

# Chat Project

---

## Authors

Quintarelli Federico, Gjergjaj Albert, Castellano Pietro, Donisan Ion Cristian, Scardoni Francesco

## 1 Introduction

### 1.1 Overview

This document defines the requirements for the Chat Protocol Project (CPP) which has been developed since 8 october until 13 december 2019 as a school project, which purpose, in addition to the chat itself, was to increase our development and team-collaboration skills. This document is designed to be beneficial to both the user/teacher as well as the development team. From this document the user/teacher will be able to determine our understanding of the requirements and verify their accuracy. The CPP should be a program which permits you to chat with other users after connecting to the main server, which controls the data streams between users as well as other various functionalities. The system must be easy to use

---

### 1.2 Goals and Objectives

CPP Goals:

1. Develop the program so that it can be maintainable and updatable easily in case of problems or future requests
  2. Include a quick login process to the chat
  3. Provide a Graphic User Interface (GUI) to ease the use of the program that allows users to enter and access the chat in an interactive environment
  4. The program must be able to establish a chat with an user group or with a private user meeting rules and boundaries imposed by the protocol
- 

### 1.3 Scope

In the end CPP will provide the user the ability to establish a connection with a specified chat room or with a private user interacting with the GUI.

---

### 1.4 Definitions

CPP - Chat Protocol Project GUI - Graphic User Interface

---

### 1.5 Assumptions

It is assumed that the CPP is be able to contact the central server and that the central server is online.

---

## 2 General Design Constraints

## 2.1 Product Environment

The CPP software can be started on any computer with an internet connection as long as the assumptions made at 1.5 are respected.

---

## 2.2 User Characteristics

There is just one category for users that may utilize the program. Normal user – Everyone that will use the chat protocol can access all the features it provides.

---

## 2.3 Mandated Constraints

There are no constraints at this time.

---

## 2.4 Potential System Evolution

The CPP software will most likely be updated in the following months when the teacher will assign us other tasks.

---

# 3 Nonfunctional Requirements

There are no nonfunctional requirements at this time.

## 3.1 Operational Requirements

There are no operational requirements at this time.

---

## 3.2 Performance Requirements

The amount of user that can connect is based on server availability, therefore the program must be able to receive, process and distinguish user messages and server control messages

---

## 3.3 Security Requirements

There are no security requirements at this time, they may be required later.

---

## 3.4 Safety Requirements

There are no safety requirements at this time.

---

## 3.5 Legal Requirements

No legal requirements at this time.

---

## 3.6 Other Quality Attributes

There are specific sections above for non-functional quality attributes such as security, performance, etc. In this section describe any other non-functional quality attributes such as portability, availability, etc.

---

### 3.7 Documentation and Training

The CPP documentation will consist in a github repository which contains all the documentation files. No training is needed.

---

## 3.8 External Interface

### 3.8.1 User Interface

Normal user Interface

**The interface is divided in two parts:**

- Login interface
- Chat interface

#### Login interface

The interface should be minimal and must allow the user to enter:

- An alias, the name that is wanted to be used during the communication (an error may occur if the name has been already taken or if it doesn't match the restrictions imposed by the protocol)
- The name of the chat room(the main chat room is named "general")

#### Chat interface

The interface should be functional and must allow the user to chat easily, it provides:

- A text field where the user can insert the message that has to be sent
  - A user list which shows online users with which is possible to establish a private chat
  - A button that permits to log off from the server
  - A field where the messages and senders are showned to the user
- 

## 4 System Features

There are no nonfunctional requirements at this time.

### 4.1 Feature: Group Chat

The group chat is where a users of the CPP can chat in a group of users of his choice. This section describes how.

#### 4.1.1 Description and Priority

While logging to the server (Login interface) the user can insert a room name, if it already exists the user will join the chat room but if it doesn't it will be created by the server. This feature is one of the main priorities of the project as well as user to user chat.

---

#### 4.1.2 **Use-case: New User**

Actors: the person using the program

Description: The use-case begins when the user starts the program and the login interface appears.

Basic Path:

1. The user will be prompted to enter alias and room name( in case of errors the program will handle the problem)
2. An acknowledgement will be sent back from the server including an ID with which the user is now able to chat providing the server to recognize him.
3. The chat interface will open and now the user can exchange messages with other users in the chat room
4. At a certain time the user can disconnect or, if any error occurred, it may be disconnected by the server

Alternate Path:

1. If step 1 is unsuccessful the user will see an error message on screen indicating what information still needs to be provided or if the user name has already been taken or if it doesn't match the boundaries
2. The user now returns to step 1 of the "Basic Path"

---

#### 4.1.3 **Additional Requirements**

Allow the user to change alias/username while chatting

---

### 4.2 Feature: user to user chat

The user to user chat permits to establish a connection with a single user at a time

#### 4.2.1 **Description and Priority**

The user must follow the same passages indicated in 4.1.1. This is also a main priority feature.

---

#### 4.2.2 **Use Case: Youth Login Screen Access**

Actors: the person using the program

Description: The use-case begins when the user starts the program and the login interface appears.

Basic Path:

1. The user will be prompted to enter alias and room name( in case of errors the program will handle the problem)
2. An acknowledgement will be sent back from the server including an ID with which the user is now able to chat providing the server to recognize him.
3. The chat interface will open and now the user can exchange messages with other users in the chat room

4. On the right side of the chat interface window the online user list will be shown and with a double click on the username we can start a private chat with someone
5. At a certain time the user can disconnect or, if any error occurred, it may be disconnected by the server

Alternate Path:

1. If step 1 is unsuccessful the user will see an error message on screen indicating what information still needs to be provided or if the user name has already been taken or if it doesn't match the boundaries
2. The user now returns to step 1 of the "Basic Path"
3. The user may be disconnected by the server for 3 reasons: the server crashed, the user has been AFK for 15 minutes or the server has been shutted down

---

#### 4.2.3 **Additional Requirement**

Same as 4.1.3

---

### 4.3 Feature: Error management

#### 4.3.1 **Description and Priority**

The errors will be handled by the program, when an error occurs the server will sent a code to the user so that the program can recognize it.