

# SWE1007

Programming in Java

J-COMPONENT PROJECT – FINAL REVIEW

**SUBMITTED TO:**

**NIRMALA M**

#### SUBMITTED BY:

#### NAME: MOHAMED RAYAN

#### REG.NO: 20MIS0098

**TITLE:**

**NOVA: CHATTING APPLICATION**

|  |  |  |
| --- | --- | --- |
| S.NO: |  | PAGE.NO: |
| 1. | Abstract | 02 |
| 2. | Introduction | 03 |
| 3. | Main Objective | 04 |
| 4. | Requirement | 06 |
| 5. | Operational Concepts and Scenarios | 08 |
| 6. | Project Scope | 10 |
| 7. | Server | 10 |
| 8. | Client | 12 |
| 9. | Future work | 13 |
| 10. | Conclusion | 16 |

**INDEX:**

**Abstract:**

Teleconferencing or chatting refers to any kind of communication that offers a real-time transmission of messages from sender to the receiver. Chatting is a method of using technology to bring people and ideas together despite the geographical barriers. The technology to provide the chatting facility has been available for years, but the acceptance is quite recent. Analysis of chatting provides an overview of the technologies used, available features, functions, system of the architecture of the application, the structure of database of an Instant Messaging application. The objective of NOVA[An chatting app] application is to facilitate text messaging, audio calling, video calling, data transfer without size restriction which is commonly seen in most of the messaging applications.

**INTRODUCTION:**

Chatting Application is a desktop based application.

This client server chat application is based on java swing, jdbc and socket programming.

This application/program is a good example of using java.io, java.net , java.sql ,java.awt  package to create a chat application.

Chatting is a method of using technology to bring people and ideas “together” despite of the geographical barriers. The technology has been available for years but the acceptance it was quit recent. Our project is an example of a multiple client chat server.

It is made up of 2 applications the client application, which runs on the user’s Pc and server application, which runs on any Pc on the network. To start chatting client should get connected to server. We will focus on TCP and UDP socket connections which are a fundamental part of socket programming.

Keywords: sockets, client-server, Java network programming-socket functions, Multicasting etc.

**MAIN OBJECTIVE:**

The aim of this project is to express how we can implement a simple chat application between a server and a client? The application is a desktop based application and is implemented using java Swing, java networking, jdbc and awt. The project is developed in Java language executed on a single stand-alone java across a network using loop back address concept.

Application consists of two programs:

Server

Client

**Server:**

The server module of the application waits for the client to connect to it. If the connection is granted, clients can interacts and communicate in network and it can also communicate with the server. The duty of the server is to let clients exchange the messages.

**Client:**

The client module is the one that utilizer sends requests to the server. Utilizer utilizes the client as the means to connect to the server. Once client establishes the connection, it can communicate to the connected server.

**REQUIREMENTS:**

External Interface Requirements

**User Interface:**

The user interface required to be developed for the system should be user friendly and attractive.

There are two sets of Java APIs for graphics programming:

AWT (Abstract Windowing Toolkit) and Swing.

1. AWT API was introduced in JDK 1.0. Most of the AWT components have become obsolete and should be replaced by newer Swing components.
2. Swing API, a much more comprehensive set of graphics libraries that enhances the AWT, was introduced as part of Java Foundation Classes (JFC) after the release of JDK 1.1. JFC consists of Swing, Java2D, Accessibility, Internationalization, and Pluggable Look-and-Feel Support APIs. JFC was an add-on to JDK 1.1 but has been integrated into core Java since JDK 1.2.

**Software Interfaces :**

Programming Language - Java

And socket programming.

**Operational Concepts and Scenarios:**Operation of the application based on the inputs given by the user:

When the run button is clicked

* A login form will initialized, the user has to enter the valid information in order to get the validation
* And then the chat form is initialized with a connection between the host and the client machine.
  + Note: server must be started at first before a client start.
* Contains a rich textbox which send messages from one user to another
* Contains a textbox for messages to be written that is sent across the network. Contains a Send button
* When the sent button is clicked, in the background, the text in the textbox is encoded and sent as a packet over the network to the client machine. Here this message is decoded and is shown in the rich textbox.
* User can interact in the network not only through text message but also through the Video call and Audio call by clicking the audio call button or video call button.
* It has the feature of getting back up of data.
* User can save their data in database as long as they want by clicking on the set backup button.
* They can view there backup data by clicking on get backup button.
* If user enter any invalid values (i.e. username and password) the programme will invoke an error page which will notify the user that they have enter invalid values

**Project Scope:**

This project can be mainly divided into two modules:

1. Server

2. Client

This project is mainly depended on client/server model. The client requests the server and server responses by granting the clients request. The proposed system should provide both of the above features along with the followed ones:

**Server:**

A server is a computer program that provides services to other computer programs (and their users) in the same or other computers. The computer that a server program runs in is also frequently referred to as a server. That machine may be a dedicated server or used for other purposes as well. Example Server, Database, Fileserver, Proxy Server, Web Server. The server will always waiting for client’s requests. The client start and stop at any time but the server remains in the same state.

A server application normally listens to a specific port waiting for connection requests from a client. When a connection request arrives, the client and the server establish a dedicated connection over which they can communicate. During the connection process, the client is assigned a local port number, and binds a socket to it. The client talks to the server by writing to the socket and gets information from the server by reading from it. Similarly, the server gets a new local port number (it needs a new port number so that it can continue to listen for connection requests on the original port). The server also binds a socket to its local port and communicates with the client by reading from and writing to it. The client and the server must agree on a protocol that is, they must agree on the language of the information transferred back and forth through the socket. Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

**Client:**

On the client site the client knows the hostname of the machine on which the server is running and the port number on which the server is listening.

To make a connection request, the client tries to rendezvous with the server on the server's machine and port. The client also needs to identify itself to the server so it binds to a local port number that it will use during this connection. This is usually assigned by the system.

**Future work:**

There is always a room for improvements in any software package, however good and efficient it may be done. But the most important thing should be flexible to accept further modification. Right now we are just dealing with text, audio and video communication. In future this software may be extended to include features such as:

Files transfer: this will enable the user to send files of different formats to others via the chat application.

Sharing of video: this will enable the user to send videos files of different formats to others via the chat application.

Sharing of audio: this will enable the user to send audio files of different formats to others via the chat application.

Sharing of images: this will enable the user to send images of different formats to others via the chat application.

**Conclusion :**

I Developed network applications in Java by using sockets, threads, and Web services.

These software is portable, efficient, and easily maintainable for large number of clients. Our developed web-based chatting software is unique in its features and more importantly easily customizable.

The java.net package provides a powerful and flexible set of classes for implementing network applications. Typically, programs running on client machines make requests to programs on a server Machine. These involve networking services provided by the transport layer. The most widely used transport protocols on the Internet are TCP (Transmission control Protocol) and UDP (User Datagram Protocol).

TCP is a connection-oriented protocol providing a reliable flow of data between two computers. On the other hand, UDP is a simpler message-based connectionless protocol which sends packets of data known as datagrams from one computer to another with no guarantees of arrival.

**CODE:**

**SERVER CODE:**

package com.chatting\_application;

import javax.swing.\*;

import java.awt.\*;

import java.net.ServerSocket;

import java.net.Socket;

import java.io.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.lang.\*;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//=== === === === === === BACKUP MESSAGES === === === === === === === === === === === === === === === === === === === === ===//

class backup extends JFrame {

JPanel p1;

JTextField t1;

JButton b1;

static JTextArea tf1;

JScrollPane sb;

JLabel l5;

public backup() {

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(7, 94, 84));

p1.setBounds(0, 0, 400, 70);

add(p1);

// Download - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/download.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

//Name - text no.01

JLabel l6 = new JLabel("Backed up messages");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 400, 40);

p1.add(l6);

//close button

b1 = new JButton("Close");

b1.setBounds(310, 555, 85, 40);

b1.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

b1.setForeground(Color.white);

b1.setBackground(new Color(7, 94, 84));

add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

//text field

// display area.

tf1 = new JTextArea();

tf1.setFont(new Font("SAN\_SERIF", Font.PLAIN, 14));

tf1.setBackground(Color.white);

tf1.setEditable(false);

tf1.setLineWrap(true);

tf1.setWrapStyleWord(true);

// adding scroll bar

sb = new JScrollPane(tf1);

sb.setBounds(5, 75, 390, 475);

sb.addNotify();

add(sb);

String BackedData = null;

String BackedDate = null;

String[] Backed\_up = new String[1000];

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

ResultSet resultSet = sts.executeQuery("select \* from chat\_backup");

while(resultSet.next()){

BackedData = resultSet.getString("chats");

BackedDate = resultSet.getString("date\_chats");

tf1.setText(tf1.getText()+"\n" + "Text: "+BackedData+" || Date: "+BackedDate);

}

System.out.println(BackedData);

System.out.println(BackedDate);

con.close();

} catch (Exception ae) {

ae.printStackTrace();

}

setLayout(null);

//Window box - Server

setSize(400, 600);

setLocation(250, 90);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

}

//=== === === === === === CALLING === === === === === === === === === === === === === === === === === === === === ===

class calling extends JFrame {

calling(String a){

JPanel p1;

JPanel p2;

JPanel p3;

JTextField t1;

JButton b1;

JTextArea tf1;

JScrollPane sb;

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(7, 94, 84));

p1.setBounds(0, 0, 400, 70);

add(p1);

// exit button - icon no.01

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3.png"));

Image i2 = i1.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(5, 25, 20, 20);

p1.add(l1);

//mouseevent for exit

l1.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent ae) {

System.exit(0);

}

});

// DP - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/elon.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

// Video call - icon no.02

ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/video.png"));

Image i8 = i7.getImage().getScaledInstance(30, 30, Image.SCALE\_DEFAULT);

ImageIcon i9 = new ImageIcon(i8);

JLabel l3 = new JLabel(i9);

l3.setBounds(230, 15, 40, 40);

p1.add(l3);

//Audio call - icon no.03

ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/phone.png"));

Image i11 = i10.getImage().getScaledInstance(33, 33, Image.SCALE\_DEFAULT);

ImageIcon i12 = new ImageIcon(i11);

JLabel l4 = new JLabel(i12);

l4.setBounds(290, 16, 40, 40);

p1.add(l4);

//More options - icon no.04

ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3icon.png"));

Image i14 = i13.getImage().getScaledInstance(12, 22, Image.SCALE\_DEFAULT);

ImageIcon i15 = new ImageIcon(i14);

JLabel l5 = new JLabel(i15);

l5.setBounds(335, 15, 40, 40);

p1.add(l5);

//Name - text no.01

JLabel l6 = new JLabel("Elon Musk");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 100, 40);

p1.add(l6);

p2 = new JPanel();

p2.setLayout(null);

p2.setBackground(new Color(7, 94, 84));

p2.setBounds(0, 510, 430, 100);

add(p2);

ImageIcon i44 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/deline button.png"));

Image i55 = i44.getImage().getScaledInstance(120, 120, Image.SCALE\_DEFAULT);

ImageIcon i66 = new ImageIcon(i55);

JLabel l22 = new JLabel(i66);

l22.setBounds(30, 20, 60, 60);

p2.add(l22);

l22.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

System.exit(0);

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

ImageIcon i45 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/speaker button.png"));

Image i56 = i45.getImage().getScaledInstance(70, 70, Image.SCALE\_DEFAULT);

ImageIcon i67 = new ImageIcon(i56);

JLabel l23 = new JLabel(i67);

l23.setBounds(130, 20, 60, 60);

p2.add(l23);

ImageIcon i46 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/mute button.png"));

Image i57 = i46.getImage().getScaledInstance(20, 30, Image.SCALE\_DEFAULT);

ImageIcon i68 = new ImageIcon(i57);

JLabel l24 = new JLabel(i68);

l24.setBounds(215, 20, 60, 60);

p2.add(l24);

ImageIcon i47 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/Video Call Mute.png "));

Image i58 = i47.getImage().getScaledInstance(35, 35, Image.SCALE\_DEFAULT);

ImageIcon i69 = new ImageIcon(i58);

JLabel l25 = new JLabel(i69);

l25.setBounds(300, 20, 60, 60);

p2.add(l25);

p3 = new JPanel();

p3.setLayout(null);

p3.setBounds(0, 100, 430, 400);

add(p3);

// DP - icon no.02

ImageIcon i444 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/astronaut.png"));

Image i555 = i444.getImage().getScaledInstance(250, 250, Image.SCALE\_DEFAULT);

ImageIcon i666 = new ImageIcon(i555);

JLabel l222 = new JLabel(i666);

l222.setBounds(50, 5, 300, 300);

p3.add(l222);

JLabel call = new JLabel("Connecting.....");

call.setBackground(Color.black);

call.setBounds(140,250,200,100);

call.setFont(new Font(Font.SANS\_SERIF,Font.BOLD,18));

p3.add(call);

JLabel call1 = new JLabel(a);

call1.setBackground(Color.black);

call1.setBounds(160,280,200,100);

call1.setFont(new Font(Font.SANS\_SERIF,Font.BOLD,16));

p3.add(call1);

setLayout(null);

//Window box - Server

setSize(400, 600);

setLocation(200, 70);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

}

//===Error Thrower === === === === === === === === === === === === === === === === === === === === === === === === ===//

class Error\_thrower\_server1 extends JFrame{

JLabel lb1;

JButton b1;

JLabel l2;

JLabel l3;

JLabel l4;

public Error\_thrower\_server1() {

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/1.1.png"));

Image i2 = i1.getImage().getScaledInstance(450,500,Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(0,0,0,0);

add(l1);

l2 = new JLabel("Incorrect");

l2.setBounds(2,2,150,150);

l2.setForeground(Color.red);

l2.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l2.setLocation(240,208);

l1.add(l2);

l3 = new JLabel("Username");

l3.setBounds(2,2,150,150);

l3.setForeground(Color.red);

l3.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l3.setLocation(240,256);

l1.add(l3);

l4 = new JLabel("Password");

l4.setBounds(2,2,150,150);

l4.setForeground(Color.red);

l4.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l4.setLocation(240,305);

l1.add(l4);

lb1 = new JLabel("ERROR");

lb1.setBounds(2,2,300,300);

lb1.setForeground(Color.red);

lb1.setFont(new Font("SAN\_SERIF",Font.BOLD,18));

lb1.setLocation(227,105);

l1.add(lb1);

b1 = new JButton("Close");

b1.setBounds(213,410, 100, 30);

b1.setForeground(Color.white);

b1.setBackground(new Color(7, 94, 84));

l1.add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

setSize(450,550);

setLocation(450,120);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setUndecorated(true);

setVisible(true);

}

}

//===LOGIN FORM === === === === === === === === === === === === === === === === === === === === === === === === ===//

class login extends JFrame {

JPanel p1;

JLabel l1;

JLabel l2;

JTextField t1;

JLabel l3;

JPasswordField t2;

JButton b1;

JButton b2;

JLabel l4;

JLabel l5;

JLabel l6;

JTextField r;

JTextField g;

JTextField b;

JLabel th;

login() {

p1 = new JPanel();

p1.setBounds(0, 0, 1000, 500);

p1.setBackground(new Color(7, 94, 84));

p1.setLayout(null);

add(p1);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/loginback.jpg"));

Image i2 = i1.getImage().getScaledInstance(1000, 500, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel bg = new JLabel(i3);

bg.setBounds(0, 0, 1000, 500);

p1.add(bg);

l1 = new JLabel("Login Form");

l1.setBounds(645, -190, 1000, 500);

l1.setFont(new Font("SANS\_SERIF", Font.BOLD, 28));

l1.setForeground(Color.white);

bg.add(l1);

l2 = new JLabel("Email:");

l2.setBounds(600, -130, 1000, 500);

l2.setFont(new Font("SANS\_SERIF", Font.BOLD, 18));

l2.setForeground(Color.white);

bg.add(l2);

t1 = new JTextField("Email");

t1.setBounds(600, 150, 250, 30);

t1.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

t1.setForeground(Color.black);

bg.add(t1);

l3 = new JLabel("Password:");

l3.setBounds(600, -25, 1000, 500);

l3.setFont(new Font("SANS\_SERIF", Font.BOLD, 18));

l3.setForeground(Color.white);

bg.add(l3);

t2 = new JPasswordField("Password");

t2.setBounds(600, 255, 250, 30);

t2.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

t2.setForeground(Color.black);

bg.add(t2);

b1 = new JButton("LogIn");

b1.setBounds(615, 320, 100, 30);

b1.setForeground(Color.black);

b1.setBackground(Color.white);

bg.add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

ResultSet resultSet = sts.executeQuery("select \* from users where name = '" + t1.getText() + "' and password = '" + t2.getText().toString() + "'");

int x = Integer.parseInt(r.getText());

int y = Integer.parseInt(g.getText());

int z = Integer.parseInt(b.getText());

if (resultSet.next()) {

new Server1(x,y,z).setVisible(true);

setVisible(false);

} else {

Error\_thrower\_server1 ets = new Error\_thrower\_server1();

}

con.close();

} catch (Exception ae) {

ae.printStackTrace();

}

}

});

b1.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

public void mouseReleased(MouseEvent e) {

}

public void mouseEntered(MouseEvent ae) {

b1.setBackground(new Color(58, 111, 89));

}

@Override

public void mouseExited(MouseEvent ae) {

b1.setBackground(Color.white);

}

});

b2 = new JButton("Cancel");

b2.setBounds(735, 320, 100, 30);

b2.setForeground(Color.black);

b2.setBackground(Color.white);

bg.add(b2);

b2.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

System.exit(0);

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {

b2.setBackground(new Color(58, 111, 89));

}

public void mouseExited(MouseEvent e) {

b2.setBackground(Color.white);

}

});

l4 = new JLabel("N🌖VA");

l4.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 70));

l4.setBounds(200, -80, 500, 500);

l4.setForeground(Color.white);

bg.add(l4);

l5 = new JLabel("Chatting");

l5.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 30));

l5.setBounds(200, -0, 500, 500);

l5.setForeground(Color.white);

bg.add(l5);

l6 = new JLabel("App");

l6.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 30));

l6.setBounds(200, 25, 500, 500);

l6.setForeground(Color.white);

bg.add(l6);

th = new JLabel("Theme:");

th.setBounds(600, 310, 100, 200);

th.setFont(new Font("SANS\_SERIF", Font.BOLD, 18));

th.setForeground(Color.white);

bg.add(th);

r = new JTextField("7");

r.setBounds(615, 440, 40, 30);

r.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

r.setForeground(Color.black);

bg.add(r);

g = new JTextField("94");

g.setBounds(705, 440, 40, 30);

g.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

g.setForeground(Color.black);

bg.add(g);

b = new JTextField("84");

b.setBounds(795, 440, 40, 30);

b.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

b.setForeground(Color.black);

bg.add(b);

setSize(1000, 500);

setLocation(200, 130);

setLayout(null);

setUndecorated(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setVisible(true);

}

}

//------------------------------------------------------------------------------------------------------------------------//

public class Server1 extends JFrame implements ActionListener {

JPanel p1;

JTextField t1;

JButton b1;

static JTextArea tf1;

JScrollPane sb;

JLabel l5;

//network

static ServerSocket skt;

static Socket s;

static DataInputStream din;

static DataOutputStream dout;

public Server1(int r, int g, int b) {

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(r,g,b));

p1.setBounds(0, 0, 400, 70);

add(p1);

// exit button - icon no.01

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3.png"));

Image i2 = i1.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(5, 25, 20, 20);

p1.add(l1);

//mouseevent for exit

l1.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent ae) {

System.exit(0);

}

});

// DP - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/elon.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

// Video call - icon no.02

ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/video.png"));

Image i8 = i7.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i9 = new ImageIcon(i8);

JLabel l3 = new JLabel(i9);

l3.setBounds(220, 20, 30, 30);

p1.add(l3);

l3.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling call = new calling("Video call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

l3.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling\_client call = new calling\_client("Video call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

//Audio call - icon no.03

ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/phone.png"));

Image i11 = i10.getImage().getScaledInstance(28, 28, Image.SCALE\_DEFAULT);

ImageIcon i12 = new ImageIcon(i11);

JLabel l4 = new JLabel(i12);

l4.setBounds(270, 20, 30, 30);

p1.add(l4);

l4.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling call = new calling("Audio call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

l4.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling\_client call = new calling\_client("Audio call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

//Backup button - icon no.04

ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/Upload.png"));

Image i14 = i13.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i15 = new ImageIcon(i14);

JLabel l5 = new JLabel(i15);

l5.setBounds(315, 20, 30, 30);

p1.add(l5);

l5.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e){

try{

String data = tf1.getText();

Date date = new Date();

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yy");

String datestr = formatter.format(date);

String sql = "INSERT INTO chat\_backup (chats,date\_chats) " + "VALUES ('" + data + "','" + datestr + "');";

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

sts.executeUpdate(sql);

System.out.println("data entered" + data);

con.close();

}

catch(Exception ee){

ee.printStackTrace();

}

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e){}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e) {}

});

//Download button - icon no.04

ImageIcon i133 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/download.png"));

Image i144 = i133.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i155 = new ImageIcon(i144);

JLabel l55 = new JLabel(i155);

l55.setBounds(360, 20, 30, 30);

p1.add(l55);

l55.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

backup obj = new backup();

}

public void mousePressed(MouseEvent e){}

public void mouseReleased(MouseEvent e){}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e) {}

});

//Name - text no.01

JLabel l6 = new JLabel("Elon Musk");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 100, 40);

p1.add(l6);

//send text field

t1 = new JTextField();

t1.setBounds(5, 555, 300, 40);

t1.setFont(new Font("SAN\_SERIF", Font.PLAIN, 16));

add(t1);

//send button

b1 = new JButton("Send");

b1.setBounds(310, 555, 85, 40);

b1.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

b1.setForeground(Color.white);

b1.setBackground(new Color(r,g,b));

b1.addActionListener(this);

add(b1);

//text field

// display area.

tf1 = new JTextArea();

tf1.setFont(new Font("SAN\_SERIF", Font.BOLD, 16));

tf1.setBackground(Color.white);

tf1.setEditable(false);

tf1.setLineWrap(true);

tf1.setWrapStyleWord(true);

// adding scroll bar

sb = new JScrollPane(tf1);

sb.setBounds(5, 75, 390, 475);

sb.addNotify();

add(sb);

// getContentPane().setBackground(Color.yellow);//background color

setLayout(null);

//Window box - Server

setSize(400, 600);

setLocation(200, 70);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

// for overriding in action performed for exit

public void actionPerformed(ActionEvent ae) {

String out = t1.getText();

tf1.setText(tf1.getText() + "\n\n\t\t" + out);

//network

try {

dout.writeUTF(out);

} catch (IOException e) {

e.printStackTrace();

}

t1.setText("");

}

public static void main(String[] args) {

login djc = new login();

String textinput = "";

try{

skt = new ServerSocket(6000);

s= skt.accept();

din = new DataInputStream(s.getInputStream());

dout = new DataOutputStream(s.getOutputStream());

while(!textinput.equals("Exit")) {

textinput = din.readUTF();

tf1.setText(tf1.getText() + "\n\n " + textinput);

}

skt.close();

s.close();

}

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

}

}

**CLIENT CODE:**

package com.chatting\_application;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.net.Socket;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.SimpleDateFormat;

import java.util.Date;

//=== === === === === === BACKUP MESSAGES === === === === === === === === === === === === === === === === === === === === ===//

class backup\_client extends JFrame {

JPanel p1;

JTextField t1;

JButton b1;

static JTextArea tf1;

JScrollPane sb;

JLabel l5;

public backup\_client() {

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(7, 94, 84));

p1.setBounds(0, 0, 400, 70);

add(p1);

// Download - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/download.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

//Name - text no.01

JLabel l6 = new JLabel("Backed up messages");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 400, 40);

p1.add(l6);

//close button

b1 = new JButton("Close");

b1.setBounds(310, 555, 85, 40);

b1.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

b1.setForeground(Color.white);

b1.setBackground(new Color(7, 94, 84));

add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

//text field

// display area.

tf1 = new JTextArea();

tf1.setFont(new Font("SAN\_SERIF", Font.PLAIN, 14));

tf1.setBackground(Color.white);

tf1.setEditable(false);

tf1.setLineWrap(true);

tf1.setWrapStyleWord(true);

// adding scroll bar

sb = new JScrollPane(tf1);

sb.setBounds(5, 75, 390, 475);

sb.addNotify();

add(sb);

String BackedData = null;

String BackedDate = null;

String[] Backed\_up = new String[1000];

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

ResultSet resultSet = sts.executeQuery("select \* from chat\_backup");

while(resultSet.next()){

BackedData = resultSet.getString("chats");

BackedDate = resultSet.getString("date\_chats");

tf1.setText(tf1.getText()+"\n" + "Text: "+BackedData+" || Date: "+BackedDate);

}

System.out.println(BackedData);

System.out.println(BackedDate);

con.close();

} catch (Exception ae) {

ae.printStackTrace();

}

setLayout(null);

//Window box - Server

setSize(400, 600);

setLocation(250, 90);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

}

//=== === === === === === CALLING === === === === === === === === === === === === === === === === === === === === ===

class calling\_client extends JFrame {

calling\_client(String a){

JPanel p1;

JPanel p2;

JPanel p3;

JTextField t1;

JButton b1;

JTextArea tf1;

JScrollPane sb;

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(7, 94, 84));

p1.setBounds(0, 0, 400, 70);

add(p1);

// exit button - icon no.01

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3.png"));

Image i2 = i1.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(5, 25, 20, 20);

p1.add(l1);

//mouseevent for exit

l1.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent ae) {

System.exit(0);

}

});

// DP - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/astronaut.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

// Video call - icon no.02

ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/video.png"));

Image i8 = i7.getImage().getScaledInstance(30, 30, Image.SCALE\_DEFAULT);

ImageIcon i9 = new ImageIcon(i8);

JLabel l3 = new JLabel(i9);

l3.setBounds(230, 15, 40, 40);

p1.add(l3);

//Audio call - icon no.03

ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/phone.png"));

Image i11 = i10.getImage().getScaledInstance(33, 33, Image.SCALE\_DEFAULT);

ImageIcon i12 = new ImageIcon(i11);

JLabel l4 = new JLabel(i12);

l4.setBounds(290, 16, 40, 40);

p1.add(l4);

//More options - icon no.04

ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3icon.png"));

Image i14 = i13.getImage().getScaledInstance(12, 22, Image.SCALE\_DEFAULT);

ImageIcon i15 = new ImageIcon(i14);

JLabel l5 = new JLabel(i15);

l5.setBounds(335, 15, 40, 40);

p1.add(l5);

//Name - text no.01

JLabel l6 = new JLabel("Rayan");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 100, 40);

p1.add(l6);

p2 = new JPanel();

p2.setLayout(null);

p2.setBackground(new Color(7, 94, 84));

p2.setBounds(0, 510, 430, 100);

add(p2);

ImageIcon i44 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/deline button.png"));

Image i55 = i44.getImage().getScaledInstance(120, 120, Image.SCALE\_DEFAULT);

ImageIcon i66 = new ImageIcon(i55);

JLabel l22 = new JLabel(i66);

l22.setBounds(30, 20, 60, 60);

p2.add(l22);

l22.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

System.exit(0);

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

ImageIcon i45 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/speaker button.png"));

Image i56 = i45.getImage().getScaledInstance(70, 70, Image.SCALE\_DEFAULT);

ImageIcon i67 = new ImageIcon(i56);

JLabel l23 = new JLabel(i67);

l23.setBounds(130, 20, 60, 60);

p2.add(l23);

ImageIcon i46 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/mute button.png"));

Image i57 = i46.getImage().getScaledInstance(20, 30, Image.SCALE\_DEFAULT);

ImageIcon i68 = new ImageIcon(i57);

JLabel l24 = new JLabel(i68);

l24.setBounds(215, 20, 60, 60);

p2.add(l24);

ImageIcon i47 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/Video Call Mute.png "));

Image i58 = i47.getImage().getScaledInstance(35, 35, Image.SCALE\_DEFAULT);

ImageIcon i69 = new ImageIcon(i58);

JLabel l25 = new JLabel(i69);

l25.setBounds(300, 20, 60, 60);

p2.add(l25);

p3 = new JPanel();

p3.setLayout(null);

p3.setBounds(0, 100, 430, 400);

add(p3);

// DP - icon no.02

ImageIcon i444 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/elon.png"));

Image i555 = i444.getImage().getScaledInstance(250, 250, Image.SCALE\_DEFAULT);

ImageIcon i666 = new ImageIcon(i555);

JLabel l222 = new JLabel(i666);

l222.setBounds(50, 5, 300, 300);

p3.add(l222);

JLabel call = new JLabel("Connecting.....");

call.setBackground(Color.black);

call.setBounds(140,250,200,100);

call.setFont(new Font(Font.SANS\_SERIF,Font.BOLD,18));

p3.add(call);

JLabel call1 = new JLabel(a);

call1.setBackground(Color.black);

call1.setBounds(160,280,200,100);

call1.setFont(new Font(Font.SANS\_SERIF,Font.BOLD,16));

p3.add(call1);

setLayout(null);

setSize(400, 600);

setLocation(700, 70);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

}

class Error\_thrower\_client1 extends JFrame{

JLabel lb1;

JButton b1;

JLabel l2;

JLabel l3;

JLabel l4;

public Error\_thrower\_client1() {

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/1.1.png"));

Image i2 = i1.getImage().getScaledInstance(450,500,Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(0,0,0,0);

add(l1);

l2 = new JLabel("Incorrect");

l2.setBounds(2,2,150,150);

l2.setForeground(Color.red);

l2.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l2.setLocation(240,208);

l1.add(l2);

l3 = new JLabel("Username");

l3.setBounds(2,2,150,150);

l3.setForeground(Color.red);

l3.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l3.setLocation(240,256);

l1.add(l3);

l4 = new JLabel("Password");

l4.setBounds(2,2,150,150);

l4.setForeground(Color.red);

l4.setFont(new Font("SAN\_SERIF",Font.BOLD,12));

l4.setLocation(240,305);

l1.add(l4);

lb1 = new JLabel("ERROR");

lb1.setBounds(2,2,300,300);

lb1.setForeground(Color.red);

lb1.setFont(new Font("SAN\_SERIF",Font.BOLD,18));

lb1.setLocation(227,105);

l1.add(lb1);

b1 = new JButton("Close");

b1.setBounds(213,410, 100, 30);

b1.setForeground(Color.white);

b1.setBackground(new Color(7, 94, 84));

l1.add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

setSize(450,550);

setLocation(450,120);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setUndecorated(true);

setVisible(true);

}

}

//===LOGIN FORM === === === === === === === === === === === === === === === === === === === === === === === === ===//

class loginClient extends JFrame {

JPanel p1;

JLabel l1;

JLabel l2;

JTextField t1;

JLabel l3;

JPasswordField t2;

JButton b1;

JButton b2;

JLabel l4;

JLabel l5;

JLabel l6;

JTextField r;

JTextField g;

JTextField b;

loginClient() {

p1 = new JPanel();

p1.setBounds(0, 0, 1000, 500);

p1.setBackground(new Color(7, 94, 84));

p1.setLayout(null);

add(p1);

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/loginback.jpg"));

Image i2 = i1.getImage().getScaledInstance(1000, 500, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel bg = new JLabel(i3);

bg.setBounds(0, 0, 1000, 500);

p1.add(bg);

l1 = new JLabel("Login Form");

l1.setBounds(645, -190, 1000, 500);

l1.setFont(new Font("SANS\_SERIF", Font.BOLD, 28));

l1.setForeground(Color.white);

bg.add(l1);

l2 = new JLabel("Email:");

l2.setBounds(600, -130, 1000, 500);

l2.setFont(new Font("SANS\_SERIF", Font.BOLD, 18));

l2.setForeground(Color.white);

bg.add(l2);

t1 = new JTextField("Email");

t1.setBounds(600, 150, 250, 30);

t1.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

t1.setForeground(Color.black);

bg.add(t1);

l3 = new JLabel("Password:");

l3.setBounds(600, -25, 1000, 500);

l3.setFont(new Font("SANS\_SERIF", Font.BOLD, 18));

l3.setForeground(Color.white);

bg.add(l3);

t2 = new JPasswordField("Password");

t2.setBounds(600, 255, 250, 30);

t2.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

t2.setForeground(Color.black);

bg.add(t2);

b1 = new JButton("LogIn");

b1.setBounds(615, 320, 100, 30);

b1.setForeground(Color.black);

b1.setBackground(Color.white);

bg.add(b1);

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

ResultSet resultSet = sts.executeQuery("select \* from users where name = '" + t1.getText() + "' and password = '" + t2.getText().toString() + "'");

int x = Integer.parseInt(r.getText());

int y = Integer.parseInt(g.getText());

int z = Integer.parseInt(b.getText());

if (resultSet.next()) {

new Client1(x,y,z).setVisible(true);

setVisible(false);

} else {

Error\_thrower\_client1 etc = new Error\_thrower\_client1();

}

con.close();

} catch (Exception ae) {

ae.printStackTrace();

}

}

});

b1.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

}

@Override

public void mousePressed(MouseEvent e) {

}

public void mouseReleased(MouseEvent e) {

}

public void mouseEntered(MouseEvent ae) {

b1.setBackground(new Color(58, 111, 89));

}

@Override

public void mouseExited(MouseEvent ae) {

b1.setBackground(Color.white);

}

});

b2 = new JButton("Cancel");

b2.setBounds(735, 320, 100, 30);

b2.setForeground(Color.black);

b2.setBackground(Color.white);

bg.add(b2);

b2.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

System.exit(0);

}

public void mousePressed(MouseEvent e) {

}

public void mouseReleased(MouseEvent e) {

}

public void mouseEntered(MouseEvent e) {

b2.setBackground(new Color(58, 111, 89));

}

public void mouseExited(MouseEvent e) {

b2.setBackground(Color.white);

}

});

l4 = new JLabel("N🌖VA");

l4.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 70));

l4.setBounds(200, -80, 500, 500);

l4.setForeground(Color.white);

bg.add(l4);

l5 = new JLabel("Chatting");

l5.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 30));

l5.setBounds(200, -0, 500, 500);

l5.setForeground(Color.white);

bg.add(l5);

l6 = new JLabel("App");

l6.setFont(new Font(Font.SANS\_SERIF, Font.ITALIC, 30));

l6.setBounds(200, 25, 500, 500);

l6.setForeground(Color.white);

bg.add(l6);

r = new JTextField("7");

r.setBounds(615, 390, 40, 30);

r.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

r.setForeground(Color.black);

bg.add(r);

g = new JTextField("94");

g.setBounds(705, 390, 40, 30);

g.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

g.setForeground(Color.black);

bg.add(g);

b = new JTextField("84");

b.setBounds(795, 390, 40, 30);

b.setFont(new Font("SANS\_SERIF", Font.ITALIC, 15));

b.setForeground(Color.black);

bg.add(b);

setSize(1000, 500);

setLocation(200, 130);

setLayout(null);

setUndecorated(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setVisible(true);

}

}

//--------------------------------------------------------------------------------------------------------------------------//

public class Client1 extends JFrame implements ActionListener {

JPanel p1;

JTextField t1;

JButton b1;

static JTextArea tf1;

JScrollPane sb;

//network

static Socket s1;

static DataInputStream din1;

static DataOutputStream dout1;

public Client1(int r, int g, int b) {

//green navbar

p1 = new JPanel();

p1.setLayout(null);

p1.setBackground(new Color(r,g,b));

p1.setBounds(0, 0, 400, 70);

add(p1);

// exit button - icon no.01

ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/3.png"));

Image i2 = i1.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i3 = new ImageIcon(i2);

JLabel l1 = new JLabel(i3);

l1.setBounds(5, 25, 20, 20);

p1.add(l1);

//mouseevent for exit

l1.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent ae) {

System.exit(0);

}

});

// DP - icon no.02

ImageIcon i4 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/astronaut.png"));

Image i5 = i4.getImage().getScaledInstance(50, 50, Image.SCALE\_DEFAULT);

ImageIcon i6 = new ImageIcon(i5);

JLabel l2 = new JLabel(i6);

//9042622676 --> samosa waley

l2.setBounds(30, 5, 60, 60);

p1.add(l2);

// Video call - icon no.02

ImageIcon i7 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/video.png"));

Image i8 = i7.getImage().getScaledInstance(30, 30, Image.SCALE\_DEFAULT);

ImageIcon i9 = new ImageIcon(i8);

JLabel l3 = new JLabel(i9);

l3.setBounds(220, 20, 30, 30);

p1.add(l3);

l3.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling call = new calling("Video call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

l3.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling\_client call = new calling\_client("Video call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

//Audio call - icon no.03

ImageIcon i10 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/phone.png"));

Image i11 = i10.getImage().getScaledInstance(33, 33, Image.SCALE\_DEFAULT);

ImageIcon i12 = new ImageIcon(i11);

JLabel l4 = new JLabel(i12);

l4.setBounds(270, 20, 30, 30);

p1.add(l4);

l4.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling\_client call = new calling\_client("Audio call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

l4.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e) {calling call = new calling("Audio call");}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public void mouseEntered(MouseEvent e) {}

public void mouseExited(MouseEvent e) {}

});

//Backup button - icon no.04

ImageIcon i13 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/Upload.png"));

Image i14 = i13.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i15 = new ImageIcon(i14);

JLabel l5 = new JLabel(i15);

l5.setBounds(315, 20, 30, 30);

p1.add(l5);

l5.addMouseListener(new MouseListener() {

public void mouseClicked(MouseEvent e){

try{

String data = tf1.getText();

Date date = new Date();

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yy");

String datestr = formatter.format(date);

String sql = "INSERT INTO chat\_backup (chats,date\_chats) " + "VALUES ('" + data + "','" + datestr + "');";

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/abc", "root", "1234");

Statement sts = con.createStatement();

sts.executeUpdate(sql);

System.out.println("data entered" + data);

con.close();

}

catch(Exception ee){

ee.printStackTrace();

}

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e){}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e) {}

});

//Download button - icon no.04

ImageIcon i133 = new ImageIcon(ClassLoader.getSystemResource("com/chatting\_application/icons/download.png"));

Image i144 = i133.getImage().getScaledInstance(25, 25, Image.SCALE\_DEFAULT);

ImageIcon i155 = new ImageIcon(i144);

JLabel l55 = new JLabel(i155);

l55.setBounds(360, 20, 30, 30);

p1.add(l55);

l55.addMouseListener(new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

backup obj = new backup();

}

public void mousePressed(MouseEvent e){}

public void mouseReleased(MouseEvent e){}

public void mouseEntered(MouseEvent e){}

public void mouseExited(MouseEvent e) {}

});

//Name - text no.01

JLabel l6 = new JLabel("Rayan");

l6.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

l6.setForeground(Color.WHITE);

l6.setBounds(95, 16, 100, 40);

p1.add(l6);

//send text field

t1 = new JTextField();

t1.setBounds(5, 555, 300, 40);

t1.setFont(new Font("SAN\_SERIF", Font.PLAIN, 16));

add(t1);

//send button

b1 = new JButton("Send");

b1.setBounds(310, 555, 85, 40);

b1.setFont(new Font("SAN\_SERIF", Font.BOLD, 18));

b1.setForeground(Color.white);

b1.setBackground(new Color(r,g,b));

b1.addActionListener(this);

add(b1);

//text field

// display area.

tf1 = new JTextArea();

tf1.setFont(new Font("SAN\_SERIF", Font.BOLD, 16));

tf1.setBackground(Color.white);

tf1.setEditable(false);

tf1.setLineWrap(true);

tf1.setWrapStyleWord(true);

// adding scroll bar

sb = new JScrollPane(tf1);

sb.setBounds(5, 75, 390, 475);

sb.addNotify();

add(sb);

// getContentPane().setBackground(Color.yellow);//background color

setLayout(null);

//Window box - Server

setSize(400, 600);

setLocation(700, 70);

//for removing the above navbar box

setUndecorated(true);

setVisible(true);

}

// for overriding in action performed for exit

public void actionPerformed(ActionEvent ae) {

String out = t1.getText();

tf1.setText(tf1.getText() + "\n\n\t\t" + out);

try {

dout1.writeUTF(out);

} catch (IOException e) {

e.printStackTrace();

}

t1.setText("");

}

public static void main(String[] args) {

loginClient lg = new loginClient();

String textinput = "";

try{

s1= new Socket("127.0.0.1",6000);

din1 = new DataInputStream(s1.getInputStream());

dout1 = new DataOutputStream(s1.getOutputStream());

while(!textinput.equals("Exit")) {

textinput = din1.readUTF();

tf1.setText(tf1.getText() + "\n\n " + textinput);

}

s1.close();

}

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

}

}