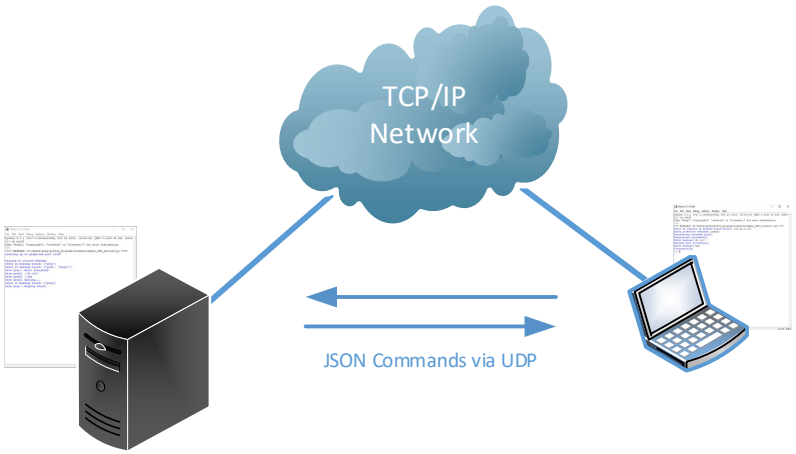
1.0 Project Description

The final project for the course is a message board. A message board allows a client application program to post messages on the server and displays that message. System setup for the message board is shown in the figure below.



The system uses UDP for communication while the application protocol follows a JSON formatted commands for posting messages, registering and deregistering users. The following are the JSON commands:

Client command codes to server

Client Commands to Server	Description
`{"command":"register", "username":"user01"}`	Message board registration
`{"command":"deregister", "username":"user01"}`	Message board deregistration
`{"command":"msg", "username":"user02", "message":"This is my message."}`	Post message on server

Server return code to client

Server Return Codes	Description
`{"command":"ret_code", "code_no":201}`	Command parameters incomplete
`{"command":"ret_code", "code_no":301}`	Command unknown
`{"command":"ret_code", "code_no":401}`	Command accepted
`{"command":"ret_code", "code_no":501}`	User not registered
`{"command":"ret_code", "code_no":502}`	User account exists

Each group will have to create the server and the client program. A group can partner with another group to test the programs developed. Applications developed by the group should adhere to the protocol above so that it is interoperable with other group's work.

2.0 Example Communication Exchange

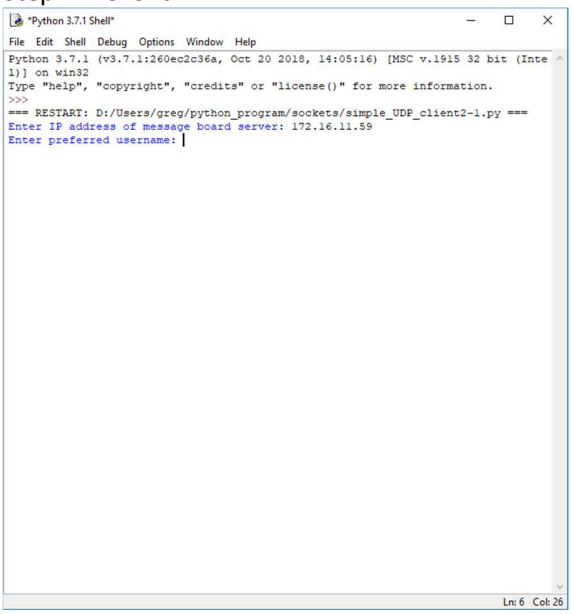
The following are example communication exchanges between server and client.

2.1 User Registration

Users are expected to register prior to posting messages. Following figures shows the step by step exchanges between server and client.

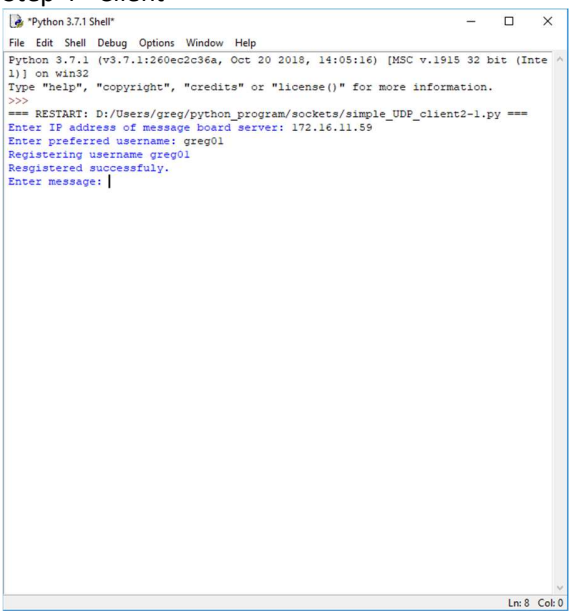
Step 1 - Server

Step 2 – Client



Step 3 - Server

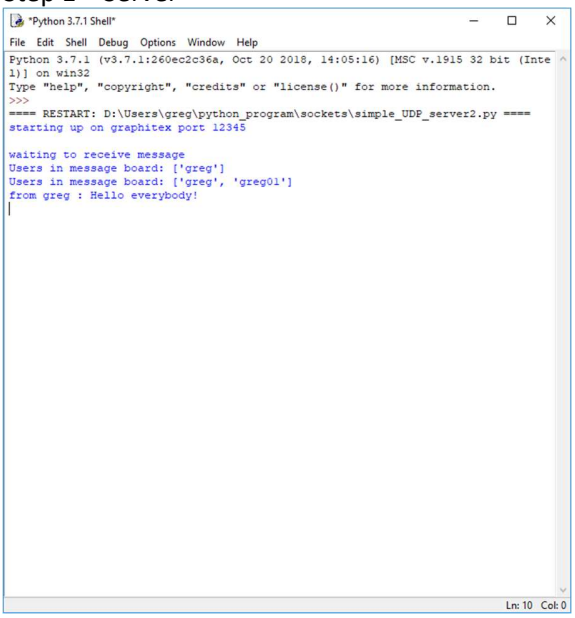
Step 4 - Client



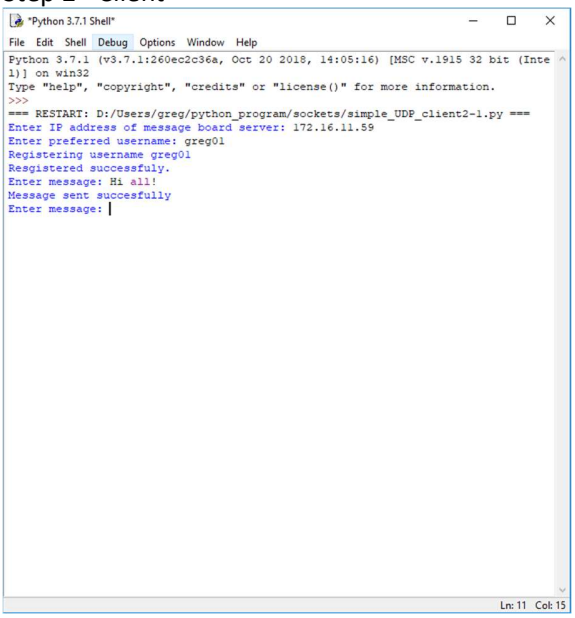
2.2 Posting Messages

All messages posted by clients should be seen on the server. Messages from other clients does not need to be seen by the clients. Following figures shows the step by step exchanges between server and client.

Step 1 – Server



Step 2 - Client



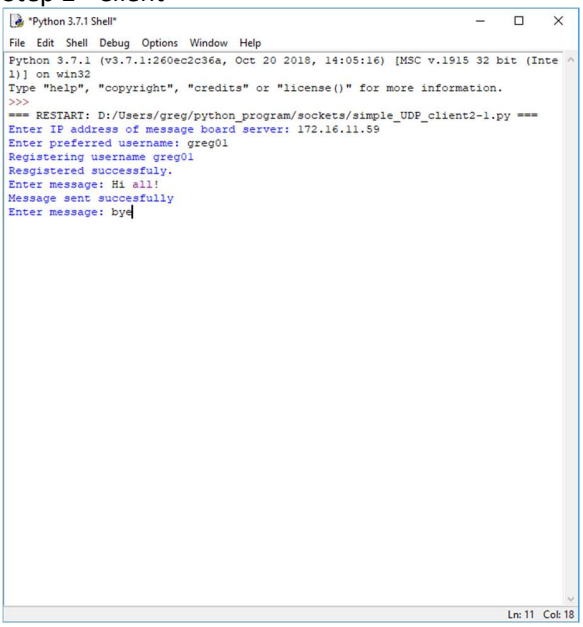
Step 3 - Server

2.3 User Deregistration

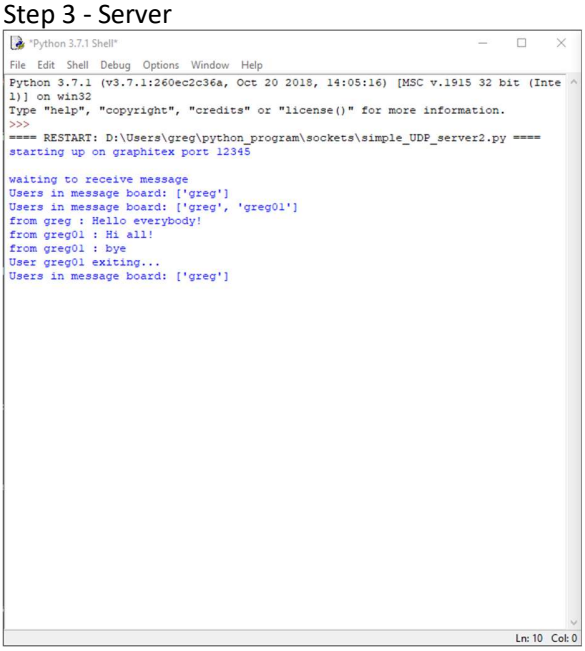
When the client is terminating, it should de-register itself from the server. The server should also show the new list of users in the server. Following figures shows the step by step exchanges between server and client.

Step 1 - Server

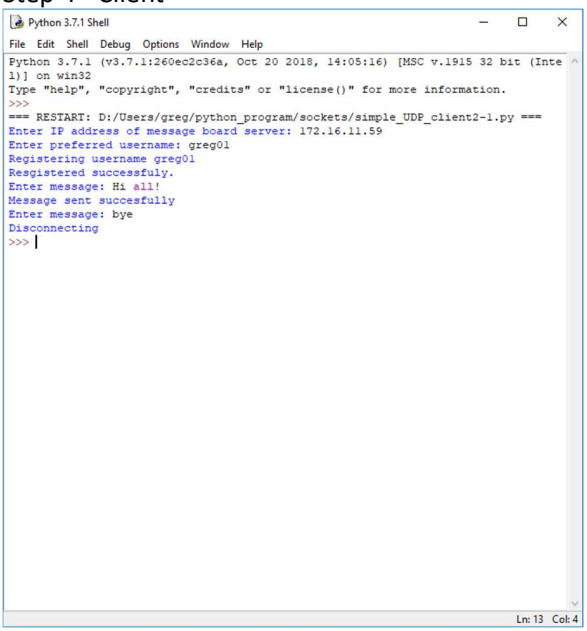
Step 2 - Client



Step 3 - Server



Step 4 - Client



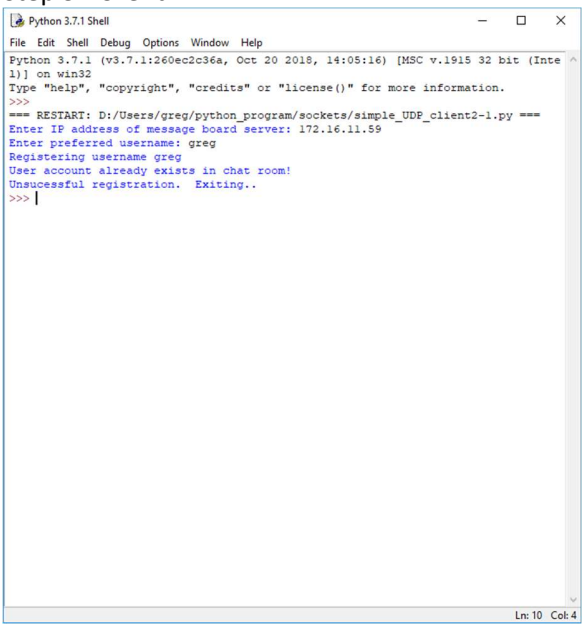
2.4 User Registration Error – Existing User

It is expected that all users should have unique “handle” in the message board. When users are registering with a non-unique handle, the server should notify the client program. Following figures shows the step by step exchanges between server and client.

Step 1 - Server

Step 2 - Client

Step 3 - Client



3.0 Project Requirements

The following are the requirements for the project:

1. Each group will create the server and client program
2. Each group must have a partner group for checking program / communication compatibility
3. The server message board and the client application should adhere to the protocol
4. Server and client applications should be interoperable with other groups
5. Server program can be put on CSNET01 or CSNET02 server.
6. Use the port number assigned from Hands-On 4.
7. Groups may use any programming language in developing the server or client application
8. Groups are required to submit codes in Canvas

Deadline for the project is posted on the Canvas assignment. Groups will need to sign up for demo appointment.