Green University of Bangladesh

Department of Computer Science & Engineering

Faculty of Sciences & Engineering

Semester: (Fall, Year: 2022), B.Sc. in CSE (Day)

Project Report
Course Title: Computer Networking Lab
Course Code: CSE 312 Section: DB

Project Name: Group Chatting Android Application(Communicator).

Student Details

Name	ID
Rayhan Kobir	******

Submission Date: January 07, 2023
Course Teacher's Name: Rusmita Halim Chaity

[For Teacher's use only: Don't write anything in this box.]

Lab Report Status

Marks: .... Signature: ....

Comments: .... Date: ....

Contents

1 Introduction

Group chatting application Communicator is an android application. That will help to communicate with many of people by using this android application. This application provide user friendly interface where user interaction will be smooth. If the user send their message it will be broadcast to every user and every user can read this message. Also they can direct message to others users..

1.1 Aim of the project

Nowadays, everyone wants connect together and make communication with each other's. At this time, if we are talking about the social media application we will found lots of social media application, where people spent lots of time to watch ads and we sometimes hear about information leak. At this point of view many company force/discouraged their employee to use external messaging application instead they encourage to use their own messaging application to make prevent their data/information leak. Considering all these things we are building this application to keep in touch with our friends. In this time maximum people are using android phone that reason we decided to make as android application. Our application shall provide the following features:

1. Prevent data/information leak.
2. Establish a communication between many of peoples.
3. User make group and when anyone send message in the group every group members will received the message..

1.2 Objective

The main objectives of this project are:

1. User can send message directly to the others user.
2. No ads on the application..
3. User friendly user interface..
4. Data safety and provide security.
5. Easy to navigate.

2 Methodology

In this application we are used Agile Model to build our application. Our application is user based multiple users are connected with our application. So we have to changed our requirements frequently that's why this agile model is good fit for our application. Another reason to chose agile that is agile is good for mobile application it's provide us to build ideal application by using iterative and incremental process.

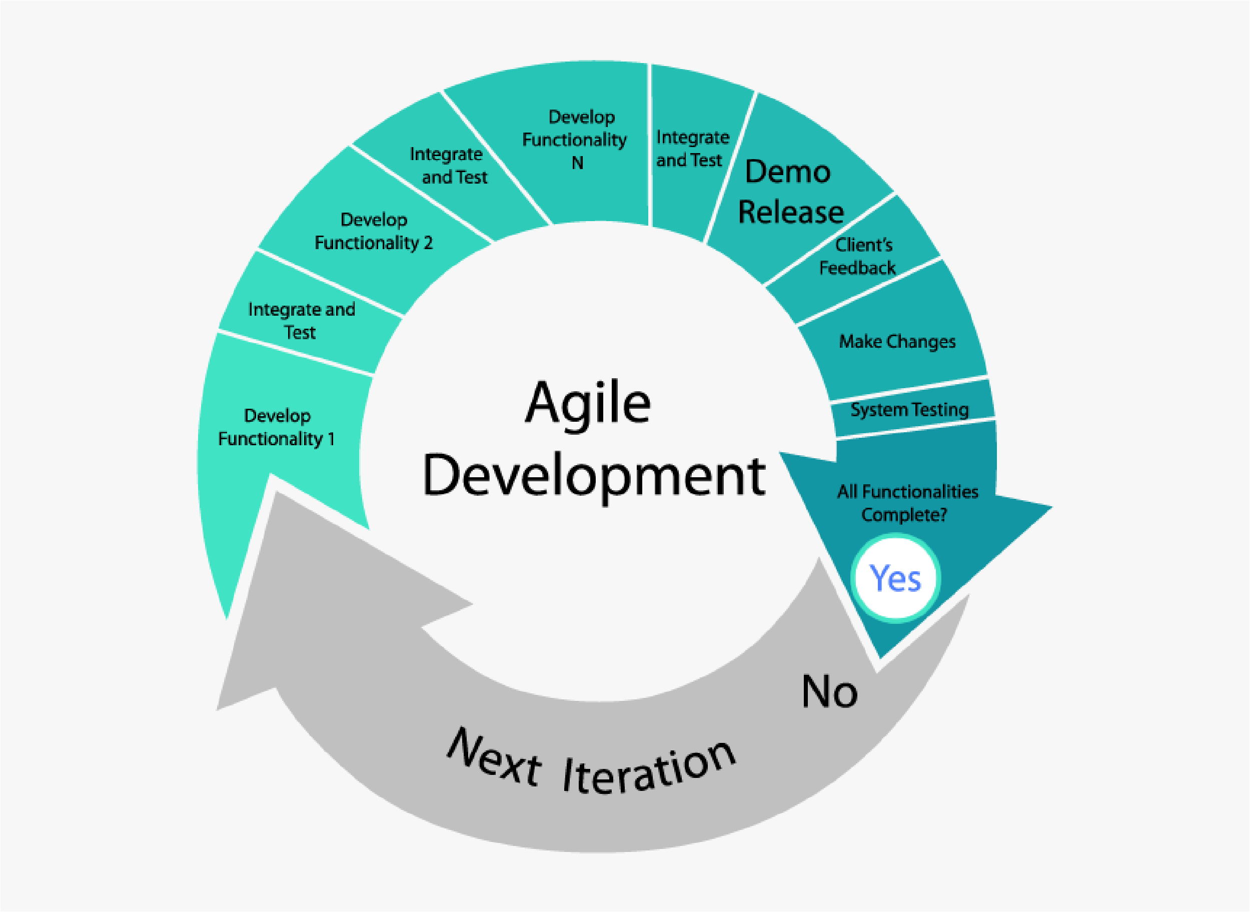


Figure 1: Agile Model

2.1 Socket & ServerSocket

This class implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.
Creates a server socket, bound to the specified port. A port number of 0 means that the port number is automatically allocated, typically from an ephemeral port range. This port number can then be retrieved by calling getLocalPort. The maximum queue length for incoming connection indications (a request to connect) is set to 50. If a connection indication arrives when the queue is full, the connection is refused. Socket are the pre-build class which provided by Java

2.2 BufferReader & BufferWriter

Java BufferedWriter class is used to provide buffering for Writer instances. It makes the performance fast. It inherits Writer class. The buffering characters are used for providing the efficient writing of single arrays, characters, and strings.

BufferReder reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines. The buffer size may be specified, or the default size may be used. The default is large enough for most purposes.

1. BufferWriter allows us to send the data one user to another.
2. BufferReader allows us to get data from client as well as server.
3. Encrypt data for further security. For more security, encryption and decryption services are offered. Message expansion and compression are simplified to guarantee that communications pass efficiently from one user to another.
4. Socket and BufferReader used TCP connection to make communicate with each other here TCP control the flow of the data and stream will allow to get data from the stream.

3 Analysis Design and Development

3.1 Requirement Gathering Technique

Functional Requirements

In nowadays we are used so many social media application and it's hard to take over the market. To get huge portion of the market we have need good requirements and execution

Analysis of the Requirement

The specifications for the system are the descriptions of the activities and constraints, and the process of discovering, evaluating, documenting, and testing these products and constraints is known as requirements engineering. Both software and hardware requirements for our project are listed below.

1. **Hardware Requirements:**
   To host the server of the communicator, It's required at-least 2 GB RAM, enough hard-drive space and internet connection.

2. **Software Requirements:**
   This following software we have used to implement our project-

   (a) Windows 8 or

   (b) Windows 10 platform.

   (c) Android OS.

   (d) Minimum API level 29

   (e) Minimum 2 GB RAM

   (f) Server.

   (g) Android OS

3. **Android (OS):**
   To run our client users need the Android OS

3.2 Data Flow Diagram

A data flow diagram is a graphical representation of system needs. It specifies the data flow the procedure, and the storage location.

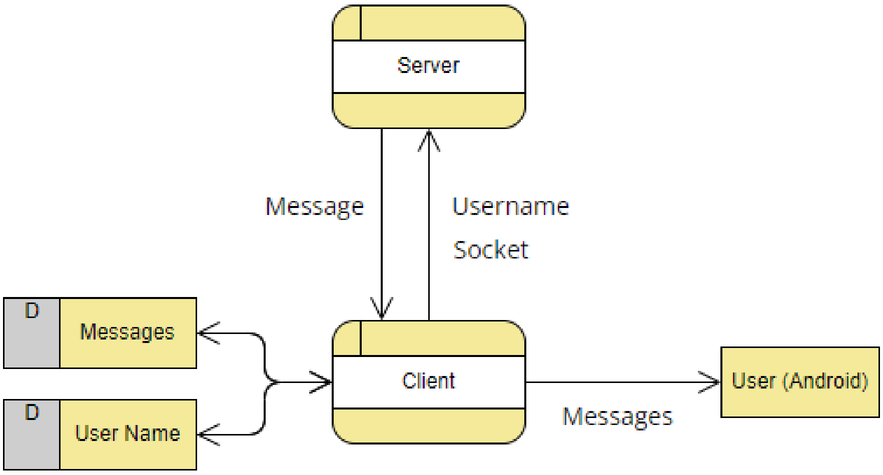


Figure 2: DFD of Communicator

Data Flow:

1. Server receives request from client with user name and the socket of the client. And it will process the client and broadcast the message to all the users.
2. Client receives the username and socket from android application and client will send to the server.
3. After that the server send the response to the client and client send the broadcast message to the android application.
4. The user receives the message and android activity will show the message on the UI.

3.3 Class Diagram

In this application we are followed the class diagram where we used client, server and clienthandler class. Each class have some private and public attributes which will access from one to another class by inherit and by creating instance of the classes.

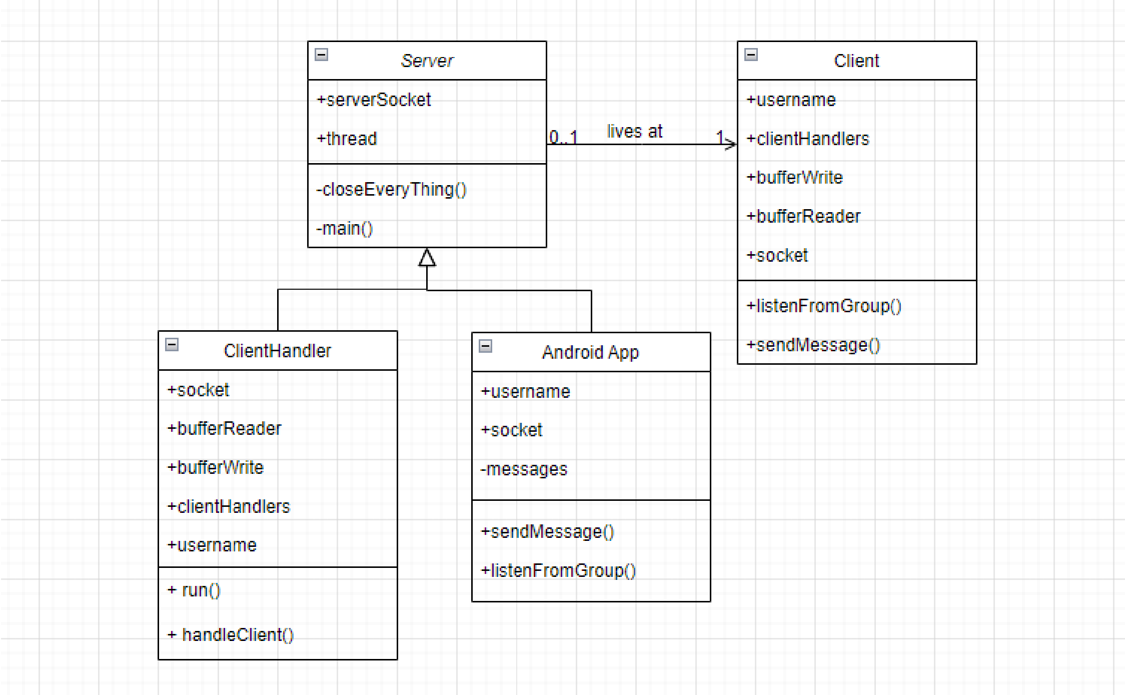


Figure 3: Class Diagram

Data Flow:

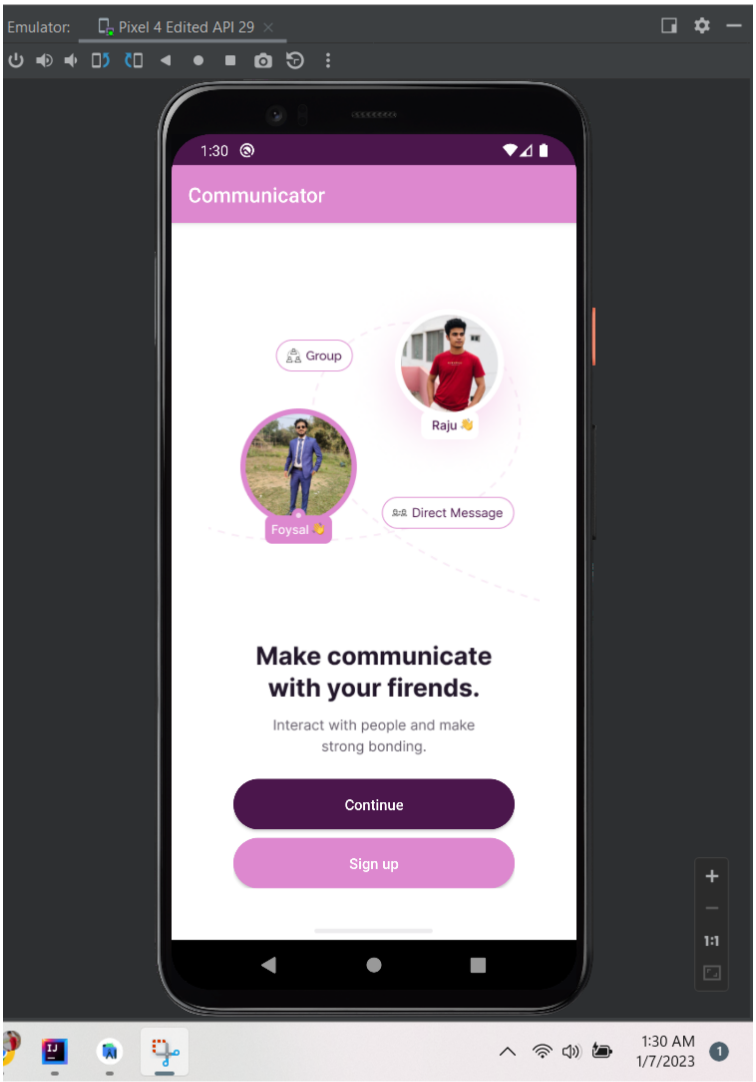
4 Implementation Of the Project

4.1 User Interface

To build user interface we are concerned about simple and minimalist and user centered design which will help to the users to use outer application. We are used Linear , Relative and Absolute Layout to make this user interface

4.1.1 Onboard Screen

In this screen user's have two options continue and signup button if they preesed on the continue button application will navigate to login page other hand if they clicked on the signup button then application will turn over to the signup page where user can signup and make communicate with their friends.



Make communicate with your firends.

Interact with people and make strong bonding.

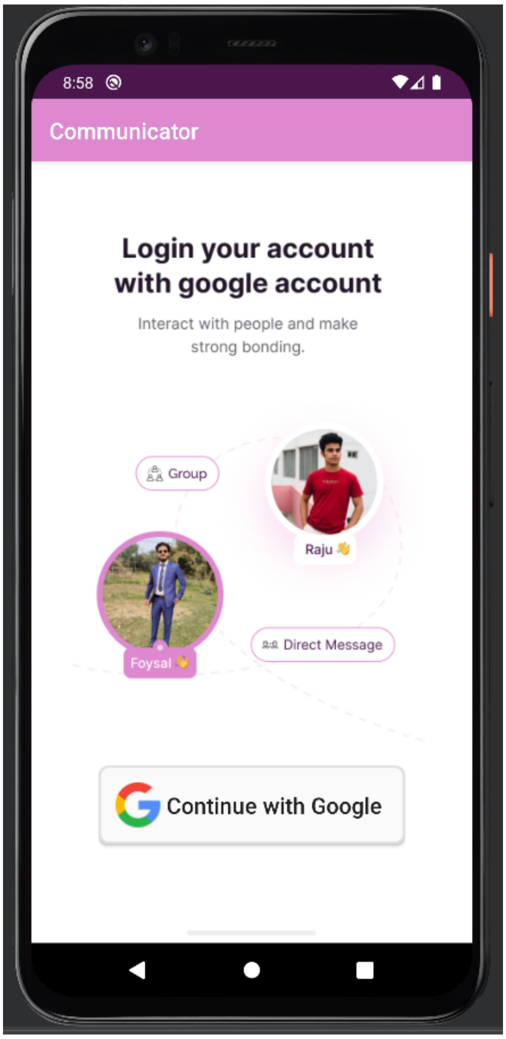
Figure 4: Onboard Screen

4.1.2 Login Signup Screen

In this screen application will provide continue with google if user pressed on this button application will authenticate the users and application navigate to the dashboard screen where user can find some option to make communicate.

4.1.3 Dashboard Screen

Dashboard screen have one button over here that is new group if user clicked on it the application will naviagte the user to the chat screen where user can text to others users.



8:58

Communicator

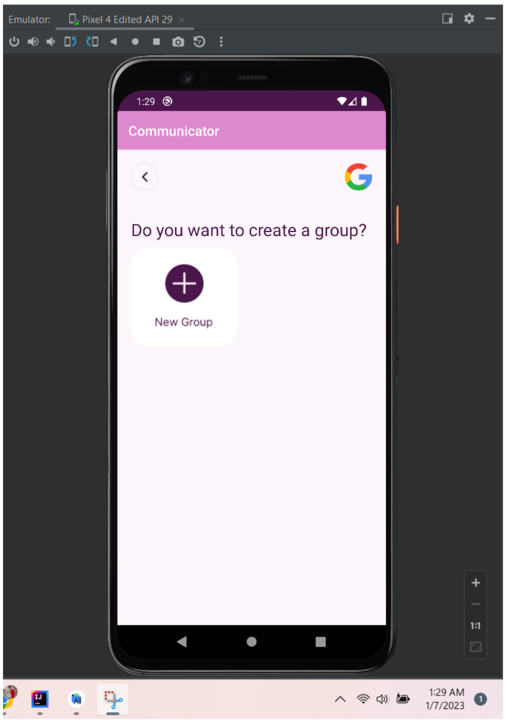
Login your account with google account

Interact with people and make strong bonding.

Figure 5: Login Signup Screen

4.1.4 Chat Screen

This screen will responsible for the user interaction here application will show the chat list where user can find the all messages as well as they can sent a message to the group. This screen provide input field and send button to the user for sending the message.

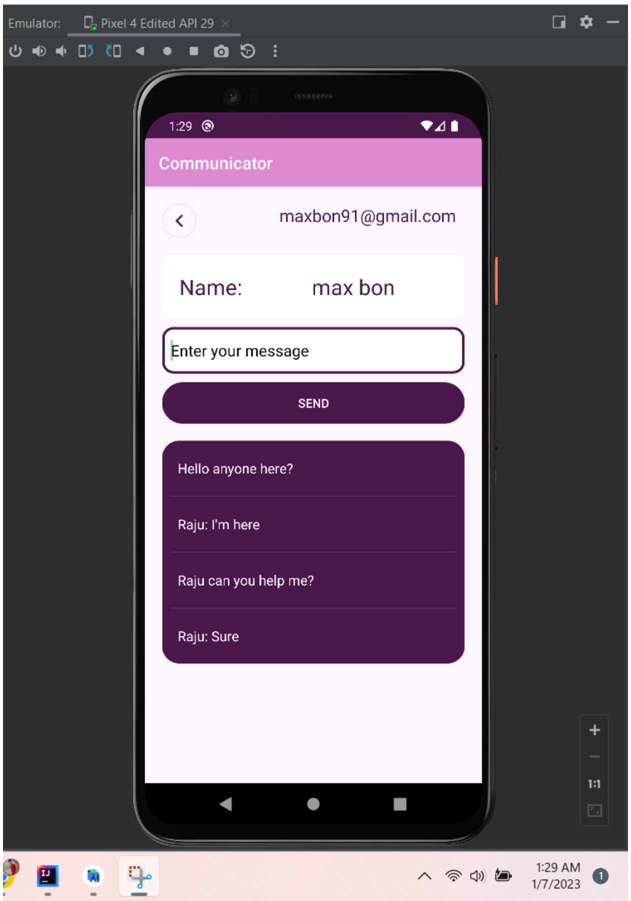


Do you want to create a group?

Figure 6: User Dashboard Screen

4.2 User Authentication Functionality

Our Project is mostly implemented on Java and Android Studio.
here we can see the first floor of our project. it is the management floor of out system.
here we can see bufferReader read the newest data from the stream and after that users give message catch the message as string and write to bufferWriter which will send to the server.



maxbon91@gmail.com

Name: max bon

Enter your message

Hello anyone here?

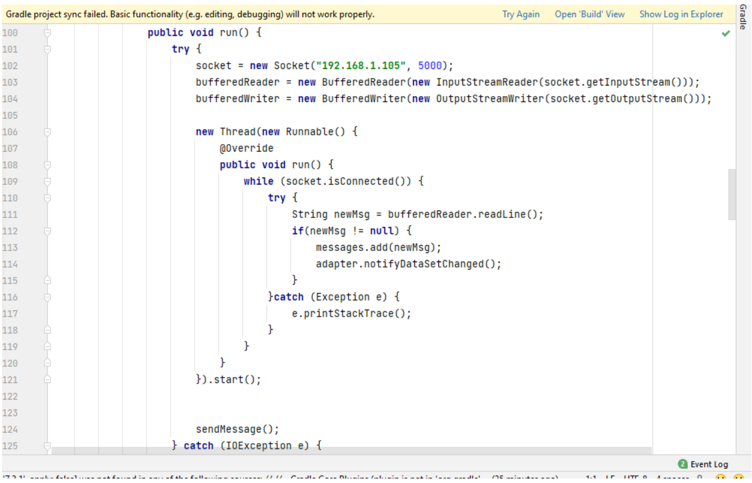
Raju: I'm here

Raju can you help me?

Raju: Sure

Figure 7: Chat Screen

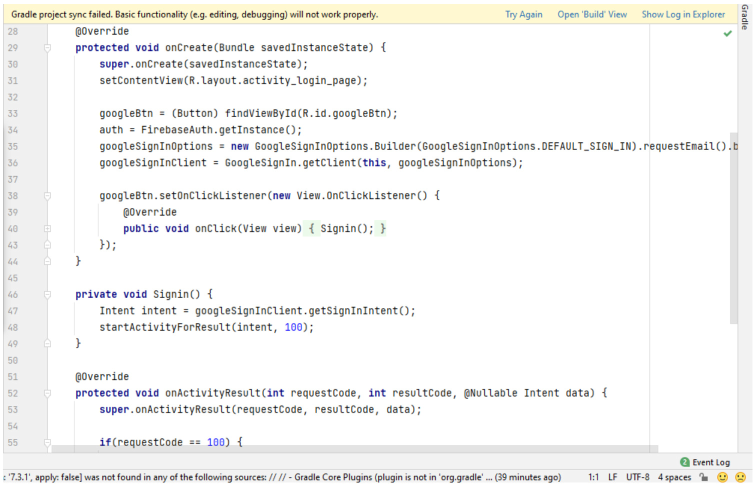
here we use Google Authentication for the login and sign up both after the login the application will navigate to dashboard page of the communicator.



Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.

```
            public void run() {
                try {
                    socket = new Socket("192.168.1.105", 5000);
                    bufferedReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
                    bufferedWriter = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));

                    new Thread(new Runnable() {
                        @Override
                        public void run() {
                            while (socket.isConnected()) {
                                try {
                                    String newMsg = bufferedReader.readLine();
                                    if(newMsg != null) {
                                        messages.add(newMsg);
                                        adapter.notifyDataSetChanged();
                                    }
                                }catch (Exception e) {
                                    e.printStackTrace();
                                }
                            }
                        }
                    }).start();


                    sendMessage();
                } catch (IOException e) {
```

Figure 8: Chatting features of communicator



Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.

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

        googleBtn = (Button) findViewById(R.id.googleBtn);
        auth = FirebaseAuth.getInstance();
        googleSignInOptions = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).requestEmail().b
        googleSignInClient = GoogleSignIn.getClient(this, googleSignInOptions);

        googleBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { Signin(); }
        });
    }

    private void Signin() {
        Intent intent = googleSignInClient.getSignInIntent();
        startActivityForResult(intent, 100);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 100) {
```

```
: '7.3.1', apply: false] was not found in any of the following sources: // // - Gradle Core Plugins (plugin is not in 'org.gradle' ... (39 minutes ago)    1:1  LF  UTF-8  4 spaces
```

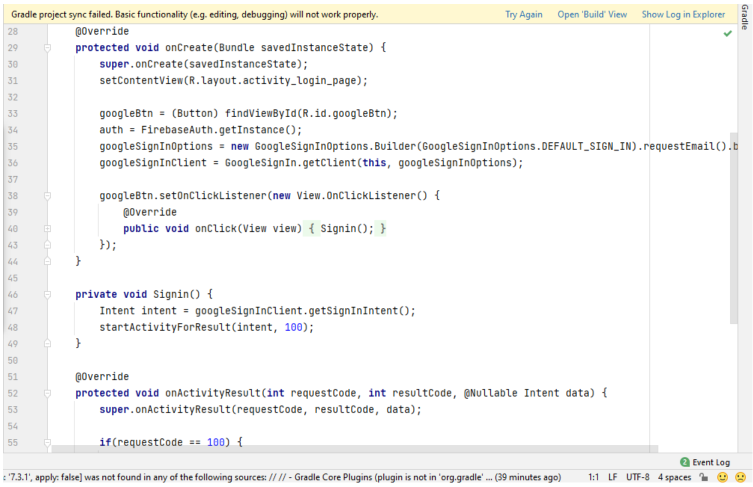
Figure 9: Message sending features

5 Results & Analysis

Communicator have server side and client side. Users are interact with the client side which is Android Application.

5.1 Results

If we run the server side it able to take request from the client and when client run the application then they connected with the server and they can communicate with each others. If we run the our android application user will landed to the onboard screen of the application Figure 3 and after login application will turn over the dashboard screen and after clicking the new group button application will navigate to the chat screen where user can see all of the messages and they can send the message by filling the message field and clicking the send button then the message will be send to others users.

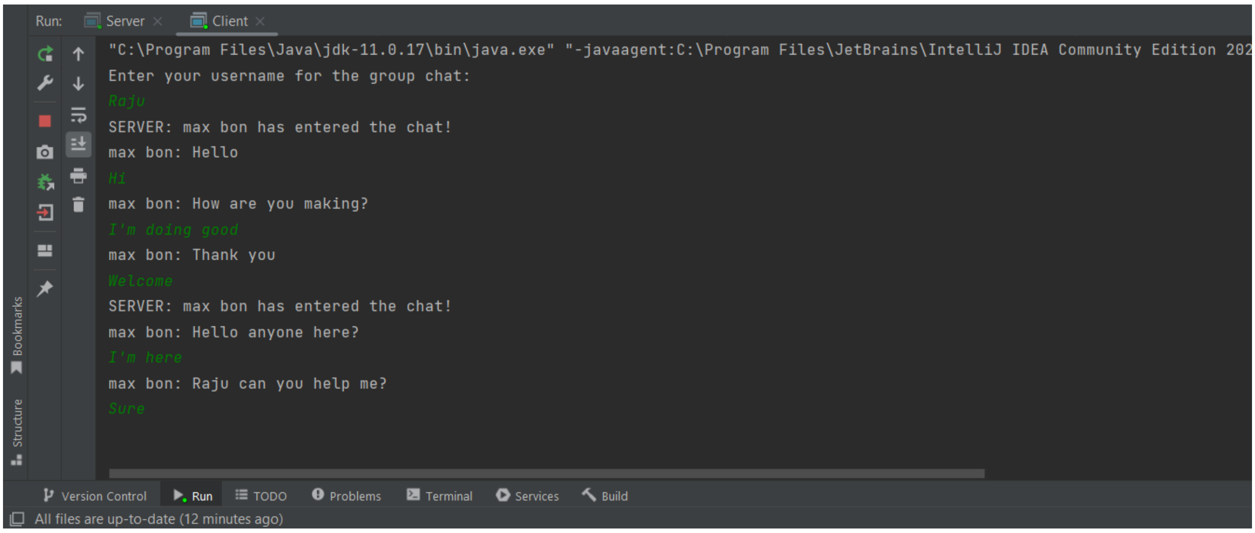


Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.

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

        googleBtn = (Button) findViewById(R.id.googleBtn);
        auth = FirebaseAuth.getInstance();
        googleSignInOptions = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN).requestEmail().b
        googleSignInClient = GoogleSignIn.getClient(this, googleSignInOptions);

        googleBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { Signin(); }
        });
    }

    private void Signin() {
        Intent intent = googleSignInClient.getSignInIntent();
        startActivityForResult(intent, 100);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 100) {
```

: '7.3.1', apply: false] was not found in any of the following sources: // // - Gradle Core Plugins (plugin is not in 'org.gradle' ... (39 minutes ago)

Figure 10: Login system of Communicator



```
"C:\Program Files\Java\jdk-11.0.17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 202
Enter your username for the group chat:
Raju
SERVER: max bon has entered the chat!
max bon: Hello
Hi
max bon: How are you making?
I'm doing good
max bon: Thank you
Welcome
SERVER: max bon has entered the chat!
max bon: Hello anyone here?
I'm here
max bon: Raju can you help me?
Sure
```

All files are up-to-date (12 minutes ago)

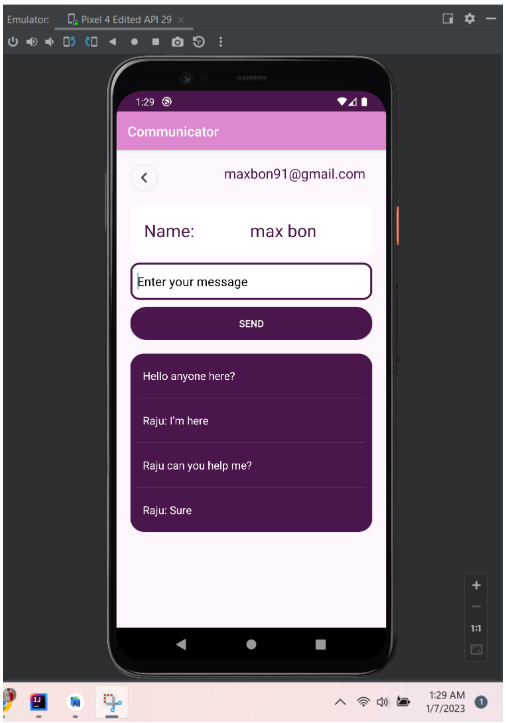
Figure 11: Server side of the Communicator

5.2 Analysis

When the application is run by the user, then the application will provide our application UI and they can able to interact with the UI and they have to login to make communication with other group members. If any user send the message in the group then application will broadcast to others users and application shows the messages on users chat screen they able to see it and they will send a message also. If any user come to the group server will notify that new users in the chat as well as when user leaves from the group everyone notified that user has leave from the group.

6 Performance

Communicator is a mobile application and performance is the most important part of the application. We have to consider memory, network, cpu and others things. In this below



maxbon91@gmail.com

Name: max bon

Enter your message

SEND

Hello anyone here?

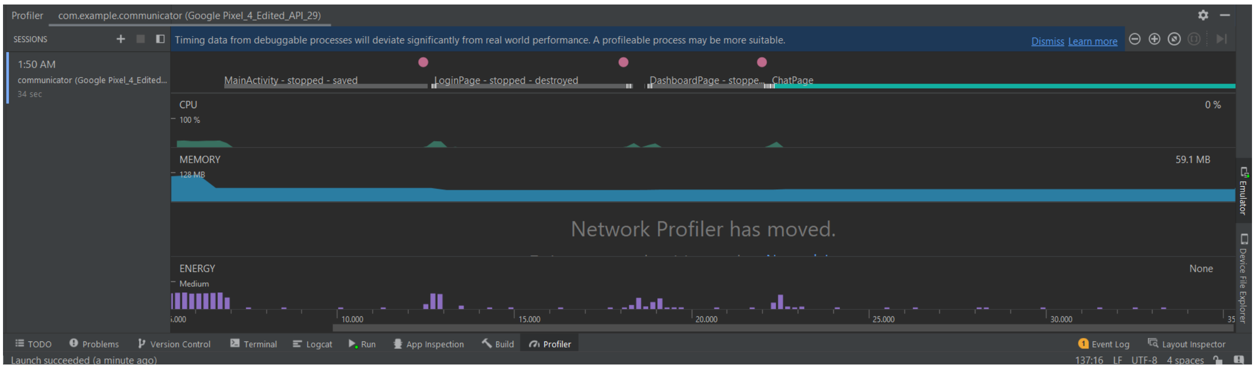
Raju: I'm here

Raju can you help me?

Raju: Sure

Figure 12: Client Chat Screen

figure we provided here the application consume the 124 MB which is quite good for us and users.



Timing data from debuggable processes will deviate significantly from real world performance. A profileable process may be more suitable.

Network Profiler has moved.

Launch succeeded (a minute ago)

When the application is run by the user then the application consume high energy at this above performance evaluation when main activity comes to run application consume high energy the it's release the energy if the any activity runs on the screen then it's consume less energy.
If we .

7 Limitation

* We are added most of the features which we needed also we faced some limitations:

- We can't delete the message if we sent once.

- Any updates to the functionality of the Application Software
- Older versions do not work with current features.

* If so many users attempt so connect it's possible to crash our server.
* If user exit from the chat room they will lose their older message.
* Not able to prevent hacking or spamming.
* This application is OS depended.
* Performance can be optimized.
* All hardware device updates, equipment re-deployments, and policy modifications must be done with the customer's mutual permission, depending on the impact on the overall security environment and network performance.
* Security can be handled more effectively.

8 Future Scope

There is a vast future scope of this Network. This Design can be improved and add more features e.g: user can direct message to anyone, user can able to share file and images, video call through WebRTC then it can be used as social media or internal company communication medium. If the limitations present in this Design are removed then, this this application will become very reliable and provide 100 percent up time.
If we added our application to the google play store then, Everyone can easily find and download our application and they can use it. In future if we added the ads revenue model in our application it we can make money from it.

9 Complex Engineering Problems that occurred

Complex Engineering Problems		
No	Name of the WP Attribute	Explain how you addressed this attribute
WP1	Depth of Knowledge Required	Knowledge is required to UI design of the system
WP2	Depth of analysis required	Analysis of the data is required to get a reliable output
WP3	Familiarity of issues	There have many of social media application this is the limitations our project