COL 334 : Assignment - 2

Sidharth Agarwal - 2019CS50661

September 16, 2021

Key Design Decisions:

I have programmed this assignment in Python with assistance from socket and threading libraries of python

- ***Two sockets:*** So for each client I have used two sockets, one for sending the messages and waiting for the acknowledgement of the message from server and second for receiving any forwarded message from the server and sending back its acknowledgement to the server.
- ***Strict registration:*** If any client tries to enter the chatroom without following the strict registration protocol then it is in not registered and receives ERROR 101 from the server. The other clients remain undisturbed and are still able to share messages. In my program I have followed the protocol of first registering the receiver socket followed by registering the sender socket. If the client has followed the protocol but tries to register a malformed username then it receives ERROR 100, and is asked to register again with proper username.
- ***Inconsistent typed message:*** So the clients should always write messages of the format *@receiver text* where text is non empty. If it fails to follow this, it is asked again to type the message in correct format. For broadcasting the message the client should use *@all*. If the client tries to send the message to a unregistered username it receives ERROR 102 from the server.
- ***Header Incomplete:*** If the client tries to send a message to a valid receiver which is even slightly deviating from the standard protocol of sending the message then it receives an ERROR 103 and is deregistered from the chatroom. The rest of the clients remain undisturbed and can still keep on sending the messages and adding more people in the chatroom following the appropriate protocol. In my program I have tried to check whether content length is present or not, whether it is having only integers or not, whether the length of the text is actually equal to the content length or not, the positioning of the new line characters and the positioning of the "SEND" and "Content-length" characters
- ***Broadcasting:*** I finally implemented sequential way of broadcasting, sending a message to one client and waiting for acknowledgement and doing this for all clients. Because I felt that this way of broadcasting made more sense when we are using a two sockets for each user and where we wait for the acknowledgement for unicast messages in the same thread as that of sending thread.
- ***Error on server and receiver side:*** So if in case sender sends a correctly structured message to a valid user and its header also passes all the test of protocol on the server. But somehow the receiver does not receive a valid message, then I notify the sender with a ERROR 102(Unable to send), and donot close any connection since the server and sender connection is still working.