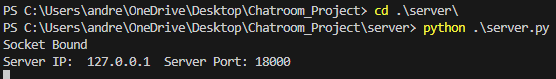
**CS 436 - Project: Network programming (Chatroom)**

**Group Members: Andrew Ortiz**

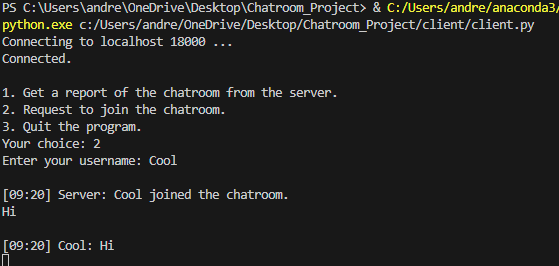
**Test your program as following:**

1. Run the server.py program. Use port 18000 at the server. Remember that your snapshots of the server should always show a brief report of what the server does at each step.

**Server:**



1. Run the client.py program. The client program displays the menu. The user selects option 2 and enters username “Cool”. Cool enters one message “Hi” to the chatroom.

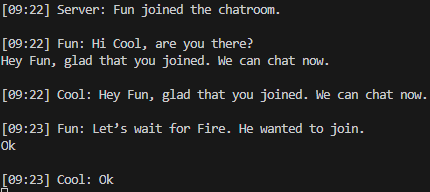
**Client:**

**Server:**

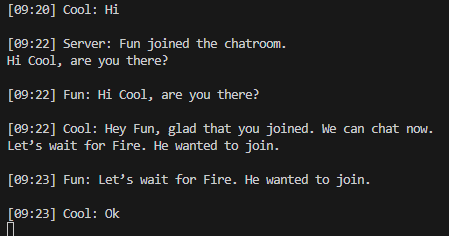
****

1. Run the client.py program again. The client program displays the menu. The user selects option 2 and enters username “Fun”. Fun will get a history of the chat and observes that Cool has entered “Hi” to the chatroom. Then, Fun enters “Hi Cool, are you there?” They both see this message in the chatroom. Then, Cool enters “Hey Fun, glad that you joined. We can chat now.” They both see this message in the chatroom. Then, Fun enters “Let’s wait for Fire. He wanted to join.” They both see this message in the chatroom. Then, Cool enters “Ok”.

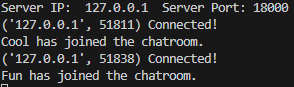
**Cool’s perspective:**

****

**Fun’s perspective:**

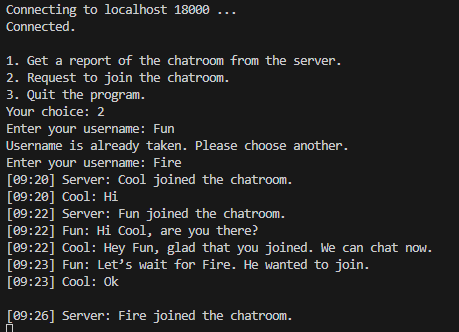
****

**Server:**

****

1. Run the client.py program again. The client program displays the menu. The user selects option 2 and enters username “Fun”. The server detects that Fun is being used by another user and rejects the request. The reject message will show up for the user. Then the client prints the menu again. The user selects option 2 and enters username “Fire”. Fire will get a history of the chat and observes the chats between Cool and Fun.

**Fire:**

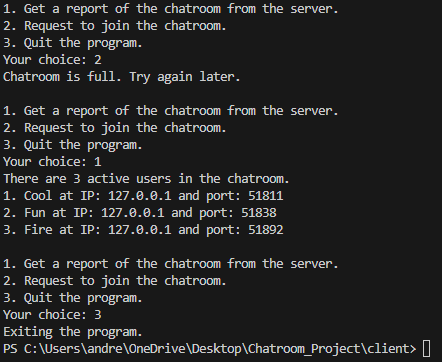
****

**Server:**

****

1. Run the client.py program again. The client program displays the menu. The user selects option 2 and enters username “Sad”. The server detects that 3 active users are in the chatroom and rejects the request. The reject message will show up for the user. Then the client prints the menu again. The user selects option 1. The server sends a report of the 3 active users, their usernames, IP addresses, and port numbers to this client. The report will show up for the user. Then the menu will show up. The user selects option 3 and the program exits.

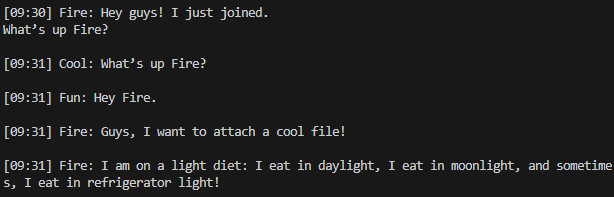
**Client:**

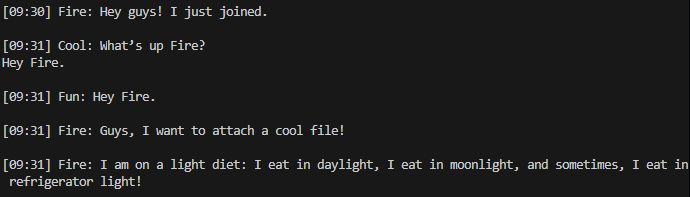
****

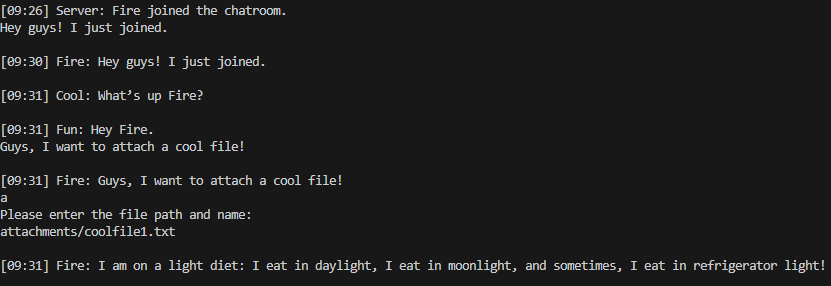
**Server:**

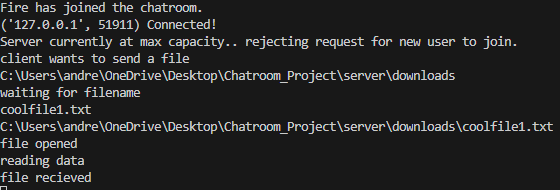
****

1. Then, Fire enters “Hey guys! I just joined.” Then Cool enters “What’s up Fire?”. Then, Fun enters “Hey Fire.” Then Fire enters “Guys, I want to attach a cool file!” Then Fire enters lowercase “a”. The client displays a message “Please enter the file path and name:” Fire enters “attachments/coolfile1.txt”. Then Fire will upload the contents of the file to the server. The server stores the file in its “downloads” folder and forwards it to all clients. All clients store the file in their “downloads” folder and display the contents of the file on the screen. Your snapshot should include the clients’ folders to show that they downloaded the file. Then all clients display the contents of the file in the chatroom. Extra credit: Instead of coolfile1.txt, upload coolfile1.png or any other image extension. Then display the image in the chatroom. Also, each message in the chatroom should include the user’s profile avatar.

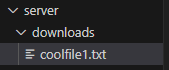
**Cool’s perspective:**

**Fun’s perspective:**

**Fire’s perspective:**

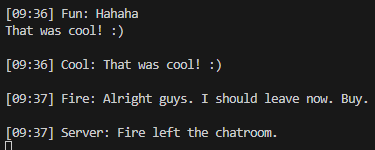
**Server:**

**Directory:**

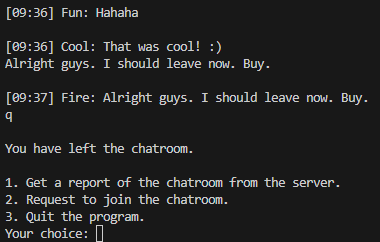
****

1. Then, Fun enters “Hahaha”. Then Cool enters “That was cool! :)” Then Fire enters “Alright guys. I should leave now. Buy.” Then Fire enters lowercase “q”. The client sends a request to quit message to the server. The server removes Fire from the active users and sends a message to everyone that “Fire left the chatroom.” Everyone will see this message. Then, the quitted Fire client will close its connection to the server.

**Cool’s perspective:**

****

**Fire’s perspective:**

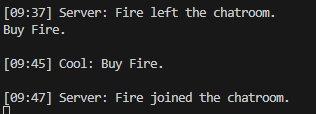
****

**Server:**

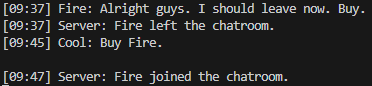
****

1. Then Cool enters “Buy Fire.” Fire does not see this message. Then, the quitted Fire client program displays the menu. The user selects option 2 and enters username “Fire”. The server accepts this request because at this moment Fire is not an active user in the chatroom. The new Fire will get a history of the chatroom, containing all messages including the last “Buy Fire” message from Cool and the messages from the server indicating that users joined and left the chatroom.

**Cool’s perspective:**

****

**Fire’s perspective:**



**Server:**

****