Title: Android Chat application (Heychat)

**Motivation:**

Online chat may refer to any kind of communication over the Internet that offers a real-time transmission of text messages from sender to receiver. Chat messages are generally short in order to enable other participants to respond quickly. Thereby, a feeling similar to a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as Internet forums and email. Online chat may address point-to-point communications as well as multicast communications from one sender to many receivers. So, our goal is to make an Android chat application called Heychat. In which user can chats with his or her friends which he or she can add him through their friend’s email id. This app will provide verification of the person by login with valid email id. And provides various features of online chatting. Simple and easy to understand UI makes the app more beautiful and easier to use.

How it works:

* In chat application We have created 5 Module:

1. Login/Register:

* The New user can register with email id and after register they can login with id.

1. Add Friend:

* The User can add friend with friend e-mail id.

1. Friend Chat:

* After Adding friend, the user can chat with friend.

1. Group chat:

* The User Can create the group with their friends.
* And can chat in group.

1. User Profile:

* The user Can change Her/his Profile picture.
* User can change the username.
* User can also reset password.

1. Forgot password

* If user forgot the password a reset password link will send to him/her.
* From the link he/she can change his/her password.

Program Code:

Java Files:

**1.RegisterActivity.java**

package com.mcwc.heychat.ui;

import android.animation.Animator;

import android.animation.AnimatorListenerAdapter;

import android.content.Intent;

import android.os.Build;

import android.os.Bundle;

import android.transition.Transition;

import android.transition.TransitionInflater;

import android.view.View;

import android.view.ViewAnimationUtils;

import android.view.animation.AccelerateInterpolator;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

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

import com.mcwc.heychat.R;

import com.mcwc.heychat.data.StaticConfig;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class RegisterActivity extends AppCompatActivity {

    FloatingActionButton fab;

    CardView cvAdd;

    private final Pattern VALID\_EMAIL\_ADDRESS\_REGEX =

            Pattern.compile("^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$", Pattern.CASE\_INSENSITIVE);

    private EditText editTextUsername, editTextPassword, editTextRepeatPassword;

    public static String STR\_EXTRA\_ACTION\_REGISTER = "register";

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_register);

        fab = (FloatingActionButton) findViewById(R.id.fab);

        cvAdd = (CardView) findViewById(R.id.cv\_add);

        editTextUsername = (EditText) findViewById(R.id.et\_username);

        editTextPassword = (EditText) findViewById(R.id.et\_password);

        editTextRepeatPassword = (EditText) findViewById(R.id.et\_repeatpassword);

        if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.LOLLIPOP) {

            ShowEnterAnimation();

        }

        fab.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                animateRevealClose();

            }

        });

    }

    private void ShowEnterAnimation() {

        Transition transition = TransitionInflater.from(this).inflateTransition(R.transition.fabtransition);

        getWindow().setSharedElementEnterTransition(transition);

        transition.addListener(new Transition.TransitionListener() {

            @Override

            public void onTransitionStart(Transition transition) {

                cvAdd.setVisibility(View.GONE);

            }

            @Override

            public void onTransitionEnd(Transition transition) {

                transition.removeListener(this);

                animateRevealShow();

            }

            @Override

            public void onTransitionCancel(Transition transition) {

            }

            @Override

            public void onTransitionPause(Transition transition) {

            }

            @Override

            public void onTransitionResume(Transition transition) {

            }

        });

    }

    public void animateRevealShow() {

        Animator mAnimator = ViewAnimationUtils.createCircularReveal(cvAdd, cvAdd.getWidth()/2,0, fab.getWidth() / 2, cvAdd.getHeight());

        mAnimator.setDuration(500);

        mAnimator.setInterpolator(new AccelerateInterpolator());

        mAnimator.addListener(new AnimatorListenerAdapter() {

            @Override

            public void onAnimationEnd(Animator animation) {

                super.onAnimationEnd(animation);

            }

            @Override

            public void onAnimationStart(Animator animation) {

                cvAdd.setVisibility(View.VISIBLE);

                super.onAnimationStart(animation);

            }

        });

        mAnimator.start();

    }

    public void animateRevealClose() {

        Animator mAnimator = ViewAnimationUtils.createCircularReveal(cvAdd,cvAdd.getWidth()/2,0, cvAdd.getHeight(), fab.getWidth() / 2);

        mAnimator.setDuration(500);

        mAnimator.setInterpolator(new AccelerateInterpolator());

        mAnimator.addListener(new AnimatorListenerAdapter() {

            @Override

            public void onAnimationEnd(Animator animation) {

                cvAdd.setVisibility(View.INVISIBLE);

                super.onAnimationEnd(animation);

                fab.setImageResource(R.drawable.ic\_signup);

                RegisterActivity.super.onBackPressed();

            }

            @Override

            public void onAnimationStart(Animator animation) {

                super.onAnimationStart(animation);

            }

        });

        mAnimator.start();

    }

    @Override

    public void onBackPressed() {

        animateRevealClose();

    }

    public void clickRegister(View view) {

        String username = editTextUsername.getText().toString();

        String password = editTextPassword.getText().toString();

        String repeatPassword = editTextRepeatPassword.getText().toString();

        if(validate(username, password, repeatPassword)){

            Intent data = new Intent();

            data.putExtra(StaticConfig.STR\_EXTRA\_USERNAME, username);

            data.putExtra(StaticConfig.STR\_EXTRA\_PASSWORD, password);

            data.putExtra(StaticConfig.STR\_EXTRA\_ACTION, STR\_EXTRA\_ACTION\_REGISTER);

            setResult(RESULT\_OK, data);

            finish();

        }else {

            Toast.makeText(this, "Invalid email or not match password", Toast.LENGTH\_SHORT).show();

        }

    }

    private boolean validate(String emailStr, String password, String repeatPassword) {

        Matcher matcher = VALID\_EMAIL\_ADDRESS\_REGEX .matcher(emailStr);

        return password.length() > 0 && repeatPassword.equals(password) && matcher.find();

    }

}

**2. LoginActivity.java**

package com.mcwc.heychat.ui;

import android.app.ActivityOptions;

import android.content.Intent;

import android.os.Build;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.EditText;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

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

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

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

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

import com.google.firebase.auth.AuthResult;

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.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.mcwc.heychat.MainActivity;

import com.mcwc.heychat.R;

import com.mcwc.heychat.data.SharedPreferenceHelper;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.User;

import com.yarolegovich.lovelydialog.LovelyInfoDialog;

import com.yarolegovich.lovelydialog.LovelyProgressDialog;

import java.util.HashMap;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class LoginActivity extends AppCompatActivity {

    private static String TAG = "LoginActivity";

    FloatingActionButton fab;

    private final Pattern VALID\_EMAIL\_ADDRESS\_REGEX =

            Pattern.compile("^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$", Pattern.CASE\_INSENSITIVE);

    private EditText editTextUsername, editTextPassword;

    private LovelyProgressDialog waitingDialog;

    private AuthUtils authUtils;

    private FirebaseAuth mAuth;

    private FirebaseAuth.AuthStateListener mAuthListener;

    private FirebaseUser user;

    private boolean firstTimeAccess;

    @Override

    protected void onStart() {

        super.onStart();

        mAuth.addAuthStateListener(mAuthListener);

    }

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_login);

        fab = (FloatingActionButton) findViewById(R.id.fab);

        editTextUsername = (EditText) findViewById(R.id.et\_username);

        editTextPassword = (EditText) findViewById(R.id.et\_password);

        firstTimeAccess = true;

        initFirebase();

    }

    private void initFirebase() {

        mAuth = FirebaseAuth.getInstance();

        authUtils = new AuthUtils();

        mAuthListener = new FirebaseAuth.AuthStateListener() {

            @Override

            public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {

                user = firebaseAuth.getCurrentUser();

                if (user != null) {

                    StaticConfig.UID = user.getUid();

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

                    if (firstTimeAccess) {

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

                        LoginActivity.this.finish();

                    }

                } else {

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

                }

                firstTimeAccess = false;

            }

        };

        waitingDialog = new LovelyProgressDialog(this).setCancelable(false);

    }

    @Override

    protected void onStop() {

        super.onStop();

        if (mAuthListener != null) {

            mAuth.removeAuthStateListener(mAuthListener);

        }

    }

    public void clickRegisterLayout(View view) {

        getWindow().setExitTransition(null);

        getWindow().setEnterTransition(null);

        if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.LOLLIPOP) {

            ActivityOptions options =

                    ActivityOptions.makeSceneTransitionAnimation(this, fab, fab.getTransitionName());

            startActivityForResult(new Intent(this, RegisterActivity.class), StaticConfig.REQUEST\_CODE\_REGISTER, options.toBundle());

        } else {

            startActivityForResult(new Intent(this, RegisterActivity.class), StaticConfig.REQUEST\_CODE\_REGISTER);

        }

    }

    @Override

    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == StaticConfig.REQUEST\_CODE\_REGISTER && resultCode == RESULT\_OK) {

            authUtils.createUser(data.getStringExtra(StaticConfig.STR\_EXTRA\_USERNAME), data.getStringExtra(StaticConfig.STR\_EXTRA\_PASSWORD));

        }

    }

    public void clickLogin(View view) {

        String username = editTextUsername.getText().toString();

        String password = editTextPassword.getText().toString();

        if (validate(username, password)) {

            authUtils.signIn(username, password);

        } else {

            Toast.makeText(this, "Invalid email or empty password", Toast.LENGTH\_SHORT).show();

        }

    }

    @Override

    public void onBackPressed() {

        super.onBackPressed();

        setResult(RESULT\_CANCELED, null);

        finish();

    }

    private boolean validate(String emailStr, String password) {

        Matcher matcher = VALID\_EMAIL\_ADDRESS\_REGEX.matcher(emailStr);

        return (password.length() > 0 || password.equals(";")) && matcher.find();

    }

    public void clickResetPassword(View view) {

        String username = editTextUsername.getText().toString();

        if (validate(username, ";")) {

            authUtils.resetPassword(username);

        } else {

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

        }

    }

    class AuthUtils {

        void createUser(String email, String password) {

            waitingDialog.setIcon(R.drawable.ic\_add\_friend)

                    .setTitle("Registering....")

                    .setTopColorRes(R.color.colorPrimary)

                    .show();

            mAuth.createUserWithEmailAndPassword(email, password)

                    .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {

                        @Override

                        public void onComplete(@NonNull Task<AuthResult> task) {

                            Log.d(TAG, "createUserWithEmail:onComplete:" + task.isSuccessful());

                            waitingDialog.dismiss();

                            if (!task.isSuccessful()) {

                                new LovelyInfoDialog(LoginActivity.this) {

                                    @Override

                                    public LovelyInfoDialog setConfirmButtonText(String text) {

                                        findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                            @Override

                                            public void onClick(View view) {

                                                dismiss();

                                            }

                                        });

                                        return super.setConfirmButtonText(text);

                                    }

                                }

                                        .setTopColorRes(R.color.colorAccent)

                                        .setIcon(R.drawable.ic\_add\_friend)

                                        .setTitle("Registration Failed")

                                        .setMessage("Email already exist or weak password!")

                                        .setConfirmButtonText("ok")

                                        .setCancelable(false)

                                        .show();

                            } else {

                                initNewUserInfo();

                                Toast.makeText(LoginActivity.this, "Register and Login success", Toast.LENGTH\_SHORT).show();

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

                                LoginActivity.this.finish();

                            }

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            waitingDialog.dismiss();

                        }

                    })

            ;

        }

        void signIn(String email, String password) {

            waitingDialog.setIcon(R.drawable.ic\_person\_low)

                    .setTitle("Login....")

                    .setTopColorRes(R.color.colorPrimary)

                    .show();

            mAuth.signInWithEmailAndPassword(email, password)

                    .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {

                        @Override

                        public void onComplete(@NonNull Task<AuthResult> task) {

                            Log.d(TAG, "signInWithEmail:onComplete:" + task.isSuccessful());

                            waitingDialog.dismiss();

                            if (!task.isSuccessful()) {

                                Log.w(TAG, "signInWithEmail:failed", task.getException());

                                new LovelyInfoDialog(LoginActivity.this) {

                                    @Override

                                    public LovelyInfoDialog setConfirmButtonText(String text) {

                                        findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                            @Override

                                            public void onClick(View view) {

                                                dismiss();

                                            }

                                        });

                                        return super.setConfirmButtonText(text);

                                    }

                                }

                                        .setTopColorRes(R.color.colorAccent)

                                        .setIcon(R.drawable.ic\_person\_low)

                                        .setTitle("Login Failed")

                                        .setMessage("Email not exist or wrong password!")

                                        .setCancelable(false)

                                        .setConfirmButtonText("Ok")

                                        .show();

                            } else {

                                saveUserInfo();

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

                                LoginActivity.this.finish();

                            }

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            waitingDialog.dismiss();

                        }

                    });

        }

        void resetPassword(final String email) {

            mAuth.sendPasswordResetEmail(email)

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            new LovelyInfoDialog(LoginActivity.this) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorPrimary)

                                    .setIcon(R.drawable.ic\_pass\_reset)

                                    .setTitle("Password Recovery")

                                    .setMessage("Sent email to " + email)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    })

            .addOnFailureListener(new OnFailureListener() {

                @Override

                public void onFailure(@NonNull Exception e) {

                    new LovelyInfoDialog(LoginActivity.this) {

                        @Override

                        public LovelyInfoDialog setConfirmButtonText(String text) {

                            findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                @Override

                                public void onClick(View view) {

                                    dismiss();

                                }

                            });

                            return super.setConfirmButtonText(text);

                        }

                    }

                            .setTopColorRes(R.color.colorAccent)

                            .setIcon(R.drawable.ic\_pass\_reset)

                            .setTitle("Failed")

                            .setMessage("Failed to sent email to " + email)

                            .setConfirmButtonText("Ok")

                            .show();

                }

            });

        }

                       void saveUserInfo() {

                    FirebaseDatabase.getInstance().getReference().child("user/" + StaticConfig.UID).addListenerForSingleValueEvent(new ValueEventListener() {

                        @Override

                        public void onDataChange(DataSnapshot dataSnapshot) {

                            waitingDialog.dismiss();

                            HashMap hashUser = (HashMap) dataSnapshot.getValue();

                            User userInfo = new User();

                            userInfo.name = (String) hashUser.get("name");

                            userInfo.email = (String) hashUser.get("email");

                            userInfo.avata = (String) hashUser.get("avata");

                            SharedPreferenceHelper.getInstance(LoginActivity.this).saveUserInfo(userInfo);

                        }

                @Override

                public void onCancelled(DatabaseError databaseError) {

                }

            });

        }

        void initNewUserInfo() {

            User newUser = new User();

            newUser.email = user.getEmail();

            newUser.name = user.getEmail().substring(0, user.getEmail().indexOf("@"));

            newUser.avata = StaticConfig.STR\_DEFAULT\_BASE64;

            FirebaseDatabase.getInstance().getReference().child("user/" + user.getUid()).setValue(newUser);

        }

    }

}

**3. FriendsFragment.java**

package com.mcwc.heychat.ui;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.res.Resources;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.util.Base64;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.DefaultItemAnimator;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

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

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

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

import com.google.firebase.auth.FirebaseAuth;

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.mcwc.heychat.R;

import com.mcwc.heychat.data.FriendDB;

import com.mcwc.heychat.data.GroupDB;

import com.mcwc.heychat.data.SharedPreferenceHelper;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.Configuration;

import com.mcwc.heychat.model.User;

import com.mcwc.heychat.service.ServiceUtils;

import com.mcwc.heychat.util.ImageUtils;

import com.yarolegovich.lovelydialog.LovelyInfoDialog;

import com.yarolegovich.lovelydialog.LovelyProgressDialog;

import java.io.FileNotFoundException;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.List;

public class UserProfileFragment extends Fragment {

    TextView tvUserName;

    ImageView avatar;

    private List<Configuration> listConfig = new ArrayList<>();

    private RecyclerView recyclerView;

    private UserInfoAdapter infoAdapter;

    private static final String USERNAME\_LABEL = "Username";

    private static final String EMAIL\_LABEL = "Email";

    private static final String SIGNOUT\_LABEL = "Sign out";

    private static final String RESETPASS\_LABEL = "Change Password";

    private static final int PICK\_IMAGE = 1994;

    private LovelyProgressDialog waitingDialog;

    private DatabaseReference userDB;

    private FirebaseAuth mAuth;

    private User myAccount;

    private Context context;

    public UserProfileFragment() {

    }

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

    }

    private ValueEventListener userListener = new ValueEventListener() {

        @Override

        public void onDataChange(DataSnapshot dataSnapshot) {

            listConfig.clear();

            myAccount = dataSnapshot.getValue(User.class);

            setupArrayListInfo(myAccount);

            if(infoAdapter != null){

                infoAdapter.notifyDataSetChanged();

            }

            if(tvUserName != null){

                tvUserName.setText(myAccount.name);

            }

            setImageAvatar(context, myAccount.avata);

            SharedPreferenceHelper preferenceHelper = SharedPreferenceHelper.getInstance(context);

            preferenceHelper.saveUserInfo(myAccount);

        }

        @Override

        public void onCancelled(DatabaseError databaseError) {

            Log.e(UserProfileFragment.class.getName(), "loadPost:onCancelled", databaseError.toException());

        }

    };

    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container,

                             Bundle savedInstanceState) {

        userDB = FirebaseDatabase.getInstance().getReference().child("user").child(StaticConfig.UID);

        userDB.addListenerForSingleValueEvent(userListener);

        mAuth = FirebaseAuth.getInstance();

        View view = inflater.inflate(R.layout.fragment\_info, container, false);

        context = view.getContext();

        avatar = (ImageView) view.findViewById(R.id.img\_avatar);

        avatar.setOnClickListener(onAvatarClick);

        tvUserName = (TextView)view.findViewById(R.id.tv\_username);

        SharedPreferenceHelper prefHelper = SharedPreferenceHelper.getInstance(context);

        myAccount = prefHelper.getUserInfo();

        setupArrayListInfo(myAccount);

        setImageAvatar(context, myAccount.avata);

        tvUserName.setText(myAccount.name);

        recyclerView = (RecyclerView)view.findViewById(R.id.info\_recycler\_view);

        infoAdapter = new UserInfoAdapter(listConfig);

        RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(view.getContext());

        recyclerView.setLayoutManager(layoutManager);

        recyclerView.setItemAnimator(new DefaultItemAnimator());

        recyclerView.setAdapter(infoAdapter);

        waitingDialog = new LovelyProgressDialog(context);

        return view;

    }

    private View.OnClickListener onAvatarClick = new View.OnClickListener() {

        @Override

        public void onClick(View view) {

            new AlertDialog.Builder(context)

                    .setTitle("Avatar")

                    .setMessage("Are you sure want to change avatar profile?")

                    .setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

                        @Override

                        public void onClick(DialogInterface dialogInterface, int i) {

                            Intent intent = new Intent();

                            intent.setType("image/\*");

                            intent.setAction(Intent.ACTION\_PICK);

                            startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK\_IMAGE);

                            dialogInterface.dismiss();

                        }

                    })

                    .setNegativeButton(android.R.string.cancel, new DialogInterface.OnClickListener() {

                        @Override

                        public void onClick(DialogInterface dialogInterface, int i) {

                            dialogInterface.dismiss();

                        }

                    }).show();

        }

    };

    @Override

    public void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK\_IMAGE && resultCode == Activity.RESULT\_OK) {

            if (data == null) {

                Toast.makeText(context, "", Toast.LENGTH\_LONG).show();

                return;

            }

            try {

                InputStream inputStream = context.getContentResolver().openInputStream(data.getData());

                Bitmap imgBitmap = BitmapFactory.decodeStream(inputStream);

                imgBitmap = ImageUtils.cropToSquare(imgBitmap);

                InputStream is = ImageUtils.convertBitmapToInputStream(imgBitmap);

                final Bitmap liteImage = ImageUtils.makeImageLite(is,

                        imgBitmap.getWidth(), imgBitmap.getHeight(),

                        ImageUtils.AVATAR\_WIDTH, ImageUtils.AVATAR\_HEIGHT);

                String imageBase64 = ImageUtils.encodeBase64(liteImage);

                myAccount.avata = imageBase64;

                waitingDialog.setCancelable(false)

                        .setTitle("Avatar updating....")

                        .setTopColorRes(R.color.colorPrimary)

                        .show();

                userDB.child("avata").setValue(imageBase64)

                        .addOnCompleteListener(new OnCompleteListener<Void>() {

                            @Override

                            public void onComplete(@NonNull Task<Void> task) {

                                if(task.isSuccessful()){

                                    waitingDialog.dismiss();

                                    SharedPreferenceHelper preferenceHelper = SharedPreferenceHelper.getInstance(context);

                                    preferenceHelper.saveUserInfo(myAccount);

                                    avatar.setImageDrawable(ImageUtils.roundedImage(context, liteImage));

                                    new LovelyInfoDialog(context)

                                            .setTopColorRes(R.color.colorPrimary)

                                            .setTitle("Success")

                                            .setMessage("Update avatar successfully!")

                                            .show();

                                }

                            }

                         })

                        .addOnFailureListener(new OnFailureListener() {

                            @Override

                            public void onFailure(@NonNull Exception e) {

                                waitingDialog.dismiss();

                                Log.d("Update Avatar", "failed");

                                new LovelyInfoDialog(context)

                                        .setTopColorRes(R.color.colorAccent)

                                        .setTitle("Failed")

                                        .setMessage("Failed to update avatar")

                                        .show();

                            }

                        });

            } catch (FileNotFoundException e) {

                e.printStackTrace();

            }

        }

    }

    public void setupArrayListInfo(User myAccount){

        listConfig.clear();

        Configuration userNameConfig = new Configuration(USERNAME\_LABEL, myAccount.name, R.mipmap.ic\_account\_box);

        listConfig.add(userNameConfig);

        Configuration emailConfig = new Configuration(EMAIL\_LABEL, myAccount.email, R.mipmap.ic\_email);

        listConfig.add(emailConfig);

        Configuration resetPass = new Configuration(RESETPASS\_LABEL, "", R.mipmap.ic\_restore);

        listConfig.add(resetPass);

        Configuration signout = new Configuration(SIGNOUT\_LABEL, "", R.mipmap.ic\_power\_settings);

        listConfig.add(signout);

    }

    private void setImageAvatar(Context context, String imgBase64){

        try {

            Resources res = getResources();

            Bitmap src;

            if (imgBase64.equals("default")) {

                src = BitmapFactory.decodeResource(res, R.drawable.default\_avata);

            } else {

                byte[] decodedString = Base64.decode(imgBase64, Base64.DEFAULT);

                src = BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length);

            }

            avatar.setImageDrawable(ImageUtils.roundedImage(context, src));

        }catch (Exception e){

        }

    }

    @Override

    public void onDestroyView (){

        super.onDestroyView();

    }

    @Override

    public void onDestroy (){

        super.onDestroy();

    }

    public class UserInfoAdapter extends RecyclerView.Adapter<UserInfoAdapter.ViewHolder>{

        private List<Configuration> profileConfig;

        public UserInfoAdapter(List<Configuration> profileConfig){

            this.profileConfig = profileConfig;

        }

        @Override

        public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

            View itemView = LayoutInflater.from(parent.getContext())

                    .inflate(R.layout.list\_info\_item\_layout, parent, false);

            return new ViewHolder(itemView);

        }

        @Override

        public void onBindViewHolder(ViewHolder holder, int position) {

            final Configuration config = profileConfig.get(position);

            holder.label.setText(config.getLabel());

            holder.value.setText(config.getValue());

            holder.icon.setImageResource(config.getIcon());

            ((RelativeLayout)holder.label.getParent()).setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View view) {

                    if(config.getLabel().equals(SIGNOUT\_LABEL)){

                        FirebaseAuth.getInstance().signOut();

                        FriendDB.getInstance(getContext()).dropDB();

                        GroupDB.getInstance(getContext()).dropDB();

                        ServiceUtils.stopServiceFriendChat(getContext().getApplicationContext(), true);

                        getActivity().finish();

                    }

                    if(config.getLabel().equals(USERNAME\_LABEL)){

                        View vewInflater = LayoutInflater.from(context)

                                .inflate(R.layout.dialog\_edit\_username,  (ViewGroup) getView(), false);

                        final EditText input = (EditText)vewInflater.findViewById(R.id.edit\_username);

                        input.setText(myAccount.name);

                        new AlertDialog.Builder(context)

                                .setTitle("Edit username")

                                .setView(vewInflater)

                                .setPositiveButton("Save", new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        String newName = input.getText().toString();

                                        if(!myAccount.name.equals(newName)){

                                            changeUserName(newName);

                                        }

                                        dialogInterface.dismiss();

                                    }

                                })

                                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        dialogInterface.dismiss();

                                    }

                                }).show();

                    }

                    if(config.getLabel().equals(RESETPASS\_LABEL)){

                        new AlertDialog.Builder(context)

                                .setTitle("Password")

                                .setMessage("Are you sure want to reset password?")

                                .setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        resetPassword(myAccount.email);

                                        dialogInterface.dismiss();

                                    }

                                })

                                .setNegativeButton(android.R.string.cancel, new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        dialogInterface.dismiss();

                                    }

                                }).show();

                    }

                }

            });

        }

        private void changeUserName(String newName){

            userDB.child("name").setValue(newName);

            myAccount.name = newName;

            SharedPreferenceHelper prefHelper = SharedPreferenceHelper.getInstance(context);

            prefHelper.saveUserInfo(myAccount);

            tvUserName.setText(newName);

            setupArrayListInfo(myAccount);

        }

        void resetPassword(final String email) {

            mAuth.sendPasswordResetEmail(email)

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            new LovelyInfoDialog(context) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorPrimary)

                                    .setIcon(R.drawable.ic\_pass\_reset)

                                    .setTitle("Password Recovery")

                                    .setMessage("Sent email to " + email)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            new LovelyInfoDialog(context) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorAccent)

                                    .setIcon(R.drawable.ic\_pass\_reset)

                                    .setTitle("False")

                                    .setMessage("False to sent email to " + email)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    });

        }

        @Override

        public int getItemCount() {

            return profileConfig.size();

        }

        class ViewHolder extends RecyclerView.ViewHolder {

            public TextView label, value;

            public ImageView icon;

            public ViewHolder(View view) {

                super(view);

                label = (TextView)view.findViewById(R.id.tv\_title);

                value = (TextView)view.findViewById(R.id.tv\_detail);

                icon = (ImageView)view.findViewById(R.id.img\_icon);

            }

        }

    }

}

**4. GroupFragment.java**

package com.mcwc.heychat.ui;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.util.Base64;

import android.util.Log;

import android.view.ContextMenu;

import android.view.LayoutInflater;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ImageButton;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.GridLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

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

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

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

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.mcwc.heychat.R;

import com.mcwc.heychat.data.FriendDB;

import com.mcwc.heychat.data.GroupDB;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.Group;

import com.mcwc.heychat.model.ListFriend;

import com.yarolegovich.lovelydialog.LovelyInfoDialog;

import com.yarolegovich.lovelydialog.LovelyProgressDialog;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

public class GroupFragment extends Fragment implements SwipeRefreshLayout.OnRefreshListener{

    private RecyclerView recyclerListGroups;

    public FragGroupClickFloatButton onClickFloatButton;

    private ArrayList<Group> listGroup;

    private ListGroupsAdapter adapter;

    private SwipeRefreshLayout mSwipeRefreshLayout;

    public static final int CONTEXT\_MENU\_DELETE = 1;

    public static final int CONTEXT\_MENU\_EDIT = 2;

    public static final int CONTEXT\_MENU\_LEAVE = 3;

    public static final int REQUEST\_EDIT\_GROUP = 0;

    public static final String CONTEXT\_MENU\_KEY\_INTENT\_DATA\_POS = "pos";

    LovelyProgressDialog progressDialog, waitingLeavingGroup;

    public GroupFragment() {

    }

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

    }

    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container,

                             Bundle savedInstanceState) {

        View layout = inflater.inflate(R.layout.fragment\_group, container, false);

        listGroup = GroupDB.getInstance(getContext()).getListGroups();

        recyclerListGroups = (RecyclerView) layout.findViewById(R.id.recycleListGroup);

        mSwipeRefreshLayout = (SwipeRefreshLayout) layout.findViewById(R.id.swipeRefreshLayout);

        mSwipeRefreshLayout.setOnRefreshListener(this);

        GridLayoutManager layoutManager = new GridLayoutManager(getContext(), 2);

        recyclerListGroups.setLayoutManager(layoutManager);

        adapter = new ListGroupsAdapter(getContext(), listGroup);

        recyclerListGroups.setAdapter(adapter);

        onClickFloatButton = new FragGroupClickFloatButton();

        progressDialog = new LovelyProgressDialog(getContext())

                .setCancelable(false)

                .setIcon(R.drawable.ic\_dialog\_delete\_group)

                .setTitle("Deleting....")

                .setTopColorRes(R.color.colorAccent);

        waitingLeavingGroup = new LovelyProgressDialog(getContext())

                .setCancelable(false)

                .setIcon(R.drawable.ic\_dialog\_delete\_group)

                .setTitle("Group leaving....")

                .setTopColorRes(R.color.colorAccent);

        if(listGroup.size() == 0){

            mSwipeRefreshLayout.setRefreshing(true);

            getListGroup();

        }

        return layout;

    }

    private void getListGroup(){

        FirebaseDatabase.getInstance().getReference().child("user/"+ StaticConfig.UID+"/group").addListenerForSingleValueEvent(new ValueEventListener() {

            @Override

            public void onDataChange(DataSnapshot dataSnapshot) {

                if(dataSnapshot.getValue() != null) {

                    HashMap mapListGroup = (HashMap) dataSnapshot.getValue();

                    Iterator iterator = mapListGroup.keySet().iterator();

                    while (iterator.hasNext()){

                        String idGroup = (String) mapListGroup.get(iterator.next().toString());

                        Group newGroup = new Group();

                        newGroup.id = idGroup;

                        listGroup.add(newGroup);

                    }

                    getGroupInfo(0);

                }else{

                    mSwipeRefreshLayout.setRefreshing(false);

                    adapter.notifyDataSetChanged();

                }

            }

            @Override

            public void onCancelled(DatabaseError databaseError) {

                mSwipeRefreshLayout.setRefreshing(false);

            }

        });

    }

    @Override

    public void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == REQUEST\_EDIT\_GROUP && resultCode == Activity.RESULT\_OK) {

            listGroup.clear();

            ListGroupsAdapter.listFriend = null;

            GroupDB.getInstance(getContext()).dropDB();

            getListGroup();

        }

    }

    private void getGroupInfo(final int indexGroup){

        if(indexGroup == listGroup.size()){

            adapter.notifyDataSetChanged();

            mSwipeRefreshLayout.setRefreshing(false);

        }else {

            FirebaseDatabase.getInstance().getReference().child("group/"+listGroup.get(indexGroup).id).addListenerForSingleValueEvent(new ValueEventListener() {

                @Override

                public void onDataChange(DataSnapshot dataSnapshot) {

                    if(dataSnapshot.getValue() != null){

                        HashMap mapGroup = (HashMap) dataSnapshot.getValue();

                        ArrayList<String> member = (ArrayList<String>) mapGroup.get("member");

                        HashMap mapGroupInfo = (HashMap) mapGroup.get("groupInfo");

                        for(String idMember: member){

                            listGroup.get(indexGroup).member.add(idMember);

                        }

                        listGroup.get(indexGroup).groupInfo.put("name", (String) mapGroupInfo.get("name"));

                        listGroup.get(indexGroup).groupInfo.put("admin", (String) mapGroupInfo.get("admin"));

                    }

                    GroupDB.getInstance(getContext()).addGroup(listGroup.get(indexGroup));

                    Log.d("GroupFragment", listGroup.get(indexGroup).id +": " + dataSnapshot.toString());

                    getGroupInfo(indexGroup +1);

                }

                @Override

                public void onCancelled(DatabaseError databaseError) {

                }

            });

        }

    }

    @Override

    public void onRefresh() {

        listGroup.clear();

        ListGroupsAdapter.listFriend = null;

        GroupDB.getInstance(getContext()).dropDB();

        adapter.notifyDataSetChanged();

        getListGroup();

    }

    @Override

    public boolean onContextItemSelected(MenuItem item) {

        switch (item.getItemId()) {

            case CONTEXT\_MENU\_DELETE:

                int posGroup = item.getIntent().getIntExtra(CONTEXT\_MENU\_KEY\_INTENT\_DATA\_POS, -1);

                if(((String)listGroup.get(posGroup).groupInfo.get("admin")).equals(StaticConfig.UID)) {

                    Group group = listGroup.get(posGroup);

                    listGroup.remove(posGroup);

                    if(group != null){

                        deleteGroup(group, 0);

                    }

                }else{

                    Toast.makeText(getActivity(), "You are not admin", Toast.LENGTH\_LONG).show();

                }

                break;

            case CONTEXT\_MENU\_EDIT:

                int posGroup1 = item.getIntent().getIntExtra(CONTEXT\_MENU\_KEY\_INTENT\_DATA\_POS, -1);

                if(((String)listGroup.get(posGroup1).groupInfo.get("admin")).equals(StaticConfig.UID)) {

                    Intent intent = new Intent(getContext(), AddGroupActivity.class);

                    intent.putExtra("groupId", listGroup.get(posGroup1).id);

                    startActivityForResult(intent, REQUEST\_EDIT\_GROUP);

                }else{

                    Toast.makeText(getActivity(), "You are not admin", Toast.LENGTH\_LONG).show();

                }

                break;

            case CONTEXT\_MENU\_LEAVE:

                int position = item.getIntent().getIntExtra(CONTEXT\_MENU\_KEY\_INTENT\_DATA\_POS, -1);

                if(((String)listGroup.get(position).groupInfo.get("admin")).equals(StaticConfig.UID)) {

                    Toast.makeText(getActivity(), "Admin cannot leave group", Toast.LENGTH\_LONG).show();

                }else{

                    waitingLeavingGroup.show();

                    Group groupLeaving = listGroup.get(position);

                    leaveGroup(groupLeaving);

                }

                break;

        }

        return super.onContextItemSelected(item);

    }

    public void deleteGroup(final Group group, final int index){

        if(index == group.member.size()){

            FirebaseDatabase.getInstance().getReference().child("group/"+group.id).removeValue()

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            progressDialog.dismiss();

                            GroupDB.getInstance(getContext()).deleteGroup(group.id);

                            listGroup.remove(group);

                            adapter.notifyDataSetChanged();

                            Toast.makeText(getContext(), "Deleted group", Toast.LENGTH\_SHORT).show();

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            progressDialog.dismiss();

                            new LovelyInfoDialog(getContext())

                                    .setTopColorRes(R.color.colorAccent)

                                    .setIcon(R.drawable.ic\_dialog\_delete\_group)

                                    .setTitle("False")

                                    .setMessage("Cannot delete group right now, please try again.")

                                    .setCancelable(false)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    })

                    ;

        }else{

            FirebaseDatabase.getInstance().getReference().child("user/"+group.member.get(index)+"/group/"+group.id).removeValue()

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            deleteGroup(group, index + 1);

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            progressDialog.dismiss();

                            new LovelyInfoDialog(getContext())

                                    .setTopColorRes(R.color.colorAccent)

                                    .setIcon(R.drawable.ic\_dialog\_delete\_group)

                                    .setTitle("False")

                                    .setMessage("Cannot connect server")

                                    .setCancelable(false)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    })

            ;

        }

    }

    public void leaveGroup(final Group group){

        FirebaseDatabase.getInstance().getReference().child("group/"+group.id+"/member")

                .orderByValue().equalTo(StaticConfig.UID)

                .addListenerForSingleValueEvent(new ValueEventListener() {

                    @Override

                    public void onDataChange(DataSnapshot dataSnapshot) {

                        if (dataSnapshot.getValue() == null) {

                            //email not found

                            waitingLeavingGroup.dismiss();

                            new LovelyInfoDialog(getContext())

                                    .setTopColorRes(R.color.colorAccent)

                                    .setTitle("Error")

                                    .setMessage("Error occurred during leaving group")

                                    .show();

                        } else {

                            String memberIndex = "";

                            ArrayList<String> result = ((ArrayList<String>)dataSnapshot.getValue());

                            for(int i = 0; i < result.size(); i++){

                                if(result.get(i) != null){

                                    memberIndex = String.valueOf(i);

                                }

                            }

                            FirebaseDatabase.getInstance().getReference().child("user").child(StaticConfig.UID)

                                    .child("group").child(group.id).removeValue();

                            FirebaseDatabase.getInstance().getReference().child("group/"+group.id+"/member")

                                    .child(memberIndex).removeValue()

                                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                                        @Override

                                        public void onComplete(@NonNull Task<Void> task) {

                                            waitingLeavingGroup.dismiss();

                                            listGroup.remove(group);

                                            adapter.notifyDataSetChanged();

                                            GroupDB.getInstance(getContext()).deleteGroup(group.id);

                                            new LovelyInfoDialog(getContext())

                                                    .setTopColorRes(R.color.colorAccent)

                                                    .setTitle("Success")

                                                    .setMessage("Group leaving successfully")

                                                    .show();

                                        }

                                    })

                                    .addOnFailureListener(new OnFailureListener() {

                                        @Override

                                        public void onFailure(@NonNull Exception e) {

                                            waitingLeavingGroup.dismiss();

                                            new LovelyInfoDialog(getContext())

                                                    .setTopColorRes(R.color.colorAccent)

                                                    .setTitle("Error")

                                                    .setMessage("Error occurred during leaving group")

                                                    .show();

                                        }

                                    });

                        }

                    }

                    @Override

                    public void onCancelled(DatabaseError databaseError) {

                        //email not found

                        waitingLeavingGroup.dismiss();

                        new LovelyInfoDialog(getContext())

                                .setTopColorRes(R.color.colorAccent)

                                .setTitle("Error")

                                .setMessage("Error occurred during leaving group")

                                .show();

                    }

                });

    }

    public class FragGroupClickFloatButton implements View.OnClickListener{

        Context context;

        public FragGroupClickFloatButton getInstance(Context context){

            this.context = context;

            return this;

        }

        @Override

        public void onClick(View view) {

            startActivity(new Intent(getContext(), AddGroupActivity.class));

        }

    }

}

class ListGroupsAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    private ArrayList<Group> listGroup;

    public static ListFriend listFriend = null;

    private Context context;

    public ListGroupsAdapter(Context context,ArrayList<Group> listGroup){

        this.context = context;

        this.listGroup = listGroup;

    }

    @Override

    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View view = LayoutInflater.from(context).inflate(R.layout.rc\_item\_group, parent, false);

        return new ItemGroupViewHolder(view);

    }

    @Override

    public void onBindViewHolder(RecyclerView.ViewHolder holder, final int position) {

        final String groupName = listGroup.get(position).groupInfo.get("name");

        if(groupName != null && groupName.length() > 0) {

            ((ItemGroupViewHolder) holder).txtGroupName.setText(groupName);

            ((ItemGroupViewHolder) holder).iconGroup.setText((groupName.charAt(0) + "").toUpperCase());

        }

        ((ItemGroupViewHolder) holder).btnMore.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                view.setTag(new Object[]{groupName, position});

                view.getParent().showContextMenuForChild(view);

            }

        });

        ((RelativeLayout)((ItemGroupViewHolder) holder).txtGroupName.getParent()).setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                if(listFriend == null){

                    listFriend = FriendDB.getInstance(context).getListFriend();

                }

                Intent intent = new Intent(context, ChatActivity.class);

                intent.putExtra(StaticConfig.INTENT\_KEY\_CHAT\_FRIEND, groupName);

                ArrayList<CharSequence> idFriend = new ArrayList<>();

                ChatActivity.bitmapAvataFriend = new HashMap<>();

                for(String id : listGroup.get(position).member) {

                    idFriend.add(id);

                    String avata = listFriend.getAvataById(id);

                    if(!avata.equals(StaticConfig.STR\_DEFAULT\_BASE64)) {

                        byte[] decodedString = Base64.decode(avata, Base64.DEFAULT);

                        ChatActivity.bitmapAvataFriend.put(id, BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length));

                    }else if(avata.equals(StaticConfig.STR\_DEFAULT\_BASE64)) {

                        ChatActivity.bitmapAvataFriend.put(id, BitmapFactory.decodeResource(context.getResources(), R.drawable.default\_avata));

                    }else {

                        ChatActivity.bitmapAvataFriend.put(id, null);

                    }

                }

                intent.putCharSequenceArrayListExtra(StaticConfig.INTENT\_KEY\_CHAT\_ID, idFriend);

                intent.putExtra(StaticConfig.INTENT\_KEY\_CHAT\_ROOM\_ID, listGroup.get(position).id);

                context.startActivity(intent);

            }

        });

    }

    @Override

    public int getItemCount() {

        return listGroup.size();

    }

}

class ItemGroupViewHolder extends RecyclerView.ViewHolder implements View.OnCreateContextMenuListener {

    public TextView iconGroup, txtGroupName;

    public ImageButton btnMore;

    public ItemGroupViewHolder(View itemView) {

        super(itemView);

        itemView.setOnCreateContextMenuListener(this);

        iconGroup = (TextView) itemView.findViewById(R.id.icon\_group);

        txtGroupName = (TextView) itemView.findViewById(R.id.txtName);

        btnMore = (ImageButton) itemView.findViewById(R.id.btnMoreAction);

    }

    @Override

    public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo contextMenuInfo) {

        menu.setHeaderTitle((String) ((Object[])btnMore.getTag())[0]);

        Intent data = new Intent();

        data.putExtra(GroupFragment.CONTEXT\_MENU\_KEY\_INTENT\_DATA\_POS, (Integer) ((Object[])btnMore.getTag())[1]);

        menu.add(Menu.NONE, GroupFragment.CONTEXT\_MENU\_EDIT, Menu.NONE, "Edit group").setIntent(data);

        menu.add(Menu.NONE, GroupFragment.CONTEXT\_MENU\_DELETE, Menu.NONE, "Delete group").setIntent(data);

        menu.add(Menu.NONE, GroupFragment.CONTEXT\_MENU\_LEAVE, Menu.NONE, "Leave group").setIntent(data);

    }

}

**5. ChatActivity.java**

package com.mcwc.heychat.ui;

import android.content.Context;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.util.Base64;

import android.view.LayoutInflater;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.google.firebase.database.ChildEventListener;

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.mcwc.heychat.R;

import com.mcwc.heychat.data.SharedPreferenceHelper;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.Conversation;

import com.mcwc.heychat.model.Message;

import java.util.ArrayList;

import java.util.HashMap;

import de.hdodenhof.circleimageview.CircleImageView;

public class ChatActivity extends AppCompatActivity implements View.OnClickListener {

    private RecyclerView recyclerChat;

    public static final int VIEW\_TYPE\_USER\_MESSAGE = 0;

    public static final int VIEW\_TYPE\_FRIEND\_MESSAGE = 1;

    private ListMessageAdapter adapter;

    private String roomId;

    private ArrayList<CharSequence> idFriend;

    private Conversation consersation;

    private ImageButton btnSend;

    private EditText editWriteMessage;

    private LinearLayoutManager linearLayoutManager;

    public static HashMap<String, Bitmap> bitmapAvataFriend;

    public Bitmap bitmapAvataUser;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_chat);

        Intent intentData = getIntent();

        idFriend = intentData.getCharSequenceArrayListExtra(StaticConfig.INTENT\_KEY\_CHAT\_ID);

        roomId = intentData.getStringExtra(StaticConfig.INTENT\_KEY\_CHAT\_ROOM\_ID);

        String nameFriend = intentData.getStringExtra(StaticConfig.INTENT\_KEY\_CHAT\_FRIEND);

        consersation = new Conversation();

        btnSend = (ImageButton) findViewById(R.id.btnSend);

        btnSend.setOnClickListener(this);

        String base64AvataUser = SharedPreferenceHelper.getInstance(this).getUserInfo().avata;

        if (!base64AvataUser.equals(StaticConfig.STR\_DEFAULT\_BASE64)) {

            byte[] decodedString = Base64.decode(base64AvataUser, Base64.DEFAULT);

            bitmapAvataUser = BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length);

        } else {

            bitmapAvataUser = null;

        }

        editWriteMessage = (EditText) findViewById(R.id.editWriteMessage);

        if (idFriend != null && nameFriend != null) {

            getSupportActionBar().setTitle(nameFriend);

            linearLayoutManager = new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);

            recyclerChat = (RecyclerView) findViewById(R.id.recyclerChat);

            recyclerChat.setLayoutManager(linearLayoutManager);

            adapter = new ListMessageAdapter(this, consersation, bitmapAvataFriend, bitmapAvataUser);

            FirebaseDatabase.getInstance().getReference().child("message/" + roomId).addChildEventListener(new ChildEventListener() {

                @Override

                public void onChildAdded(DataSnapshot dataSnapshot, String s) {

                    if (dataSnapshot.getValue() != null) {

                        HashMap mapMessage = (HashMap) dataSnapshot.getValue();

                        Message newMessage = new Message();

                        newMessage.idSender = (String) mapMessage.get("idSender");

                        newMessage.idReceiver = (String) mapMessage.get("idReceiver");

                        newMessage.text = (String) mapMessage.get("text");

                        newMessage.timestamp = (long) mapMessage.get("timestamp");

                        consersation.getListMessageData().add(newMessage);

                        adapter.notifyDataSetChanged();

                        linearLayoutManager.scrollToPosition(consersation.getListMessageData().size() - 1);

                    }

                }

                @Override

                public void onChildChanged(DataSnapshot dataSnapshot, String s) {

                }

                @Override

                public void onChildRemoved(DataSnapshot dataSnapshot) {

                }

                @Override

                public void onChildMoved(DataSnapshot dataSnapshot, String s) {

                }

                @Override

                public void onCancelled(DatabaseError databaseError) {

                }

            });

            recyclerChat.setAdapter(adapter);

        }

    }

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

        if(item.getItemId() == android.R.id.home){

            Intent result = new Intent();

            result.putExtra("idFriend", idFriend.get(0));

            setResult(RESULT\_OK, result);

            this.finish();

        }

        return true;

    }

    @Override

    public void onBackPressed() {

        Intent result = new Intent();

        result.putExtra("idFriend", idFriend.get(0));

        setResult(RESULT\_OK, result);

        this.finish();

    }

    @Override

    public void onClick(View view) {

        if (view.getId() == R.id.btnSend) {

            String content = editWriteMessage.getText().toString().trim();

            if (content.length() > 0) {

                editWriteMessage.setText("");

                Message newMessage = new Message();

                newMessage.text = content;

                newMessage.idSender = StaticConfig.UID;

                newMessage.idReceiver = roomId;

                newMessage.timestamp = System.currentTimeMillis();

                FirebaseDatabase.getInstance().getReference().child("message/" + roomId).push().setValue(newMessage);

            }

        }

    }

}

class ListMessageAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    private Context context;

    private Conversation consersation;

    private HashMap<String, Bitmap> bitmapAvata;

    private HashMap<String, DatabaseReference> bitmapAvataDB;

    private Bitmap bitmapAvataUser;

    public ListMessageAdapter(Context context, Conversation consersation, HashMap<String, Bitmap> bitmapAvata, Bitmap bitmapAvataUser) {

        this.context = context;

        this.consersation = consersation;

        this.bitmapAvata = bitmapAvata;

        this.bitmapAvataUser = bitmapAvataUser;

        bitmapAvataDB = new HashMap<>();

    }

    @Override

    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        if (viewType == ChatActivity.VIEW\_TYPE\_FRIEND\_MESSAGE) {

            View view = LayoutInflater.from(context).inflate(R.layout.rc\_item\_message\_friend, parent, false);

            return new ItemMessageFriendHolder(view);

        } else if (viewType == ChatActivity.VIEW\_TYPE\_USER\_MESSAGE) {

            View view = LayoutInflater.from(context).inflate(R.layout.rc\_item\_message\_user, parent, false);

            return new ItemMessageUserHolder(view);

        }

        return null;

    }

    @Override

    public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {

        if (holder instanceof ItemMessageFriendHolder) {

            ((ItemMessageFriendHolder) holder).txtContent.setText(consersation.getListMessageData().get(position).text);

            Bitmap currentAvata = bitmapAvata.get(consersation.getListMessageData().get(position).idSender);

            if (currentAvata != null) {

                ((ItemMessageFriendHolder) holder).avata.setImageBitmap(currentAvata);

            } else {

                final String id = consersation.getListMessageData().get(position).idSender;

                if(bitmapAvataDB.get(id) == null){

                    bitmapAvataDB.put(id, FirebaseDatabase.getInstance().getReference().child("user/" + id + "/avata"));

                    bitmapAvataDB.get(id).addListenerForSingleValueEvent(new ValueEventListener() {

                        @Override

                        public void onDataChange(DataSnapshot dataSnapshot) {

                            if (dataSnapshot.getValue() != null) {

                                String avataStr = (String) dataSnapshot.getValue();

                                if(!avataStr.equals(StaticConfig.STR\_DEFAULT\_BASE64)) {

                                    byte[] decodedString = Base64.decode(avataStr, Base64.DEFAULT);

                                    ChatActivity.bitmapAvataFriend.put(id, BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length));

                                }else{

                                    ChatActivity.bitmapAvataFriend.put(id, BitmapFactory.decodeResource(context.getResources(), R.drawable.default\_avata));

                                }

                                notifyDataSetChanged();

                            }

                        }

                        @Override

                        public void onCancelled(DatabaseError databaseError) {

                        }

                    });

                }

            }

        } else if (holder instanceof ItemMessageUserHolder) {

            ((ItemMessageUserHolder) holder).txtContent.setText(consersation.getListMessageData().get(position).text);

            if (bitmapAvataUser != null) {

                ((ItemMessageUserHolder) holder).avata.setImageBitmap(bitmapAvataUser);

            }

        }

    }

    @Override

    public int getItemViewType(int position) {

        return consersation.getListMessageData().get(position).idSender.equals(StaticConfig.UID) ? ChatActivity.VIEW\_TYPE\_USER\_MESSAGE : ChatActivity.VIEW\_TYPE\_FRIEND\_MESSAGE;

    }

    @Override

    public int getItemCount() {

        return consersation.getListMessageData().size();

    }

}

class ItemMessageUserHolder extends RecyclerView.ViewHolder {

    public TextView txtContent;

    public CircleImageView avata;

    public ItemMessageUserHolder(View itemView) {

        super(itemView);

        txtContent = (TextView) itemView.findViewById(R.id.textContentUser);

        avata = (CircleImageView) itemView.findViewById(R.id.imageView2);

    }

}

class ItemMessageFriendHolder extends RecyclerView.ViewHolder {

    public TextView txtContent;

    public CircleImageView avata;

    public ItemMessageFriendHolder(View itemView) {

        super(itemView);

        txtContent = (TextView) itemView.findViewById(R.id.textContentFriend);

        avata = (CircleImageView) itemView.findViewById(R.id.imageView3);

    }

}

1. **AddGroupActivity.java**

package com.mcwc.heychat.ui;

import android.content.Context;

import android.content.Intent;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.text.Editable;

import android.text.TextWatcher;

import android.util.Base64;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.CheckBox;

import android.widget.CompoundButton;

import android.widget.EditText;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

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

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

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

import com.google.firebase.database.FirebaseDatabase;

import com.mcwc.heychat.R;

import com.mcwc.heychat.data.FriendDB;

import com.mcwc.heychat.data.GroupDB;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.Group;

import com.mcwc.heychat.model.ListFriend;

import com.mcwc.heychat.model.Room;

import com.yarolegovich.lovelydialog.LovelyInfoDialog;

import com.yarolegovich.lovelydialog.LovelyProgressDialog;

import java.util.HashSet;

import java.util.Set;

import de.hdodenhof.circleimageview.CircleImageView;

public class AddGroupActivity extends AppCompatActivity {

    private RecyclerView recyclerListFriend;

    private ListPeopleAdapter adapter;

    private ListFriend listFriend;

    private LinearLayout btnAddGroup;

    private Set<String> listIDChoose;

    private Set<String> listIDRemove;

    private EditText editTextGroupName;

    private TextView txtGroupIcon, txtActionName;

    private LovelyProgressDialog dialogWait;

    private boolean isEditGroup;

    private Group groupEdit;

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_add\_group);

        Intent intentData = getIntent();

        txtActionName = (TextView) findViewById(R.id.txtActionName);

        LinearLayoutManager linearLayoutManager = new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);

        listFriend = FriendDB.getInstance(this).getListFriend();

        listIDChoose = new HashSet<>();

        listIDRemove = new HashSet<>();

        listIDChoose.add(StaticConfig.UID);

        btnAddGroup = (LinearLayout) findViewById(R.id.btnAddGroup);

        editTextGroupName = (EditText) findViewById(R.id.editGroupName);

        txtGroupIcon = (TextView) findViewById(R.id.icon\_group);

        dialogWait = new LovelyProgressDialog(this).setCancelable(false);

        editTextGroupName.addTextChangedListener(new TextWatcher() {

            @Override

            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override

            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

                if (charSequence.length() >= 1) {

                    txtGroupIcon.setText((charSequence.charAt(0) + "").toUpperCase());

                } else {

                    txtGroupIcon.setText("R");

                }

            }

            @Override

            public void afterTextChanged(Editable editable) {

            }

        });

        btnAddGroup.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                if (listIDChoose.size() < 3) {

                    Toast.makeText(AddGroupActivity.this, "Add at lease two people to create group", Toast.LENGTH\_SHORT).show();

                } else {

                    if (editTextGroupName.getText().length() == 0) {

                        Toast.makeText(AddGroupActivity.this, "Enter group name", Toast.LENGTH\_SHORT).show();

                    } else {

                        if (isEditGroup) {

                            editGroup();

                        } else {

                            createGroup();

                        }

                    }

                }

            }

        });

        if (intentData.getStringExtra("groupId") != null) {

            isEditGroup = true;

            String idGroup = intentData.getStringExtra("groupId");

            txtActionName.setText("Save");

            btnAddGroup.setBackgroundColor(getResources().getColor(R.color.colorPrimary));

            groupEdit = GroupDB.getInstance(this).getGroup(idGroup);

            editTextGroupName.setText(groupEdit.groupInfo.get("name"));

        } else {

            isEditGroup = false;

        }

        recyclerListFriend = (RecyclerView) findViewById(R.id.recycleListFriend);

        recyclerListFriend.setLayoutManager(linearLayoutManager);

        adapter = new ListPeopleAdapter(this, listFriend, btnAddGroup, listIDChoose, listIDRemove, isEditGroup, groupEdit);

        recyclerListFriend.setAdapter(adapter);

    }

    private void editGroup() {

        dialogWait.setIcon(R.drawable.ic\_add\_group\_dialog)

                .setTitle("Editing....")

                .setTopColorRes(R.color.colorPrimary)

                .show();

        final String idGroup = groupEdit.id;

        Room room = new Room();

        for (String id : listIDChoose) {

            room.member.add(id);

        }

        room.groupInfo.put("name", editTextGroupName.getText().toString());

        room.groupInfo.put("admin", StaticConfig.UID);

        FirebaseDatabase.getInstance().getReference().child("group/" + idGroup).setValue(room)

                .addOnCompleteListener(new OnCompleteListener<Void>() {

                    @Override

                    public void onComplete(@NonNull Task<Void> task) {

                        addRoomForUser(idGroup, 0);

                    }

                })

                .addOnFailureListener(new OnFailureListener() {

                    @Override

                    public void onFailure(@NonNull Exception e) {

                        dialogWait.dismiss();

                        new LovelyInfoDialog(AddGroupActivity.this) {

                            @Override

                            public LovelyInfoDialog setConfirmButtonText(String text) {

                                findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                    @Override

                                    public void onClick(View view) {

                                        dismiss();

                                    }

                                });

                                return super.setConfirmButtonText(text);

                            }

                        }

                                .setTopColorRes(R.color.colorAccent)

                                .setIcon(R.drawable.ic\_add\_group\_dialog)

                                .setTitle("False")

                                .setMessage("Cannot connect database")

                                .setCancelable(false)

                                .setConfirmButtonText("Ok")

                                .show();

                    }

                })

        ;

    }

    private void createGroup() {

        dialogWait.setIcon(R.drawable.ic\_add\_group\_dialog)

                .setTitle("Registering....")

                .setTopColorRes(R.color.colorPrimary)

                .show();

        final String idGroup = (StaticConfig.UID + System.currentTimeMillis()).hashCode() + "";

        Room room = new Room();

        for (String id : listIDChoose) {

            room.member.add(id);

        }

        room.groupInfo.put("name", editTextGroupName.getText().toString());

        room.groupInfo.put("admin", StaticConfig.UID);

        FirebaseDatabase.getInstance().getReference().child("group/" + idGroup).setValue(room).addOnCompleteListener(new OnCompleteListener<Void>() {

            @Override

            public void onComplete(@NonNull Task<Void> task) {

                addRoomForUser(idGroup, 0);

            }

        });

    }

    private void deleteRoomForUser(final String roomId, final int userIndex) {

        if (userIndex == listIDRemove.size()) {

            dialogWait.dismiss();

            Toast.makeText(this, "Edit group success", Toast.LENGTH\_SHORT).show();

            setResult(RESULT\_OK, null);

            AddGroupActivity.this.finish();

        } else {

            FirebaseDatabase.getInstance().getReference().child("user/" + listIDRemove.toArray()[userIndex] + "/group/" + roomId).removeValue()

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            deleteRoomForUser(roomId, userIndex + 1);

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            dialogWait.dismiss();

                            new LovelyInfoDialog(AddGroupActivity.this) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorAccent)

                                    .setIcon(R.drawable.ic\_add\_group\_dialog)

                                    .setTitle("False")

                                    .setMessage("Cannot connect database")

                                    .setCancelable(false)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    });

        }

    }

    private void addRoomForUser(final String roomId, final int userIndex) {

        if (userIndex == listIDChoose.size()) {

            if (!isEditGroup) {

                dialogWait.dismiss();

                Toast.makeText(this, "Create group success", Toast.LENGTH\_SHORT).show();

                setResult(RESULT\_OK, null);

                AddGroupActivity.this.finish();

            } else {

                deleteRoomForUser(roomId, 0);

            }

        } else {

            FirebaseDatabase.getInstance().getReference().child("user/" + listIDChoose.toArray()[userIndex] + "/group/" + roomId).setValue(roomId).addOnCompleteListener(new OnCompleteListener<Void>() {

                @Override

                public void onComplete(@NonNull Task<Void> task) {

                    addRoomForUser(roomId, userIndex + 1);

                }

            }).addOnFailureListener(new OnFailureListener() {

                @Override

                public void onFailure(@NonNull Exception e) {

                    dialogWait.dismiss();

                    new LovelyInfoDialog(AddGroupActivity.this) {

                        @Override

                        public LovelyInfoDialog setConfirmButtonText(String text) {

                            findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                @Override

                                public void onClick(View view) {

                                    dismiss();

                                }

                            });

                            return super.setConfirmButtonText(text);

                        }

                    }

                            .setTopColorRes(R.color.colorAccent)

                            .setIcon(R.drawable.ic\_add\_group\_dialog)

                            .setTitle("False")

                            .setMessage("Create group false")

                            .setCancelable(false)

                            .setConfirmButtonText("Ok")

                            .show();

                }

            });

        }

    }

}

class ListPeopleAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {

    private Context context;

    private ListFriend listFriend;

    private LinearLayout btnAddGroup;

    private Set<String> listIDChoose;

    private Set<String> listIDRemove;

    private boolean isEdit;

    private Group editGroup;

    public ListPeopleAdapter(Context context, ListFriend listFriend, LinearLayout btnAddGroup, Set<String> listIDChoose, Set<String> listIDRemove, boolean isEdit, Group editGroup) {

        this.context = context;

        this.listFriend = listFriend;

        this.btnAddGroup = btnAddGroup;

        this.listIDChoose = listIDChoose;

        this.listIDRemove = listIDRemove;

        this.isEdit = isEdit;

        this.editGroup = editGroup;

    }

    @Override

    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View view = LayoutInflater.from(context).inflate(R.layout.rc\_item\_add\_friend, parent, false);

        return new ItemFriendHolder(view);

    }

    @Override

    public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {

        ((ItemFriendHolder) holder).txtName.setText(listFriend.getListFriend().get(position).name);

        ((ItemFriendHolder) holder).txtEmail.setText(listFriend.getListFriend().get(position).email);

        String avata = listFriend.getListFriend().get(position).avata;

        final String id = listFriend.getListFriend().get(position).id;

        if (!avata.equals(StaticConfig.STR\_DEFAULT\_BASE64)) {

            byte[] decodedString = Base64.decode(avata, Base64.DEFAULT);

            ((ItemFriendHolder) holder).avata.setImageBitmap(BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length));

        }else{

            ((ItemFriendHolder) holder).avata.setImageBitmap(BitmapFactory.decodeResource(context.getResources(), R.drawable.default\_avata));

        }

        ((ItemFriendHolder) holder).checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {

            @Override

            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {

                if (b) {

                    listIDChoose.add(id);

                    listIDRemove.remove(id);

                } else {

                    listIDRemove.add(id);

                    listIDChoose.remove(id);

                }

                if (listIDChoose.size() >= 3) {

                    btnAddGroup.setBackgroundColor(context.getResources().getColor(R.color.colorPrimary));

                } else {

                    btnAddGroup.setBackgroundColor(context.getResources().getColor(R.color.grey\_500));

                }

            }

        });

        if (isEdit && editGroup.member.contains(id)) {

            ((ItemFriendHolder) holder).checkBox.setChecked(true);

        }else if(editGroup != null && !editGroup.member.contains(id)){

            ((ItemFriendHolder) holder).checkBox.setChecked(false);

        }

    }

    @Override

    public int getItemCount() {

        return listFriend.getListFriend().size();

    }

}

class ItemFriendHolder extends RecyclerView.ViewHolder {

    public TextView txtName, txtEmail;

    public CircleImageView avata;

    public CheckBox checkBox;

    public ItemFriendHolder(View itemView) {

        super(itemView);

        txtName = (TextView) itemView.findViewById(R.id.txtName);

        txtEmail = (TextView) itemView.findViewById(R.id.txtEmail);

        avata = (CircleImageView) itemView.findViewById(R.id.icon\_avata);

        checkBox = (CheckBox) itemView.findViewById(R.id.checkAddPeople);

    }

}

1. **UserProfileFragment.java**

package com.mcwc.heychat.ui;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.res.Resources;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.Bundle;

import android.util.Base64;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.DefaultItemAnimator;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

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

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

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

import com.google.firebase.auth.FirebaseAuth;

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.mcwc.heychat.R;

import com.mcwc.heychat.data.FriendDB;

import com.mcwc.heychat.data.GroupDB;

import com.mcwc.heychat.data.SharedPreferenceHelper;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.model.Configuration;

import com.mcwc.heychat.model.User;

import com.mcwc.heychat.service.ServiceUtils;

import com.mcwc.heychat.util.ImageUtils;

import com.yarolegovich.lovelydialog.LovelyInfoDialog;

import com.yarolegovich.lovelydialog.LovelyProgressDialog;

import java.io.FileNotFoundException;

import java.io.InputStream;

import java.util.ArrayList;

import java.util.List;

public class UserProfileFragment extends Fragment {

    TextView tvUserName;

    ImageView avatar;

    private List<Configuration> listConfig = new ArrayList<>();

    private RecyclerView recyclerView;

    private UserInfoAdapter infoAdapter;

    private static final String USERNAME\_LABEL = "Username";

    private static final String EMAIL\_LABEL = "Email";

    private static final String SIGNOUT\_LABEL = "Sign out";

    private static final String RESETPASS\_LABEL = "Change Password";

    private static final int PICK\_IMAGE = 1994;

    private LovelyProgressDialog waitingDialog;

    private DatabaseReference userDB;

    private FirebaseAuth mAuth;

    private User myAccount;

    private Context context;

    public UserProfileFragment() {

    }

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

    }

    private ValueEventListener userListener = new ValueEventListener() {

        @Override

        public void onDataChange(DataSnapshot dataSnapshot) {

            listConfig.clear();

            myAccount = dataSnapshot.getValue(User.class);

            setupArrayListInfo(myAccount);

            if(infoAdapter != null){

                infoAdapter.notifyDataSetChanged();

            }

            if(tvUserName != null){

                tvUserName.setText(myAccount.name);

            }

            setImageAvatar(context, myAccount.avata);

            SharedPreferenceHelper preferenceHelper = SharedPreferenceHelper.getInstance(context);

            preferenceHelper.saveUserInfo(myAccount);

        }

        @Override

        public void onCancelled(DatabaseError databaseError) {

            Log.e(UserProfileFragment.class.getName(), "loadPost:onCancelled", databaseError.toException());

        }

    };

    @Override

    public View onCreateView(LayoutInflater inflater, ViewGroup container,

                             Bundle savedInstanceState) {

        userDB = FirebaseDatabase.getInstance().getReference().child("user").child(StaticConfig.UID);

        userDB.addListenerForSingleValueEvent(userListener);

        mAuth = FirebaseAuth.getInstance();

        View view = inflater.inflate(R.layout.fragment\_info, container, false);

        context = view.getContext();

        avatar = (ImageView) view.findViewById(R.id.img\_avatar);

        avatar.setOnClickListener(onAvatarClick);

        tvUserName = (TextView)view.findViewById(R.id.tv\_username);

        SharedPreferenceHelper prefHelper = SharedPreferenceHelper.getInstance(context);

        myAccount = prefHelper.getUserInfo();

        setupArrayListInfo(myAccount);

        setImageAvatar(context, myAccount.avata);

        tvUserName.setText(myAccount.name);

        recyclerView = (RecyclerView)view.findViewById(R.id.info\_recycler\_view);

        infoAdapter = new UserInfoAdapter(listConfig);

        RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(view.getContext());

        recyclerView.setLayoutManager(layoutManager);

        recyclerView.setItemAnimator(new DefaultItemAnimator());

        recyclerView.setAdapter(infoAdapter);

        waitingDialog = new LovelyProgressDialog(context);

        return view;

    }

    private View.OnClickListener onAvatarClick = new View.OnClickListener() {

        @Override

        public void onClick(View view) {

            new AlertDialog.Builder(context)

                    .setTitle("Avatar")

                    .setMessage("Are you sure want to change avatar profile?")

                    .setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

                        @Override

                        public void onClick(DialogInterface dialogInterface, int i) {

                            Intent intent = new Intent();

                            intent.setType("image/\*");

                            intent.setAction(Intent.ACTION\_PICK);

                            startActivityForResult(Intent.createChooser(intent, "Select Picture"), PICK\_IMAGE);

                            dialogInterface.dismiss();

                        }

                    })

                    .setNegativeButton(android.R.string.cancel, new DialogInterface.OnClickListener() {

                        @Override

                        public void onClick(DialogInterface dialogInterface, int i) {

                            dialogInterface.dismiss();

                        }

                    }).show();

        }

    };

    @Override

    public void onActivityResult(int requestCode, int resultCode, Intent data) {

        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK\_IMAGE && resultCode == Activity.RESULT\_OK) {

            if (data == null) {

                Toast.makeText(context, "", Toast.LENGTH\_LONG).show();

                return;

            }

            try {

                InputStream inputStream = context.getContentResolver().openInputStream(data.getData());

                Bitmap imgBitmap = BitmapFactory.decodeStream(inputStream);

                imgBitmap = ImageUtils.cropToSquare(imgBitmap);

                InputStream is = ImageUtils.convertBitmapToInputStream(imgBitmap);

                final Bitmap liteImage = ImageUtils.makeImageLite(is,

                        imgBitmap.getWidth(), imgBitmap.getHeight(),

                        ImageUtils.AVATAR\_WIDTH, ImageUtils.AVATAR\_HEIGHT);

                String imageBase64 = ImageUtils.encodeBase64(liteImage);

                myAccount.avata = imageBase64;

                waitingDialog.setCancelable(false)

                        .setTitle("Avatar updating....")

                        .setTopColorRes(R.color.colorPrimary)

                        .show();

                userDB.child("avata").setValue(imageBase64)

                        .addOnCompleteListener(new OnCompleteListener<Void>() {

                            @Override

                            public void onComplete(@NonNull Task<Void> task) {

                                if(task.isSuccessful()){

                                    waitingDialog.dismiss();

                                    SharedPreferenceHelper preferenceHelper = SharedPreferenceHelper.getInstance(context);

                                    preferenceHelper.saveUserInfo(myAccount);

                                    avatar.setImageDrawable(ImageUtils.roundedImage(context, liteImage));

                                    new LovelyInfoDialog(context)

                                            .setTopColorRes(R.color.colorPrimary)

                                            .setTitle("Success")

                                            .setMessage("Update avatar successfully!")

                                            .show();

                                }

                            }

                         })

                        .addOnFailureListener(new OnFailureListener() {

                            @Override

                            public void onFailure(@NonNull Exception e) {

                                waitingDialog.dismiss();

                                Log.d("Update Avatar", "failed");

                                new LovelyInfoDialog(context)

                                        .setTopColorRes(R.color.colorAccent)

                                        .setTitle("Failed")

                                        .setMessage("Failed to update avatar")

                                        .show();

                            }

                        });

            } catch (FileNotFoundException e) {

                e.printStackTrace();

            }

        }

    }

    public void setupArrayListInfo(User myAccount){

        listConfig.clear();

        Configuration userNameConfig = new Configuration(USERNAME\_LABEL, myAccount.name, R.mipmap.ic\_account\_box);

        listConfig.add(userNameConfig);

        Configuration emailConfig = new Configuration(EMAIL\_LABEL, myAccount.email, R.mipmap.ic\_email);

        listConfig.add(emailConfig);

        Configuration resetPass = new Configuration(RESETPASS\_LABEL, "", R.mipmap.ic\_restore);

        listConfig.add(resetPass);

        Configuration signout = new Configuration(SIGNOUT\_LABEL, "", R.mipmap.ic\_power\_settings);

        listConfig.add(signout);

    }

    private void setImageAvatar(Context context, String imgBase64){

        try {

            Resources res = getResources();

            Bitmap src;

            if (imgBase64.equals("default")) {

                src = BitmapFactory.decodeResource(res, R.drawable.default\_avata);

            } else {

                byte[] decodedString = Base64.decode(imgBase64, Base64.DEFAULT);

                src = BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length);

            }

            avatar.setImageDrawable(ImageUtils.roundedImage(context, src));

        }catch (Exception e){

        }

    }

    @Override

    public void onDestroyView (){

        super.onDestroyView();

    }

    @Override

    public void onDestroy (){

        super.onDestroy();

    }

    public class UserInfoAdapter extends RecyclerView.Adapter<UserInfoAdapter.ViewHolder>{

        private List<Configuration> profileConfig;

        public UserInfoAdapter(List<Configuration> profileConfig){

            this.profileConfig = profileConfig;

        }

        @Override

        public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

            View itemView = LayoutInflater.from(parent.getContext())

                    .inflate(R.layout.list\_info\_item\_layout, parent, false);

            return new ViewHolder(itemView);

        }

        @Override

        public void onBindViewHolder(ViewHolder holder, int position) {

            final Configuration config = profileConfig.get(position);

            holder.label.setText(config.getLabel());

            holder.value.setText(config.getValue());

            holder.icon.setImageResource(config.getIcon());

            ((RelativeLayout)holder.label.getParent()).setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View view) {

                    if(config.getLabel().equals(SIGNOUT\_LABEL)){

                        FirebaseAuth.getInstance().signOut();

                        FriendDB.getInstance(getContext()).dropDB();

                        GroupDB.getInstance(getContext()).dropDB();

                        ServiceUtils.stopServiceFriendChat(getContext().getApplicationContext(), true);

                        getActivity().finish();

                    }

                    if(config.getLabel().equals(USERNAME\_LABEL)){

                        View vewInflater = LayoutInflater.from(context)

                                .inflate(R.layout.dialog\_edit\_username,  (ViewGroup) getView(), false);

                        final EditText input = (EditText)vewInflater.findViewById(R.id.edit\_username);

                        input.setText(myAccount.name);

                        new AlertDialog.Builder(context)

                                .setTitle("Edit username")

                                .setView(vewInflater)

                                .setPositiveButton("Save", new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        String newName = input.getText().toString();

                                        if(!myAccount.name.equals(newName)){

                                            changeUserName(newName);

                                        }

                                        dialogInterface.dismiss();

                                    }

                                })

                                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        dialogInterface.dismiss();

                                    }

                                }).show();

                    }

                    if(config.getLabel().equals(RESETPASS\_LABEL)){

                        new AlertDialog.Builder(context)

                                .setTitle("Password")

                                .setMessage("Are you sure want to reset password?")

                                .setPositiveButton(android.R.string.ok, new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        resetPassword(myAccount.email);

                                        dialogInterface.dismiss();

                                    }

                                })

                                .setNegativeButton(android.R.string.cancel, new DialogInterface.OnClickListener() {

                                    @Override

                                    public void onClick(DialogInterface dialogInterface, int i) {

                                        dialogInterface.dismiss();

                                    }

                                }).show();

                    }

                }

            });

        }

        private void changeUserName(String newName){

            userDB.child("name").setValue(newName);

            myAccount.name = newName;

            SharedPreferenceHelper prefHelper = SharedPreferenceHelper.getInstance(context);

            prefHelper.saveUserInfo(myAccount);

            tvUserName.setText(newName);

            setupArrayListInfo(myAccount);

        }

        void resetPassword(final String email) {

            mAuth.sendPasswordResetEmail(email)

                    .addOnCompleteListener(new OnCompleteListener<Void>() {

                        @Override

                        public void onComplete(@NonNull Task<Void> task) {

                            new LovelyInfoDialog(context) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorPrimary)

                                    .setIcon(R.drawable.ic\_pass\_reset)

                                    .setTitle("Password Recovery")

                                    .setMessage("Sent email to " + email)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    })

                    .addOnFailureListener(new OnFailureListener() {

                        @Override

                        public void onFailure(@NonNull Exception e) {

                            new LovelyInfoDialog(context) {

                                @Override

                                public LovelyInfoDialog setConfirmButtonText(String text) {

                                    findView(com.yarolegovich.lovelydialog.R.id.ld\_btn\_confirm).setOnClickListener(new View.OnClickListener() {

                                        @Override

                                        public void onClick(View view) {

                                            dismiss();

                                        }

                                    });

                                    return super.setConfirmButtonText(text);

                                }

                            }

                                    .setTopColorRes(R.color.colorAccent)

                                    .setIcon(R.drawable.ic\_pass\_reset)

                                    .setTitle("False")

                                    .setMessage("False to sent email to " + email)

                                    .setConfirmButtonText("Ok")

                                    .show();

                        }

                    });

        }

        @Override

        public int getItemCount() {

            return profileConfig.size();

        }

        class ViewHolder extends RecyclerView.ViewHolder {

            public TextView label, value;

            public ImageView icon;

            public ViewHolder(View view) {

                super(view);

                label = (TextView)view.findViewById(R.id.tv\_title);

                value = (TextView)view.findViewById(R.id.tv\_detail);

                icon = (ImageView)view.findViewById(R.id.img\_icon);

            }

        }

    }

}

1. **MainActivity.java**

package com.mcwc.heychat;

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.view.Window;

import android.view.WindowManager;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import androidx.fragment.app.Fragment;

import androidx.fragment.app.FragmentManager;

import androidx.fragment.app.FragmentPagerAdapter;

import androidx.viewpager.widget.ViewPager;

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

import com.google.android.material.tabs.TabLayout;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.mcwc.heychat.data.StaticConfig;

import com.mcwc.heychat.service.ServiceUtils;

import com.mcwc.heychat.ui.FriendsFragment;

import com.mcwc.heychat.ui.GroupFragment;

import com.mcwc.heychat.ui.LoginActivity;

import com.mcwc.heychat.ui.UserProfileFragment;

import java.util.ArrayList;

import java.util.List;

public class MainActivity extends AppCompatActivity {

    private static String TAG = "MainActivity";

    private ViewPager viewPager;

    private TabLayout tabLayout = null;

    public static String STR\_FRIEND\_FRAGMENT = "FRIEND";

    public static String STR\_GROUP\_FRAGMENT = "GROUP";

    public static String STR\_INFO\_FRAGMENT = "INFO";

    private FloatingActionButton floatButton;

    private ViewPagerAdapter adapter;

    private FirebaseAuth mAuth;

    private FirebaseAuth.AuthStateListener mAuthListener;

    private FirebaseUser user;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        requestWindowFeature(Window.FEATURE\_NO\_TITLE);

        getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

                WindowManager.LayoutParams.FLAG\_FULLSCREEN);

        setContentView(R.layout.activity\_main);

        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

        if(toolbar != null) {

            setSupportActionBar(toolbar);

            getSupportActionBar().setTitle("Heychat");

        }

        viewPager = (ViewPager) findViewById(R.id.viewpager);

        floatButton = (FloatingActionButton) findViewById(R.id.fab);

        initTab();

        initFirebase();

    }

    private void initFirebase() {

        mAuth = FirebaseAuth.getInstance();

        mAuthListener = new FirebaseAuth.AuthStateListener() {

            @Override

            public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {

                user = firebaseAuth.getCurrentUser();

                if (user != null) {

                    StaticConfig.UID = user.getUid();

                } else {

                    MainActivity.this.finish();

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

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

                }

                // ...

            }

        };

    }

    @Override

    protected void onStart() {

        super.onStart();

        mAuth.addAuthStateListener(mAuthListener);

        ServiceUtils.stopServiceFriendChat(getApplicationContext(), false);

    }

    @Override

    protected void onStop() {

        super.onStop();

        if (mAuthListener != null) {

            mAuth.removeAuthStateListener(mAuthListener);

        }

    }

    @Override

    protected void onDestroy() {

        ServiceUtils.startServiceFriendChat(getApplicationContext());

        super.onDestroy();

    }

    private void initTab() {

        tabLayout = (TabLayout) findViewById(R.id.tabs);

        tabLayout.setSelectedTabIndicatorColor(getResources().getColor(R.color.colorIndivateTab));

        setupViewPager(viewPager);

        tabLayout.setupWithViewPager(viewPager);

        setupTabIcons();

    }

    private void setupTabIcons() {

        int[] tabIcons = {

                R.drawable.ic\_tab\_person,

                R.drawable.ic\_tab\_group,

                R.drawable.ic\_tab\_infor

        };

        tabLayout.getTabAt(0).setIcon(tabIcons[0]);

        tabLayout.getTabAt(1).setIcon(tabIcons[1]);

        tabLayout.getTabAt(2).setIcon(tabIcons[2]);

    }

    private void setupViewPager(ViewPager viewPager) {

        adapter = new ViewPagerAdapter(getSupportFragmentManager());

        adapter.addFrag(new FriendsFragment(), STR\_FRIEND\_FRAGMENT);

        adapter.addFrag(new GroupFragment(), STR\_GROUP\_FRAGMENT);

        adapter.addFrag(new UserProfileFragment(), STR\_INFO\_FRAGMENT);

        floatButton.setOnClickListener(((FriendsFragment) adapter.getItem(0)).onClickFloatButton.getInstance(this));

        viewPager.setAdapter(adapter);

        viewPager.setOffscreenPageLimit(3);

        viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {

            @Override

            public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

            }

            @Override

            public void onPageSelected(int position) {

                ServiceUtils.stopServiceFriendChat(MainActivity.this.getApplicationContext(), false);

                if (adapter.getItem(position) instanceof FriendsFragment) {

                    floatButton.setVisibility(View.VISIBLE);

                    floatButton.setOnClickListener(((FriendsFragment) adapter.getItem(position)).onClickFloatButton.getInstance(MainActivity.this));

                    floatButton.setImageResource(R.drawable.plus);

                } else if (adapter.getItem(position) instanceof GroupFragment) {

                    floatButton.setVisibility(View.VISIBLE);

                    floatButton.setOnClickListener(((GroupFragment) adapter.getItem(position)).onClickFloatButton.getInstance(MainActivity.this));

                    floatButton.setImageResource(R.drawable.ic\_float\_add\_group);

                } else {

                    floatButton.setVisibility(View.GONE);

                }

            }

            @Override

            public void onPageScrollStateChanged(int state) {

            }

        });

    }

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.menu\_main, menu);

        return true;

    }

    @Override

    public boolean onOptionsItemSelected(MenuItem item) {

        int id = item.getItemId();

        if (id == R.id.about) {

            Toast.makeText(this, "Hey Chat! Created by Group 12 - MCWC", Toast.LENGTH\_LONG).show();

            return true;

        }

        return super.onOptionsItemSelected(item);

    }

    class ViewPagerAdapter extends FragmentPagerAdapter {

        private final List<Fragment> mFragmentList = new ArrayList<>();

        private final List<String> mFragmentTitleList = new ArrayList<>();

        public ViewPagerAdapter(FragmentManager manager) {

            super(manager);

        }

        @Override

        public Fragment getItem(int position) {

            return mFragmentList.get(position);

        }

        @Override

        public int getCount() {

            return mFragmentList.size();

        }

        public void addFrag(Fragment fragment, String title) {

            mFragmentList.add(fragment);

            mFragmentTitleList.add(title);

        }

        @Override

        public CharSequence getPageTitle(int position) {

            return null;

        }

    }

}

**XML FILE**

1. **Activity\_add\_group.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical"

    android:background="@color/colorIndivateTab">

    <LinearLayout

        android:id="@+id/btnAddGroup"

        android:layout\_width="match\_parent"

        android:layout\_height="50dp"

        android:background="@color/grey\_500"

        android:gravity="center|right"

        android:orientation="horizontal"

        android:paddingRight="20dp">

        <ImageView

            android:layout\_width="40dp"

            android:layout\_height="40dp"

            android:padding="7dp"

            android:src="@drawable/ic\_accept\_cr\_group" />

        <TextView

            android:id="@+id/txtActionName"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="Create group"

            android:textAllCaps="true"

            android:textColor="@color/colorIndivateTab"

            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout

        android:layout\_width="match\_parent"

        android:layout\_height="100dp"

        android:gravity="center"

        android:orientation="horizontal"

        android:paddingEnd="20dp"

        android:paddingStart="20dp"

        android:focusable="true"

        android:focusableInTouchMode="true">

        <TextView

            android:id="@+id/icon\_group"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:background="@drawable/rounded\_textview"

            android:gravity="center"

            android:text="R"

            android:textColor="@android:color/white"

            android:textSize="30sp" />

        <EditText

            android:id="@+id/editGroupName"

            android:layout\_width="0dp"

            android:layout\_height="wrap\_content"

            android:layout\_marginLeft="20dp"

            android:layout\_weight="3"

            android:hint="@string/enter\_group\_name"

            android:paddingBottom="20dp"

            android:paddingLeft="20dp" />

    </LinearLayout>

    <View

        android:id="@+id/line"

        android:layout\_width="match\_parent"

        android:layout\_height="1dp"

        android:background="@color/grey\_200" />

    <androidx.recyclerview.widget.RecyclerView

        android:id="@+id/recycleListFriend"

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent"

        android:paddingStart="10dp"

        android:paddingEnd="10dp"

        />

</LinearLayout>

1. **Activity\_login.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"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent">

    <androidx.cardview.widget.CardView

        android:id="@+id/cv"

        android:layout\_width="300dp"

        android:layout\_height="300dp"

        android:layout\_centerInParent="true"

        app:cardCornerRadius="6dp"

        app:cardElevation="3dp"

        android:layout\_alignParentBottom="true"

        android:layout\_marginBottom="60dp"

        app:cardUseCompatPadding="true">

        <LinearLayout

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:orientation="vertical"

            android:background="#16ADEE">

            <RelativeLayout

                android:layout\_width="301dp"

                android:layout\_height="28dp"

                android:layout\_marginTop="10dp">

                <View

                    android:layout\_width="8dp"

                    android:layout\_height="match\_parent"

                    android:layout\_alignParentStart="true"

                    android:background="#41CBC7" />

                <TextView

                    android:layout\_width="wrap\_content"

                    android:layout\_height="wrap\_content"

                    android:layout\_centerVertical="true"

                    android:layout\_marginStart="50dp"

                    android:text="@string/login"

                    android:textColor="#FFFFFF"

                    android:textSize="18sp"

                    android:textStyle="bold" />

            </RelativeLayout>

            <LinearLayout

                android:layout\_width="match\_parent"

                android:layout\_height="50dp"

                android:layout\_marginTop="10dp"

                android:orientation="horizontal"

                android:paddingEnd="30dp"

                android:paddingStart="50dp">

                <com.google.android.material.textfield.TextInputLayout

                    android:layout\_width="match\_parent"

                    android:layout\_height="wrap\_content"

                    android:textColorHint="#FFFFFF">

                    <EditText

                        android:id="@+id/et\_username"

                        android:layout\_width="match\_parent"

                        android:layout\_height="wrap\_content"

                        android:background="@drawable/selector\_bg\_edit"

                        android:hint="@string/Username"

                        android:inputType="textEmailAddress"

                        android:drawablePadding="15dp"

                        android:drawableLeft="@drawable/ic\_baseline\_email\_24"

                        android:paddingBottom="2dp"

                        android:textColor="#FFFFFF"

                        android:textCursorDrawable="@drawable/bg\_input\_cursor"

                        android:textSize="13sp" />

                </com.google.android.material.textfield.TextInputLayout>

            </LinearLayout>

            <LinearLayout

                android:layout\_width="match\_parent"

                android:layout\_height="50dp"

                android:orientation="horizontal"

                android:paddingEnd="30dp"

                android:paddingStart="50dp">

                <com.google.android.material.textfield.TextInputLayout

                    android:layout\_width="match\_parent"

                    android:layout\_height="wrap\_content"

                    android:textColorHint="#FFFFFF">

                    <EditText

                        android:id="@+id/et\_password"

                        android:layout\_width="match\_parent"

                        android:layout\_height="wrap\_content"

                        android:background="@drawable/selector\_bg\_edit"

                        android:hint="@string/Password"

                        android:inputType="textPassword"

                        android:paddingBottom="10dp"

                        android:textColor="#FFFFFF"

                        android:textCursorDrawable="@drawable/bg\_input\_cursor"

                        android:drawableLeft="@drawable/ic\_baseline\_lock\_24"

                        android:drawablePadding="15dp"

                        android:textSize="13sp" />

                </com.google.android.material.textfield.TextInputLayout>

            </LinearLayout>

            <RelativeLayout

                android:layout\_width="match\_parent"

                android:layout\_height="60dp"

                android:layout\_marginTop="25dp"

                android:gravity="center">

                <Button

                    android:id="@+id/bt\_go"

                    android:layout\_width="150dp"

                    android:layout\_height="50dp"

                    android:background="@drawable/bt\_shape"

                    android:stateListAnimator="@drawable/state\_list\_animator\_z"

                    android:text="@string/go"

                    android:textColor="#FFFFFF"

                    android:onClick="clickLogin"/>

            </RelativeLayout>

            <TextView

                android:layout\_width="wrap\_content"

                android:layout\_height="wrap\_content"

                android:layout\_gravity="center\_horizontal"

                android:layout\_marginTop="5dp"

                android:text="@string/forgot\_your\_password"

                android:textColor="#FFFFFF"

                android:textSize="12sp"

                android:drawableLeft="@drawable/ic\_pass\_reset"

                android:drawablePadding="15dp"

                android:onClick="clickResetPassword"/>

        </LinearLayout>

    </androidx.cardview.widget.CardView>

    <com.google.android.material.floatingactionbutton.FloatingActionButton

        android:id="@+id/fab"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignEnd="@id/cv"

        android:layout\_alignTop="@id/cv"

        android:layout\_marginEnd="-20dp"

        android:layout\_marginTop="25dp"

        android:onClick="clickRegisterLayout"

        android:src="@drawable/ic\_signup"

        android:transitionName="loginFab"

        app:fabSize="normal" />

</RelativeLayout>

1. **Activity\_chat.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical"

    android:background="@color/colorIndivateTab">

    <ImageButton

        android:id="@+id/btnSend"

        android:layout\_width="45dp"

        android:layout\_height="45dp"

        android:layout\_alignParentBottom="true"

        android:layout\_alignParentEnd="true"

        android:background="#00000000"

        android:padding="10dp"

        android:scaleType="centerInside"

        android:src="@drawable/ic\_send" />

    <View

        android:id="@+id/line"

        android:layout\_width="match\_parent"

        android:layout\_height="1dp"

        android:background="@color/grey\_200"

        android:layout\_above="@id/btnSend"/>

    <androidx.recyclerview.widget.RecyclerView

        android:id="@+id/recyclerChat"

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent"

        android:layout\_above="@id/line">

    </androidx.recyclerview.widget.RecyclerView>

    <EditText

        android:id="@+id/editWriteMessage"

        android:layout\_width="match\_parent"

        android:layout\_height="45dp"

        android:layout\_alignParentBottom="true"

        android:layout\_alignParentStart="true"

        android:layout\_marginEnd="20dp"

        android:layout\_marginTop="10dp"

        android:layout\_toStartOf="@id/btnSend"

        android:background="@drawable/edittext\_boder"

        android:drawablePadding="20dp"

        android:inputType="textLongMessage"

        android:maxLines="1"

        android:paddingLeft="7dp"

        android:paddingRight="7dp"

        android:textColorHint="#aeafaf"

        android:textSize="15sp" />

</RelativeLayout>

1. **Activity\_register.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">

    <FrameLayout

        android:layout\_width="320dp"

        android:layout\_height="400dp"

        android:layout\_centerInParent="true"

        android:layout\_alignParentBottom="true"

        android:paddingTop="30dp">

        <androidx.cardview.widget.CardView

            android:id="@+id/cv\_add"

            android:layout\_width="match\_parent"

            android:layout\_height="335dp"

            android:layout\_gravity="center\_horizontal"

            android:layout\_marginTop="10dp"

            app:cardBackgroundColor="#251532"

            app:cardCornerRadius="6dp"

            app:cardElevation="3dp"

            app:cardUseCompatPadding="true">

            <LinearLayout

                android:layout\_width="match\_parent"

                android:layout\_height="match\_parent"

                android:orientation="vertical">

                <TextView

                    android:layout\_width="wrap\_content"

                    android:layout\_height="wrap\_content"

                    android:layout\_marginStart="50dp"

                    android:layout\_marginTop="50dp"

                    android:text="@string/register"

                    android:textColor="#fff"

                    android:textSize="16sp"

                    android:textStyle="bold" />

                <LinearLayout

                    android:layout\_width="match\_parent"

                    android:layout\_height="40dp"

                    android:layout\_marginTop="10dp"

                    android:paddingEnd="30dp"

                    android:paddingStart="50dp">

                    <com.google.android.material.textfield.TextInputLayout

                        android:layout\_width="match\_parent"

                        android:layout\_height="wrap\_content"

                        android:textColorHint="#f0f7f4"

                        android:theme="@style/TextLabel">

                        <EditText

                            android:id="@+id/et\_username"

                            android:layout\_width="match\_parent"

                            android:layout\_height="wrap\_content"

                            android:background="@drawable/selector\_bg\_edit\_2"

                            android:hint="@string/Username"

                            android:inputType="textEmailAddress"

                            android:paddingBottom="2dp"

                            android:textColor="#FFFFFF"

                            android:textCursorDrawable="@drawable/bg\_input\_cursor\_2"

                            android:drawablePadding="15dp"

                            android:drawableLeft="@drawable/ic\_baseline\_email\_24"

                            android:textSize="13sp" />

                    </com.google.android.material.textfield.TextInputLayout>

                </LinearLayout>

                <LinearLayout

                    android:layout\_width="match\_parent"

                    android:layout\_height="40dp"

                    android:orientation="horizontal"

                    android:paddingEnd="30dp"

                    android:paddingStart="50dp">

                    <com.google.android.material.textfield.TextInputLayout

                        android:layout\_width="match\_parent"

                        android:layout\_height="wrap\_content"

                        android:textColorHint="#f0f7f4"

                        android:theme="@style/TextLabel">

                        <EditText

                            android:id="@+id/et\_password"

                            android:layout\_width="match\_parent"

                            android:layout\_height="wrap\_content"

                            android:background="@drawable/selector\_bg\_edit\_2"

                            android:hint="@string/Password"

                            android:inputType="textPassword"

                            android:paddingBottom="10dp"

                            android:textColor="#f0f7f4"

                            android:textCursorDrawable="@drawable/bg\_input\_cursor\_2"

                            android:drawablePadding="15dp"

                            android:drawableLeft="@drawable/ic\_baseline\_lock\_24"

                            android:textSize="13sp" />

                    </com.google.android.material.textfield.TextInputLayout>

                </LinearLayout>

                <LinearLayout

                    android:layout\_width="match\_parent"

                    android:layout\_height="40dp"

                    android:orientation="horizontal"

                    android:paddingEnd="30dp"

                    android:paddingStart="50dp">

                    <com.google.android.material.textfield.TextInputLayout

                        android:layout\_width="match\_parent"

                        android:layout\_height="wrap\_content"

                        android:textColorHint="#f0f7f4"

                        android:theme="@style/TextLabel">

                        <EditText

                            android:id="@+id/et\_repeatpassword"

                            android:layout\_width="match\_parent"

                            android:layout\_height="wrap\_content"

                            android:background="@drawable/selector\_bg\_edit\_2"

                            android:hint="@string/repeat\_password"

                            android:inputType="textPassword"

                            android:paddingBottom="10dp"

                            android:textColor="#f0f7f4"

                            android:drawablePadding="15dp"

                            android:drawableLeft="@drawable/ic\_baseline\_phonelink\_lock\_24"

                            android:textCursorDrawable="@drawable/bg\_input\_cursor\_2"

                            android:textSize="13sp" />

                    </com.google.android.material.textfield.TextInputLayout>

                </LinearLayout>

                <Button

                    android:id="@+id/bt\_go"

                    android:layout\_width="150dp"

                    android:layout\_height="50dp"

                    android:layout\_gravity="center\_horizontal"

                    android:layout\_marginTop="20dp"

                    android:background="@drawable/bt\_shape\_2"

                    android:stateListAnimator="@drawable/state\_list\_animator\_z"

                    android:text="@string/go"

                    android:textColor="#251532"

                    android:onClick="clickRegister"/>

            </LinearLayout>

        </androidx.cardview.widget.CardView>

        <com.google.android.material.floatingactionbutton.FloatingActionButton

            android:id="@+id/fab"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_gravity="center\_horizontal|top"

            android:onClick="clickBackLogin"

            android:src="@drawable/plus\_x"

            android:transitionName="loginFab"

            app:fabSize="normal" />

    </FrameLayout>

</RelativeLayout>

1. **activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.coordinatorlayout.widget.CoordinatorLayout 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:fitsSystemWindows="true"

    tools:context="com.mcwc.heychat.MainActivity">

    <com.google.android.material.appbar.AppBarLayout

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:background="?attr/colorPrimary"

        android:focusable="true"

        android:focusableInTouchMode="true"

        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar

            android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"

            android:layout\_height="wrap\_content"

            android:layout\_width="match\_parent"

            android:minHeight="?attr/actionBarSize"

            android:background="?attr/colorPrimary"

            android:id="@+id/toolbar" />

        <com.google.android.material.tabs.TabLayout

            android:id="@+id/tabs"

            android:layout\_width="match\_parent"

            android:layout\_height="50dp"

            android:layout\_gravity="bottom"

            app:tabGravity="fill"

            app:tabMode="fixed" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content\_main" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton

        android:id="@+id/fab"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_gravity="bottom|end"

        android:layout\_margin="@dimen/fab\_margin"

        app:srcCompat="@drawable/ic\_float\_button"

        app:backgroundTint="@color/colorPrimary"

        />

</androidx.coordinatorlayout.widget.CoordinatorLayout>

1. **fragment\_group.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent">

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout

        android:id="@+id/swipeRefreshLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent">

        <androidx.recyclerview.widget.RecyclerView

            android:id="@+id/recycleListGroup"

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_marginEnd="10dp"

            android:layout\_marginStart="10dp"

            android:layout\_marginTop="15dp" />

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

</RelativeLayout>

1. **fragment\_info.xml**

<androidx.percentlayout.widget.PercentRelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent">

    <RelativeLayout

        android:id="@+id/profile"

        android:background="@color/grey\_100"

        app:layout\_aspectRatio="150%"

        app:layout\_widthPercent="100%"

        android:layout\_height="wrap\_content"

        android:layout\_width="wrap\_content">

        <LinearLayout

            android:layout\_centerVertical="true"

            android:layout\_width="match\_parent"

            android:orientation="vertical"

            android:layout\_marginTop="10dp"

            android:layout\_marginBottom="10dp"

            android:layout\_height="wrap\_content">

            <ImageView

                android:id="@+id/img\_avatar"

                android:layout\_gravity="center"

                android:layout\_width="104dp"

                android:layout\_height="104dp"

                android:src="@drawable/default\_avata"

                android:background="#00000000"/>

            <TextView

                android:layout\_gravity="center"

                android:textSize="24sp"

                android:textColor="@android:color/black"

                android:layout\_marginTop="32dp"

                android:id="@+id/tv\_username"

                android:layout\_width="wrap\_content"

                android:layout\_height="wrap\_content"/>

        </LinearLayout>

    </RelativeLayout>

    <androidx.recyclerview.widget.RecyclerView

        android:layout\_below="@+id/profile"

        android:id="@+id/info\_recycler\_view"

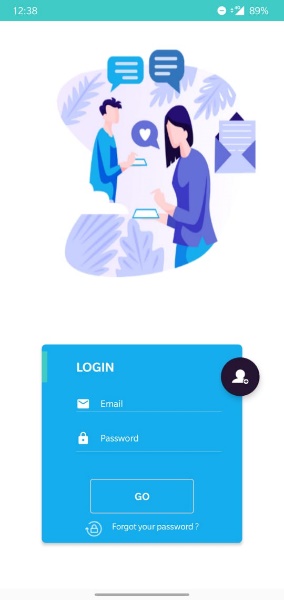
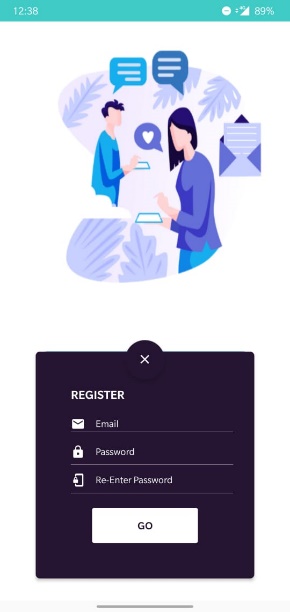
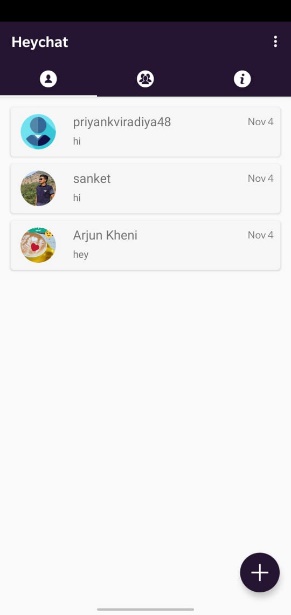
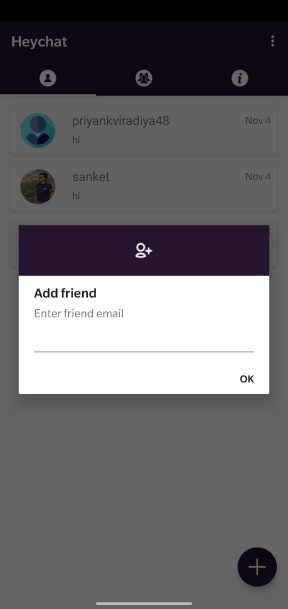
        android:layout\_width="match\_parent"

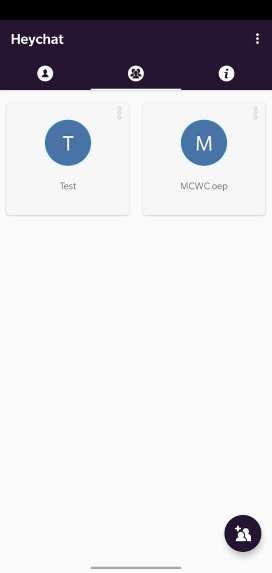
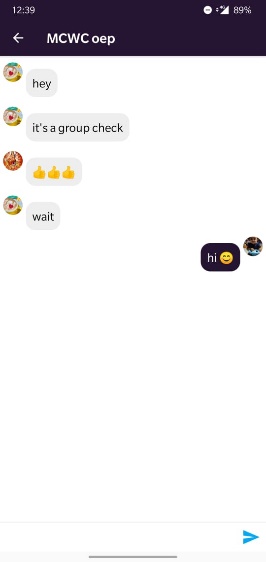
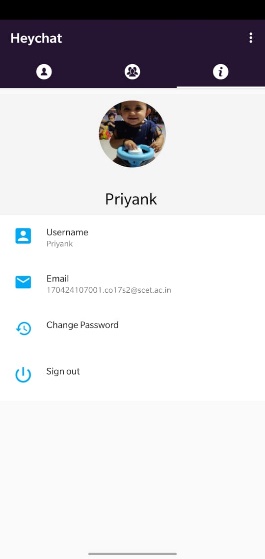
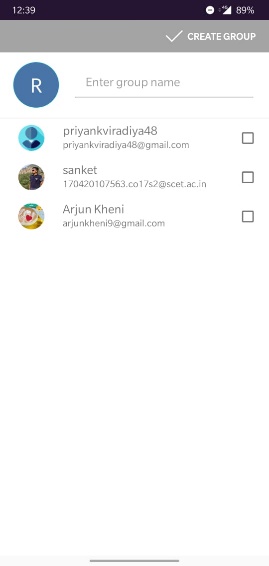
        android:layout\_height="wrap\_content"

        android:scrollbars="vertical" />

</androidx.percentlayout.widget.PercentRelativeLayout>

**Output:**

**   **

**    **