Socket Programming: Group chat room mini project

Developed by:
6030226121 Thaworn Kangwansinghanat 28
6030609921 Suchut Sapsathien 28
6031045021 Bhoome Suitcharit 28
6031061021 Anan Methasate 28

Github repo: https://github.com/daidew/group_chat_web_socket

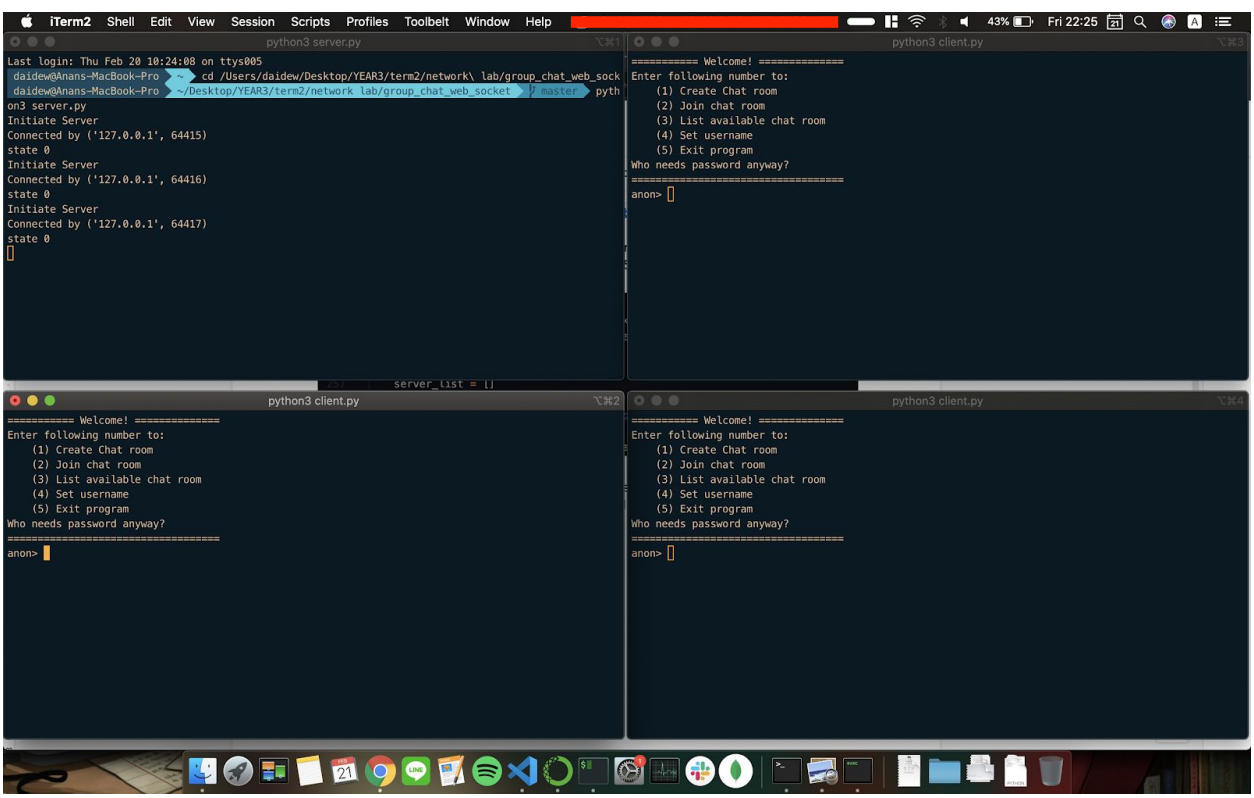
How to use it?

Before running, configure the number of clients you want your server to be able to handle by specifying in `s.listen(n)` and `for i in range(n)` in line 256 and 260 respectively. By default, the server will be able to accept a maximum of 5 clients.

```
252 if __name__ == '__main__':
253     s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
254     s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
255     s.bind(('', 50006))
256     s.listen(5)
257     server_list = []
258     # conn, addr = s.accept()
259     # server_list.append(server(conn, addr))
260     for i in range(5):
261         threading.Thread(target=server, args=(s.accept())).start()
```

Then, start the server by executing command `python3 server.py` (in project directory) for initiating server.

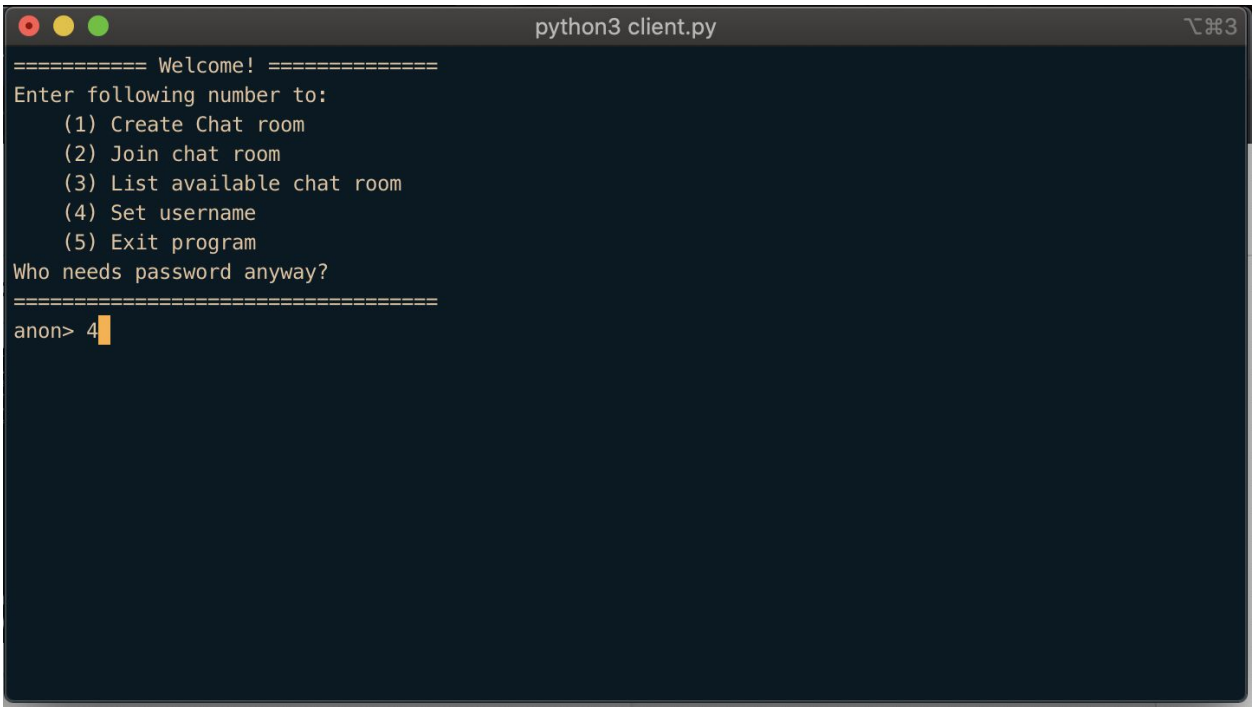
Start the client by executing `python3 client.py`. For this prototype, we have not implemented ngrok tunneling yet, so it can only be used locally (local network). As an example, I have created 3 clients and 1 server as shown below. (top left terminal is the host server)



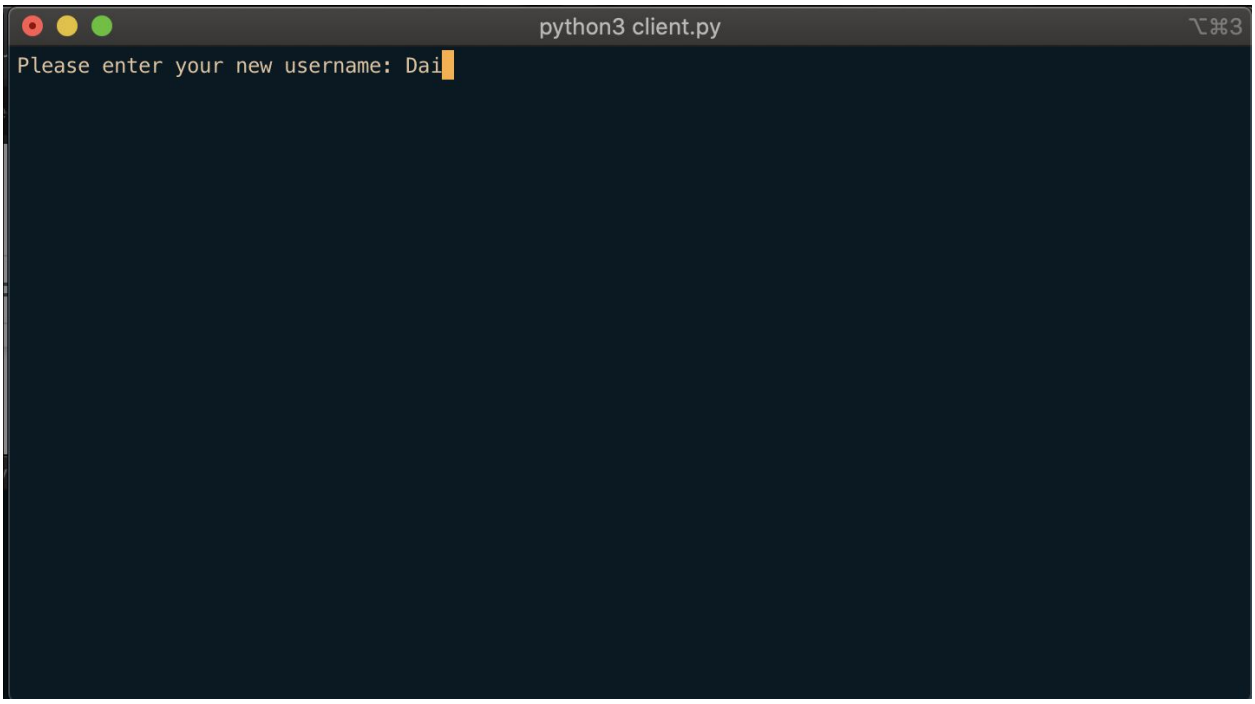
After starting the client program, we will see the main menu rendered through the terminal screen. For clients, there are 5 options available;

1. Create chat room
2. Join chat room
3. List available chat room
4. Set username (default is “anon”)
5. Exit program

To set username, type `4` then enter.



When prompted, enter your desired name and enter.



If the process went successfully, you will be able to see your handle changed from anon to your newly entered name as shown below.

```
python3 client.py
Update username successful
=========== Welcome! ==============
Enter following number to:
    (1) Create Chat room
    (2) Join chat room
    (3) List available chat room
    (4) Set username
    (5) Exit program
Who needs password anyway?
===================================
Dai>
```

Next, we will try to create a new chat room by typing `1` and enter.

```
python3 client.py
Enter room name you wish to create: 1150
```

If successful, the message will be returned as `Create room XXXX successful`.

```
python3 client.py
Create room 1150 successful
========== Welcome! ==============
Enter following number to:
    (1) Create Chat room
    (2) Join chat room
    (3) List available chat room
    (4) Set username
    (5) Exit program
Who needs password anyway?
==================================
Dai>
```

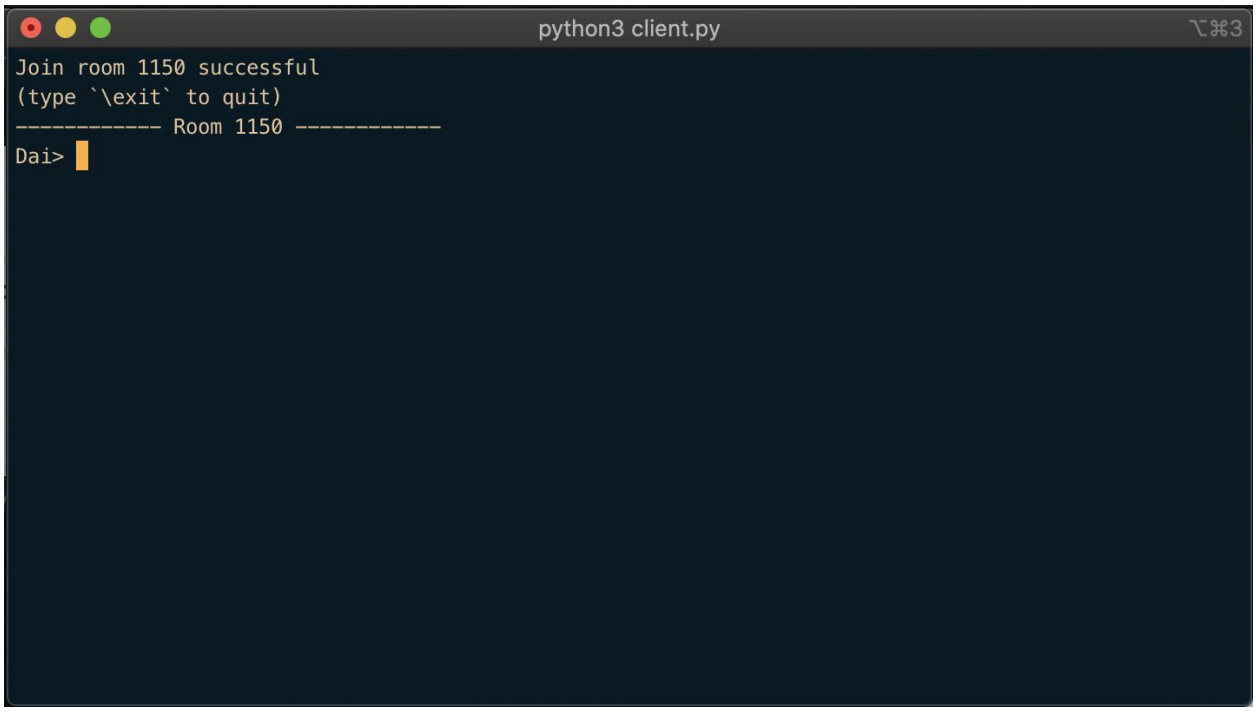
We can also list all created chat rooms by typing `3` and enter.

```
python3 client.py
Active Room(s): 1150
========== Welcome! ==============
Enter following number to:
    (1) Create Chat room
    (2) Join chat room
    (3) List available chat room
    (4) Set username
    (5) Exit program
Who needs password anyway?
==================================
Dai>
```

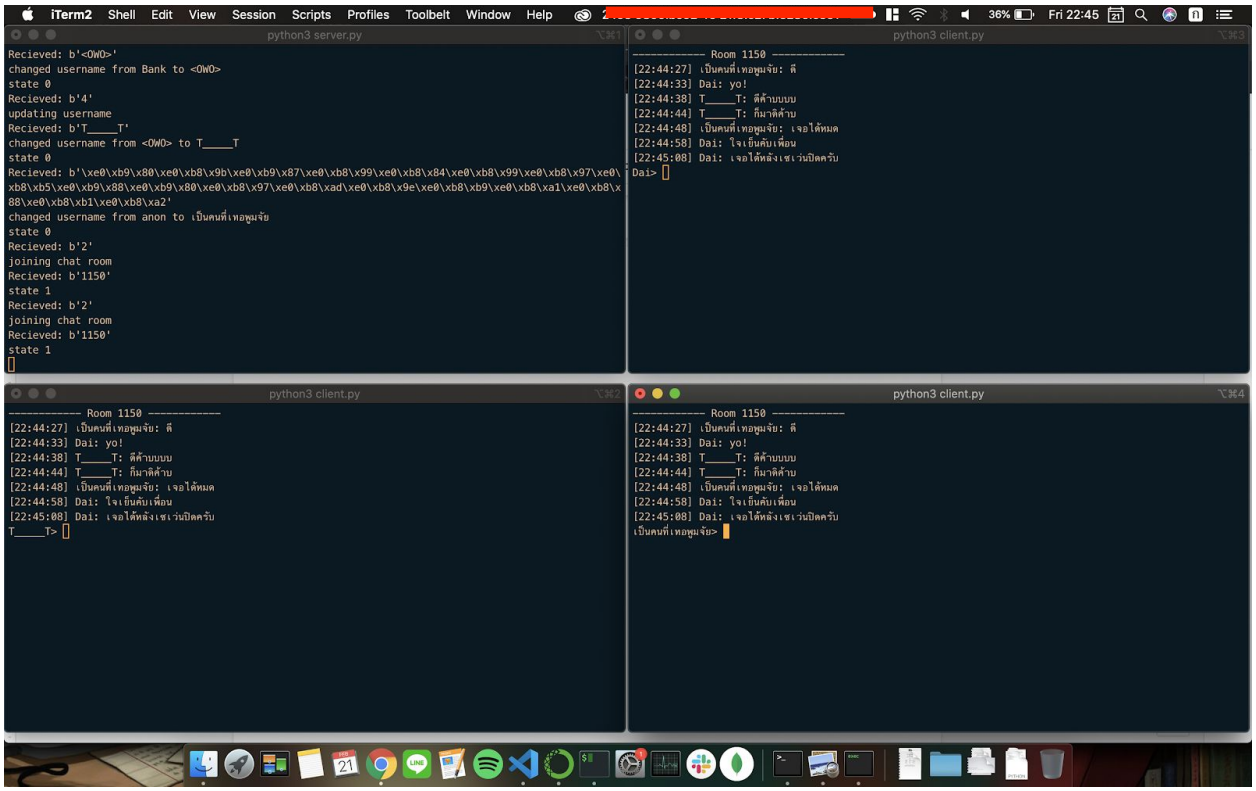
If we want to join a chat room, type `2` and enter.

Then enter a room number you wish to join.

If successful, the screen will now look something like below. You are now in a chat room. Hooray!



This is an example of a simulated chat scenario where 3 people are talking to each other. Each terminal screen is updated in real time. (to exit chat room, type `\exit`)



Algorithm

I have used a state machine to track the state of each client, where the session is maintained in different function threads in a server side. The client is almost like an input/output relay. The server does rendering and processing like a 1st tier architecture.
Code structure design is OOP based.

Problems Encountered

My major problem was to deal with input handling, where the client must receive the data from both stdin and socket at the same time, so some kind of polling technique was used to process input. The second problem was the screen is a terminal, so when clearing the screen (to update chat), the typed input will also get cleared as well. I have tried to alleviate this by capturing input without waiting for newline characters, but till this day I have not managed to be able to fix it.