextView.text = donor["name"]
holder.cityTextView.text = donor["city"]
holder.phoneTextView.text = donor["phone"]
holder.emailTextView.text = donor["email"]
holder.bloodTextView.text = donor["bloodGroup"]
```

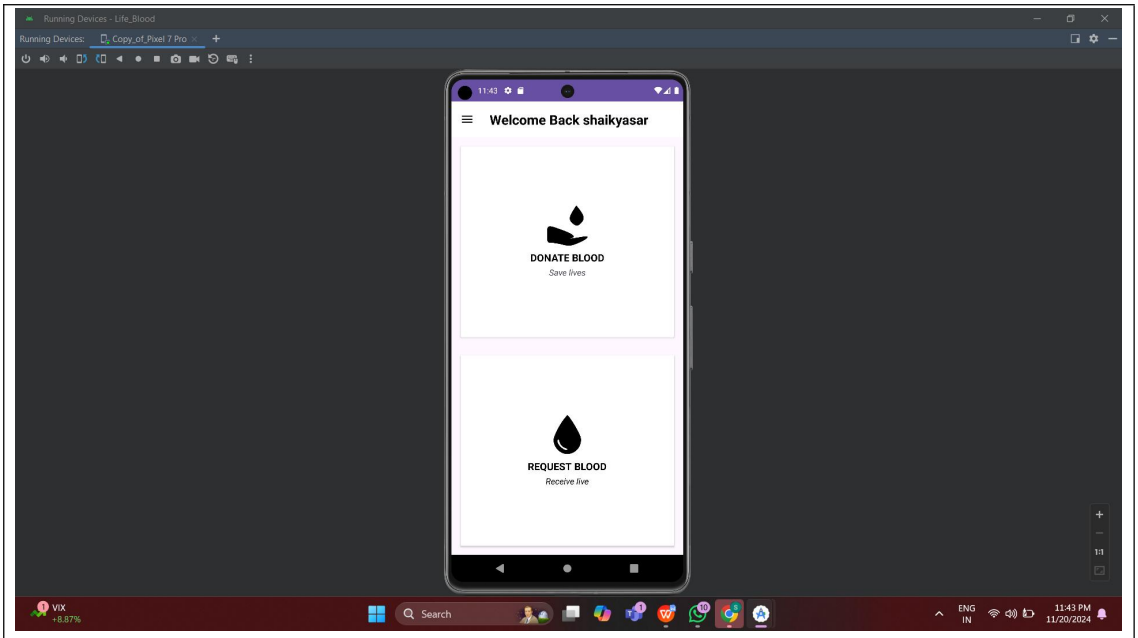
4. Emulator Screenshots

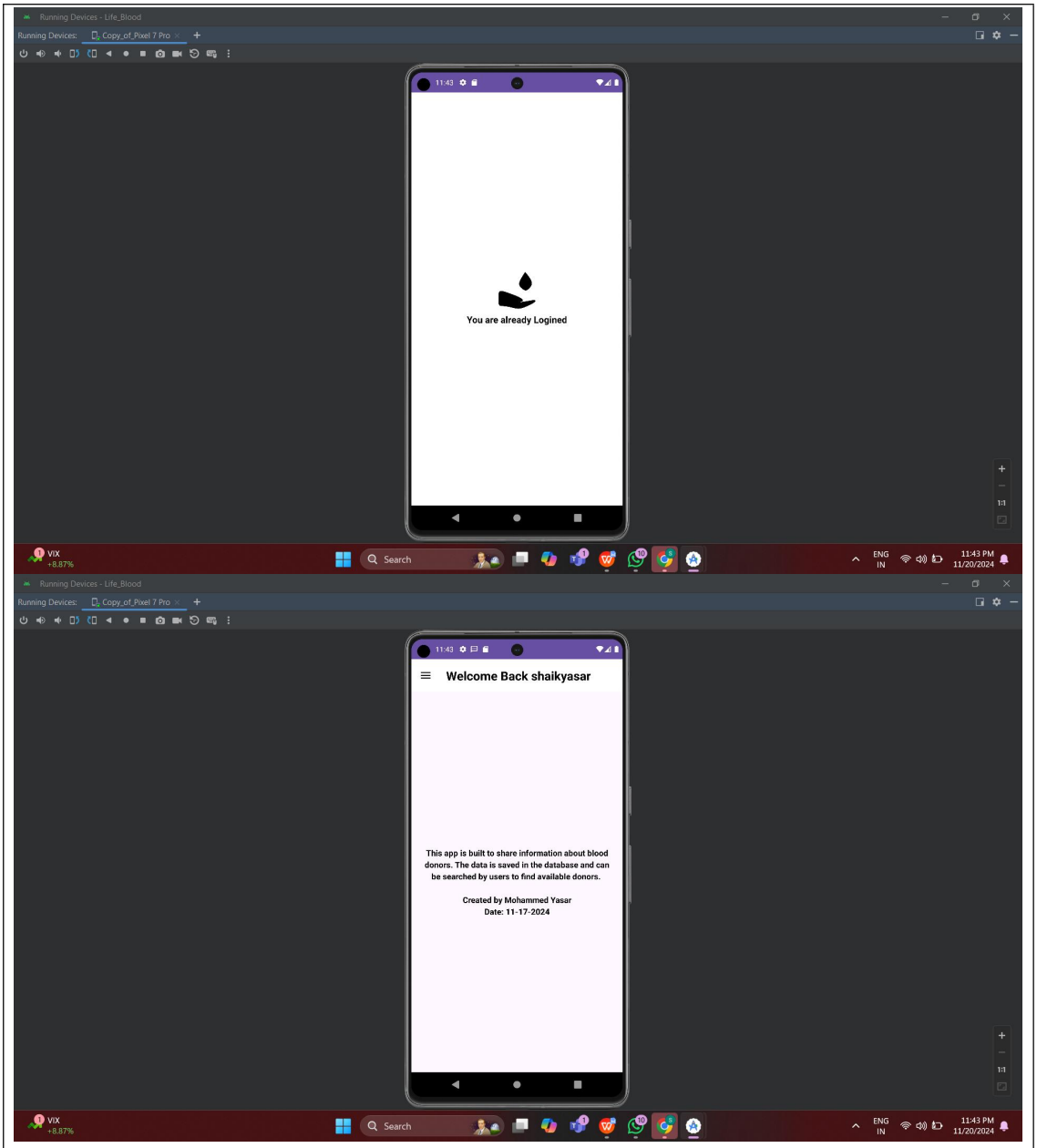





Running Devices - Life_Blood
Running Devices: Copy_of_Pixel 7 Pro
11:42
NULL
NULL
profile
Home
About
login
logout
Search
ENG
IN
11:42 PM
11/20/2024
Running Devices - Life_Blood
Running Devices: Copy_of_Pixel 7 Pro
11:42
DONATE
BLOOD
Enter your full name
Create PassWord
Enter your age
Male
91-123456789
Eg-shaikyasar905@gmail.com
Click to find location
Select Blood Group
A+
Search
ENG
IN
11:42 PM
11/20/2024


Running Devices - Life_Blood
Running Devices:
Copy_of_Pixel 7 Pro
11:42
Receive
BLOOD
Click to find location
Select Blood Group
A+
A-
B+
B-
AB+
AB-
O+
O-
Find
NASDAQ
-0.74%
Search
ENG
IN
11:42 PM
11/20/2024
Running Devices - Life_Blood
Running Devices:
Copy_of_Pixel 7 Pro
11:43
shaikyasar
shaikyasar99@gmail.com
profile
Home
About
login
logout
VIX
+8.87%
Search
ENG
IN
11:43 PM
11/20/2024


Running Devices - Life_Blood
Running Devices:
Copy_of_Pixel 7 Pro
11:43
Welcome Back shaikyasar
DONATE BLOOD
Save lives
REQUEST BLOOD
Receive live
Search
ENG
IN
11:43 PM
11/20/2024


Running Devices - Life_Blood
Running Devices:
Copy_of_Pixel 7 Pro
11:43
You are already Logined
Search
ENG
IN
11:43 PM
11/20/2024
Running Devices - Life_Blood
Running Devices:
Copy_of_Pixel 7 Pro
11:43
Welcome Back shaikyasar
This app is built to share information about blood donors. The data is saved in the database and can be searched by users to find available donors.
Created by Mohammed Yasar
Date: 11-17-2024
Search
ENG
IN
11:43 PM
11/20/2024

5. Conclusion & Future Scope

The **LifeBlood** application successfully addresses the critical need for an efficient and user-friendly blood donation and retrieval platform. By integrating essential features like donor registration, real-time donor search, and session management, the app serves as a comprehensive solution for connecting blood donors and recipients.

Conclusion

User-Centric Design:
The app's simple and modern user interface, including animated transitions, personalized greetings, and intuitive navigation, ensures a seamless user experience. The ability to easily toggle between donor and recipient functionalities empowers users to actively participate in life-saving efforts

Robust Backend Support:
With the use of SQLite for persistent data storage, the app maintains critical information about donors and user sessions efficiently. The robust querying and data retrieval mechanisms ensure accurate and fast results

Session Management:
The inclusion of session handling simplifies user workflows, allowing returning users to bypass redundant login processes while ensuring secure data access. It also enhances the user experience by personalizing the main activity based on active session data.

Scalability:
The modular structure of the app allows for the addition of future features without disrupting the existing functionality. This ensures the app can grow alongside the evolving needs of its users.

Impact:
The app effectively bridges the gap between blood donors and recipients, making the process of saving lives faster and more efficient. It reduces the time spent searching for donors and encourages more people to register as donors.

Future Scope

Integration with Cloud Databases:
To support larger datasets and multiple users simultaneously, the app can be upgraded to use cloud-based databases like Firebase or AWS.

Real-Time Notifications:
Implementing push notifications can alert donors when someone nearby requires their blood type or inform recipients of newly registered donors in their area.

GPS-Based Donor Matching:
Enhancing the location search feature with real-time GPS tracking can make it easier to find the nearest available donor.

User Authentication Enhancements:
Incorporating advanced authentication methods like biometric login (fingerprint or face recognition) can further enhance security and convenience.

Data Analytics Dashboard:
A dashboard for administrators could provide insights into donation trends, popular blood types, and areas with high demand, helping streamline blood donation efforts.

Multi-Language Support:
Adding support for regional languages can make the app more accessible to a broader audience.

Collaboration with Healthcare Organizations:
Partnering with hospitals and blood banks can help expand the app's reach and credibility.

Offline Functionality:
Allowing users to access donor details offline can be useful in areas with limited internet connectivity.

In conclusion, the **LifeBlood** app is a vital tool that combines technology and humanity to promote blood donation and save lives. With its current features and potential for future enhancements, it can become a game-changer in the domain of blood donation systems, fostering a more connected and responsive healthcare ecosystem.

6. Github:

https://github.com/shaikyasar/LIFE_BLOOD