An-Najah National University

Faculty of Engineering

جامعة النجاح الوطنية

كلية الهندسة

Computer Engineering Department

Computer Networks 1(10636454)

Course Mini-Project & Homework

Dr. Raed Alqadi

Instructor Name: Dr. Raed Alqadi
Academic Year:2017/2018
Semester: Summer
Credit Hours: 3
Date: Part1: July 6
Due Date Part2: July 17

Student Name: Ahmad
Registration Number:
Section: 2
Total Project Mark:
Project +HW Weight: 15%

Student Grades

Part	Description	Points	ILO's		Part Grade
Part 1	Basic Network Programming		2		
Part 2	Client Server and Peer to Peer chat		2		
Student Grade (of 30)					

Project Notes:

1- Submit every part of the Program (Software on time)
2- Use good programming practices and style.
3- Read the specs of the program on the next page carefully.

Project: Network Programming

In this project, you will develop a Reliable Client-Server and Peer to Peer Application that runs on a Network. To simplify the process of developing the application, this project will be composed of 2 assignments and you will be graded on each part. You must submit each part on due time. The due time of each part will be announced to you in class according to the parts covered during the course.

Project Parts (Assignments):

1. **Part 1 (HW1):** In this part you are required to apply Java Socket programming to develop Client and Server Applications by using Java
   1.1 Server:
   - The Server shall be a Multi-threaded Server with GUI.
   - The GUI should allow the User to Select the Port Number for the Server and the ability for the Server to chat with any Client. There should be Text Areas for the Send and Receive Messages. Also, it should show the list of Active Clients. Add buttons for Send, initialize, and any other buttons an GUI elements you wantyou need.
   - The Server should have a main thread for the Listening Socket.
   - The Server must create a new thread for each client that establishes a connection with the server. Sequential Server is Not permitted. Chatting with a client should be handled from the thread created for this client.
   - Keep a List of the active clients and Display it.
   - Any client can send a message to the server and the server can send and receive messages to that client. Use chatting for messages.
   - Clients Cannot talk to each other

   1.2 Client: The client should create a connection to the server and should be able to send and receive messages.
   - The Client can create and close the connection.
   - The Client can chat with the Server.
   - There should be GUI with Text Areas for Sed and Receive Messages and also required buttons and text boxes to enter the IP and Port number of the server.

2. **Part 2 (HW2)** In this part, you are required to modify the Applications above such that they have TCP Client Server interaction as well as peer to peer UDP chat program. You will need to modify the Server and Client Apps described above as follows.
   2.1 TCP Client/Peer-to-peer application:
   The old client in hw1 will become will have a TCP client and a UDP peer to peer .
   - The TCp Client creates a connection with the TCP server and registers its UDP IP address and its UDP port number. The server will inform all clients of which clients are active.
   - After registering with the TCP clients, it should be able to see all clients that are online and it should be able to select a client to chat with by using UDP. The two clients will chat with each other as peer to peer by using UDP. The list of active clients is provided by the server.

   2.2 Server:
   - The TCP server will only do the book-keeping of the clients that want to chat with each other as peer to peer.

- The Server accepts registration requests from TCP clients in the Client/UDP-Peer-to-Peer applications. The TCP server keeps track of “who is online” and sends this information to all clients. The second program is the TCP-Client/UDP-Peer-to-Peer. Your TCP-Client/UDP-Peer-to-Peer should first register at the TCP server to get the online clients. control to select a file to be transferred. Also develop a simple protocol to inform anther client of the file transfer request.