



Department Name: Computer Engineering

Course name : Networks 1

Course Number: 10636454

Programming HW 2
Chatting Between Peers

Instructor Name:

Academic Year:2021/2022

Semester: 2nd

Credit Hours: 3

Date: 4/4/2022

Student Name:.....

Registration Number:.....

Section:

Exam Weight:15%

Question	Points	CLO	Question Grade	Required Time
Q1	6	iii		
Q2	9	iii		
Student Grade				

Exam Notes:

- 1- Close Books & Notes.
- 2- Read each problem carefully before attempting to solve it.
- 3- Write all work on this exam paper.

Good Luck

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

Project Parts (Assignments):

1. Part 1 (Question1) Deadline 17/4/2022 midnight

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 3](#). 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 (Question2)

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

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 4, 5](#)

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 4, 5](#)

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. On the left, there's a chat log with three messages: 'Rem: Hello from 6000' (blue), 'Rem: Hello2' (blue), and 'Me: Hooelo from 5000 124444444444' (red). Below the log is a text input field with the placeholder 'enter text here'. At the bottom left, a status bar displays 'Received From:IP=192.168.1.112, Port = 6000'. On the right side, there are configuration fields: 'TCP Server IP' (127.0.0.1), 'TCP Server Port' (8888), 'Available Interfaces' (a dropdown menu showing 'Wi-Fi:192.168.1.112'), 'Local IP' (192.168.1.112), 'Local Port' (5000), 'Remote IP' (192.168.1.112), and 'Remote Port' (6000). There are 'Login' and 'Logout' buttons at the top left, and 'Send' and 'test Button' at the bottom right. An 'Online Users' list is on the far right.

This screenshot shows the same 'ClientChat' application window after the 'Available Interfaces' dropdown has been changed to 'Ethernet:169.254.49.56'. The chat log now shows three messages: 'Me: Hello from 6000' (red), 'Me: Hello2' (blue), and 'Rem: Hooelo from 5000 124444444444' (blue). The status bar at the bottom left now displays 'Received From:IP=192.168.1.112, Port = 5000'. The other configuration fields and buttons remain the same as in the previous screenshot.

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

Status:

TCP Server IP:

TCP Server Port:

Available Interfaces

Loopback Pseudo-Interface 1:127.0.0.1

Local IP:

Local Port:

Remote IP:

Remote Port:

Send

test Button

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

Status:

TCP Server IP:

TCP Server Port:

Available Interfaces

Loopback Pseudo-Interface 1:127.0.0.1

Local IP:

Local Port:

Remote IP:

Remote Port:

Send

test Button

Online Users

127.0.0.1,7000

127.0.0.1,5000

127.0.0.1,8000