

# COMPENG 4DN4 Lab 4 Report

Aaron Pinto  
pintoa9  
400190637

Raeed Hassan  
hassam41  
400188200

April 9, 2023

## Server Class

The server class code is run from the terminal and will run a Chat Room Directory Server (CRDS) whose function is to maintain a chat room directory (CRD) for the clients. The server listens for client connections on its Chat Room Directory port (CRDP), which is port 50001, and accepts CRDS commands from clients that involve listing, creating and deleting chat rooms.

On startup, the client will create a TCP listening socket on port 50001. This socket will handle client connections and handle client commands. The server is also able to handle multiple client connections at the same time. This is achieved by using the `select` module, where multiple client connections are added to a list of read sockets, which are then monitored by the `select` module. When something is available on one of these sockets to read, the server will process the received commands from the client without interrupting any other client connection.

When a command is read from a client, the connection handler will attempt to receive bytes over the TCP connection. If no bytes are received from the connection, then we close the connection. Otherwise, we receive a command input from the client that we can use to perform CRD operations. The server will receive one byte from the client that will contain the command issued by the client. Depending on whether a `getdir`, `makeroom`, `deleteroom`, or `chat` command is received, the server will handle the operation requested by the client differently.

When a `getdir` command is received, the server will send eight bytes to the client which encode the response size, and then will send the entire listing response which contains the chat room names, chat room addresses, and chat room ports.

When a `makeroom` command is received, the server will receive one byte encoding the size of the chat room name requested, followed by the chat room name, followed by the same for the chat room address and port (one byte for size, followed by contents for each address and port). The new chat room will be added to a dictionary on the server that contains all the available chat rooms.

When a `deleteroom` command is received, the server will receive one byte encoding the size of the chat room name requested, followed by the chat room name. The server will search for the chat room in its dictionary, and if found it will remove the chat room from the dictionary.

When a `chat` command is received, the server will receive one byte encoding the size of the chat room name requested, followed by the chat room name. The server will search for the chat room in its dictionary, and if found it will send one byte encoding the chat room address, followed by the address, then one byte encoding the chat room port. If the chat room is not found, it will return a 0 encoded to a single byte.

The server will close the connection on the socket when an error occurs, when no response is received from the client, or when the client has closed the connection on their end (`bye` command is issued by the client).

## Client Class

The client class code is run from the terminal and will be access the Chat Room Directory Server (CRDS) and to chat with other clients using IP multicast communications.

The client will allow the user to connect to the CRDS on the CRDP port, and establish a TCP connection using the **connect** command. This will establish a connection with the CRDS and allow the user to input client-to-CRDS commands. When the client has established a connection with the CRDS, it will have access to additional commands, **getdir**, **makeroom**, and **deleteroom**. The three commands will send a packet with the format of the packet determined by the command.

When a **getdir** command is issued, the client will send a single byte integer designated for the **getdir** command, and receive eight bytes from the server which encode the response size, and then will receive the number of bytes required for the listing response.

When a **makeroom** command is issued, the client will send a single byte integer designated for the **makeroom** command, one byte encoding the size of the chat room name requested, followed by the chat room name, followed by the same for the chat room address and port (one byte for size, followed by contents for each address and port). The client will be able to confirm the chat room has been created with the **getdir** command.

When a **deleteroom** command is issued, the client will send a single byte integer designated for the **deleteroom** command, followed by one byte that encodes the chat room name size, followed by the chat room name. The client will be able to confirm the chat room has been deleted with the **getdir** command.

The client will close the connection on the socket when an error occurs, or when no response is received from the server. The client will also close the connection when the **bye** command is issued.

Outside of the client-to-CRDS communications, the client can allow issue **chat** and **name** commands. The **name** command simply changes the displayed name of the user when chatting, stored as a string on the client. The **chat** command will create a TCP connection the CRDS, request a specific chat room (similar to the **deleteroom** command) and receive the address and port (similar to the **getdir** command) to connect to, and then immediately close the TCP connection. The client will then transmit multicast broadcasts on the chat room's associated address and port.

## Member Contributions

Aaron contributed to the server class, Raed contributed to the client class.