

Project Name: Chat Application

Table of Contents

Chapter 1 Introduction	3
1.1 Introduction.....	3
1.2 Design Goals/Objective	3
 Chapter 2 Design/Development/Implementation of the Project.....	4-9
2.1 Section (Choose the name of this section as appropriate with your project)Error! Bookmark not defined.	
2.2.1 Subsection.....	Error! Bookmark not defined.-9
 Chapter 3 Performance Evaluation	10-14
3.1 Simulation Environment/ Simulation Procedure.....	10-12
3.2 Results and Discussions	13-14
 Chapter 4 Conclusion.....	15
4.1 Introduction.....	15
4.2 Scope of Future Work.....	15
 References.....	16

Chapter 1

Introduction

1.1 Introduction

Our Chat Application made in java using concepts of Java Networking and Socket Programming that allows communication between a server and client. The application has an easy to use User Interface which has been created using Java Swing.

As we use socket here, we need to know “What is Socket?”

A socket is one endpoint of a two-way communication link between two programs running on a network. It is one of the most fundamental technologies of computer network programming. Sockets allow network software applications to communicate using standard mechanisms built into network hardware and operating systems. A socket represents a single connection between exactly two pieces of software (a so-called point-to-point connection). More than two pieces of software can communicate with client/server or distributed systems by using multiple sockets.

1.2 Design Goals/Objective

- This project will be developed in java. The project is planned to introduce an online chat system solution.
- A real time communication chat system will in the project to make a in-time communication channel.
- This project is to develop a java chat application.
- It will be an instant messaging solution to enable users to seamlessly communicate with each other.
- The project should be very easy to use enabling even a beginner to use it.

Chapter 2

Design/Development/Implementation of the Project

2.1 Section

2.1.1 Subsection

Server.Java Code:

```
import java.net.*;

import java.io.*;
class Server
{

    ServerSocket Server;
    Socket socket;
    BufferedReader br;
    PrintWriter out;

    public Server()
    {
        try {
            Server=new ServerSocket(7777);
            System.out.println("Server is ready to accept connection");
            System.out.println("Waiting...");
            socket=Server.accept();

            br=new BufferedReader(new

            InputStreamReader(socket.getInputStream())); out=new

            PrintWriter(socket.getOutputStream());

            startReading();
            startWriting();

        } catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

```
}
```

```
public void startReading()
```

```
{
```

4 | Page

```
//thread-read kore kore data dite thakbe
```

```
Runnable r1=()-> {
```

```
    System.out.println("Reader Started..");
```

```
    try {
```

```
        while(true) {
```

```
            String msg=br.readLine();
```

```
            if(msg.equals("exit")){
```

```
                System.out.println("Client terminated the chat");
```

```
                socket.close();
```

```
                break;
```

```
            }
```

```
            System.out.println("Client: "+msg);
```

```
        }
```

```
    } catch (Exception e) {
```

```
        //e.printStackTrace();
```

```
        System.out.println("Connection is closed");
```

```
    }
```

```
};
```

```
new Thread(r1).start();
```

```
}
```

```
public void startWriting()
```

```
{
```

```
//thread-data user er theke nibe and clint porjonto send korte
```

```
thkbe Runnable r2=()-> {
```

```
    System.out.println("Writer started...");
```

```
    try {
```

```
        while(!socket.isClosed())
```

```
        {
```

```
            BufferedReader br1= new BufferedReader(new
```

```
InputStreamReader(System.in)); String content=br1.readLine();
```

```
out.println(content);  
out.flush();
```

```
if (content.equals("exit")){  
    socket.close();
```

5 | Page

```
    break;
```

```
}
```

```
}
```

```
} catch (Exception e) {  
    //e.printStackTrace();
```

```
System.out.println("Connection is closed");
```

```
}
```

```
System.out.println("Connection is  
closed"); };
```

```
new Thread(r2).start();
```

```
}
```

```
public static void main (String[] args) {
```

```
    System.out.println("This is Server...");
```

```
    new Server();
```

```
}
```

```
}
```

2.1.2 Subsection

Client.Java Code:

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

public class Client {

    Socket socket;

    BufferedReader br;
    PrintWriter out;

    public Client()
    {

        try {
            System.out.println("Sending Request to Server");
            socket=new Socket("127.0.0.1",7777);
            System.out.println("Connection done.");

            br=new BufferedReader(new
            InputStreamReader(socket.getInputStream())); out=new
            PrintWriter(socket.getOutputStream());
```

```
startReading();
startWriting();
```

```
    } catch (Exception e) {
        //TODO: handle exception
    }
```

```
}
```

```
public void startReading()
{
    //thread-read kore kore data dite thakbe
    Runnable r1=()-> {
        System.out.println("Reader Started..");

        try{

            while(true)
            {
```

7 | Page

```
        String msg=br.readLine();
        if(msg.equals("exit"))
        {
            System.out.println("Server terminated the chat");
            socket.close();
            break;
        }
```

```
        System.out.println("Server: "+msg);
```

```
    }
```

```
} catch (Exception e) {
    //e.printStackTrace();
    System.out.println("Connection is closed");
```

```
}
```

```
};
```

```
new Thread(r1).start();
```

```
}
```

```
public void startWriting()
```

```

{
    //thread-data user er theke nibe and clint porjonto send korte
    thkbe Runnable r2=()-> {

        System.out.println("Writer started...");

        try{

            while(!socket.isClosed())
            {

                BufferedReader br1= new BufferedReader(new
                InputStreamReader(System.in)); String content=br1.readLine();

                out.println(content);
                out.flush();

                if (content.equals("exit")){
                    socket.close();
                    break;
                }

            }

            System.out.println("Connection is closed");
        } catch (Exception e) {
            e.printStackTrace();
        }

    };

    new Thread(r2).start();

}

public static void main (String[] args){

    System.out.println("This is
    Client..."); new Client();

}

}

```


Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

Server Part: (Console base)

```
C:\Users\Galaxy Computer\Documents\ChatAPP>javac Server.java

C:\Users\Galaxy Computer\Documents\ChatAPP>java Server
This is Server...
Server is ready to accept connection
Waiting...
Reader Started..
Writer started...
Client: hi
Client: hlw
hi
hlwww how are you
```

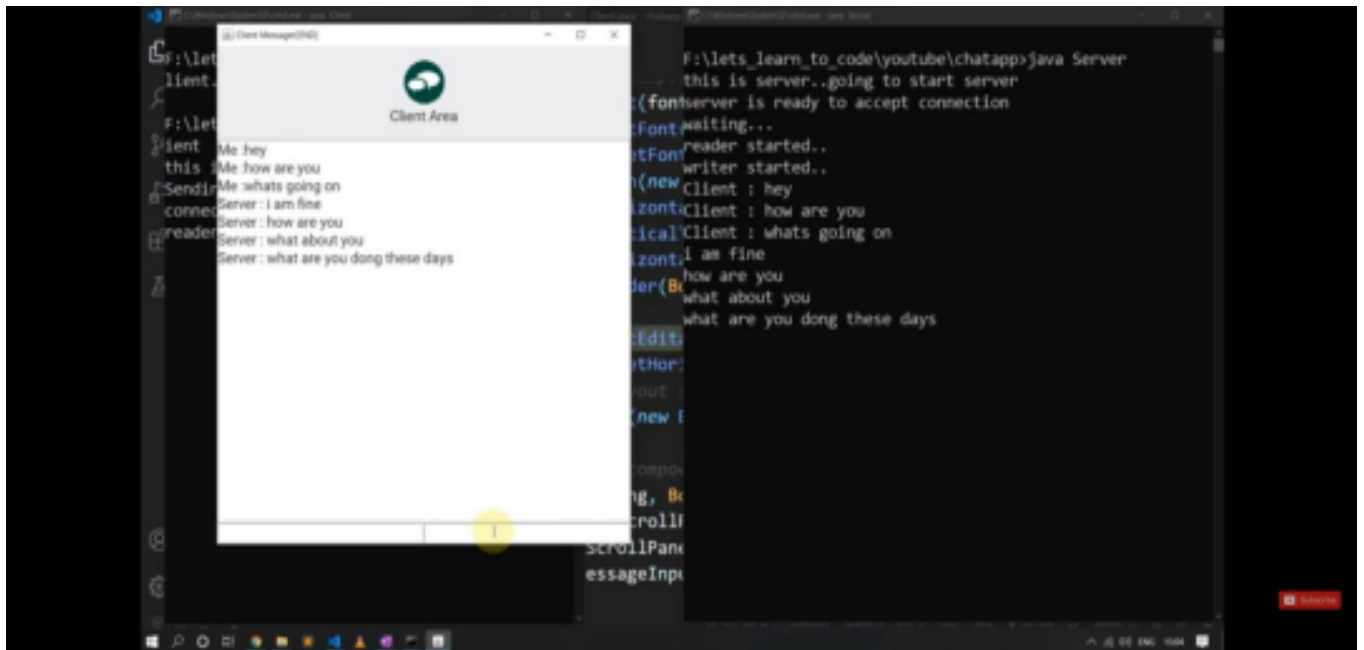
```
C:\Windows\System32\cmd.exe - java Client

C:\Users\Galaxy Computer\Documents\ChatAPP>javac Client.java

C:\Users\Galaxy Computer\Documents\ChatAPP>java client
Error: Could not find or load main class client
Caused by: java.lang.NoClassDefFoundError: Client (wrong name: cl
ient)

C:\Users\Galaxy Computer\Documents\ChatAPP>java Client
This is Client...
Sending Request to Server
Connection done.
Reader Started..
Writer started...
hi
hlw
Server: hi
Server: hlwww how are you
```

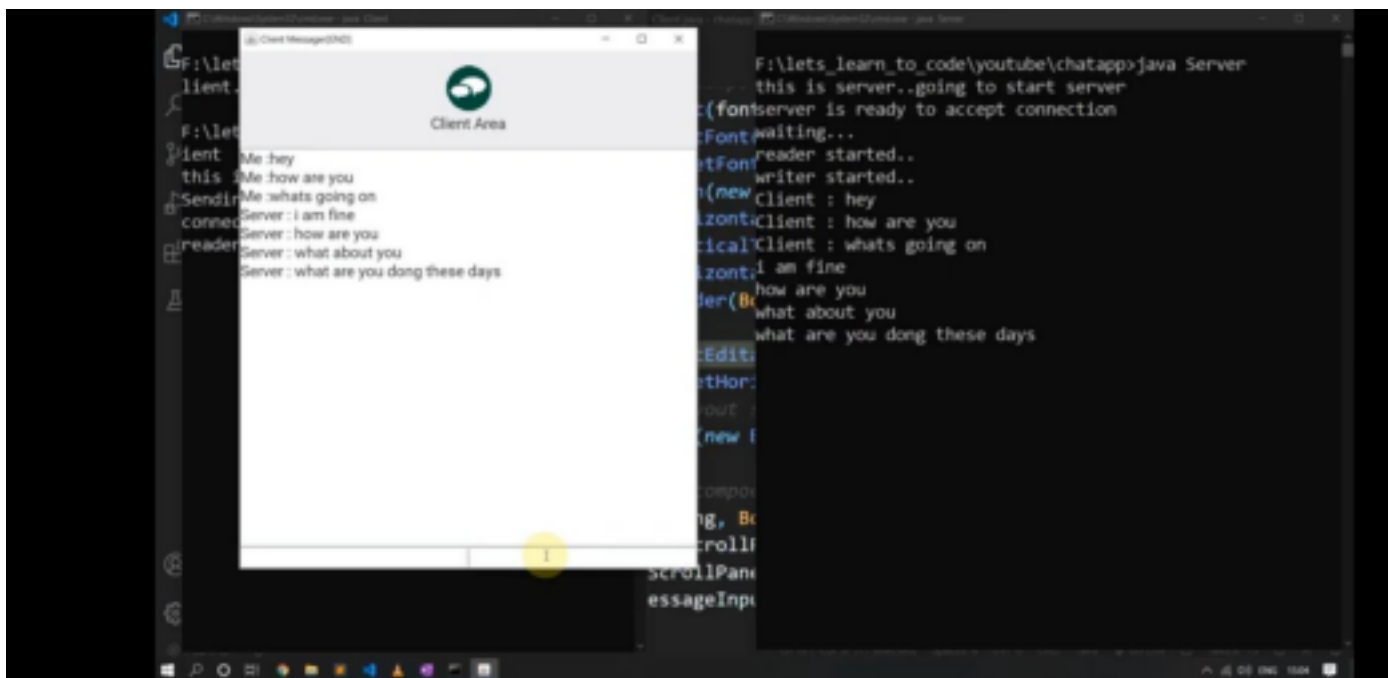
Client part: (GUI base)



3.2 Results and Discussions

3.2.1 Results

Final Result of the project is:



3.2.2 Analysis and Outcome

The main objective of the project is to develop a Secure Chat Application. I had taken a wide range of literature review in order to achieve all the tasks, where I came to know about some of the products that are existing in the market. I made a detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in our application. In the process of research I came to know about the latest technologies and different algorithms.

Conclusion

4.1 Introduction

It is a simple chat programmed in Java and Swing, using sockets. There is a client and a server, which can send messages between them using a username. The server defines the IP and the port, and the client needs to specify it in order to talk to the desired server. Once connected, they will be able to send messages between them.

4.1 Scope of Future Work

With the knowledge I have gained by developing this application, I am confident that in the future I can make the application more effectively by adding this services.

- Extending this application by providing Authorisation service.
- Creating Database and maintaining users.
- Increasing the effectiveness of the application by providing Voice Chat.
- Extending it to Web Support.

References

[1] <https://www.wikipedia.org/>

[2] <https://www.youtube.com/>

[3] <https://1000projects.org/conclusion-and-future-scope-of-secure-chat-java-application.html>