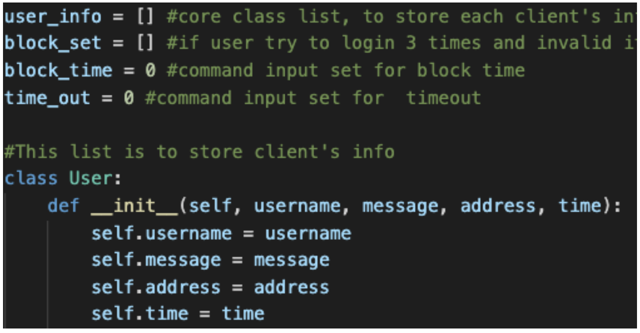
Zid:z5380108

Program design:

This program is basically design through two part: server.py and client.py
The main model for them run is that server will create multi-thread and one thread to connect one client, the client will have two threads for its own after the login process which means the program send and receive the socket data synchronously after login or it will stuck in recv if no message sent from server.

Data Structure:

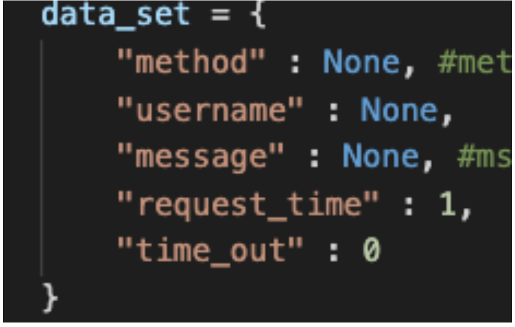
The main data structure used in server is a structure class list which is user_info list store each client's information for executing many functions. User_info will store the user's name and client socket and that work efficiency in send message to another user



```
user_info = [] #core class list, to store each client's in
block_set = [] #if user try to login 3 times and invalid i
block_time = 0 #command input set for block time
time_out = 0 #command input set for  timeout

#This list is to store client's info
class User:
    def __init__(self, username, message, address, time):
        self.username = username
        self.message = message
        self.address = address
        self.time = time
```

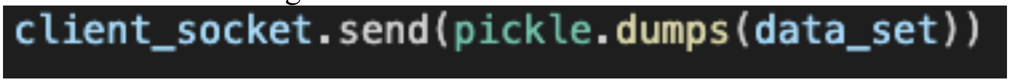
Besides I pass the message through client and server used by dictionary, because it works as little protocol to make the format sent and format receive. In dictionary data_set, the "method" is to identify the function the client wants to do, and 'message' is to store the content server or client want to. Like send some specific msg to other and it will store in message.



```
data_set = {
    "method" : None, #met
    "username" : None,
    "message" : None, #ms
    "request_time" : 1,
    "time_out" : 0
}
```

The application layer message format

The file sent between server and client is packed by pickle, it can pack the dict into byte format which is easy to access the info in dictionary from client and server. But p2p did not facilitate. It packs the dict with dumps and unpack by load and I use send all in my program to send the specific client. In usual there will be client_socket which is the socket by sent client, sometimes like message to others, it will be user.address.send to send another socket to finish the interacting



```
client_socket.send(pickle.dumps(data_set))
```

System work:

System will work like sending and receiving through client and server, one server will connect to multi-client through multi-thread but one thread only works for one client.

My program often run in **python3**, please make sure it, and I often test it in mac, it won’t be fine in Vlab.

My program hasn’t implemented

1, Peer to peer Messaging
2, user blocking and unblocking
3, whoelsesince
4, offline messaging

Reference

```
self.clientAlive = True
```

There are some basic connection and var like clientAlive borrowed from sample given by Wei Song