Table of Contents

1 - Introduction

1.1 - Overview The *Project: Chat Protocol* will be a pc application available for Windows and Linux alike. The application will provide the user with the ability to create an account and connect to a chat server to communicate with other users using simple text messages. They will also be able to check which users are online and to choose whether to write to a single user, multiple users or all active users at that specific moment

This document provides information on the requirements for the *Project: Chat Protocol* application. Project goals, scope and definitions are given in the introduction. Design constraints and application environment are described in the following section. Non-functional requirements are outlined for later verification. Functional requirements are given to show the system features and expected user interaction. Project constraints will be included in separate documentation. The Software Project Management Plan will give specifics on project budget and schedule. A separate Test Plan document will address test specifications and procedures.

1.2 - Goals and Objectives This project's main objective is to test the development team's ability to define internet protocols. The app's main use will be to allow classes of the Marconi institute to communicate between each other and with teachers

The *Project: Chat Protocol* application is expected to: 1. Provide a visual interface on pc to be able to see all online users and manage chats with them 2. Be functional but simple to use and understand 3. Provide users with the tools to chat with anyone who uses the same application and server

1.3 - Scope The *Project: Chat Protocol* application will provide users with the ability to create an account and connect to a server in which they will be able to chat among other users from a pc connected to the same network of said users. During the registration process, users will be able to input their profile info and set a password to ensure protection. Moreover, after logging into their account they will be able to view a list of all users they can interact with. Currently none of the functions of the app will be accessible without creating and logging into a valid account.

1.4 - Definitions ***Project: Chat Protocol***: The application that is being described; the software system specified in this document; the final product.

Project: The activities that will result in the development of the application

User: The person who will interact with the Application

Use case: Describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

Scenario: One path through a user case

Actor: User or other software system that receives value from a user case.

2 - General Design Constraints

2.1 - Project: Chat Protocol Application environment The Project: Chat Protocol product will include a pc app designed to work on a Windows or Linux platform. This application will interface with a TCP server of our design. The server will allow more efficient maintenance of the software system.

2.2 - User Characteristics Chat Users: students, faculty or staff who have access to a computer connected to the Marconi network. Since Marconi is an IT School, most students and teachers are expected to know how to operate simple pc applications. Moreover, usage of login pages and chats is very common in this age.

2.3 - Mandated Constraints Since the application is written and based in python, there are next to no constraints except those set by the teacher in the assignment, such as packet length and naming conventions.

3 - Nonfunctional Constraints

3.1 - Operational Requirements *Usability* : 90% of users will not need to read the user manual to be able to use the application.

3.2 - Performance Requirements *Maintainability* : Changes made to the client can be adopted without altering the server application. Changes to the protocol itself, such as packet length, adding a new type of packet or deleting an unused one, will lead to changes in the code of the client, more specifically to the protocol definition.

3.3 - Security Requirements The Project: Chat Protocol app's features can be utilized only by users after a secure login in a page initialized as the first view of the app. Passwords can be changed at any time and periodical changes of all passwords are not mandatory, but suggested.

3.4 - Documentation and Training The Project: Chat Protocol application will be delivered to users as a download without documentation or training except for a readme file containing basic information on the app, its stakeholders and creators. System documentation will be provided to project stakeholders as well.

3.5 - External Interface

3.5.1 - User Interface The user interface will be eye-catching but more functional than visually appealing. When users access their accounts, the interface will move to the chat portion of the app modeled with Python Tkinter, which has a simple and easily understandable look and feel.

The interface will be intuitive. No training will be provided and it is expected that 90% of users will be able to use the app without any training.

3.5.2 - Software Interface The Project: Chat Protocol server will serve as an interface between the clients. It will enable interaction between the users and give the single users information about the state of other users in the network.

4 - Functional Requirements

4.1 - Required features

4.1.1 - Use Case: 1 Description: User Login / Sending a message

Actors: Any student, teacher or school personnel

Basic Path

```
1. User opens the Project: Chat Protocol application
2. The app opens on a login page
3. The user inserts his username and his password in the forms. If he does not
 have an account yet, he can click on register to create one
4. The user inserts correct information,the app opens a main page where the
user can send messages and check recieved ones.
5. User selects the user/s he wants to send a message to and types the
text, then presses the send button
```

```
6. User continues to chat until he sees fit
7. User clicks Logoff
8. System exits and reverts to the login page
```

Alternate path 1. User opens the Project: Chat Protocol application 2. The app opens on a login page 3. The user inserts his username and his password in the forms. 4. The user inserts wrong information, the app displays an error message. 5. The app prompts the user to either retry to insert his information or to exit 6. If this option is selected, the system exits to desktop

4.1.2 - Use Case: 2

Description: Check all online users

Actors: Any student, teacher or school personnel

Basic Path

```
1. User opens the Project: Chat Protocol application
2. Following Login [Use Case 1 Step 3:4]: System opens main page.
3. User clicks the Check Users button
4. The app returns a list of all users currently logged in that the user
can interact with
5. The app returns to its main page, prompting either to log off or to send
a message to Users
6. User clicks Logoff
7. System exits and reverts to the login page
```

4.2 Optional Features 4.2.1 Use Case: 3

Description: Check all users, online or offline

Actors: Any student, teacher or school personnel

Basic Path

```
1. Following Login [Use Case 1 Step 6]: System displays main page.
2. User clicks Check Users.
3. System displays all online users.
4. User checks a "Display offline users" box below Check Users.
5. System displays in a different color all users that the server
has interacted with, online or offline.
6. After pressing a back button, the app returns to the main page
7. User clicks Logoff
8. System exits and reverts to the login page
```