Thomas Hotard

COMs 319

Prof Jannesari

Sept 23, 2019

Homework 1 Report

Solution Approach:

I found that most of the basis for this project can be found in the code for the first lab in ListServer.java and ListClient.java. The key difference between the code from the first lab and our project is that the code from the lab has inputs/outputs coded in and the clients do not interact.

Because of this I used the code from the first lab as the outline for my code. The first thing I changed was making the connecting clients must log in and create a username. I changed the ListClient code so that before users connected to the server, they would be prompted to give a username and put in the correct passcode.

Since the users would have to be able to communicate, I realized that I must have a way to store all the connected clients' info. I set up an ArrayList to accomplish this task. At first I created the ArrayList of the threads themselves so I would be able to get socket info. But after running into some issues with this is decided just to have the ArrayList hold the socket info to make the process slightly easier.

The most important change I made to the Lab 1 code was changing the constructor code for ListClientHandler. I decided that I had to pass in the ArrayList I created in my server code as an argument in the constructor so the code to send messages would be able to access the socket info of other clients and send the messages from the clients to each other. That way, the message distributor would be able to look for its own socket info in the ArrayList, ignore it, and send the message to every other client in a for loop.

I didn't change the client code all that much. I got rid of the prompt handling code as I didn't need any prompts and kept most of the rest the same. One thing I added to the client side however was formatting the messages so that it included the usernames that the client entered upon running the client program. That way the server would not have to have any client name info and wouldn't have to format anything, just send the already formatted message to the other clients.

Screenshot:

The below screenshot shows two clients in the consoles on the left and the server in the console on the right. The middle console also shows what happens when the passcode is entered incorrectly.