CSC645 COMPUTER NETWORKS

Project: Socket Programming

1. Goal

The goal of this project is to write simple client and server programs for text messaging with socket programming.

2. Instruction

Write a server program and a client program. The server manages a number of client accounts, each having a username and a password. The server program starts first. When a client process is started, it **repeatedly** shows the following menu to the user and asks the user for an option:

0. Connect to the server
1. Get the user list
2. Send a message
3. Get my messages
4. Exit

If the user chooses Option 0, the client program establishes a TCP connection with the server. Then, it prompts the user for their username and password to log in. The client program then sends the credentials to the server to verify. If what the user entered matches the username and password stored at the server, grant the user access by printing "Access Granted" at the client. Otherwise, print "Access Denied – Username/Password Incorrect", and keep asking the user for the credentials until the user gets them correct to successfully log onto the server.

If the user chooses Option 1, the client program receives the list of usernames from the server and print them out to the user.

If the user chooses Option 2, the user can leave a text message to another user. The client program prompts the user for the recipient of the message, sends this text message to the server, and the server will store it for the recipient.

If the user chooses Option 3, the user can retrieve their own text messages that were left by other users at the server. The client program will display those messages to the user.

If the user chooses Option 4, the client program terminates the TCP connection with the server and exit.

3. Sample Code

We provide sample code for a simple server program and a client program. The client establishes a TCP connection with the server, and sends a message to the server, which will send a capitalized message back to the client for display. Essentially, it gives you a programming template that you can follow to write your own networking program.

4. Testing

Use the following scenario for testing. The server maintains two client accounts: One is for Alice and the password is "1234"; the other is for Bob and the password is "5678". Alice logs onto the server first. She gets the user list from the server and sees Bob. She sends a text message to Bob and exit from the client program. Next, Bob logs onto the server. He then retrieves his text messages left by Alice, and exits from the client program.

More details can be found from the demo video posted on Canvas.

5. Submission

Name the server program as *TextServer.java* and the client program as *TextClient.java*. Create one ZIP file consisting of these two source files and executables. Name the ZIP file as "*YourFirstName_YourLastName*" and submit it to Canvas.