Introduction:

Let's chat is an Android Chatting Application that supports logging in with phone number, sending of text messages and images, push notifications, and also supports group chat.

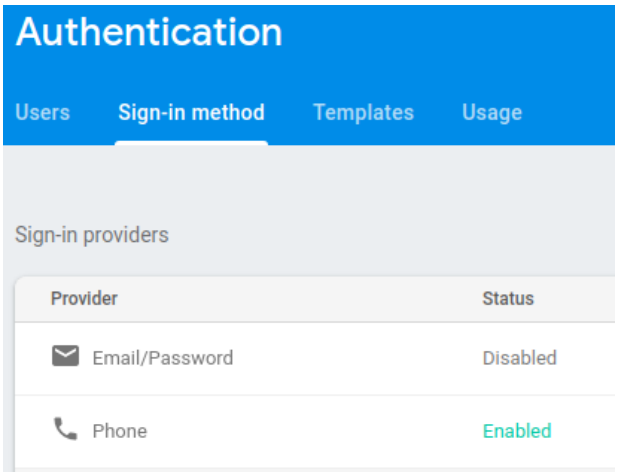
The back end of the project is done using Firebase, and the following features of firebase has been used:

1) Firebase Auth
2) Firebase Storage
3) Firebase Database

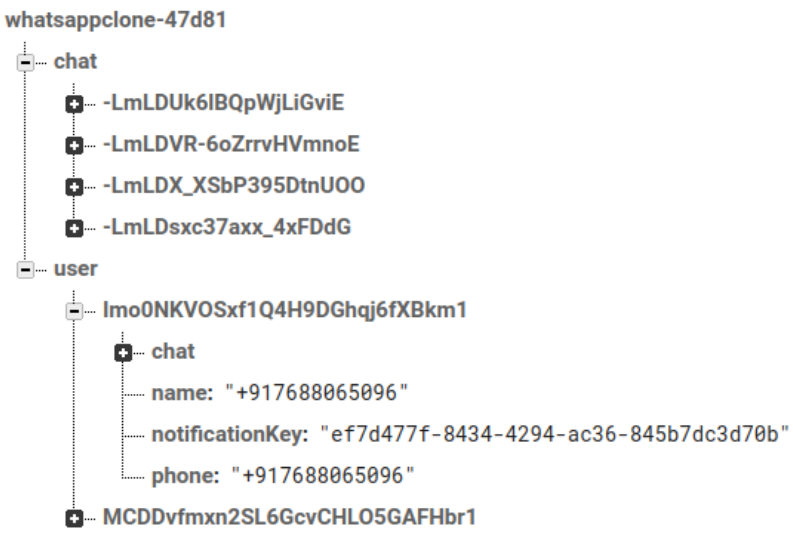
Also, for image sharing, Fresco, and Android library for managing images has been used, (https://github.com/facebook/fresco) and for push notifications, a service named OneSignal (https://onesignal.com/) is used.

Getting Phone Authentication & Displaying Contacts:

Initially, a new project was setup in Firebase Console, and the app was connected to it using Android Studio. Then, sign-in provider for phone was set to Enabled.



After authentication, the new user was pushed as an Entry to the database tree structure as follows:



After authentication, the contact list from the user device was obtained and the database was checked for similar entries. If an entry matches, it means that contact also has the app installed, and his name will be displayed in the create chat room page. Screenshots are attached at the end.

Creating Chat Room and Starting Chat Activity:

On clicking “find users” button in the home page, a chat room creation activity is started.
A ChatObject is created, which contains the chatId of the chat. It is updated to the “chat” child of the root of the database, and the id is inflated into the Chatroom list using the ChatList Adapter.
On clicking each entry in the chatroom list, we can enter into the corresponding chat room, which starts the chat activity.

Sending and receiving media and message:

The text message is put into an arraylist containing an user-defined MessageObject which contains messageId, senderID, message and mediaUriList (for images). The message is then stored into the database as following structure.

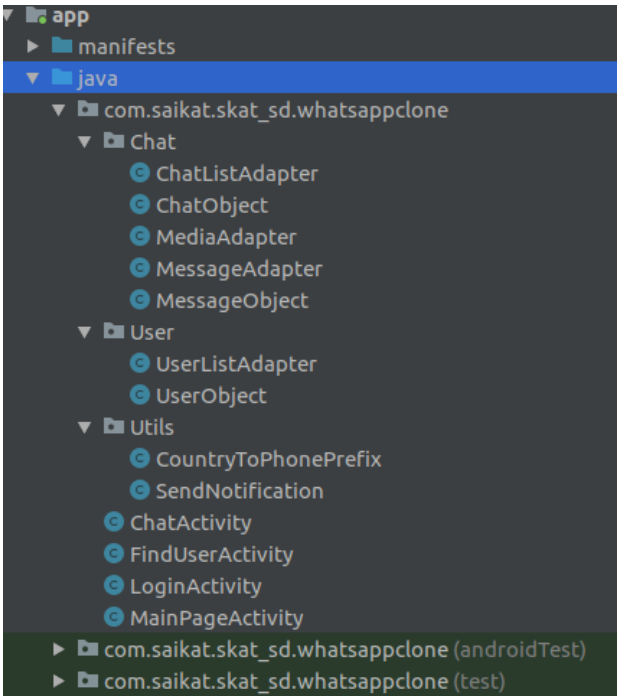
It is then inflated from the database into the Chat room in the recycler view using a Message Adapter. For media objects, a Media Adapter is used. The image is stored Online in Firebase Storage and the Uri of The image is stored in the Firebase DB.

The images are viewed using Fresco Image Viewer.

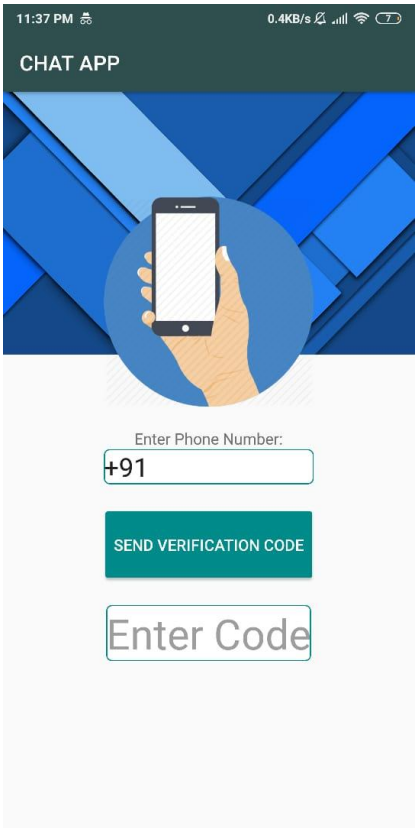
Sending Notifications:

For sending push notifications, an online service called OneSignal was used. A JSON object containing message, notificationKey and heading was created, and the postNotification() method of OneSignal was used to push notifications to the user as soon as a message is received.

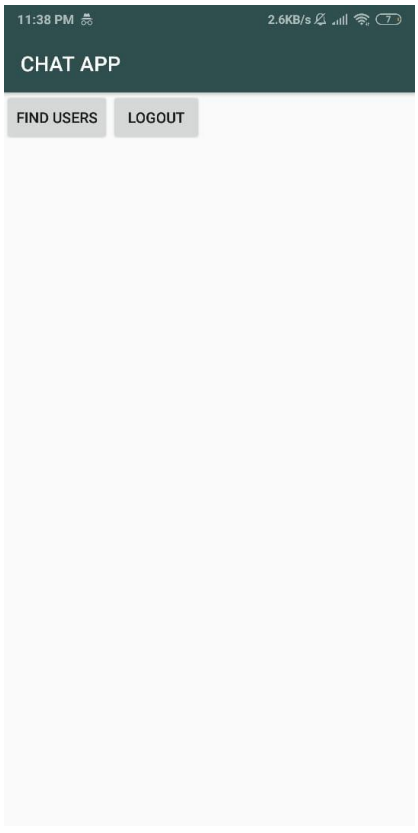
Project Structure:



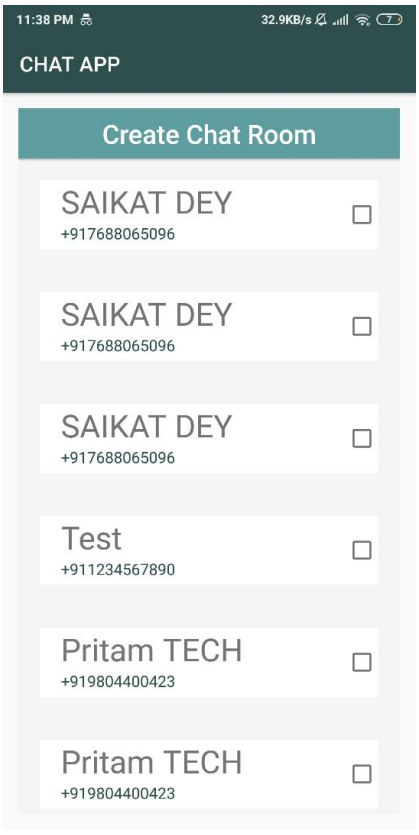
ScreenShots:



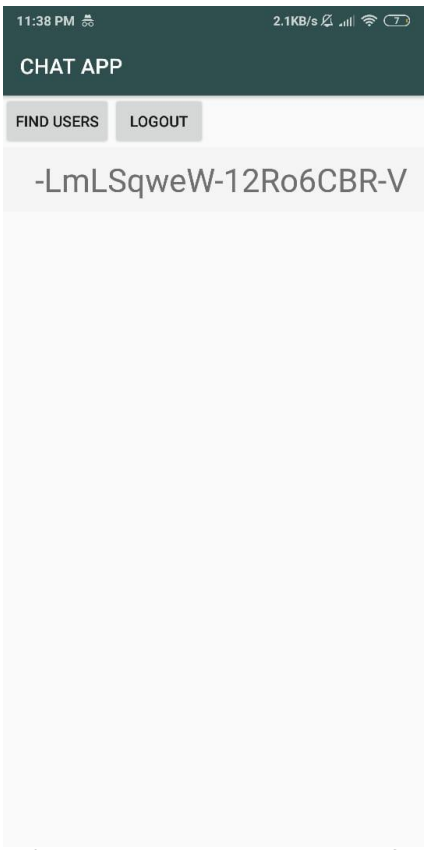
1) The login page



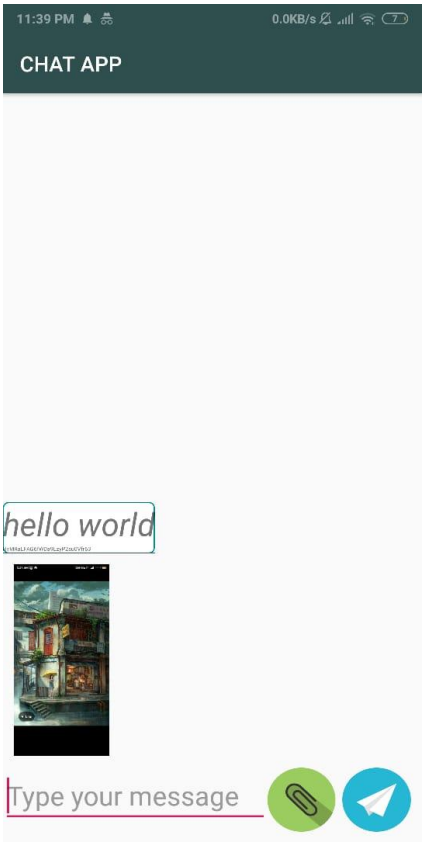
2)The homepage



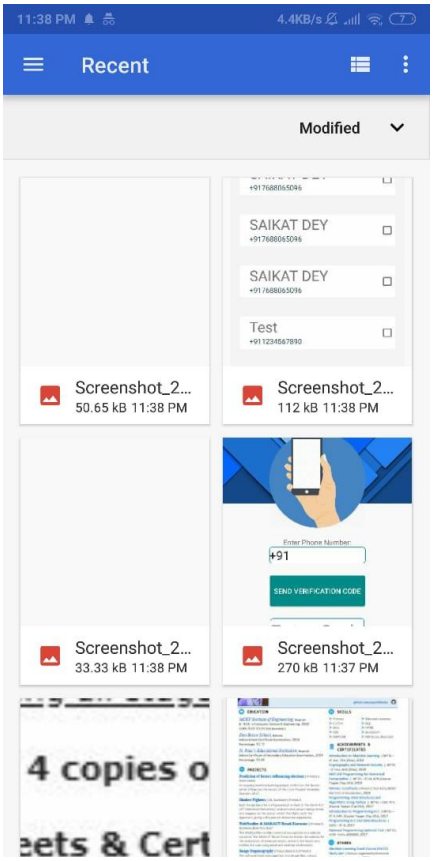
3) The create chat room



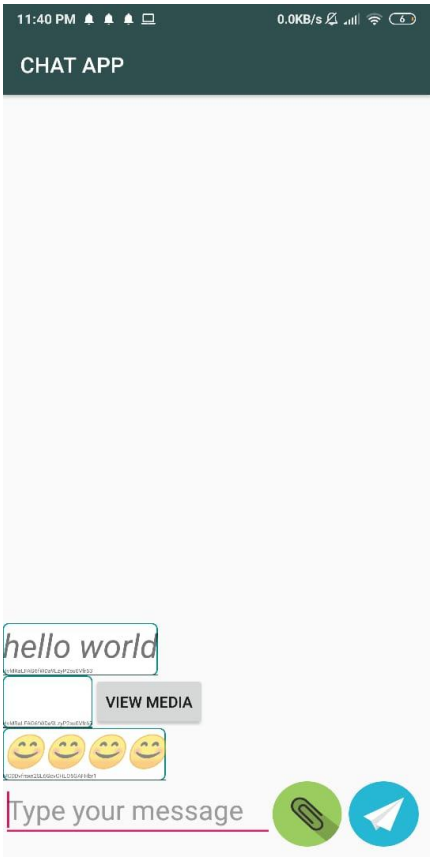
4) Home page updated after creating a chat room



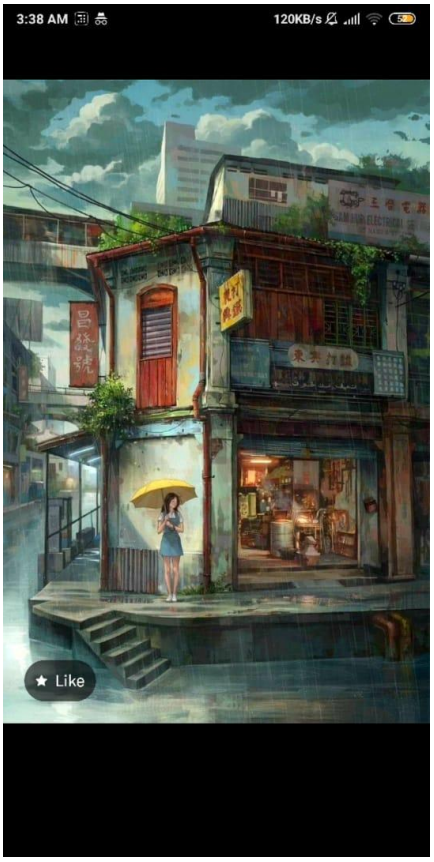
5) The main chat room



6) Selecting images to send



7) Image and text sent



8) Image viewer using Fresco

Improvement Areas:

1) The chat room is denoted by the unique chat ID generated, which is inflated in the Recycler View. It is not read-friendly as it is an unique ID. It can be replaced later by implementing chat room names.

2) Also, the sender of a message in a chat room is also denoted by their unique ID and not their name. A mapping from ID to name can be performed later to get the names of the users in the chat room.

3) UI improvements needs to be made.
