MINI PROJECT
(2020-21)

CoChat - A Messaging Application

MID-TERM REPORT

Institute of Engineering & Technology

Submitted by-

Rahul Saxena	**(181500538)**
Prankit Agarwal	**(181500481)**
Prakhar Agarwal	**(181500469)**
Aman Vikrant Garg	**(181500080)**
Ishita Goel	**(181500287)**

Supervised By: -
Mr. Neeraj Khanna
Technical Trainer
Department of Computer Engineering & Applications

Contents

Abstract

As we all know in this today's world where technology is changing really quick. Everyone is switching from desktops to mobile devices. Those days are gone when people used to send text messages and paid for it, also there is no much time to waste to call via PC's video calling softwares. As no one can carry PC's and laptops with themselves everywhere. Nowadays everyone gets switched to mobile. People are engaging more in mobile applications rather than using PC's softwares.

So, keeping all this in mind, we decided to make a *messaging application*. So that the user has to not waste his precious time in downloading or keep waiting for others as what happens in desktop softwares. He/She can get in touch with anyone in the world using this application from anywhere. So the idea behind this is simple, it's just to connect people at once so that they can share their precious moments with their loved ones.

Introduction

General Introduction to the topic-

We made a messaging application named CoChat. As we all know that there are multiple messaging applications available in the market today. So what is different in our application as our name says CoChat means Collective Chat, basically our main idea behind making this application is just to provide users a hassle free experience. No ads, No privacy concern. We just want to provide a user- friendly environment.

This awesome messaging application will help users to simplify their needs.

Area of Computer Science-

Android is an open source software. Any phone manufacturer can use android without an expensive license fee from google because it is open for all. Manufacturers can modify Android without restriction, allowing it to fit the device they are making total freedom. The ability to run tens of thousands of apps is another big incentive.
Android is the present trend and future.

About CoChat Application: -

Before starting to develop a messaging app, let's talk about the essential features of a messaging application.

- ❏ Authentication in chat apps can be implemented in various ways: via phone number, email, or social media profiles. Most instant messaging apps require users to sign in with a phone number.
- ❏ Customising profiles helps users express their individuality. They might change names and nicknames, background colours, patterns, and fonts, or choose an avatar photo from the camera roll or by taking a photo instantly. In a lot of apps, people can see user status, namely when people last used the messenger app, who's online, and when someone is typing.

- ❏ Instant messaging feature operates over the internet, but if users go offline they can receive all the messages they miss as soon as they get back online. Also, messages can have statuses (delivered/failed/seen/edited).

Hardware Requirements-

- Memory [2GB RAM (or higher)]
- Android OS

Software Used-

- Google Firebase
- Android Studio

Objective

From yahoo chat that started in those early days of the internet to Snapchat, WhatsApp and Facebook's highly sophisticated chat messenger, online Chat applications have evolved dramatically over the last decade.

With multi-tasking mechanisms playing the major focus, today's chat apps are explored globally by billions of users for both personal as well as commercial fulfilment.

At the heart of these chat app innovation lies fascination for mobile technology that was earlier seen as a fleeting fad. The reason why chat apps are exercised with great sincerity is the fact that there is a subtle opportunity to transform chat application services into a major marketing. As now more than 70% of total visits on top digital news websites come from mobile devices, the need for developing mobile-specific chat apps has escalated.

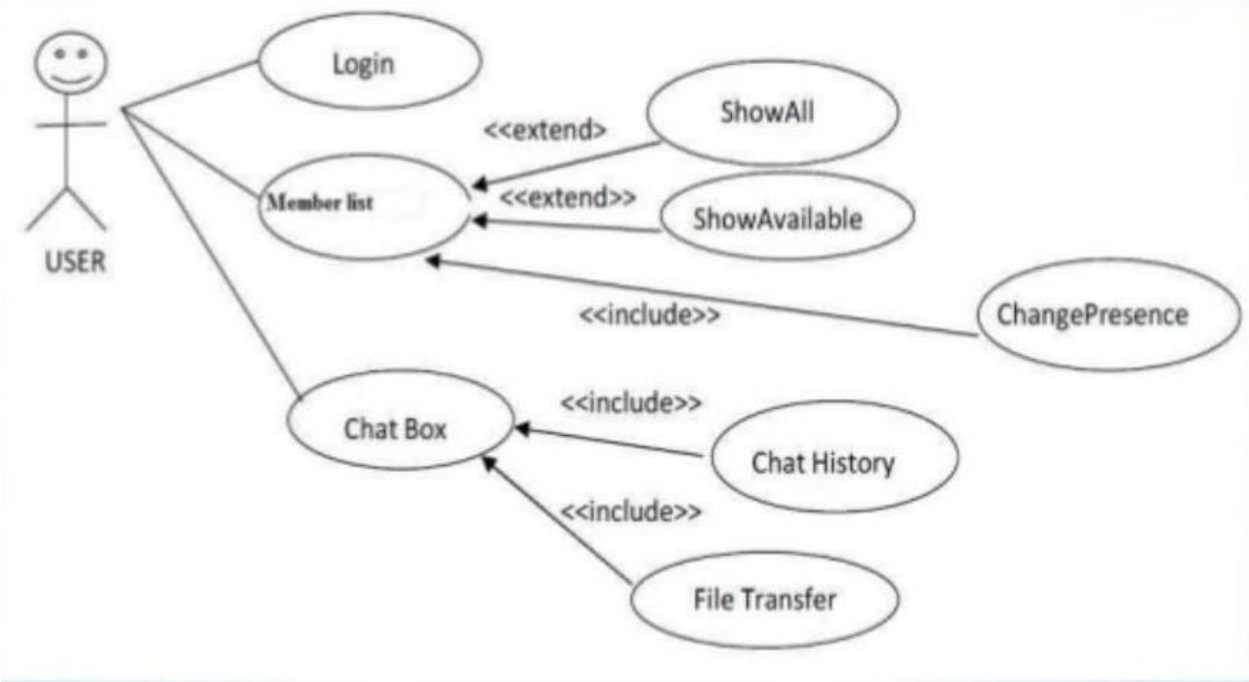
A lot of traffic and referrals observed on news organisations belong to social platforms, which translates into a great opportunity for relationship building and audience engagement.

Chat messengers present themselves as representatives of socials on mobile and drive the size of traffic too big to ignore.

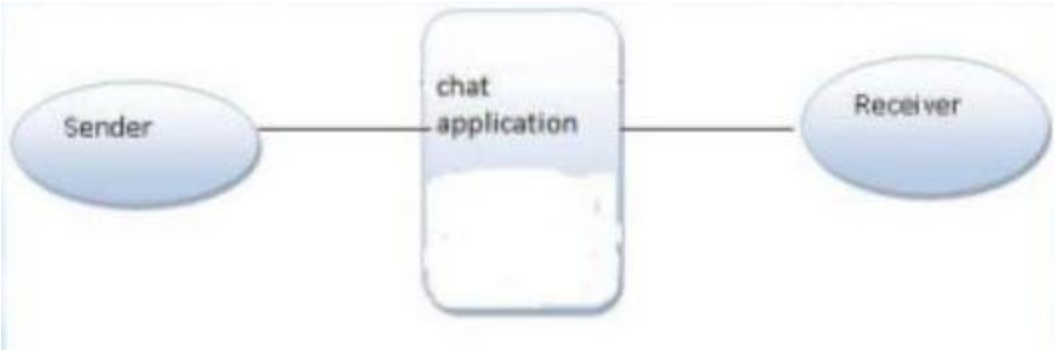
From above discussion, it is evident how chat applications are proposed to play various roles in addition to acting as a chat communication medium.

As days pass by, chat messengers are going to be even more advanced in its capability. While app publishers exhibit immense joy, optimism and excitement for their success so far, they all almost agree that the chat app industry is still its experimental and exploratory phase.

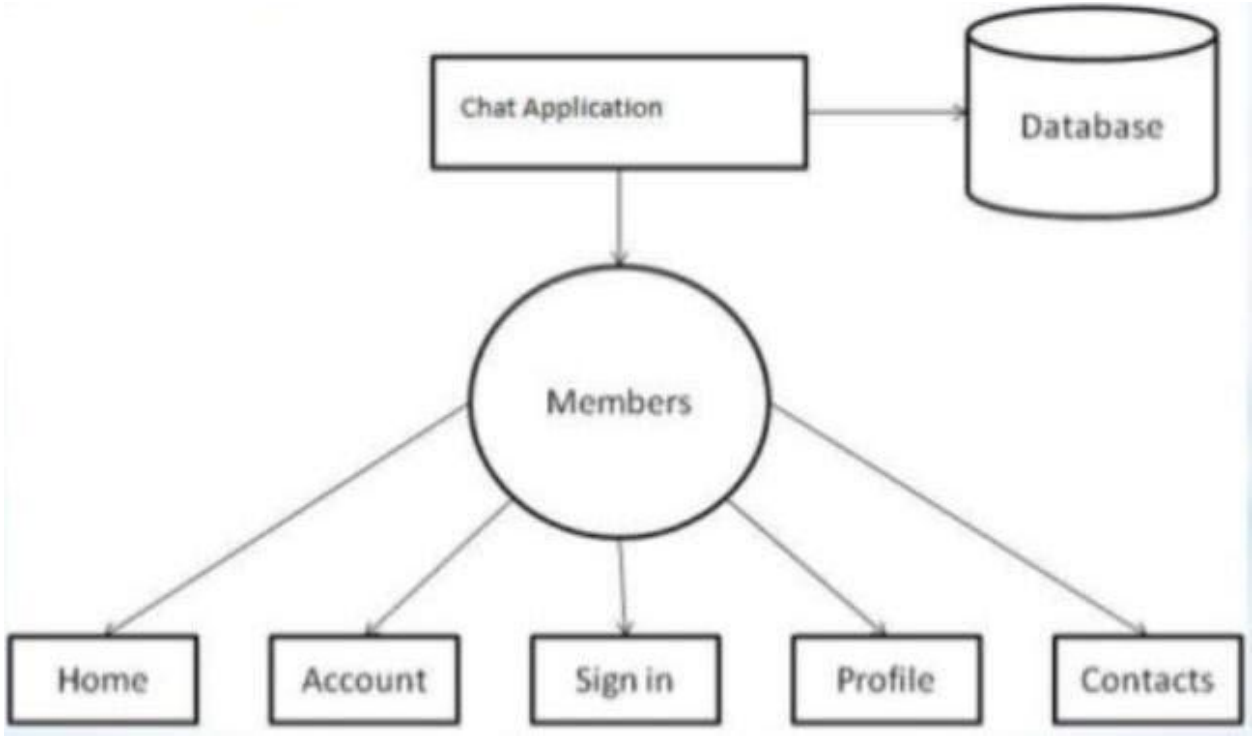
Implementation Details



Initializing and basic thought process of working on a chat application.



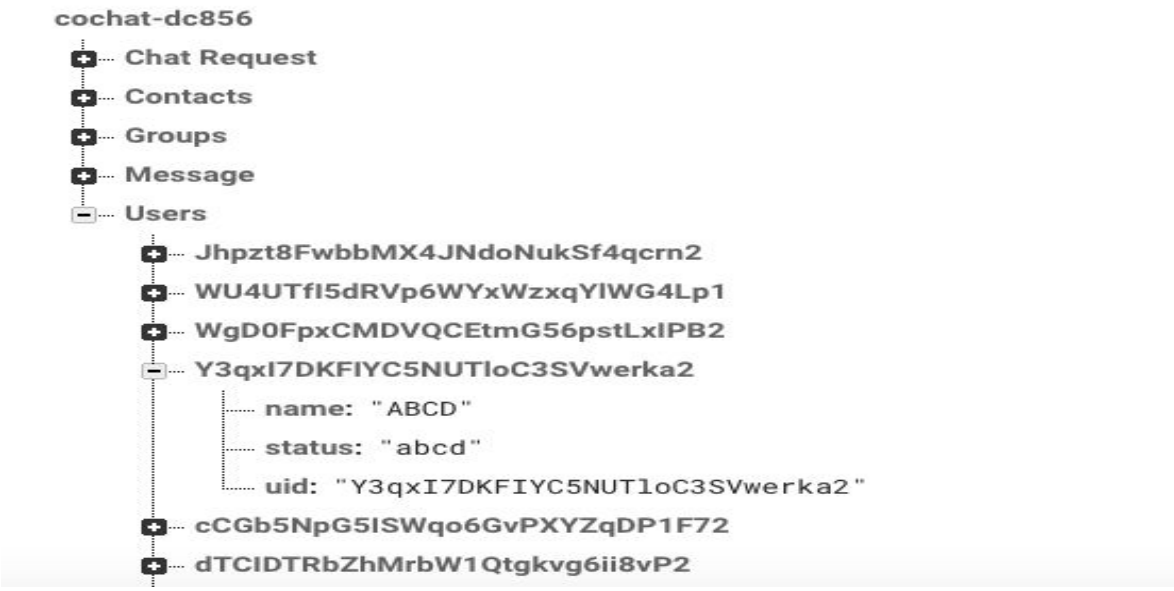
Through this DFD we are showing the basics of any contacting application. As we are seeing that a sender is sending a message using the application and the receiver is getting the message through that application.



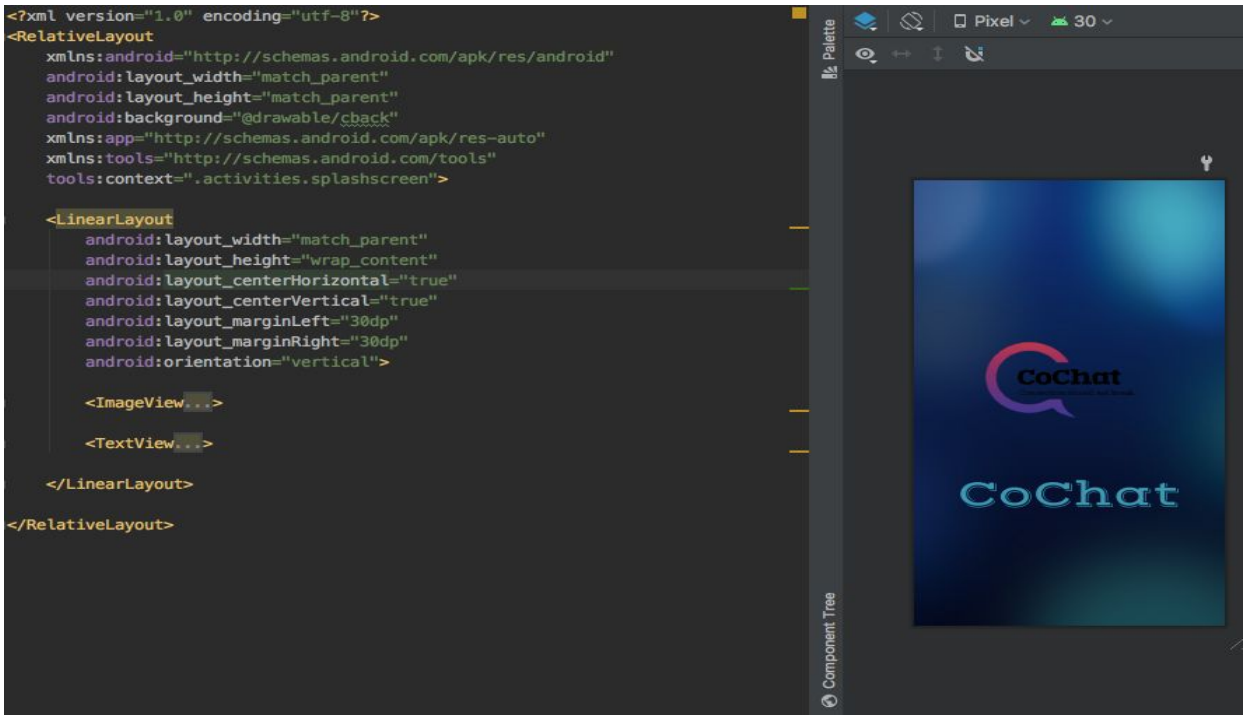
Through this DFD we can see following things-

- Users first need to login via their contact number or email id.
- Users have to upload their avatar or profile picture. (Not Mandatory)
- Users can chat with multiple people at a time but that contact must be saved.
- All the data is being stored in the database in the background.

Screenshots

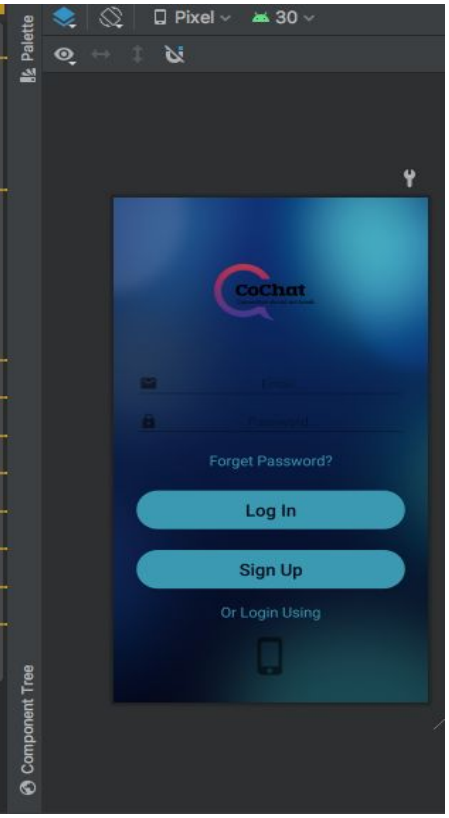


Firebase provides Realtime Database and backend as a service. Our database looks like this.



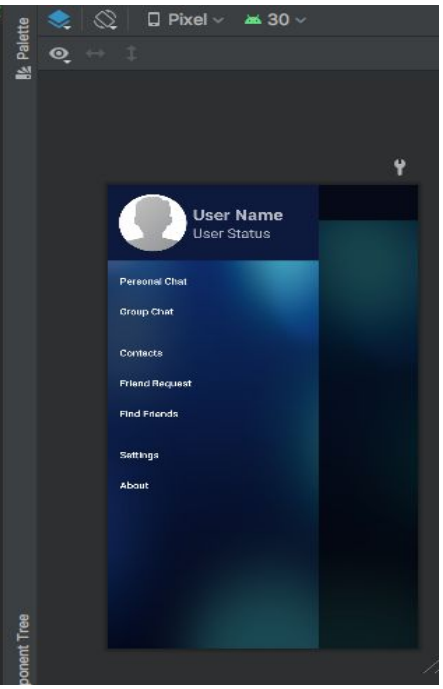
Welcome sliders provides you a brief introduction about chat application.

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/cback"
    tools:context=".activities.LoginActivity">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_marginRight="30dp"
            android:layout_marginLeft="30dp"
            android:layout_centerHorizontal="true"
            android:orientation="vertical">

            <ImageView...>

            <EditText...>

            <EditText...>

            <TextView...>

            <Button...>

            <Button...>

            <TextView...>

            <de.hdodenhof.circleimageview.CircleImageView...>

        </LinearLayout>
```

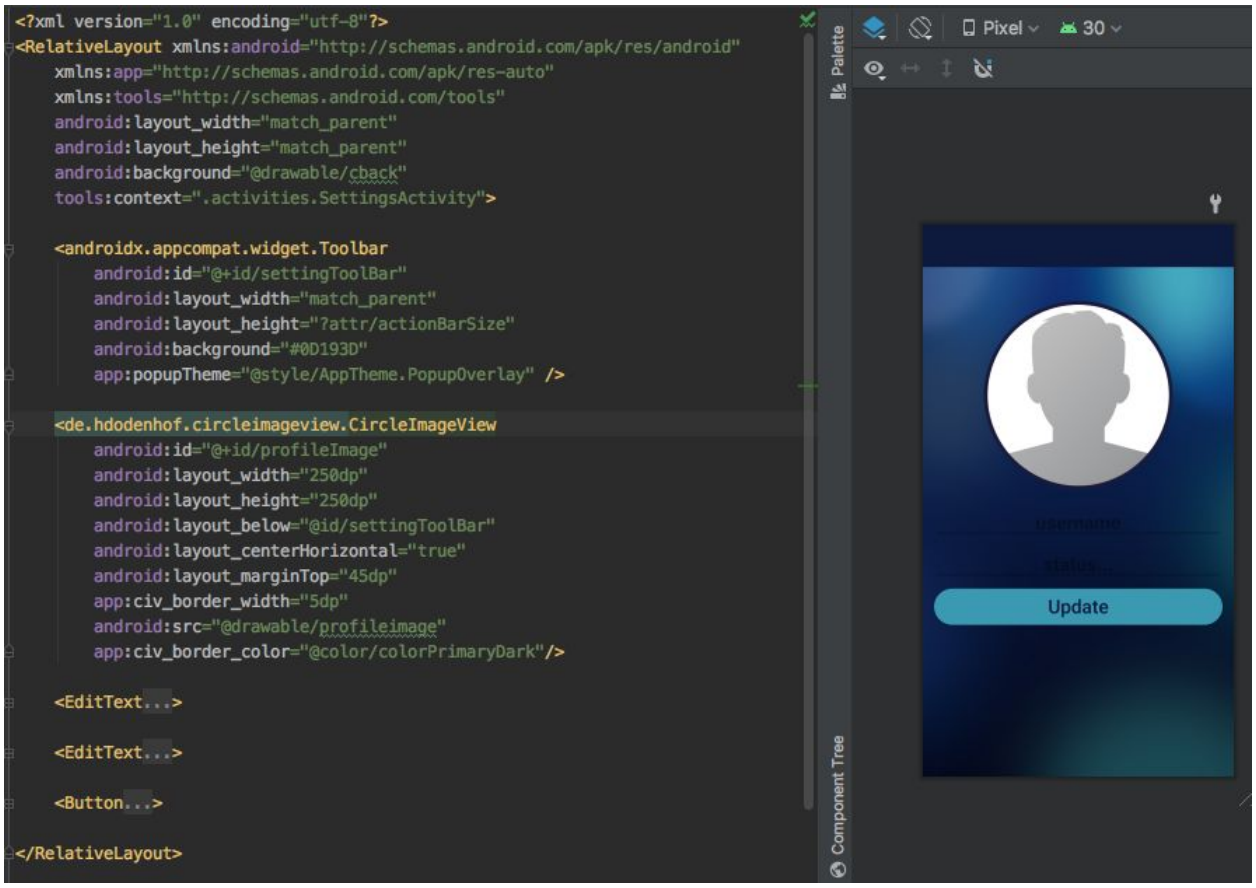



```java
public void allowUserToLogin(){
    String email = emailEditText.getText().toString();
    String password = passwordEditText.getText().toString();
    //  check whether users enters email and password or not
    if (email.equals(""))
        Toast.makeText( context: this,  text: "Please enter email...", Toast.LENGTH_SHORT).show();
    else if (password.equals(""))
        Toast.makeText( context: this,  text: "Please enter password...", Toast.LENGTH_SHORT).show();
    //  if both have entered
    else {
        //  add loading bar to indicate the user
        loadingBar.setTitle("SignIn");
        loadingBar.setMessage("Please wait...");
        loadingBar.setCanceledOnTouchOutside(true);
        loadingBar.show();
        //  send email and password to firebase for sign in
        firebaseAuth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener((task) → {
                        if (task.isSuccessful()) {
                            sendUserToMainActivity();
                            Toast.makeText( context: LoginActivity.this,  text: "Logged in successful", Toast.LENGTH_SHORT).show();
                        }
                        else {
                            new AlertDialog.Builder( context: LoginActivity.this)
                                    .setIcon(android.R.drawable.ic_dialog_alert)
                                    .setTitle("Error in login")
                                    .setMessage(task.getException().getMessage())
                                    .setPositiveButton( text: "Ok",  listener: null)
                                    .show();
                        }
                        loadingBar.dismiss();
                });
    }
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:background="@drawable/cback"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:itemTextColor="@android:color/background_light"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer" />
</androidx.drawerlayout.widget.DrawerLayout>
```




```java
@Override
public boolean onSupportNavigateUp() {
    NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);
    return NavigationUI.navigateUp(navController, mAppBarConfiguration)
            || super.onSupportNavigateUp();
}

@Override
protected void onStart() {...}

private void verifyUserExistence() {...}

private void updateHeader() {
    NavigationView navigationView = findViewById(R.id.nav_view);
    View headerView = navigationView.getHeaderView( index: 0);
    final TextView headerName = headerView.findViewById(R.id.headerProfileName);
    final TextView headerStatus = headerView.findViewById(R.id.headerProfileStatus);
    final CircleImageView headerImage = headerView.findViewById(R.id.headerProfileImage);

    final String currentUserId = firebaseAuth.getCurrentUser().getUid();
    dbReference.child("Users").child(currentUserId).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.hasChild( path: "image")){
                String name = snapshot.child("name").getValue().toString();
                String status = snapshot.child("status").getValue().toString();
                String image = snapshot.child("image").getValue().toString();
                headerName.setText(name);
                headerStatus.setText(status);
                Glide.with( activity: MainActivity.this).load(image).placeholder(R.drawable.profileimage).into(headerImage);
            }
            else {
                String name = snapshot.child("name").getValue().toString();
                String status = snapshot.child("status").getValue().toString();
                headerName.setText(name);
                headerStatus.setText(status);
            }
```




```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/cback"
    tools:context=".activities.SettingsActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/settingToolBar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="#0D193D"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/profileImage"
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:layout_below="@id/settingToolBar"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="45dp"
        app:civ_border_width="5dp"
        android:src="@drawable/profileimage"
        app:civ_border_color="@color/colorPrimaryDark"/>

    <EditText...>

    <EditText...>

    <Button...>

</RelativeLayout>
```

```
private void updateSettings(){

    if (userNameEditText.getText().toString().equals(""))
        Toast.makeText( context: this,  text: "Please write username!", Toast.LENGTH_SHORT).show();
    else if (statusEditText.getText().toString().equals(""))
        Toast.makeText( context: this,  text: "Please write status!", Toast.LENGTH_SHORT).show();
    else {
        HashMap<String, Object> profileMap = new HashMap<>();
        profileMap.put("uid", currentUserId);
        profileMap.put("name", userNameEditText.getText().toString());
        profileMap.put("status", statusEditText.getText().toString());

        dbReference.child("Users").child(currentUserId).updateChildren(profileMap)
                .addOnCompleteListener((task) → {
                        if (task.isSuccessful()) {
                            sendUserToMainActivity();
                            Toast.makeText( context: SettingsActivity.this,  text: "Profile updated.", Toast.LENGTH_SHORT).show();
                        }
                        else {
                            new AlertDialog.Builder( context: SettingsActivity.this)
                                    .setIcon(android.R.drawable.ic_dialog_alert)
                                    .setTitle("Failed to update settings")
                                    .setMessage(task.getException().getMessage())
                                    .setPositiveButton( text: "Ok",  listener: null)
                                    .show();
                        }
                });
    }
}
```

```
private void saveMessageInfoToDatabase() {
    String message = inputGroupMessage.getText().toString();
    String messageKey = groupNameReference.push().getKey();
    if (message.equals(""))
        Toast.makeText( context: this,  text: "Please input the message first..", Toast.LENGTH_SHORT).show();
    else {
        Calendar calForDate = Calendar.getInstance();
        SimpleDateFormat currentDateFormat = new SimpleDateFormat( pattern: "dd MM, yyyy");
        String currentDate = currentDateFormat.format(calForDate.getTime());

        Calendar calForTime = Calendar.getInstance();
        @SuppressLint("SimpleDateFormat") SimpleDateFormat currentTimeFormat = new SimpleDateFormat( pattern: "hh:mm a");
        String currentTime = currentTimeFormat.format(calForTime.getTime());

        HashMap<String, Object> groupMessageKey = new HashMap<>();
        groupNameReference.updateChildren(groupMessageKey);
        DatabaseReference groupMessageKeyRef = groupNameReference.child(messageKey);

        HashMap<String, Object> messageInfoMap = new HashMap<>();
        messageInfoMap.put("name", currentUserName);
        messageInfoMap.put("message", message);
        messageInfoMap.put("date", currentDate);
        messageInfoMap.put("time", currentTime);
        groupMessageKeyRef.updateChildren(messageInfoMap);
    }
}
```

```
FirebaseRecyclerOptions options = new FirebaseRecyclerOptions.Builder<Contact>()
        .setQuery(contactsRef, Contact.class)
        .build();

FirebaseRecyclerAdapter<Contact, ContactsViewHolder> adapter = new
        FirebaseRecyclerAdapter<>(options) {
    @Override
    protected void onBindViewHolder(@NonNull final ContactsViewHolder holder, final int position, @NonNull Contact model)
        String userId = getRef(position).getKey();
        userRef.child(userId).addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists() && snapshot.hasChild( path: "image")){
                    String userImage = snapshot.child("image").getValue().toString();
                    String profileName = snapshot.child("name").getValue().toString();
                    String profileStatus = snapshot.child("status").getValue().toString();
                    holder.userName.setText(profileName);
                    holder.userStatus.setText(profileStatus);
                    Glide.with( fragment: ContactsFragment.this).load(userImage)
                            .placeholder(R.drawable.profileimage).into(holder.profileImage);
                }
                else if (snapshot.exists()){
                    String profileName = snapshot.child("name").getValue().toString();
                    String profileStatus = snapshot.child("status").getValue().toString();
                    holder.userName.setText(profileName);
                    holder.userStatus.setText(profileStatus);
                }
                else {
                    Toast.makeText(getContext(),  text: "No contact is available.", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
```

References

- Wikipedia
- Google
- Youtube