

223CCS Lab Mini Project

Student ID: _

Name: Engineer Fayez Al-Qarni Student

Java GUI-Based Client-Server Application for Currency Conversion

Problem Statement:

Create a Java GUI-based application to implement a **Client-Server** system for converting currency. The system should work as follows:

1. **Server:**
 - Listens for connections on **port 1234**. ○ Accepts an integer input from the client, representing an amount in **USD**.
 - Converts the USD amount to its equivalent in **SAR** using the conversion rate of **1 USD = 3.75 SAR**.
 - Sends the converted SAR amount back to the client. ○ Ends the communication when the client sends the value **0** and responds with "Bye".
2. **Client:**
 - Provides a **Graphical User Interface (GUI)** for the user to input the amount in USD. ○ Displays the converted SAR amount received from the server in a text area.
 - Ends the session when the user inputs **0**, displaying a goodbye message.

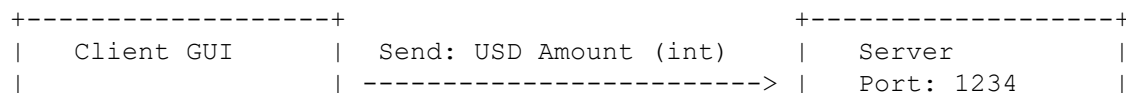
Functional Requirements:

1. The **server** must handle continuous client requests until the client sends 0.
2. The **client GUI** must:
 - Accept USD input from the user.
 - Display the SAR equivalent sent by the server.
 - Terminate the session gracefully when 0 is entered.

Figure:

Below is a graphical representation of the problem:

sql Copy
code



- Input: USD			- Convert: USD -> SAR	
- Display: SAR		<-----	- Respond: SAR Amount	
- Terminate on 0		Receive: SAR Amount	- Say "Bye" on 0	
+-----+			+-----+	

Design Considerations:

1. **Server:**
 - o Implemented using `ServerSocket` to listen on port 1234.
 - o Uses a loop to process continuous client requests.
2. **Client:**
 - o Designed with a simple GUI using `JFrame`.
 - o Sends user input to the server via `Socket`.
 - o Displays the server's response in a `JTextArea`.

Example Interaction:

1. The client sends 10 USD. o **Server Response:** 37.5 SAR.
 2. The client sends 0.
 - o **Server Response:** "Bye" and terminates the session.
-

Tasks: 1. Implement the Server:

- o Write a Java program to handle client connections, perform currency conversion, and terminate upon receiving 0.
2. **Design the Client GUI:**
 - o Develop a Java program with a GUI to input USD values, send them to the server, and display the SAR equivalent.
3. **Submit:**
 - o Source code for both the server and client.

Client

```
package com.mycompany.client;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import java.net.*;
```

```

public class Client {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Currency Converter (USD to SAR)");
        JTextField inputField = new JTextField(10);
        JButton sendButton = new JButton("Convert");
        JTextArea resultArea = new JTextArea(10, 30);
        resultArea.setEditable(false);

        frame.setLayout(new java.awt.FlowLayout());
        frame.add(new JLabel("Enter amount in USD:"));
        frame.add(inputField);
        frame.add(sendButton);
        frame.add(new JScrollPane(resultArea));

        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        sendButton.addActionListener(new ActionListener() {
            Socket socket = null;
            PrintWriter out = null;
            BufferedReader in = null;

            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    if (socket == null) {
                        socket = new Socket("localhost", 1234);
                        out = new PrintWriter(socket.getOutputStream(), true);
                        in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                    }

                    String inputText = inputField.getText();
                    int usd;

                    try {
                        usd = Integer.parseInt(inputText);
                    } catch (NumberFormatException ex) {
                        resultArea.append("Invalid input. Please enter a valid
integer.\n");
                    }

                    return;
                }
            }
        });
    }
}

```

```

        out.println(usd);
        String response = in.readLine();

        if (response.equals("Bye")) {
            resultArea.append("Server: Bye\n");
            socket.close();
            socket = null;
            return;
        }

        resultArea.append("Converted: " + usd + " USD = " + response
+ " SAR\n");

    } catch (IOException ioException) {
        resultArea.append("Error: " + ioException.getMessage() +
"\n");
    }
}
});
}
}

```

Server:

```

package com.mycompany.server;
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) {
        int port = 1234;
        double conversionRate = 3.75;

        try (ServerSocket serverSocket = new ServerSocket(port)) {
            System.out.println("Server is listening on port " + port);

            while (true) {
                try (Socket socket = serverSocket.accept();
                    BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                    PrintWriter out = new PrintWriter(socket.getOutputStream(),
true)) {

```

```

        System.out.println("New client connected");

        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            int usd = Integer.parseInt(inputLine);

            if (usd == 0) {
                out.println("Bye");
                System.out.println("Client disconnected.");
                break;
            }

            double sar = usd * conversionRate;
            out.println(sar);
            System.out.println("Converted " + usd + " USD to " + sar
+ " SAR");
        }
    } catch (IOException | NumberFormatException e) {
        System.out.println("Error handling client: " +
e.getMessage());
    }

    } catch (IOException e) {
        System.out.println("Server exception: " + e.getMessage());
    }
}
}

```

- A screenshot showing the GUI in action, with interaction logs.

