Project Charter

- **Start date:** 15/10/2019
- **End date:** 29/10/2019
- **Project Manger** Lorenzo DeCarli

Purpose

Everyday people use messages to communicate with other people or share information like photos, videos and documents.
All these opportunities are made possible thanks to networks, computer science and protocols.
Students must implement a chat protocol to learn how it works, what could be possible problems and how to solve them.
It gives students the possibility to write a program that emulates a real chat, based on a protocol and is an opportunity to understand all the aspects of everyday things.

Goals and Objectives

The overall goal is to create a sort of system where authorized students can communicate with each other in private or in groups.
The program is expected to:

- Provide a textual or graphical interface to communicate.
- Send and control packets that respect the protocol.
- Send and receive messages, view their content and provide general information about the chat, for example who is logged in.

Schedule Information

Our schedule is planned, as we already have a deadline to our project. The project has to last 3 lessons of 3 hours each, for a total of 9 hours, at the end of which each student has to have either a complete client or a server implementing the Chat Protocol.

- *15/10/2019:* Project Explanation
- *29/10/2019:* Project Submission

Finantial Information

The project has no budget, and no employee is getting paid.

Project Priorities and degrees of freedom

In the execution of the program, the elasticity that was given to us was a couple of weeks. When the three weeks were up we had to submit of the project in question. This is purely an educational project, so the work is paid in school hours without any budget to spend on the execution of this project. It is an immediate and simple software and maintenance is easily done through Python programming.

Approach

The approach to program execution started with building a client and a server via a socket opening a connection between them. The professor has provided us with his own protocol and we have created our functions in Python from a set of rules divided by steps to implement the chat protocol. Once the program was over, the professor resolved each of us' doubts about some problems encountered during the implementation.

Constraints

The final solution must not depend on a specific programming language (Python and Java are recommended). Both on the server and client side it is advised to create a graphical interface for better user interaction with the final product.

Assumptions

Customers through a graphical interface (client) can access a chat with other users through the server. To connect to the server, the customer must provide a user and a password.

Success Criteria

The project will be considered a success if a developer delivers a working client or server program.

Scope

The students are divided between who will develop the client and who will develop the server.

- **Client:** the program must provide the ability to
    - Send and receive packets and view related messages. Messages can be private, public or multicast.
- Register, log in and log out.

- Request a list of connected users.
- **Server:** the program must:
    - Check the correctness of the packets that arrive, and return one with the result of the verification.
    - Forward messages to its recipients.
- Manage the users who want to register, log in and log out.

Risks and Obstacles to Success

A risk we could face is the lack of experience and skill some students could have programming an entire application, whether it be a client or a server. The lack of experience with threads could pose a challenge, as not many students were able to successfully develop a multithreading application in two years.