

## Assignment 1 and 2 Small Project

### Chatting Between Peers

Spring 2021

Dr. Raed Alqadi and Dr. Abdullah Rashed

In this project Peer to Peer Application that runs on a Network. You will Write it in Java.

#### Project Parts (Assignments):

##### 1. Part 1 (HW1)

In this part, you are required to write Peer-to-peer Application Chatting Java UDP Socket Programming: Write a GUI application by using Java. Your program should have Text boxes, Buttons, Text Areas, Drop Downs etc. You should enter Source and destination IP addresses, port numbers. Also, you should display sent and received messages. See the sample for GUI Interface on [Page 2](#). Two clients Are shown. The Buttons for Login, Logout, and the List of Online users is Not Required for This part. They are Required in Par2. Blocking Code is Not Accepted.

#### Page 2 Shows Two Clients and sample GUI

##### 2. Part 2 (HW2):

In this part you are required to apply TCP Java Socket programming but we will add Two Parts:

2.1 A TCP Client to the Application described in 1. This Client requires the additions of buttons to register to the TCP Server Described in 2.2. The Server will simply keep a list of UDP clients involved in the Chatting. The TCP Server will send a message to each TCP client in the Active Chatting Client described in 1 to inform it of the List of Active Clients. This is similar to Skype and other chatting Servers. The server just keeps a list of those that Active Chatting Clients. So, you will need to modify the code in 1 to accommodate this Requirement.

The Actual chatting will remain peer-to-peer. But the User chooses which Client to talk to from the List Provided by the Server. The login in the Clients is used for Registration to the TCP server to keep track of online Clients. You Should display the List of Active Client. [See pages 3, 4](#)

Add a TCP Server that keeps track of the Active Chatting Clients as described in 2.1. It should Show a GUI with the Active List of Clients. The login in the Clients is used for Registration. [See pages 3, 4](#)

**Blocking Code is Not Accepted.**

**Part1(HW1):**This Figure Shows Two Clients used in Part 1 and 2. The Login/Logout and TCP Server are not required in Part1 but will be Required in Part2.

The screenshot shows the 'ClientChat' application window. At the top, there's a title bar with a green icon and the text 'ClientChat'. Below the title bar, on the left, is a 'Username:' label followed by an empty text box, a red 'Login' button, and a grey 'Logout' button. In the center, there's a large text area containing three lines of chat history: 'Rem: Hello from 6000' (blue), 'Rem: Hello2' (blue), and 'Me: Hoeeelo from 5000 1244444444444' (red). Below this is a text input field with the placeholder 'enter text here'. On the right side, there are configuration fields: 'TCP Server IP:' with '127.0.0.1', 'TCP Server Port:' with '8888', 'Available Interfaces' with a dropdown menu showing 'Wi-Fi:192.168.1.112', 'Local IP:' with '192.168.1.112', 'Local Port:' with '5000', 'Remote IP:' with '192.168.1.112', and 'Remote Port:' with '6000'. Below these are 'Send' and 'test Button' buttons. On the far right, there's an 'Online Users' label above an empty list box. At the bottom, a 'Status:' label is followed by a text box displaying 'Received From:IP=192.168.1.112, Port = 6000'.

This screenshot shows the 'ClientChat' application window after a configuration change. The layout is identical to the first screenshot, but the 'Available Interfaces' dropdown menu now shows 'Ethernet:169.254.49.56'. The chat history in the central text area has been updated: 'Me: Hello from 6000' (red), 'Me: Hello2' (red), and 'Rem: Hoeeelo from 5000 1244444444444' (blue). The 'Status:' text box at the bottom now displays 'Received From:IP=192.168.1.112, Port = 5000', indicating a change in the received data.

**Part 2(HW2):** Here we show The TCP Server and # Chatting Clients each has a UDP and a TCP Client in the Code. Same Code for the Clients

TCPServerN

Start Listening Port: 8888

Loopback Pseudo-Interface 1: 127.0.0.1

127.0.0.1,7000  
127.0.0.1,5000  
127.0.0.1,8000

Status Address:127.0.0.1 Port:8888

## Client1

ClientChat

Username:  Login Logout

Me: Hi From 7000  
Rem: Hi From 8000

enter text here

Status: Received From:IP=127.0.0.1, Port = 8000

TCP Server IP: 127.0.0.1  
TCP Server Port: 8888

Available Interfaces  
Loopback Pseudo-Interface 1:127.0.0.1

Local IP: 127.0.0.1  
Local Port: 7000  
Remote IP: 127.0.0.1  
Remote Port: 5000

Send test Button

Online Users  
127.0.0.1,7000  
127.0.0.1,5000  
127.0.0.1,8000

## Client 2:

ClientChat

Username:

Login

Logout

Me: Hello from 5000

Rem: Hi From 7000

enter text here

Send

test Button

Status:

Received From:IP=127.0.0.1, Port = 7000

TCP Server IP:

TCP Server Port:

Available Interfaces

Loopback Pseudo-Interface 1:127.0.0.1

Local IP:

Local Port:

Remote IP:

Remote Port:

Online Users

127.0.0.1,7000

127.0.0.1,5000

127.0.0.1,8000

## Client 3

ClientChat

Username:

Login

Logout

Rem: Hello from 5000

Me: Hi From 8000

enter text here

Send

test Button

Status:

Received From:IP=127.0.0.1, Port = 5000

TCP Server IP:

TCP Server Port:

Available Interfaces

Loopback Pseudo-Interface 1:127.0.0.1

Local IP:

Local Port:

Remote IP:

Remote Port:

Online Users

127.0.0.1,7000

127.0.0.1,5000

127.0.0.1,8000