COMP3331/9331 ASSIGNMENT REPORT

1. Project specification:

Project: Instant messaging application
Language: Python 3.7

2.Run the program

Start client: python3 client.py IP port (command line)
Stop client: logout or using Keyboard Interrupt (ctrl + c)
Start server: python3 server.py port block_time timeout (command line)
Stop server: Keyboard Interrupt (ctrl + c)
It is better to rerun the program after the connection has closed(waiting for the tineout time) , or it may cause the error 'OSError: [Error 98] Address already in use'.

3. Application layer protocol:

TCP is the protocol for communication between client and server. And server can handle the requests from multiple clients which also allows the clients to communicate with other clients and start private connection with other clients

4.Design of my program

●Client side:

Firstly, I create an initial socket which aims to receive the current port number
Then, I recreate one socket with the previous port number and setup connection with the server side. One of the purpose of this sockets is to distinguish with other clients, and the other is to implementing the sending and receiving function.
In order to implementing these two functions, I use the multithread method for executing the receive and send function at the same time.' t1.setDaemon(True)' is for the child thread can be closed as the main thread is closed.
Then using the while function to continuously receive and send message from or to the server.
If 'welcome' in the receive message, meaning that the connection has been successfully established.
If 'time out' in the receive message, the connection will be closed and the client program will exit.
If 'log out' in the receive message, program will raise an error for the signal of exit.if the connection of clients has already time out, it will show one sentence in client program(Your connection is timeout,please tap Enter button),tapping the ENTER button can exit the program.

●Server side:

I will divide my server side program into several sections and talk about it briefly. And I use a large dictionary to store the information of one user, including the password,status,ban_time,active_time,socket and log_in_time etc, , which the format is like the picture below.

The first part is to setup connection with each clients, which I implement in 'each_client' function.

After the connection has established, server side should receive the input from the client side to have the user authentication(waiting the input from client side) and check the correctness of username and password, the server will provide at most three times login authentication to the clients. And after three time wrong log in attempt, the server will block this particular user during the block time duration

Broadcast function is to reform other clients the online or offline of one user. When user log in, server will immediately send a message to all the online users.

Online message: If the receiver is online, the server will find it's socket and forward the message to this receiver

Offline message: the any message which was sent to other offline user will be stored in the offline message dictionary. And whenever the user is online, the server will send the message to this user.

Whoelse is to find who is online expect yourself(the sender). The method of my design is to check the status in that time to find who is online and send it to the sender.

Whoelsesince time is to list all the online users since this time except the command sender, which I implement it by checking the time and log in time of the users and this given value.

Blacklisting: if one client block hans, then he will not be received any information from hans. And he can also use unblock command to remove hans in the block list.

The method I use for timeout is to start a new thread to refresh value in the dictionary

and check the active time per second for each clients. If one clients do nothing more than time out time, it will be forced to log out, and send a return message to client side, the server will log this user out. And each functionality ablove is refresh the active time of the users. And whenever the connection has closed, there will a message showing in the server command line.

5.main modules

Socket , thread ,pickles ,sys

6.improve of my design

I think I can use class object in the future because it will reduce some codes
And I can also make a progress in peer to peer communication.