***Abstract* — This article demonstrates How to set up a simple Chat Room Android App and allow multiple clients to connect to it. The code along with the GUI created enables the user to login using credentials(email id, password) and chat with his/her acquaintances in a secure manner.**

***Keywords*** *—Android App Development, GUI Development, Database Management Systems(DBMS), App Architecture.*

INTRODUCTION

We have made a simple chat room Android App which has a feature of end to end encryption like Whatsapp. The safety and security of our users and their messages matters a lot to us . After using our chat room app for a while , the user will understand how trustworthy this app is .This app is also very easy to understand. Basically ,this chatroom project is a form of communication that utilizes apps that allow for two - way conversations between users in real time . Typically , the users will connect to a chat server using a chat client and meet in a chatroom . Conversations are then carried on by reading the messages entered by the other users in the chatroom and responding to them .

The main aim of this project is to attract the people towards a user - friendly interface . We have built a chatroom which is simple , secure and reliable to massage all the people . We want to give people a better experience of chatting through this application.

# Methodology

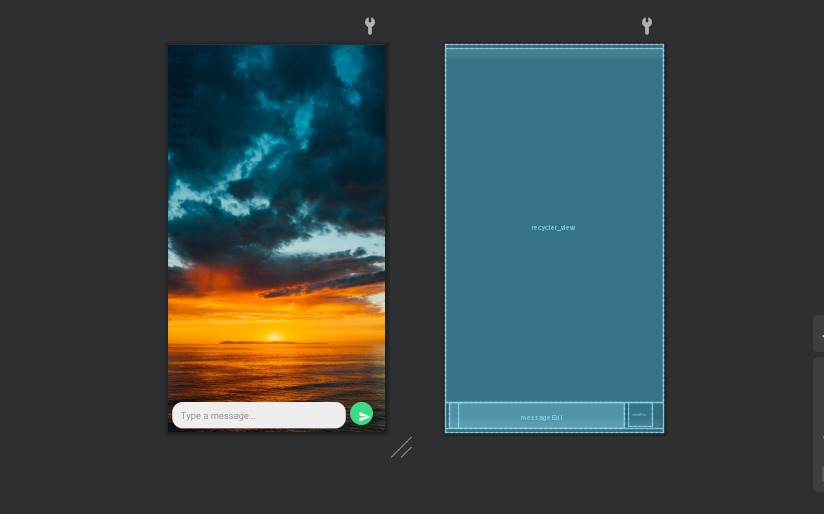
Basics of Communication

Communication can be defined as the process of sharing information in a secure and efficient way. It can be point-to-point(P2P) and point-to-multipoint(PMP). Also the communication channel through which two or more nodes communicate with each other must be secure and noise immune.

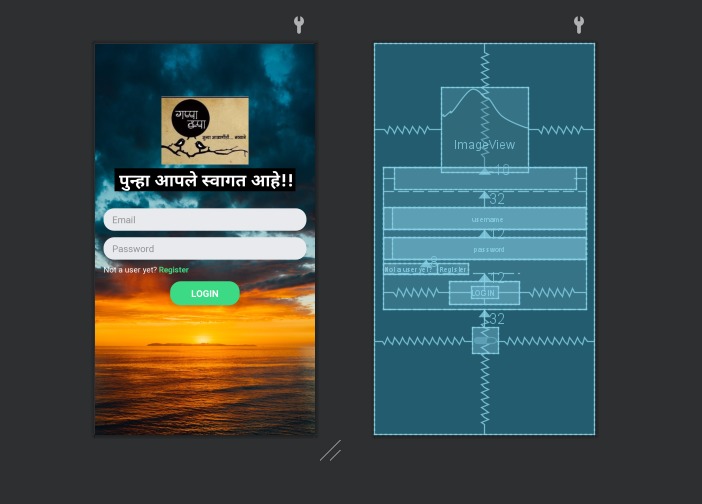
Usage

Here, we have developed a front-end XML page and back-end Java Script for each activity. The activities used in the app are as follows-

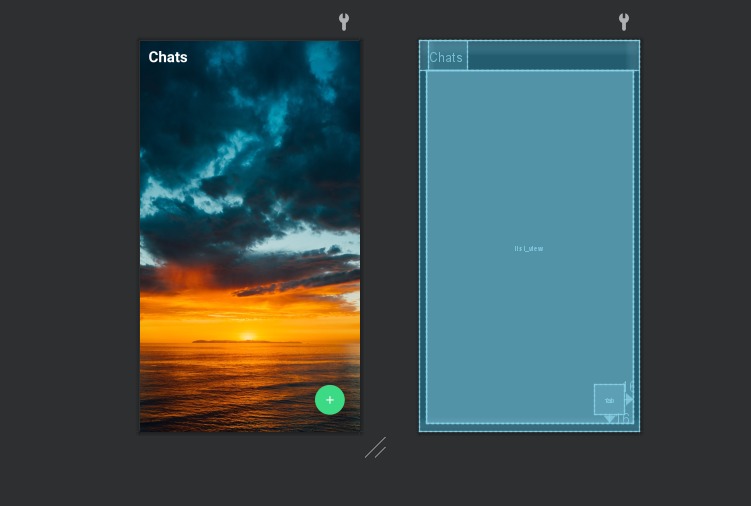
1. Chat



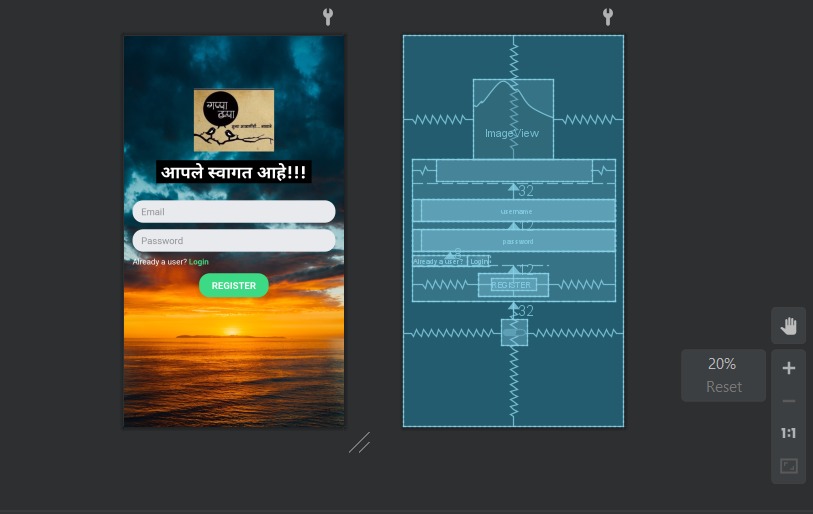
1. Login



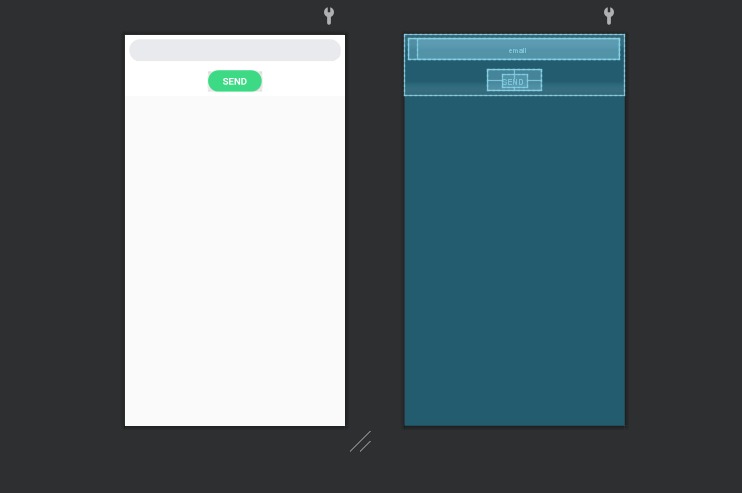
1. Main



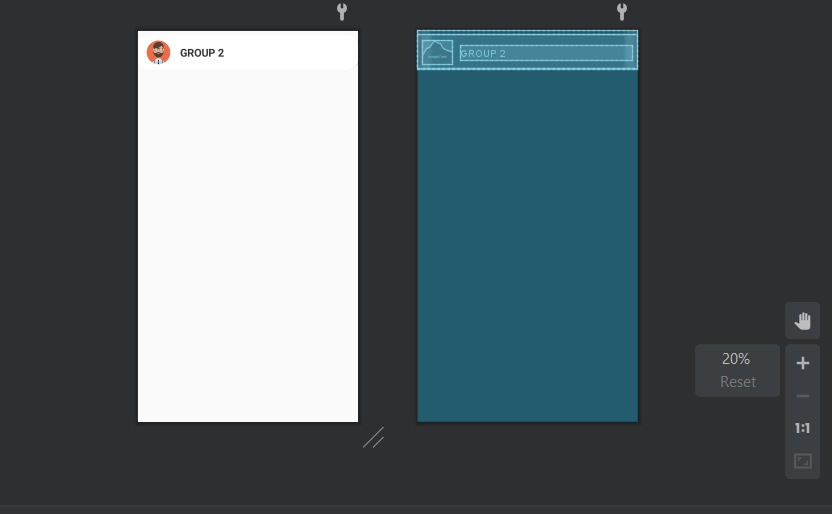
1. Register



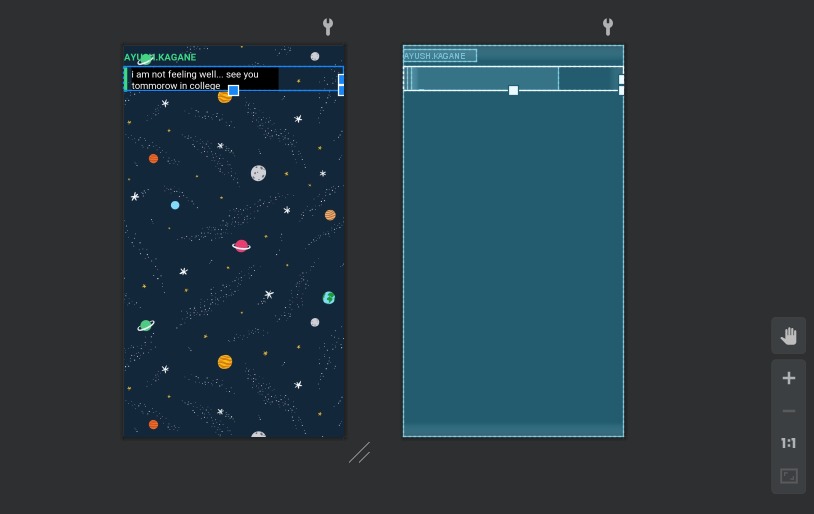
1. New Chat



1. Layout Chat



1. Layout Message



For better understanding, we have the JavaScript for Main Activity here below:-

package com.example.gappa\_tappa;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.material.floatingactionbutton.FloatingActionButton;

import com.google.firebase.FirebaseApp;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.example.gappa\_tappa.ChatRoom;

import com.example.gappa\_tappa.NewChatDialog;

import com.example.gappa\_tappa.R;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity implements View.OnClickListener,

SwipeRefreshLayout.OnRefreshListener {

//widgets

private ListView mListView;

private FloatingActionButton fab;

private SwipeRefreshLayout swipe;

//vars

private static final String TAG = "MainActivity";

private ArrayList<ChatRoom> mChatRoomList = new ArrayList<>();

private ArrayList<String> mSenderList = new ArrayList<>();

private ArrayAdapter<String> adapter;

private String currentUserEmail;

private final String chatrooms = "chatrooms";

private final String to = "to";

private final String from = "from";

//firebase authetication

private FirebaseAuth.AuthStateListener mAuthListener;

//Firebase Database

private DatabaseReference mRoot =FirebaseDatabase.getInstance("https://gappa-tappa-fe82e-default-rtdb.firebaseio.com/").getReference();

private DatabaseReference mChatData = mRoot.child(chatrooms);

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Log.d(TAG, "onCreate: started");

setupFirebaseAuth();

//getting the intentExtra passed from login and register activity

currentUserEmail = getIntent().getStringExtra("email");

Log.d(TAG, "onCreate: Current Email" + currentUserEmail);

//set title to the action bar

getSupportActionBar().setTitle("Gappa-Tappa");

//find the widgets

mListView = findViewById(R.id.list\_view);

fab = findViewById(R.id.fab);

swipe = findViewById(R.id.swipe);

//set onRefresh Listener to the swipe refresh layout

swipe.setOnRefreshListener(this);

//set onClick listener to floating action button

fab.setOnClickListener(this);

//set up list view

adapter = new ArrayAdapter<String>(this,

R.layout.layout\_chat\_room, R.id.chat\_room\_email, mSenderList);

mListView.setAdapter(adapter);

//get the chatrooms from database

getChatRooms();

//add onItem click listener

mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

onChatClick(i);

}

});

}

//inflate the menu in options

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.main\_menu, menu);

return true;

}

//get the clicks in options menu

@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

if (item.getItemId() == R.id.sign\_out) {

//signOut button is pressed

signOut();

}

return true;

}

//log out the current user

private void signOut() {

Log.d(TAG, "signOut: sigining out.");

FirebaseAuth.getInstance().signOut();

Toast.makeText(this, "Bye! Bye! ", Toast.LENGTH\_SHORT).show();

Log.d(TAG, "signOut: User signed out");

startActivity(new Intent(this, LoginActivity.class));

}

private void setupFirebaseAuth() {

Log.d(TAG, "setupFirebaseAuth: started.");

FirebaseApp.initializeApp(this);

mAuthListener = new FirebaseAuth.AuthStateListener() {

@Override

public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user != null) {

//user is signed in

Log.d(TAG, "onAuthStateChanged:signed\_in:" + user.getUid());

currentUserEmail = user.getEmail();

} else {

//user is signed out... revert to login page

Log.d(TAG, "onAuthStateChanged:signed\_out");

//Toast.makeText(MainActivity.this, "Please Login", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(MainActivity.this, LoginActivity.class);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);

startActivity(intent);

finish();

}

}

};

}

@Override

public void onStart() {

super.onStart();

FirebaseAuth.getInstance().addAuthStateListener(mAuthListener);

}

@Override

public void onStop() {

super.onStop();

if (mAuthListener != null) {

FirebaseAuth.getInstance().removeAuthStateListener(mAuthListener);

}

}

//executed when fab is clicked

@Override

public void onClick(View view) {

//TODO: code for sending the request to user using dialog box

Toast.makeText(this, "Add Chat", Toast.LENGTH\_SHORT).show();

NewChatDialog dialog = new NewChatDialog();

dialog.show(getSupportFragmentManager(), "New Chat");

}

private void getChatRooms() {

Log.d(TAG, "getChatRooms: started");

//get chat room in which current user is added

mChatData.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

//clear the chatroom list or else the chatrooms will repeat

mChatRoomList.clear();

mSenderList.clear();

for (DataSnapshot room : dataSnapshot.getChildren()) {

ChatRoom chatRoom = room.getValue(ChatRoom.class);

if (currentUserEmail.equals(chatRoom.getTo()) ||

currentUserEmail.equals(chatRoom.getFrom())) {

//user is present in this chatroom

mChatRoomList.add(chatRoom);

if (currentUserEmail.equals(chatRoom.getFrom())) {

String [] names = chatRoom.getTo().split("@");

mSenderList.add(names[0]);

} else {

String [] names = chatRoom.getFrom().split("@");

mSenderList.add(names[0]);

}

}

}

//notify the adapter about the change

adapter.notifyDataSetChanged();

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

Log.e(TAG, "onCancelled: ", databaseError.toException());

}

});

Log.d(TAG, "getChatRooms: current chat rooms" + mChatRoomList.toString());

}

//called from inside onClick

public void createNewChatRoom(final String email) {

String key = mChatData.push().getKey();

ChatRoom mChatRoom = new ChatRoom(email, currentUserEmail, key);

mChatData.child(key).setValue(mChatRoom).addOnSuccessListener(

new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Log.d(TAG, "onSuccess: chatroom added successfully");

}

}

).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.e(TAG, "onFailure: chat room not added", e);

}

});

}

//called when any of the list view item is clicked

public void onChatClick(int position) {

Intent intent = new Intent(MainActivity.this, ChatActivity.class);

ChatRoom mchatRoom = mChatRoomList.get(position);

intent.putExtra("key", mchatRoom.getId());

Log.d(TAG, "onChatClick: passing key " + mchatRoom.getId());

startActivity(intent);

}

//called every time the window is refreshed

@Override

public void onRefresh() {

getChatRooms();

swipe.setRefreshing(false);

Toast.makeText(this, "Refreshed", Toast.LENGTH\_SHORT).show();

}

Output: Initially, the user will login to the Chatroom App. After that, he/she can communicate with any of his friends or relatives(only point-to-point communication allowed). For Authentication purposes, Google Firebase keeps track of all the users logged in to the Chatroom App.

## Testing

The Android App was tested and was found to be in working condition.

## Software used

The whole program is compiled on Android Studio and collaborated with Google Firebase.

# Results and Discussions

1.The App was tested successfully and the Chatroom was running Successfully.

2.The Login, Chat, Main, Register, New Chat, Layout Chat, Layout Message Activities are working successfully in the ChatRoom.

3.Anyone can Sign-in to the ChatRoom,as it is User friendly.

4.Only One-to-one Conversation and sharing of data is available.

# Limitations

1. The App does not have support for point-to-multipoint(PMP) communications.
2. The users can chat using text messages. JPEG images, GIFs and videos cannot be shared.
3. It is mandatory to link the Android App to Google Firebase otherwise the App doesn’t work.

# Future Scope

1. We are going to Develop an actual Android App using Android Studio.
2. More Graphical interfaces can be added in it.
3. We can use The App to make it more User Friendly.

# Conclusion

1. The main purpose of this project is to develop a secure chat application .

2. You can use this application very comfortably as well as smoothly .

3 . We have used the java programming language in the coding of this chat room project .

4. Basically , This chatroom project presents a software application for the launching of a real time communication between two users .

5 . This system is developed on android which will enable the users to communicate with other users through text messages with the help of the internet .

6 . So , we have tried to make this chat room application very unique as compared to some existing applications.

References

1. <https://youtu.be/Hn_wfTqFbdg>
2. <https://youtu.be/9cnKfO_Ywmc>
3. <https://www.scaledrone.com/blog/android-chat-tutorial/>
4. <http://www.devexchanges.info/2016/12/simple-chat-application-using-firebase.html?m=1>
5. <http://www.devexchanges.info/2016/12/simple-chat-application-using-firebase.html?m=1>