

```

package cop2805;

import javax.swing.*.*;
import javax.swing.border.EmptyBorder;
import java.awt.*.*;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.Socket;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;

public class Client
{
    private static void constructGUI()
    {
        JFrame.setDefaultLookAndFeelDecorated(true);
        MyFrame frame = new MyFrame();
        frame.setVisible(true);
    }

    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(Client::constructGUI);
    }
}

class MyFrame extends JFrame
{
    public JTextField userInput;
    public JButton btnTransmit;
    public JLabel instances;
    public DefaultListModel<Integer> listModel;
    public JList<Integer> list;

    public MyFrame()
    {
        super();
        init();
    }

    private void init()
    {
        userInput = new JTextField();
        btnTransmit = new JButton("Transmit");
        instances = new JLabel();
        listModel = new DefaultListModel<>();
        list = new JList<>(listModel);

        btnTransmit.addActionListener(event ->
        {
            listModel.clear();
            try {
                String userString = userInput.getText();
            }
        }
    }
}

```

```

        Socket connection = new Socket("127.0.0.1", 1236);
        OutputStream output = connection.getOutputStream();

        output.write(userString.length());
        output.write(userString.getBytes());

        BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));

        String serverResponse = "";
        while (reader.readLine() != null)
        {
            String occurrences = readFile("hamlet-
index.txt").get(0).replace("[", "").replace("]", "");
            String[] occArr = occurrences.split(", ");

            for(String value: occArr)
            {
                listModel.addElement(Integer.parseInt(value.trim()));
            }
        }

        if (!connection.isClosed())
            connection.close();

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

this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setTitle("Word Searcher");

this.setLayout(new BorderLayout());

JPanel inputPanel = new JPanel(new GridLayout(2, 1));
inputPanel.add(new JLabel("Find:"));
inputPanel.add(userInput);
inputPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
this.add(inputPanel, BorderLayout.NORTH);

JScrollPane responsePane = new JScrollPane();
responsePane.setViewportView(list);
list.setLayoutOrientation(JList.VERTICAL);
JPanel responsePanel = new JPanel(new GridLayout(2, 1));
responsePanel.add(new JLabel("Response:"));
responsePanel.add(responsePane);
responsePanel.setBorder(new EmptyBorder(0, 10, 0, 10));
this.add(responsePanel, BorderLayout.CENTER);

JPanel searchBtn = new JPanel();
searchBtn.add(btnTransmit);
searchBtn.setBorder(new EmptyBorder(10, 10, 10, 10));
this.add(searchBtn, BorderLayout.SOUTH);

int frameWidth = 500;

```

```

        int frameHeight = 250;
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        this.setBounds((int) (screenSize.getWidth() / 2) - (frameWidth / 2),
            ((int) (screenSize.getHeight() / 2) - frameHeight / 2),
            frameWidth,
            frameHeight);
    }

    public static List<String> readFile(String file)
    {
        List<String> lines = null;
        try
        {
            lines = Files.readAllLines(Paths.get(file));
            System.out.println(lines);
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
        return lines;
    }
}

```