CSCI 4311/5311

Socket Programming

Programming Assignment 1

Due Date: Feb 27, 2026, 11:59 PM

Goal of the assignment

In this assignment, we build a simple group chat application. The protocol between the client and the server is as follows.

- You can choose to use either TCP or UDP in your implementation. (TCP preferred)
- The server is first started on a known port.
- The client program is started (server IP and port are provided on the command line).
- The client connects to the server. The server asks the user for input. The user types the username in the following format “username = ComNet” message on the terminal.
- If the user doesn’t provide a username, the server doesn’t accept the user’s messages.
- After the user enters a username, the server broadcast to everyone “Server: Welcome username”.
- After that, the user can send messages (e.g., "Hi", "Bye", "How are you"). The user's input is sent to the server via the connected socket.
- The server reads the user's input from the client's socket. If the user has typed "Bye", the server must broadcast to everyone with "Goodbye username" e.g. “Server: Goodbye ComNet”.
- If a user enters “AllUsers”, the server needs to send all active users to that user.

Rules:

- The server must be able to monitor and handle messages from multiple clients simultaneously. Therefore, your server needs to be multithreaded.
- Do NOT implement a peer-to-peer application. E.g., clients talk directly to each other.
- The architecture here is server-client. All messages from the clients go through the server. The server distributes the messages to all other clients who have a connection to the server.
- You can implement the assignment with your preferred programming language. However, in case you have any questions, I may help you if you implement your code with Java. Other programming languages will not receive support.

- You need to write a report to explain your code, put some screenshots for your outputs, etc. Save your report in PDF format.
- Without the report, you don't get any points.
- GUI is optional with 20 bonus points. If you implement the GUI version well, you will get an additional 20 points. Example:

Step 1: Execute Server code

e.g. java Server 8989

This starts the server listening on the port number 8989

Step 2: Execute Client code

java Client localhost 8989

It tells the client to connect to the server at localhost on port 8989. Then you see the following message in the server's console:

The program asks for the username:

e.g. Enter your username:

Let say client 1 enters UNO.

Server prints "Welcome UNO"

Now, UNO can enter any message

Step 3: Execute Client code one more time to get a new user

The program asks for the username:

e.g. Enter your username:

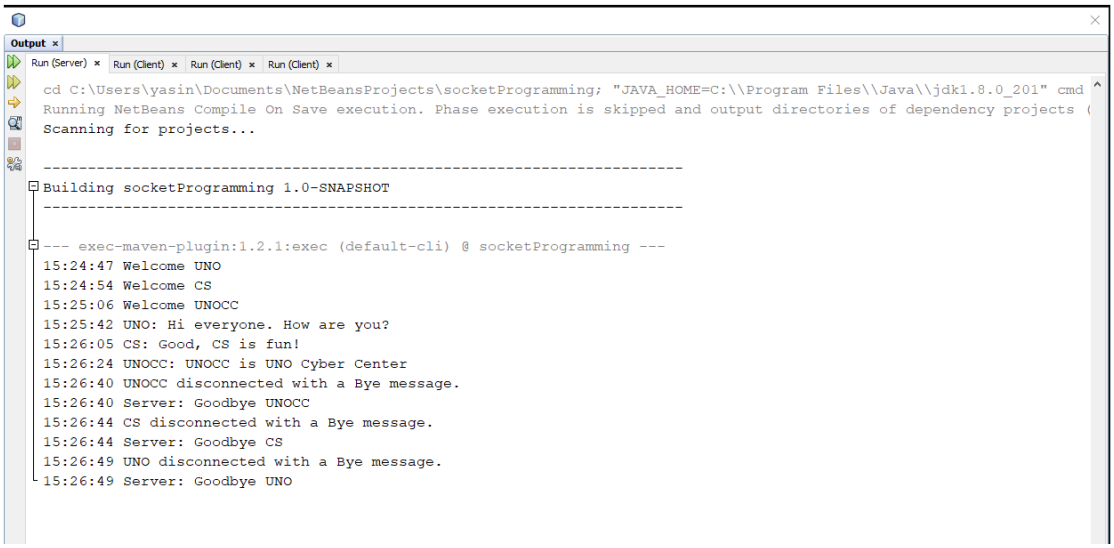
Let say client 2 enters CS.

Server prints "Welcome CS"

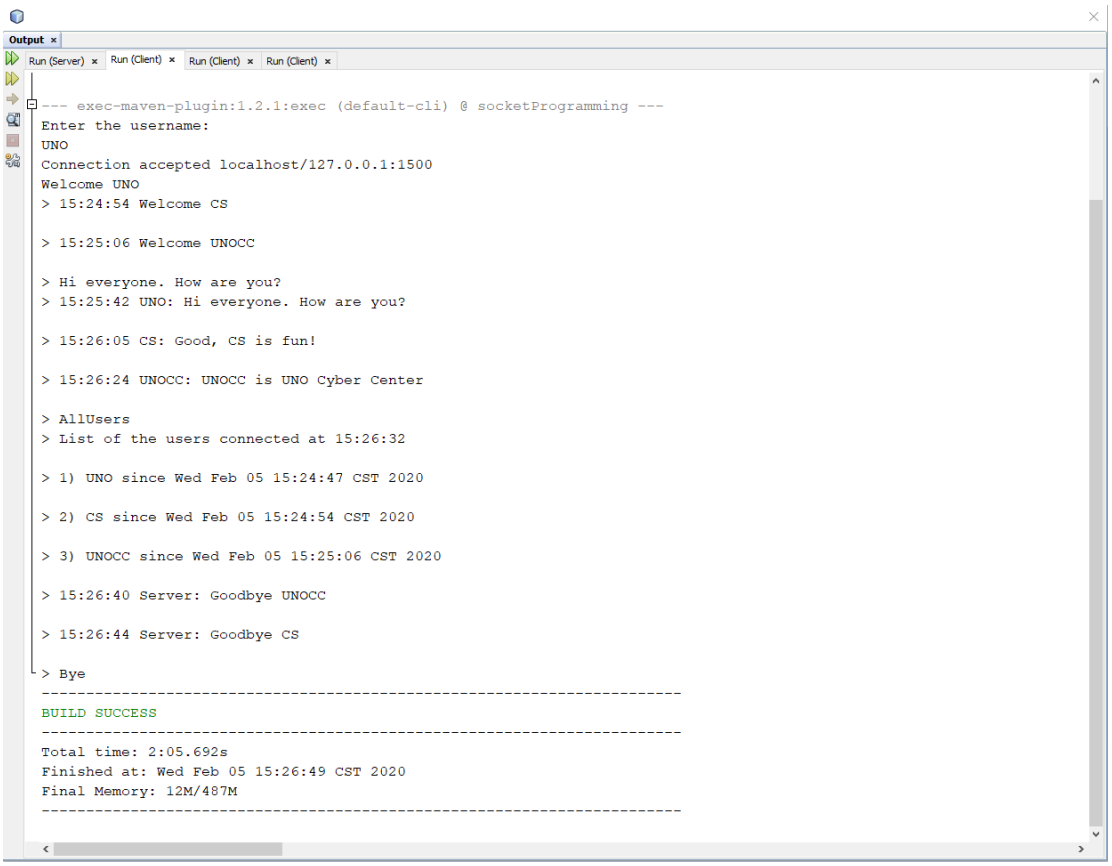
Now, CS can enter any message

EXAMPLE:

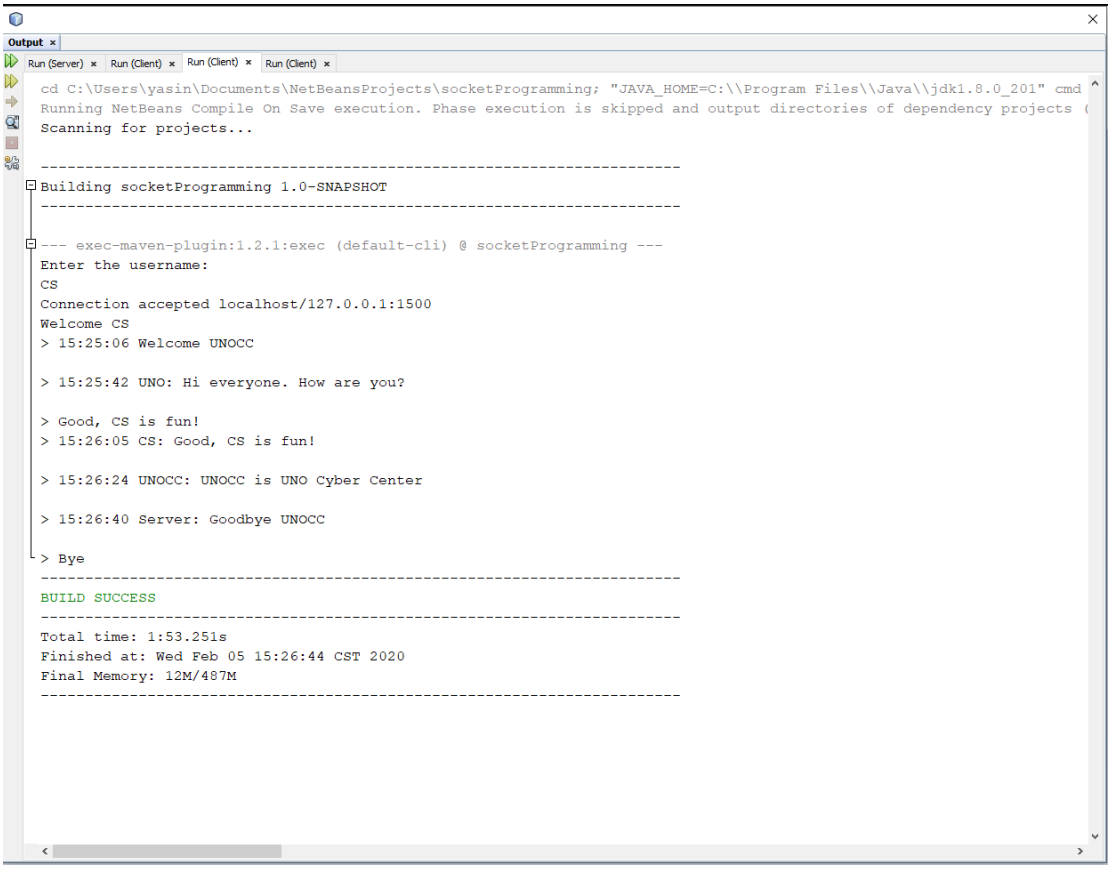
Server Output:



Client 1:



Client 2:



Client 3: