

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)



Project Report

On

Message Application for Mobiles

“PingMe Message Application”

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

Submitted by:

Shriya Jaiswal

(0901IT193D04)

Project – 160801

Faculty Mentor:

Prof. Vikas Sejwar,

Asst. Professor

(Department of Information Technology)

Submitted to:

DEPARTMENT OF INFORMATION TECHNOLOGY

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR – 474005

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

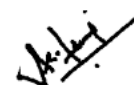
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Shriya Jaiswal** (0901IT193D04) has submitted the project report titled **Message Application for Mobiles “PingMe Message Application”** under the mentorship of Prof. **Vikas Sejwar**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Information Technology** from Madhav Institute of Technology and Science, Gwalior.



Prof. Vikas Sejwar
Asst. Professor
Department of IT



Dr. Akhilesh Tiwari
Professor and Head,
Department of IT

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Information Technology at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of Prof. **Vikas Sejwar**, Asst. Professor, Department of Information Technology.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Date: 26/05/2022

Place: Gwalior



Shriya Jaiswal

0901IT193D04

IV Year,

Information Technology

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Information Technology**, for allowing me to explore this project. I humbly thank **Dr. Akhilesh Tiwari**, Professor and Head, Department of Information Technology, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of Prof. **Vikas Sejwar**, Asst. Professor, Department of Information Technology for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.



Shriya Jaiswal

0901It193D04

IV Year,

Information Technology

TABLE OF CONTENTS

Abstract

Chapter 1: Introduction

- 1.1 Purpose
- 1.2 Requirement
 - 1.2.1 System Requirement
 - 1.2.1.1 Hardware Requirement
 - 1.2.2.2 Software Requirement
 - 1.2.2 Functional Requirement
 - 1.2.3 User Requirement
 - 1.2.4 Environment Requirement
- 1.3 Goals
- 1.4 Project Features
- 1.5 Future Scope

Chapter 2: Literature Review

- 2.1 Literature Review
- 2.2 Latest Technology

Chapter 3: Preliminary design

- 3.1 Technology
- 3.2 Tools
- 3.3 Flow of Start Application
- 3.4 Implementation
 - 3.4.1 Firebase Authentication
 - 3.4.2 Firebase Database
 - 3.4.3 Messages Implementation

Chapter 4: Final Analysis and Design

4.1 Graphical User Interface (GUI) Layout

4.1.1 Launch Page

4.1.2 Sign UP Page

4.1.3 Sign In

4.1.3.1 Email Validation

4.1.3.2 Password Validation

4.2.4 User view Layout

4.2.5 Chat Layout

4.2.6.1 Text field validation

4.2.6 Menu Layout

4.2.6.1 Settings

4.2.6.1.1 Update profile

4.2.6.2 Group Chatting

4.2.6.3 Logout

4.3 Future Scope

Conclusion

References

ABSTRACT

Communication plays a very important role in the real world. People have been communicating with each other through various apps/platforms or mediums. Initially, people faced problems with communication. Letters or other resources were used for people to communicate with each other. And since these mediums take a long time to convey the message. So in this digital world there are various applications and mediums available to communicate with each other. Where online communication is the most popular medium which allows the user to communicate with each other in less time without any delay. And users can pass their messages from one end to the other. So in this project I am presenting an application which is based on Android operating system. To develop this project I am using Android Studio as a tool and the programming language is Java. This project is tested on an Android Emulator which is a tool that allows developers to easily test an application without having to install the application on a real device. In this application user can pass messages faster with minimum delay or no delay. I am using firebase database to store user and user related data. Firebase is a platform that provides real time database cloud storage and other services for building applications.

Keyword: Communication, Online messaging, group messaging, firebase real time database, android application.

Chapter 1

Introduction

1.1 Purpose

The purpose of this project is to develop a mobile application that utilizes a message feature where the user can communicate with each other and pass the message over the networks. The main overall goal of this app is to achieve something that positively contributes towards the safety of the person using it.

This project is for developing a mobile application which provides a messaging facility where users can communicate with each other and send messages over the network. The main goal of this application is to provide a platform or medium where a user can communicate his/her messages to another user. Android based "PingMe" messaging application provides a lot of convenience. Where the user can register themselves or create an account in the application, send or receive a text message with the other recipient user through the internet. The system needs to connect both devices via the internet. In this application, Firebase database is used as a backbone to store user information.

1.2 Requirement

1.2.1 System Requirement

1.2.1.1 Hardware Requirement

- Laptop/Desktop.
- Windows OS
- Minimum 4GB RAM and More.
- 1TB HDD

1.2.1.2 Software Requirement

- Android Studio.
- Firebase Console
- Installation of java
- Installation of Android studio.

Step 1: Install "Android Studio IDE"

Reference:

"Install Android Studio" @ <https://developer.android.com/studio/install>

(For Windows)

1. Check that environment variable JAVA_HOME is set to the JDK installation directory via command "set JAVA_HOME". Otherwise, Follow the steps [HERE](#).
2. Check the system requirements for Android Studio/SDK @ <https://developer.android.com/studio#Requirements> e.g., For Windows 10, 8GB of RAM, 8GB of disk space, and 1280x800 minimum screen resolution. Take note that you should have enough space on C drive. Insufficient space on C drive will take you many days.
3. Go to "Android Studio" under "Android Developers" @ <https://developer.android.com/studio> ⇒ Click "Download Android Studio" (Android Studio Bumblebee 2021.x.x for Windows 64-bit (872MiB)) to download the executable installer "android-studio-2021.x.x.xx-windows.exe".

4. Run the downloaded installer ⇒ you may watch a short video @ <https://developer.android.com/studio/install>.
 - a. In "Choose Components", select "Android Studio" and "Android Virtual Device" (space required: 2.7GB).
 - b. In "Configuration Settings Install Location", accept the default "C:\Program Files\Android\Android Studio".
 - c. In "Choose Start Menu Folder", accept the default ⇒ Install.
 - d. Launch Android Studio. Continue to Step 2.

by default, the "Android Studio IDE" will be installed in "C:\Program Files\Android\Android Studio", and the "Android SDK" in "c:\Users*username*\AppData\Local\Android\Sdk".

1.2.2 Functional Requirement

➤ App Navigation

- a) The application should make impeccable transitions between the users activities and application layout.
- b) Different layout pages should be clearly defined for the users.
- c) Features should be clearly visible and working to the users.
- d) Available options on each layout page should be clearly visible and working to the users.

➤ Phone Authentication with Database Connection

- a) The Database should be continuously available for new user registration or Sign-Up.
- b) The database should be updated on new user registration.

➤ Profile

- a) The user have to put their email address and create a personalized password to register into the app.
- b) The user may be able to use an alternative login i.e. login with Google accounts and Facebook accounts.

➤ Messaging

- a) The user will be able to pass the Messages to the other user.
- b) The user may be able to select a group or individual Messaging.
- c) User must send a message if user is connected to a specified internet connection.
- d) Message must include the user name and Last message in a sent message.

➤ Update Profile

- a) The user shall be able to update their profile like change the profile picture, user name, and add bio.

➤ Logout Profile

- a) The User can be logout their profile

1.2.3 User Requirement

- A user will be required to have connected with the internet on their smartphone.
- A new user will need to register or use the login method provided to access to the key features of the app.
- An existing user will be able to login using the email password or the alternate login method provided in the sign in page.

- The user must be able to connect to at least one Wi-Fi network for the messaging feature to work.
- A user may be able to select individuals or group to receive or send a Message.

1.2.4 Environmental Requirement

- A phone or device should have to be connected with the internet or WI-FI network.
- Internet or Wi-Fi network signal capability must be at a level where a device can make an authenticated connection.
- The connection must be held long enough for the application to send the message.

1.3 Goals

The basic goal is to create an application that can be used to create a connection between two or more users so that they can communicate easily without any delay or very little lag and create chat rooms according to the users and related to store data in one location. To implement a real time messaging application that can allow the user to communicate instantly in the form of messages. Implementing Google Firebase cloud and real time database to store data.

1.4 Project Features

- Create account
- Login
- Email & password validation field
- Messaging
- Text validation field
- Group messaging
- Update Profile
- Logout

1.5 Future Scope

To experience the new modern paradigm shifts in technology humans will need to be at one with the technologies they create and the ability to interface in real time. AS the basic idea behind our app is “PingMe” message Application which in future can be updated to more features such as: -

1. Group Calls
2. Voice Calls
3. Video Calls
4. Voice messages
5. Send Emoji’s

Chapter 2

Literature Review

Communication plays a very important role in the real world. People have been communicating with each other through various apps/platforms or mediums. Initially, people faced problems with communication. Letters or other resources were used for people to communicate with each other. And since these mediums take a long time to convey the message. So in this digital world there are various applications and mediums available to communicate with each other. Where online communication is the most popular medium which allows the user to communicate with each other in less time without any delay. And users can pass their messages from one end to the other.

Over the years, mobile learning and messaging are one of the fastest growing technologies which are extensively used by numerous individuals, companies and educational institutions. Messaging has been exploited as a foundation for education institutions to encourage forthcoming events and notify students with news. Educational institutions like colleges and universities have discovered the ability of messaging tools to be useful tools within collaborative learning environment, and the fact that the tools allow learning to occur anytime and anywhere.

Latest Technology:

Android is an operation system for mobiles that was developed by Google. This operating system allows the application to be used on the mobile. Android user can develop mobile application and it can be sold through Android application store like Play Store.

Firestore

In the real world the communication plays a very important role People have been communicating with each other through various platforms or mediums. Firestore is a platform that provides real time database cloud storage and other services for building applications. Nowadays, there are many types of mediums or databases available for messaging applications which provide real-time services, storage, and quick response time and interact quickly without any delay. Among them, there exists a very famous database which was developed by Google, which is Firestore Database.

Firestore is a NOSQL database where the data is stored in a key value manner. This allows the database user to store and retrieve the data from the database. If we want to use firestore on our android application then the version of android studio and firestore dependency should be compatible and added in the application. Firestore provides a various kind of services such as:

Firestore Authentication: Firestore Authentication facilitates databases or back-end services. It provides an easy-to-operate SDK and fully featured user interface library for users to authenticate in the application. Firestore supports authentication using email passwords, phone numbers and Google, Facebook and Twitter etc.

Firestore cloud: Firestore Cloud Messaging is a multi-platform cloud-based solution for messaging and notifications for android, IOS and web applications, we can utilize it for free.

Real-time database: It is a cloud hosted NoSQL database. Firestore Real-time Database is a database that is hosted on the cloud. Data is stored in JSON format and synchronized in real time for each connected user's device. In addition to authentication, Firestore also provides a service for cloud service and real time database crash reporting.

This paper deals with software applications for establishing a real time communication between users. Message applications, which are based on multiuser communication systems where users can exchange messages with other users. User can create chat group as per requirement or can also exchange message individually.

Chapter 3

Preliminary Design

3.1 Technology

Java: Java is a popular programming language, created in 1995. It is owned by Oracle, and more than **3 billion** devices run Java. It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!



Need to install Java Dev. Kit (JDK) latest or 8 version to write Java (Android) programs

- Don't install Java Runtime Env. (JRE); JDK is different!
- Newer versions of JDK can cause issues with Android

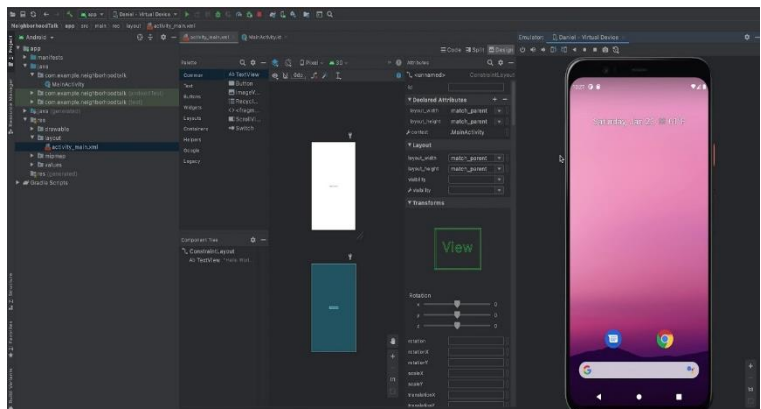
• Can download JDK (free): <https://adoptopenjdk.net/> – Oracle's JDK (<http://java.oracle.com>) free for dev. only; payment for commercial use

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful

3.2 Tools

Android Studio: Android Studio is the official IDE for android platform development and is freely available under the apache license. It is designed specifically for Android development and I am using it to develop the app, as thus far the app is exclusive to android.



Firestore: Google Firestore is a mobile platform for helping to develop app, and increase your user base. It's made up of features that can be mixed and matched to fit the needs of the user. Most of the features on Firestore are free to use. I will be using Firestore for app sign in options and authentication and to maintain a user database.



3.3 Flow of Start or Launch the Application

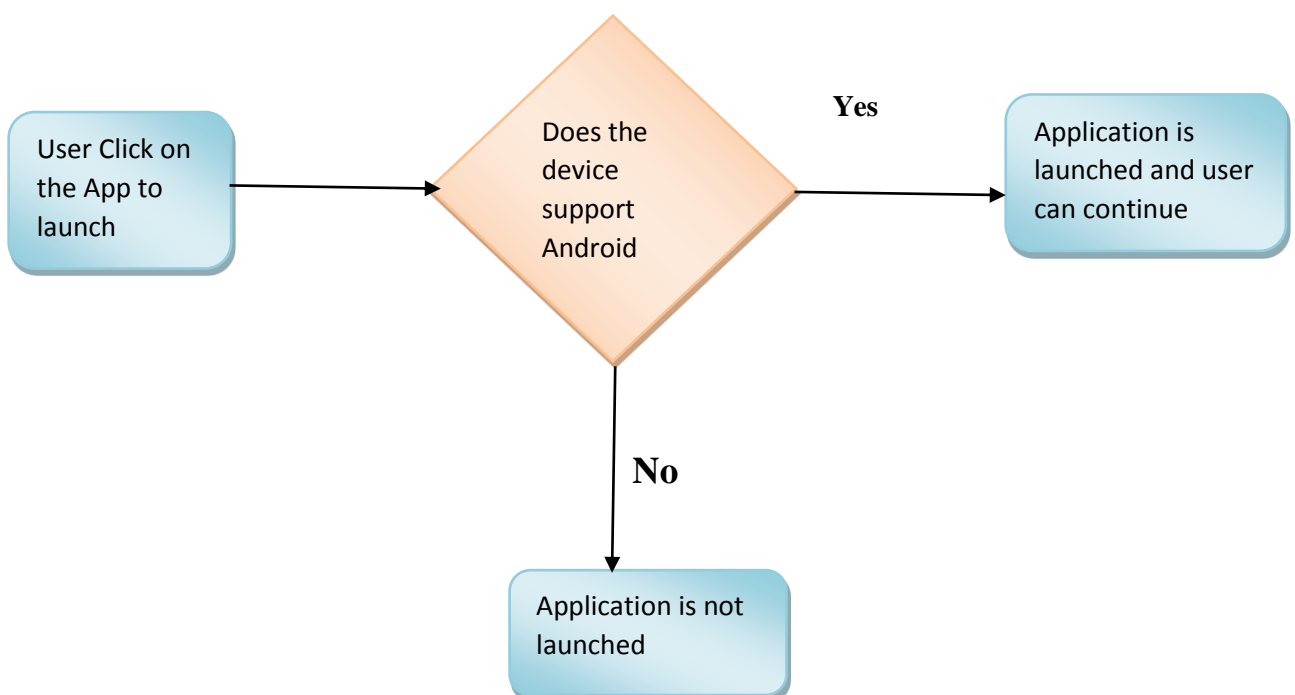
Description:

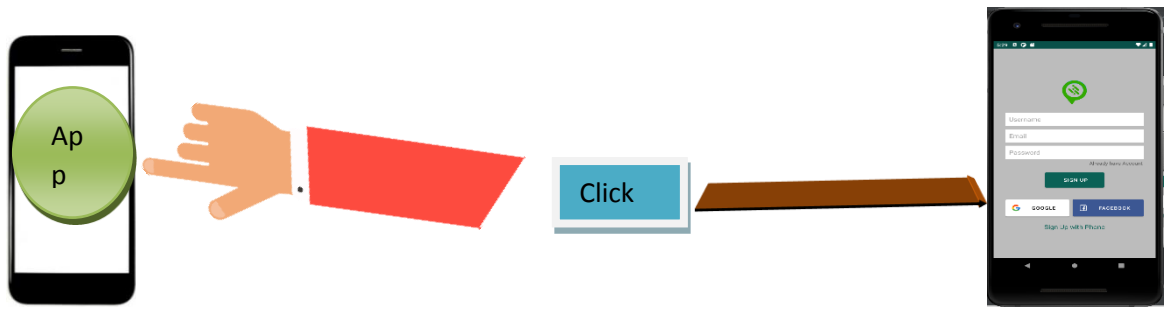
This use case is first on the priority list and describes the key functions that are offered to the user starting the application. If the user does not have the right or appropriate hardware, then they cannot launch the applications.

Scope

The scope of this use case is to describe how the user can launch the application and Get access to the primary functions that are displayed to the user.

Use case Diagram



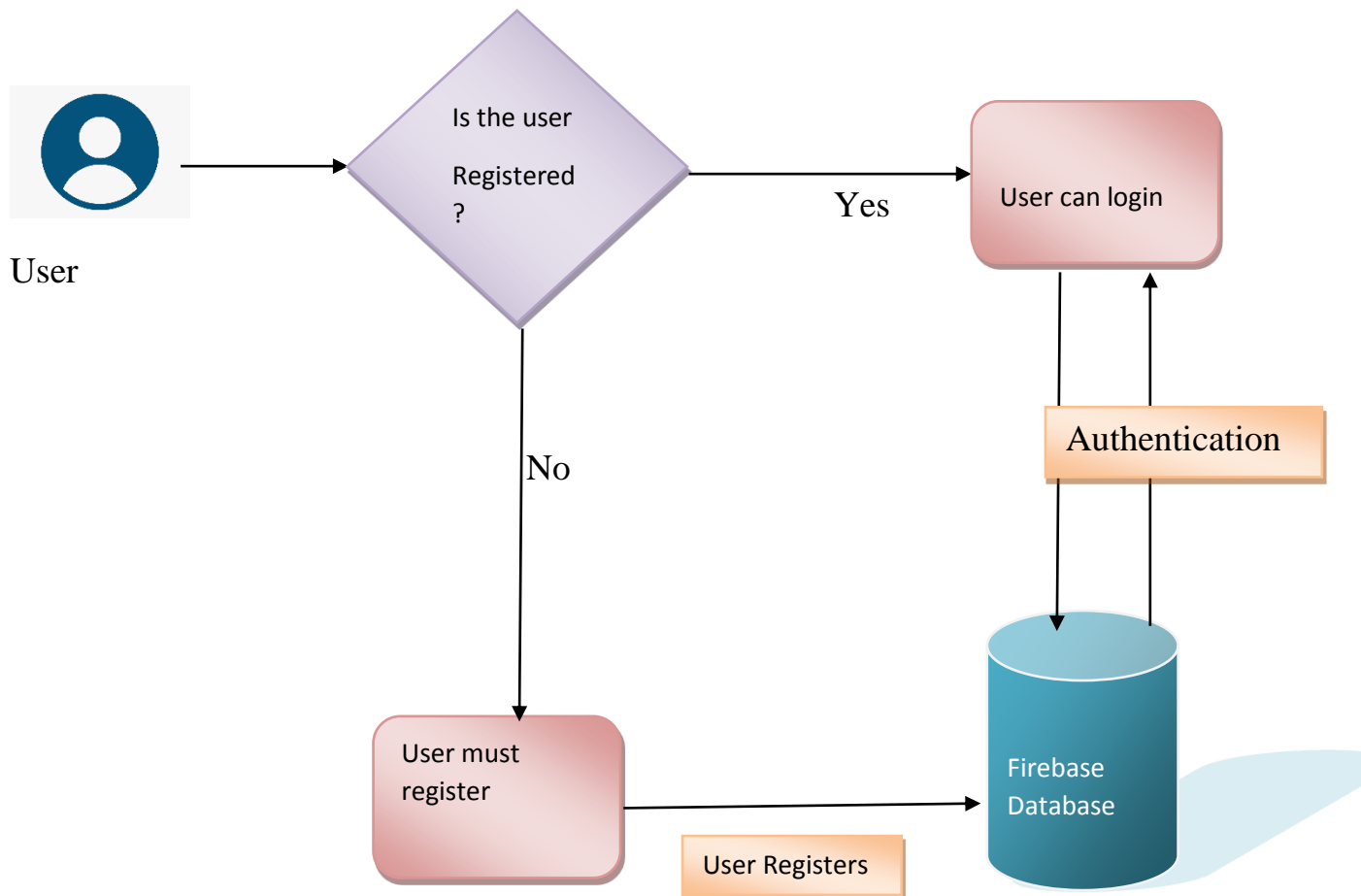


Requirement to Register or Login into Application

Priority: This use case is second in the priority list and describes the process of logging in and registering, once the application has booted up the user must be able to choose whether to login or register.

Use case: The scope of this use case is to describe how the user can begin logging in or registering.

Diagram



Pre-condition

The application has been booted up and is running in a waiting state.

Activation

This use case will start when the user has launched the application.

Main Flow

Actor Action: User clicks on the login.

System Response: Login tasks displayed for user.

Alternate Flow

Actor Action: The user is not registered and clicks on login.

System Response: The user is prompted to create an account.

Actor Action: The user does NOT click on login.

System Response: Login no register tasks will be launched.

Termination

This use case is terminated when the user logs in or registers successfully.

Post Condition

The user will register and be logged in.

Sending Messages

Description

This use case is third in the priority list and describes the process of sending a message on the basis of user id to the other user via the internet.

Precondition

The user s logged in as an authenticated user.

Activation This use case starts when the user has logged in.

Main Flow

Actor Action: User begins to type a message and presses send.

System Response: Sending process initiated and executed.

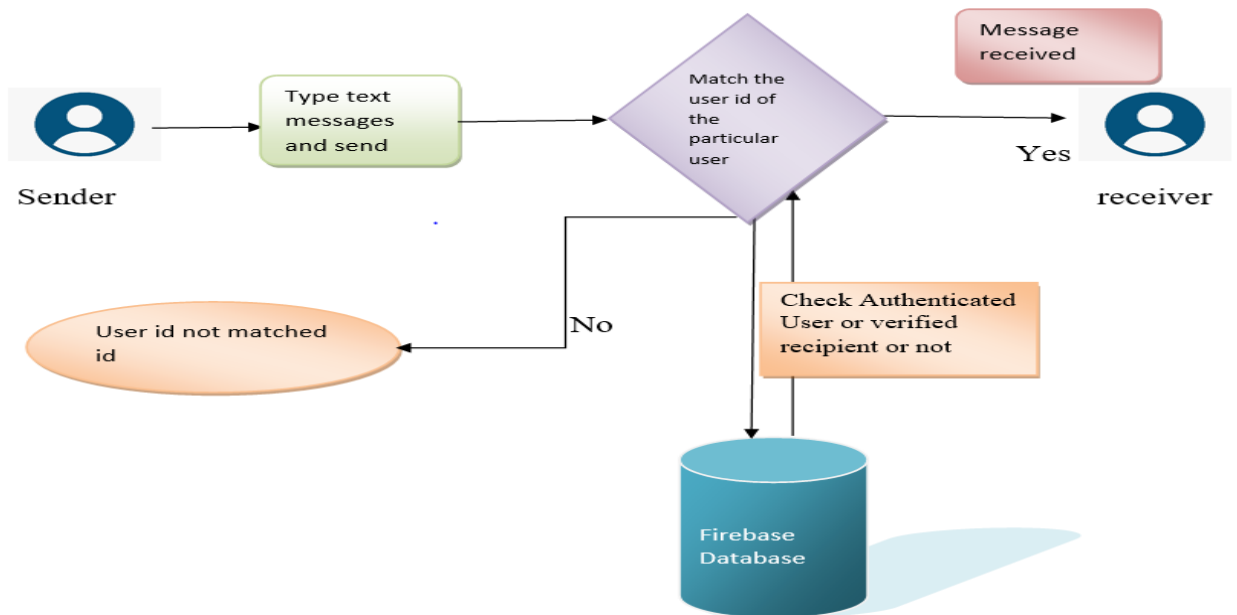
Alternate Flow

Actor Action: The user does not send the message.

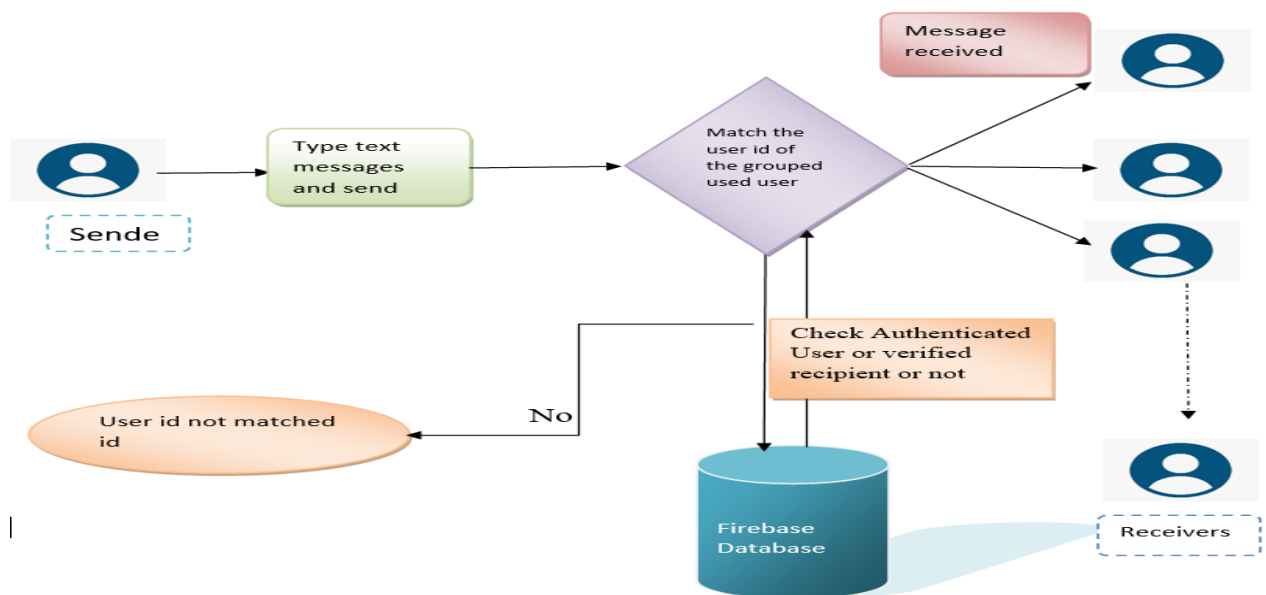
System Response: System stays idle, user has the option to sign out or exit the application.

Termination This flow is terminated when the user sends message or signs out and user id not matched and recipient not verified.

One to One Messaging



Group Messaging



3.4 Implementation

The implementation languages used in this project are mainly Java which is the language used while developing in Android Studio. At the back-end side I am using the Google Firebase authentication code to enable users to sign up and sign in using email password and their Google account, to me this made the most sense because the app is Android specific.

3.4.1 Firebase Authentication

Firebase Authentication provides backend services, an easy-to-use SDK, and a ready-to-use UI library for authenticating users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers such as Google, Facebook and Twitter, etc.

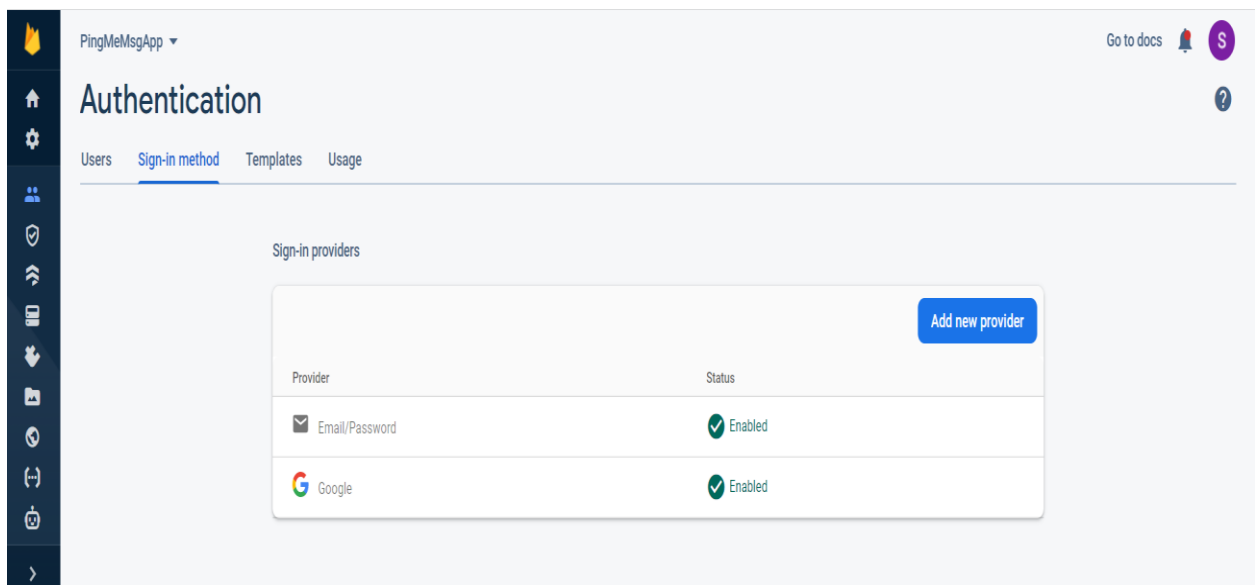


Behind the scenes of Authentication → Firebase Authentication

Before talking about how our app authenticates users, let's first connect to our App with the Firebase. And add dependency in our project modules.

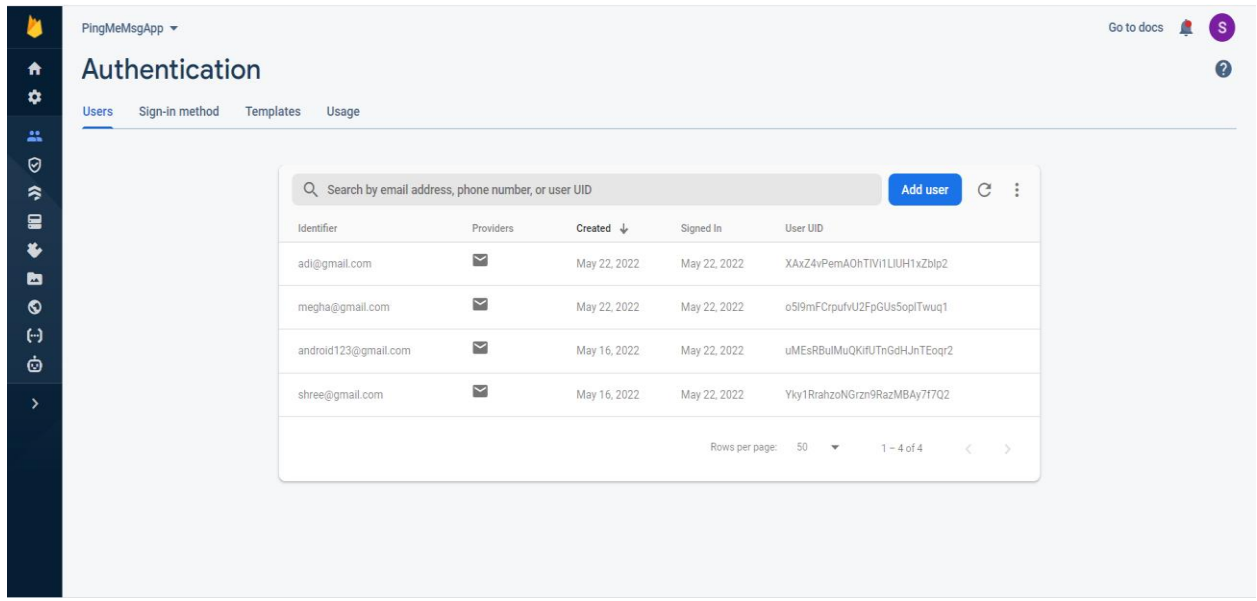
Using the Authentication emulator involves just a few steps:

1. Adding a line of code to your app's test config to connect to the emulator.
2. From the root of your local project directory, running `firebase emulators:start`.
3. Using the Local Emulator Suite UI for interactive prototyping, or the Authentication emulator REST API for non-interactive testing.



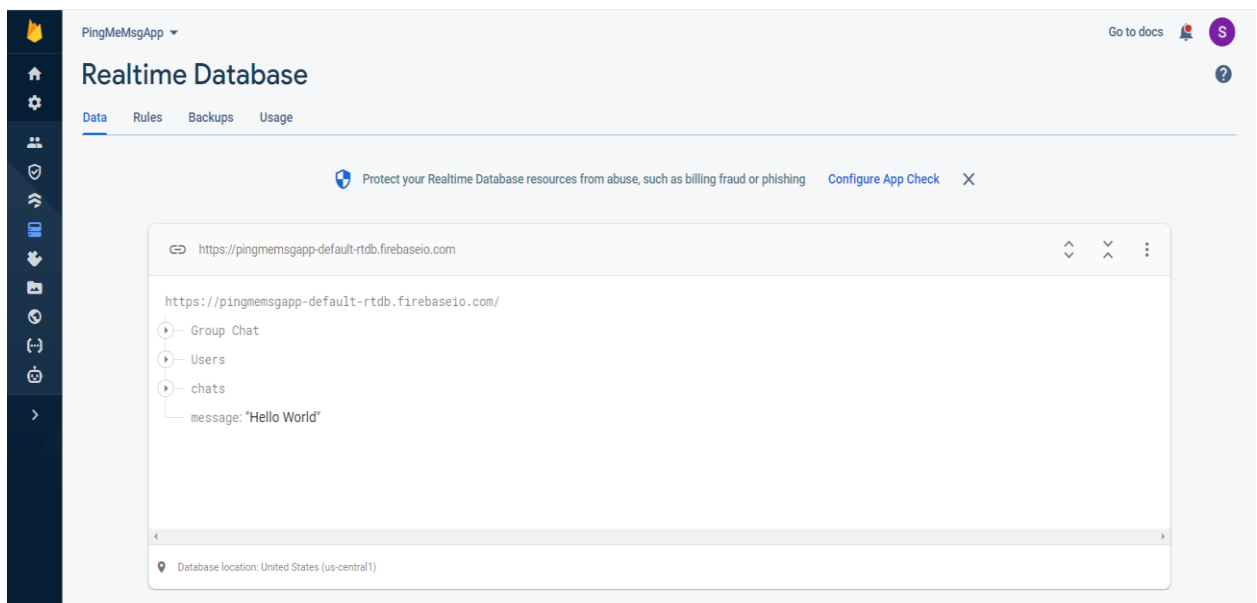
After that user want to create an account i.e. sign-up Activity and then he/she will be able to login in the app.

Authenticated user using Email :



3.4.2 Firebase Database

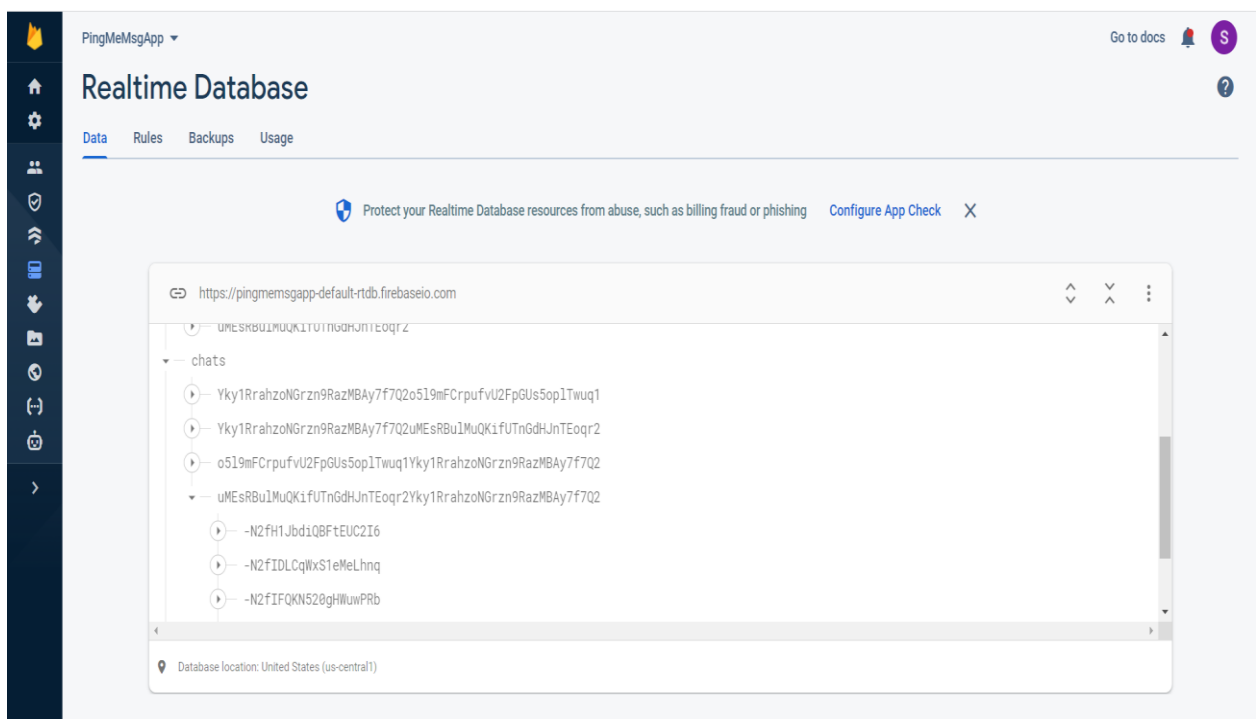
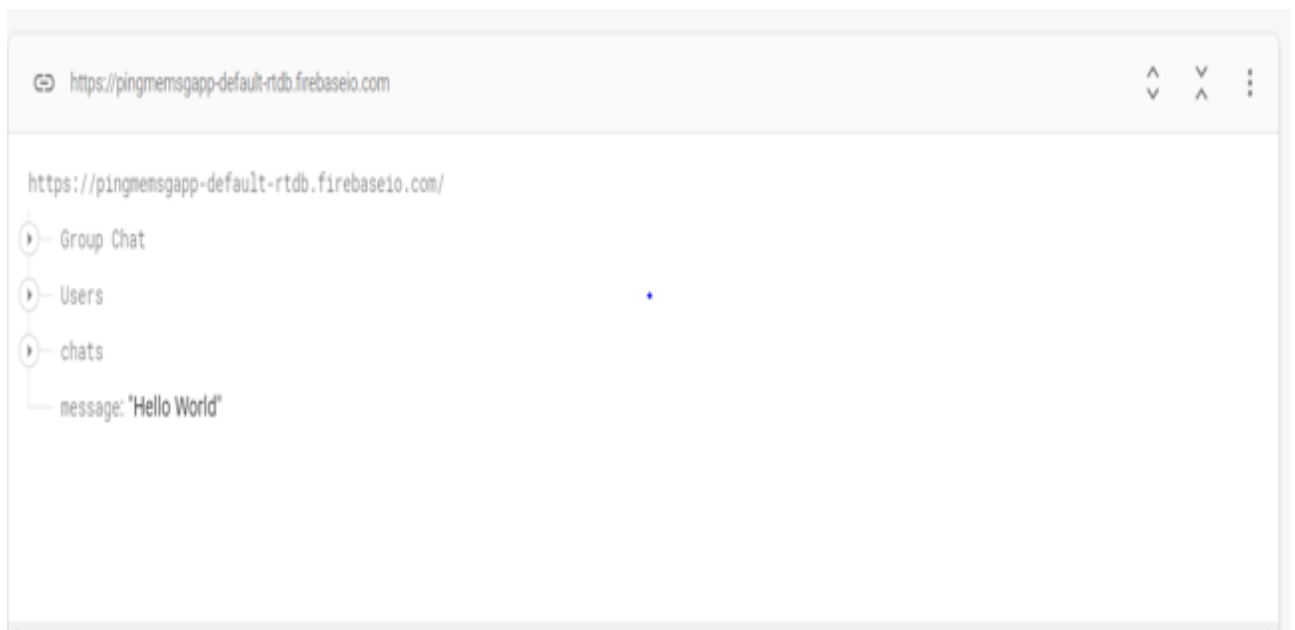
The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in realtime. Not only is the database being used to store user information but it is also being used to store user text messages and send them back to the application for all users to see. Below is an example of the message and user structure as seen on Firebase.



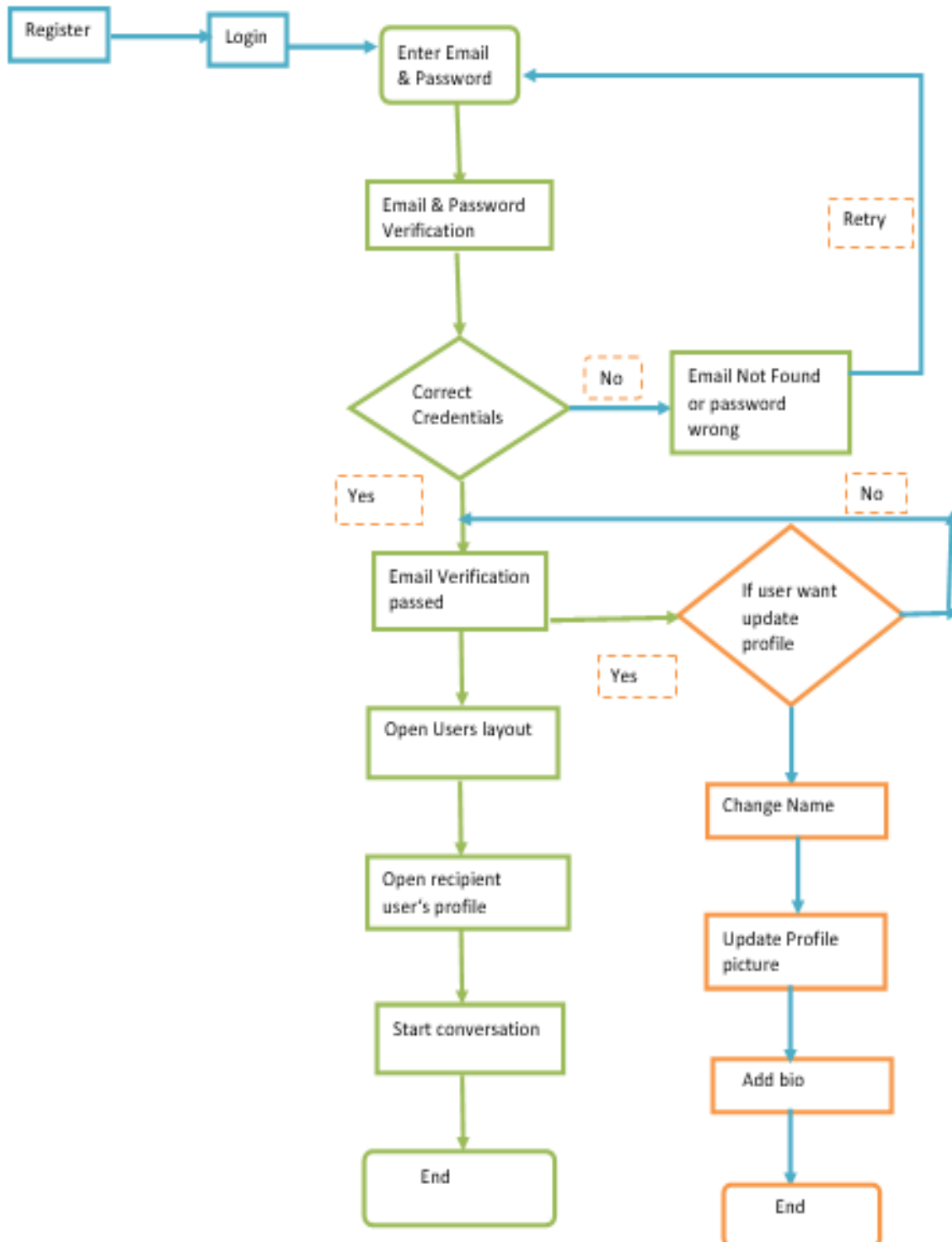
3.4.3 Message Implementation

If credentials are matched with the database, then the user is successfully logged-in, in the application and user can able to do messaging with the other recipient user. The messages are implemented using some modified Firebase code and sender and receiver user id. At the Backend of the, messages are passing to the both ends on the basis of the user id.

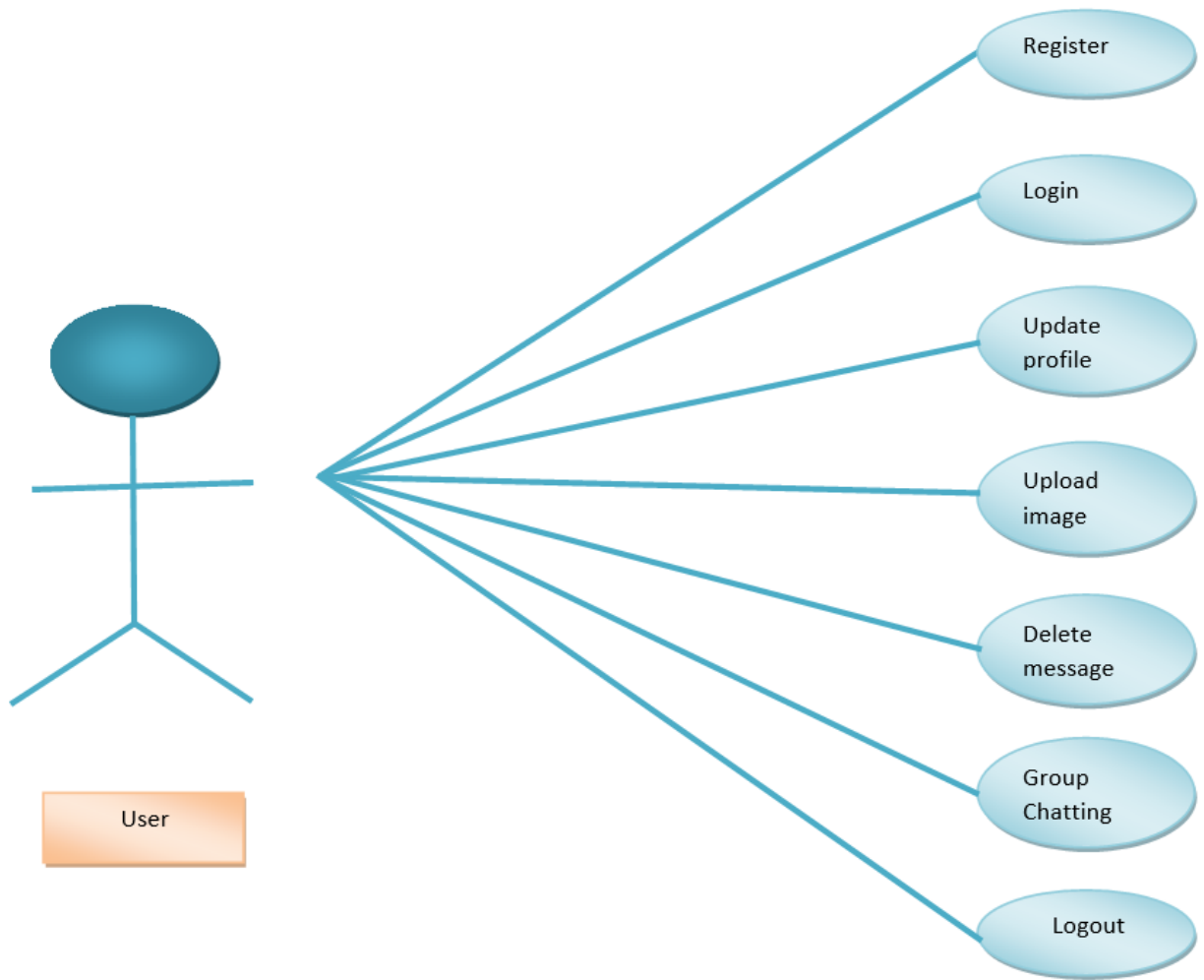
If *the user starts* the messaging with the other user then one node *will be created in* the database and this node is called chats node and this node will be store the user's chats.



Flowchart Of “ PingMe” Messaging application



Use case Diagram



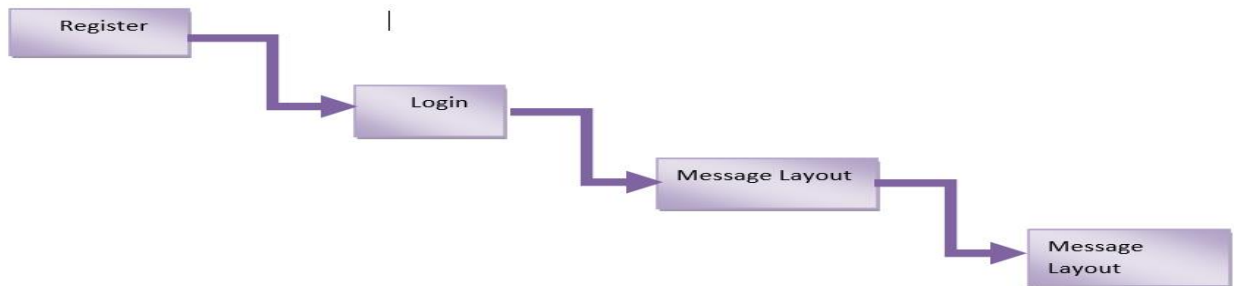
Chapter 4

Final Analysis and Design

4.1 Result

The final system will result in a real time communication application which facilitates the users to communicate with each other easily. The application will have a login page through which the user can register and login himself. After that, user can set his name and profile picture and start chatting with other users.

DFD Level 0:



DFD Level 1:



4.2 Graphical User Interface Layout:

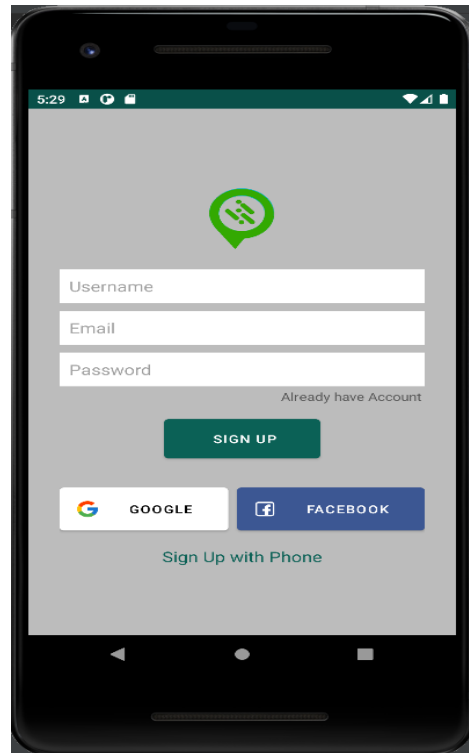
This is an actual Visualization of the implementation of this project.

4.2.1 Launch Page



4.2.2 Sign Up Activity

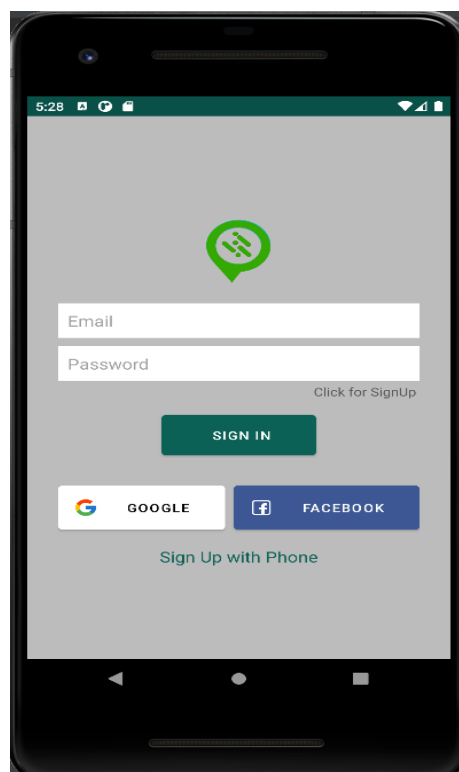
This is a starting Activity layout of the application where user can register themselves in to the application.



4.2.2 Sign In Activity

This is a second Activity layout page of the application. Where User can put their login credential.

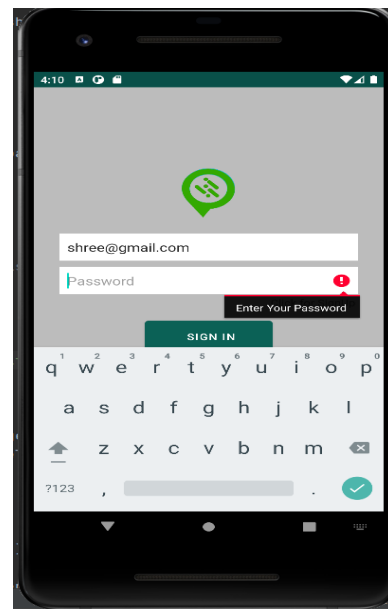
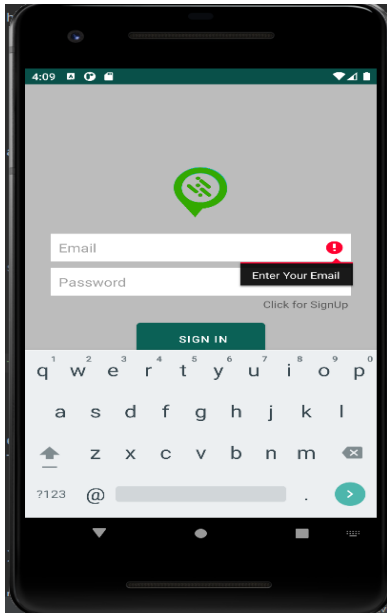
Login Process: If the user *successfully completes* their sign up activity, then he/she is able to login in the app.



4.2.2.1 Email & Password Validation

User cannot leave the email and password field. Email is the most preferred mode of communication and an innate part of running a business. It is important to keep in mind that the list of email addresses is valid and without any errors. Making errors while typing an email address is quite common and the best way to resolve this issue is to use Email Validation or user can't leave the empty field without writing their email.

Email Validation is a method through which it is verified if the email address is accurate and valid or not and whether the address's inbox can receive the mail or not.



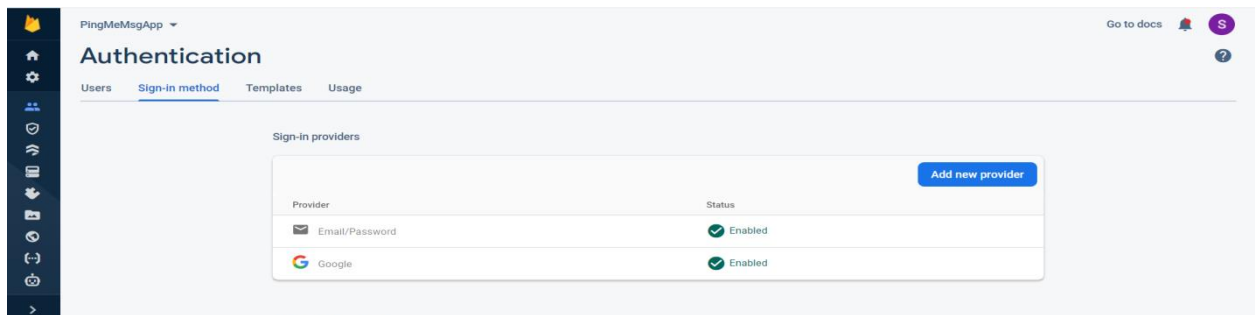
Behind the scenes of Authentication → Firebase Authentication

Before talking about how our app authenticates users, let's first connect to our App with the Firebase. And add dependency in our project modules.

Using the Authentication emulator involves just a few steps:

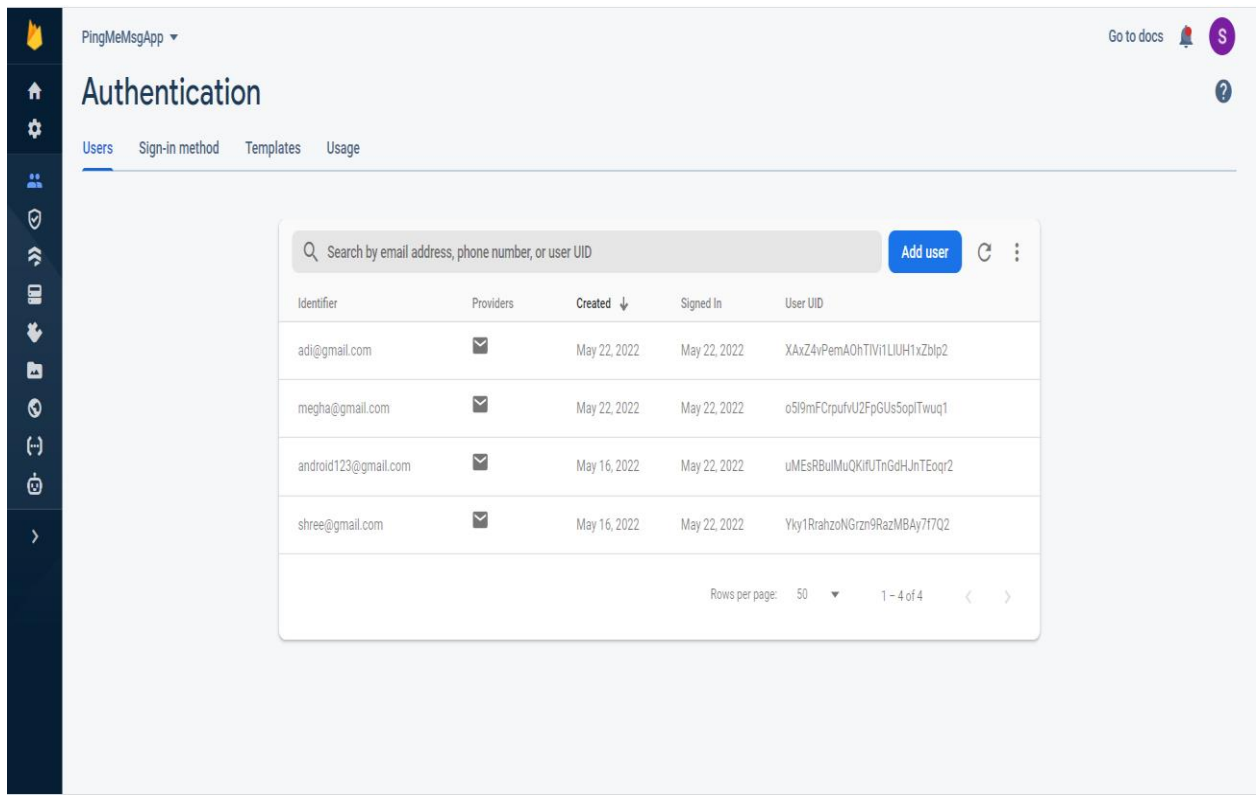
4. Adding a line of code to your app's test config to connect to the emulator.
5. From the root of your local project directory, running `firebase emulators:start`.
6. Using the Local Emulator Suite UI for interactive prototyping, or the Authentication emulator REST API for non-interactive testing.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.



After that user want to create an account i.e. sign-up Activity and then he/she will be able to login in the app.

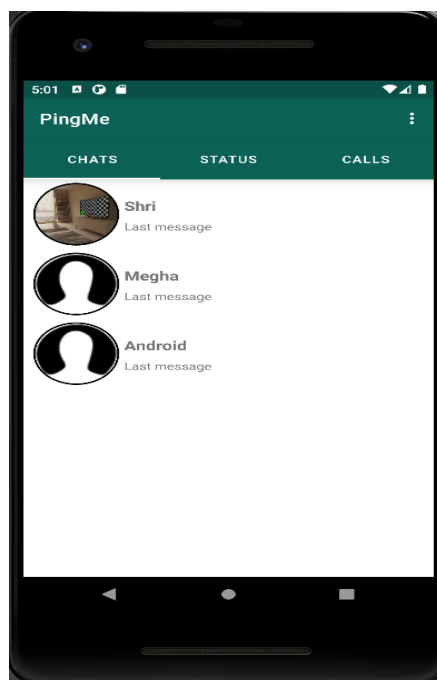
Authenticated user using Email :



If credentials are matched with the database, then the user is successfully logged-in, in the application and user can able to do chatting with the other recipient user.

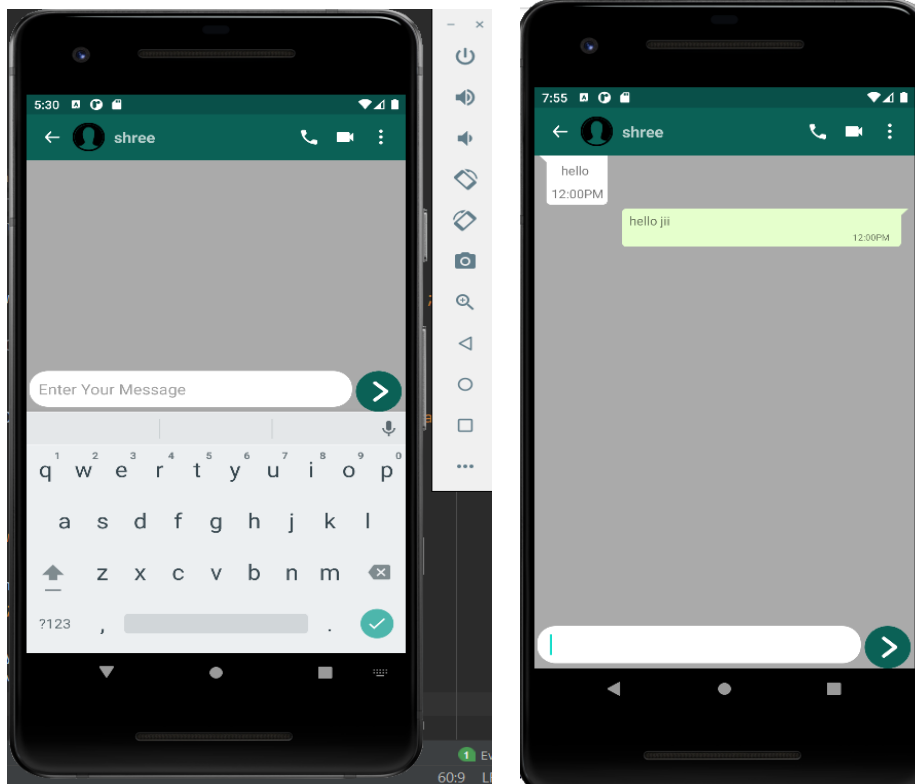
4.2.3 User View Layout

After login in to the application there are a layout called user view layout where all the *user* profiles are *shown*.



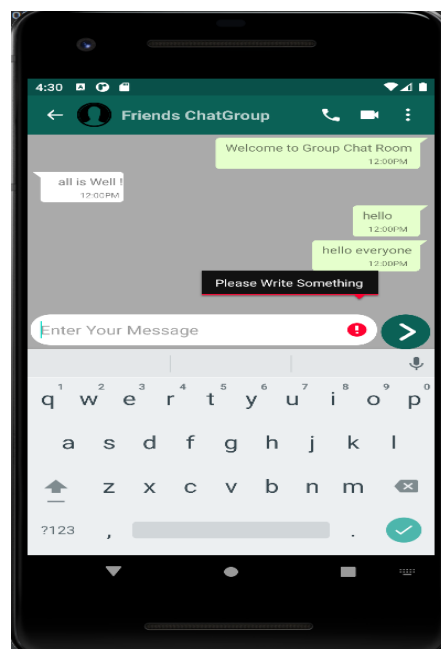
4.2.4 Message Layout:

User can start the messaging with the other user.



4.2.4.1 Text field Validation:

If user click the send button without write any messages then it show some validation error. User cannot click the send button without writing something.



Firestore Database

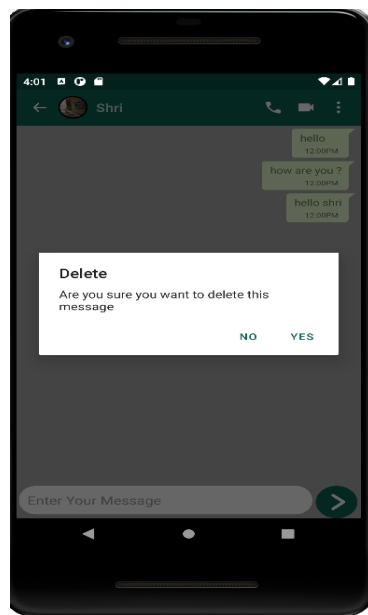
Not only is the database being used to store user information, but it is also being used to store user text messages and send them back to the application for all users to see.



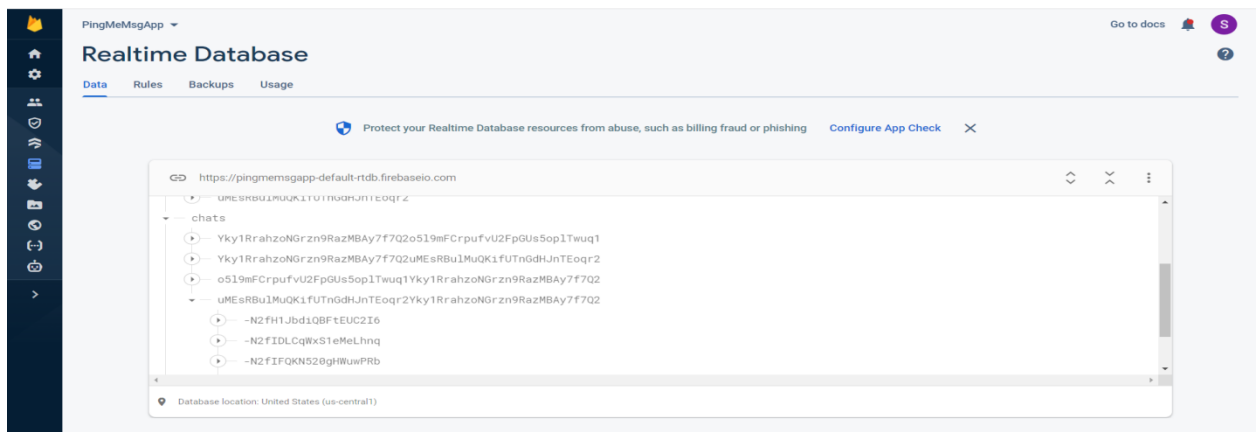
If the user starts the chatting with the other user then one node will be created in the database and this node is called chats node and this node will be store the user's chats.

Delete Messages

User can delete the message from the same end.



Behind the scenes of Chats in Firestore Database Storage



4.2.5 Menu field

There are a menu field where user can handle the profile setting and all. Menu field contains 3 menu i.e.

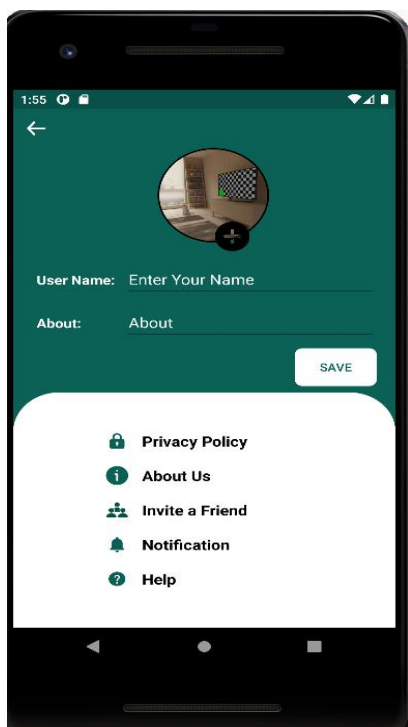
- Setting
- Group Chatting
- Logout

Setting Field:

In this field user can go to profile setting and update the profile. Here user can update profile picture, update user name, and add bio.

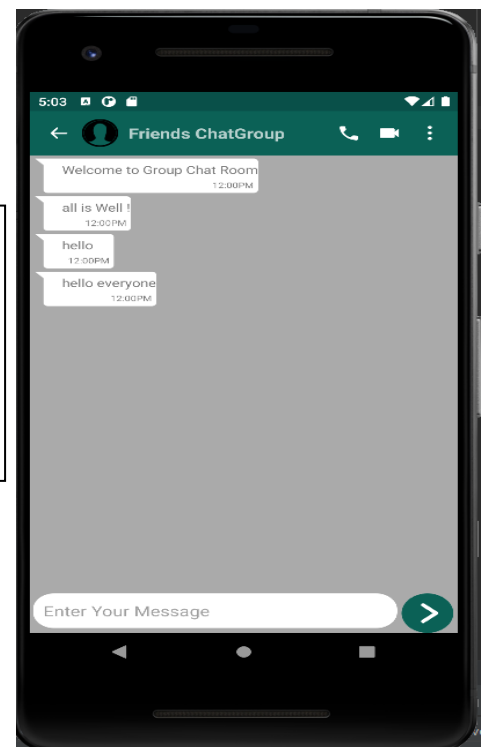
Update Profile

In this app update is just an introduction to features like updating user/group name, change profile picture and add bio.



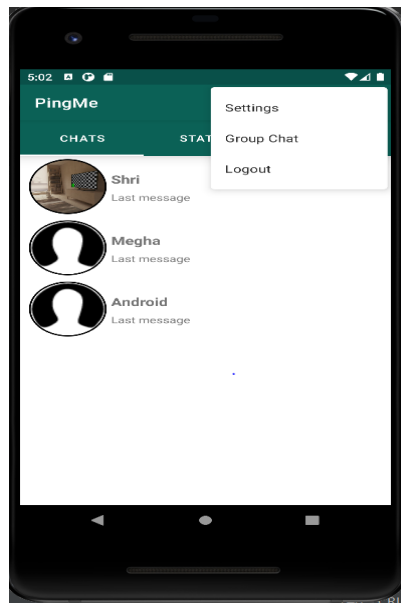
Group Messaging

In this application user can also do group chatting. Group chat is the ability to chat with



Logout

User can also Logout from the application.



4.3 Future Scope

To experience the new modern paradigm shifts in technology humans will need to be at one with the technologies they create and the ability to interface in real time. AS the basic idea behind our app is “PingMe” message Application which in future can be updated to more features such as: -

1. Group Calls
2. Voice Calls
3. Video Calls
4. Voice messages
5. Send Emoji's

Application

1. User can use this application for the simple text messaging.
2. In this application, user can update their profile like: change profile picture, change user name, and user can add bio.
3. User can also do the group chatting.
4. User can delete the messages.
5. User can Logout from the application.

Conclusion

We have made a basic version of messaging app which in future will require some updates and side by side we are going to work on them. Up to how much extent application is efficient, there is always some room of improvement.

References

- 1.”Android Developers”, <https://developer.android.com/>
- 2.”Android SDK”, <https://developer.android.com/reference/packages.html>
3. <https://androidcommunity.com/>
- 4.”Firebase”, <https://firebase.google.com/docs/android/setup>

Plagiarism Report



Similarity Report ID: oid:28506:17887001

PAPER NAME

PingMe_MsgApp Project paper (2).pdf

AUTHOR

S. Jaiswal

WORD COUNT

1781 Words

CHARACTER COUNT

9293 Characters

PAGE COUNT

4 Pages

FILE SIZE

976.4KB

SUBMISSION DATE

May 30, 2022 11:41 AM GMT+5:30

REPORT DATE

May 30, 2022 11:42 AM GMT+5:30

● 7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- 1% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 14 words)

Summary