

# **Robby Chat Application**

## **Introduction**

- Robby Chat app it Is the Name Of Android Application.
- Basically This Application Build for Messaging purpose.
- Robby Chat app is a multi-activity Application which aims to User Can Easily Communicate with other User.
- It also have Current users Status so that other user can seen Friends online Status.
- You can send friend request And Accept Friend Request. Also Unfriend your friend.

## **Features**

- Send Free Message via Internet.
- Add Friends , Remove Friend.
- Set Your Profile Picture
- Set your Status.

# Requirements

## Specification

- **Operating System** : - Android
- **Version Compatibility** :- Android Nougat And Above

## Software Specification

### software's :-

Android Studio.

Android Studio is the official integrated development environment for Google's Android operating system, built on Jet Brains IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

### Language Used :-

Java.

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented.

Xml.

XML stands for eXtensible Markup Language.

XML was designed to store and transport data.

Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.

You can also use Android Studio's Layout Editor to build your XML layout using a drag-and-drop interface.

## **Back End :-**

### **Firebase:**

Firebase is a Backend-as-a-Service — BaaS .

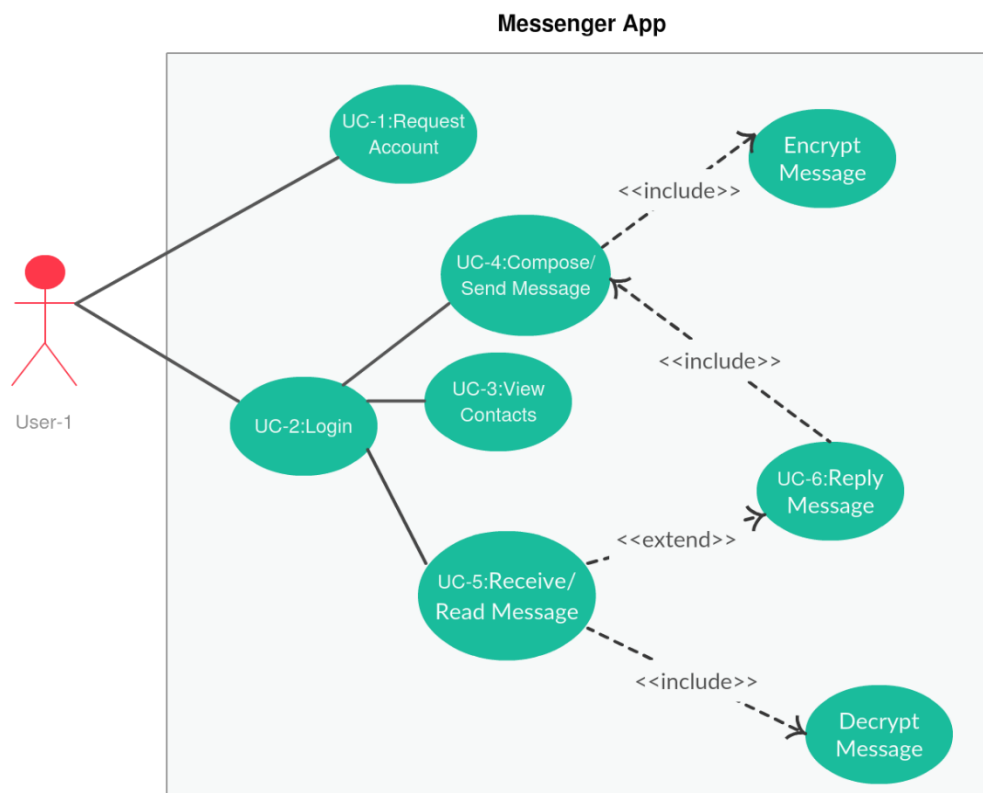
Firebase is built on Google infrastructure and scales automatically, for even the largest apps.

Firebase gives you functionality like analytics, databases, messaging and crash reporting so you can move quickly and focus on your users.

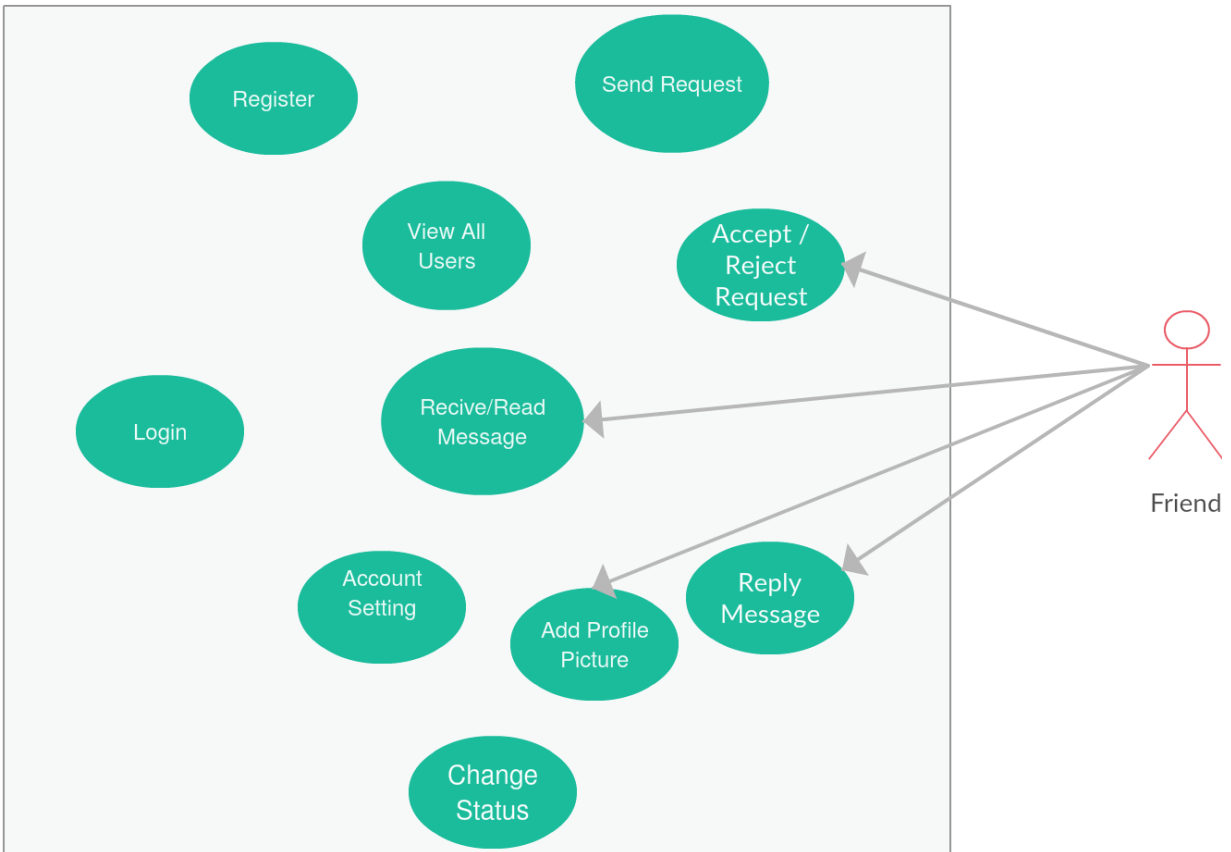
# System Design Details

## Use Case Diagrams

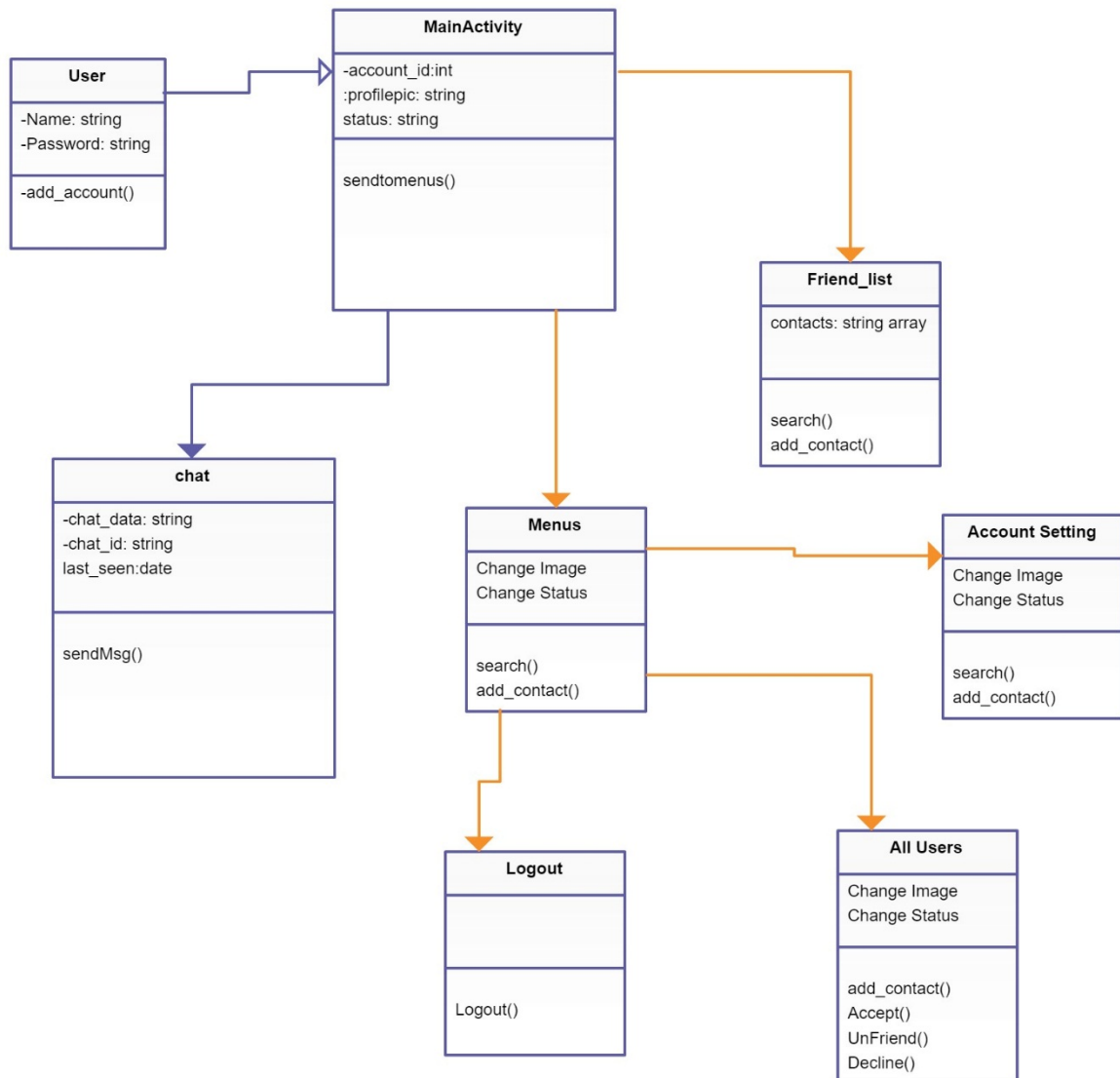
### 1. User-1



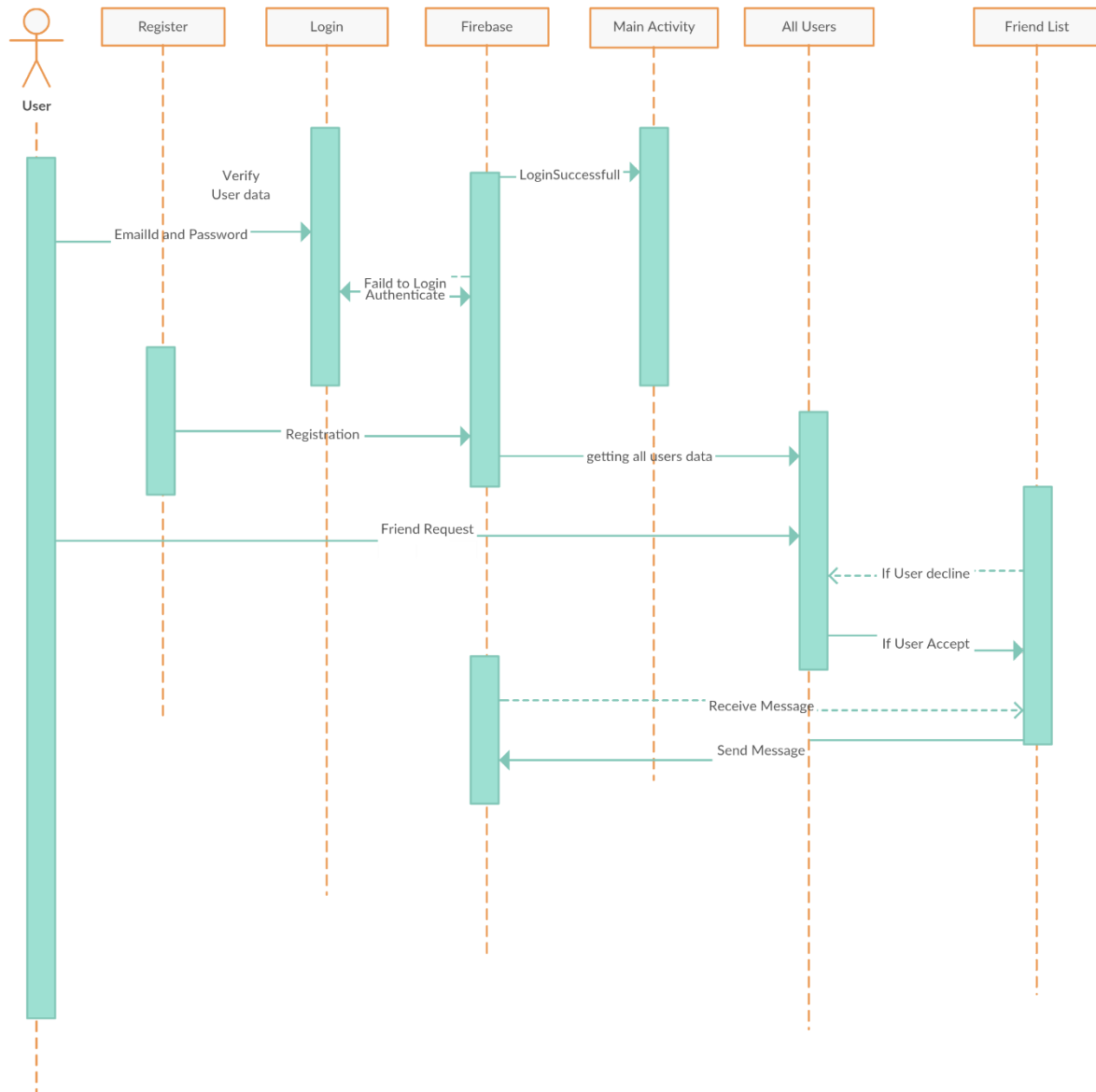
## 2. User-2



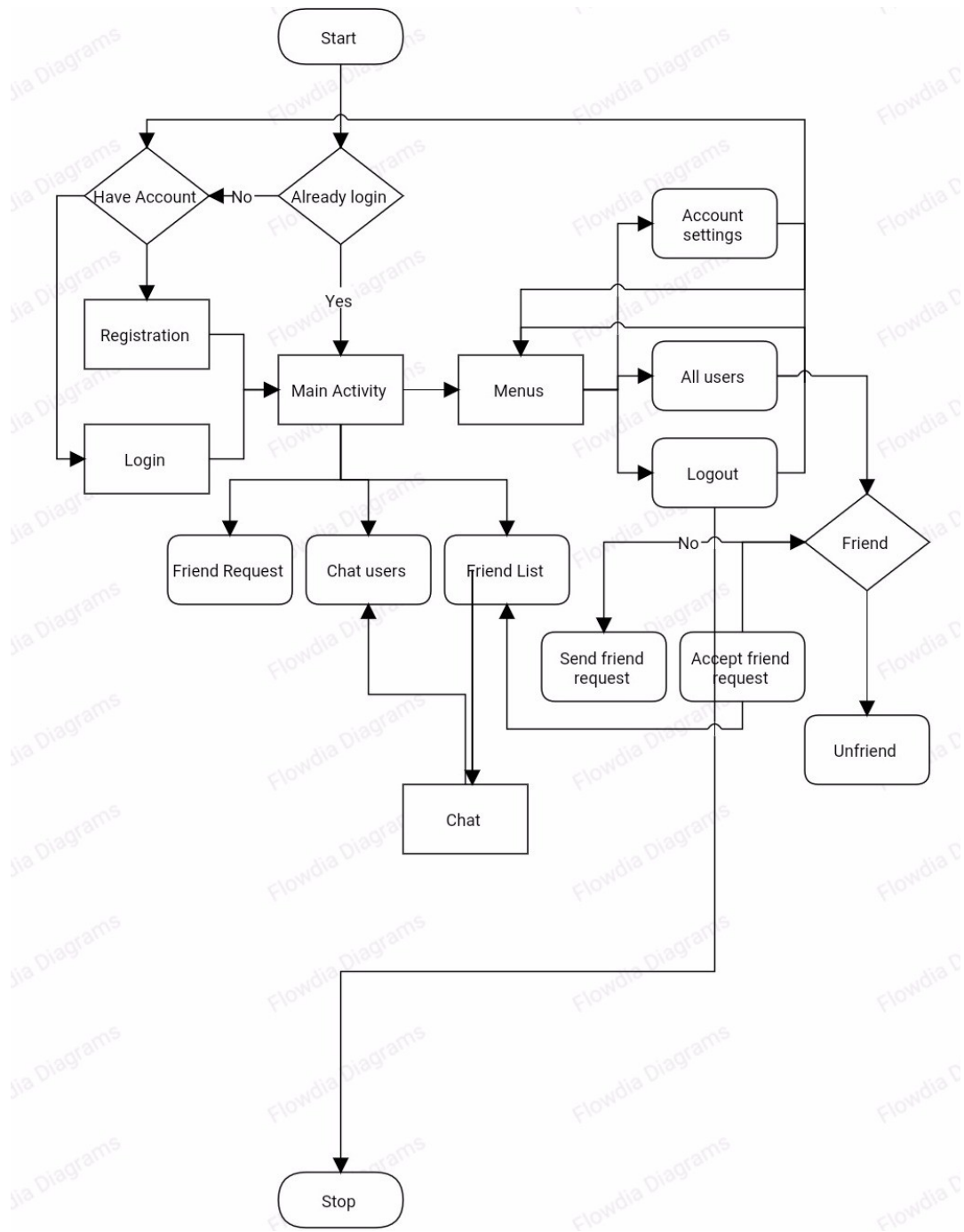
# Class Diagrams



# Sequence Diagrams



## Flowchart:-





# **System Implementation**

## **User**

User can see the other Users status and profile pitchers.

User can send the friend request also cancel friend request

Other user can see the friend request and he decide to accept or decline that friend request.

If user accept that request the he will become a friend of that other user.

User can see the friend also send the message to other friends

Other user also does the same things.

Also User can set profile pitcher and set status.

One more function is user can unfriend to other user so that

Other user can't send message.

# SYSTEM DESIGN DETAILS

## Classes

- ChatActivity
- ChatFragment
- FirebaseMessangingService
- Friends
- FriendsFragment
- GetTimeAgo
- LoginActivity
- MainActivity
- MessageAdapter
- Message
- ProfileActivity
- RegisterActivity
- RequestFragment
- RobbyChat
- SessionPagerAdapter
- SettingActivity
- StatusActivity
- Users
- UsersActivity

# Coding

## **MainActivity.java**

```
package com.jadhavrupesh22.robbychatapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ServerValue;

public class MainActivity extends AppCompatActivity {

    private FirebaseAuth mAuth;
    private Toolbar mToolbar;
    private ViewPager mViewPager;
    private SectionsPagerAdapter mSectionsPagerAdapter;
    private DatabaseReference mUserRef;
```

```

private TabLayout mTabLayout;

@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mAuth = FirebaseAuth.getInstance();
    mToolbar = (Toolbar) findViewById(R.id.main_page_toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("Lapit Chat");
    if (mAuth.getCurrentUser() != null) {
        mUserRef =
        FirebaseDatabase.getInstance().getReference().child("Users").child(mAuth.getCurrentUser().get
        Uid());
    }
    //Tabs
    mViewPager = (ViewPager) findViewById(R.id.main_tabPager);
    mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());
    mViewPager.setAdapter(mSectionsPagerAdapter);
    mTabLayout = (TabLayout) findViewById(R.id.main_tabs);
    mTabLayout.setupWithViewPager(mViewPager);
}

@Override

public void onStart() {
    super.onStart();
    // Check if user is signed in (non-null) and update UI accordingly.
    FirebaseUser currentUser = mAuth.getCurrentUser();
    if(currentUser == null){
        sendToStart();
    } else {

```

```

mUserRef.child("online").setValue("true");
}
}
@Override
protected void onStop() {
super.onStop();
FirebaseUser currentUser = mAuth.getCurrentUser();
if(currentUser != null) {
mUserRef.child("online").setValue(ServerValue.TIMESTAMP);
}
}
private void sendToStart() {
Intent startIntent = new Intent(MainActivity.this, StartActivity.class);
startActivity(startIntent);
finish();
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
super.onCreateOptionsMenu(menu);
getMenuInflater().inflate(R.menu.main_menu, menu);
return true;
}
public void Logout(MenuItem item) {
FirebaseAuth.getInstance().signOut();
sendToStart();
mUserRef.child("online").setValue(ServerValue.TIMESTAMP);
}
//setting btn

```

```
public void alu(Menuitem item) {  
    Intent userIntent=new Intent(MainActivity.this,UsersActivity.class);  
    startActivity(userIntent);  
}  
public void as(Menuitem item) {  
    Intent settingIntent=new Intent(MainActivity.this,SettingsActivity.class);  
    startActivity(settingIntent);  
}  
}
```

## **ChatActivity.java**

```
package com.jadhavrupesh22.robbychatapp;  
import android.content.Context;  
import android.content.Intent;  
import android.net.Uri;  
import android.support.annotation.NonNull;  
import android.support.v4.widget.SwipeRefreshLayout;  
import android.support.v7.app.ActionBar;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.support.v7.widget.LinearLayoutManager;  
import android.support.v7.widget.RecyclerView;  
import android.support.v7.widget.Toolbar;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;
```

```
import android.widget.TextView;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
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.Query;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import de.hdodenhof.circleimageview.CircleImageView;
public class ChatActivity extends AppCompatActivity {
    private String mChatUser;
    private Toolbar mChatToolbar;
    private DatabaseReference mRootRef;
    private TextView mTitleView;
    private TextView mLastSeenView;
    private CircleImageView mProfileImage;
```

```

private FirebaseAuth mAuth;
private String mCurrentUserId;
private ImageButton mChatAddBtn;
private ImageButton mChatSendBtn;
private EditText mChatMessageView;
private RecyclerView mMessagingList;
private SwipeRefreshLayout mRefreshLayout;
private final List<Messages> messagesList = new ArrayList<>();
private LinearLayoutManager mLayoutManager;
private MessageAdapter mAdapter;
private static final int TOTAL_ITEMS_TO_LOAD = 10;
private int mCurrentPage = 1;
private static final int GALLERY_PICK = 1;
// Storage Firebase
private StorageReference mImageStorage;
//New Solution
private int itemPos = 0;
private String mLastKey = "";
private String mPrevKey = "";
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_chat);
    mChatToolbar = (Toolbar) findViewById(R.id.chat_app_bar);
    setSupportActionBar(mChatToolbar);
    ActionBar actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setDisplayShowCustomEnabled(true);

```



```

mRootRef = FirebaseDatabase.getInstance().getReference();
mAuth = FirebaseAuth.getInstance();
mCurrentUserId = mAuth.getCurrentUser().getUid();
mChatUser = getIntent().getStringExtra("user_id");
String userName = getIntent().getStringExtra("user_name");
LayoutInflater inflater = (LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
View action_bar_view = inflater.inflate(R.layout.chat_custom_bar, null);
actionBar.setCustomView(action_bar_view);
// ---- Custom Action bar Items ----
mTitleView = (TextView) findViewById(R.id.custom_bar_title);
mLastSeenView = (TextView) findViewById(R.id.custom_bar_seen);
mProfileImage = (CircleImageView) findViewById(R.id.custom_bar_image);
mChatAddBtn = (ImageButton) findViewById(R.id.chat_add_btn);
mChatSendBtn = (ImageButton) findViewById(R.id.chat_send_btn);
mChatMessageView = (EditText) findViewById(R.id.chat_message_view);
mAdapter = new MessageAdapter(messagesList);
mMessagesList = (RecyclerView) findViewById(R.id.messages_list);
mRefreshLayout = (SwipeRefreshLayout) findViewById(R.id.message_swipe_layout);
mLinearLayout = new LinearLayoutManager(this);
mMessagesList.setHasFixedSize(true);
mMessagesList.setLayoutManager(mLinearLayout);
mMessagesList.setAdapter(mAdapter);
//----- IMAGE STORAGE -----
mImageStorage = FirebaseStorage.getInstance().getReference();
mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true);
loadMessages();
mTitleView.setText(userName);
mRootRef.child("Users").child(mChatUser).addValueEventListener(new ValueEventListener() {

```

```

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
    String online=dataSnapshot.child("online").getValue().toString();
    String image=dataSnapshot.child("image").getValue().toString();
    Picasso.get().load(image).placeholder(R.drawable.pp).into(mProfileImage);
    if (online.equals("true")){
        mLastSeenView.setText("online");
    }
    else {
        GetTimeAgo getTimeAgo = new GetTimeAgo();
        long lastTime = Long.parseLong(online);
        String lastSeenTime = getTimeAgo.getTimeAgo(lastTime, getApplicationContext());
        mLastSeenView.setText(lastSeenTime);
    }
}

```

```

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {
}

});

mRootRef.child("chat").child(mCurrentUserId).addValueEventListener(new
 ValueEventListener() {

```

```

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
    if(!dataSnapshot.hasChild(mChatUser)){
        Map chatAddMap = new HashMap();
        chatAddMap.put("seen", false);
        chatAddMap.put("timestamp", ServerValue.TIMESTAMP);
        Map chatUserMap = new HashMap();
        chatUserMap.put("Chat/" + mCurrentUserId + "/" + mChatUser, chatAddMap);
    }
}

```

```

chatUserMap.put("Chat/" + mChatUser + "/" + mCurrentUserId, chatAddMap);
mRootRef.updateChildren(chatUserMap, new DatabaseReference.CompletionListener() {
    @Override
    public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
    {
        if(databaseError != null){
            Log.d("CHAT_LOG", databaseError.getMessage().toString());
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
});
mChatSendBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sendMessage();
    }
});
mChatAddBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent galleryIntent = new Intent();
        galleryIntent.setType("image/*");
        galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(galleryIntent, "SELECT IMAGE"), GALLERY_PICK);
    }
});
mRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override

```



```

messageMap.put("time", ServerValue.TIMESTAMP);
messageMap.put("from", mCurrentUserId);
Map messageUserMap = new HashMap();
messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);
mChatMessageView.setText("");
mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {
@Override
public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
{
if(databaseError != null){
Log.d("CHAT_LOG", databaseError.getMessage().toString());
}
}});
}}
});}
}

private void loadMoreMessages() {
DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);
Query messageQuery = messageRef.orderByKey().endAt(mLastKey).limitToLast(10);
messageQuery.addChildEventListener(new ChildEventListener() {
@Override
public void onChildAdded(DataSnapshot dataSnapshot, String s) {
Messages message = dataSnapshot.getValue(Messages.class);
String messageKey = dataSnapshot.getKey();
if(!mPrevKey.equals(messageKey)){
messagesList.add(itemPos++, message);
} else {

```

```

mPrevKey = mLastKey;
}

if(itemPos == 1) {
mLastKey = messageKey;
}

Log.d("TOTALKEYS", "Last Key : " + mLastKey + " | Prev Key : " + mPrevKey + " | Message Key : "
+ messageKey);

mAdapter.notifyDataSetChanged();
mRefreshLayout.setRefreshing(false);
mLinearLayout.scrollToPositionWithOffset(10, 0);
}

@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) {
}});
}

private void loadMessages() {
DatabaseReference messageRef =
mRootRef.child("messages").child(mCurrentUserId).child(mChatUser);

Query messageQuery = messageRef.limitToLast(mCurrentPage * TOTAL_ITEMS_TO_LOAD);
messageQuery.addChildEventListener(new ChildEventListener() {

```

```

@Override

public void onChildAdded(DataSnapshot dataSnapshot, String s) {
    Messages message = dataSnapshot.getValue(Messages.class);
    itemPos++;
    if(itemPos == 1){
        String messageKey = dataSnapshot.getKey();
        mLastKey = messageKey;
        mPrevKey = messageKey;
    }
    messagesList.add(message);
    mAdapter.notifyDataSetChanged();
    mMessagesList.scrollToPosition(messagesList.size() - 1);
    mRefreshLayout.setRefreshing(false);
}

@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) {}

});

}

private void sendMessage() {
    String message = mChatMessageView.getText().toString();
    if(!TextUtils.isEmpty(message)){
        String current_user_ref = "messages/" + mCurrentUserId + "/" + mChatUser;

```

```

String chat_user_ref = "messages/" + mChatUser + "/" + mCurrentUserId;
DatabaseReference user_message_push = mRootRef.child("messages")
.child(mCurrentUserId).child(mChatUser).push();
String push_id = user_message_push.getKey();
Map messageMap = new HashMap();
messageMap.put("message", message);
messageMap.put("seen", false);
messageMap.put("type", "text");
messageMap.put("time", ServerValue.TIMESTAMP);
messageMap.put("from", mCurrentUserId);
Map messageUserMap = new HashMap();
messageUserMap.put(current_user_ref + "/" + push_id, messageMap);
messageUserMap.put(chat_user_ref + "/" + push_id, messageMap);
mChatMessageView.setText("");
mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("seen").setValue(true);
mRootRef.child("Chat").child(mCurrentUserId).child(mChatUser).child("timestamp").setValue(S
erverValue.TIMESTAMP);
mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("seen").setValue(false);
mRootRef.child("Chat").child(mChatUser).child(mCurrentUserId).child("timestamp").setValue(S
erverValue.TIMESTAMP);
mRootRef.updateChildren(messageUserMap, new DatabaseReference.CompletionListener() {
@Override
public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
{
if(databaseError != null){
Log.d("CHAT_LOG", databaseError.getMessage().toString());
}
}}
});}}}

```



## ChatFragment.java

```
package com.jadhavrupesh22.robbychatapp;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.firebase.auth.FirebaseAuth;
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.Query;
import com.google.firebase.database.ValueEventListener;
```

```

import com.squareup.picasso.Picasso;
import de.hdodenhof.circleimageview.CircleImageView;

/**
 * A simple {@link Fragment} subclass.
 */
public class ChatsFragment extends Fragment {
    private RecyclerView mFriendsList;
    private DatabaseReference mFriendsDatabase;
    private DatabaseReference mFriendsDatabases;
    private DatabaseReference mUsersDatabase;
    private DatabaseReference mLastMsg;
    private FirebaseAuth mAuth;
    private String mCurrent_user_id;
    private View mMainView;
    private DatabaseReference mUserDatabase;
    public ChatsFragment() {
        // Required empty public constructor
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        mMainView = inflater.inflate(R.layout.fragment_friends, container, false);
        mFriendsList = (RecyclerView) mMainView.findViewById(R.id.friends_list);
        mAuth = FirebaseAuth.getInstance();
        mCurrent_user_id = mAuth.getCurrentUser().getUid();
        mFriendsDatabases =
            FirebaseDatabase.getInstance().getReference().child("Friends").child(mCurrent_user_id);
        mFriendsDatabases.keepSynced(true);
    }

```

```

mFriendsDatabase =
FirebaseDatabase.getInstance().getReference().child("Chat").child(mCurrent_user_id);

mFriendsDatabase.keepSynced(true);

mUsersDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
mUsersDatabase.keepSynced(true);

mLastMsg =
FirebaseDatabase.getInstance().getReference().child("messages").child(mCurrent_user_id);
mLastMsg.keepSynced(true);

//mDatabase=FirebaseDatabase.getInstance();

mFriendsList.setLayoutManager(new LinearLayoutManager(getContext()));

// Inflate the layout for this fragment
return mMainView;
}

@Override

public void onStart() {
super.onStart();

FirebaseRecyclerOptions<Friends> options =
new FirebaseRecyclerOptions.Builder<Friends>()

.setQuery(mFriendsDatabase, Friends.class)

.build();

FirebaseRecyclerAdapter<Friends, FriendsFragment.FriendsViewHolder> adapter= new
FirebaseRecyclerAdapter<Friends, FriendsFragment.FriendsViewHolder>(options){

@NonNull

@Override

public FriendsFragment.FriendsViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.users_single_layout,
parent, false);

FriendsFragment.FriendsViewHolder friendsViewHolder=new
FriendsFragment.FriendsViewHolder(view);

return friendsViewHolder;

```

```

}

@Override

protected void onBindViewHolder(@NonNull final FriendsFragment.FriendsViewHolder
friendsViewHolder, int i, @NonNull Friends friends) {

friendsViewHolder.setDate(friends.getDate());

final String list_user_id = getRef(i).getKey();

mUsersDatabase.child(list_user_id).addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

final String userName = dataSnapshot.child("name").getValue().toString();

String userThumb = dataSnapshot.child("thumb_image").getValue().toString();

if (dataSnapshot.hasChild("online")){

String userOnline= dataSnapshot.child("online").getValue().toString();

friendsViewHolder.setUserOnline(userOnline);

}

friendsViewHolder.setName(userName);

friendsViewHolder.setUserImage(userThumb,getContext());

friendsViewHolder.mView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent chatIntent = new Intent(getContext(), ChatActivity.class);

chatIntent.putExtra("user_id", list_user_id);

chatIntent.putExtra("user_name", userName);

startActivity(chatIntent);

}

});

}

@Override

public void onCancelled(@NonNull DatabaseError databaseError) {

```

```

}
}};}};
mFriendsList.setAdapter(adapter);
adapter.startListening();}

public static class ConvViewHolder extends RecyclerView.ViewHolder {
View mView;

public ConvViewHolder(View itemView) {
super(itemView);
mView = itemView;}

public void setDate(String date){
TextView userStatusView = (TextView) mView.findViewById(R.id.user_single_status);
userStatusView.setText(date);}

public void setName(String name){
TextView userNameView = (TextView) mView.findViewById(R.id.user_single_name);
userNameView.setText(name);}

public void setUserImage(String thumb_image, Context ctx){
CircleImageView userImageView = (CircleImageView)
mView.findViewById(R.id.user_single_image);

Picasso.get().load(thumb_image).into(userImageView);}

public void setUserOnline(String online_status){
ImageView userOnlineView=(ImageView)mView.findViewById(R.id.user_single_online_icon);
if (online_status.equals("true")){
userOnlineView.setVisibility(View.VISIBLE);
}
else{
userOnlineView.setVisibility(View.INVISIBLE);
}}}}

```

## Conv.java

```
package com.jadhavrupesh22.robbychatapp;

public class Conv {

    public boolean seen;

    public long timestamp;

    public Conv(){

    }

    public boolean isSeen() {

        return seen;}

    public void setSeen(boolean seen) {

        this.seen = seen}

    public long getTimestamp() {

        return timestamp;}

    public void setTimestamp(long timestamp) {

        this.timestamp = timestamp;

    }

    public Conv(boolean seen, long timestamp) {

        this.seen = seen;

        this.timestamp = timestamp;

    }}
}
```

## FirebaseMessageService.java

```
package com.jadhavrupesh22.robbychatapp;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import com.google.firebase.messaging.RemoteMessage;

public class FirebaseMessagingService extends
com.google.firebase.messaging.FirebaseMessagingService {

    @Override

    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);

        String notification_title = remoteMessage.getNotification().getTitle();
        String notification_message = remoteMessage.getNotification().getBody();
        String click_action = remoteMessage.getNotification().getClickAction();
        String from_user_id = remoteMessage.getData().get("from_user_id");

        NotificationCompat.Builder mBuilder =new
        NotificationCompat.Builder(this).setSmallIcon(R.mipmap.ic_launcher).setContentTitle("Fried
        Request").setContentText("New Friend Request..");

        Intent resultIntent = new Intent(click_action);
        resultIntent.putExtra("user_id", from_user_id);

        PendingIntent resultPendingIntent =
        PendingIntent.getActivity(this,0,resultIntent,PendingIntent.FLAG_UPDATE_CURRENT);
        mBuilder.setContentIntent(resultPendingIntent);

        int mNotificationId = (int) System.currentTimeMillis();

        NotificationManager
        mNotifyMgr=(NotificationManager) getSystemService(NOTIFICATION_SERVICE);
        mNotifyMgr.notify(mNotificationId, mBuilder.build());
    }
}
```

```
}}
```

## FriendReq.java

```
package com.jadhavrupesh22.robbychatapp;

public class Friend_req {

    public String request_type;

    public Friend_req(String request_type) {

        this.request_type = request_type;}

    public String getRequest_type() {

        return request_type;}

    public void setRequest_type(String request_type) {

        this.request_type = request_type;

    }

}
```

## Friends.java

```
package com.jadhavrupesh22.robbychatapp;

public class Friends {

    public String date;

    public Friends() {}

    public Friends(String date) {

        this.date = date;}

    public String getDate() {

        return date;}

    public void setDate(String date) {

        this.date = date;

    }

}
```



## GetTimeAgo.java

```
package com.jadhavrupesh22.robbychatapp;

import android.content.Context;

public class GetTimeAgo {

    private static final int SECOND_MILLIS = 1000;

    private static final int MINUTE_MILLIS = 60 * SECOND_MILLIS;

    private static final int HOUR_MILLIS = 60 * MINUTE_MILLIS;

    private static final int DAY_MILLIS = 24 * HOUR_MILLIS;

    public static String getTimeAgo(long time, Context ctx) {

        if (time < 10000000000000L) {

            // if timestamp given in seconds, convert to millis

            time *= 1000;}

        long now = System.currentTimeMillis();

        if (time > now || time <= 0) {

            return null;}

        // TODO: localize

        final long diff = now - time;

        if (diff < MINUTE_MILLIS) {

            return "just now";

        } else if (diff < 2 * MINUTE_MILLIS) {

            return "a minute ago";

        } else if (diff < 50 * MINUTE_MILLIS) {

            return diff / MINUTE_MILLIS + " minutes ago";

        } else if (diff < 90 * MINUTE_MILLIS) {

            return "an hour ago";

        } else if (diff < 24 * HOUR_MILLIS) {

            return diff / HOUR_MILLIS + " hours ago";
```

```
} else if (diff < 48 * HOUR_MILLIS) {  
    return "yesterday";  
} else {  
    return diff / DAY_MILLIS + " days ago";}}}
```

## LoginActivity.java

```
package com.jadhavrupesh22.robbychatapp;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.support.annotation.NonNull;  
import android.support.design.widget.TextInputLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.iid.FirebaseInstanceId;  
  
public class LoginActivity extends AppCompatActivity {  
    private TextInputLayout mLoginEmail;  
    private TextInputLayout mLoginPassword;
```

```

//Firebase
private FirebaseAuth mAuth;

//Toolbar
private android.support.v7.widget.Toolbar mToolbar;

//progressBar
private ProgressDialog mLoginProgress;
private DatabaseReference mUserDatabase;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    //Toolbar
    mToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.login_toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("Login");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    //Firebase
    mAuth = FirebaseAuth.getInstance();
    mUserDatabase=FirebaseDatabase.getInstance().getReference().child("Users");

    //Progress
    mLoginProgress = new ProgressDialog(this);
    mLoginEmail = (TextInputLayout) findViewById(R.id.login_email);
    mLoginPassword = (TextInputLayout) findViewById(R.id.login_password);}

    public void lb(View view) {
        String email = mLoginEmail.getEditText().getText().toString();
        String password = mLoginPassword.getEditText().getText().toString();
        if (!TextUtils.isEmpty(email) || !TextUtils.isEmpty(password)) {
            mLoginProgress.setTitle("Loggin In");

```

```

mLoginProgress.setMessage("Please Waite while Login...");
mLoginProgress.setCanceledOnTouchOutside(false);
mLoginProgress.show();
loginUser(email, password);}

else{

Toast.makeText(LoginActivity.this,"Please Enter Email and
Password.....",Toast.LENGTH_LONG).show();}}

private void loginUser(String email, String password) {

 mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(new
 OnCompleteListener<AuthResult>() {

@Override

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

if (task.isSuccessful()) {

mLoginProgress.dismiss();

String mCurrentUserId= mAuth.getCurrentUser().getUid();

String deviceToken=FirebaseInstanceId.getInstance().getToken();

mUserDatabase.child(mCurrentUserId).child("device_token").setValue(deviceToken).addOnSuc
cessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

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

mainIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);

startActivity(mainIntent);

finish();

}});} else {

mLoginProgress.hide();

Toast.makeText(LoginActivity.this, "Invalid ID or password. Please try again.",
Toast.LENGTH_LONG).show();}

}});}}}

```

## MessageAdapter .java

```
package com.jadhavrupesh22.robbychatapp;

import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
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.squareup.picasso.Picasso;
import java.util.List;
import de.hdodenhof.circleimageview.CircleImageView;

public class MessageAdapter extends
RecyclerView.Adapter<MessageAdapter.MessageViewHolder>{

    private List<Messages> mMessageList;

    private DatabaseReference mUserDatabase;

    public MessageAdapter(List<Messages> mMessageList) {

        this.mMessageList = mMessageList;
    }

    @Override

    public MessageViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

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

        .inflate(R.layout.message_single_layout ,parent, false);
```

```

return new MessageViewHolder(v);
}

public class MessageViewHolder extends RecyclerView.ViewHolder {
    public TextView messageText;
    public CircleImageView profileImage;
    public TextView displayName;
    public ImageView messageImage;
    public TextView Time;
    public MessageViewHolder(View view) {
        super(view);
        messageText = (TextView) view.findViewById(R.id.message_text_layout);
        profileImage = (CircleImageView) view.findViewById(R.id.message_profile_layout);
        displayName = (TextView) view.findViewById(R.id.name_text_layout);
        messageImage = (ImageView) view.findViewById(R.id.message_image_layout);}

    @Override
    public void onBindViewHolder(final MessageViewHolder viewHolder, int i) {
        Messages c = mMessageList.get(i);
        String from_user = c.getFrom();
        String message_type = c.getType();
        long time=c.getTime();

        mUserDatabase =
        FirebaseDatabase.getInstance().getReference().child("Users").child(from_user);
        mUserDatabase.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String name = dataSnapshot.child("name").getValue().toString();
                String image = dataSnapshot.child("thumb_image").getValue().toString();
                viewHolder.displayName.setText(name);
                Picasso.get().load(image).placeholder(R.drawable.pp).into(viewHolder.profileImage);}

```

```

@Override

public void onCancelled(DatabaseError databaseError) {}

if(message_type.equals("text")) {
viewHolder.messageText.setText(c.getMessage());
viewHolder.messageImage.setVisibility(View.INVISIBLE);}
else {
viewHolder.messageText.setVisibility(View.INVISIBLE);
Picasso.get().load(c.getMessage()).placeholder(R.drawable.pp).into(viewHolder.messageImage)
;}}

@Override

public int getItemCount() {
return mMMessageList.size();}

}

```

## MessageAdapter .java

```

package com.jadhavrupesh22.robbychatapp;

public class Messages {

private String message, type;

private long time;

private boolean seen;

private String from;

public Messages(String from) {
this.from = from;}

public String getFrom() {
return from;}

public void setFrom(String from) {
this.from = from;}

public Messages(String message, String type, long time, boolean seen) {

```

```
this.message = message;
this.type = type;
this.time = time;
this.seen = seen;
}
public String getMessage() {
return message;}
public void setMessage(String message) {
this.message = message;}
public String getType() {
return type;}
public void setType(String type) {
this.type = type;}
public long getTime() {
return time;}
public void setTime(long time) {
this.time = time;
}
public boolean isSeen() {
return seen;
}
public void setSeen(boolean seen) {
this.seen = seen;
}
public Messages(){
}
}
```



## **ProfileActivity.jav**

```
package com.jadhavrupesh22.robbychatapp;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Callback;
import com.squareup.picasso.NetworkPolicy;
import com.squareup.picasso.Picasso;
```

```
import java.text.DateFormat;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
public class ProfileActivity extends AppCompatActivity {
    private ImageView mProfileImage;
    private Button mProfileSendReqBtn, mDeclineBtn;
    private TextView mProfileName, mProfileStatus, mProfileFriendCount;
    private DatabaseReference mUserDatabase;
    private DatabaseReference mFriendDatabase;
    private DatabaseReference mFriendReqDatabase;
    //progress dialog
    private ProgressDialog mProgressDialog;
    private String mCurrent_state;
    private FirebaseUser mCurrent_user;
    private FirebaseAuth mAuth;
    private DatabaseReference mNotificationDatabase;
    private DatabaseReference mRootRef;
    private DatabaseReference mUserRef;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
        final String user_Id = getIntent().getStringExtra("user_Id");
        mProgressDialog = new ProgressDialog(this);
        mProgressDialog.setTitle("Loading User Data");
        mProgressDialog.setMessage("Please wait while loading user Data");
        mProgressDialog.setCanceledOnTouchOutside(false);
```

```

mProgressDialog.show();

mAuth = FirebaseAuth.getInstance();

mUserRef=FirebaseDatabase.getInstance().getReference().child("Users").child(mAuth.getCurrentUser().getUid());

mFriendReqDatabase = FirebaseDatabase.getInstance().getReference().child("Friend_req");
mFriendDatabase = FirebaseDatabase.getInstance().getReference().child("Friends");
mUserDatabase = FirebaseDatabase.getInstance().getReference().child("Users").child(user_Id);
mRootRef = FirebaseDatabase.getInstance().getReference();
mNotificationDatabase = FirebaseDatabase.getInstance().getReference().child("notifications");
mCurrent_user = FirebaseAuth.getInstance().getCurrentUser();
mCurrent_state = "not_friends";
mProfileImage = (ImageView) findViewById(R.id.profile_image);
mProfileSendReqBtn = (Button) findViewById(R.id.profile_send_req_btn);
mDeclineBtn = (Button) findViewById(R.id.profile_decline_Friend_requenst);
mProfileName = (TextView) findViewById(R.id.profile_displayName);
mProfileStatus = (TextView) findViewById(R.id.profile_status);
mProfileFriendCount = (TextView) findViewById(R.id.profile_totalFriends);
mDeclineBtn.setVisibility(View.INVISIBLE);
mDeclineBtn.setEnabled(false);
mUserDatabase.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        String displayName = dataSnapshot.child("name").getValue().toString();
        String status = dataSnapshot.child("status").getValue().toString();
        final String image = dataSnapshot.child("image").getValue().toString();
        mProfileName.setText(displayName);
        mProfileStatus.setText(status);
        Picasso.get().load(image).placeholder(R.drawable.pp).into(mProfileImage);
    }
});
//---Friend list // request Feature-----//

```

```
mFriendReqDatabase.child(mCurrent_user.getId()).addListenerForSingleValueEvent(new ValueEventListener() {
```

```
@Override
```

```
public void onDataChange(DataSnapshot dataSnapshot) {
```

```
if(dataSnapshot.hasChild(user_Id)){
```

```
String req_type = dataSnapshot.child(user_Id).child("request_type").getValue().toString();
```

```
if(req_type.equals("received")){
```

```
mCurrent_state = "req_received";
```

```
mProfileSendReqBtn.setText("Accept Friend Request");
```

```
mDeclineBtn.setVisibility(View.VISIBLE);
```

```
mDeclineBtn.setEnabled(true);
```

```
} else if(req_type.equals("sent")) {
```

```
mCurrent_state = "req_sent";
```

```
mProfileSendReqBtn.setText("Cancel Friend Request");
```

```
mDeclineBtn.setVisibility(View.INVISIBLE);
```

```
mDeclineBtn.setEnabled(false);
```

```
}
```

```
mProgressDialog.dismiss();
```

```
} else {
```

```
mFriendDatabase.child(mCurrent_user.getId()).addListenerForSingleValueEvent(new ValueEventListener() {
```

```
@Override
```

```
public void onDataChange(DataSnapshot dataSnapshot) {
```

```
if(dataSnapshot.hasChild(user_Id)){
```

```
mCurrent_state = "friends";
```

```
mProfileSendReqBtn.setText("Unfriend this Person");
```

```
mDeclineBtn.setVisibility(View.INVISIBLE);
```

```
mDeclineBtn.setEnabled(false);
```

```
}
```

```

mProgressDialog.dismiss();
}

@Override
public void onCancelled(DatabaseError databaseError) {
mProgressDialog.dismiss();
}});
}

@Override
public void onCancelled(DatabaseError databaseError) {});
//-----Friend list // request Feature-----
}

@Override
public void onCancelled(@NonNull DatabaseError databaseError) {});
mProfileSendReqBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
mProfileSendReqBtn.setEnabled(false);
//-----Not Friend State----//
if (mCurrent_state.equals("not_friends")) {
DatabaseReference newNotificationref = mRootRef.child("notifications").child(user_Id).push();
String newNotificationId = newNotificationref.getKey();
HashMap<String, String> notificationData = new HashMap<>();
notificationData.put("from", mCurrent_user.getUid());
notificationData.put("type", "request");
Map requestMap = new HashMap();
requestMap.put("Friend_req/" + mCurrent_user.getUid() + "/" + user_Id + "/request_type",
"sent");
requestMap.put("Friend_req/" + user_Id + "/" + mCurrent_user.getUid() + "/request_type",
"received");
}
}
}

```

```

requestMap.put("notifications/" + user_Id + "/" + newNotificationId, notificationData);
mRootRef.updateChildren(requestMap, new DatabaseReference.CompletionListener() {

@Override

public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
{

if (databaseError != null) {

Toast.makeText(ProfileActivity.this, "There was some error in sending request",
Toast.LENGTH_SHORT).show();

} else {

mCurrent_state = "req_sent";
mProfileSendReqBtn.setText("Cancel Friend Request");

}

mProfileSendReqBtn.setEnabled(true);

}});

//-----candle Friend request----//

if (mCurrent_state.equals("req_sent")) {

mFriendReqDatabase.child(mCurrent_user.getId()).child(user_Id).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

mFriendReqDatabase.child(user_Id).child(mCurrent_user.getId()).removeValue().addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

mProfileSendReqBtn.setEnabled(true);

mCurrent_state = "not_friends";

mProfileSendReqBtn.setText("Send Friend Request");

mDeclineBtn.setVisibility(View.INVISIBLE);

mDeclineBtn.setEnabled(false);

}

}

}

}

}

```

```

});
Toast.makeText(ProfileActivity.this, "Request is Canceled.", Toast.LENGTH_LONG).show();
});
}
//-----request receive state-----//
if (mCurrent_state.equals("req_received")) {
    final String currentDate = DateFormat.getDateInstance().format(new Date());
    Map friendsMap = new HashMap();
    friendsMap.put("Friends/" + mCurrent_user.getId() + "/" + user_Id + "/date", currentDate);
    friendsMap.put("Friends/" + user_Id + "/" + mCurrent_user.getId() + "/date", currentDate);
    friendsMap.put("Friend_req/" + mCurrent_user.getId() + "/" + user_Id, null);
    friendsMap.put("Friend_req/" + user_Id + "/" + mCurrent_user.getId(), null);
    mRootRef.updateChildren(friendsMap, new DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
        {
            if (databaseError == null) {
                mProfileSendReqBtn.setEnabled(true);
                mCurrent_state = "friends";
                mProfileSendReqBtn.setText("Unfriend this Person");
                mDeclineBtn.setVisibility(View.INVISIBLE);
                mDeclineBtn.setEnabled(false);
            } else {
                String error = databaseError.getMessage();
                Toast.makeText(ProfileActivity.this, error, Toast.LENGTH_SHORT).show();
            }
        }
    });
}
// ----- UNFRIENDS -----
if(mCurrent_state.equals("friends")){

```

```

Map unfriendMap = new HashMap();
unfriendMap.put("Friends/" + mCurrent_user.getUid() + "/" + user_Id, null);
unfriendMap.put("Friends/" + user_Id + "/" + mCurrent_user.getUid(), null);
unfriendMap.put("Chat/" + mCurrent_user.getUid() + "/" + user_Id, null);
unfriendMap.put("Chat/" + user_Id + "/" + mCurrent_user.getUid(), null);
mRootRef.updateChildren(unfriendMap, new DatabaseReference.CompletionListener() {
@Override
public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
{
if(databaseError == null){
mCurrent_state = "not_friends";
mProfileSendReqBtn.setText("Send Friend Request");
mDeclineBtn.setVisibility(View.INVISIBLE);
mDeclineBtn.setEnabled(false);
} else {
String error = databaseError.getMessage();
Toast.makeText(ProfileActivity.this, error, Toast.LENGTH_SHORT).show();}
mProfileSendReqBtn.setEnabled(true);}}});
}
});
mDeclineBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
// ----- Decline_FRIENDS -----
if(mCurrent_state.equals("req_received")){
Map declinefriendMap = new HashMap();
declinefriendMap.put("Friend_req/" + mCurrent_user.getUid() + "/" + user_Id, null);
declinefriendMap.put("Friend_req/" + user_Id + "/" + mCurrent_user.getUid(), null);
mRootRef.updateChildren(declinefriendMap, new DatabaseReference.CompletionListener() {

```



```
@Override

public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference)
{
    if(databaseError == null){
        mCurrent_state = "not_friends";
        mProfileSendReqBtn.setText("Send Friend Request");
        mDeclineBtn.setVisibility(View.INVISIBLE);
        mDeclineBtn.setEnabled(false);
    } else {
        String error = databaseError.getMessage();
        Toast.makeText(ProfileActivity.this, error, Toast.LENGTH_SHORT).show();
    }
    mProfileSendReqBtn.setEnabled(true);
});

}

}

}

);

}

}
```

## RegisterActivity.java

```
package com.jadhavrupesh22.robbychatapp;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;
import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity {

    //Firebase
    private FirebaseAuth mAuth;

    //Toolbar
    private android.support.v7.widget.Toolbar mToolbar;
    private TextInputLayout mDisplayName;
```

```
private TextInputLayout mEmail;
private TextInputLayout mPassword;
//progressBar
private ProgressDialog mRegProgress;
//Firebase Database
private DatabaseReference mDatabase;
private DatabaseReference mUserDatabase;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    //Toolbar
    mToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.register_toolbar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("Registration");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    //Firebase
    mAuth = FirebaseAuth.getInstance();
    mUserDatabase = FirebaseDatabase.getInstance().getReference().child("Users");
    mDisplayName = (TextInputLayout) findViewById(R.id.reg_display_name);
    mEmail = (TextInputLayout) findViewById(R.id.login_email);
    mPassword = (TextInputLayout) findViewById(R.id.login_password);
    //Progress
    mRegProgress = new ProgressDialog(this);
}
public void rcb(View view) {
    String display_name = mDisplayName.getEditText().getText().toString();
    String email = mEmail.getEditText().getText().toString();
```

```

String password = mPassword.getEditText().getText().toString();
if (!TextUtils.isEmpty(display_name) || !TextUtils.isEmpty(email) ||
!TextUtils.isEmpty(password)) {
mRegProgress.setTitle("Registration.");
mRegProgress.setMessage("Please Wait while processing.....");
mRegProgress.setCanceledOnTouchOutside(false);
mRegProgress.show();
register_account(display_name, email, password);
} else {
Toast.makeText(RegisterActivity.this, "Please Fill All Details.... ", Toast.LENGTH_LONG).show();
}
}

private void register_account(final String display_name, String email, String password) {
mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {
FirebaseUser current_user = FirebaseAuth.getInstance().getCurrentUser();
String uid = current_user.getId();
mDatabase = FirebaseDatabase.getInstance().getReference().child("Users").child(uid);
HashMap<String, String> userMap = new HashMap<>();
userMap.put("name", display_name);
userMap.put("status", "hii there, i m Using Robby chat");
userMap.put("image", "default");
userMap.put("thumb_image", "default");
mDatabase.setValue(userMap).addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {

```

```

if (task.isSuccessful()) {
    String current_user_id = mAuth.getCurrentUser().getUid();
    String deviceToken = FirebaseInstanceId.getInstance().getToken();
    mUserDatabase.child(current_user_id).child("device_token").setValue(deviceToken).addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            mRegProgress.dismiss();
            Intent mainIntent = new Intent(RegisterActivity.this, MainActivity.class);
            startActivity(mainIntent);
            finish();
        }
    });
} else {
    mRegProgress.hide();
    Toast.makeText(RegisterActivity.this, "Cannot Sign In. Please Check the form and try again.. ",
        Toast.LENGTH_LONG).show();
}
}
});
}
}
}

```

## SettingsActivity.java

```
package com.jadhavrupesh22.robbychatapp;

import android.app.ProgressDialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.net.Uri;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Callback;
import com.squareup.picasso.NetworkPolicy;
```

```
import com.squareup.picasso.Picasso;
import com.theartofdev.edmodo.cropper.CropImage;
import com.theartofdev.edmodo.cropper.CropImageView;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.util.Set;
import de.hdodenhof.circleimageview.CircleImageView;
import id.zelory.compressor.Compressor;
public class SettingsActivity extends AppCompatActivity {
    FirebaseStorage storage = FirebaseStorage.getInstance();
    DataSnapshot dataSnapshot;
    private DatabaseReference mUserDatabase;
    private FirebaseUser mCurrentUser;
    private StorageReference mStorageRef;
    private static final int GALLERY_PICK = 1;
    private android.support.v7.widget.Toolbar mToolbar;
    //Android Layout
    private CircleImageView mDisplayImage;
    private TextView mName;
    private TextView mStatus;
    //Firebase
    private FirebaseAuth mAuth;
    private DatabaseReference mUserRef;
    //Store Image in FirebaseDatabase
    private StorageReference mImageStorage;
    //progress dialog
    private ProgressDialog mProgressDialog;
```

```

private StorageReference thumb_imageRef;

Bitmap thumb_bitmap=null;

@Override

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_settings);
    mDisplayImage = (CircleImageView) findViewById(R.id.settings_image);
    mName = (TextView) findViewById(R.id.settings_name);
    mStatus = (TextView) findViewById(R.id.settings_status);
    mImageStorage = FirebaseStorage.getInstance().getReference();
    thumb_imageRef = FirebaseStorage.getInstance().getReference().child("thumb_images");
    //Firebase

    mAuth = FirebaseAuth.getInstance();

    mUserRef=FirebaseDatabase.getInstance().getReference().child("Users").child(mAuth.getCurrentUser().getUid());

    //Toolbar

    mToolbar = (android.support.v7.widget.Toolbar) findViewById(R.id.setting_app_bar);
    setSupportActionBar(mToolbar);
    getSupportActionBar().setTitle("Setting");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    mCurrentUser = FirebaseAuth.getInstance().getCurrentUser();
    String current_uid = mCurrentUser.getUid();

    mUserDatabase =
    FirebaseDatabase.getInstance().getReference().child("Users").child(current_uid);
    mUserDatabase.keepSynced(true);

    mStorageRef = FirebaseStorage.getInstance().getReference();

    mUserDatabase.addValueEventListener(new ValueEventListener() {

        @Override

        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

```



```

//retrive value from database
String name = dataSnapshot.child("name").getValue().toString();
final String image = dataSnapshot.child("image").getValue().toString();
String status = dataSnapshot.child("status").getValue().toString();
String thumb_image = dataSnapshot.child("thumb_image").getValue().toString();
mName.setText(name);
mStatus.setText(status);
if (!image.equals("default")) {
//          Picasso.get().load(image).placeholder(R.drawable.pp).into(mDisplayImage);
Picasso.get().load(image).networkPolicy(NetworkPolicy.OFFLINE)
.placeholder(R.drawable.pp).into(mDisplayImage, new Callback() {
@Override
public void onSuccess() {
}
@Override
public void onError(Exception e) {
Picasso.get().load(image).placeholder(R.drawable.pp).into(mDisplayImage);
}
});
}
@Override
public void onCancelled(@NonNull DatabaseError databaseError) {
}
});
}
//Move To Status Activity and set Status
public void cs(View view) {

```

```

String status_value = mStatus.getText().toString();
Intent statusIntent = new Intent(SettingsActivity.this, StatusActivity.class);
statusIntent.putExtra("status_value", status_value);
startActivity(statusIntent);
finish();
}

//Change Profile Image
public void ci(View view) {
    CropImage.activity()
        .setAspectRatio(1, 1)
        .setCropShape(CropImageView.CropShape.OVAL)
        .setBorderLineColor(Color.rgb(57, 76, 168))
        .setGuidelinesColor(Color.WHITE)
        .setGuidelines(CropImageView.Guidelines.ON)
        .start(this);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);
        if (resultCode == RESULT_OK) {
            mProgressDialog = new ProgressDialog(SettingsActivity.this);
            mProgressDialog.setTitle("Uploading Image");
            mProgressDialog.setMessage("Please wait while uploading");
            mProgressDialog.setCanceledOnTouchOutside(false);
            mProgressDialog.show();
            final String current_user_id = mCurrentUser.getId();

```

```

//image Uri
final Uri resultUri = result.getUri();

//Thumb_File
final File thumb_filePath = new File(resultUri.getPath());

try {
thumb_bitmap = new Compressor(this)
.setMaxWidth(200)
.setMaxHeight(200)
.setQuality(75)
.compressToBitmap(thumb_filePath);
} catch (IOException e) {
e.printStackTrace();
}

ByteArrayOutputStream baos = new ByteArrayOutputStream();
thumb_bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
final byte[] thumb_byte = baos.toByteArray();

//Thumb_File
final StorageReference filepath = mImageStorage.child("profile_images").child(current_user_id
+ ".jpg");

final StorageReference thumb_filepath =
mImageStorage.child("profile_images").child("thumbs").child(current_user_id + ".jpg");

filepath.putFile(resultUri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {

@Override

public void onComplete(@NonNull final Task<UploadTask.TaskSnapshot> task) {
if (task.isSuccessful()) {
filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {

@Override

public void onSuccess(Uri uri) {
final String download_url = uri.toString();

```

```

mUserDatabase.child("image").setValue(download_url);

thumb_filepath.putBytes(thumb_byte).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {

@Override

public void onComplete(@NonNull Task<UploadTask.TaskSnapshot> thumb_task) {

if (thumb_task.isSuccessful()) {

thumb_filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {

@Override

public void onSuccess(Uri uri) {

mProgressDialog.dismiss();

final String thumb_download_url = uri.toString();

mUserDatabase.child("thumb_image").setValue(thumb_download_url);

Toast.makeText(SettingsActivity.this, "Image added Successfully....",
Toast.LENGTH_SHORT).show();

}

}}}});

});

} else {

Toast.makeText(SettingsActivity.this, "Error in uploading", Toast.LENGTH_LONG).show();

}

});

mDisplayImage.setImageURI(resultUri);

//image Url Close

} else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {

Exception error = result.getError();

Toast.makeText(SettingsActivity.this, "" + error, Toast.LENGTH_LONG).show();

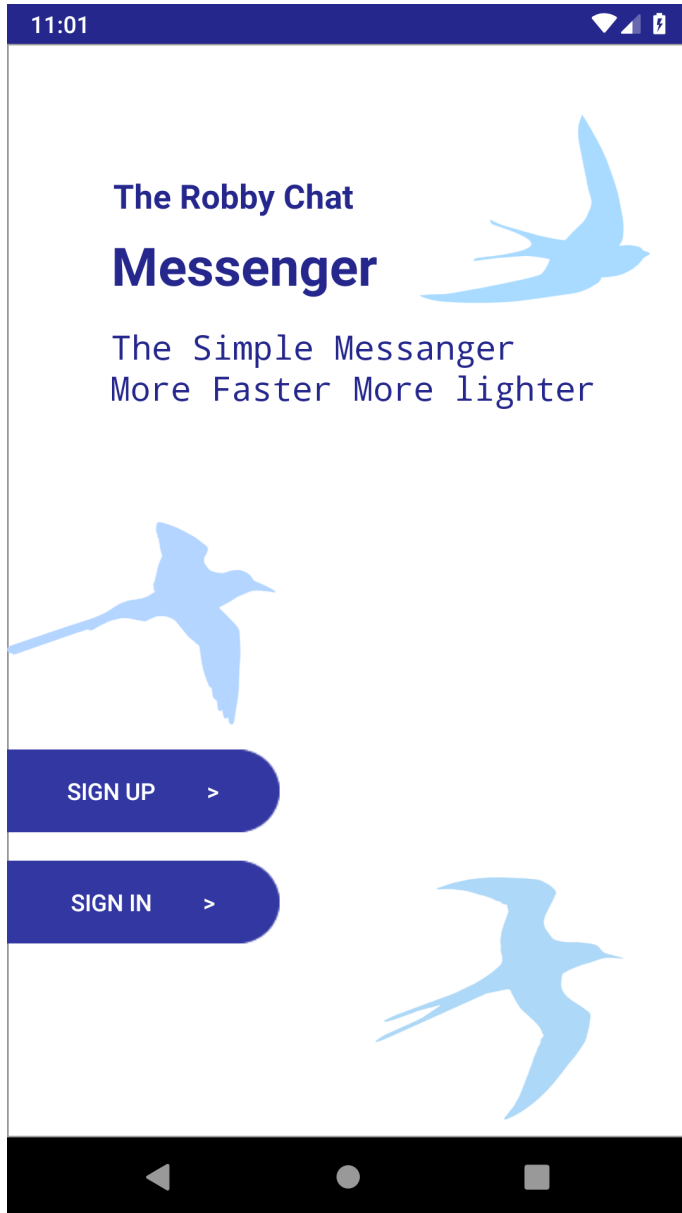
}}

}}

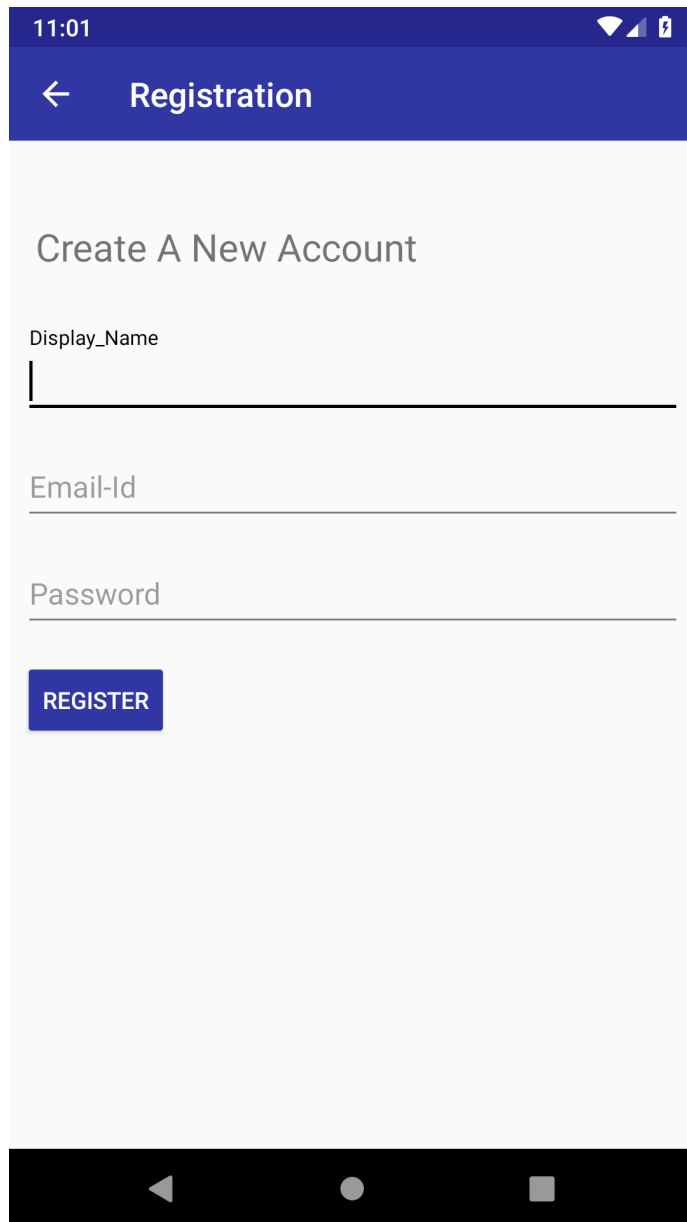
```

# RESULTS

## Starting page



## Registration page.



A mobile application registration screen. At the top is a dark blue header bar with the time '11:01' on the left and signal, Wi-Fi, and battery icons on the right. Below the header is a blue bar with a white back arrow and the text 'Registration'. The main area is light gray and contains the text 'Create A New Account'. Below this are three input fields: 'Display\_Name', 'Email-Id', and 'Password', each with a vertical line on the left and a horizontal line below. A blue 'REGISTER' button is positioned below the 'Password' field. At the bottom is a black navigation bar with three white icons: a triangle, a circle, and a square.

11:01

← Registration

Create A New Account

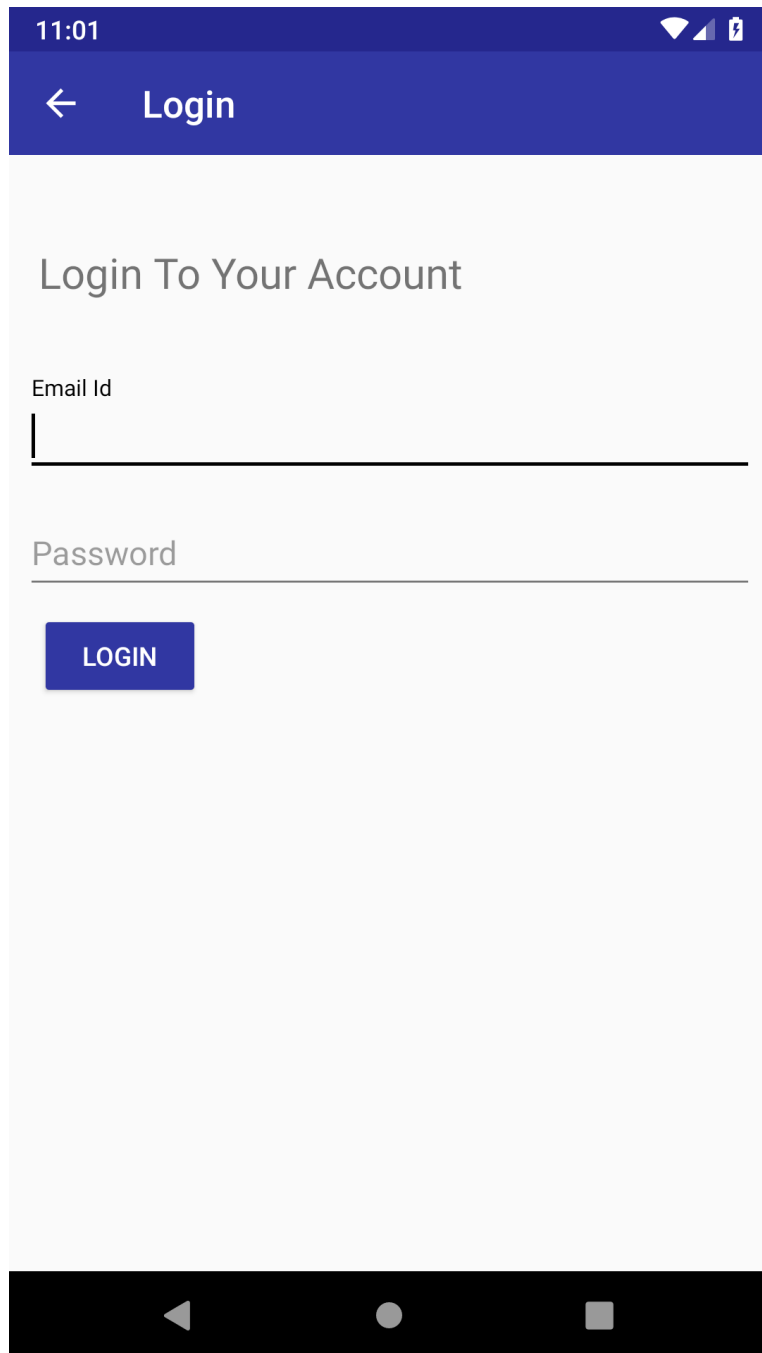
Display\_Name

Email-Id

Password

REGISTER

## Login page



A mobile application login screen. At the top is a dark blue header bar containing the time '11:01' on the left and status icons (Wi-Fi, cellular signal, battery) on the right. Below the header is a lighter blue bar with a white back arrow and the text 'Login'. The main area is light gray and contains the text 'Login To Your Account'. Below this are two input fields: 'Email Id' and 'Password', each with a vertical line on the left and a horizontal line below. A blue 'LOGIN' button is positioned below the 'Password' field. At the bottom is a black navigation bar with three white icons: a triangle, a circle, and a square.

11:01

← Login

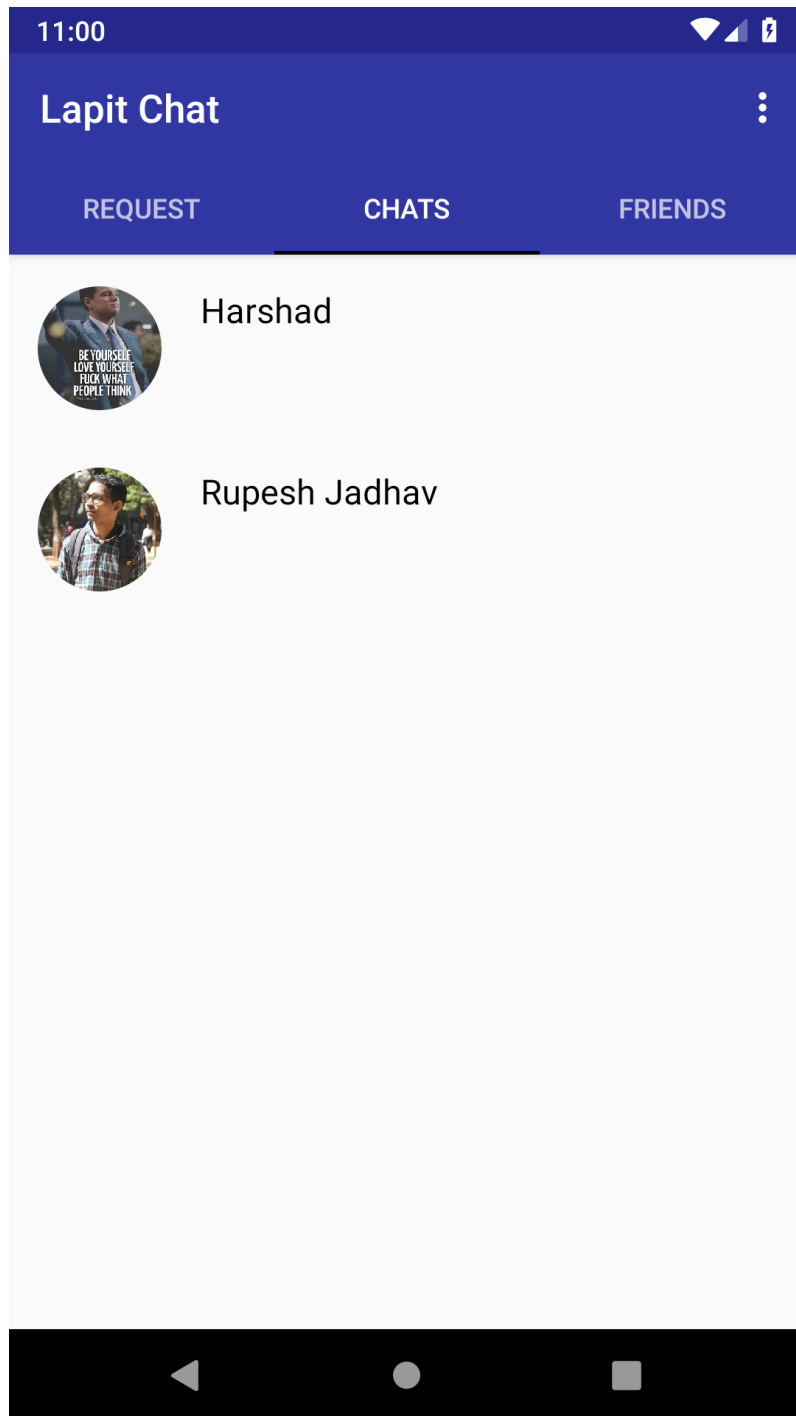
Login To Your Account

Email Id

Password

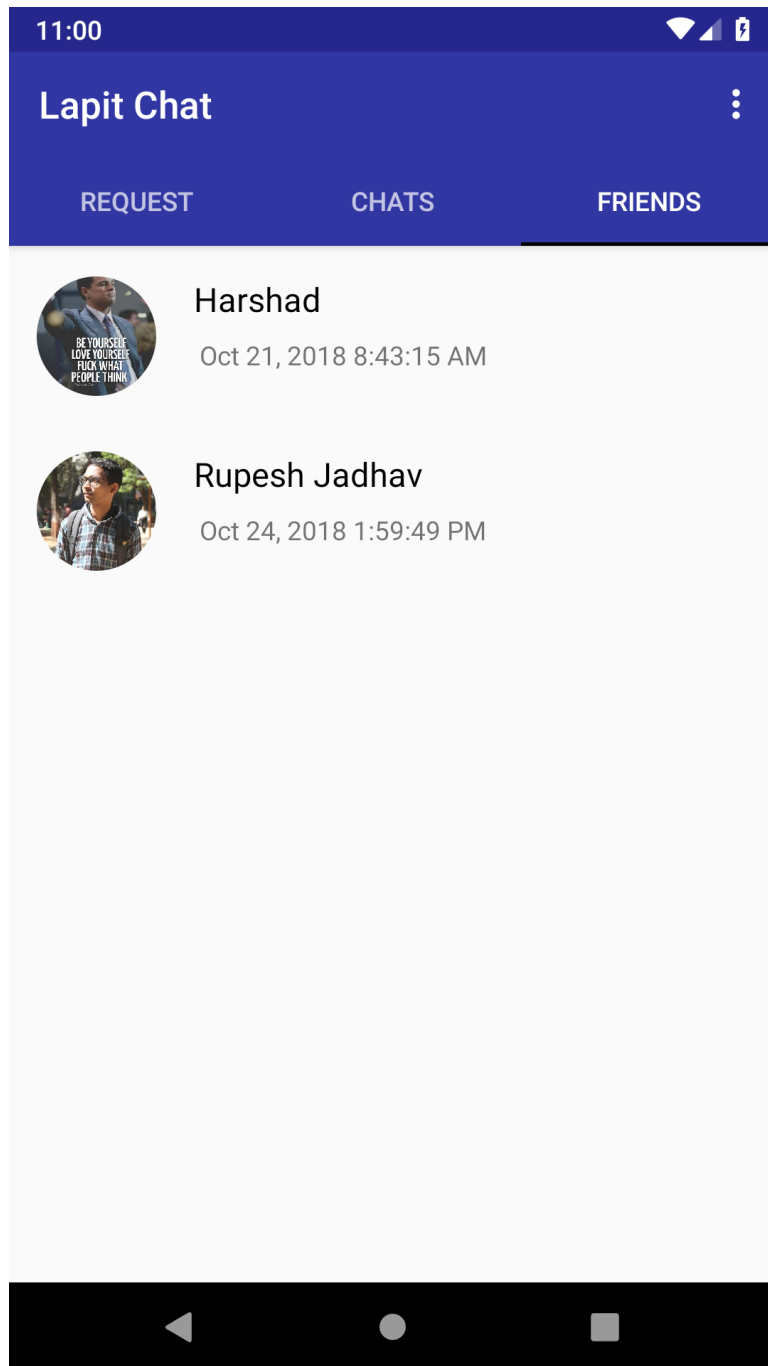
LOGIN

## Main page

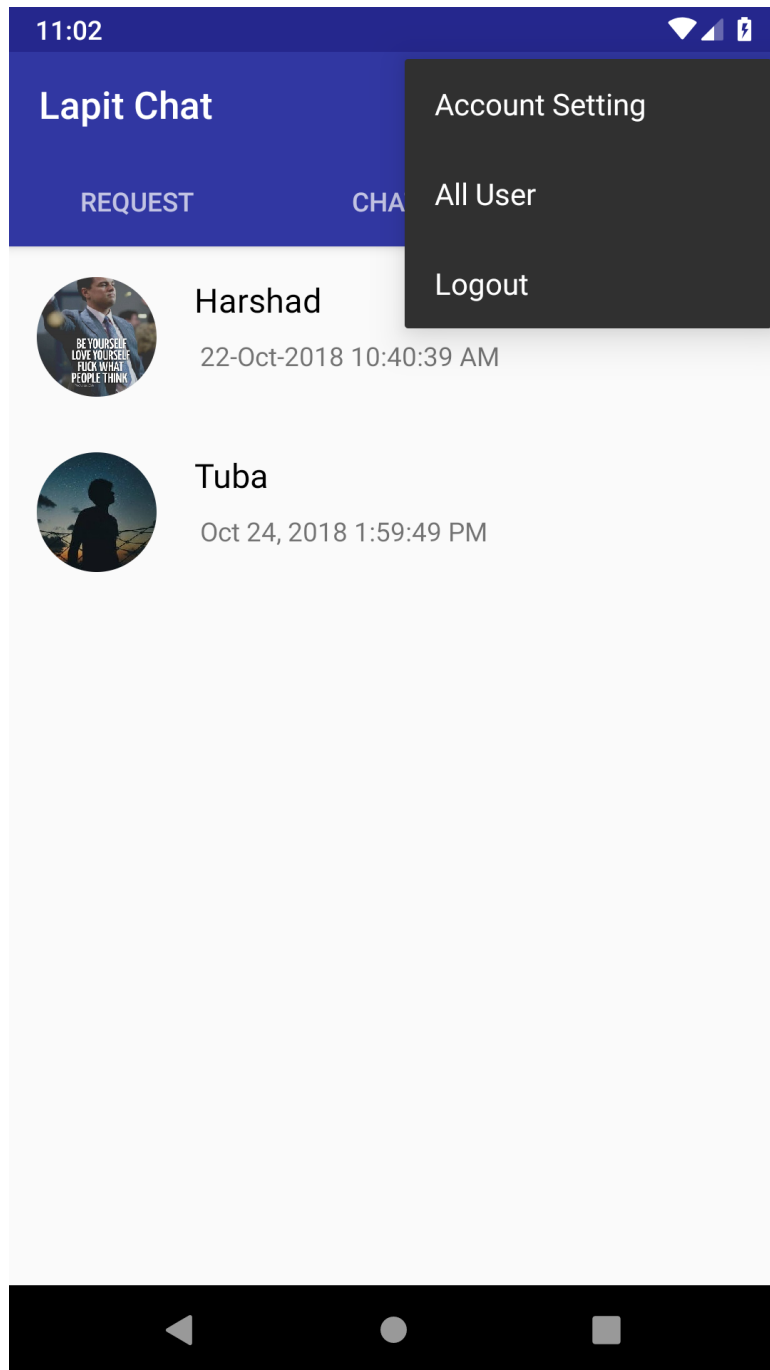




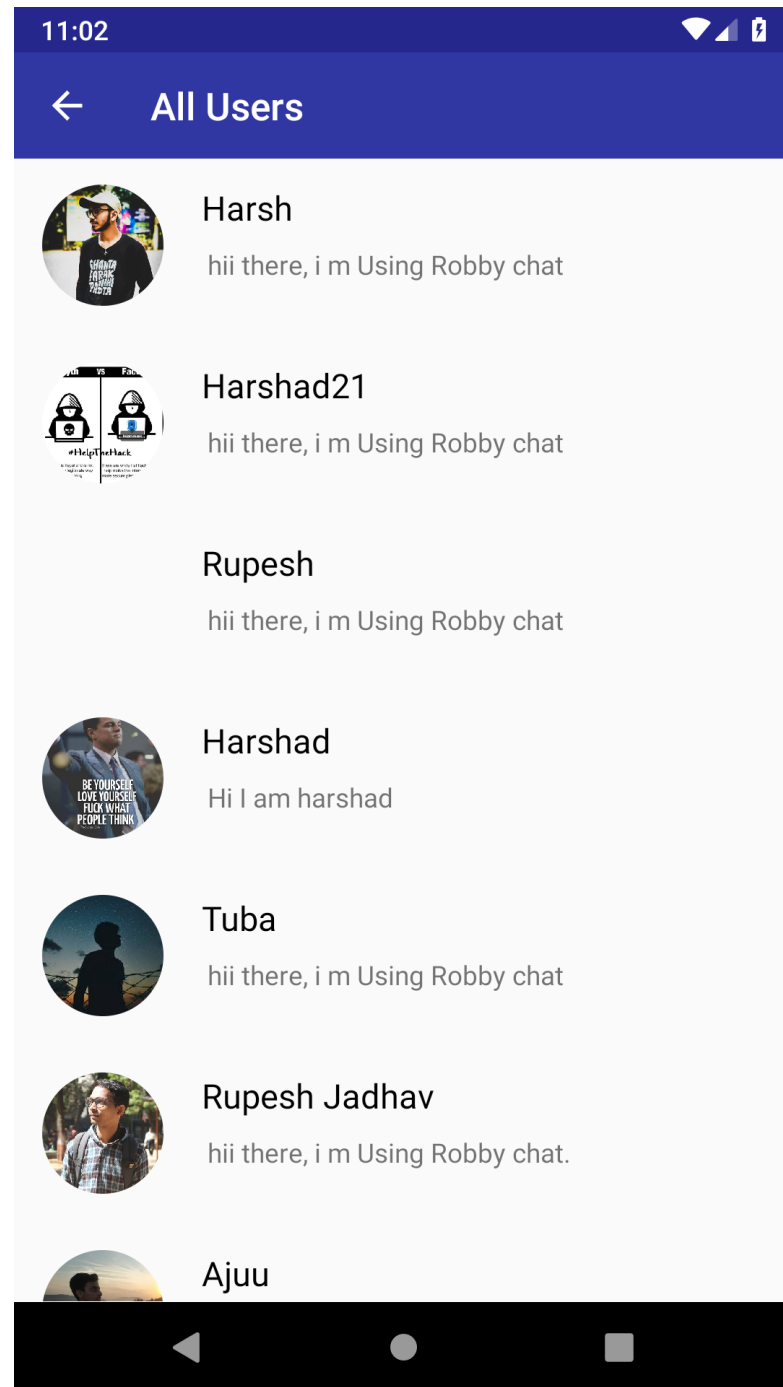
## Friends page



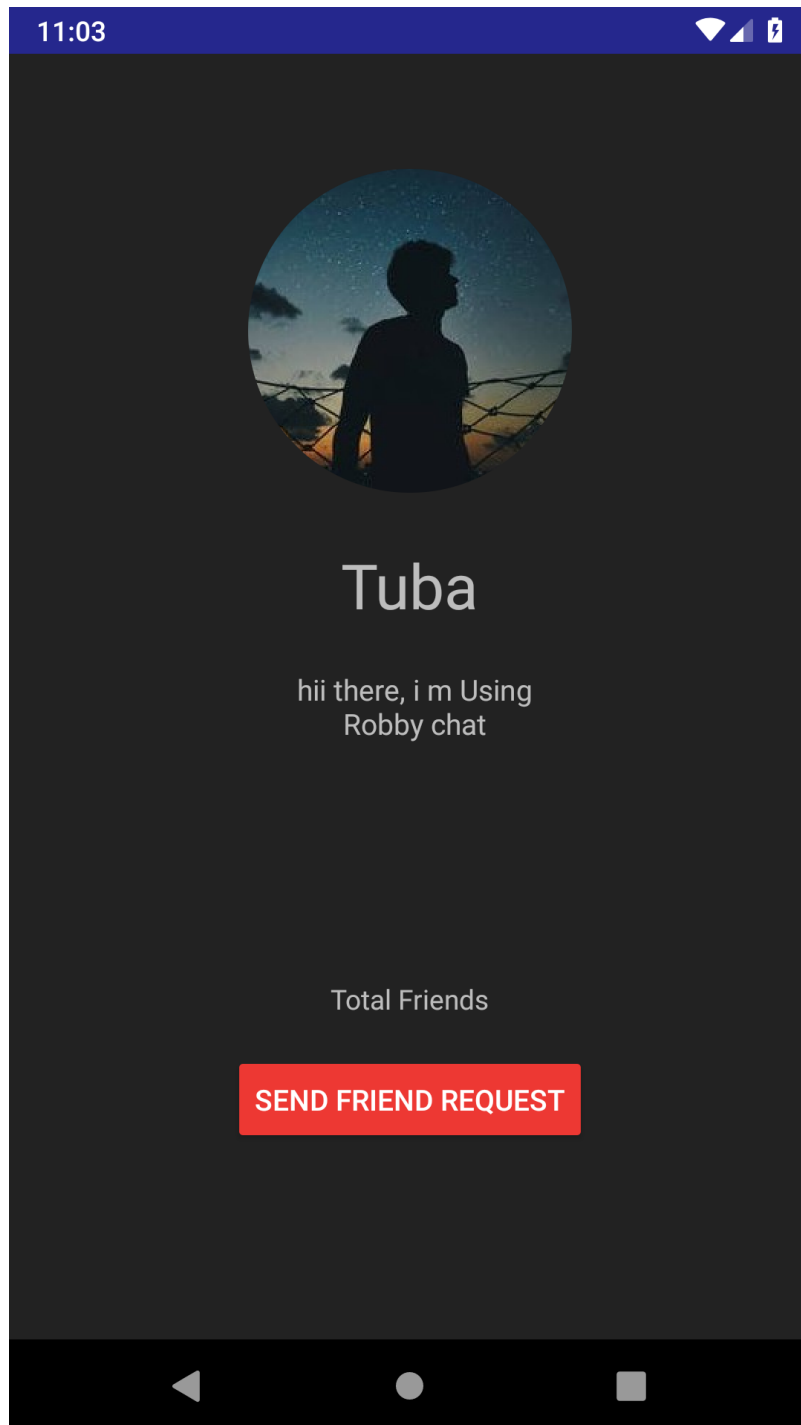
## Menus



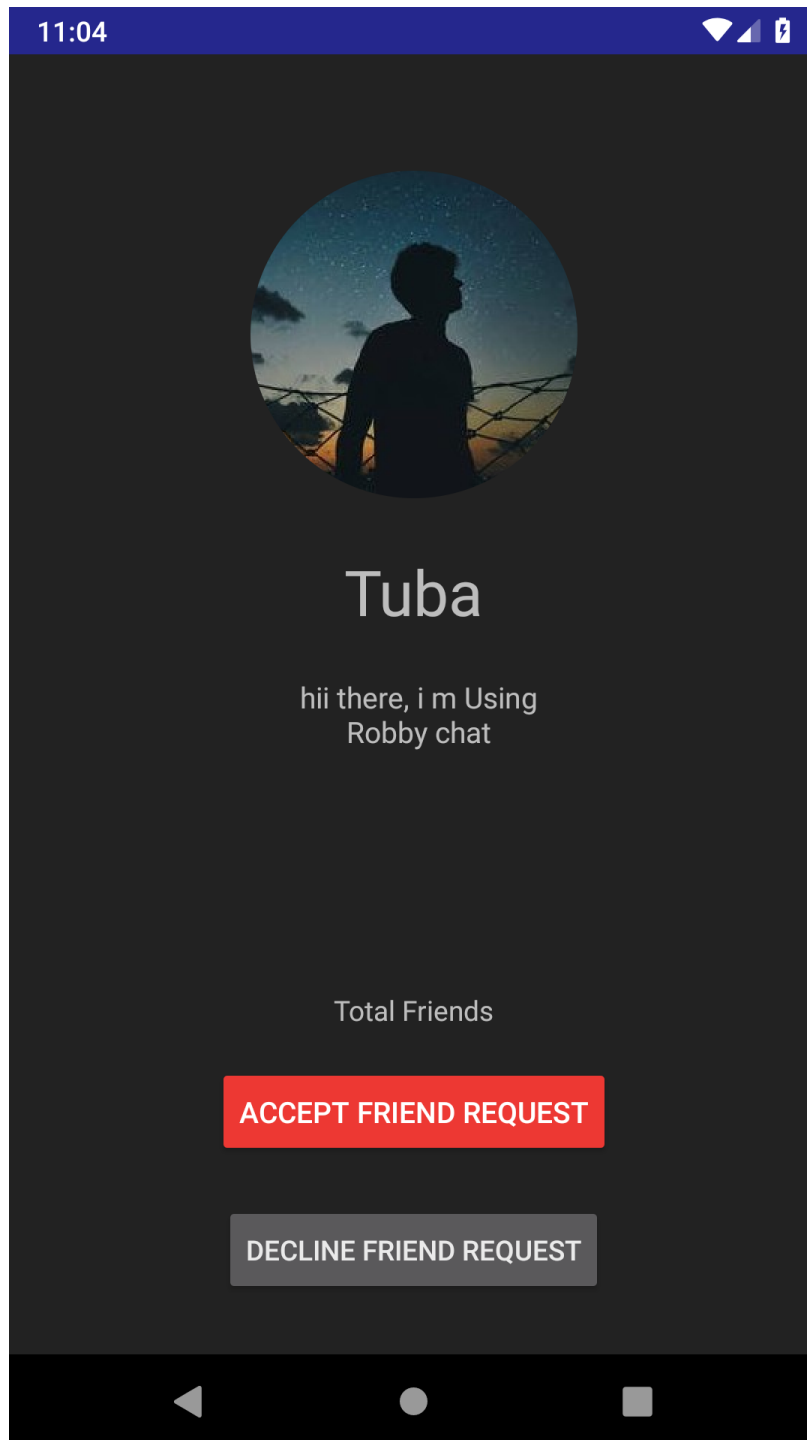
## See all the users



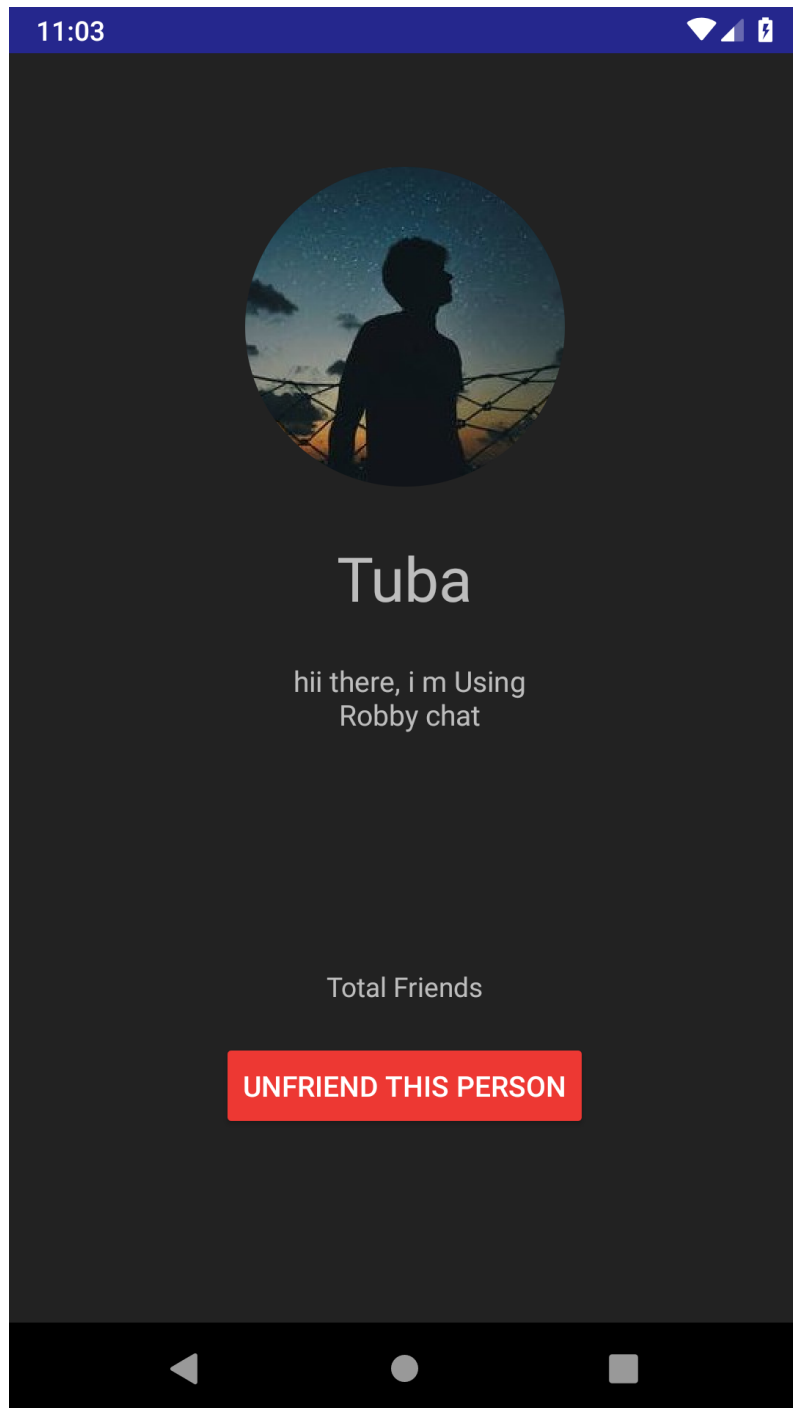
## Send friend request



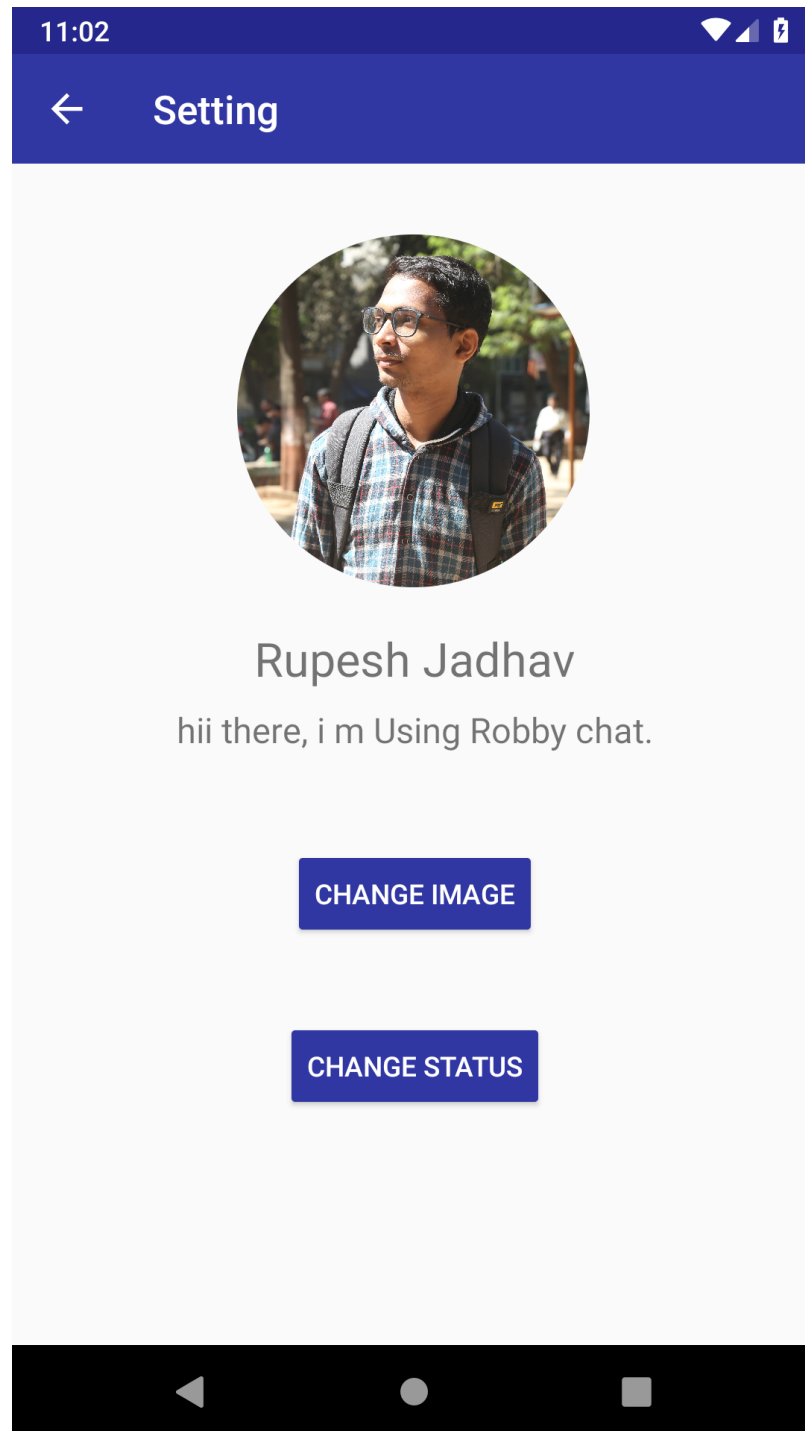
## Accept and decline friend request



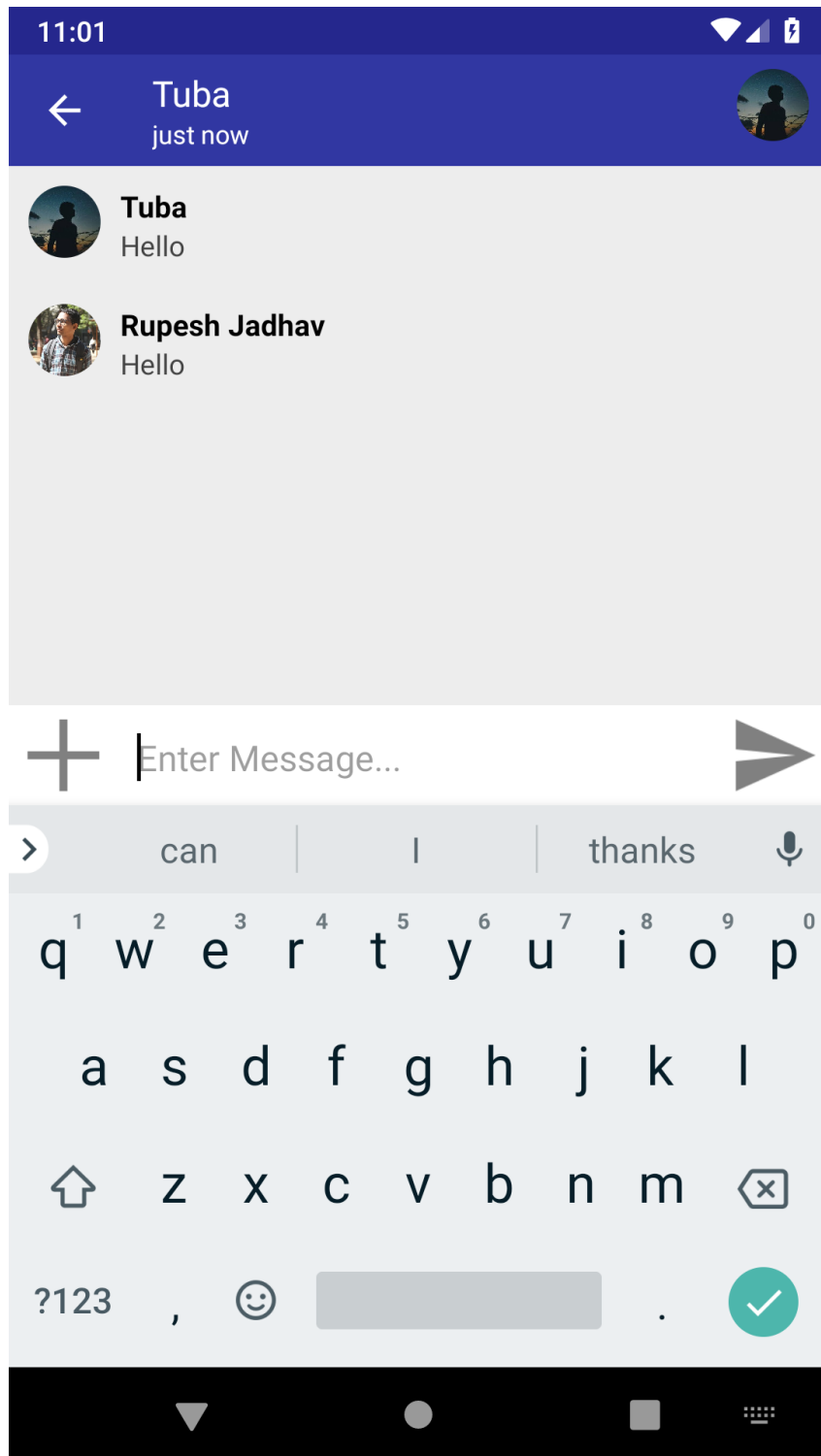
## Unfriend



## User profile page



## Chat page





# **Future Enhancement**

- We Can Provide More Facilities Like Users Can See The mag timing, sending images, video call.

## **References**

### **Links:-**

<https://www.youtube.com/watch?v=Cw0SmoroyNg>