**Chat Room Client & Server Tan Karageldi**

**Comp 4911 – Networks**

This document describes a Chat Room Client & Server as a final project for this course. The project was motivated by my interest on chat rooms, especially growing up playing video games with my friends. Starting with TeamSpeak, Skype, Discord, my interest on advanced chat rooms, and importance of real time communication between people especially playing video games, motivated me to pick this topic for my project.

**Introduction**

Chat rooms are online systems, that allow multiple users to communicate over a network, with real time communication. Chat rooms are hosted on a server, enabling users, that have the client software, around the world to communicate with each other. This project aims to develop a robust chat room server and client system to facilitate communication. By using client-server architecture, chat room will have a centralized server, which is the host of the program, will provide data for multiple clients that are using the chat room client. Chat rooms can be used for various topics of communication and can be used for different purposes. It’s mainly used for sharing text-based information among a group of users simultaneously. Unlike instant messaging tools, which are primarily intended for one-on-one conversations, chat rooms allow users to engage in discussions with multiple participants within the same conversation. There are many advanced chat rooms that are currently used by millions, which has extended usages other than just communication. (Skype, Discord, Microsoft Teams.)

**Socket**

First, we need to know about Sockets. A network socket is a structure of a computer network, that serves as an interface, for sending and receiving data across the network. Socket, in networking terminologies, serves as a connector, the application layer to the transport layer in the TCP/IP protocol. Sockets are used for establishing communication between a server and client/s. Socket programming involves establishing communication between two nodes on a network. One node act as a listener, listening on a specific port at an IP address, while the other node initiates the connection. The node acting as the listener creates a socket to accept incoming connections, while the other node connects to it.

**Python**

This project, I will be using python language, for server development, which has a library on socket programming. (References, No 3). Python is a great programming language for this project, because of the clear and easy to understand syntax, which makes the development faster and efficient. It allows the developers to focus more on the logic rather than getting stuck down in the language complexity. Another reason to use python, is it comes with a rich library that includes handling sockets, http requests, and other networking tasks, which is what are we looking for in this project. Also, if we need to use, there are more advanced networking libraries, that makes it easier to develop a server-client model.

**Server**

Chat room server is the host of our system. The server will run a specific environment, which will stay open and running as long as we want the users to communicate and use the chat room. Server accepts connections from the clients, retrieve messages from the clients and distribute that message to other online clients which are currently in the chat room. The server will establish a socket, which has specified IP address and a port number assigned by the server user. The server should stay open and running to receive connection requests. Server needs to use multi-threading to handle with multiple user connections. Multi-threading, as its defined below, is handling multiple thread executions from simultaneous messages coming from different clients.

**Client**

The chat room server is practically useless without clients. Client program is like the user interface, which our user is going to be using to establish connection and send messages to other users throughout the server. Chat room client will try to connect to the server socket, which has an assigned IP address and port number client should know. Client program will check if there is any input coming from either sides, client, and server, after connection. And will do respective actions, such as sending the message to the server for server to process and display the message for the other users online or displaying the message from server on client’s terminal.

**Multi-Threading**

A thread refers to a sequence of code the computer must execute. In computer architecture, multithreading is the ability of a central processing unit (CPU) to provide multiple threads of execution concurrently, supported by the operating system. (Ref. 4, lines 1-4). In this context of our online chat room, we will use multi-threading to handle concurrent connections and communication between clients and the server. By using this approach, we will ensure that server will handle multiple client connections simultaneously without blocking or enabling real-time connection.

**Responsibilities**

***Server:***

* Accepting Connections
* Managing Client Connections
* Processing Messages
* Printing Messages
* Keeping the State of the Chat Room

***Client:***

* Establishing Connection
* Sending Messages
* Receiving Messages
* Displaying Active Users
* Establish Disconnection

**Server Script**

Starting with server-side script. For this python script, we need to import both of the libraries I have mentioned above.

**Conclusion**

Included with the final submission of this project, there will be a server-side script, which will implement the ideas I have talked about above, to offer secure, successful, and real time connection with client users. Also, the submission will include client-side script, which will also implement the ideas above, to connect to the respective server using the IP address client is connecting to, send and receive messages, and print the messages on command line. There are advanced extensions that are in consideration to incorporate in this architecture. (Private messaging, secure connection using usernames and passwords.)

***References:***

1. *What is a chat room?*

[*https://www.techtarget.com/whatis/definition/chat-room#:~:text=A%20chat%20room%20is%20an,hold%20conversations%20about%20various%20topics*](https://www.techtarget.com/whatis/definition/chat-room#:~:text=A%20chat%20room%20is%20an,hold%20conversations%20about%20various%20topics)*.*

[*https://en.wikipedia.org/wiki/Chat\_room*](https://en.wikipedia.org/wiki/Chat_room)

1. *Python socket library.*

[*https://docs.python.org/3/library/socket.html*](https://docs.python.org/3/library/socket.html)

[*https://docs.python.org/3/library/socketserver.html*](https://docs.python.org/3/library/socketserver.html)

1. *Multi-Threading*

[*https://en.wikipedia.org/wiki/Multithreading\_(computer\_architecture)*](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture))