# SOFTWARE REQUIREMENT SPECIFICATIONS

|  |
| --- |
| **Project Name:** Custom Messaging Application  **Document Title:** SRS  **Project Timeline:** 03.01.2023 to 11.01.2023 |

**TABLE OF CONTENTS**

# 1. Introduction

1.1 Overview

1.2 Purpose

1.3 Project scope

# 2. Overall Description

2.1 Project Features

2.2 User Needs

2.3 Operating Environment

# 3. System Features

3.1 Functional Requirements

3.2 Technical Requirements

3.3 System Features

1. **Non-Functional Requirements**

4.1 Performance Requirements

1. **Future Enhancements**
2. **Appendix**

# INTRODUCTION

Custom Messaging Application is a system for messaging between two or more systems. The messages will be exchanged via client server socket mechanism. This will enable two systems to talk to each other in real time.

**1.1 Overview:**

The Custom Messaging Application is a system in which the user can re login into the system using the valid username and password. After successful authentication, the user will be able to communicate with present active users and create private chat groups with the required users. The users can also logout from the application by entering exit command.

**1.2 Project Scope:**

The objective of the project is to provide a messaging application that ensures communication between users in an optimal manner with ease. To summarize, it covers the following aspects -

* User login to the application
* Chat with active users
* User data will be stored and monitored based on login and logout.

# OVERALL DESCRIPTION

## 2.1 Project Feature

* This application is used to connect to multiple clients at the same time and exchange messages.
* The first and foremost step in the efficient and smooth use of this application is the establishment of a server on one of the terminals.
* Users can get connected to the server once the server is established. Multiple users can connect to this server at the very same time.
* This list of active users is displayed to the newly logged in user and on the server side.

## 2.2 User Needs

1. User Characteristics: The user should be familiar with menu-driven Applications.
2. General Constraints: A full internet connection is required for Linux (Operating System).

## 2.3 Operating Environment

The operating environment for the application is listed below

● Operating system: Any Linux-based OS

# SYSTEM FEATURES

## 3.1 Functional Requirements

3.1.1.G3\_FR01: User will be asked to create username and password if he was logging in to the app for the first time. User can select Login from the menu.

3.1.2.G3\_FR02: User can enter special characters and the minimum length of the password should be 2 characters. The special characters can be @, $, %, &, \*.

3.1.3.G3\_FR03: User data will be stored and monitored based on login and logout.

3.1.4.G3\_FR04: The server will be matching the entered username with users list and if found, it will send a message to client saying username already exists. If not, it will ask the user to register.

3.1.5.G3\_FR05: User will be displayed active users list and given the option to select when private chat is selected.

3.1.6.G3\_FR06: Users can select a person from the list and continue to talk with him.

3.1.7.G3\_FR07: User can log off from private chat by entering a ‘exit’ command and the other user will be notified and exit from private chat and his data will be removed from the server.

3.1.8.G3\_FR08: User will be displayed all active group names and can select a group or can create a new group.

3.1.9.G3\_FR09: If another client selected the same group, then he will be joined, and this notification will be sent to all the people in the group. This is an optional requirement.

3.1.10.G3\_FR10: User can start messaging and can exit by giving a command

3.1.11.G3\_FR11: User data will be removed from the server once he types ‘exit’

3.1.12.G3\_FR12 : Users can send the files to the server and store them into the server in a separate folder. This is an optional requirement.

3.1.13.G3\_FR13: User should provide the complete path name of the file and can start sharing. This is an optional requirement.

3.1.14.G3\_FR14: If the file name is incorrect or any error in sending data then the user will exit the app. This is an optional requirement

3.1.15.G3\_FR15: User can share the files with other users and the server will send files to other clients. This is an optional requirement.

**3.2 Technical Requirements**

3.2.1.G3\_TR01: Process synchronization -It is the way by which processes that share the same memory space are managed in an operating system. Here, it ensures that multiple clients accessing the common data i.e., Client messages is synchronized, thereby avoiding conflicts.

3.2.2.G3\_TR02: Shared Memory in Linux - All data related to the client data such as active users list etc. are shared by multiple clients. Mutex can be used for locking and unlocking the shared resources to avoid data corruptions and errors

3.2.3.G3\_TR03: Socket Programming in C – TCP -Socket programming is a way of connecting two nodes, here the client and server, on a network to communicate with each other and coordinate

3.2.4.G3\_TR04: Support for statistics - Server is responsible for the display of statistics related to availability of clients such as number of clients are active and exit.

## 3.3 System Requirements

System Requirements are types of functional requirements. These are features that are required for a system to function.

Software Interface:

Operating System: Windows XP (32/62 bit) and Linux which supports networking.

Hardware Interface:

Hardware requirements are:

* Processor: i3 or above
* ROM: 1TB (SSD/HDD)
* RAM: 8 GB or above

## NON-FUNCTIONAL REQUIREMENTS

1. **Maintainability**: Software must be capable of being maintained cost-effectively throughout its lifetime and can be modified with additional requirements.
2. **Compatibility**: Software must be compatible with all Linux environments.
3. **Availability**: Users must access the system whenever they want.

# EXTERNAL INTERFACE REQUIREMENTS

1. User Interface:
   1. GUI: There is no GUI involved or created for the project / application.
   2. CLI: The application is based on CLI, and the commands are given through it.
2. Hardware Interface:

The Application uses/accesses the hard disk for storing the data and to access the files. Access to the hardware requirements is managed by the operating system and the application.

* 1. LINUX-based operating system.
  2. Terminal to run.

# FUTURE ENHANCEMENTS

NA

# APPENDIX

* https://www.coursera.org/learn/linux-system-programming-introduction-to-buildroot/home/week/1