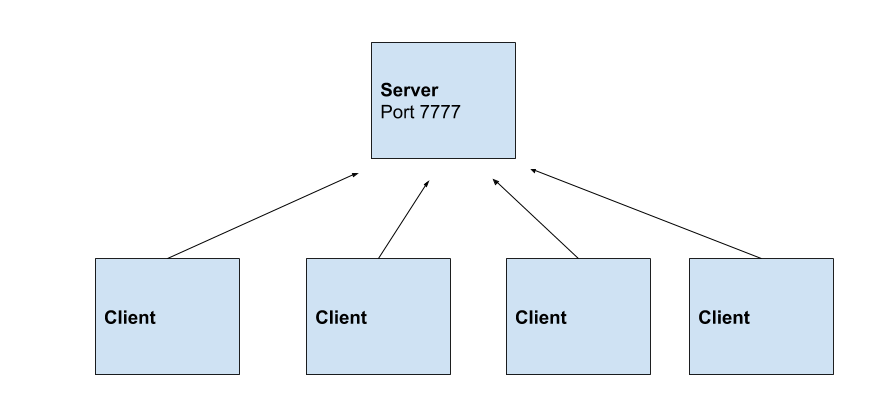
NDC  
Assignment 1  
**Group Chat Application with Integrated Commands and Custom Protocol  
  
Date Given:** 8th Feb **Date Due:** 22nd March  
**Marks:** 30%

**Assignment Outline**

In this CA, you are required to build a chat client/server application. The server is designed to listen for connections from clients, and then open a socket to each client. Whenever a client is connected, all messages that are sent to the server should be relayed out to each of the individual clients. Below is a diagram of client/server model we will be working with.



Although we are working with multiple clients, this can all be replicated on your local machine.

The server is responsible for keeping track of the different users that are currently connected to the server and to relay out different messages to the clients.

A protocol should be designed, that will allow the sever to interpret the different commands that are being sent to the server. Similar to HTML, a simple markup using tags e.g., **<cmd>message</cmd>** can be used. When the server receives theses commands, it will then parse the tags and extract the relevant information and perform the desired task.

**System Specifics**

* As an element of the protocol, a hash of the payload (the data inside of the tag) should also be added, so the server can verify if all the content has been delivered successfully without any tampering. This requires an encryption algorithm to hash the data on the client side and add it into the payload. On the server side, the payload data should be hashed and compared with the one it received to verify it.
* A custom “ping” command should be created on the server, which allows the client to ping the server and send out a message to the client saying “pong” and the amount of time the response took.
* When a user is sending a message to the server, their chat username should also be added into the message, allowing the server to know who exactly sent the message.
* A feature should exist is in the server allowing the user to send a command asking the server for the current time, which will then be sent back to the client.
* Whenever a new user has joined the chat session, a message should be relayed to all clients saying “XXXX has just joined the chat session”.
* A “total” command should be created, that will tell a client how many messages are currently in the buffer.
* A feature should be added to get a list of all users currently in the chat.
* A method should exist to kick a user off the chat by issuing a “kick username” command. This should disconnect them from the chat.

**A-Grade**

You are encouraged to add some of your own functionality to the system if you are looking to get an A grade. An example of this is to review existing functionality of the IRC protocol and see what you can include in your application.

**Tips**

One of the core advantages of using Python is the low-level string manipulation that is available. Functions such as finding the index of a specific word inside of a string can help the server identify where a command starts and where a command ends. The text in the middle is the actual payload that is being delivered.

Python offers different data structures to hold information such as lists, these can be used to keep track of the different users and addresses that are currently connected to the server.

Before beginning the coding process, write out a list of commands and how the server should respond to each command, outline the process in a diagram to show the flow of the information between the client and the server.

**Deliverables**

* Source code for the application with comments
* Document (2000 words max) outlining the planning and functionality of the application and how the protocol you designed works.
* In this document please also outline who is in your group (Student Number & Names)
* Uploader link will be provided on Moodle. Only one member needs to upload the zip file.

**Mark Allocation (Out of 100%)**

50% - **Design**

* Quality and thoroughness of custom protocol
* Design and functionality of custom protocol
* Documentation and diagrams outlining protocol

50% - **Implementation**

* Implementation of client and server applications.
* Good use of functions and low-level string manipulation
* Use of threading in the client and server applications
* Quality of implementation of each function designed in system specifics (above)