**Multi-Party Conference Chat (MPCC)**

**Project Documentation**

**Published By**

Maddali Santhosh

Table of Contents

[1. Introduction 3](#_Toc192210651)

[1.1 Purpose 3](#_Toc192210652)

[1.2 Scope 3](#_Toc192210653)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc192210654)

[1.5 Overview 3](#_Toc192210655)

[2. Overall Description 3](#_Toc192210656)

[2.1 Product Perspective 3](#_Toc192210657)

[2.2 Product Functions 4](#_Toc192210658)

[2.3 User Classes and Characteristics 4](#_Toc192210659)

[2.4 Operating Environment 4](#_Toc192210660)

[2.5 Design and Implementation Constraints 4](#_Toc192210661)

[3. Specific Requirements 4](#_Toc192210662)

[3.1 Functionality 4](#_Toc192210663)

[3.2 Usability 6](#_Toc192210664)

[3.3 Reliability & Availability 6](#_Toc192210665)

[3.5 Supportability 6](#_Toc192210666)

[3.7 Architecture and Use case Diagrams 7](#_Toc192210667)

[Use Case Diagram 7](#_Toc192210668)

[3.9 Data Storage 8](#_Toc192210669)

[4. Technical Information 9](#_Toc192210670)

[4.1 System Calls 9](#_Toc192210671)

[5. Final Features 9](#_Toc192210672)

[6. Conclusion 9](#_Toc192210673)

**System Requirements Specifications:**

# 1. Introduction

## 1.1 Purpose

This document details the development of the Multi-Party Conference Chat (MPCC) system, a real-time, text-based chat application designed for multiple users.

## 1.2 Scope

The project includes the design, development, and testing of both client and server applications, facilitating real-time communication through a commandline interface.

## 1.3 Definitions, Acronyms, and Abbreviations

* **MPCC:** Multi-Party Conference Chat
* **TCP:** Transmission Control Protocol
* **CLI:** Command-Line Interface
* **XOR:** Exclusive OR (encryption method)

**1.4 References**

N/A

## 1.5 Overview

MPCC utilizes a client-server architecture with TCP sockets. The server, implemented in C, manages client connections, message broadcasting, and user registration. The client, also in C, provides a CLI for user interaction. A simple XOR encryption is used for basic security.

# 2. Overall Description

## 2.1 Product Perspective

MPCC is a network application with a server component deployed on a server and client applications running on user machines.

## 2.2 Product Functions

* User registration and authentication.
* Real-time message broadcasting.
* Session management.
* Basic message encryption and decryption.
* Server activity logging.

## 2.3 User Classes and Characteristics

* **Users:** Use the CLI client to participate in chat.
* **Administrators:** Maintain and monitor the server.

## 2.4 Operating Environment

* **Server:** Linux-based OS.
* **Client:** Cross-platform CLI.

## 2.5 Design and Implementation Constraints

* Implemented in C.
* **TCP** sockets for network communication.
* **select()** system call for handling multiple clients.
* **XOR** encryption.

# 3. Specific Requirements

## 3.1 Functionality

* **Client-Server Connection:**
  + - TCP socket connections. o The server must listen for incoming client connections on a specified port.
    - **Technologies:** socket(), bind(), listen(), accept(), connect(), send(), recv().
* **User Registration:**
  + - New users must be able to register with a unique username and password.
    - Stores encrypted user credentials in a file.
    - **Technologies:** fopen(), fprintf(), fgets(), strcspn().
* **Client Authentication:**
  + Existing users must be able to authenticate with their username and password. o The user cant connect and receiving message like authentication error
* **Active Client Session:**
  + The server must maintain an active session for each connected client. o In case of connection loss, the client must be able to reconnect and resume the session.
* **Message Broadcasting:**
  + Broadcasts messages to all connected active clients. o Simultaneously in server side also it should display.
  + **Technologies:** send(), recv().
* **Session Termination:**
  + Sessions must be terminated when a client disconnects.
  + Prevent the inactive users from consuming system resources.
  + Close(socket\_fd) to terminate the session.
* **Session Management:**

Handles client connections and disconnections.

* + **Technologies:** select().
* **Encryption/Decryption:**
  + XOR encryption for basic security.
  + **Technologies:** XOR bitwise operation.
* **Server Logging:**
  + Logs server activity to a file.
  + **Technologies:** fopen(), fprintf(), time(), localtime(), strftime().
* **Concurrent Clients:**
  + - Server handles multiple client connections using select().
    - **Technologies:** select(), file descriptors.

## 3.2 Usability

* CLI for user interaction.
* Informative feedback.

## 3.3 Reliability & Availability

* Error handling and recovery.
* Handles network disruptions.
* Designed for high availability, minimizing downtime

**3.4 Security**

* XOR encryption of user credentials and messages.

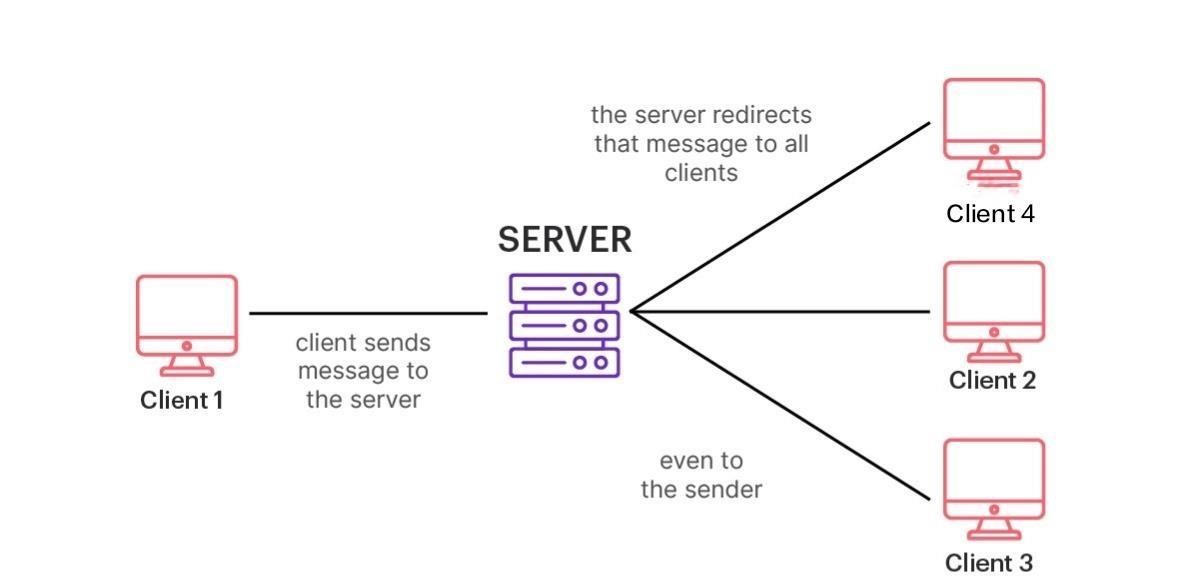
## 3.5 Supportability

* Server activity logging.

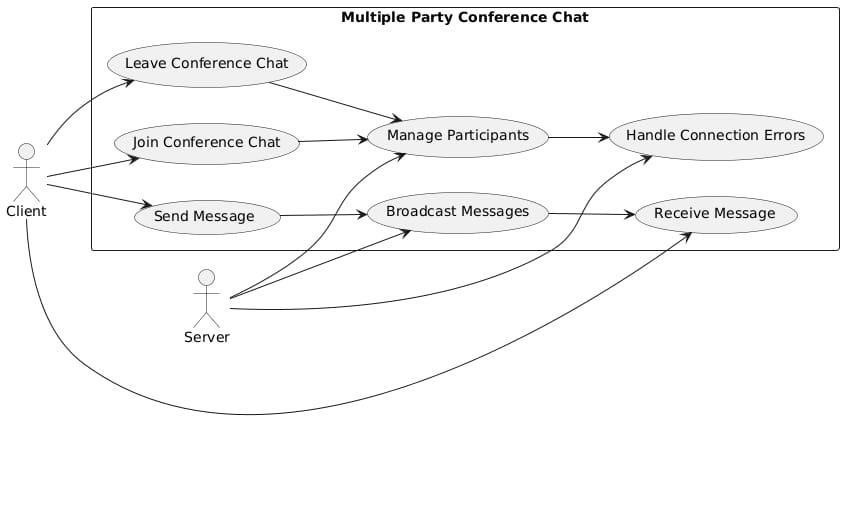
**3.6 Design Constraints** • C implementation.

* TCP sockets.
* select() system call.
* XOR encryption.

## 3.7 Architecture and Use case Diagrams



# Use Case Diagram



The diagram shows how the "Client" interacts with the MPCC system to join a chat, send and receive messages, manage participants, and leave the chat. The "Server" handles broadcasting messages and managing connection errors.

**3.8 Error Handling**

• Handles connection errors, file I/O errors, and network errors.

## 3.9 Data Storage

* Encrypted user credentials in a file ("registered\_users.txt").
* Server activity log file ("server\_log.txt").

**3.10 Scalability**

* Handles a limited number of concurrent clients.

**3.11 User Interface**

* CLI.

**3.12 Testing**

* Unit, integration, and system testing.

# 4. Technical Information

## 4.1 System Calls

* socket(), bind(), listen(), accept(), connect(), send(), recv(), select(), fopen(), fprintf(), fgets(), strcspn(), time(), localtime(), strftime(), close().

**4.2 TCP Sockets**

* Reliable network communication.

**4.3 Encryption/Decryption**

* XOR encryption.

**4.4 Select() System Call**

# 5. Final Features

* User registration and authentication.
* Real-time chat.
* Basic security.
* Server logging.

# 6. Conclusion

This documentation provides a comprehensive overview of the MPCC system.

It also provides information about how the connection between server and multiple clients takesplace .