

FUNCTIONALITY

TCP connection between client and server

```
[pichons@hawk A1]> python chatserver3.py 8433
Server is listening on hawk:8433
Connected with ('130.179.28.119', 35084)
Client created (first_user)
handling client: ('130.179.28.119', 35084)
Connected with ('130.179.28.119', 46722)
Client created (second_user)
handling client: ('130.179.28.119', 46722)

[pichons@hawk A1]> python chatclient2.py hawk.cs.umanitoba.ca 8433
Enter your username: first_user
Connected to server. Use '/help' to view a list of commands.

[pichons@hawk A1]> python chatclient2.py hawk.cs.umanitoba.ca 8433
Enter your username: second_user
Connected to server. Use '/help' to view a list of commands.
```

Create chat rooms, view list of existing chat rooms, and view the number and list of connected users for each room

```
/create
Successfully created room #1
/list
===== ROOMS =====
ROOM 1 : 1/5
- first_user
=====

[pichons@hawk A1]> python chatclient2.py hawk.cs.umanitoba.ca 8433
Enter your username: second_user
Connected to server. Use '/help' to view a list of commands.
/list
===== ROOMS =====
ROOM 1 : 1/5
- first_user
=====
```

- Both clients are able to see the newly created room, as well as its capacity and a list of all users currently in it.

- When a room is created, the client that created it is automatically put into it.

Join existing chat rooms

```
/join 1
Successfully joined room #1
/list
===== ROOMS =====
ROOM 1 : 2/5
- first_user
- second_user
=====
```

- second_user has joined the created room, and is properly shown in the list

Send messages to chat rooms

```
second_user has joined the room.
second_user> hello
wow is that second_user my favourite user of all time?
first_user> wow is that second_user my favourite user of all time?
second_user> yeah that's me, im definitely not the same person typing to himself no siree

hello
second_user> hello
first_user> wow is that second_user my favourite user of all time?
yeah that's me, im definitely not the same person typing to himself no siree
second_user> yeah that's me, im definitely not the same person typing to himself no siree
```

- Both users can see the messages from other users sent in the same room

Leaving a chat room

```
/leave
Left room #1. Use '/list' to view existing rooms or '/help' for other commands.
/list
===== ROOMS =====
ROOM 1 : 1/5
- second_user
=====
i am gone

first_user has left the room. 1/5
hey where did you go
second_user> hey where did you go
```

- first_user has left the chat room. Their messages are not seen by second_user, and they cannot see second_user's messages from the room that they left.

Additional functionality:

- “/exit” to disconnect from the server

```
/exit
Disconnecting.
[pichons@hawk A1]>
===== CLIENTS =====
Client second_user is in room 1
=====
```

- “/help” to view a list of available commands

```
/help
Valid commands:
/help: Shows valid commands
/exit: Disconnect from server
/create: Create a new chat room
/list: List existing rooms
/join <#>: Join an existing room
/leave: Leave the current room, bringing you back to the lobby
```

TESTING

Testing was done manually on the aviary servers using multiple terminals. One server was running, and multiple clients would connect.

- 1) Connecting to the server with clients
 - a. Connecting with clients one by one was successful
 - b. Connecting with multiple clients without picking a username, then picking a username was also successful
- 2) Viewing list of existing chat rooms / connected users for each room
 - a. If no rooms exist, the client is told that no rooms exist and is prompted to create one.
 - b. After creating a room, the room appears in the list appropriately.
 - c. Leaving the room (and therefore deleting it) will now show the same message as (a)
 - d. Leaving a room that had another person in it will show that the room still exists, and that the other client is still there.
- 3) Joining existing chat rooms
 - a. Attempting to join a chat room without giving a number will give an appropriate error message to only that client and nothing bad will happen.
 - i. No number given: "Invalid use of join command"
 - ii. Argument is not a number: "Please enter a valid digit"
 - b. Attempting to join a chat room that doesn't exist will give an appropriate error message only to that client and nothing bad will happen.
 - c. Attempting to join a chat room that the client is already in will give an appropriate error only to that client and nothing bad will happen.
 - d. Attempting to join a chat room that is full will give an appropriate error only to that client and nothing bad will happen.
 - e. Attempting to join a chat room that exists and has room for more clients will successfully add the client to the room and broadcast a message to anyone in that room to show they have joined.
 - i. The client shows up properly when viewing a list of rooms/clients.
 - ii. If the client was in another chat room, they are successfully removed from it, and if that room is now empty it is deleted. Otherwise, a message is broadcasted to anyone in that room to show that a user has left.
- 4) Check room capacity and restrict access to full capacity rooms
 - a. To test this, the MAX_CAPACITY variable was lowered to 1 to make testing easier.
 - b. Checking the list of rooms correctly shows the current capacity as well as the max capacity.
 - c. If there is room, a client may join a chat room.

- d. If the room is already at max capacity, the client is told that the room is already full, and they do not join the room.
- 5) Creating chat rooms
- a. Creating a chat room correctly places the client into the chat room
 - b. Chat rooms cannot have the same room number as another chat room (they are unique)
 - c. Chat rooms are correctly assigned the lowest available number
 - d. Creating a new chat room makes a client leave the previous room they were in, and the leaving is handled properly (removed from that room's list, lets everyone in that room know someone has left, if the room is empty it gets deleted if it isn't the lobby).
- 6) Sending messages to chat rooms
- a. Messages are correctly not broadcasted if the client is in the lobby (room 0)
 - b. Clients do not receive messages from clients in a different room
 - c. Clients correctly receive messages from clients in the same room
 - d. If a message is received in the middle of inputting a message, it is printed over the input which looks odd, but otherwise has no negative effect. The message is still sent properly.
 - e. Clients correctly see other clients' usernames as well as the intended message being sent
- 7) Leaving a chat room
- a. The user is correctly not allowed to leave the lobby, and an error message is shown only to that client.
 - b. Clients can leave an existing chat room, and a message is broadcasted to anyone in that room that that specific client has left, and the client is shown a separate message to inform them that leaving was successful.
 - c. Leaving a chat room when the client was the last client in that room will correctly delete the now empty chat room.