**Name: Manohar Rajaram**

**Student ID: 1001544414**

# Note

* Language used: **Python**
* Python version: **3.6.7**
* IDE used: **PyCharm**

# Project Description

Implementation of multithreaded server-client communication using HTTP

# Execution Steps

Step 1

* Run the file ‘server.py’.
* If running from command window, change directory to the folder containing the server file. Execute the following command from command window/terminal or via PyCharm:

**python server.py**

* If running from PyCharm, run the server.py file.
* A server window pops up.

Step 2

* Run the file ‘client.py’.
* If running from command window, change directory to the folder containing the client file. Execute the following command from command window/terminal or via PyCharm:

**python client.py**

* If running from PyCharm, open the client.py in a new terminal and execute the above command.
* A client window pops up
* Repeat the above steps two more times for two more client windows.
* We can follow the prompts on the client from here on or follow the below steps
* You may need to move the windows to adjust visibility

Step 3

* On the client window, enter the name to identify the client with.
* Message displayed on server window that the client has joined. On the top-right side of the window, active client list is displayed.
* Enter the name in the rest of the 2 client windows similarly.

Step 4

* From any client window, enter any message to be transmitted. A prompt comes asking for delivery method (1-1, 1-N). Choose accordingly by entering 1 or 2.
* If 1 is selected, the previously entered message Next a prompt comes asking for the destination client by displaying the active client list. The name of the desired client should be entered.
* We can see the message displayed on the destination client along with delivery method.
* Also, we can see the HTTP request on the server window of the transmitted message.

Step 5

* Repeat the steps from the other 2 client windows.

Step 6

* Disconnect one of the clients by clicking on ‘Quit’. The server window displays message about the disconnection.

Step 7

* Disconnect the server by clicking on the ‘Disconnect’ button on the server window. The clients are notified about the disconnection and will no longer be able to send messages.

# Limitations

* Clients with same name is not handled currently
* Messages with ‘=’ or ‘&&’ symbol in them is not transmitted properly because I am using these as delimiter in the HTTP request parsing.

# References

* [Chat App- Start Up](https://medium.com/swlh/lets-write-a-chat-app-in-python-f6783a9ac170)
* [Date representation](https://stackoverflow.com/questions/225086/rfc-1123-date-representation-in-python)
* [HTTP Formats](https://www.jmarshall.com/easy/http/)