# “CONVERSE”

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

**RGUKT**

**Rajiv Gandhi University of Knowledge Technologies**

**R.K.VALLEY**

Submitted by

**N Venkata Raju ---R171148**

**the Esteemed guidance of Mr. Satya Nandaram N RGUKT RK Valley.**

## DECLARATION

I am here by declare that the report of the B.Tech mini Project Work entitled "**CONVERSE**" which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by Me. The material contained in this report has not been submitted to any university or institution for award of any degree.

**N Venkataraju ---R171148**

**Dept. Of Computer Science and Engineering.**

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

**RGUKT**

(A.P.Government Act 18 of 2008)

RGUKT, RK VALLEY

Department of Computer Science and Engineering

## CERTIFICATE FOR PROJECT COMPLETION

This is certify that the project entitled "**CONVERSE**" submitted by **N Venkataraju(R171148)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester-2 2021-2022 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

| | |
|---|---|
| **Project Internal Guide** | **Head of the Department** |
| Mr.N.Satya Nandaram | Mr. Harinaadha P |
| Assistant Professor | HOD Of CSE |
| RGUKT, RK Valley | RGUKT, RK Valley |

## **Abstract**

Converse is a Chatting Application For Communicate with others through Internet . Where people can chat to  other people using this app. To Chat with your friends both users should install this application in  their devices. Using This app People can Improve Their Communication  Skills This app is Designed for the people who want to talk to others  through Messaging whenever you don’t have access to call to others you Just need internet to message your friend  so that your friend will respond to you.

# Index

1. Abstraction 4
2. Introduction 6
2.1 Purpose 6
2.2 Intended Audience 6
2.3 Product Vision 6
2.4 Technologies 6
3. Android Studio 7
4. Creating Project 10
5. Kotlin 12
6. XML 13
7. Firebase 13
8. Features of Firebase 14
8.1 Connecting Android App with Firebase 16
9. System in Context 17
9.1 Context Diagram 17
10. System-Wide Requirements 18
11. Function Requirements 19
11.1 Use Case Diagram 19
12. ER Diagram 20
13. App Interface 25
14. Conclusion 29

15. Future Enhancements 30

16. References 31

# Converse SRS Document

## Introduction:

This document has the requirements of Chatting with Friends. The Converse is used to develop the People Communication skills.

### 1.1: Purpose

The purpose of this document is to gather the requirements that are needed for implementing the Converse. It also focuses on various key features, the product, product vision and scope, product overview. The main purpose of Converse is to provide a platform to the People who connect with Friends through Chatting.

### 1.2:Intended Audience:

The intended audience will be the users who can access the platform to Chat with The Friends and Improve Their Communication Skills.

**Users:** People

## Product Vision:

The product vision is to develop the Communication between the users, which Is Helpful to the Students and Working Professionals. They can Communicate with team mates And their friends.

**Technologies:**

- ➢ Android Studio
- ➢ Kotlin
- ➢ XML
- ➢ Firebase

## Android Studio

Android Studio is the IDE to Build Android Apps. It is integrated Development Kit For the android Development. Android Support Both Java and Kotlin to Write Code. Google Announces that Android Development Will be Kotlin-first. **Android Studio** is the official **IDE (Integrated Development Environment)** for Android app development and it is based on **JetBrains' IntelliJ IDEA** software. Android Studio provides many excellent features that enhance productivity when building Android apps, such as:

- A blended environment where one can develop for all Android devices
- Apply Changes to push code and resource changes to the running app without restarting the app
- A flexible Gradle-based build system
- A fast and feature-rich emulator
- GitHub and Code template integration to assist you to develop common app features and import sample code
- Extensive testing tools and frameworks
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine, and many more.

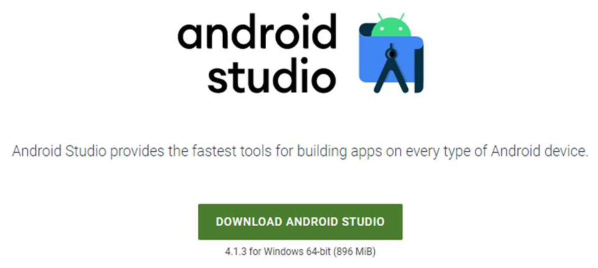
## System Requirements

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

## Installation Guide

**Step 1:** Head over to this link to get the Android Studio executable or zip file.

**Step 2:** Click on the **Download Android Studio Button**



Android Studio provides the fastest tools for building apps on every type of Android device.

Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.

Click on the Save file button in the appeared prompt box and the file will start downloading.

**Step 3:** After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.

Click on next. In the next prompt, it’ll ask for a path for installation. Choose a path and hit next.

**Step 4:** It will start the installation, and once it is completed, it will be like the image shown below

**Step 5:** Once “**Finish**” is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not. It is better to choose the ‘Don’t import Settings option’.

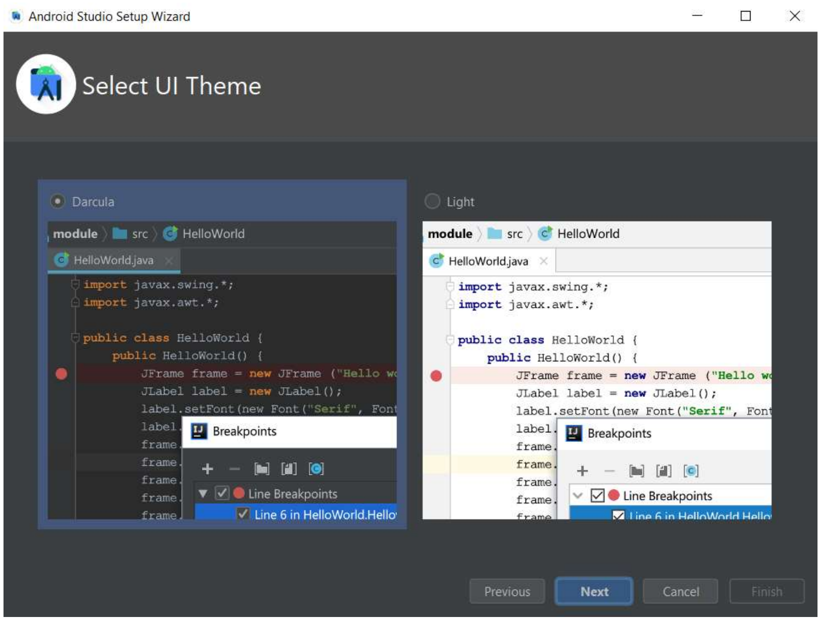
Click the **OK** button.

**Step 6:** This will start the Android Studio.

Meanwhile, it will be finding the available SDK components

**Step 7:** After it has found the SDK components, it will redirect to the Welcome dialog box.

Choose Standard and click on Next. Now choose the theme, whether the **Light** theme or the **Dark** one. The light one is called the **IntelliJ** theme whereas the dark theme is called **Darcula**. Choose as required.



```
import javax.swing.*;
import javax.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame ("Hello wo
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
```

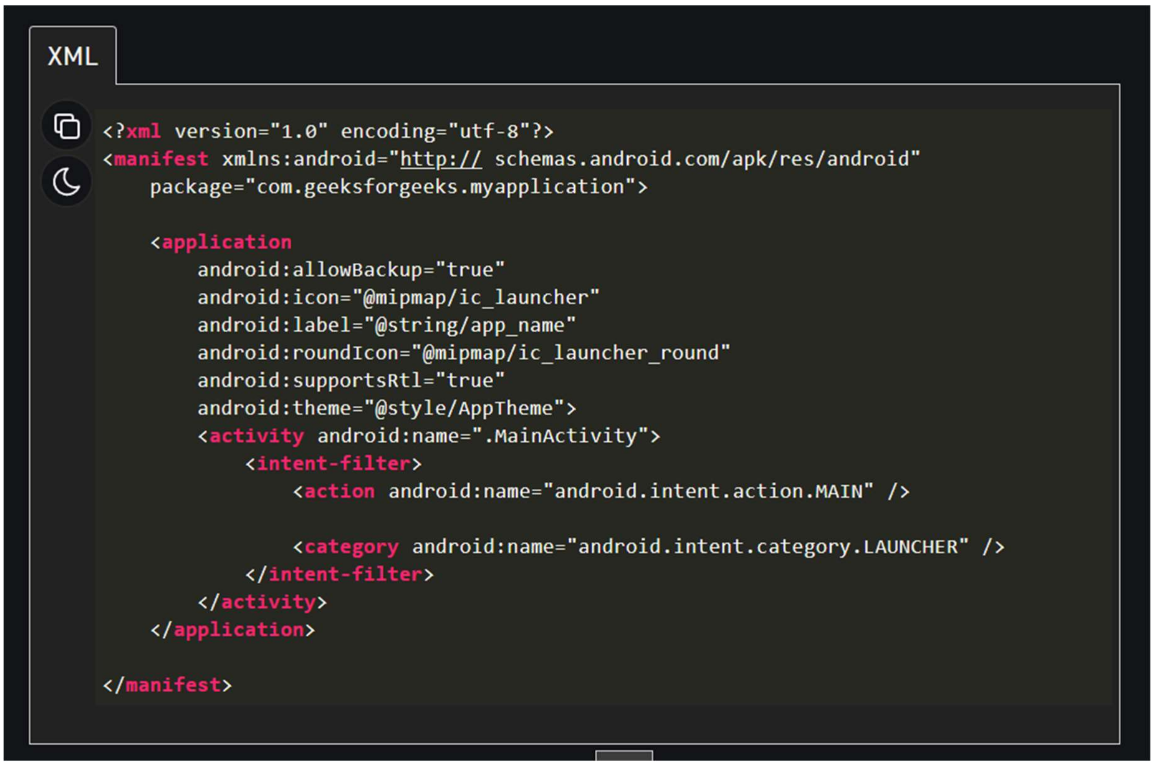
## Android Project folder Structure

The android project contains different types of app modules, source code files, and resource files. We will explore all the folders and files in the android app.

1. Manifests Folder
2. Java Folder
3. res (Resources) Folder
    - Drawable Folder
    - Layout Folder
    - Mipmap Folder
    - Values Folder
4. Gradle Scripts

### Manifests Folder

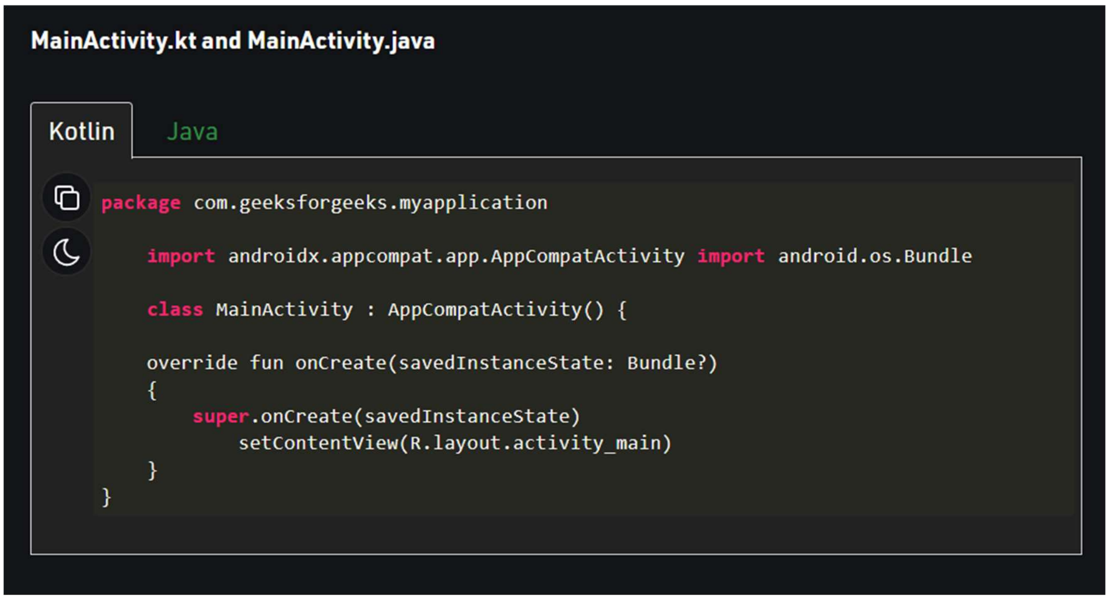
Manifests folder contains **AndroidManifest.xml** for creating our android application. This file contains information about our application such as the Android version, metadata, states package for Kotlin file, and other application components. It acts as an intermediator between android OS and our application.



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http:// schemas.android.com/apk/res/android"
    package="com.geeksforgeeks.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

## Java folder

The Java folder contains all the java and Kotlin source code (.java) files that we create during the app development, including other Test files. If we create any new project using Kotlin, by default the class file MainActivity.kt file will create automatically under the package name


MainActivity.kt and MainActivity.java

```
package com.geeksforgeeks.myapplication

    import androidx.appcompat.app.AppCompatActivity import android.os.Bundle

    class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
            setContentView(R.layout.activity_main)
    }
}
```

## Resource (res) folder

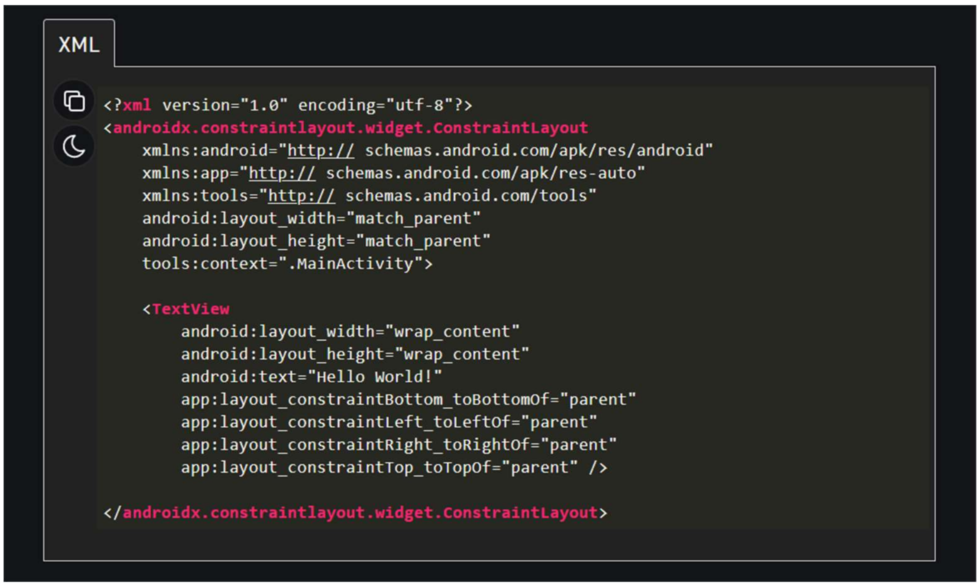
The resource folder is the most important folder because it contains all the non-code sources like images, XML layouts, and UI strings for our android application.

## res/drawable folder

It contains the different types of images used for the development of the application. We need to add all the images in a drawable folder for the application development.

## res/layout folder

The layout folder contains all XML layout files which we used to define the user interface of our application. It contains the **activity_main.xml** file.



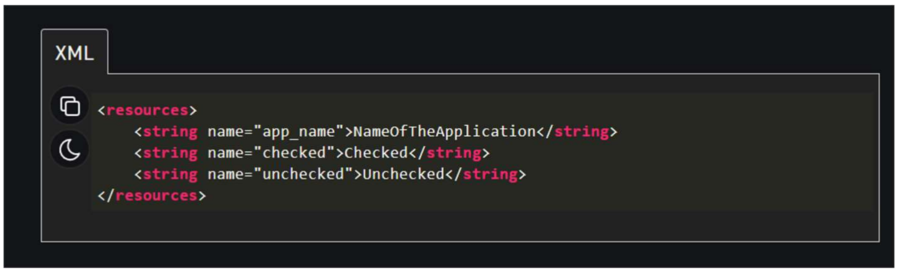
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http:// schemas.android.com/apk/res/android"
    xmlns:app="http:// schemas.android.com/apk/res-auto"
    xmlns:tools="http:// schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

## res/mipmap folder

This folder contains launcher.xml files to define icons that are used to show on the home screen. It contains different density types of icons depending upon the size of the device such as hdpi, mdpi, xhdpi.

## res/values folder

Values folder contains a number of XML files like strings, dimensions, colors, and style definitions. One of the most important files is the **strings.xml** file which contains the resources.



```xml
<resources>
    <string name="app_name">NameOfTheApplication</string>
    <string name="checked">Checked</string>
    <string name="unchecked">Unchecked</string>
</resources>
```

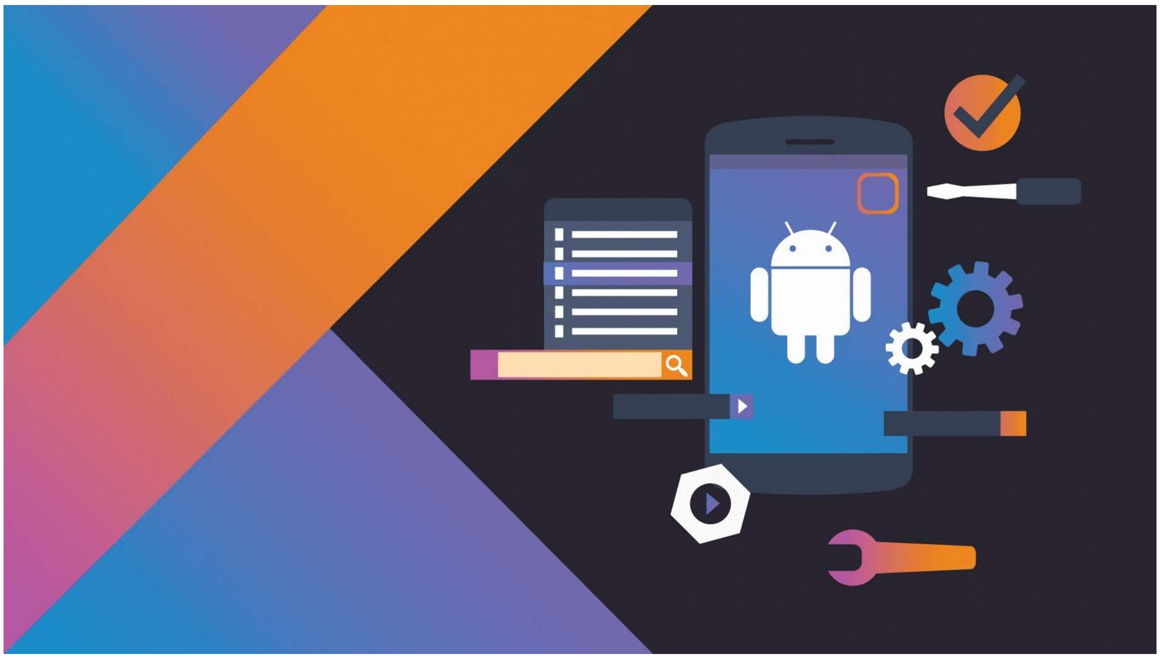
## Gradle Scripts folder

Gradle means automated build system and it contains a number of files that are used to define a build configuration that can be applied to all modules in our application. In build.gradle (Project) there are buildscripts and in build.gradle (Module) plugins and implementations are used to build configurations that can be applied to all our application modules.

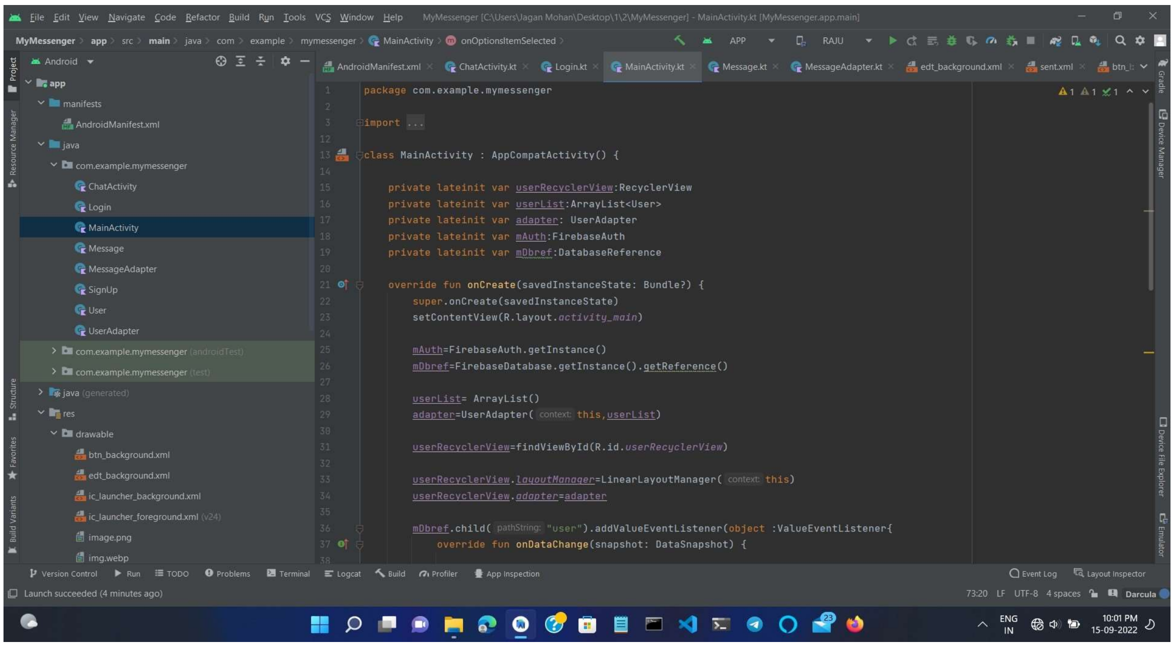
## Kotlin

Kotlin is a statically typed, general-purpose programming language developed by JetBrains, that has built world-class IDEs like IntelliJ IDEA, PhpStorm, Appcode, etc. It was first introduced by JetBrains in 2011 and is a new language for the JVM. Kotlin is an object-oriented language, and a “better language” than Java, but still be fully interoperable with Java code.

Kotlin is sponsored by Google, announced as one of the official languages for Android Development in 2017.



## Kotlin code in Android Studio



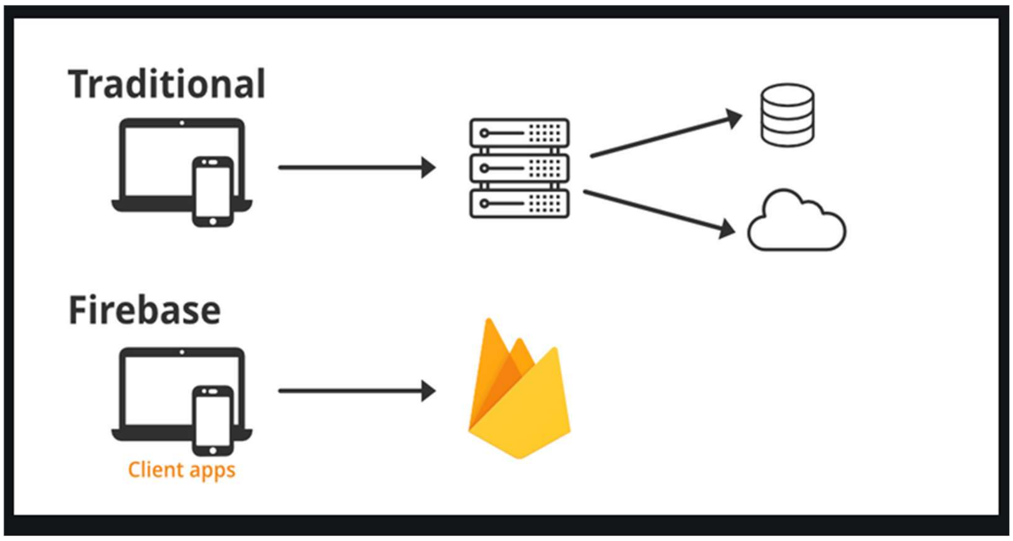
```
package com.example.mymessenger

import ...

class MainActivity : AppCompatActivity() {

    private lateinit var userRecyclerView:RecyclerView
    private lateinit var userList:ArrayList<User>
    private lateinit var adapter: UserAdapter
    private lateinit var mAuth:FirebaseAuth
    private lateinit var mDbref:DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        mAuth=FirebaseAuth.getInstance()
        mDbref=FirebaseDatabase.getInstance().getReference()

        userList= ArrayList()
        adapter=UserAdapter( context: this,userList)

        userRecyclerView=findViewById(R.id.userRecyclerView)

        userRecyclerView.layoutManager=LinearLayoutManager( context: this)
        userRecyclerView.adapter=adapter

        mDbref.child( pathString: "user").addValueEventListener(object :ValueEventListener{
            override fun onDataChange(snapshot: DataSnapshot) {
```

## XML

XML stands for Extensible Markup Language. XML is a markup language much like HTML used to describe data. It is derived from Standard Generalized Markup Language(SMGL). Basically, the XML tags are not predefined in XML. We need to implement and define the tags in XML. XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked.

## Firebase

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, ios, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

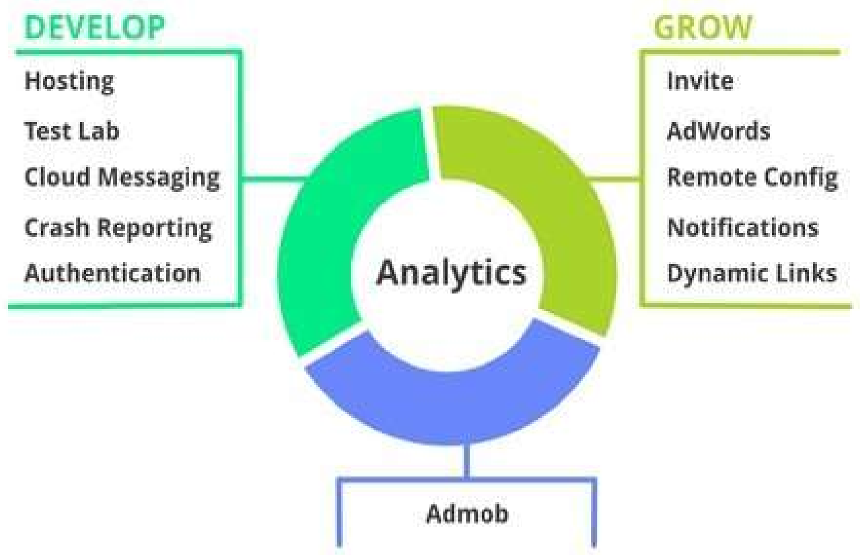


## Brief History of Firebase:

Firebase initially was an online chat service provider to various websites through API and ran with the name **Envolve**. It got popular as developers used it to exchange application data like a game state in real time across their users more than the chats. This resulted in the separation of the Envolve architecture and it's chat system. The Envolve architecture was further evolved by it's founders James Tamplin and Andrew Lee,to what modern day Firebase is in the year 2012.

## Features of Firebase:

Mainly there are 3 categories in which firebase provides its services.



## Connecting Firebase to Android Studio

### Step 1: Create a Firebase project

Before you can add Firebase to your Android app, you need to create a Firebase project to connect to your Android app. Visit Understand Firebase Projects to learn more about Firebase projects.

**Create a Firebase project**

### Step 2: Register your app with Firebase

To use Firebase in your Android app, you need to register your app with your Firebase project. Registering your app is often called "adding" your app to your project.

Go to the Firebase console.

 In the center of the project overview page, click the **Android** icon (plat_android) or **Add app** to launch the setup workflow.

 Enter your app's package name in the **Android package name** field.

What's a package name, and where do you find it?

 *(Optional)* Enter other app information: **App nickname** and **Debug signing certificate SHA-1**.

How are the *App nickname* and the *Debug signing certificate SHA-1* used within Firebase?

 Click **Register app**.

## Step 3: Add a Firebase configuration file

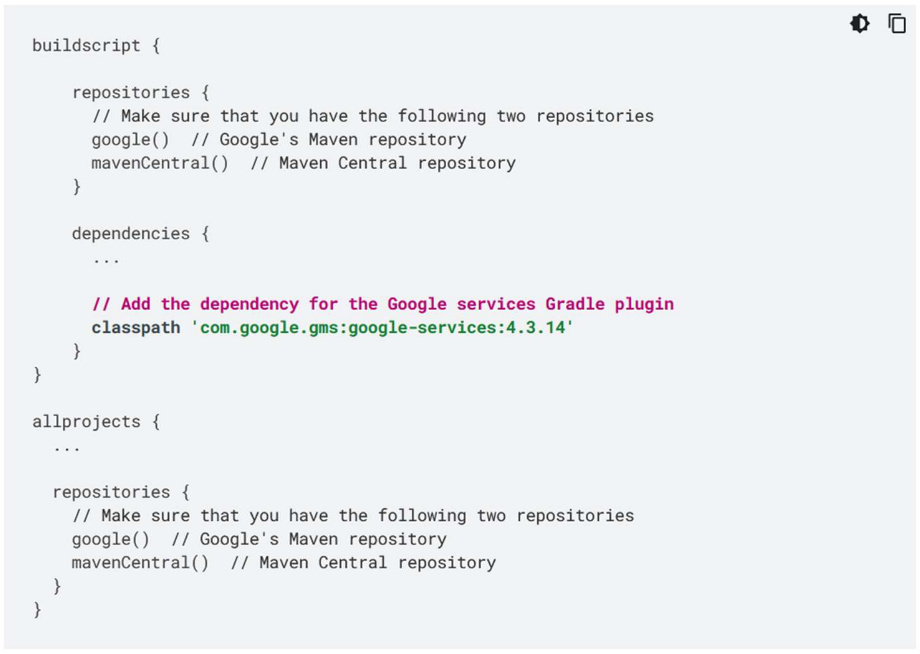
 Download and then add the Firebase Android configuration file (google-services.json) to your app:

1. Click **Download google-services.json** to obtain your Firebase Android config file.
2. Move your config file into the **module (app-level)** root directory of your app.

What do you need to know about this config file?

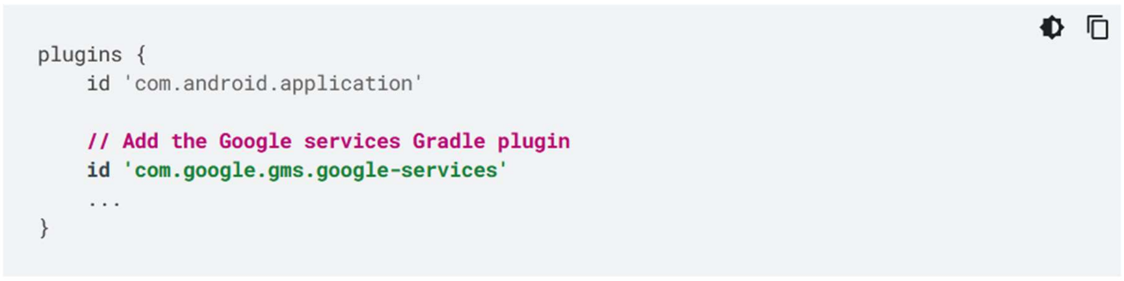
 To make the values in your google-services.json config file accessible to Firebase SDKs, you need the Google services Gradle plugin (google-services).

1. In your **root-level (project-level)** Gradle file (<project>/build.gradle), add the Google services plugin as a buildscript dependency:



```
buildscript {

    repositories {
      // Make sure that you have the following two repositories
      google()  // Google's Maven repository
      mavenCentral()  // Maven Central repository
    }

    dependencies {
      ...

      // Add the dependency for the Google services Gradle plugin
      classpath 'com.google.gms:google-services:4.3.14'
    }
}

allprojects {
  ...

  repositories {
    // Make sure that you have the following two repositories
    google()  // Google's Maven repository
    mavenCentral()  // Maven Central repository
  }
}
```

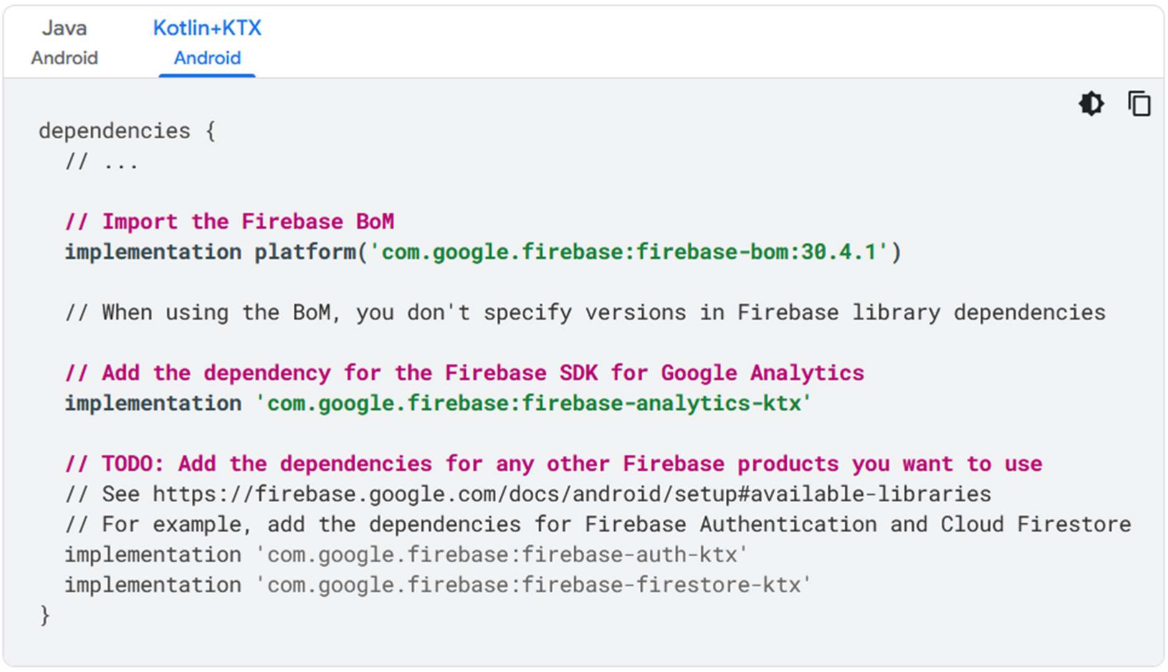
In your **module (app-level)** Gradle file (usually <project>/<app-module>/build.gradle), add the Google services plugin:



```
plugins {
    id 'com.android.application'

    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}
```

## Step 4: Add Firebase SDKs to your app

 In your **module (app-level) Gradle file** (usually <project>/<app-module>/build.gradle), add the dependencies for the Firebase products that you want to use in your app. We recommend using the Firebase Android BoM to control library versioning.



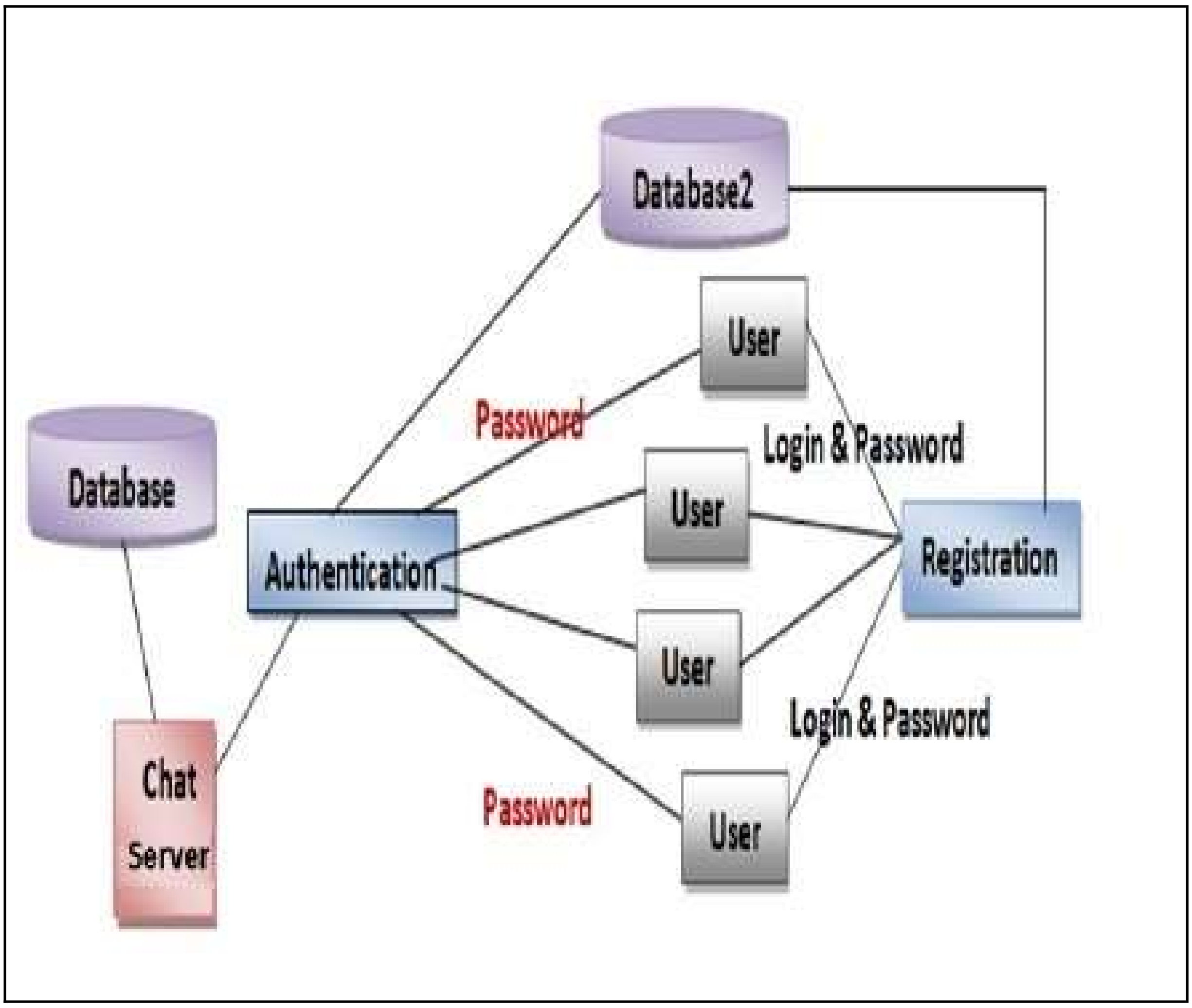
```
dependencies {
  // ...

  // Import the Firebase BoM
  implementation platform('com.google.firebase:firebase-bom:30.4.1')

  // When using the BoM, you don't specify versions in Firebase library dependencies

  // Add the dependency for the Firebase SDK for Google Analytics
  implementation 'com.google.firebase:firebase-analytics-ktx'

  // TODO: Add the dependencies for any other Firebase products you want to use
  // See https://firebase.google.com/docs/android/setup#available-libraries
  // For example, add the dependencies for Firebase Authentication and Cloud Firestore
  implementation 'com.google.firebase:firebase-auth-ktx'
  implementation 'com.google.firebase:firebase-firestore-ktx'
}
```

After adding the dependencies for the products you want to use, sync your Android project with Gradle files.

## System in Context:

The Converse provide the users to Chat with their Friends and Family and their Team members and also provides security to the Users. App Provides the Users names Who already Using this application. Users can Login With Email and Password. Authentication is Provided To the users.

## Context Diagram:



## System-wide Requirements(Received):

### Actors:

The system interacts with Two users. Each user has its own functions to access with the system. The functionalities of users are dependent on each other.

### Events:

Converse is a multiuser system which provides user to chat with each other for day to day operations.

**The most critical events are:**

1. Gets register first using the Name,Email address,Password.
2. Users login using the email and password and can select another user.
3. After select the user can chat with that user.
4. one user can select any other user to chat with.
5. User can Send message to others and Receive messages from another users 6. User Can Logout from the App.

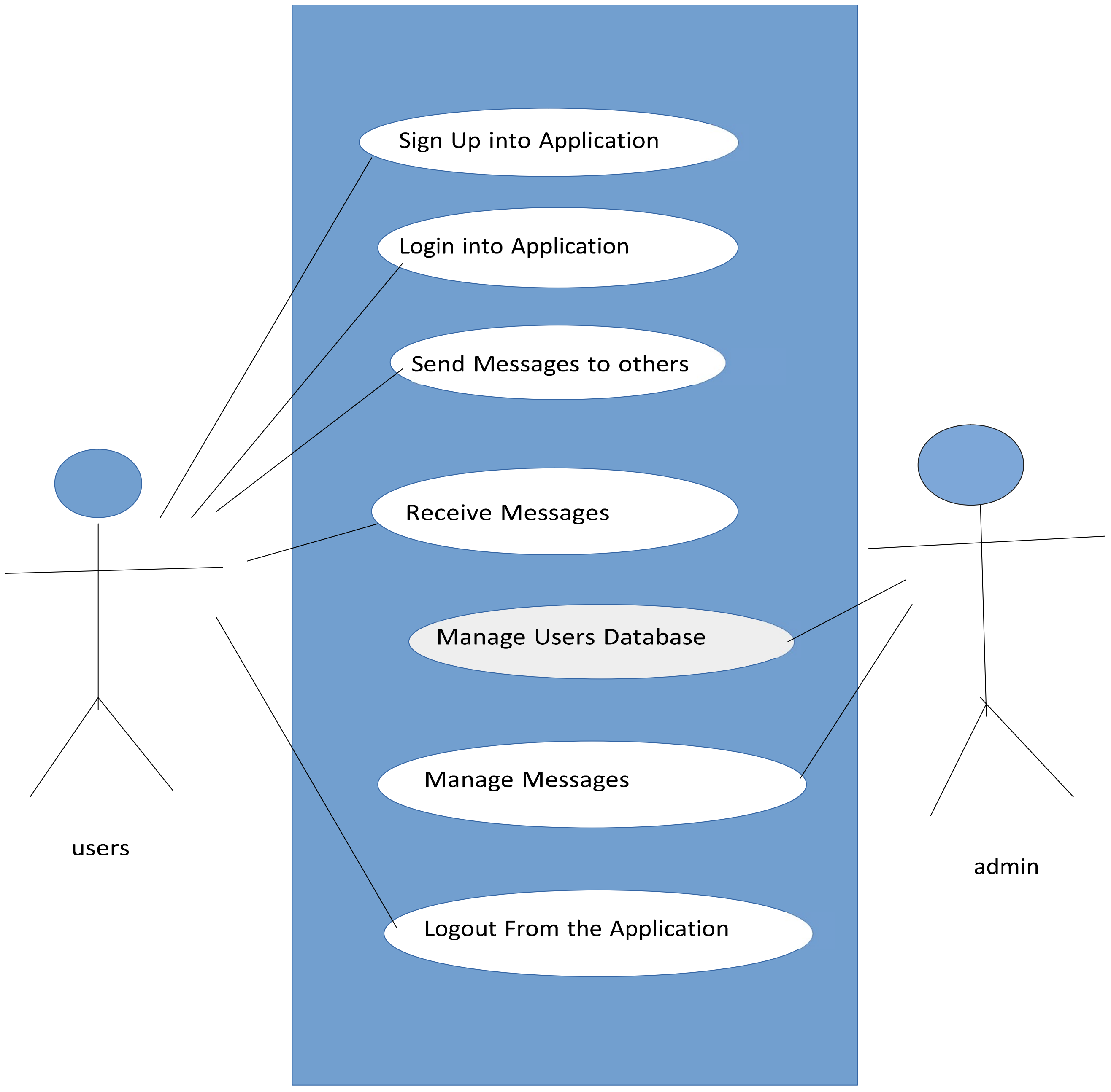
**The below table provides a set of user visible events that define the functionalities that are in Converse.**

| | Actor | Action | Object | Frequenc y | Arrival Pattern | Response |
|---|---|---|---|---|---|---|
| 1. | User | onclick | Signup | 1/day | Episodic | It asks Name, Email address,create password |
| 2. | User | Onclick | Login | 1/day | Episodic | It ask email address and password |
| 3. | User | select | Another User | 1/day | Episodic | It show All Users |

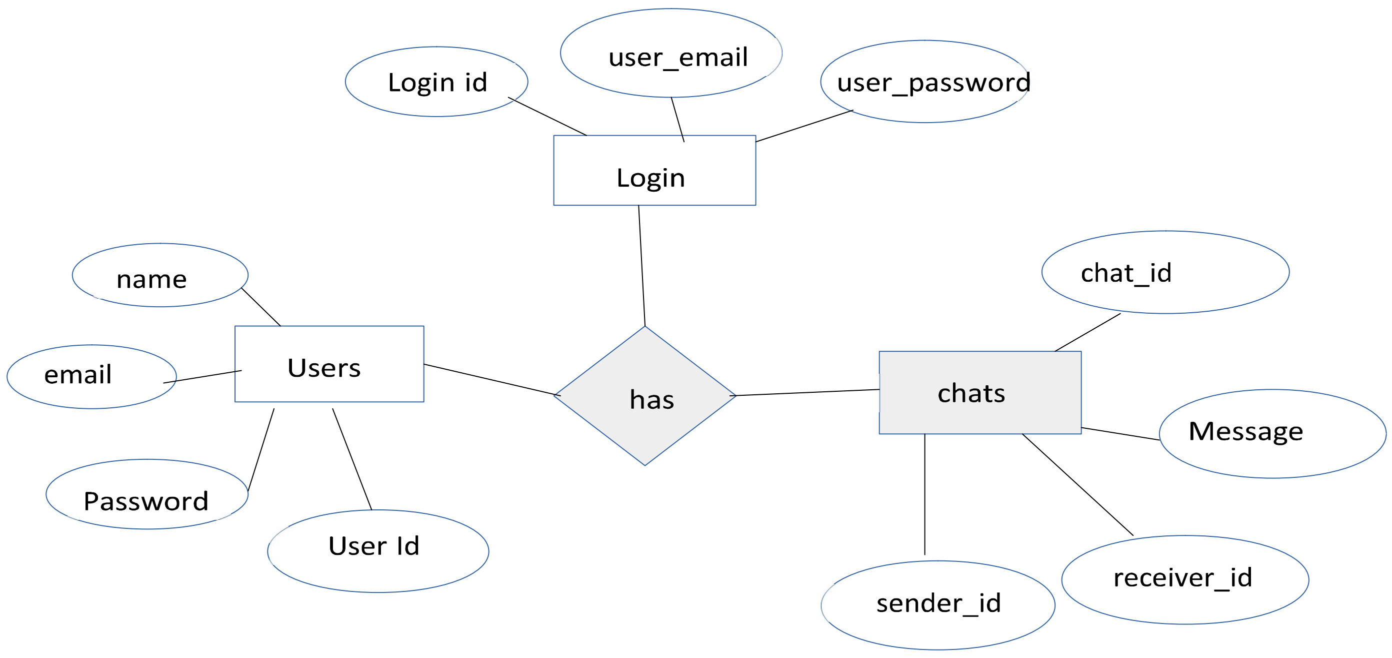
| | | | | | | |
|---|---|---|---|---|---|---|
| 4. | User | onclick | User | 1/day | Episodic | It show the Chat Screen |
| 5. | User | onclick | Input Message | 1/day | Episodic | It ask the Message to type And send to Another user |
| 6. | User | onclick | Send Button | 1/day | Episodic | User Entered Message Sent to the Intended user |
| 7. | User | onclick | Back button | 1/day | Episodic | It shows List of user |
| 8. | User | onclick | Another user | 1/day | Episodic | Another user Chat window opened |
| 9. | User | onclick | Home button | 1/day | Episodic | User will move to the Home Screen |
| 10. | User | onclick | Logout | 1/day | Episodic | After onclick logout user logout from page and reach to homeScreen. |

## Functional Requirements:

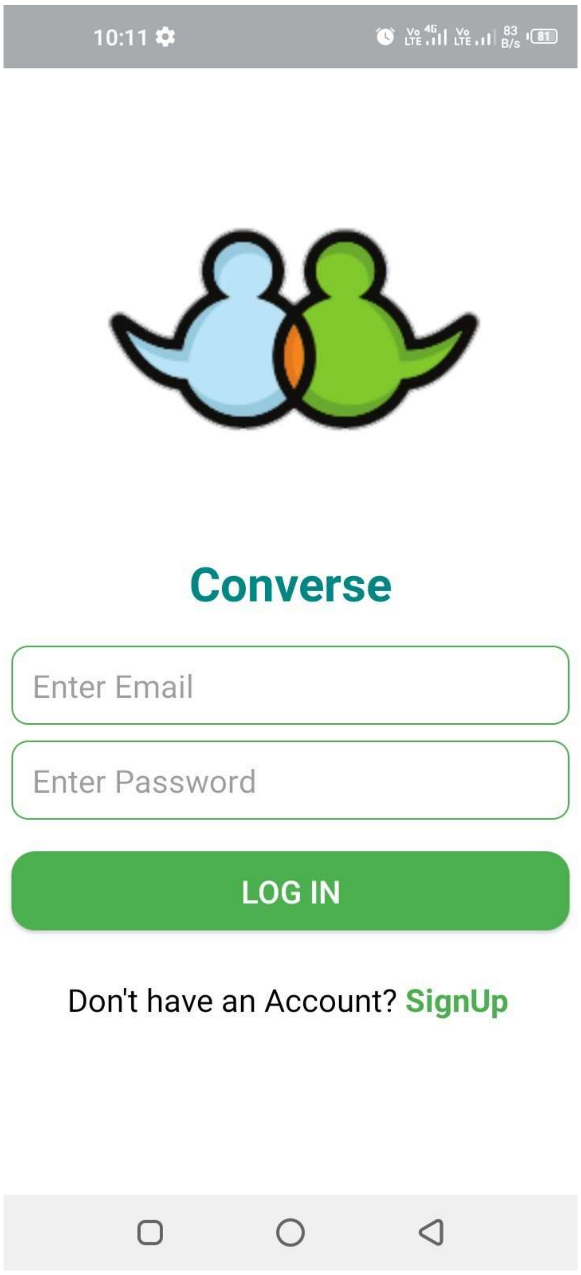
### Use case Diagram:



## ER Diagram:


Login id
user_email
user_password
Login
name
email
Users
Password
User Id
has
chat_id
chats
Message
sender_id
receiver_id

## UI Of Converse


10:11

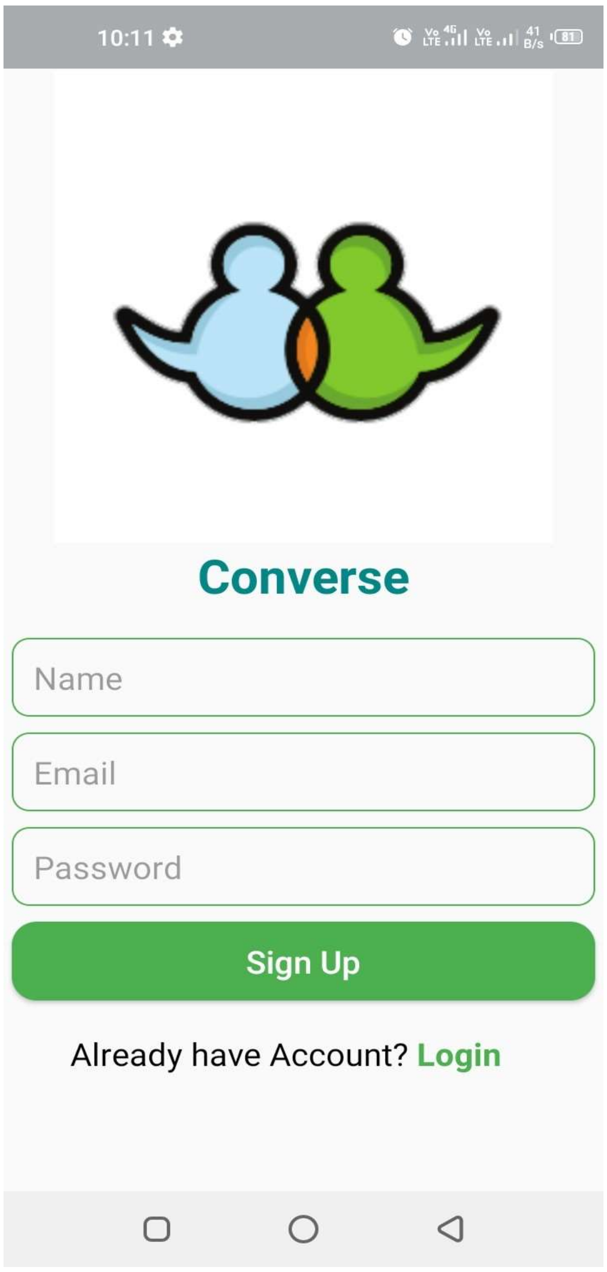
Converse

Enter Email

Enter Password

LOG IN

Don't have an Account? SignUp


10:11

Converse

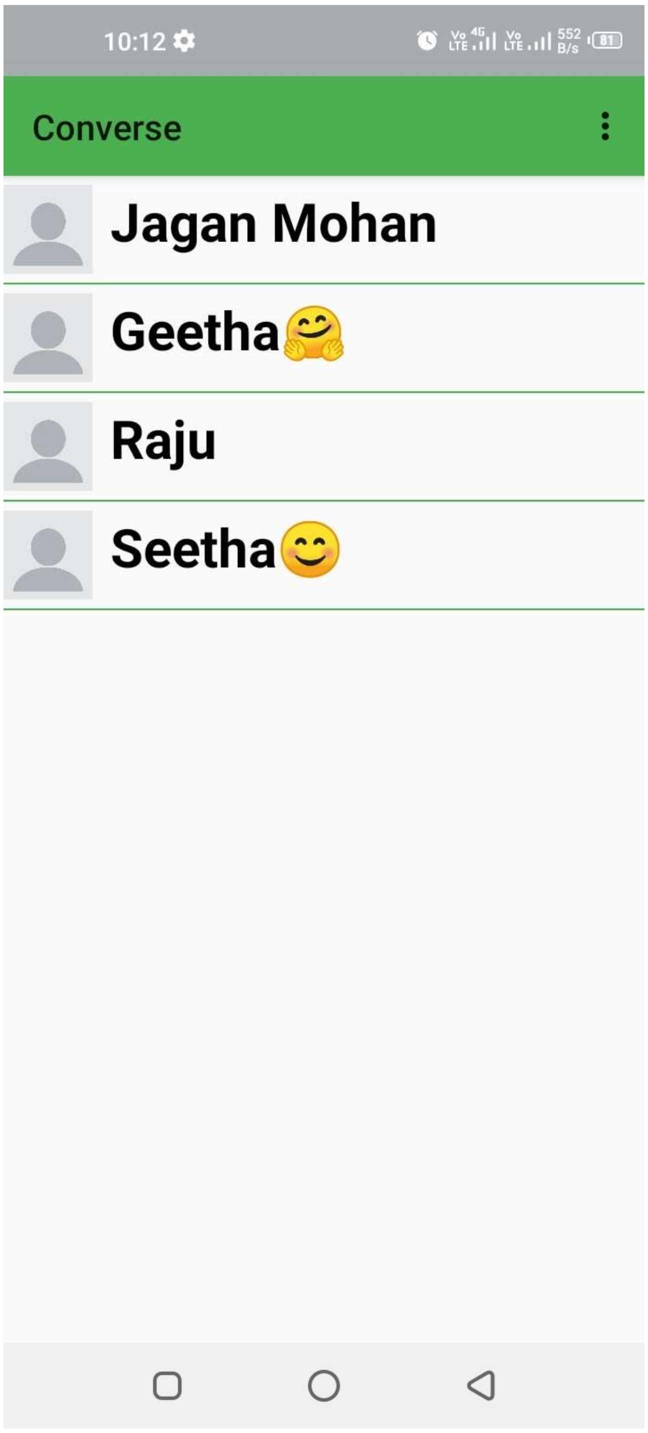
Name

Email

Password

Sign Up

Already have Account? Login


10:12

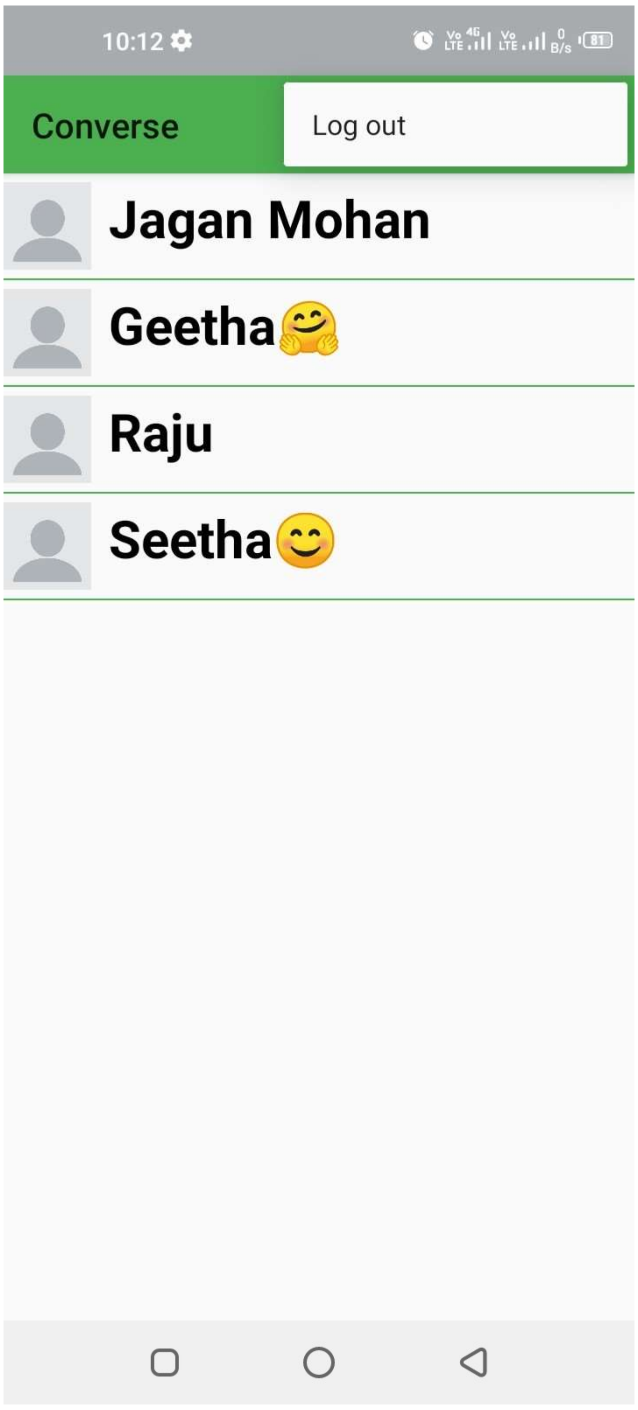
Converse

Jagan Mohan

Geetha🤗

Raju

Seetha😊


10:12

Converse

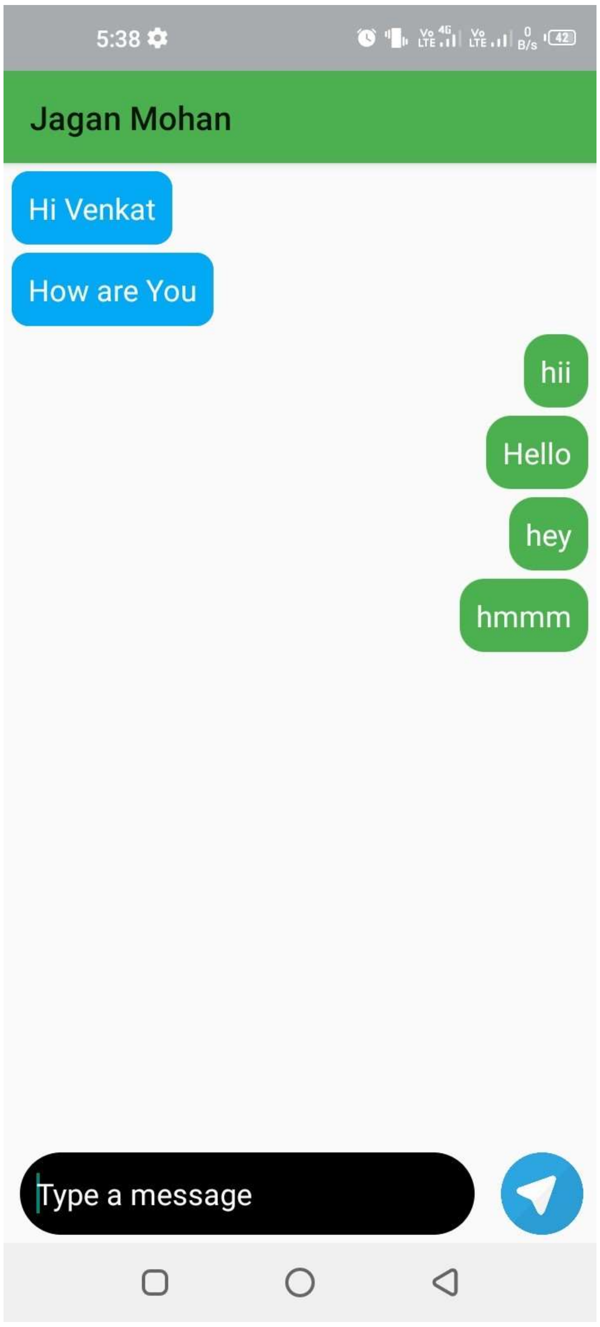
Log out

Jagan Mohan

Geetha🤗

Raju

Seetha😊



Jagan Mohan

Hi Venkat

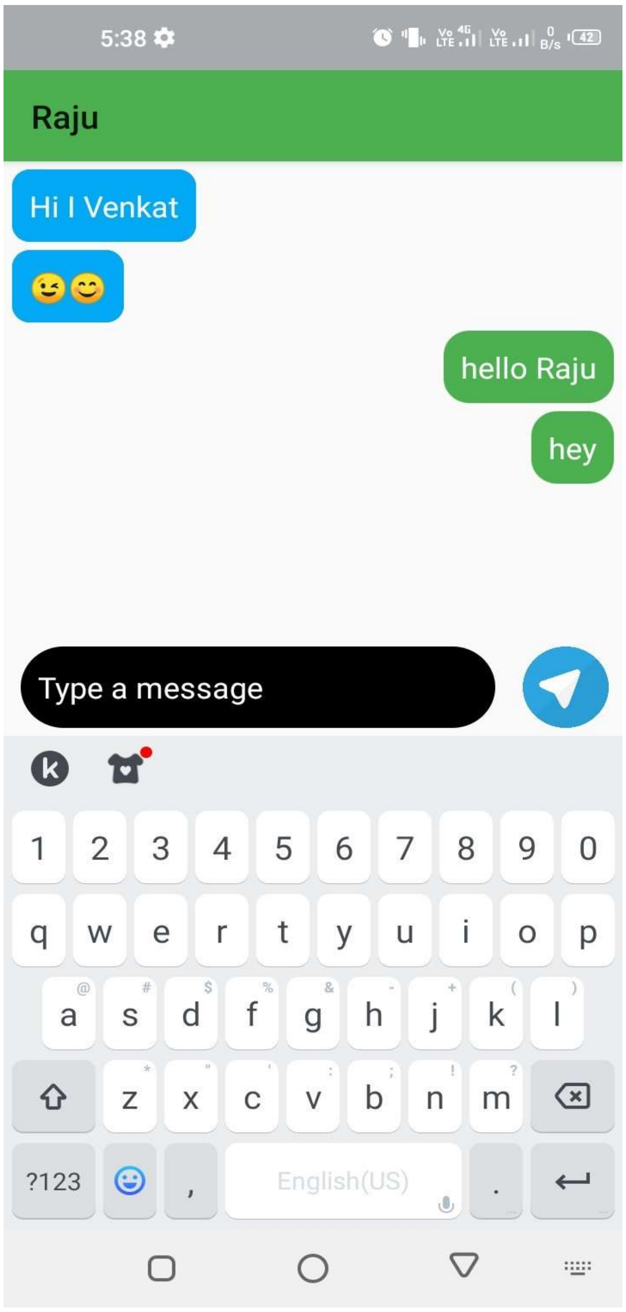
How are You

hii

Hello

hey

hmmm



Raju

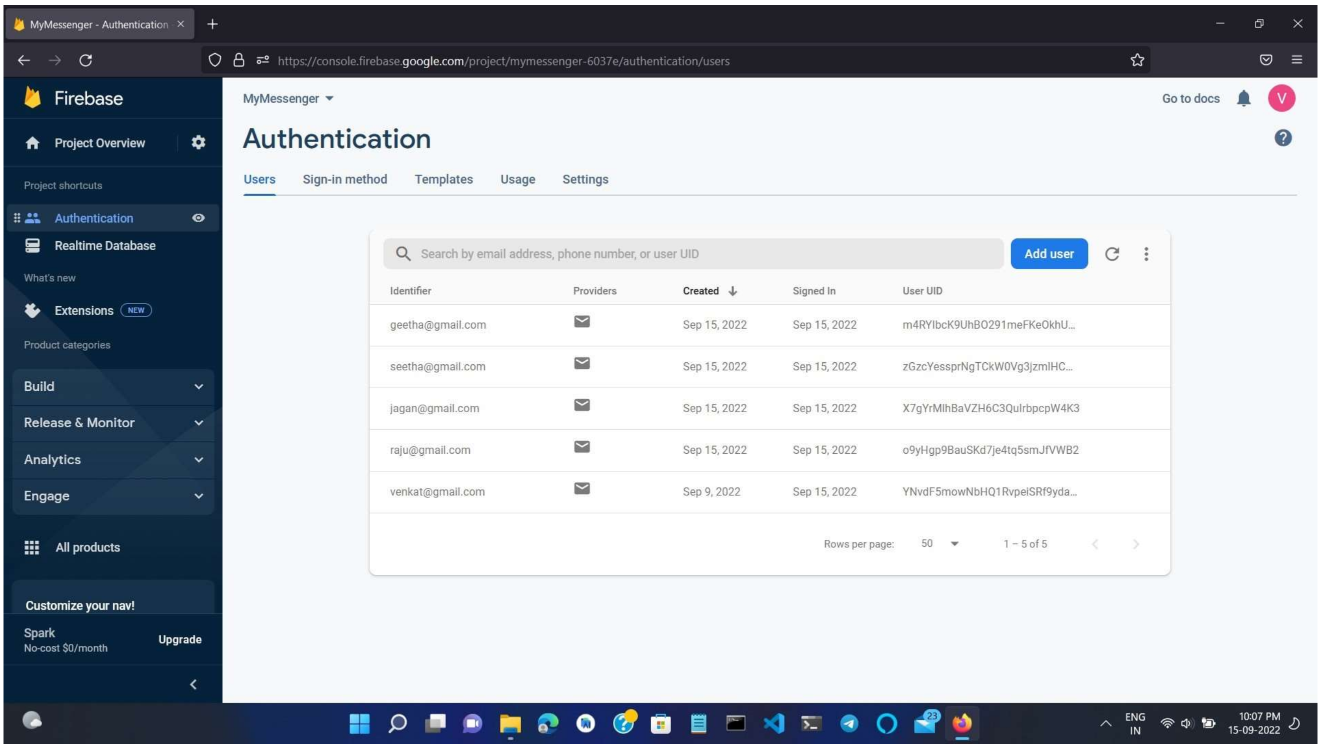
Hi I Venkat

😉😊

hello Raju

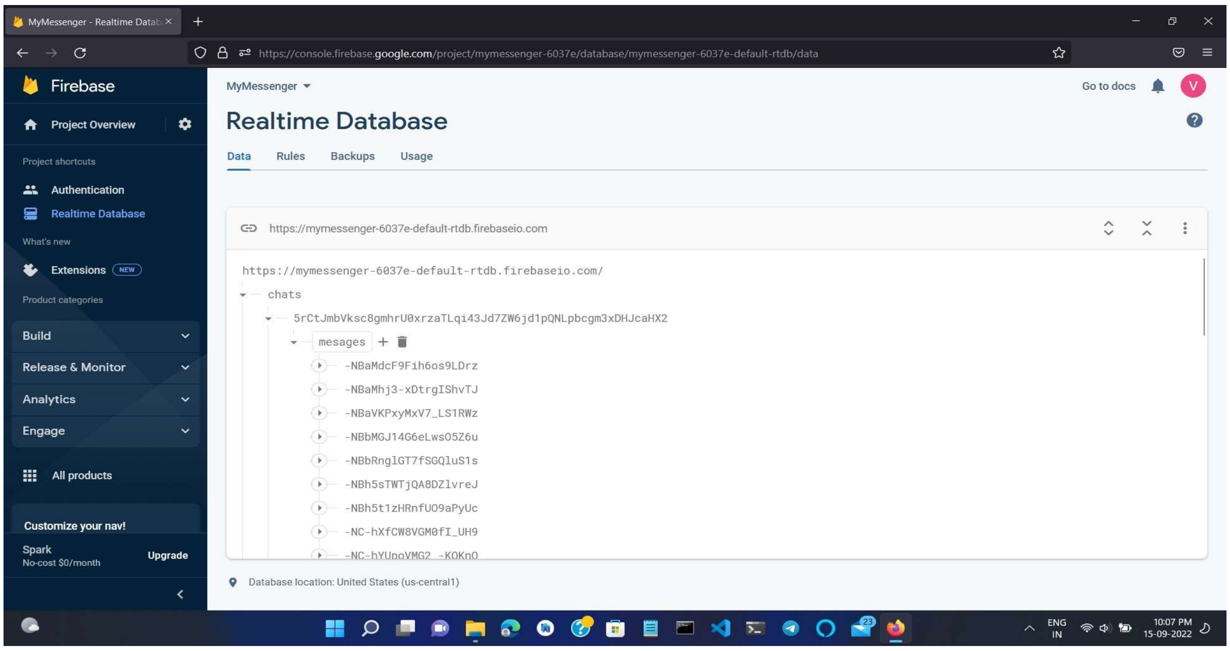
hey

## Connecting to Firebase



| Identifier | Providers | Created | Signed In | User UID |
|---|---|---|---|---|
| geetha@gmail.com | ✉ | Sep 15, 2022 | Sep 15, 2022 | m4RYIbcK9UhBO291meFKeOkhU... |
| seetha@gmail.com | ✉ | Sep 15, 2022 | Sep 15, 2022 | zGzcYessprNgTCkW0Vg3jzmIHC... |
| jagan@gmail.com | ✉ | Sep 15, 2022 | Sep 15, 2022 | X7gYrMlhBaVZH6C3QuIrbpcpW4K3 |
| raju@gmail.com | ✉ | Sep 15, 2022 | Sep 15, 2022 | o9yHgp9BauSKd7je4tq5smJfVWB2 |
| venkat@gmail.com | ✉ | Sep 9, 2022 | Sep 15, 2022 | YNvdF5mowNbHQ1RvpeiSRf9yda... |

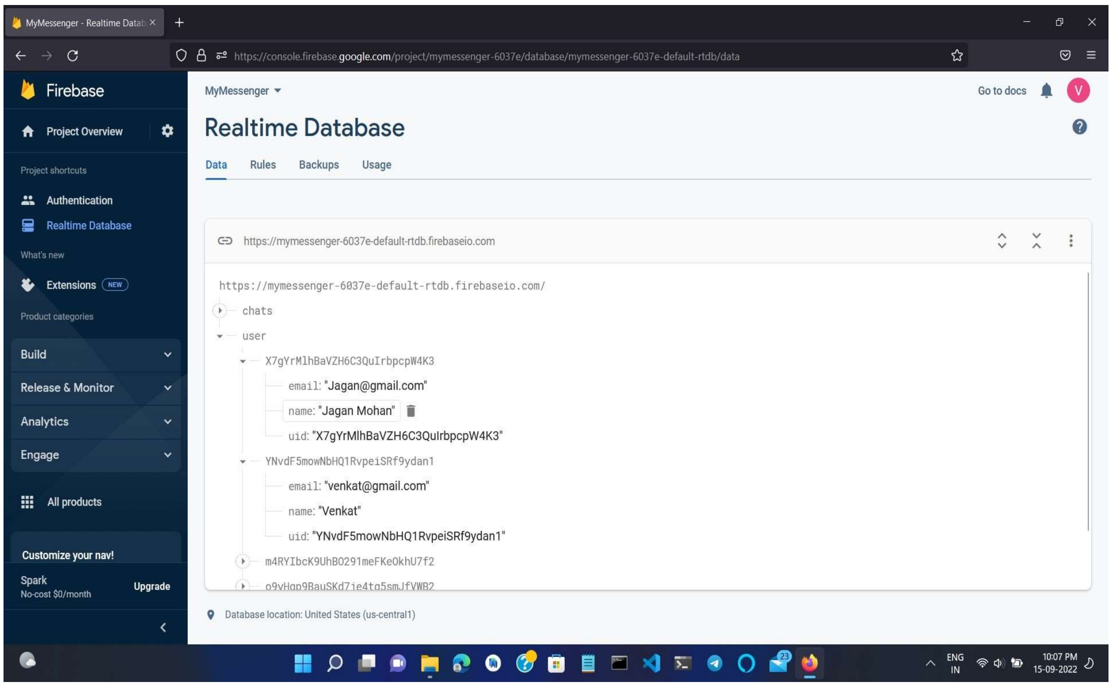
Rows per page: 50 1 – 5 of 5



https://mymessenger-6037e-default-rtdb.firebaseio.com

```
https://mymessenger-6037e-default-rtdb.firebaseio.com/
  chats
    5rCtJmbVksc8gmhrU0xrzaTLqi43Jd7ZW6jd1pQNLpbcgm3xDHJcaHX2
      mesages
        -NBaMdcF9Fih6os9LDrz
        -NBaMhj3-xDtrgIShvTJ
        -NBaVKPxyMxV7_LS1RWz
        -NBbMGJ14G6eLws05Z6u
        -NBbRnglGT7fSGQluS1s
        -NBh5sTWTjQA8DZlvreJ
        -NBh5t1zHRnfUO9aPyUc
        -NC-hXfCW8VGM0fI_UH9
        -NC-hYUpoVMG2_-KOKnO
```

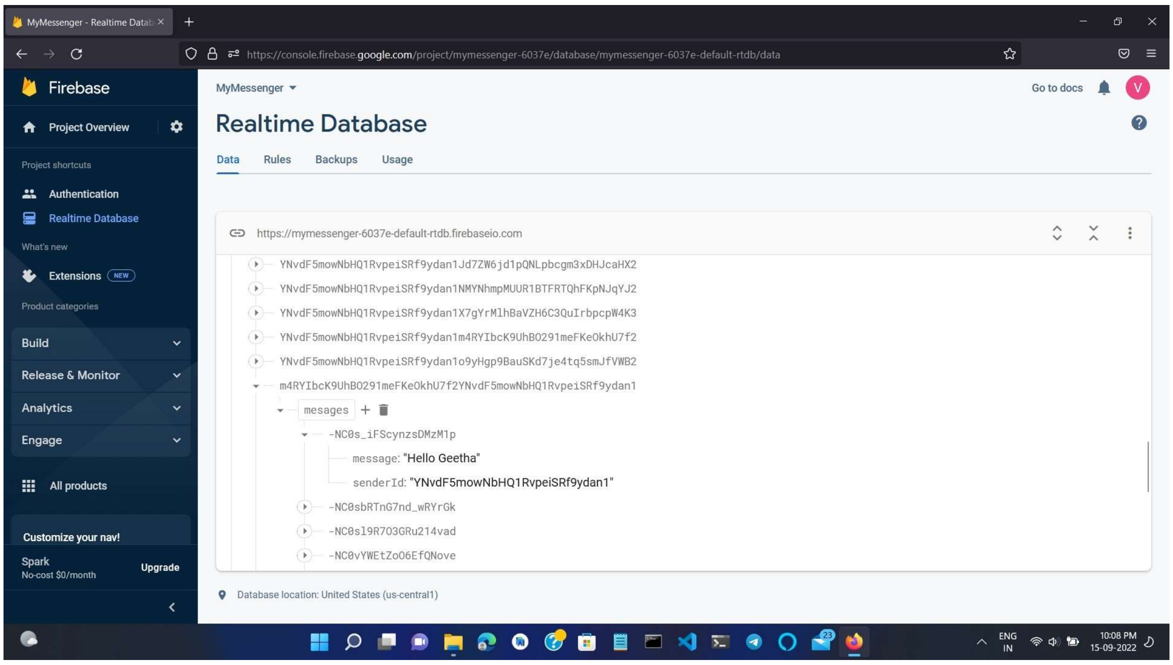
Database location: United States (us-central1)



https://mymessenger-6037e-default-rtdb.firebaseio.com

```
https://mymessenger-6037e-default-rtdb.firebaseio.com/
  chats
  user
    X7gYrMlhBaVZH6C3QuIrbpcpW4K3
      email: "Jagan@gmail.com"
      name: "Jagan Mohan"
      uid: "X7gYrMlhBaVZH6C3QuIrbpcpW4K3"
    YNvdF5mowNbHQ1RvpeiSRf9ydan1
      email: "venkat@gmail.com"
      name: "Venkat"
      uid: "YNvdF5mowNbHQ1RvpeiSRf9ydan1"
    m4RYIbcK9UhB0291meFKeOkhU7f2
    o9yHgp9BauSKd7je4tq5smJfVWB2
```

Database location: United States (us-central1)



https://mymessenger-6037e-default-rtdb.firebaseio.com

```
  YNvdF5mowNbHQ1RvpeiSRf9ydan1Jd7ZW6jd1pQNLpbcgm3xDHJcaHX2
  YNvdF5mowNbHQ1RvpeiSRf9ydan1NMYNhmpMUUR1BTFRTQhFKpNJqYJ2
  YNvdF5mowNbHQ1RvpeiSRf9ydan1X7gYrMlhBaVZH6C3QuIrbpcpW4K3
  YNvdF5mowNbHQ1RvpeiSRf9ydan1m4RYIbcK9UhB0291meFKeOkhU7f2
  YNvdF5mowNbHQ1RvpeiSRf9ydan1o9yHgp9BauSKd7je4tq5smJfVWB2
  m4RYIbcK9UhB0291meFKeOkhU7f2YNvdF5mowNbHQ1RvpeiSRf9ydan1
    mesages
      -NC0s_iFScynzsDMzM1p
        message: "Hello Geetha"
        senderId: "YNvdF5mowNbHQ1RvpeiSRf9ydan1"
      -NC0sbRTnG7nd_wRYrGk
      -NC0sl9R703GRu214vad
      -NC0vYWEtZo06EfQNove
```

Database location: United States (us-central1)

## Conclusion

The Converse Application Provides users to communicate with others by which they can improve their communication skills and they can also share information. This App can be used any person who want to chat with their friends and family  and others.

## Future Enhancement

- User can Upload their profile pic so that other users can easily identify them
- Sharing files option will be implemented
- Groupchat option will be enabled and some more functionalities like who are in online.
- Audio video calling will be implemented.

## References

https://www.geeksforgeeks.org/guide-to-install-and-set-up-android-studio/

https://firebase.google.com/docs/android/setup

https://developer.android.com/kotlin/first