VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER NETWORKS LAB (CO3094)

Lab 2c

Socket Programming in Java
Chat Application

Advisor:	Nguyễn Mạnh Thìn
Class:	CC06
Students:	Nguyễn Minh Tâm - 1952968

HO CHI MINH CITY, OCTOBER 2021

Contents

1 Exercise 1 2

2 Exercise 2 4

3 Exercise 3 11

1 Exercise 1

Code:

```
import java.io.*;
import java.net.URL;
import java.net.MalformedURLException;

public class download {

    public static void DownloadWebPage(String webpage) {
        try {
            URL url = new URL(webpage);
            BufferedReader reader = new BufferedReader(new InputStreamReader(
    url.openStream()));
            BufferedWriter writer = new BufferedWriter(new FileWriter("Download
    .html"));
            String line;
            while ((line = reader.readLine()) != null) {
                writer.write(line);
            }
            reader.close();
            writer.close();
            System.out.println("Successfully Downloaded.");
        }

        catch (MalformedURLException mue) {
            System.out.println("Malformed URL Exception raised");
        } catch (IOException ie) {
            System.out.println("IOException raised");
        }
    }

    public static void main(String args[]) throws IOException {
        String url = "https://www.google.com/";
        DownloadWebPage(url);
    }
}
```

Result:

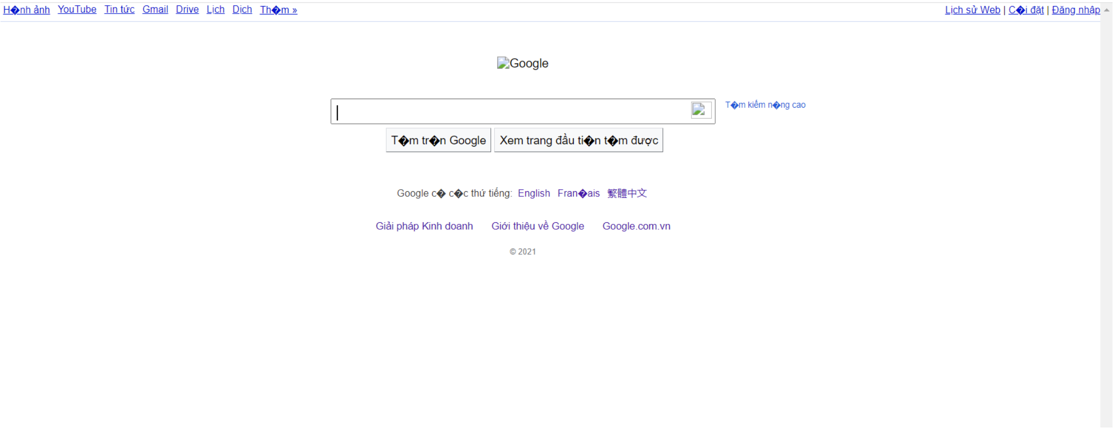
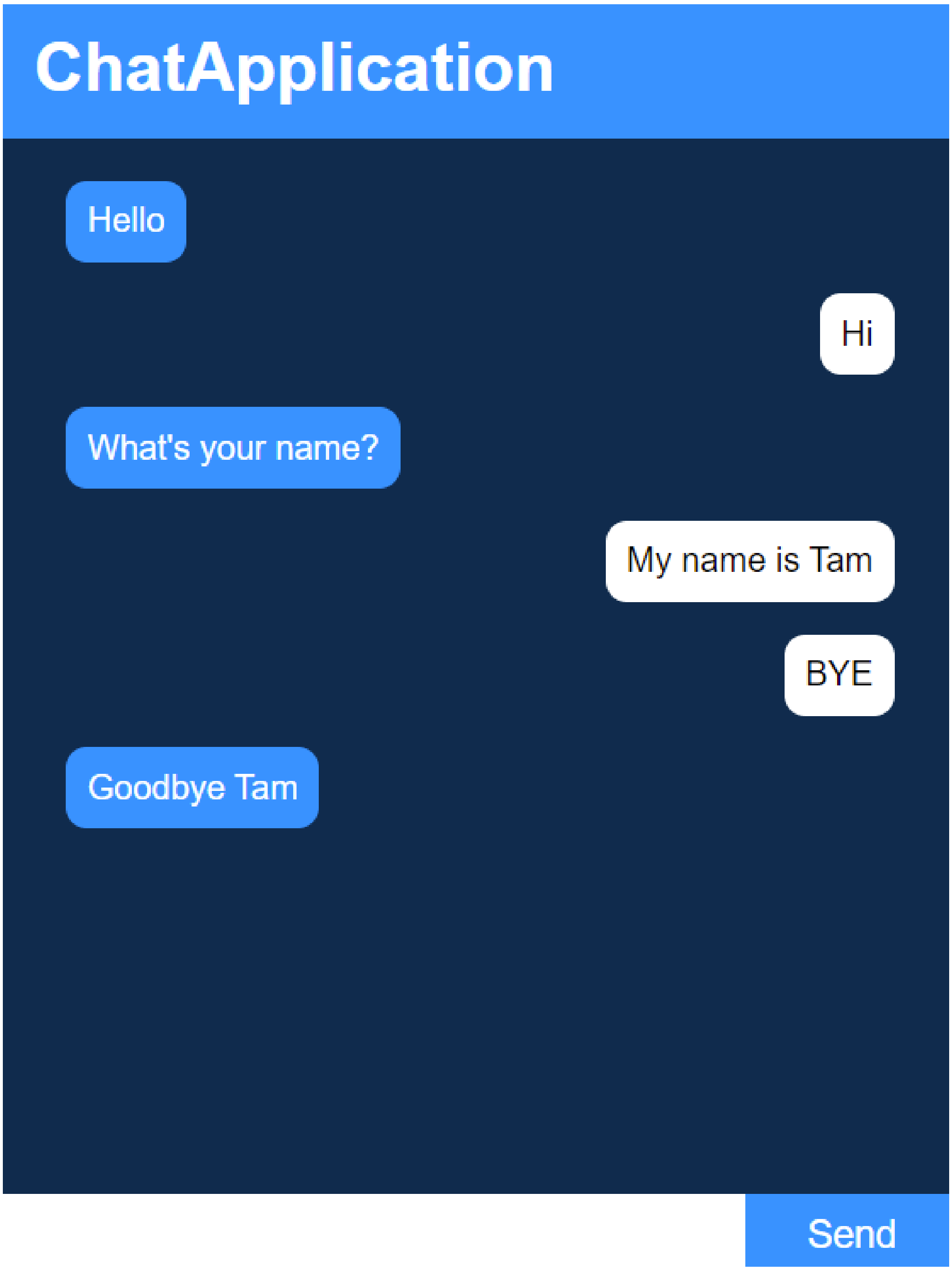

H�nh ảnh YouTube Tin tức Gmail Drive Lịch Dịch Th�m »
Lịch sử Web | C�i đặt | Đăng nhập
Google
T�m kiếm n�ng cao
T�m tr�n Google
Xem trang đầu ti�n t�m được
Google c� c�c thứ tiếng: English Fran�ais 繁體中文
Giải pháp Kinh doanh
Giới thiệu về Google
Google.com.vn
© 2021

Figure 1: Google webpage after downloading

2 Exercise 2

UI interface:



Hello

Hi

What's your name?

My name is Tam

BYE

Goodbye Tam

Figure 2: Chat application UI interface

HTML file for Chat Application:

```html
<!DOCTYPE html>
<html>
<head>
    <title>Chat</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <div class="container">
        <div class="header">
            <h1>ChatApplication</h1>
        </div>
        <div class="body">
            <p class="message">Hello</p>
            <p class="message user_message">Hi</p>
            <p class="message">What’s your name?</p>
            <p class="message user_message">My name is Tam</p>
            <p class="message user_message">BYE</p>
            <p class="message">Goodbye Tam</p>
        </div>
        <div class="footer">
            <form>
                <input type="text" name="">
                <button>Send</button>
            </form>
        </div>
    </div>
</body>
```

CSS file for Chat Application HTML:

```
* {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  box-sizing: border-box;
}
body {
  height: 100vh;
  background-color: #f8f8f8;
  display: flex;
  justify-content: center;
  align-items: center;
}
.container {
  width: 450px;
  height: 80vh;
  display: flex;
  flex-direction: column;
  box-shadow: 2px 2px 20pcx rgba(0, 0, 0, 0.4);
  background-color: dodgerblue;
}
.header h1 {
  color: white;
  padding: 15px;
}
.body {
  flex: 1;
  color: white;
  background-color: rgba(0, 0, 0, 0.7);
  padding: 20px 30px;
}
.message {
  background-color: dodgerblue;
  padding: 10px;
  color: white;
  width: fit-content;
  border-radius: 10px;
  margin-bottom: 15px;
}
.user_message {
  margin-left: auto;
  background-color: white;
  color: black;
}
.footer form {
  display: flex;
}
form input {
  flex: 1;
  height: 40px;
  border: none;
  outline: none;
  padding-left: 5px;
  font-size: 16px;
}
form button {
  width: 100px;
  font-size: 18px;
  border: none;
  outline: none;
  background-color: dodgerblue;
  color: white;
  cursor: pointer;
}
form button:hover {
  background-color: blue;
  transition: 0.2s ease;
}
```

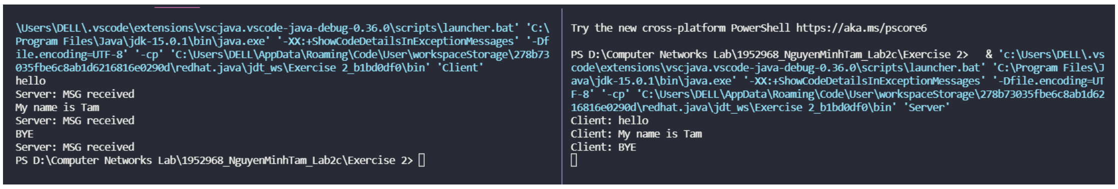
Server side:

```java
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    public static void main(String[] args) throws IOException {
        Socket socket = null;
        ServerSocket serverSocket = null;
        InputStreamReader inputStreamReader = null;
        OutputStreamWriter outputStreamWriter = null;
        BufferedReader bufferedReader = null;
        BufferedWriter bufferedWriter = null;

        serverSocket = new ServerSocket(1234);

        while (true) {
            try {
                socket = serverSocket.accept();
                inputStreamReader = new InputStreamReader(socket.getInputStream
    ());
                outputStreamWriter = new OutputStreamWriter(socket.
    getOutputStream());
                bufferedReader = new BufferedReader(inputStreamReader);
                bufferedWriter = new BufferedWriter(outputStreamWriter);
                while (true) {
                    String clientMsg = bufferedReader.readLine();
                    System.out.println("Client: " + clientMsg);
                    bufferedWriter.write("MSG received");
                    bufferedWriter.newLine();
                    bufferedWriter.flush();
                    if (clientMsg.equalsIgnoreCase("BYE"))
                        break;
                }
                socket.close();
                inputStreamReader.close();
                outputStreamWriter.close();
                bufferedReader.close();
                bufferedWriter.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Client side:

```
import java.io.*;
import java.net.Socket;
import java.util.Scanner;

public class Client {
    public static void main(String[] args) {
        Socket socket = null;
        InputStreamReader inputStreamReader = null;
        OutputStreamWriter outputStreamWriter = null;
        BufferedReader bufferedReader = null;
        BufferedWriter bufferedWriter = null;

        try {
            socket = new Socket("localhost", 1234);
            inputStreamReader = new InputStreamReader(socket.getInputStream());
            outputStreamWriter = new OutputStreamWriter(socket.getOutputStream
    ());
            bufferedReader = new BufferedReader(inputStreamReader);
            bufferedWriter = new BufferedWriter(outputStreamWriter);
            Scanner scanner = new Scanner(System.in);
            while (true) {
                String sendingMsg = scanner.nextLine();
                bufferedWriter.write(sendingMsg);
                bufferedWriter.newLine();
                bufferedWriter.flush();
                System.out.println("Server: " + bufferedReader.readLine());
                if (sendingMsg.equalsIgnoreCase("BYE"))
                    break;
            }

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (socket != null)
                    socket.close();
                if (inputStreamReader != null)
                    inputStreamReader.close();
                if (outputStreamWriter != null)
                    outputStreamWriter.close();
                if (bufferedReader != null)
                    bufferedReader.close();
                if (bufferedWriter != null)
                    bufferedWriter.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Demonstration:



```
\Users\DELL\.vscode\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\
Program Files\Java\jdk-15.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-Df
ile.encoding=UTF-8' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\278b73
035fbe6c8ab1d6216816e0290d\redhat.java\jdt_ws\Exercise 2_b1bd0df0\bin' 'Client'
hello
Server: MSG received
My name is Tam
Server: MSG received
BYE
Server: MSG received
PS D:\Computer Networks Lab\1952968_NguyenMinhTam_Lab2c\Exercise 2> ▯
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Computer Networks Lab\1952968_NguyenMinhTam_Lab2c\Exercise 2>   & 'c:\Users\DELL\.vs
code\extensions\vscjava.vscode-java-debug-0.36.0\scripts\launcher.bat' 'C:\Program Files\J
ava\jdk-15.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UT
F-8' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\278b73035fbe6c8ab1d62
16816e0290d\redhat.java\jdt_ws\Exercise 2_b1bd0df0\bin' 'Server'
Client: hello
Client: My name is Tam
Client: BYE
▯
```

Figure 3: Demonstration between Client & Server

3 Exercise 3

Server side:

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    private ServerSocket serverSocket;

    public Server(ServerSocket serverSocket) {
        this.serverSocket = serverSocket;
    }

    public void startServer() {
        try {
            while (!serverSocket.isClosed()) {
                Socket socket = serverSocket.accept();
                System.out.println("A new client has connected");
                ClientHandler clientHandler = new ClientHandler(socket);
                Thread thread = new Thread(clientHandler);
                thread.start();
            }
        } catch (IOException e) {

        }
    }

    public void closeServerSocket() {
        try {
            if (serverSocket != null)
                serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(1234);
        Server server = new Server(serverSocket);
        server.startServer();
    }
}
```

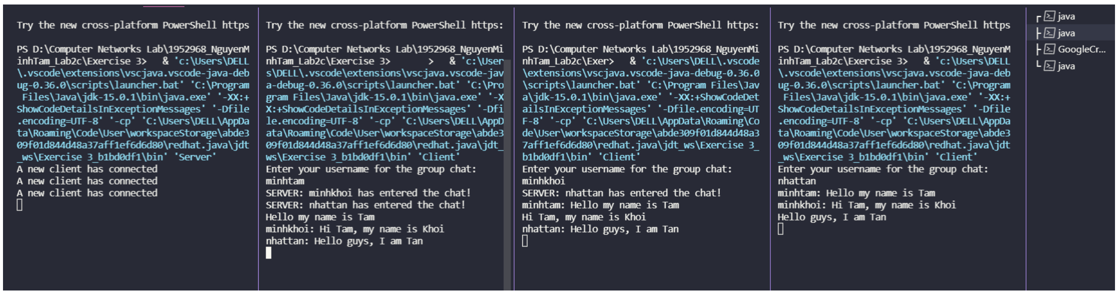
Client Handler:

```
import java.io.*;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.Socket;
import java.util.ArrayList;

public class ClientHandler implements Runnable {
    public static ArrayList<ClientHandler> clientHandlers = new ArrayList<>();
    private Socket socket;
    private BufferedReader bufferedReader;
    private BufferedWriter bufferedWriter;
    private String clientUsername;

    public ClientHandler(Socket socket) {
        try {
            this.socket = socket;
            this.bufferedWriter = new BufferedWriter(new OutputStreamWriter(
    socket.getOutputStream()));
            this.bufferedReader = new BufferedReader(new InputStreamReader(
    socket.getInputStream()));
            this.clientUsername = bufferedReader.readLine();
            clientHandlers.add(this);
            broadcastMessage("SERVER: " + clientUsername + " has entered the
    chat!");
        } catch (IOException e) {
            closeEverything(socket, bufferedReader, bufferedWriter);
        }
    }

    @Override
    public void run() {
        String messageFromClient;
        while (socket.isConnected()) {
            try {
                messageFromClient = bufferedReader.readLine();
                broadcastMessage(messageFromClient);
            } catch (IOException e) {
                closeEverything(socket, bufferedReader, bufferedWriter);
                break;
            }
        }
    }

    public void broadcastMessage(String sendingMsg) {
        for (ClientHandler clientHandler : clientHandlers) {
            try {
                if (!clientHandler.clientUsername.equals(clientUsername)) {
                    clientHandler.bufferedWriter.write(sendingMsg);
                    clientHandler.bufferedWriter.newLine();
                    clientHandler.bufferedWriter.flush();
                }
            } catch (IOException e) {
                closeEverything(socket, bufferedReader, bufferedWriter);
            }
        }
    }

    public void removeClientHandler() {
        clientHandlers.remove(this);
        broadcastMessage("SERVER: " + clientUsername + " has left the chat!");
    }

    public void closeEverything(Socket socket, BufferedReader bufferedReader,
    BufferedWriter bufferedWriter) {
        removeClientHandler();
        try {
            if (bufferedReader != null)
                bufferedReader.close();
            if (bufferedWriter != null)
                bufferedWriter.close();
            if (socket != null)
                socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client side:

```
import java.io.*;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.Socket;
import java.util.Scanner;

public class Client {
    private Socket socket;
    private BufferedReader bufferedReader;
    private BufferedWriter bufferedWriter;
    private String username;

    public Client(Socket socket, String username) {
        try {
            this.socket = socket;
            this.bufferedWriter = new BufferedWriter(new OutputStreamWriter(
    socket.getOutputStream()));
            this.bufferedReader = new BufferedReader(new InputStreamReader(
    socket.getInputStream()));
            this.username = username;
        } catch (IOException e) {
            closeEverything(socket, bufferedReader, bufferedWriter);
        }
    }

    public void sendMessage() {
        try {
            bufferedWriter.write(username);
            bufferedWriter.newLine();
            bufferedWriter.flush();
            Scanner scanner = new Scanner(System.in);
            while (socket.isConnected()) {
                String sendingMsg = scanner.nextLine();
                bufferedWriter.write(username + ": " + sendingMsg);
                bufferedWriter.newLine();
                bufferedWriter.flush();
            }
        } catch (IOException e) {
            closeEverything(socket, bufferedReader, bufferedWriter);
        }
    }

    public void listenForMessage() {
        new Thread(new Runnable() {
            @Override
            public void run() {
                String msgFromGroupChat;
                while (socket.isConnected()) {
                    try {
                        msgFromGroupChat = bufferedReader.readLine();
                        System.out.println(msgFromGroupChat);
                    } catch (IOException e) {
                        closeEverything(socket, bufferedReader, bufferedWriter)
    ;
                    }
                }
            }
        }).start();
    }

    public void closeEverything(Socket socket, BufferedReader bufferedReader,
    BufferedWriter bufferedWriter) {
        try {
            if (bufferedReader != null)
                bufferedReader.close();
            if (bufferedWriter != null)
                bufferedWriter.close();
            if (socket != null)
                socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your username for the group chat: ");
        String username = scanner.nextLine();
        Socket socket = new Socket("localhost", 1234);
        Client client = new Client(socket, username);
        client.listenForMessage();
        client.sendMessage();
    }
}
```

Demonstration:



```
Try the new cross-platform PowerShell https

PS D:\Computer Networks Lab\1952968_NguyenM
inhTam_Lab2c\Exercise 3>   & 'c:\Users\DELL
\.vscode\extensions\vscjava.vscode-java-deb
ug-0.36.0\scripts\launcher.bat' 'C:\Program
 Files\Java\jdk-15.0.1\bin\java.exe' '-XX:+
ShowCodeDetailsInExceptionMessages' '-Dfile
.encoding=UTF-8' '-cp' 'C:\Users\DELL\AppDa
ta\Roaming\Code\User\workspaceStorage\abde3
09f01d844d48a37aff1ef6d6d80\redhat.java\jdt
_ws\Exercise 3_b1bd0df1\bin' 'Server'
A new client has connected
A new client has connected
A new client has connected
```

```
Try the new cross-platform PowerShell https:

PS D:\Computer Networks Lab\1952968_NguyenMi
nhTam_Lab2c\Exercise 3>       >   & 'c:\User
s\DELL\.vscode\extensions\vscjava.vscode-jav
a-debug-0.36.0\scripts\launcher.bat' 'C:\Pro
gram Files\Java\jdk-15.0.1\bin\java.exe' '-X
X:+ShowCodeDetailsInExceptionMessages' '-Dfi
le.encoding=UTF-8' '-cp' 'C:\Users\DELL\AppD
ata\Roaming\Code\User\workspaceStorage\abde3
09f01d844d48a37aff1ef6d6d80\redhat.java\jdt_
ws\Exercise 3_b1bd0df1\bin' 'Client'
Enter your username for the group chat:
minhtam
SERVER: minhkhoi has entered the chat!
SERVER: nhattan has entered the chat!
Hello my name is Tam
minhkhoi: Hi Tam, my name is Khoi
nhattan: Hello guys, I am Tan
```

```
Try the new cross-platform PowerShell https:

PS D:\Computer Networks Lab\1952968_NguyenMi
nhTam_Lab2c\Exer>   & 'c:\Users\DELL\.vscode
\extensions\vscjava.vscode-java-debug-0.36.0
\scripts\launcher.bat' 'C:\Program Files\Jav
a\jdk-15.0.1\bin\java.exe' '-XX:+ShowCodeDet
ailsInExceptionMessages' '-Dfile.encoding=UT
F-8' '-cp' 'C:\Users\DELL\AppData\Roaming\Co
de\User\workspaceStorage\abde309f01d844d48a3
7aff1ef6d6d80\redhat.java\jdt_ws\Exercise 3_
b1bd0df1\bin' 'Client'
Enter your username for the group chat:
minhkhoi
SERVER: nhattan has entered the chat!
minhtam: Hello my name is Tam
Hi Tam, my name is Khoi
nhattan: Hello guys, I am Tan
```

```
Try the new cross-platform PowerShell https

PS D:\Computer Networks Lab\1952968_NguyenM
inhTam_Lab2c\Exercise 3>   & 'c:\Users\DELL
\.vscode\extensions\vscjava.vscode-java-deb
ug-0.36.0\scripts\launcher.bat' 'C:\Program
 Files\Java\jdk-15.0.1\bin\java.exe' '-XX:+
ShowCodeDetailsInExceptionMessages' '-Dfile
.encoding=UTF-8' '-cp' 'C:\Users\DELL\AppDa
ta\Roaming\Code\User\workspaceStorage\abde3
09f01d844d48a37aff1ef6d6d80\redhat.java\jdt
_ws\Exercise 3_b1bd0df1\bin' 'Client'
Enter your username for the group chat:
nhattan
minhtam: Hello my name is Tam
minhkhoi: Hi Tam, my name is Khoi
Hello guys, I am Tan
```

Figure 4: Chat Application allows multiple users talking to each other concurrently