**Description of design/ implementation:**

A chat server and chat client was created as per the project specifications.

The server starts and using the Java Socket API, waits for any socket connections using port 13.

The application has been designed to allow multiple users/ clients to connect to the server, maps were used to store any connected clients.

Upon start up, the chat server will send the client a message to ask them to enter their username. This is to allow a distinguishment between different clients, even if they are using the same IP address.

Upon successful connection, the chat server will confirm through a message to the client.

Threads have been implemented to allow for multi-user chat sessions. Multiple clients can all connect to the server and send messages. The application has been set up where a message sent will not be received by the person who sent it, but everyone else in the chat server will.

It has also been set up to not allow the client to send messages to other clients on the server if there is no characters in the message.

When the clients send a message, it will include their username next to their message.

As per the project specification, if the client enters \q, they will disconnect from the chat server, followed by a confirmation message from the chat server “Disconnected from the chat server, goodbye"

IO Exceptions and Unknown Host Exceptions have been included for possible errors.

**Instructions for running the application:**

1. Start the chat server, do this by opening up a command line prompt and navigating to the ChatServer.java file location, once here run the command “java ChatServer.java”
2. Next, start the Chat Client. Similarly to the server, navigate to the java file and run it in the command prompt using a command “java ChatClient.java””
3. You will now be prompted for a username, and once provided, you will then be able to send messages to the server and receive them.
4. Repeat steps 2 and 3 for however many clients you would like.
5. Whenever a client sends a message, it will appear on every other clients console.
6. Press \q to exit as a client.

**References:**

# Bibliography

Anon., 2015. *Java Sockets TCP send and receive.* [Online]   
Available at: https://stackoverflow.com/questions/32098630/java-sockets-tcp-send-and-receive  
[Accessed 01 2025].

Schools, W., n.d. *Java User Input.* [Online]   
Available at: https://www.w3schools.com/java/java\_user\_input.asp  
[Accessed 01 2025].