****

**Multi Party Conference Chat**

**MPCC**



**Software Requirements Specification Version 0.1**

**Document Control :**

| **Project Revision History** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

| **Team Members** |
| --- |

| **Employee ID:** | **Name** |
| --- | --- |
| 46252675 | Shreya Kamath |
| 46252660 | Rajesh Kannan Mookkandi |
| 46252693 | Nikhitha Ravi |
| 46252343 | Sneha Mariam Chacko |

|  |  |  | **Table of Contents** | | | |  | | |  | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **.** [**I**](#bookmark=id.gjdgxs)**NTRODUCTION** | | | | | |  | | |  | | | **4** |
|  | 1.1 | | | Purpose | | |  | | |  | | | 4 |
|  | 1.2 | | | Scope | | |  | | |  | | | 4 |
|  | 1.3 | | | Definitions, Acronyms, and Abbreviations | | |  | | |  | | | 4 |
|  | 1.4 | | | References | | |  | | |  | | | 5 |
|  | 1.5 | | | Overview | | |  | | |  | | | 5 |
| **2.** | [**O**](#bookmark=id.30j0zll)**VERALL DESCRIPTION** | | | | | |  | | |  | | | **5** |
| **3.** | [**S**](#bookmark=id.1fob9te)**PECIFIC REQUIREMENTS** | | | | | |  | | |  | | | **6** |
|  | 3.1 | | | Functionality | | |  | | |  | | | 6 |
|  |  | | |  | 3.1.1 Client Server Connection | | |  | | |  | 6 | | |
|  |  | | |  | 3.1.2 Client Registration | | |  | | |  | 7 | | |
|  |  | | |  | 3.1.3 Client Authentication | | |  | | |  | 7 | | |
|  |  | | |  | 3.1.4 Active Client Session  3.1.5 Concurrent Server to serve multiple clients  3.1.6 Message Broadcast  3.1.7 User Registration  3.1.8 Session Termination  3.1.9 Encryption and Decryption  3.1.10 Debug Log Messages | | |  | | |  | 7  7  7  7  8  8  8 | | |
|  | 3.2 | | | Usability | |  | | |  | | | | 8 | | |
|  | 3.3 | | | Security | |  | | |  | | | | 8 | | |
|  | 3.6 | | | Design Constraints | |  | | |  | | | | 9 | | |
|  | 3.7 | | | Use Case Diagram | |  | | |  | | | | 10 | | |

**Software Requirements Specification**

1. **Introduction**

The introduction of Software Requirements Specification(SRS) ) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather, analyze and give an in-depth insight into the complete Multi Party Conference Chat by defining all the requirements in detail. The intended audience includes developers, testers, project manager and the client. The detailed requirement of the Multi Party Conference Chat is provided in this document.

**1.1 Purpose**

Purpose of this document is to provide communication between multiple users. Whenever the client sends a message, it will be broadcasted to all other active clients who are connected to the server.

**1.2 Scope**

In this project the multi party conference chat allows multiple clients to connect with a server and each client will have their own unique user ID, when each client sends a message it will be broadcasted to all other active clients and these messages will be displayed.

**1.3** **Definitions, Acronyms, and Abbreviations**

| MPCC | Multi Party Conference Chat |
| --- | --- |
| SRS | Software Requirement Specification |
| TCP | Transmission Control Protocol |

**1.4 References**

1. <https://stackoverflow.com/questions/18681742/what-is-better-for-instant-messenger-tcp-or-udp>
2. System Requirements Specification Document
3. <https://www.ijert.org/research/end-to-end-multi-client-encrypted-windows-chat-system-IJERTV7IS030178.pdf>

**1.5 Overview**

The remaining sections of this document provide a general description, including characteristics of the users of this project, functionalities, and other requirements of the proposed system such as functional requirements, supporting information etc.

1. **Overall Description**

The Multi party conference chat is mainly based on the concept of achieving an effective communication channel allowing multiple users to communicate with each other easily. The MPCC allows the user to get registered and upon successful registration communicate with other users .It also enables broadcasting of messages sent by one user to all the other users active in the conference chat. The MPCC supports encryption and Decryption of login credentials such as user ID and password and also the message sent to attain security in the chat system. It works in a client-server model which uses TCP/IP protocol. The TCP/IP protocol is used as it is a reliable protocol which provides acknowledgement regarding the packet delivery whereas UDP protocol does not provide an acknowledgement for successful message delivery which might lead to data loss. The server performs client authentication and broadcasts the message from one user to all the other users in the chat.

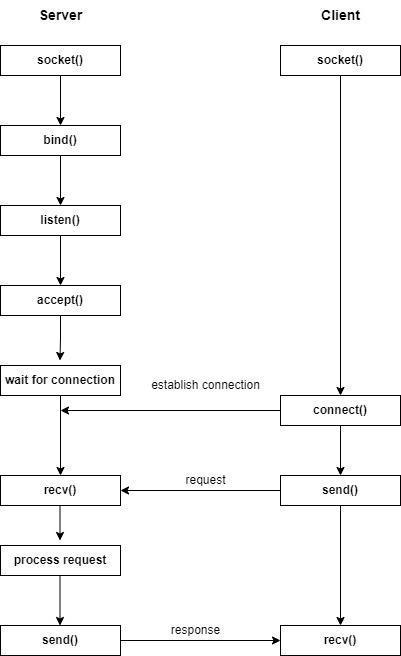
**3. Specific Requirements**

**3.1 Functionality**

**3.1.1 Client-Server Connection**

The Client-Server Connection is achieved by TCP/IP protocol. Transmission Control Protocol (TCP) is a connection-oriented protocol . It makes sure that the Server and client establish a connection with each other.

The below block diagram shows the steps followed to establish a connection between client and server. In MPCC, a single server supports multiple clients. Initially a socket is created and an address is assigned to the socket.The listen() system call marks the socket created as a passive socket and accepts incoming connection requests. After the server accepts the client, connection is established between client and server.



**3.1.2 Client Registration**

After we establish a successful connection between client and server, the users have to register using UserID and password . The registered credentials are stored into a file.

**3.1.3 Client Authentication**

Authentication is required to verify the credentials when the user logins to access the session. It is crucial because it's a key step in the process that keeps unauthorized users from gaining access to the session.

**3.1.4 Active Client Session**

The server should be always available to establish connection with the clients. If the client connection is lost then the client needs to reconnect with the server.

**3.1.5 Concurrent Server to Serve Multiple Clients**

Concurrent Server can handle multiple clients at the same time. A concurrent server can be achieved by multithreading or multiprocessing. Every client will send the chat request and this has to be maintained by the concurrent server.

**3.1.6 Message Broadcast**

Multiple clients are connected with a server. Each client will have their own unique user ID. Whenever the client is sending a message, it will be broadcasted to all other active clients who are connected to the server.These messages will be displayed and it can be identified by the client’s user ID.

**3.1.7 Registered Users**

Each client has to register with a unique userID and valid password so that it can be connected to the server. The server will store the list of registered users in a file using a file handling mechanism which enables the registered clients to participate in the upcoming session as an existing user.

**3.1.8 Session Termination**

Every client will have their own session while connected to the server. Once the session is completed the client gets disconnected.

**3.1.9 Encryption and Decryption**

Encryption is the process of encoding information to prevent anyone other than its intended recipient from viewing it. Encryption converts data into scrambled text. The unreadable text can only be decoded with a secret key.The purpose of encryption is to protect the confidentiality of data.In order to attain security in the chat system encryption is used in sending messages, user ID and password.

Decryption is the process that transforms the encrypted information into its original format. Decryption also provides confidentiality to private data and it helps to ensure that the record or file remains unchanged.

**3.1.10 Debug Log Messages**

* FATAL - Any error that is forcing a shutdown of the service or application to prevent data loss
* INFO - Generally useful information to log (service start/stop, configuration assumptions, etc).
* WARNING - Anything that can potentially cause application oddities
* DEBUG- Information that is diagnostically helpful to people more than developers.

**3.2 Usability**

The system is user-friendly enabling the users to communicate with each other.

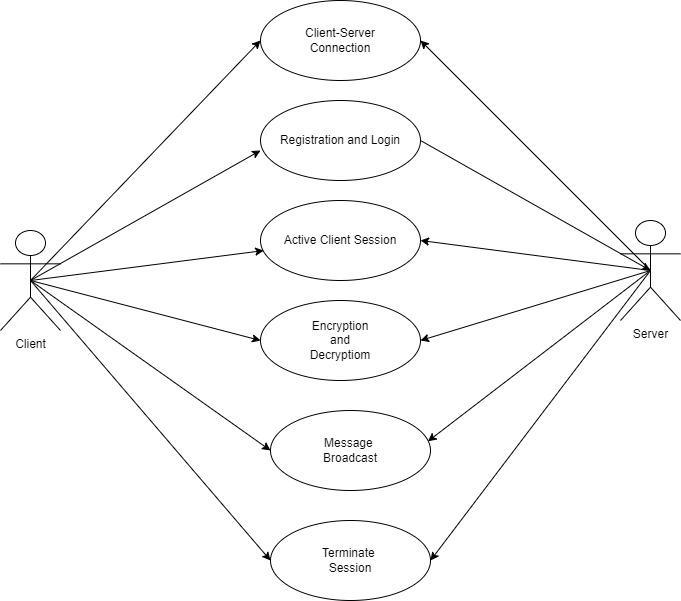
**3.3 Security**

Encryption and Decryption techniques are used which enables security in chat system.

**3.4 Design Constraints**

The system is built using only C++ language.

**3.5 Use Case Diagram**

****