WhatsApp - Chatting Application

Programming in Java (SWE1007)

Submitted By :

Srividya. S - 20MIS0083

Debdatta Ray - 20MIS0112

Abstract:

This Project Entitled 'WhatsApp - Chatting Application' is used basically for chatting purposes with remote clients or users on the Internet or local networks. Here in this project, a java client/server combination is used to chat with remote users. Server and client classes are created and frames are developed using java swing. Using socket programming the server and client are connected which enables ono-to-one chatting.

The project also includes the feature of group chatting. This feature enables multiple users to chat among themselves. The frames for the users is designed using swing. The messages are passed among different users using socket programming.

This project enables a user to send a message to multiple people at the same time saving time. Applets are machine-independent and so java programs can run on any computer on the internet.

The term client/server is used in the context of networking, what it actually means.

The concepts used are **swing** and **java networking** (socket programming)

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.

- AWT API was introduced in JDK 1.0. Most of the AWT components have become obsolete and should be replaced by newer Swing components.

- 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

Implementation:

Creating users- Multiple users are created having the same port address who can have a group chat.

Designing the frame- Using swing the frames for each user are created. It includes displaying the name of the sender along with the messages in the text area and changing the status from 'Active now' to 'typing…' using thread sleep() method.

Server- It connects the users using socket programming and keeps a tab on the user getting connected, sending and receiving the messages

Code:

One-to-one chatting

Server Code:

```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.border.EmptyBorder;


import java.net.*;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.io.*;


public class Server extends JFrame implements ActionListener {

        JPanel p1;

        JTextField t1;

        JButton b1;

        static JPanel a1;
```

```
    static JFrame f1 = new JFrame();

    static Box vertical = Box.createVerticalBox();
    static ServerSocket skt;
    static Socket s;
    static DataInputStream din;
    static DataOutputStream dout;

    Boolean typing;

        Server(){

        f1.setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOS
E);
                p1= new JPanel();
                p1.setLayout(null);
                p1.setBackground(new Color(7,94,84));
                p1.setBounds(0, 0, 416, 70);
      f1.add(p1);

                ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("chatting/application/icons/3.png"));
            Image i2 = i1.getImage().getScaledInstance(30, 30,
Image.SCALE_DEFAULT);
```

```
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l1 = new JLabel(i3);
        l1.setBounds(5, 17, 30, 30);
        p1.add(l1);

        l1.addMouseListener(new MouseAdapter(){
           public void mouseClicked(MouseEvent ae){
              System.exit(0);
           }
        });

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("chatting/application/icons/dp.png"))
;
        Image i5 = i4.getImage().getScaledInstance(60, 60,
Image.SCALE_DEFAULT);
        ImageIcon i6 = new ImageIcon(i5);
        JLabel l2 = new JLabel(i6);
        l2.setBounds(40, 5, 60, 60);
        p1.add(l2);

        ImageIcon i7 = new
ImageIcon(ClassLoader.getSystemResource("chatting/application/icons/video.png
"));
        Image i8 = i7.getImage().getScaledInstance(30, 30,
Image.SCALE_DEFAULT);
```

```
        ImageIcon i9 = new ImageIcon(i8);
        JLabel l5 = new JLabel(i9);
        l5.setBounds(290, 20, 30, 30);
        p1.add(l5);

        ImageIcon i11 = new
ImageIcon(ClassLoader.getSystemResource("chatting/application/icons/phone.pn
g"));
        Image i12 = i11.getImage().getScaledInstance(35, 30,
Image.SCALE_DEFAULT);
        ImageIcon i13 = new ImageIcon(i12);
        JLabel l6 = new JLabel(i13);
        l6.setBounds(343, 20, 35, 30);
        p1.add(l6);

        ImageIcon i14 = new
ImageIcon(ClassLoader.getSystemResource("chatting/application/icons/3icon.png
"));
        Image i15 = i14.getImage().getScaledInstance(13, 25,
Image.SCALE_DEFAULT);
        ImageIcon i16 = new ImageIcon(i15);
        JLabel l7 = new JLabel(i16);
        l7.setBounds(395, 20, 13, 25);
        p1.add(l7);
```

```
        JLabel l3=new JLabel("Debdatta");
        l3.setFont(new Font("SAN_SERIF",Font.BOLD,16));
        l3.setForeground(Color.WHITE);
        l3.setBounds(110, 15, 100, 18);
        p1.add(l3);

        JLabel l4=new JLabel("Active Now");
        l4.setFont(new Font("SAN_SERIF",Font.PLAIN,14));
        l4.setForeground(Color.WHITE);
        l4.setBounds(110, 38, 100, 22);
        p1.add(l4);

    Timer t = new Timer(1, new ActionListener(){
        public void actionPerformed(ActionEvent ae){
            if(!typing){
                l4.setText("Active Now");
            }
        }
      });

    t.setInitialDelay(2000);

    a1 = new JPanel();
    a1.setBounds(5, 75, 440, 570);
```

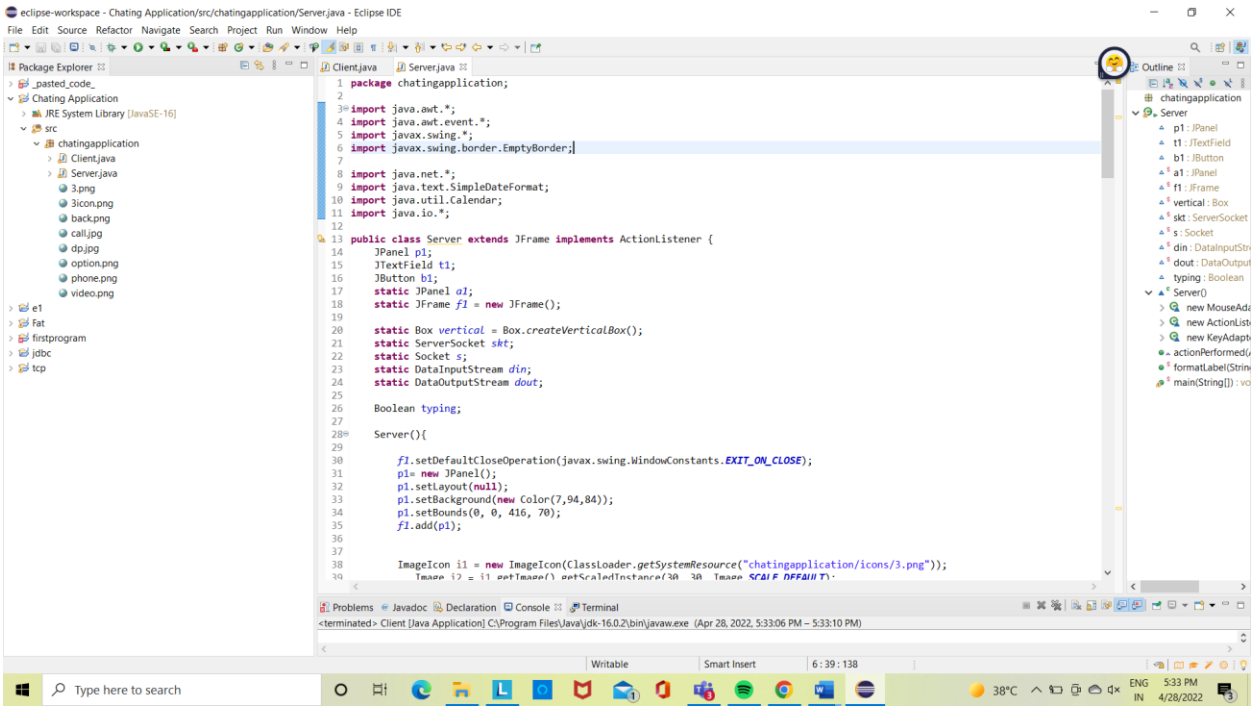
```
a1.setFont(new Font("SAN_SERIF", Font.PLAIN, 16));
f1.add(a1);

    t1=new JTextField();
    t1.setBounds(5, 560, 340, 30);
    t1.setFont(new Font("SAN_SERIF",Font.PLAIN,16));
    f1.add(t1);

    t1.addKeyListener(new  KeyAdapter(){
    public void keyPressed(KeyEvent  ke){
       l4.setText("typing...");

       t.stop();

       typing = true;
    }
    public void keyReleased(KeyEvent  ke){
       typing = false;

       if(!t.isRunning()){
          t.start();
       }
    }
  });
```

```
        b1=new JButton("Send");
        b1.setBounds(350, 557, 65, 35);
        b1.setBackground(new Color(7,94,84));
        b1.setForeground(Color.WHITE);
b1.setFont(new Font("SAN_SERIF", Font.PLAIN, 12));
b1.addActionListener(this);
        f1.add(b1);

        f1.getContentPane().setBackground(Color.WHITE);
        f1.setLayout(null);
        f1.setSize(419,600);
        f1.setLocation(400,200);
        f1.setUndecorated(true);
        f1.setVisible(true);
  }
  public void actionPerformed(ActionEvent ae) {
        try {
              String out = t1.getText();
              JPanel p2 = formatLabel(out);
   a1.setLayout(new BorderLayout());

   JPanel right = new JPanel(new BorderLayout());
   right.add(p2, BorderLayout.LINE_END);
```

```
            vertical.add(right);
            vertical.add(Box.createVerticalStrut(15));

            a1.add(vertical, BorderLayout.PAGE_START);
                        dout.writeUTF(out);
                        t1.setText("");
                    }catch(Exception e) {System.out.println(e);}

            }

            public static JPanel formatLabel(String out){
        JPanel p3 = new JPanel();
        p3.setLayout(new BoxLayout(p3, BoxLayout.Y_AXIS));

        JLabel l1 = new JLabel("<html><p  style = \"width :
150px\">"+out+"</p></html>");
        l1.setFont(new Font("Tahoma", Font.PLAIN, 16));
        l1.setBackground(new Color(37, 211, 102));
        l1.setOpaque(true);
        l1.setBorder(new EmptyBorder(15,15,15,50));

        Calendar cal = Calendar.getInstance();
        SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");

        JLabel l2 = new JLabel();
```

```
    l2.setText(sdf.format(cal.getTime()));

    p3.add(l1);
    p3.add(l2);
    return p3;
}

      public static void main(String[] args) {
              new Server().f1.setVisible(true);
              String msginput = "";

              try{
      skt = new ServerSocket(1248);
      while(true){
         s = skt.accept();
         din = new DataInputStream(s.getInputStream());
         dout = new DataOutputStream(s.getOutputStream());

            while(true){
                msginput = din.readUTF();
               JPanel p2 = formatLabel(msginput);

               JPanel left = new JPanel(new BorderLayout());
               left.add(p2, BorderLayout.LINE_START);
```

```
                vertical.add(left);
                f1.validate();
               }

        }
      }catch(Exception e){System.out.println(e);}
        }

}
```



Client code:

```
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
```

```
import java.net.*;
import java.io.*;
import java.util.Calendar;
import java.text.SimpleDateFormat;import javax.swing.*;

public class Client extends JFrame implements ActionListener {
        JPanel p1;
        JTextField t1;
        JButton b1;
        static JPanel a1;
         static JFrame f1 = new JFrame();
        static Box vertical = Box.createVerticalBox();

   static Socket s;
   static DataInputStream din;
   static DataOutputStream dout;

   Boolean typing;

        Client(){

        f1.setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOS
E);
```

```
        p1= new JPanel();
        p1.setLayout(null);
        p1.setBackground(new Color(7,94,84));
        p1.setBounds(0, 0, 416, 70);
        add(p1);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("chatingapplication/3.png"));
        Image i2 = i1.getImage().getScaledInstance(30, 30,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel l1 = new JLabel(i3);
        l1.setBounds(5, 17, 30, 30);
        p1.add(l1);

        l1.addMouseListener(new MouseAdapter(){
          public void mouseClicked(MouseEvent ae){
            System.exit(0);
          }
        });

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("chatingapplication/dp.jpg"));
        Image i5 = i4.getImage().getScaledInstance(60, 60,
Image.SCALE_DEFAULT);
        ImageIcon i6 = new ImageIcon(i5);
```

```
        JLabel l2 = new JLabel(i6);
        l2.setBounds(40, 5, 60, 60);
        p1.add(l2);

        ImageIcon i7 = new
ImageIcon(ClassLoader.getSystemResource("chatingapplication/video.png"));
        Image i8 = i7.getImage().getScaledInstance(30, 30,
Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel l5 = new JLabel(i9);
        l5.setBounds(290, 20, 30, 30);
        p1.add(l5);

        ImageIcon i11 = new
ImageIcon(ClassLoader.getSystemResource("chatingapplication/phone.png"));
        Image i12 = i11.getImage().getScaledInstance(35, 30,
Image.SCALE_DEFAULT);
        ImageIcon i13 = new ImageIcon(i12);
        JLabel l6 = new JLabel(i13);
        l6.setBounds(343, 20, 35, 30);
        p1.add(l6);

        ImageIcon i14 = new
ImageIcon(ClassLoader.getSystemResource("chatingapplication/3icon.png"));
        Image i15 = i14.getImage().getScaledInstance(13, 25,
Image.SCALE_DEFAULT);
```

```
ImageIcon i16 = new ImageIcon(i15);
JLabel l7 = new JLabel(i16);
l7.setBounds(395, 20, 13, 25);
p1.add(l7);



   JLabel l3=new JLabel("Srividya");
   l3.setFont(new Font("SAN_SERIF",Font.BOLD,16));
   l3.setForeground(Color.WHITE);
   l3.setBounds(110, 15, 100, 18);
   p1.add(l3);

   JLabel l4=new JLabel("Active Now");
   l4.setFont(new Font("SAN_SERIF",Font.PLAIN,14));
   l4.setForeground(Color.WHITE);
   l4.setBounds(110, 38, 100, 22);
   p1.add(l4);
   Timer t = new Timer(1, new ActionListener(){
   public void actionPerformed(ActionEvent ae){
      if(!typing){
         l4.setText("Active Now");
      }
   }
});
```

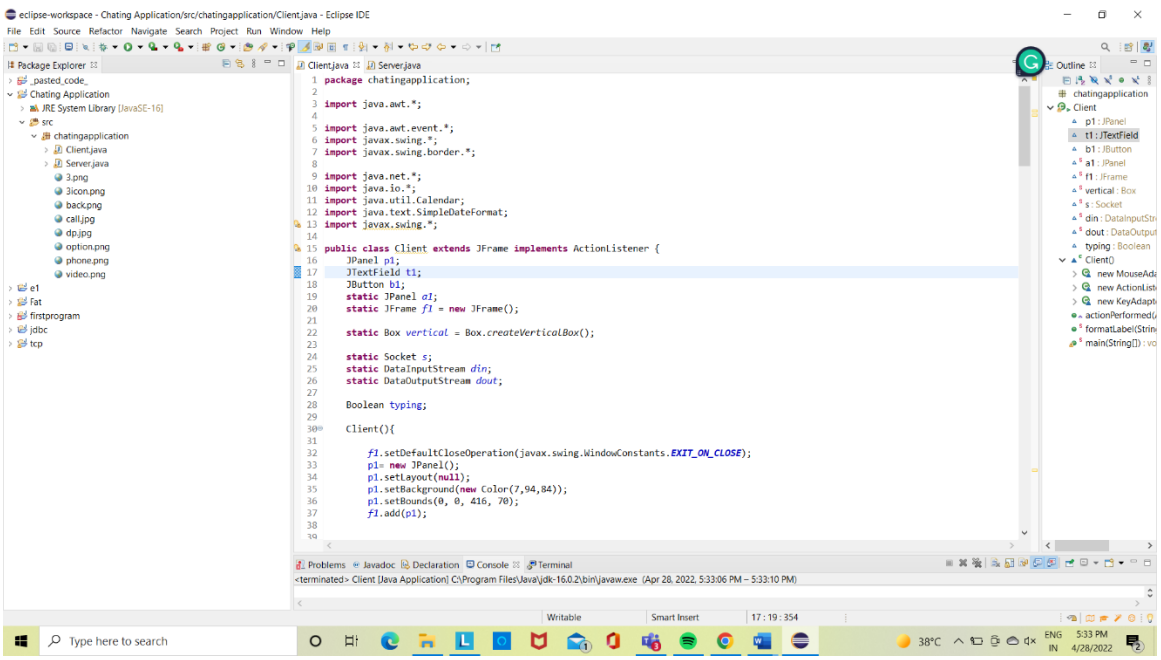
```
t.setInitialDelay(2000);

a1 = new JPanel();
a1.setBounds(5, 75, 440, 570);
a1.setFont(new Font("SAN_SERIF", Font.PLAIN, 16));
f1.add(a1);


   t1=new JTextField();
   t1.setBounds(5, 560, 340, 30);
   t1.setFont(new Font("SAN_SERIF",Font.PLAIN,16));
   add(t1);
   t1.addKeyListener(new  KeyAdapter(){
   public void keyPressed(KeyEvent  ke){
      l4.setText("typing...");

      t.stop();

      typing = true;
   }

   public void keyReleased(KeyEvent  ke){
      typing = false;
```

```
                if(!t.isRunning()){
                    t.start();
                }
            }
        });

            b1=new JButton("Send");
            b1.setBounds(350, 557, 65, 35);
            b1.setBackground(new Color(7,94,84));
            b1.setForeground(Color.WHITE);
b1.setFont(new Font("SAN_SERIF", Font.PLAIN, 12));
b1.addActionListener(this);
            add(b1);

            getContentPane().setBackground(Color.WHITE);

            setLayout(null);
            setSize(419,600);
            setLocation(1000,200);
            setUndecorated(true);
            setVisible(true);
    }
    public void actionPerformed(ActionEvent  ae) {
```

```
            try {
                    String out = t1.getText();
                    JPanel p2 = formatLabel(out);
                     a1.setLayout(new BorderLayout());
                     JPanel right = new JPanel(new BorderLayout());
                     right.add(p2, BorderLayout.LINE_END);
               vertical.add(right);
               vertical.add(Box.createVerticalStrut(15));

               a1.add(vertical, BorderLayout.PAGE_START);
            dout.writeUTF(out);
            t1.setText("");
              }catch(Exception e) {System.out.println(e);}

        }
        public static JPanel formatLabel(String out){
      JPanel p3 = new JPanel();
      p3.setLayout(new BoxLayout(p3, BoxLayout.Y_AXIS));

      JLabel l1 = new JLabel("<html><p  style = \"width :
150px\">"+out+"</p></html>");
      l1.setFont(new Font("Tahoma", Font.PLAIN, 16));
      l1.setBackground(new Color(37, 211, 102));
      l1.setOpaque(true);
      l1.setBorder(new EmptyBorder(15,15,15,50));
```

```
    Calendar cal = Calendar.getInstance();
    SimpleDateFormat sdf = new SimpleDateFormat("HH:mm");

    JLabel l2 = new JLabel();
    l2.setText(sdf.format(cal.getTime()));

    p3.add(l1);
    p3.add(l2);
    return p3;
}
      public static void main(String[] args) {
              new Client().setVisible(true);


    try{

       s = new Socket("127.0.0.1", 1248);

       din  = new DataInputStream(s.getInputStream());
       dout = new DataOutputStream(s.getOutputStream());
       String msginput = "";
       while(true){
          a1.setLayout(new BorderLayout());
```

```
            msginput = din.readUTF();
                JPanel p2 = formatLabel(msginput);
            JPanel left = new JPanel(new BorderLayout());
            left.add(p2, BorderLayout.LINE_START);

            vertical.add(left);
            vertical.add(Box.createVerticalStrut(15));
            a1.add(vertical, BorderLayout.PAGE_START);
            f1.validate();
        }

    }catch(Exception e){}
      }

}
```

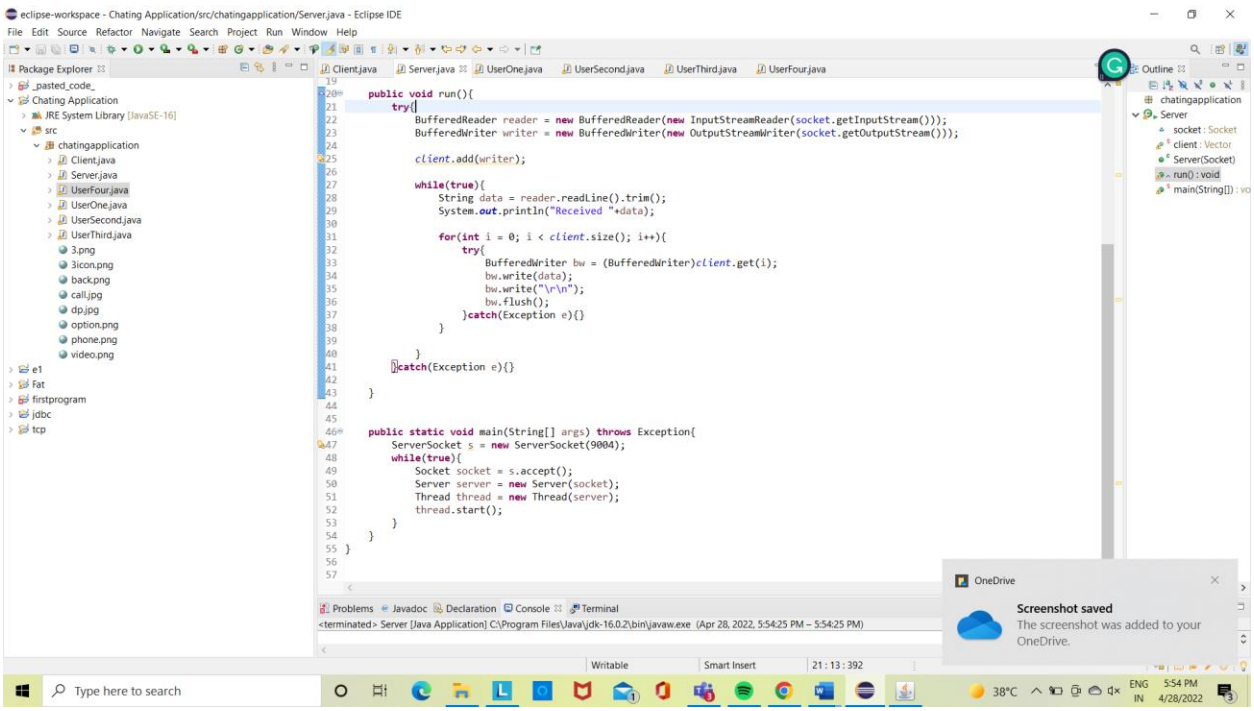


Group chatting

Server Code:

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

public class Server implements Runnable{

   Socket socket;

   public static Vector client = new Vector();

   public Server(Socket socket){
      try{
         this.socket = socket;
      }catch(Exception e){}
   }


   public void run(){
      try{
         BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
         BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()));

         client.add(writer);

         while(true){
            String data = reader.readLine().trim();
            System.out.println("Received "+data);

            for(int i = 0; i < client.size(); i++){
               try{
                  BufferedWriter bw = (BufferedWriter)client.get(i);
                  bw.write(data);
```

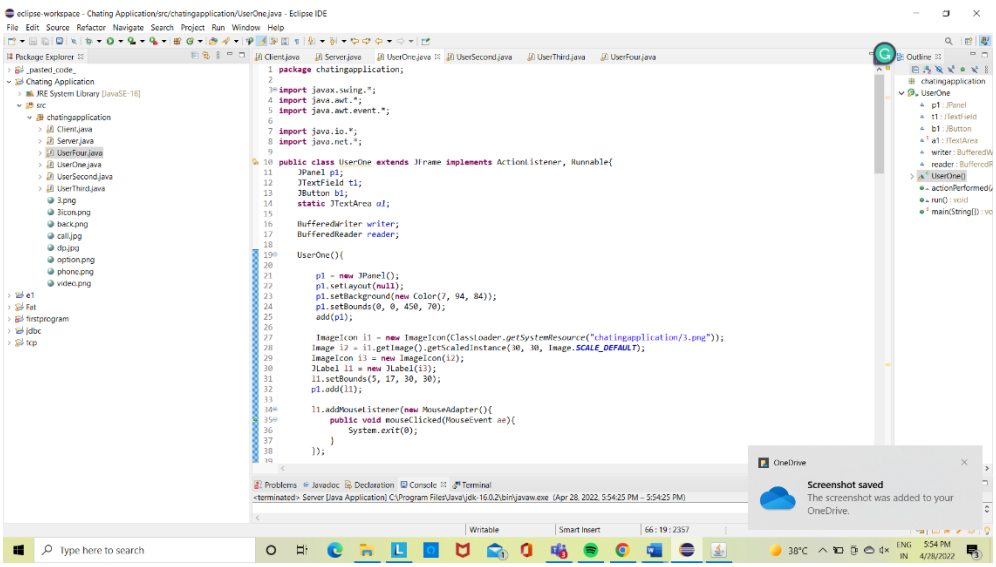
```
                bw.write("\r\n");
                bw.flush();
            }catch(Exception e){}
        }

      }
    }catch(Exception e){}

  }


  public static void main(String[] args) throws Exception{
    ServerSocket s = new ServerSocket(9004);
    while(true){
      Socket socket = s.accept();
      Server server = new Server(socket);
      Thread thread = new Thread(server);
      thread.start();
    }
  }
}
```

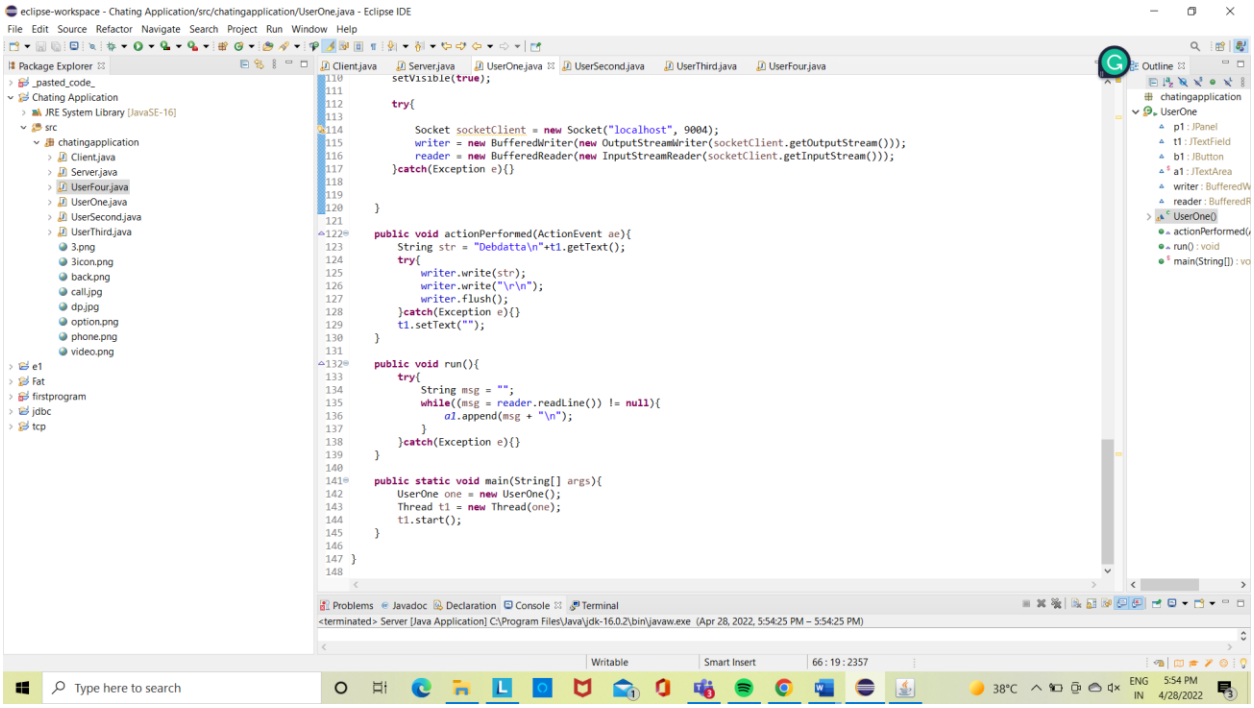


UserOne:

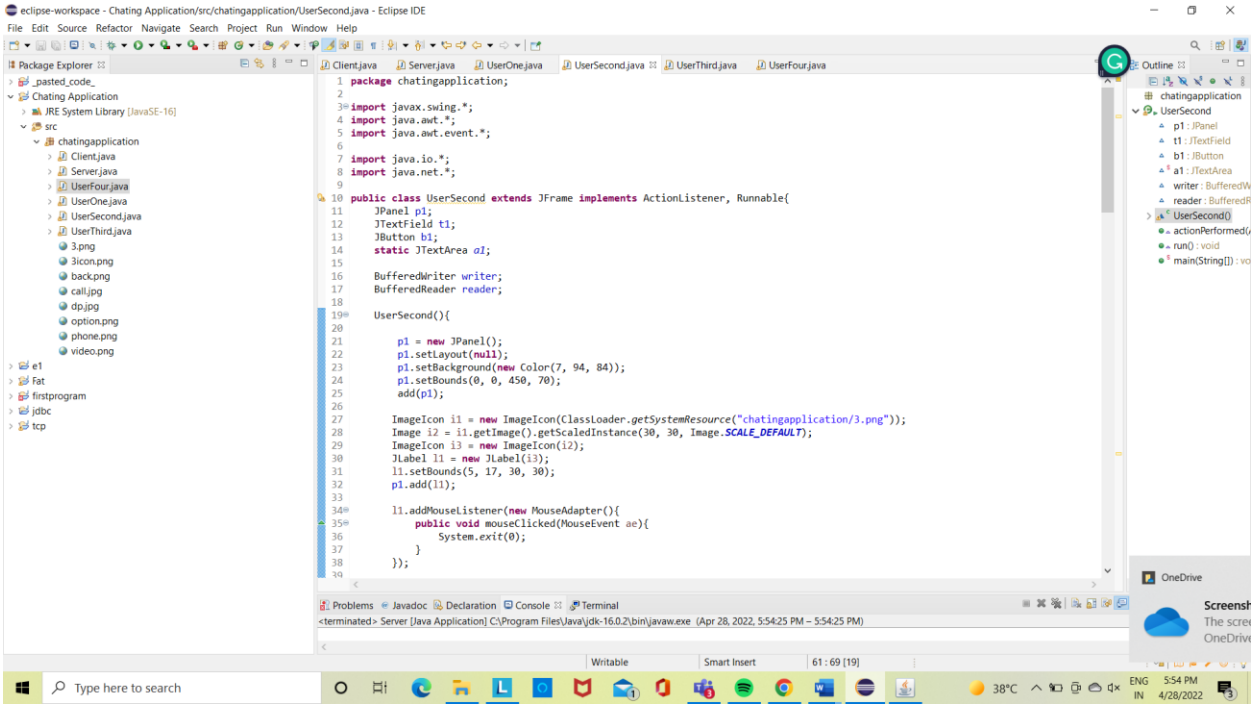
```
try{

        Socket socketClient  = new Socket("localhost",  4004);
        writer = new BufferedWriter(new
OutputStreamWriter(socketClient.getOutputStream()));
        reader = new BufferedReader(new
InputStreamReader(socketClient.getInputStream()));
     }catch(Exception e){}


   }

   public void actionPerformed(ActionEvent  ae){
      String str = "Debdatta\n"+t1.getText();
      try{
         writer.write(str);
         writer.write("\r\n");
```

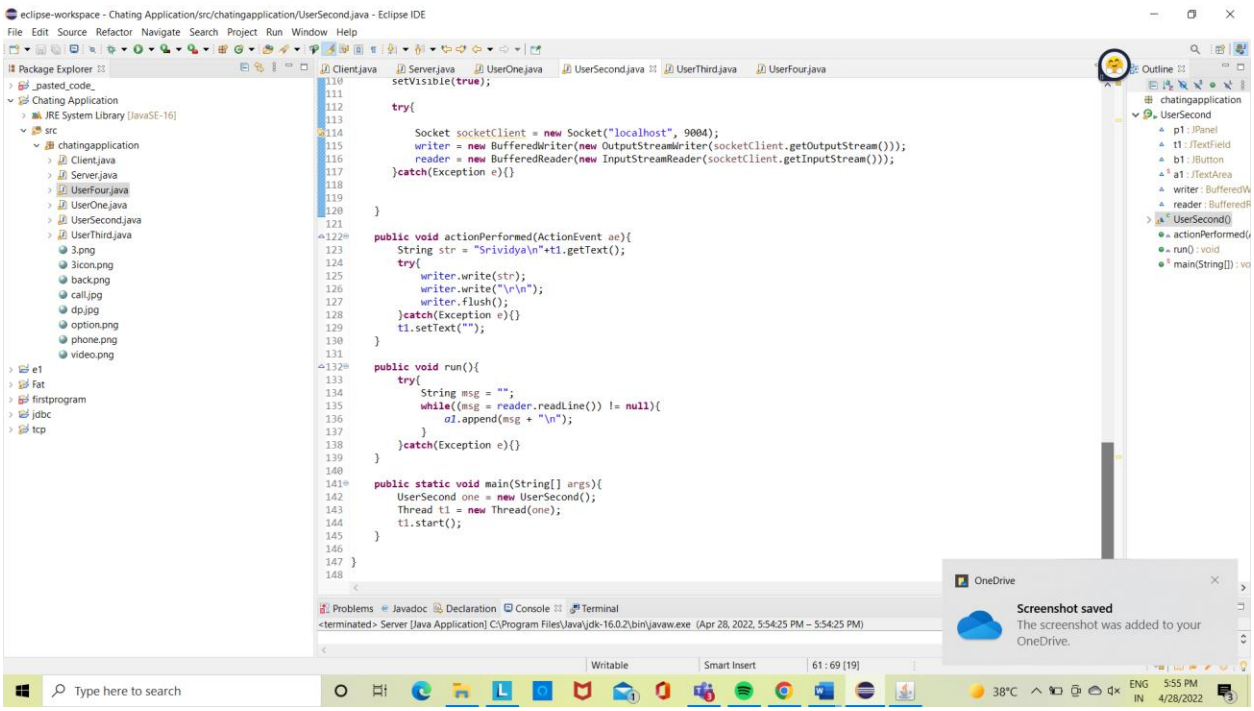
```
            writer.flush();
        }catch(Exception e){}
        t1.setText("");
    }

    public void run(){
        try{
            String msg = "";
            while((msg = reader.readLine()) != null){
                a1.append(msg + "\n");
            }
        }catch(Exception e){}
    }

    public static void main(String[] args){
        UserOne one = new UserOne();
        Thread t1 = new Thread(one);
        t1.start();
    }

}
```



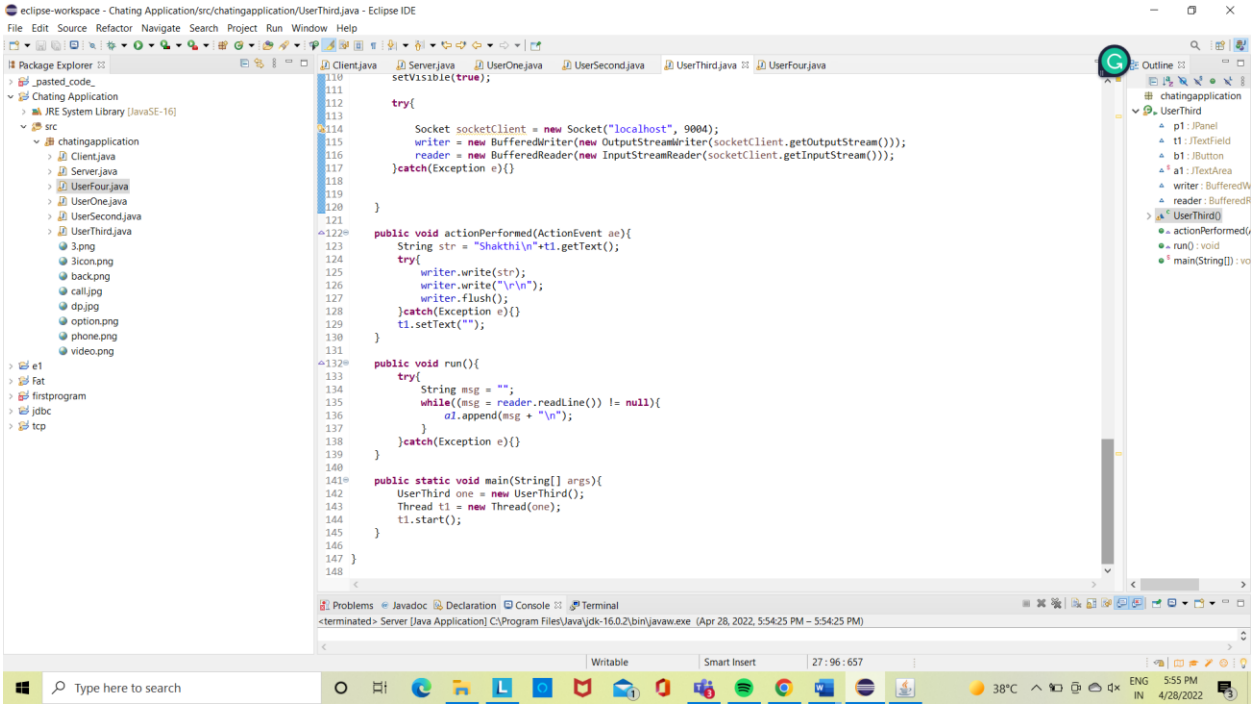

eclipse-workspace - Chating Application/src/chatingapplication/UserOne.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
setVisible(true);
try{
Socket socketClient = new Socket("localhost", 9004);
writer = new BufferedWriter(new OutputStreamWriter(socketClient.getOutputStream()));
reader = new BufferedReader(new InputStreamReader(socketClient.getInputStream()));
}catch(Exception e){}
}
public void actionPerformed(ActionEvent ae){
String str = "Debdatta\n"+t1.getText();
try{
writer.write(str);
writer.write("\r\n");
writer.flush();
}catch(Exception e){}
t1.setText("");
}
public void run(){
try{
String msg = "";
while((msg = reader.readLine()) != null){
a1.append(msg + "\n");
}
}catch(Exception e){}
}
public static void main(String[] args){
UserOne one = new UserOne();
Thread t1 = new Thread(one);
t1.start();
}
}
Outline
chatingapplication
UserOne
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserOne()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
Writable
Smart Insert
66 : 19 : 2357
Type here to search
38°C
ENG
IN
5:54 PM
4/28/2022

UserSecond

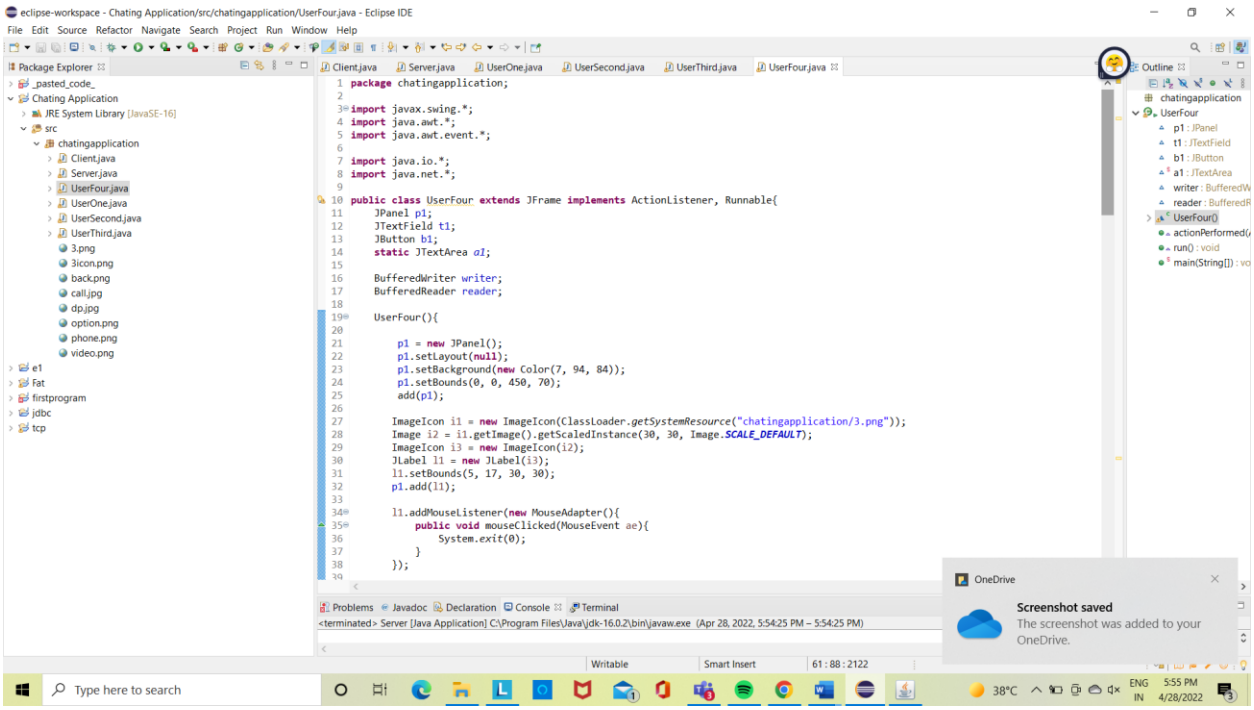

eclipse-workspace - Chating Application/src/chatingapplication/UserSecond.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
package chatingapplication;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
public class UserSecond extends JFrame implements ActionListener, Runnable{
JPanel p1;
JTextField t1;
JButton b1;
static JTextArea a1;
BufferedWriter writer;
BufferedReader reader;
UserSecond(){
p1 = new JPanel();
p1.setLayout(null);
p1.setBackground(new Color(7, 94, 84));
p1.setBounds(0, 0, 450, 70);
add(p1);
ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("chatingapplication/3.png"));
Image i2 = i1.getImage().getScaledInstance(30, 30, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel l1 = new JLabel(i3);
l1.setBounds(5, 17, 30, 30);
p1.add(l1);
l1.addMouseListener(new MouseAdapter(){
public void mouseClicked(MouseEvent ae){
System.exit(0);
}
});
Outline
chatingapplication
UserSecond
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserSecond()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
OneDrive
Writable
Smart Insert
61 : 69 [19]
Type here to search
38°C
ENG
IN
5:54 PM
4/28/2022

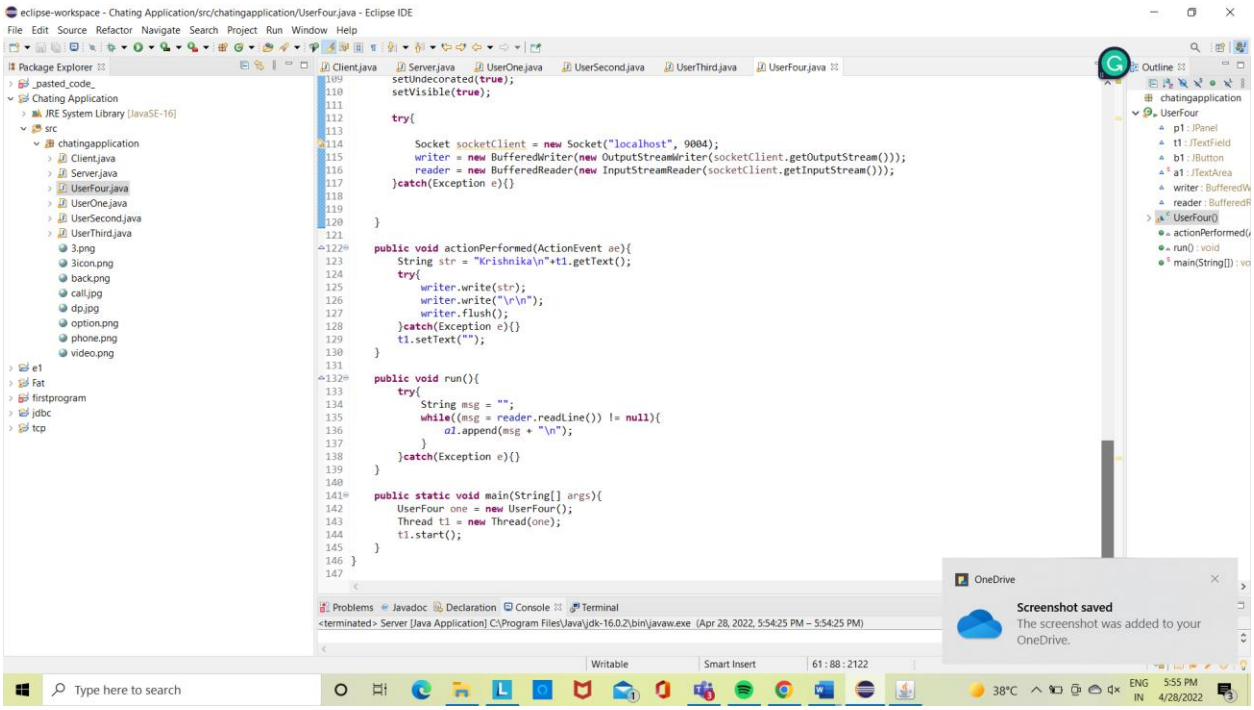

eclipse-workspace - Chating Application/src/chatingapplication/UserSecond.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
setVisible(true);
try{
Socket socketClient = new Socket("localhost", 9004);
writer = new BufferedWriter(new OutputStreamWriter(socketClient.getOutputStream()));
reader = new BufferedReader(new InputStreamReader(socketClient.getInputStream()));
}catch(Exception e){}
public void actionPerformed(ActionEvent ae){
String str = "Srividya\n"+t1.getText();
try{
writer.write(str);
writer.write("\r\n");
writer.flush();
}catch(Exception e){}
t1.setText("");
}
public void run(){
try{
String msg = "";
while((msg = reader.readLine()) != null){
a1.append(msg + "\n");
}
}catch(Exception e){}
}
public static void main(String[] args){
UserSecond one = new UserSecond();
Thread t1 = new Thread(one);
t1.start();
}
}
Outline
chatingapplication
UserSecond
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserSecond()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
Writable
Smart Insert
61 : 69 [19]
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Type here to search
38°C
ENG
IN
5:55 PM
4/28/2022

UserThird


eclipse-workspace - Chating Application/src/chatingapplication/UserThird.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
setVisible(true);
try{
Socket socketClient = new Socket("localhost", 9004);
writer = new BufferedWriter(new OutputStreamWriter(socketClient.getOutputStream()));
reader = new BufferedReader(new InputStreamReader(socketClient.getInputStream()));
}catch(Exception e){}
public void actionPerformed(ActionEvent ae){
String str = "Shakthi\n"+t1.getText();
try{
writer.write(str);
writer.write("\r\n");
writer.flush();
}catch(Exception e){}
t1.setText("");
}
public void run(){
try{
String msg = "";
while((msg = reader.readLine()) != null){
a1.append(msg + "\n");
}
}catch(Exception e){}
}
public static void main(String[] args){
UserThird one = new UserThird();
Thread t1 = new Thread(one);
t1.start();
}
}
Outline
chatingapplication
UserThird
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserThird()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
Writable
Smart Insert
27 : 96 : 657
Type here to search
38°C
ENG
IN
5:55 PM
4/28/2022

UserFour

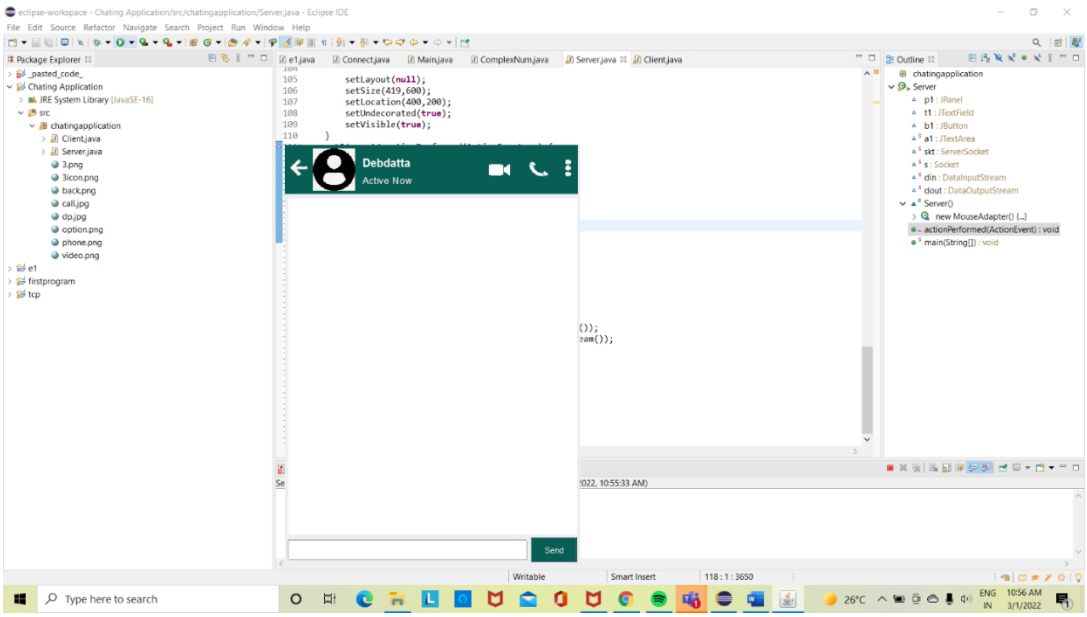

eclipse-workspace - Chating Application/src/chatingapplication/UserFour.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
package chatingapplication;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
public class UserFour extends JFrame implements ActionListener, Runnable{
JPanel p1;
JTextField t1;
JButton b1;
static JTextArea a1;
BufferedWriter writer;
BufferedReader reader;
UserFour(){
p1 = new JPanel();
p1.setLayout(null);
p1.setBackground(new Color(7, 94, 84));
p1.setBounds(0, 0, 450, 70);
add(p1);
ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("chatingapplication/3.png"));
Image i2 = i1.getImage().getScaledInstance(30, 30, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel l1 = new JLabel(i3);
l1.setBounds(5, 17, 30, 30);
p1.add(l1);
l1.addMouseListener(new MouseAdapter(){
public void mouseClicked(MouseEvent ae){
System.exit(0);
}
});
Outline
chatingapplication
UserFour
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserFour()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
Writable
Smart Insert
61 : 88 : 2122
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Type here to search
38°C
ENG
IN
5:55 PM
4/28/2022


eclipse-workspace - Chating Application/src/chatingapplication/UserFour.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
Chating Application
JRE System Library [JavaSE-16]
src
chatingapplication
Client.java
Server.java
UserFour.java
UserOne.java
UserSecond.java
UserThird.java
3.png
3icon.png
back.png
call.jpg
dp.jpg
option.png
phone.png
video.png
e1
Fat
firstprogram
jdbc
tcp
Client.java
Server.java
UserOne.java
UserSecond.java
UserThird.java
UserFour.java
setUndecorated(true);
setVisible(true);
try{
Socket socketClient = new Socket("localhost", 9004);
writer = new BufferedWriter(new OutputStreamWriter(socketClient.getOutputStream()));
reader = new BufferedReader(new InputStreamReader(socketClient.getInputStream()));
}catch(Exception e){}
}
public void actionPerformed(ActionEvent ae){
String str = "Krishnika\n"+t1.getText();
try{
writer.write(str);
writer.write("\r\n");
writer.flush();
}catch(Exception e){}
t1.setText("");
}
public void run(){
try{
String msg = "";
while((msg = reader.readLine()) != null){
a1.append(msg + "\n");
}
}catch(Exception e){}
}
public static void main(String[] args){
UserFour one = new UserFour();
Thread t1 = new Thread(one);
t1.start();
}
}
Outline
chatingapplication
UserFour
p1 : JPanel
t1 : JTextField
b1 : JButton
a1 : JTextArea
writer : BufferedW
reader : BufferedR
UserFour()
actionPerformed(
run() : void
main(String[]) : vo
Problems Javadoc Declaration Console Terminal
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (Apr 28, 2022, 5:54:25 PM – 5:54:25 PM)
Writable
Smart Insert
61 : 88 : 2122
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Type here to search
38°C
ENG
IN
5:55 PM
4/28/2022

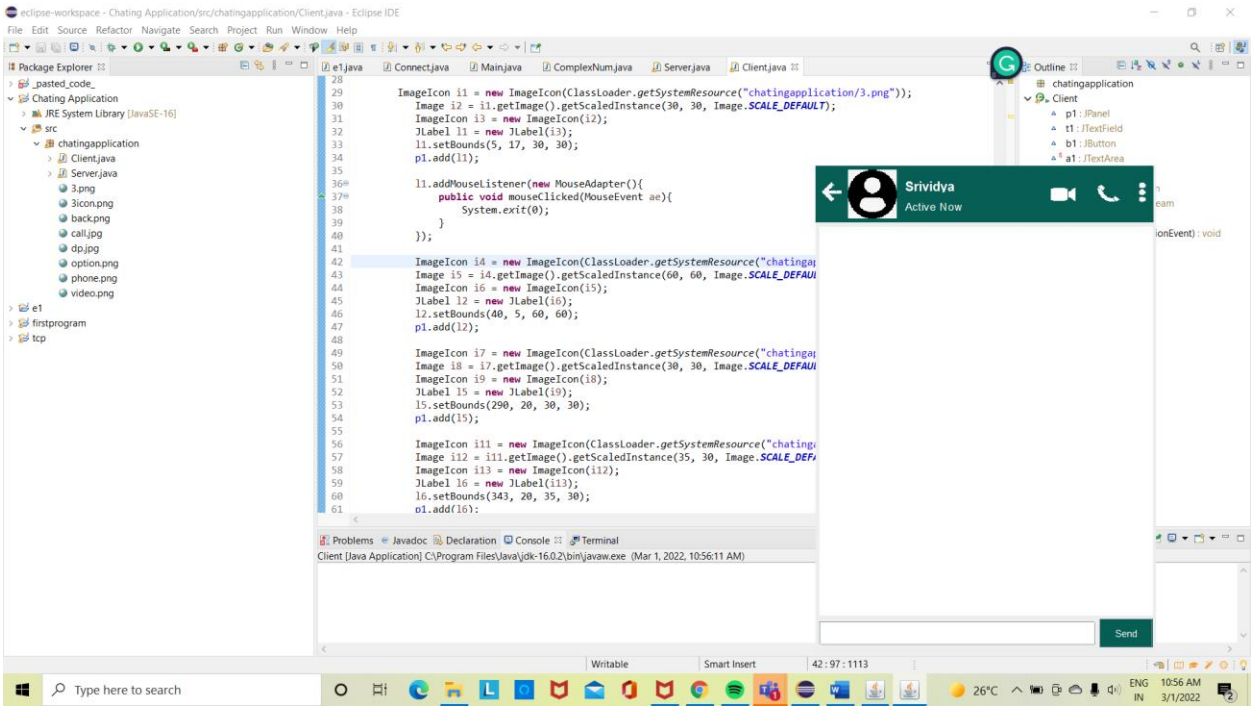
Output:

One-to-one chatting system

Server Screen:

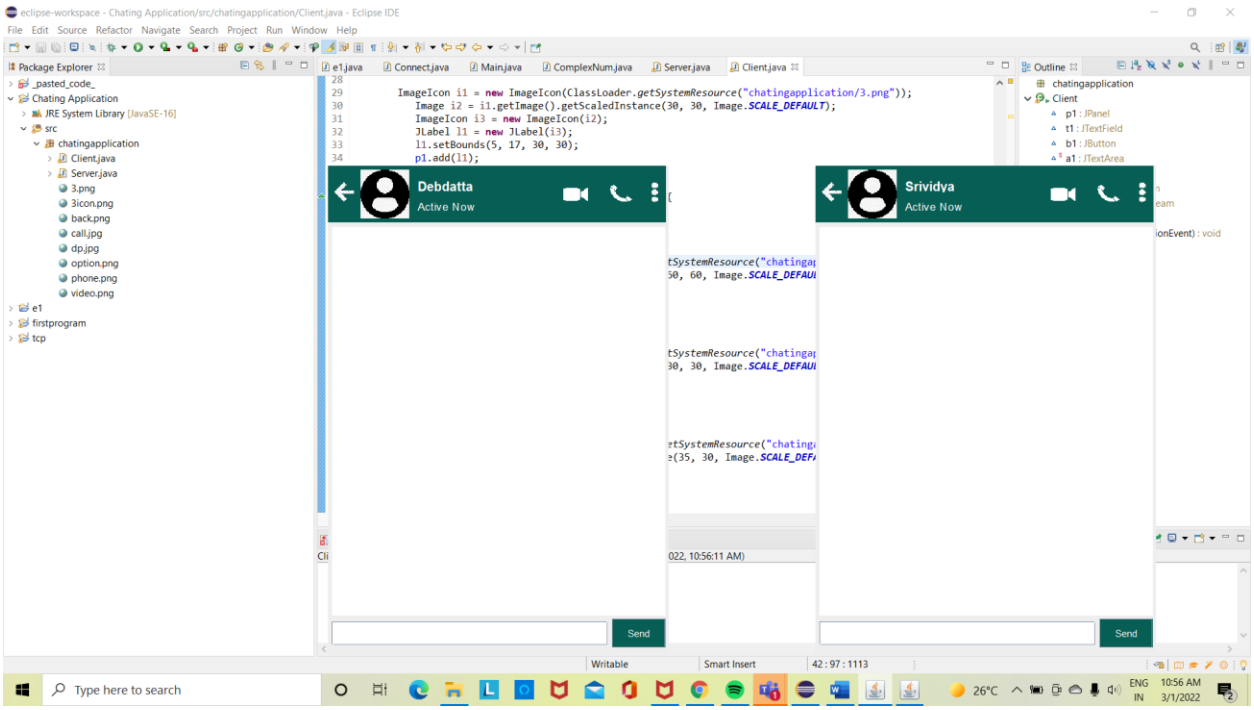


Client Screen:

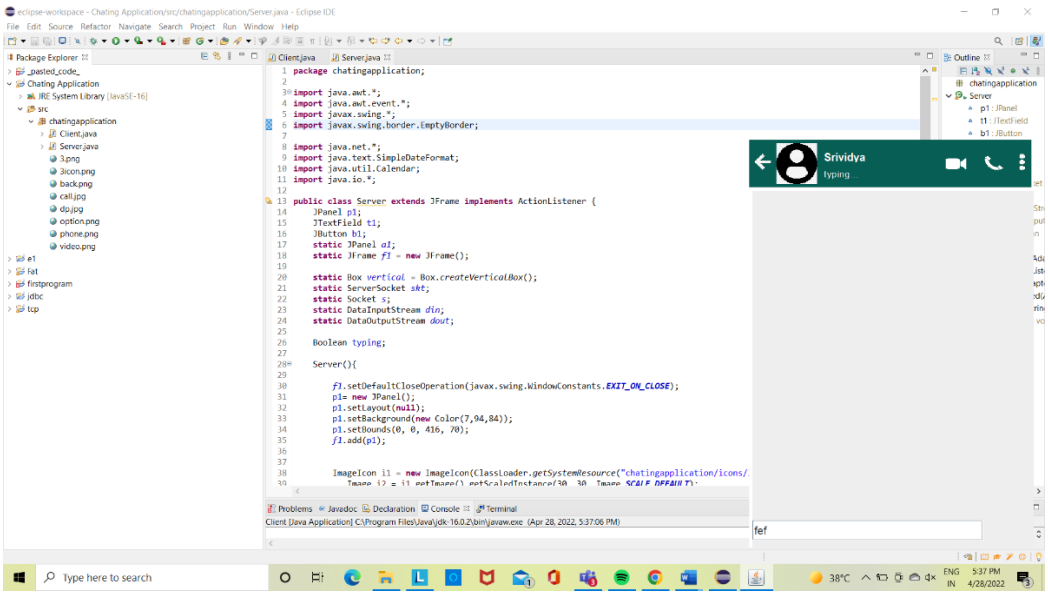


Server and Client Screen:

Client and Server are connected which enables one-to-one chatting. The messages sent by the user appears on the left side of the panel in a colored textbox for the receiver and vice versa.



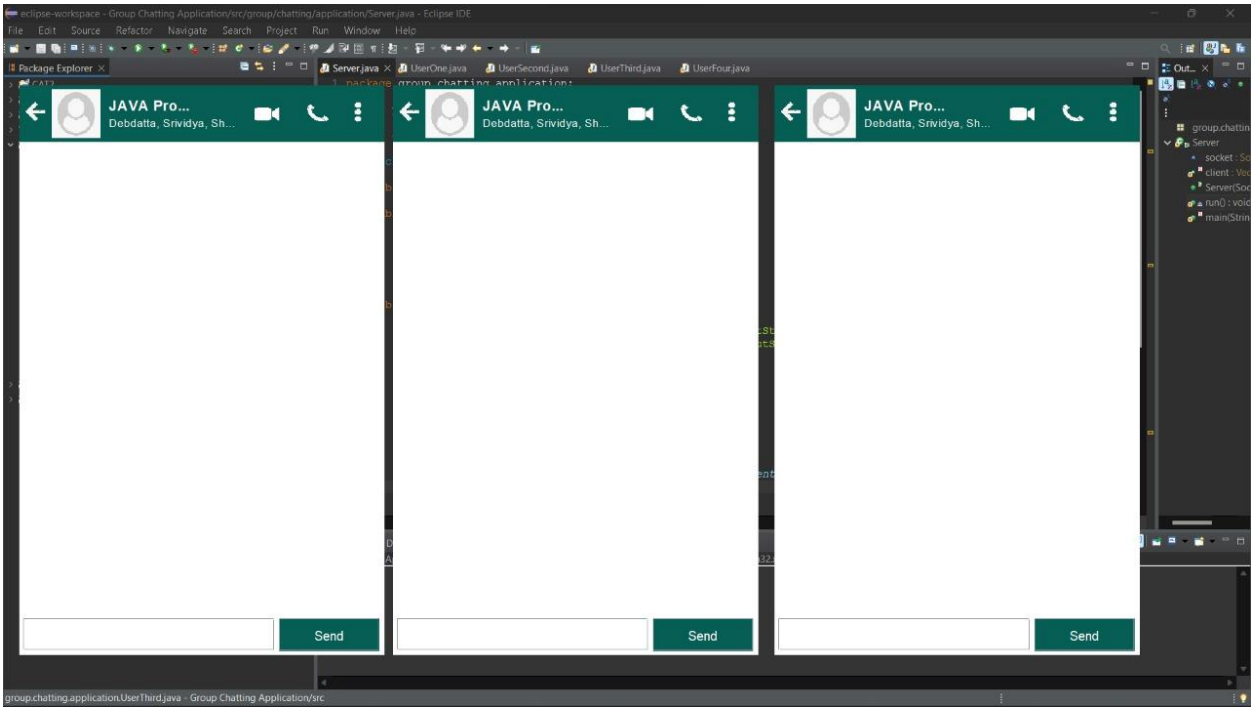
<u>When the user is typing</u>: Whenever the user types a text in the text field the status is changed to ‘typing’ . It changes back to ‘Active now’ within a second of stopping to type. This is enabled by thread sleep(1000).



Group Chatting :

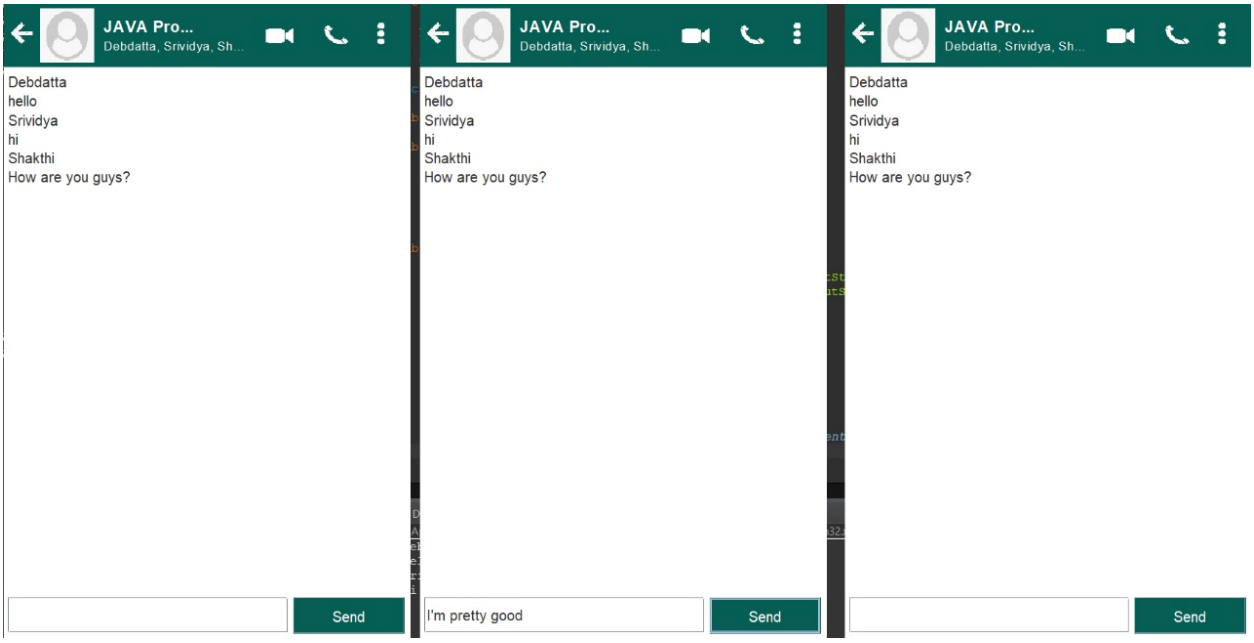
The concepts used for one-to-one chatting are extended to introduce a group chatting application where multiple users can converse in a group.

A server class is created which connects the user classes(UserOne, UserSecond, UserThird, UserFour) using threads and networking.



All the three users are connected. The messages sent by the users are displayed to others along with name and timestamp at the left side of the panel.

For every user that starts typing, the status is changed from ‘Active now’ to ‘typing…’


JAVA Pro...
Debdatta, Srividya, Sh...
Debdatta
hello
Srividya
hi
Shakthi
How are you guys?
Send
JAVA Pro...
Debdatta, Srividya, Sh...
Debdatta
hello
Srividya
hi
Shakthi
How are you guys?
I'm pretty good
Send
JAVA Pro...
Debdatta, Srividya, Sh...
Debdatta
hello
Srividya
hi
Shakthi
How are you guys?
Send