1 Introduction

In this assignment you will implement virtual group chats. The purpose of this assignment is to improve your programming skills and deeply expose the transport layer and the application layer (layers 4-5). The assignment applies the *client-server* architecture as seen at class. Note that the *client-server* architecture has two entities: **client** who consume services, and **server** who provides services. In our case:

- The **client** is a user whose interested at group chatting.
- The **server** is the computer whose responsible of managing the group chats. Its services include: access to group chats, print new messages, etc. .

2 Requirements

At the beginning of any connection, the client receives an opening message with the following options:

- Connect to a group chat.
- Create a group chat.
- Exit the server.

Following are the requirements for each option:

1. If the client chooses option 1, the server needs to ask for the client name, group ID and password. In case that the group ID does exist in the server and the password matches it, the server needs to connect the client to the relevant group chat.
2. If the client chooses option 2, the server needs to ask the client for its name and password, and then it generates group ID and connects the client to a new group chat (with the respective group ID and password). Note that the client should be notified what group ID was generated.
3. If the client chooses option 3, the server needs to disconnect the client immediately. Note that the server keeps listening after the disconnection of the client.

When a client is connected to a group chat, the client should get any message that was sent **in the specific relevant group chat**, including the sender name and the message output to the console.

3 Example

∗ Client connecting to the server ∗
Server:
Hello client, please choose an option:

1. Connect to a group chat.
2. Create a group chat.
3. Exit the server.

Client:
1
Server:
Enter your name:
Client:
Bar
Server:
Enter group ID:
Client:
8888
Server:
Enter password:
Client:
1234
∗ Client connecting to a group chat ∗
Server:
You're connected to group chat #8888
Client:
Hi friends
Server:
client12: Hello, how are you?
client20: Welcome, new client.

4 Technical Highlights

- Make sure to check extreme cases. Use your own judgement to decide where a validation is necessary. Input type validation is not required – for example, if you expect to get a string you can assume it.
- You are responsible at establishing a reliable connection between the server and clients. Use the functions in *Socket* library wisely.
- You are **NOT** required to output in a **specific** pattern. However, you should put your efforts at the following:
  - Provide a well-documented quality code.

  - Provide a documentation in the PDF file.
  - Fully understand the theoretical material.

- Implementing a nice GUI (Graphical User Interface) will give you an extra credit of 10 points.

5 Instructions

5.1 Implementation

- You can implement your code in any operating system and any code language you wish. *Python* is recommended, as seen at class. Following is a link where you can learn python easily: https://www.w3schools.com/python

- You **must not** use any non-standard libraries for socket programming. Use only the libraries introduced at class.

5.2 Documentation

- You're required to well-document your code.

- You're required to attach a document in PDF format including the following:

  1. Full name and ID.
  2. Operating system, code language and version.
  3. Theoretical background.
  4. An explanation about any file you created – including any classes and functions.
  5. An explanation about the *socket* handshake – which protocol you used, what are the commands, are they *blocking* commands.

- Include illustration pictures (usage examples) for any option in the server.

5.3 Submission

- Submit in pairs or individuals.

- Compress all the files in a ZIP format and submit it with the IDs of the students.

- Final submission date: 01.01.2023
  Note that you have plenty of time. Use it wisely.