

Eric Wilson

Project13

ChitChat.java

```
import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.net.Inet4Address;
import java.net.UnknownHostException;

import javax.swing.*;
import javax.swing.border.Border;

public class ChitChat extends JFrame {
    private static final int LARGE_FONT_SIZE = 28;
    private static final int MEDIUM_FONT_SIZE = 20;
    String addressIP4 = null;

    // defined at the class level so addAMessage()
    // can update it with incoming messages.
    private JPanel messagesReceivedPanel;

    public ChitChat() {
        setTitle("Chit-Chat");

        // The default layout for the content pane
        // of a JFrame is BorderLayout
        Container contentPane = getContentPane();

        try {
            addressIP4 = Inet4Address.getLocalHost().getHostAddress();
        } catch (UnknownHostException e) {
            e.printStackTrace();
        }
        // Note, to find our IP address under Windows
        // execute ipconfig from cmd line.
        JLabel ipLabel = new JLabel(addressIP4);
        ipLabel.setFont(new Font("SansSerif", Font.PLAIN, LARGE_FONT_SIZE));
        ipLabel.setHorizontalAlignment(SwingConstants.CENTER);
        contentPane.add(ipLabel, BorderLayout.NORTH);

        JComponent scrollableMessagesReceived = buildMessagesReceivedPanel();
        contentPane.add(scrollableMessagesReceived, BorderLayout.CENTER);

        JComponent sendMessagePanel = buildSendMessagePanel();
        contentPane.add(sendMessagePanel, BorderLayout.SOUTH);
    }

    private JComponent buildMessagesReceivedPanel() {
        messagesReceivedPanel = new JPanel();

        messagesReceivedPanel.setLayout(new GridBagLayout());

        //addAMessage("From:Larry @ 134.193.12.34\nHey Jimmy! What's up? Want to go play frisbee?");
        //addAMessage("From:Mindy @ 134.193.52.77\nJimmy, can you help me with my Java homework?");

        JScrollPane scrollPane = new JScrollPane(messagesReceivedPanel);

        Border titledBorder = BorderFactory.createTitledBorder("Receive");
        Border compoundBorder = BorderFactory.createCompoundBorder(
            titledBorder,
            scrollPane.getBorder());
        scrollPane.setBorder(compoundBorder);

        return scrollPane;
    }

    public void addAMessage(String msg) {
        GridBagConstraints constraints = new GridBagConstraints();

        // Defaults
        constraints.gridx = 0;
        constraints.gridy = GridBagConstraints.RELATIVE;
```

```

constraints.weightx = 1;
constraints.weighty = 1;
constraints.insets = new Insets(2,2,2,2);
constraints.fill = GridBagConstraints.HORIZONTAL;
constraints.anchor = GridBagConstraints.NORTH;

JTextArea messageTextArea = new JTextArea(msg,4,20);
messageTextArea.setEditable(false);
messageTextArea.setLineWrap(true);
messageTextArea.setWrapStyleWord(true);

messageTextArea.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
JScrollPane sourceScrollPane = new JScrollPane(messageTextArea);
JScrollBar vertical = sourceScrollPane.getVerticalScrollBar();
vertical.setValue( vertical.getMaximum() );
messagesReceivedPanel.add(sourceScrollPane,constraints);
this.validate();

}

private JComponent buildSendMessagePanel() {
    JPanel messagePanel = new JPanel();

    messagePanel.setLayout(new GridBagLayout());

    GridBagConstraints constraints = new GridBagConstraints();

    // Defaults
    constraints.weightx = 0;
    constraints.weighty = 0;
    constraints.fill = GridBagConstraints.NONE;
    constraints.insets = new Insets(2,2,2,2);
    constraints.anchor = GridBagConstraints.NORTHWEST;

    JLabel ipLabel = new JLabel("Receiver's IP:");
    ipLabel.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
    constraints.gridx = 0;
    constraints.gridy = 0;
    constraints.gridwidth = 1;
    constraints.gridheight = 1;
    messagePanel.add(ipLabel,constraints);

    constraints.gridy = 1;
    JLabel senderNameLabel = new JLabel("Sender's Name:");
    senderNameLabel.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
    messagePanel.add(senderNameLabel,constraints);

    constraints.weighty = 1;
    constraints.gridy = 2;
    JLabel messageLabel = new JLabel("Message:");
    messageLabel.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
    messagePanel.add(messageLabel,constraints);

    constraints.gridx = 1;
    constraints.gridy = 0;
    constraints.gridwidth = 3;
    constraints.weightx = 1;
    constraints.weighty = 0;
    constraints.fill = GridBagConstraints.HORIZONTAL;
    JTextField ipTextField = new JTextField(20);
    ipTextField.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
    messagePanel.add(ipTextField,constraints);

    constraints.gridy = 1;
    JTextField senderNameTextField = new JTextField(20);
    senderNameTextField.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
    messagePanel.add(senderNameTextField,constraints);

    constraints.gridy = 2;
    constraints.weighty = 1;
    JTextArea messageTextArea = new JTextArea(5,20);
    messageTextArea.setLineWrap(true);
    messageTextArea.setWrapStyleWord(true);

    messageTextArea.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));

```

```

JScrollPane sourceScrollPane = new JScrollPane(messageTextArea);
messagePanel.add(sourceScrollPane, constraints);

constraints.weighty = 0;
constraints.gridy = 3;
constraints.fill = GridBagConstraints.NONE;
JButton sendButton = new JButton("Send");

sendButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        String newMessage = new String(senderNameTextField.getText() + " @ "
            + addressIP4 + "\n" + messageTextArea.getText());
        Client client = new Client();
        try {
            client.sendMessage(ipTextField.getText(), newMessage);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
});

sendButton.setFont(new Font("SansSerif", Font.PLAIN, MEDIUM_FONT_SIZE));
messagePanel.add(sendButton, constraints);

Border titledBorder = BorderFactory.createTitledBorder("Send");
messagePanel.setBorder(titledBorder);
return messagePanel;
}

public static void main(String[] args) {
    ChitChat frame = new ChitChat();
    frame.pack();
    frame.setVisible(true);
    LittleServer server = new LittleServer(frame);
    server.listen();
}
}

```

LittleServer.java

```

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

public class LittleServer {

    private static final int PortNumber = 8086;
    static ChitChat chitchat;

    LittleServer(ChitChat cc) {
        chitchat = cc;
    }

    public static void listen () {
        try {
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            ServerSocket ss = new ServerSocket(PortNumber);
            System.out.println("Listening on port: " + PortNumber);

            while (true) {
                Socket clientSocket = ss.accept();
                String hostName = clientSocket.getInetAddress().getHostName();

                OutputStream out = clientSocket.getOutputStream();
                InputStream in = clientSocket.getInputStream();

                int nextByte = in.read();
                while (nextByte != '\n') {
                    baos.write(nextByte);
                    System.out.write(nextByte);
                    nextByte = in.read();
                }

                String Sender = baos.toString();
                baos.reset();
            }
        }
    }
}

```

```

        System.out.print("CONVERTED STRING: " + Sender + "\n");

        nextByte = in.read();
        while (nextByte != -1) {
            baos.write(nextByte);
            System.out.write(nextByte);
            nextByte = in.read();
        }

        String baosString = baos.toString();

        baos.reset();

        System.out.print("CONVERTED STRING: " + baosString + "\n");

        //build string
        String newMessage = new String("From:" + Sender + "\n" + baosString);

        chitchat.addAMessage(newMessage);
        System.out.println(hostName + " disconnecting");

        if (in != null)
            in.close();
        if (out != null)
            out.close();
        if (clientSocket != null)
            clientSocket.close();
    }
}
catch (Exception e) {
    System.out.println("Error in RemoteConnection thread:" + e);
}
}
}

```

Client.java

```

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

// Connect to: http://sce.umkc.edu/BIT/burris/cs349/lessons/networking/
public class Client {
    Client() {
        System.out.print("Client created. It's pretty nice.\n");
    }
    public static void sendMessage (String server, String message) throws IOException {
        String serverName = server;
        int serverPort = 8086;

        try {
            Socket socketToServer = new Socket(serverName, serverPort);
            System.out.println("SocketToServer: " + socketToServer);

            OutputStream out = socketToServer.getOutputStream();
            BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(out));

            InputStream in = socketToServer.getInputStream();
            BufferedReader reader = new BufferedReader(new InputStreamReader(in));


            bw.write(message + "\n");
            // Machine hosts multiple domains on the same IP address
            //bw.write("Host:" + serverName + "\n\n");
            bw.flush();

            //String line = null;
            //while ((line = reader.readLine()) != null) {
            //    System.out.println(line);
            //}

            bw.close();
            in.close();
            out.close();
            socketToServer.close();
        }
        catch (Exception e) {
            System.out.println("Ex: " + e);
        }
    }
}

```

```
}  
}  
}
```

 Chit-Chat — □ ×

192.168.1.177

Receive

From:Eric @ 192.168.1.177
HI ERIC


Send

Receiver's IP: 192.168.1.177

Sender's Name: Eric

Message: HI ERIC

Send

Chit-Chat

X

192.168.1.177

Receive

From:Eric @ 192.168.1.177
HI ERIC

From:Eric @ 192.168.1.177
You made a thing!

Send

Receiver's IP:192.168.1.177

Sender's Name:Eric

Message:You made a thing!

Send

I tested this over my local network also. It worked for a little bit and then broke everything until I reset my router. I'm not sure exactly why and I think that might be outside of the scope of where we're at