Instructions

Download and unzip the example project: C# TCP Chat Program.zip

Inside the zip file you will finda project that allows a user to choose between being a server or client for the chat application. The server accepts connections and broadcasts messages to all clients. Clients can connect to the server, send and receive chat messages. You will also find a readme.txt inside the project for more guidance.

Currently the project is limited and needs extending to make it a fuller chat experience.

Step 1

Users currently do not have usernames. When a user joins the server, they should also send their proposed username, they should send this data in the form of !username [new_username] e.g !username Bob. When the server receives this message, it needs to see if this name is in use. If no other user is using this name, then let the user know it was a success. If it is already in use, send a message to the user to tell them it failed. If the user gets a failed message, they should be disconnected on both client and server side.

Step 2

After adding usernames, chat messages broadcast from the server should include the username of the person who sent them in.

Step 3

Clientscan change their username whenever they want, using the !user [new_username] and if successful, the change should be announced to the rest of the clients.

Step 4

There are a number of special commands that need to be implemented. Currently if a user types in !commands to the server it will send a message back stating what commands can be done. These other commands have not been implemented yet and need to be done by you. The commands need to do the following:

!who – When server receives this message, it sends back messages containing the names of the connected users to the client to be output to the chat window

!about – Server should send information back to the client about its creator, purpose and year of development

!whisper [username] – sends a message to a specified user directly or lets sender know it cannot find anyone by that username

Also implement a custom command of your own devising

Step 5

The server can designate a moderator with !mod [username] e.g. !mod *Bob*. Moderators can kick other clients using !kick [username]. If the server uses the !mod command on a moderator, then they are demoted back to normal client. Server can also use !mods command to see a list of moderators output to the chat window.

Referencing

It is essential that you use appropriate APA style for citing and referencing research. Please see more information on referencing here https://library.torrens.edu.au/academicskills/apa/tool

Submission Instructions

Zip all projects, related files and any instructional readme.txt files for submission. Name this zip file using this format: NDS203 [your name]_[studentID]_Assignment2.zip

Submit this task via the **Assessment** link in the main navigation menu in NDS203 Networking and Database Systems. The Learning Facilitator will provide feedback via the Grade Centre in the LMS portal. Feedback can be viewed in My Grades.

Before you submit your assessment, please ensure you have read and understand the conditions outlined in the Academic Integrity Code Handbook. If you are unsure about anything in the Handbook, please reach out to your Learning Facilitator.

Academic Integrity Declaration

I declare that except where I have referenced, the work I am submitting for this assessment task is my own work. I have read and am aware of Torrens University Australia Academic Integrity Policy and Procedure viewable online at http://www.torrens.edu.au/policies-and-forms

I am aware that I need to keep a copy of all submitted material and their drafts, and I will do so accordingly.