Ananth Kuchibhotla

For my internship coding challenge, I created a simple multithreaded Client-Server architecture, which allows Clients to send messages to the Server. The Server is hosted on localhost, port 4444. The port number can be changed in Constants.java.

Running The Server / Connecting Clients
To run the server, compile the Server.java file and run ‘java Server’. I suggest running this through a terminal, as I use outputs to STDIN/OUT to signify actions taking place. This instantiates a single server object and leaves it open for Client connections.

After running ‘java Server’, compile the Client.java file. Then run ‘java Client’ from another terminal. Each time this command is run, a new Client is instantiated and connects to the server. The server then creates a new Thread for each Client, modularizing all Server-Client interactions for a single Client to be within a distinct Thread.

After running the command above, Clients can send messages to the server. The server will receive these messages and display both the Client who sent the message and the message itself. On the Client end, the Server will respond ‘Received!’ to indicate that the Client’s message was successfully sent. To ensure successful transmission, I used Server Codes, which I put in Constants.java.

To disconnect, Clients can type ‘QUIT’ which kills the connection, kills the Client’s individual Thread, and kills the program itself. To kill the Server, simply kill the process like any other.

Multithreading and Handling Clients
The Server can handle a preset maximum number of Clients at the same time. This number is not hard-coded into my implementation, and you may vary it as you please. You can find this number in Constants.java. If a Client attempts to connect to the Server after the capacity has been reached, the Client is sent a rejection methods and the connection closes. However, if Clients disconnect from the Server, reducing the number of Clients below the capacity, new Clients can join the same server. This is due to the way I handle the number of active Threads on the main Server Thread.

I was going to implement a chat between all connected users, but did not have the time complete that feature. This was my first experience with Server architecture, so it was definitely a learning experience. Please let me know if you have any issues running the program or about my implementation. I have heavily commented my code to make it as readable as possible.

NOTE: For any strange compilation issues, please compile all of the following files before running: Client.java, ClientRunnable.java, Constants.java, Server.java.