MINI-PROJECT
(2020-2021)

(Tele Chat APP)

PROJECT REPORT

Department of Computer Engineering & Applications

Institute of Engineering & Technology

Submitted by :-

Shivam shukla (181500671)
Ayush agrawal (181500167)

Submitted to :-

Mrs. Harvinder Kaur Mam

TABLE OF CONTENTS

5.About Android

6.About Android Virtual Device

7.Creating Our Android app: Displaying “Tele Chat app”

8.1 Coding for Tele chat app

1. MainActivity.java

2.Activity_main.xml

Second page:

1. MainActivity.java (for second page)

2.Activity_main.xml

Third page:

1. MainActivity.java (for third page)

2.Activity_main.xml

final page:

1. MainActivity.java (for fourth page)

2.Activity_main.xml

8.Contribution Summary

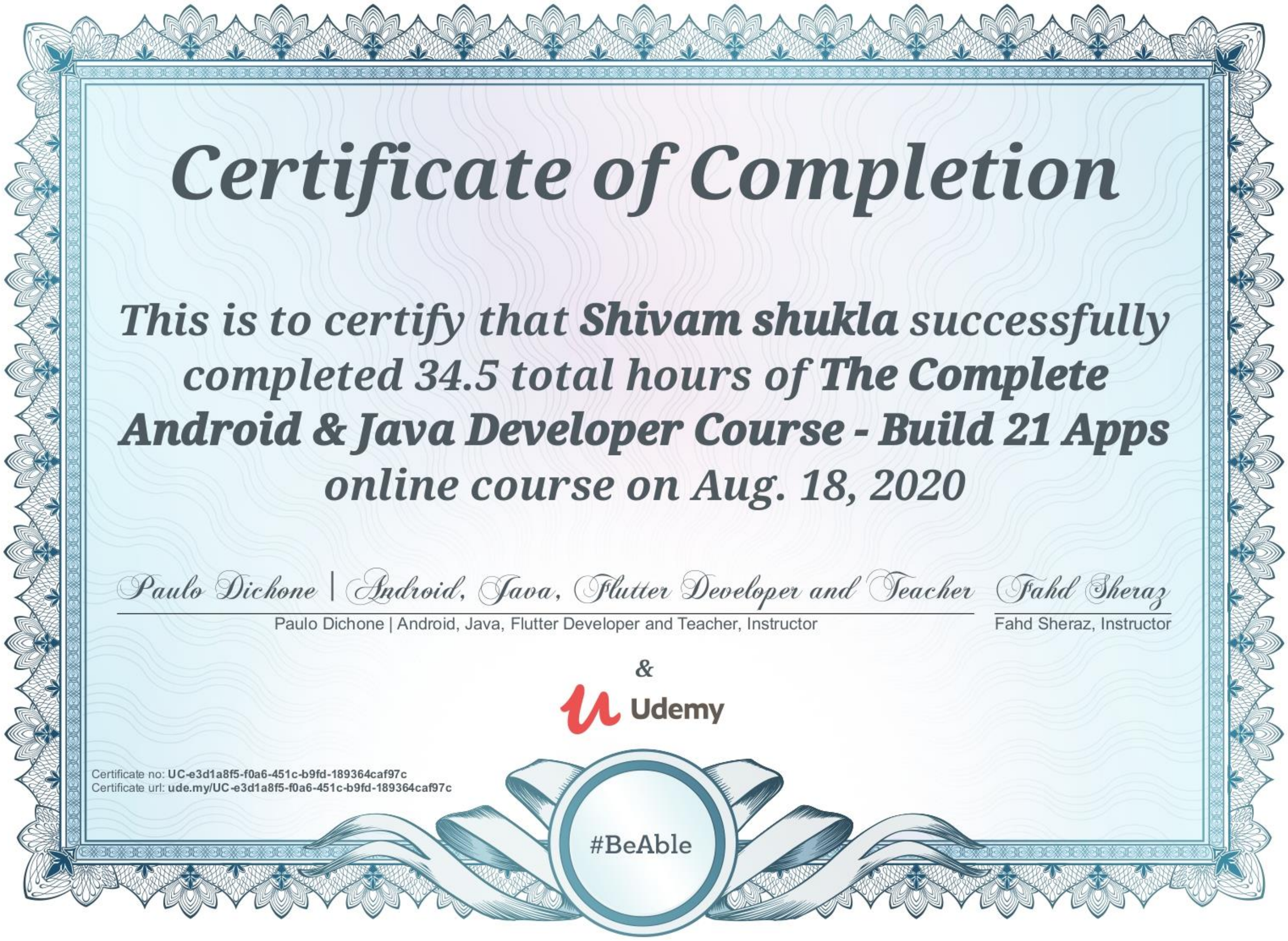
9. REFERENCE

Certificate

This is to certify that Shivam Shukla , Ayush agrawal , Up adhyay students of B.Tech (CSE) 3rd year has successfully Completed the MINI PROJECT named Tele chat APP under the Guidance of Mrs. Harvinder Kaur . During 2020-21.

Summer training certificate-

Shivam Shukla -



Certificate of Completion

This is to certify that **Shivam shukla** successfully completed 34.5 total hours of **The Complete Android & Java Developer Course - Build 21 Apps** online course on Aug. 18, 2020

Paulo Dichone | Android, Java, Flutter Developer and Teacher

Paulo Dichone | Android, Java, Flutter Developer and Teacher, Instructor

Fahd Sheraz

Fahd Sheraz, Instructor

&

Udemy

Certificate no: **UC-e3d1a8f5-f0a6-451c-b9fd-189364caf97c**
Certificate url: **ude.my/UC-e3d1a8f5-f0a6-451c-b9fd-189364caf97c**

#BeAble

Ayush agrawal -

Signature:

Mrs. Harvinder Kaur Mam

(Mentor)

SYNOPSIS -

Project Information:

Title Of Project/Training/Task-
Tele Chat app(WhatsApp clone)

Role & Responsibility
SHIVAM SHUKLA: Front-end AYUSH Agrawal : Back-end

Technical Details
Hardware Requirements:
Minimum Marshmallow version 7.0 (for android)
Pentium IV or higher, (PIV-300GHz recommended)
Hard-Disk 250GB or more
RAM 4GB or more
Processor i3 (7th Gen)
1024 x 768 Display

Internet

Software Requirements:

SYSTEM SOFTWARE: -
Operating System (Windows, Linux, MacOS)

APPLICATION SOFTWARE:-
a) java
b) Xml
c) GitHub
d) Web Browser (Google Chrome, Firefox, Safari, Mozilla, Internet Explorer)

Acknowlegment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many Teachers. I would like to extend my sincere thanks to all of them.

We are highly indebted to Mrs. Harvinder Kaur for her guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.
We would like to express my gratitude towards our parents & member of GLA University for their kind co-operation and encouragement which help us in completion of this project.

Abstract

The report presents the three tasks that is being in continuation which are listed below:

1. Understanding of the Problem objective & implication.
2. Understanding of the data & building of the model.
3. Evaluation of the model.

All these tasks have been completed successfully and results were according to expectations. All the tasks were need very systematic approach, starting from the collection of the data to the implementation of the solution and till evaluation of the System. The most challenging task was the domain knowledge, to understand the language. It is one of the major areas and really need very fundamental and conceptual knowledge of java and Xml.

1. INTRODUCTION -

1.1 Overview

- All the functional/non-functional requirements, corresponding DFD's, UML and Use Case Diagrams have been organized in this report. Along with these designs, this report also contains the essential data of this project.
- The complete description of the application followed by the functionalities has been listed initially. Later on, the media player app has been described diagrammatically with the help of different designing tools like Data Flow Diagram, Use Case Diagram, Interaction Diagram and E-R Diagram.

1.2 Motivation-

Tele chat app has a number of advanced security features. It has end-to-end encryption, just like Apple's iMessage and Signal. All messages that flow through the platform are protected so that only the sender and the recipient can see them. Tele chat app will not be able to read your messages, even if it wanted to. The application does not store your personal information, and only the people you approve as contacts can send you messages.

1.3 Problem Statement

In general, it is well-known that smartphone usage influences user privacy. As phones tend to remain on, their usage patterns provide deep insights into their owners’ lives. With this project, we aim to raise awareness of the fact that this is particularly true for messenger apps.

While the upload of users' address books has, so far, been the main concern with messenger apps, we will shed light on the more subtle privacy implications of unintentional system leaks. These are inherent side effects of using a specific messenger service.

One of the most popular — and, at the same time, probably the most underestimated privacy-related system leak — is the user's presence status. Popular messenger apps like WhatsApp or Telegram display the current "online" status of a contact whenever a contact opens the messenger app while connected to the Internet. Moreover, a "last seen” timestamp is displayed for each contact. This refers to the time the user was last connected.

1.4 Objective

our objective of this project is to design and implement user friendly, platform Tele chat app is for two or more people to communicate almost simultaneously through the Internet connection and through text messages. It has all the benefits of an enhanced chat system with the facilities to share all kinds of documents. It is available on both iOS and Android or even Windows Phone.

2. System and Requirement Analysis

2.1 System Analysis

System analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of the system into its components.

It is a process of studying a system in order to define its goals or purposes and to discover operations and procedures for accomplishing them most efficiently.

2.2 Role of System Analyst

The system analyst is a person who is thoroughly aware of the system and guides the system development project by giving proper directions. He is an expert having technical and interpersonal skills to carry out development tasks

required at each phase.

2.2.1 Main Roles of System Analyst:

- Defining and understanding the requirement of user through various fact-finding techniques.
- Prioritizing the requirements by obtaining user consensus.

- Maintains analysis and evaluation to arrive at appropriate
- te system which is more user friendly.
- Draw certain specifications which are easily understood by users and programmer in precise and detailed form.
- Implement the logical design of system which must be modular.

2.3 Users

The User listen music with the help of app.

2.4 Methodology

Our methodology is designed to help you take maximum advantage of the internet technologies. It incorporates all aspects related to our APP and allows us to ensure that the final product is of the highest standards. Below are the steps we will take to ensure that all your deliverables are completed in time, within budget also we will try to solve each and every problem efficiently.

Requirements analysis

The first step for us is to analyze your and your target market's requirements. Who will be visiting your APP, what will be the purpose of their visit, what is the primary goal of your APP, how can your organization best cater to their needs etc.? Many such questions are analyses for the Needs Analysis stage.

If we are given access to the current APP statistics, we would also like to analyze your current page views, average user time spent on the site, top landing pages, existing search engine rankings, existing bounce rates and many such factors. We analyze your online target audience and assess your differentiation strategy to best attract and retain your online visitors.

Your APP will also undergo comprehensive search engine analysis twice during the course of the project; once during this stage and once again after the deployment (Go Live phase) of your app.

HARDWARE REQUIREMENT (MINIMUM)

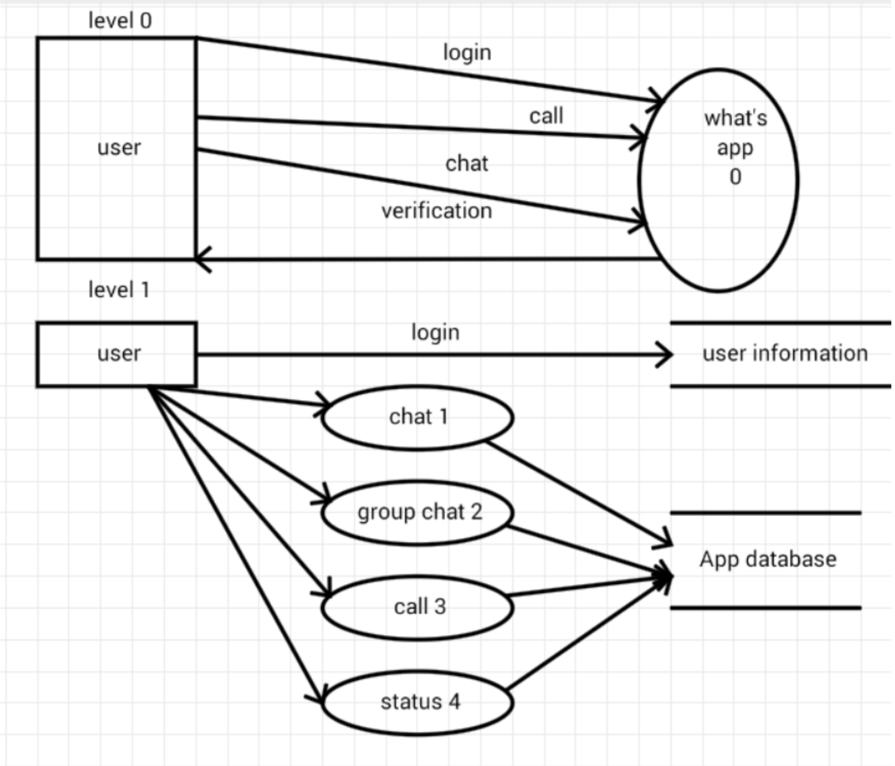
- 20 GB OF HARDDISK
- Processor i3 (7th Gen)
- 1024 x 768 Display
- Internet
- Minimum Marshmallow version 7.0 (for android)
- Pentium IV or higher, (PIV-300GHz recommended)
- Hard-Disk 250GB or more
- RAM 4GB or more
- Processor i3 (7th Gen)
-

SOFTWARE REQUIREMENT-

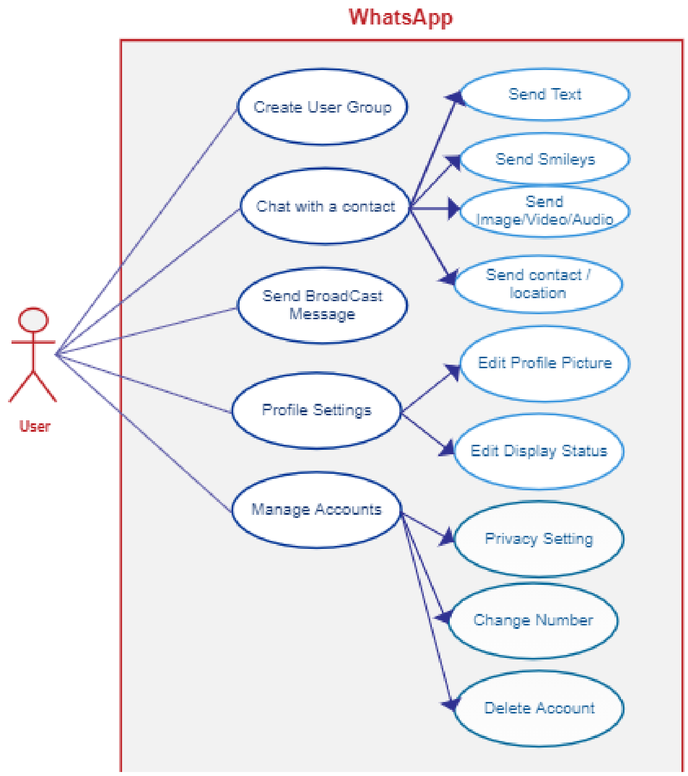
- SYSTEM SOFTWARE
  - Operating System (Windows 7, Linux)
- APPLICATION SOFTWARE
  - ➢ JavaScript (back-end)
  - ➢ Xml
  - ➢ GitHub
- Web Browser:-
  - ➢ Google Chrome, Firefox, Safari, Mozilla, Internet Explorer

2.5 DFD

LEVEL 0 or 1 DFD


level 0
user
login
call
chat
verification
what's
app
0
level 1
user
login
user information
chat 1
group chat 2
call 3
status 4
App database

USE-CASE DIAGRAM



3.Implementation Details

I am creating a telechat app in which i am using java, and Xml.

We making a different layout of app.

To connecting all type layout, we use coding on java

4.Progress

1.) Part 1 is completed

Create front page of Tele Chat app

a. Create a login page .
b. Changing background color white to black.

2.) Part 2 is completed

Create Second page of Tele Chat app

a. Create a find user page .
b. Also create logout button.

3.) Part 3 is completed

Create Third page of Tele Chat app

a. Create a Chat room page.

4.) Part 4 is completed

Create final page of Tele Chat app

a. Complete final deigns of Tele chat app by creating a send message, media buttons .

.

5.About the Android

Android is an operating system based on Linux with a Java programming interface. Android is a mobile operating system (OS) developed by Google. Android is the first completely open source mobile OS. Building on the contributions of the open-source Linux community and more than 300 hardware, software, and carrier partners, Android has rapidly become the fastest-growing mobile OS.

Android versions

1.0 G1 (2008)

Android 1.0 debuted on the HTC Dream (aka T-Mobile G1) and served up apps through Android Market with 35 apps at launch. Its Google Maps used the phone's GPS and Wi-Fi, and it had an Android browser built right in.

1.5 Cupcake (2009)

As the first major update of Android, Cupcake added widgets for the home screen, an onscreen keyboard, video recording in the camera and a copy-and-paste function to the web browser.

1.6 Donut (2009)

Android Donut brought search capabilities for everything on your phone, introduced the Android Market (the precursor to the Google Play store) and additional battery usage info.

2.0 Eclair (2009)

Microsoft users rejoiced at Eclair's support for Exchange. The OS also supported multiple Google accounts and let you search within text and SMS messages. It also added multitouch support and the camera improved with flash and digital zoom.

2.2 Froyo (2010)

Froyo introduced Flash Player 10.1, which bridged gaps by allowing phones to play video and stream audio. The camera flash now worked on video, Bluetooth compatibility was boosted and you could use your phone as a Wi-Fi hotspot.

2.3 Gingerbread (2011)

With Gingerbread, the world got to know near-field communication (NFC), which allowed phones to connect with other nearby devices. The OS also allowed video calling using the front-facing camera and added a download manager.

3.0 Honeycomb (2011)

Honeycomb was the first update for tablets only. It included 3D graphics support, side-by-side browser tabs, video chatting with Google Talk, Bluetooth tethering and full-screen mode within the photo gallery.

4.0 Ice Cream Sandwich (2011)

With its clunky name, ICS merged the phone and tablet operating systems. It added face recognition for unlocking phones, canned text responses to declined calls and live video effects in the camera.

4.1 Jelly Bean (2012)

Jelly Bean had faster, smoother performance thanks to "Project Butter." It let you interact more with expandable notifications, the Chrome browser became the default, widgets became resizable and Google Now came preinstalled.

4.4 KitKat (2013)

Google partnered with an actual food company to create the KitKat OS. As for the software itself, it added emojis to the Google Keyboard, had a smaller memory footprint to support lower-end phones and let you print on the go with Google Cloud Print.

5.0 Lollipop (2014)

Google overhauled its aesthetic completely with a flat interface known as Material Design. Notifications came in banners across the lock screen or as pop-up alerts. The OS also had priority mode, multiuser support, screen pinning and recent apps were renamed Overview.

6.0 Marshmallow (2015)

Marshmallow introduced Doze Mode to save on battery life. It also added built-in support for a fingerprint reader, USB Type-C and 4K mode for apps.

7.0 Nougat (2016)

With Nougat, you could finally clear all your apps in Overview with a single tap. It let you adjust the skin tone of your emojis and tap through more Quick Setting options. It also supported Google's VR endeavor, Daydream.

8.0 Oreo (2017)

Timed to launch right alongside the 2017 US solar eclipse, Android Oreo served up picture-in-picture apps for even more multitasking. There was also a better copy-and-paste experience, improved security and better battery management.

9.0 Pie (2018)

Android Pie focused on behind-the-scenes improvements that made Android phones work faster and save battery life. That included AI tools that suggested apps and shortcuts you might want to use, and it loaded "slices" of an online page that gave precise information, without the need to download an app search through a webpage.

10.0 Q (2019)

This year's Android Q, which hasn't been named or rolled out in full yet, introduces the beloved Dark Theme, a new pop-up notifications style called Bubbles and live captioning on videos that are being played for more accessibility.

3.2 Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games etc.

3.3 IT services offered

3.3.1 App design & develop

Having you're App designed and built professionally will improve your company image and help to enforce your brand.

3.3.3 Android Technical Support

Technologies Languages

XML, JAVASCRIPT

Android app development platform

Android Studio

Virtual device

Server platform

windows

`

6. Introduction to the Development tool “Android Studio”

Android Studio is the official integrated development environment (IDE) for Android platform 3development. Android Studio is freely available under the apache license. Android Studio is designed specifically for Android development. Android application development can be started on either of the following operating systems –

Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit).

Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks).

GNOME or KDE desktop. All the required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's that is needed before starting Android application programming.

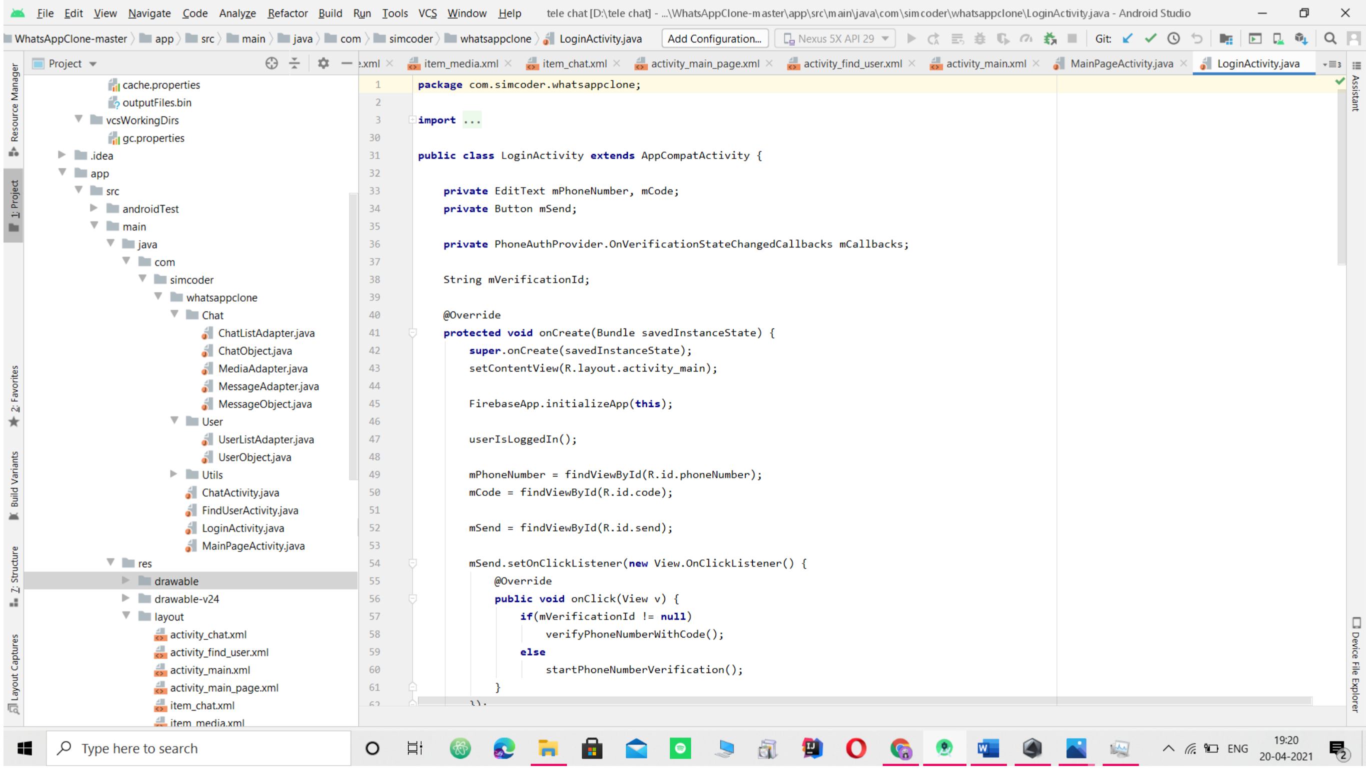
Java JDK5 or later version Java Runtime Environment (JRE) 6

Android Studio.

7. Creating Our Android app: Displaying “Tele Chat App”

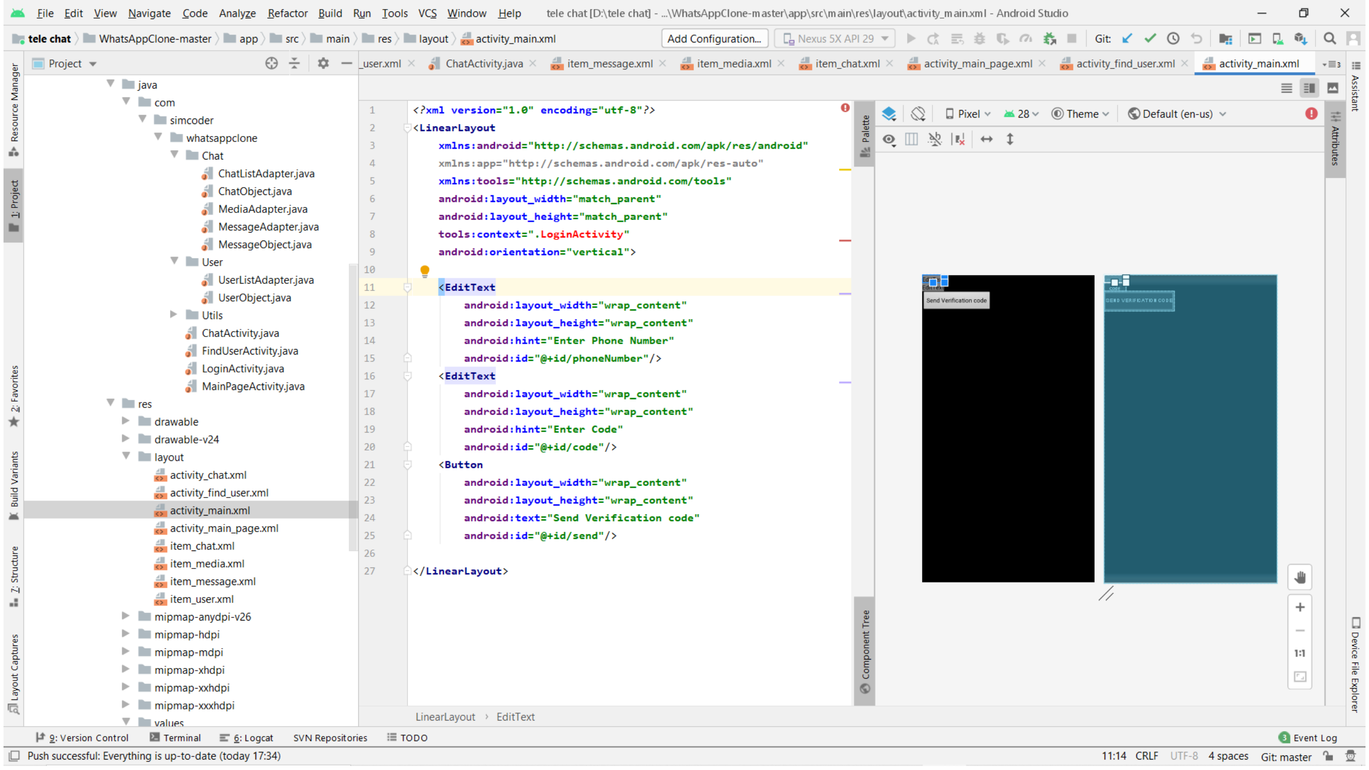
First page:

1.MainActivity.java



```java
package com.simcoder.whatsappclone;

import ...

public class LoginActivity extends AppCompatActivity {

    private EditText mPhoneNumber, mCode;
    private Button mSend;

    private PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks;

    String mVerificationId;

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

        FirebaseApp.initializeApp(this);

        userIsLoggedIn();

        mPhoneNumber = findViewById(R.id.phoneNumber);
        mCode = findViewById(R.id.code);

        mSend = findViewById(R.id.send);

        mSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(mVerificationId != null)
                    verifyPhoneNumberWithCode();
                else
                    startPhoneNumberVerification();
            }
        });
```

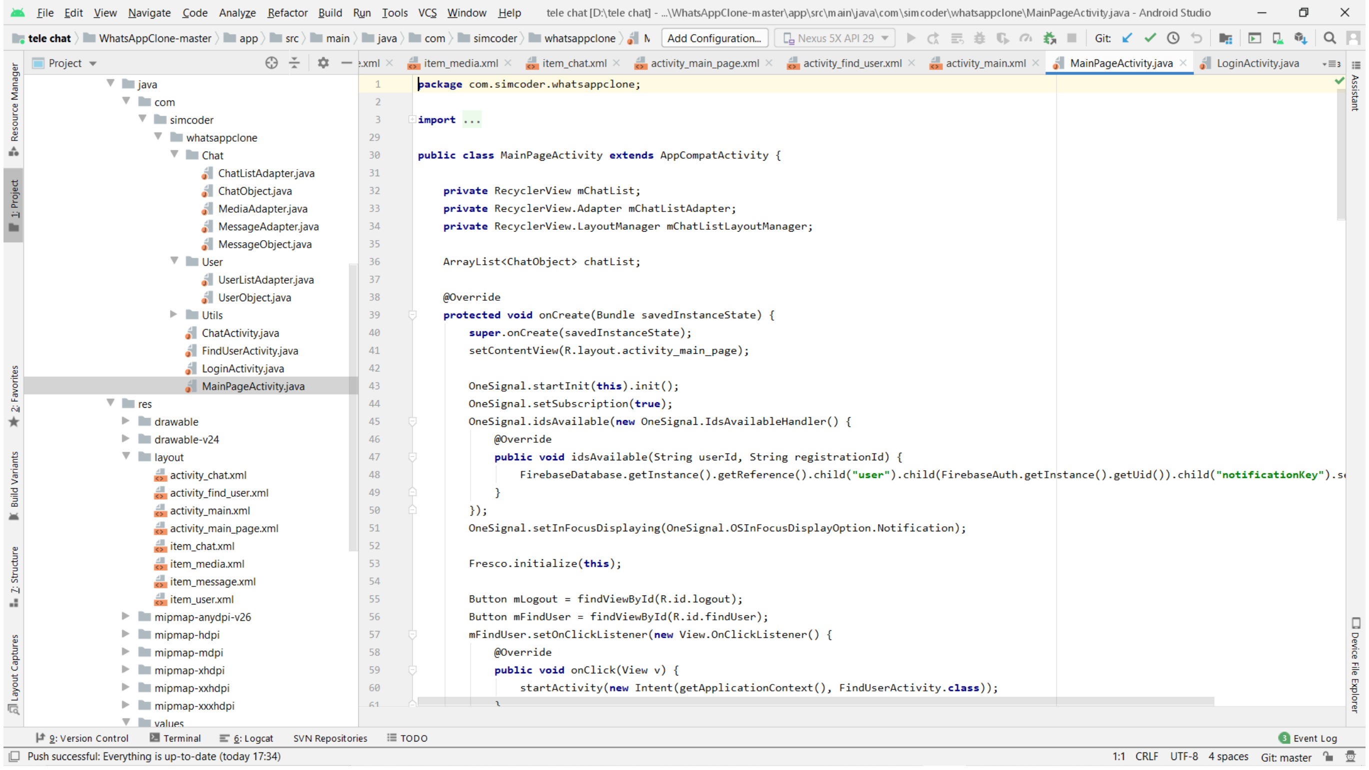
2.Activity_main.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    tools:context=".LoginActivity"
    android:orientation="vertical">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter Phone Number"
        android:id="@+id/phoneNumber"/>
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter Code"
        android:id="@+id/code"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Verification code"
        android:id="@+id/send"/>

</LinearLayout>
```

Second page:

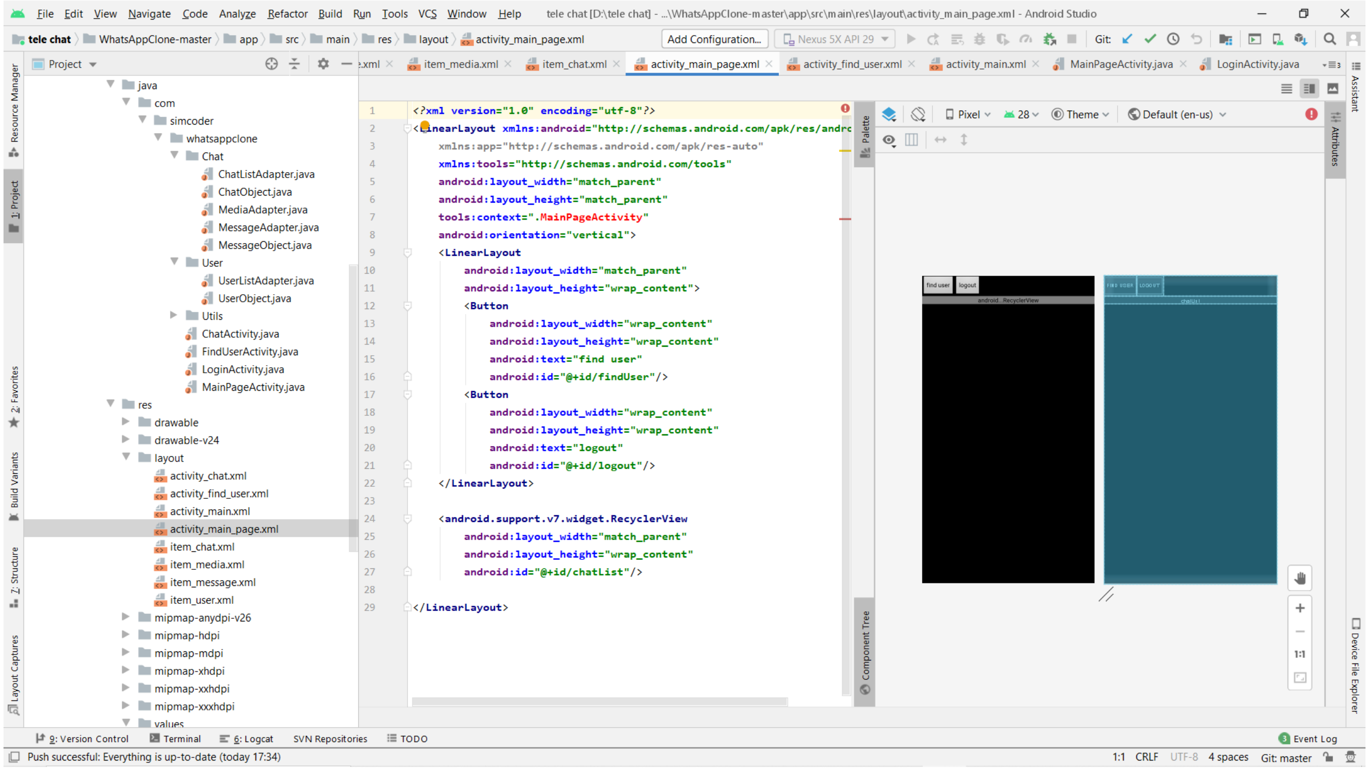
1.MainActivity.java



```java
package com.simcoder.whatsappclone;

import ...

public class MainPageActivity extends AppCompatActivity {

    private RecyclerView mChatList;
    private RecyclerView.Adapter mChatListAdapter;
    private RecyclerView.LayoutManager mChatListLayoutManager;

    ArrayList<ChatObject> chatList;

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

        OneSignal.startInit(this).init();
        OneSignal.setSubscription(true);
        OneSignal.idsAvailable(new OneSignal.IdsAvailableHandler() {
            @Override
            public void idsAvailable(String userId, String registrationId) {
                FirebaseDatabase.getInstance().getReference().child("user").child(FirebaseAuth.getInstance().getUid()).child("notificationKey").s
            }
        });
        OneSignal.setInFocusDisplaying(OneSignal.OSInFocusDisplayOption.Notification);

        Fresco.initialize(this);

        Button mLogout = findViewById(R.id.logout);
        Button mFindUser = findViewById(R.id.findUser);
        mFindUser.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), FindUserActivity.class));
            }
```

Push successful: Everything is up-to-date (today 17:34)

2.Activity_main.xml

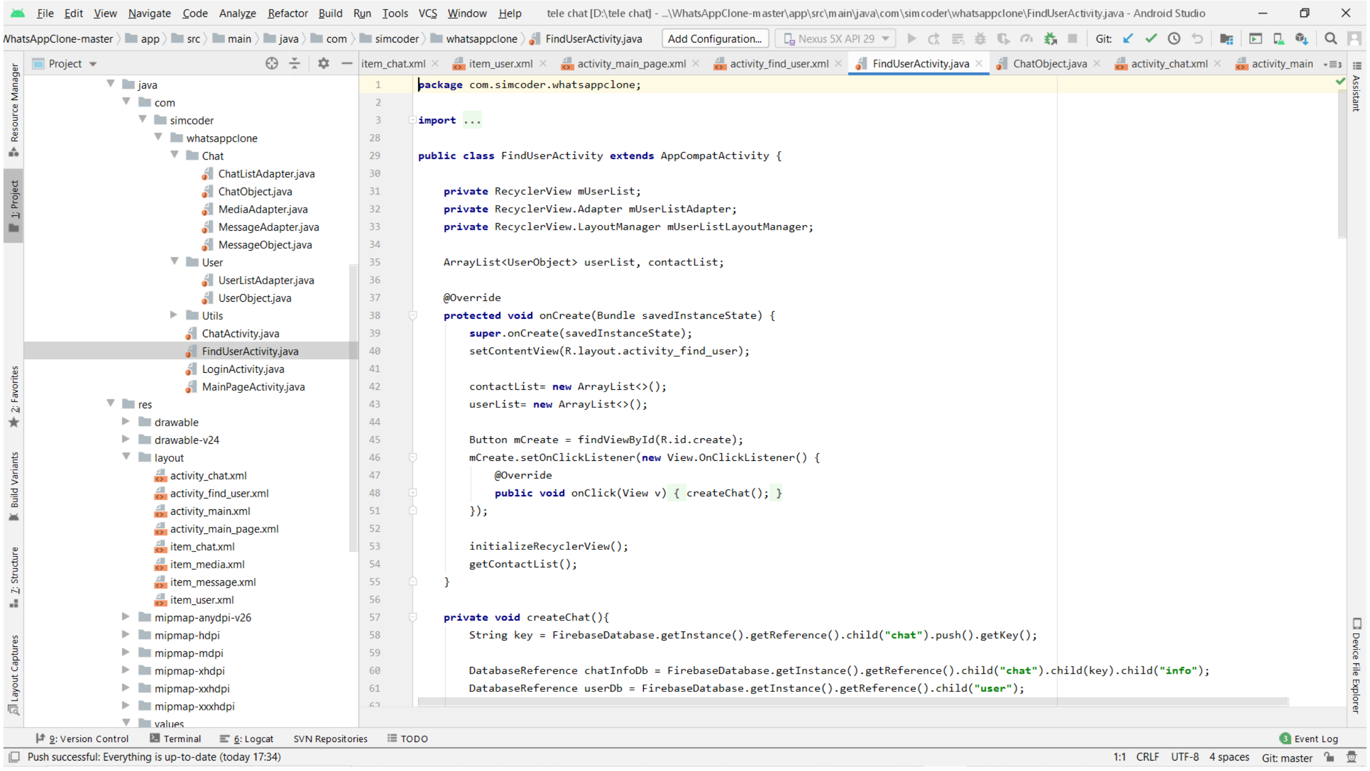


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andro
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainPageActivity"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="find user"
            android:id="@+id/findUser"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="logout"
            android:id="@+id/logout"/>
    </LinearLayout>

    <android.support.v7.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/chatList"/>

</LinearLayout>
```

Push successful: Everything is up-to-date (today 17:34)

Third page:

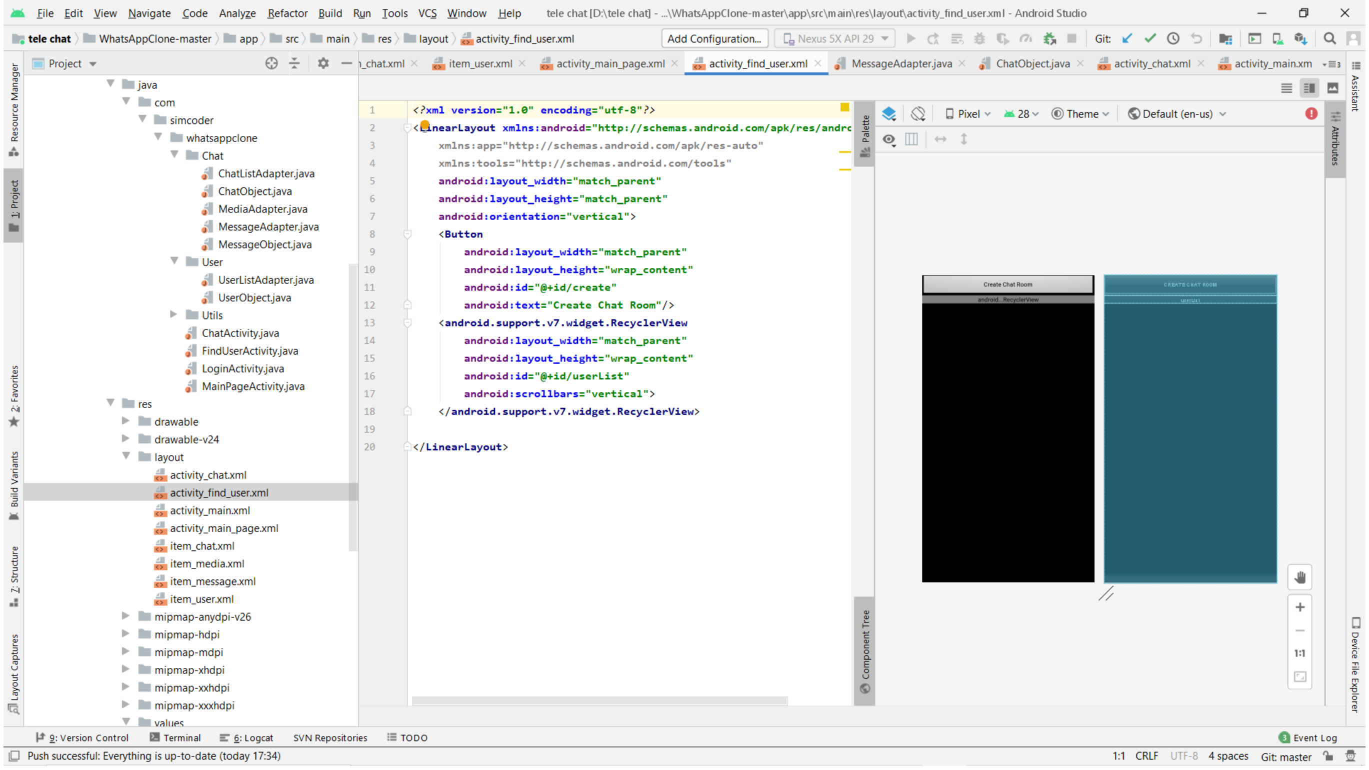
1.MainActivity.java



```
package com.simcoder.whatsappclone;

import ...

public class FindUserActivity extends AppCompatActivity {

    private RecyclerView mUserList;
    private RecyclerView.Adapter mUserListAdapter;
    private RecyclerView.LayoutManager mUserListLayoutManager;

    ArrayList<UserObject> userList, contactList;

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

        contactList= new ArrayList<>();
        userList= new ArrayList<>();

        Button mCreate = findViewById(R.id.create);
        mCreate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { createChat(); }
        });

        initializeRecyclerView();
        getContactList();
    }

    private void createChat(){
        String key = FirebaseDatabase.getInstance().getReference().child("chat").push().getKey();

        DatabaseReference chatInfoDb = FirebaseDatabase.getInstance().getReference().child("chat").child(key).child("info");
        DatabaseReference userDb = FirebaseDatabase.getInstance().getReference().child("user");
```

Push successful: Everything is up-to-date (today 17:34)

2.Activity_main.xml



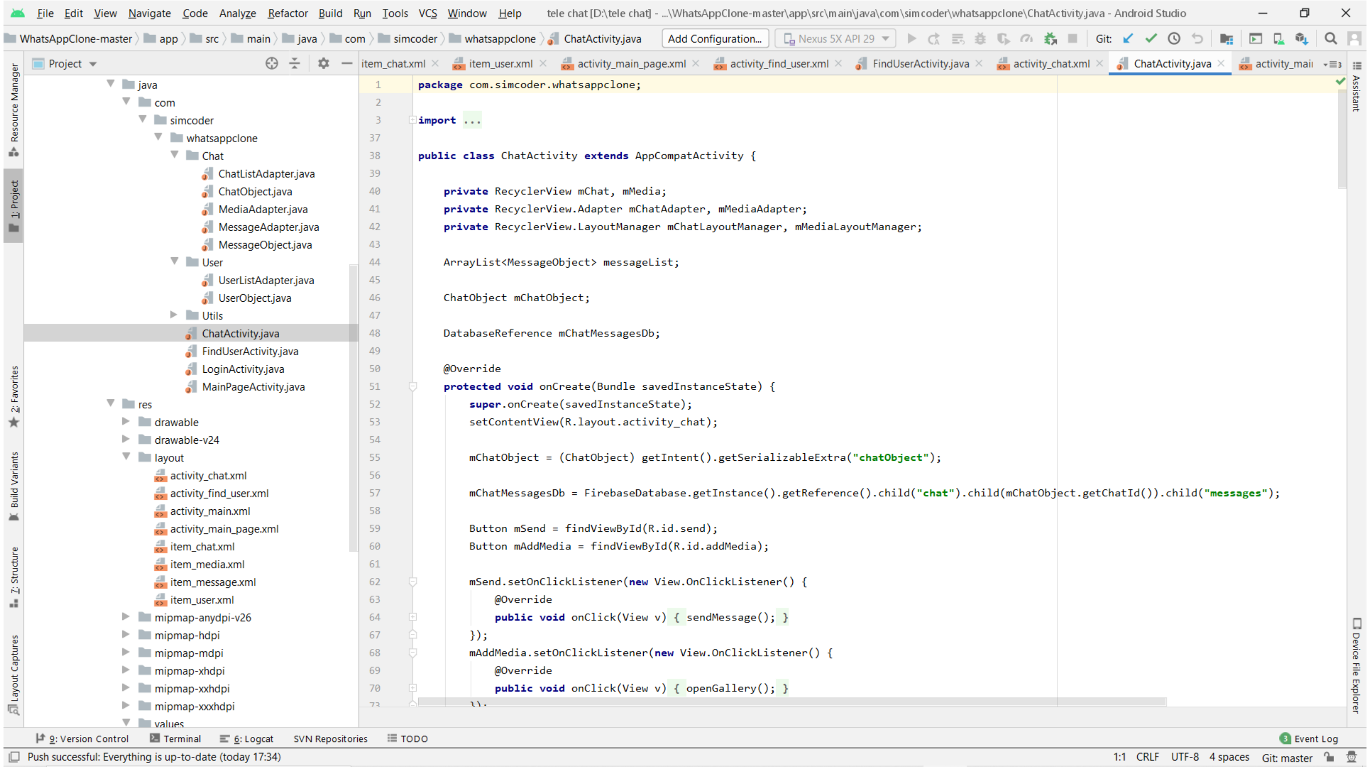
tele chat [D:\tele chat] - ...\WhatsAppClone-master\app\src\main\res\layout\activity_find_user.xml - Android Studio

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andrc
    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:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/create"
        android:text="Create Chat Room"/>
    <android.support.v7.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/userList"
        android:scrollbars="vertical">
    </android.support.v7.widget.RecyclerView>

</LinearLayout>
```

Push successful: Everything is up-to-date (today 17:34)

Final page:

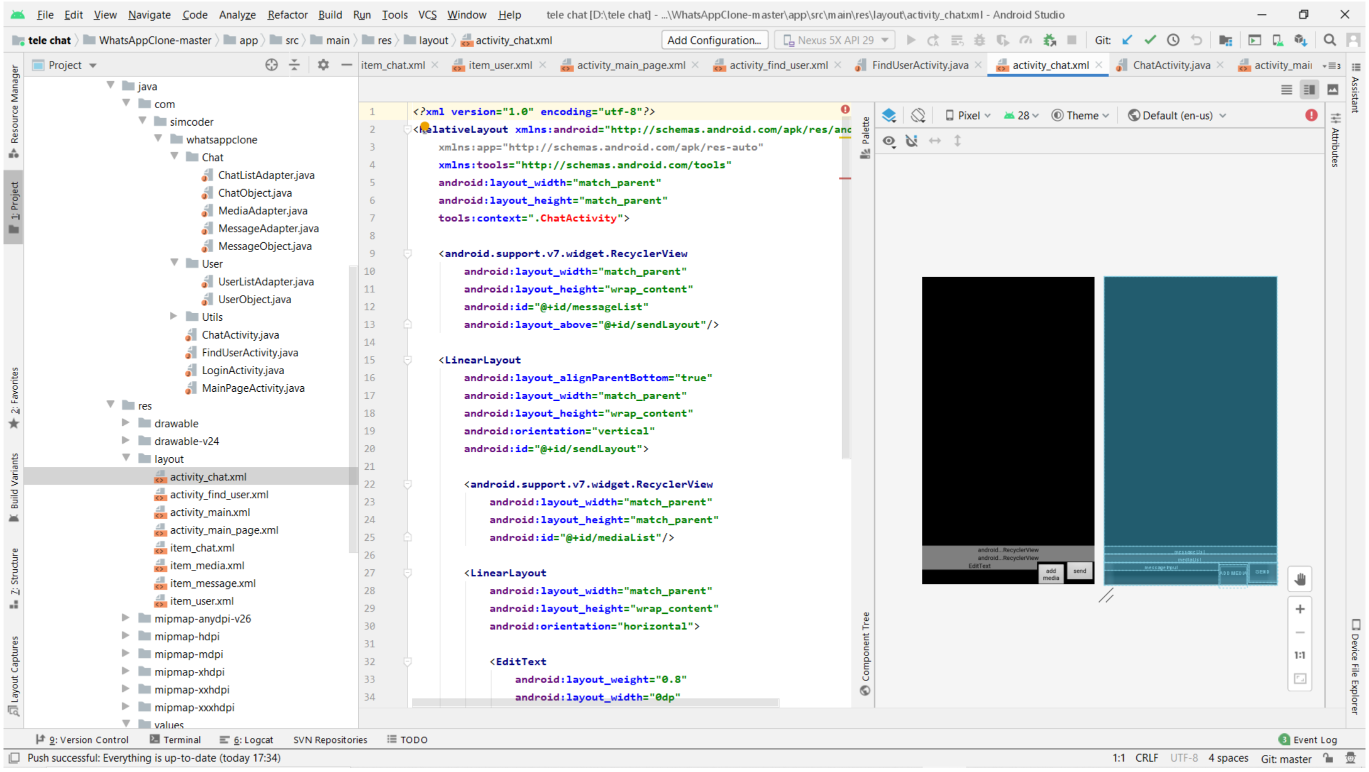
1.MainActivity.java



```java
package com.simcoder.whatsappclone;

import ...

public class ChatActivity extends AppCompatActivity {

    private RecyclerView mChat, mMedia;
    private RecyclerView.Adapter mChatAdapter, mMediaAdapter;
    private RecyclerView.LayoutManager mChatLayoutManager, mMediaLayoutManager;

    ArrayList<MessageObject> messageList;

    ChatObject mChatObject;

    DatabaseReference mChatMessagesDb;

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

        mChatObject = (ChatObject) getIntent().getSerializableExtra("chatObject");

        mChatMessagesDb = FirebaseDatabase.getInstance().getReference().child("chat").child(mChatObject.getChatId()).child("messages");

        Button mSend = findViewById(R.id.send);
        Button mAddMedia = findViewById(R.id.addMedia);

        mSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { sendMessage(); }
        });
        mAddMedia.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { openGallery(); }
```

Push successful: Everything is up-to-date (today 17:34)

2.Activity_main.xml



tele chat [D:\tele chat] - ...\WhatsAppClone-master\app\src\main\res\layout\activity_chat.xml - Android Studio

tele chat 〉 WhatsAppClone-master 〉 app 〉 src 〉 main 〉 res 〉 layout 〉 activity_chat.xml

```
java
  com
    simcoder
      whatsappclone
        Chat
          ChatListAdapter.java
          ChatObject.java
          MediaAdapter.java
          MessageAdapter.java
          MessageObject.java
        User
          UserListAdapter.java
          UserObject.java
        Utils
        ChatActivity.java
        FindUserActivity.java
        LoginActivity.java
        MainPageActivity.java
res
  drawable
  drawable-v24
  layout
    activity_chat.xml
    activity_find_user.xml
    activity_main.xml
    activity_main_page.xml
    item_chat.xml
    item_media.xml
    item_message.xml
    item_user.xml
  mipmap-anydpi-v26
  mipmap-hdpi
  mipmap-mdpi
  mipmap-xhdpi
  mipmap-xxhdpi
  mipmap-xxxhdpi
  values
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/anc
    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"
    tools:context=".ChatActivity">

    <android.support.v7.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/messageList"
        android:layout_above="@+id/sendLayout"/>

    <LinearLayout
        android:layout_alignParentBottom="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:id="@+id/sendLayout">

        <android.support.v7.widget.RecyclerView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/mediaList"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <EditText
                android:layout_weight="0.8"
                android:layout_width="0dp"
```

Push successful: Everything is up-to-date (today 17:34)

8.Contribution Summary

OUR TEAM:

Frontend Developer: - shivam shukla

Database Manager: - ayush agrawal

9. REFERENCE

- www.wikipedia.org/
- www.udemy.com/
- www.google.com/