

**CUSTOM MESSAGING APPLICATION**

**Software Requirement Specification Document**

**Project Timeline: 28/11/2022 to 9/12/2022**

**Team Members**

Gowthami Bonu

Sameer Raj

Vara Lakshmi Pujari

Supriya Nulu

Seelam Navya Sahitya

**Table of Contents**

**1. Introduction**

1.1 Purpose

1.2 Intended Audience and Reading Suggestions

1.3 Scope of Application

**2. Overall Description**

2.1 Application Perspective

2.2 Application Features

2.3 Design and Implementation Constraints

2.4 Assumptions and Dependencies

**3. System Features**

3.1 Functional Requirements

**4. Technical Requirements**

**5. Non-Functional Requirements**

1. **Introduction**

**1.1 Purpose**

the purpose of Custom Messaging Application is, it makes easy to communicate with people anywhere in the world by sending and receiving messages in real time. With this chat Application, users can receive the same engaging and lively interactions through custom messaging features, just as they would in person.

**1.2 Intended Audience**

This document explains our team architecture, our team’s initial understanding of the user needs. It will help our team understand the system specifications and analyze our project's critical aspects. This Project targets all age groups and provides real time messaging services, so they can connect to the person they want to connect to or chat.

**1.3 Scope of Application**

The scope of the Custom Messaging Application makes it easy to communicate with people anywhere in the world by sending and receiving messages. The messages will be exchanged via client server socket mechanism. This will enable the two systems to talk to each other in real time.

**2. Overall Description**

**2.1 Application Perspective**

Custom messaging application gives the following information.

* Create User Id & Password:

If the user logs in for the first time, they must create the user id and password first.

* Login:

After creating user id and password user can login to the application with user id and password which is generated by the client server.

* Select Chat Types:

After Successfully login, server provides the active user list to the client. where they have the option to choose to chat in private or in a group.

**2.2 Application Features**

The major feature of this application is its client-server relationship with socket mechanism. It provides real time text transmission; it stores client data in the server with a separate folder. Users can send files to other users with the help of TCP socket.

**2.3 Design and Implementation Constraints**

1. The fragmentation schema and maintaining server-client relation.

2. Using Data structures and System calls for above application.

3. Generating response at user Interface for every phase in server.

4. Implementing process separately for each type of algorithm.

**2.4 Assumption and Dependencies**

It needs a client server socket mechanism to send or receive messages from one user to another user.

**3. System Features**

**3.1 Functional Requirements**

The Custom Messaging Application should display a main menu to User that shows the option to Create User id and password if the user is logging for the first time. After creating user id and password user data will be stored and monitored based on login and logout.

The server will match the username entered with the user list and if found, it will send a message to client that username already exists.

After Successfully login user will display the active user list and given the option to select the chat option, whether its private chat or group chat. The user can select any person from the list and continue to talk with him.

User can log off from a private chat by giving some command and the other users will be notified and exit from private chat and his data will be removed from the server.

Users will be displayed the Active group names and can select a group or create a new group. If another client selected the same group, then he will join, and this notification will be sent to all people in the group.

Users can start messaging and exit by giving a command. Users' data will be removed from the server once he types exit command.

Users can send files to the server and store them in the server in a separate folder by giving them a complete path name and then they can start sharing the files. If the file name is incorrect in sending the data, then user will exit from app.

**4. Technical Requirements**

* Process synchronization
* Shared Memory in Linux
* Socket Programming in TCP
* Support for statistics
* I/O Multiplexing (in Linux using CPP)
* Logging and debugging framework

**5. Non-Functional Requirements**

* Multi-file multi-directory solution is expected.
* Make-file to build applications.
* Use Val-grind tool on application executable to detect memory leak. Final Val-grind report to be submitted in “reports” directory.
* Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and 2 flowcharts showing core functions logic.
* Use CPP Unit to automate unit testing.
* Any other UML diagrams developed while designing the system.
* HLD, LLD of the system.
* RTM, Plan, Presentation
* Unit test cases and Integration test cases in UT\_IT document. Both types of test cases i.e., sunny, and rainy should be present in this document