DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ENCS3320 - Computer Networks
Course Project

Dr. Mohammad Helal

Project Title : Peer-to-Peer-based Chatting Application
Team Size : 3 members, if 4 then extra requirement applies
Submission Due : Aug 25, 2022
Total Score : 25 points

Project Description:
Develop a *UDP*-based *Chatting Application* based on a *Peer-to-Peer* architecture. Although it is a P2P system, it would still need a *Server* in order to provide connectivity between the *Clients*, and keep tracking on who's online and using what *IP* and *Port* numbers.

- The system would require a *Server* application that keeps listening to a *UDP Socket*, when a *Client* is active, it sends its *User ID* in a *UDP Datagram* to the *Server*, then *Server* saves *IP*, *Port #* and *User ID* in a *List* or an *Array*.
- *Client* keeps sending the aforementioned message every 5 seconds, *Server* keeps updating the entry in the *List*, every time change has happened, it sends an updated *List* to all online *Clients*.
- An updated version of *List* of online *Clients* is sent to every online Client in the system, Client application presents the list of User IDs only, it does not need to display the *IP* and *Port #.*
- *Client* then allows the user to select target *Client* to chat with, either by double-clicking on his record, or by writing target *Client User ID* in some *Text Box*.
- After selecting target *User*, *Client application* send *UDP* message to *Server* asking for target *IP* and *Port #.*
- *Server* responds using UDP message that includes target *IP* and *Port #.*
- Once *Client* receives target IP and Port #, it sends a UDP message to the target.
- When a *Client* receives a message, it displays it in a text box right next to the User ID of the sender.

Extra requirement for teams with 4 students:
4-member-teams should use *Threads* to handle message sending and message receiving services separately and in parallel fashion.

Submission Requirements:

1. **Report (5 points)**
   Report Explaining the structure of the project, the methods used, the format of the messaging (the suggested Protocol). Provide screenshots and explain the codes.
2. **The Project Files (10 points)**
   Includes at least two files, *Server* and *Client*.
3. **Project Discussion (10 points)**
   There will be a face-to-face discussion with the *Teaching Assistants.* Students are expected to show deep understanding of every part of the project, please be hesitant to say “my partner did this part”, because its another way to say “I dont deserve the full score because I dont understand the whole project” :)

Best Luck ^^