**Group #4 – Design**

Design Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| mm/dd/yyyy | 1.0 | Initial Version | Your Name |
| 03/30/2022 | 1.1 | Updated Document | Anthony Lopez |
| 4/4/2022 | 1.2 | Updated Document | Matthew Baron |
| 04/06/20222 | 1.2.1 | Fixed formatting, added class diagram | Owen Casebeer |
| 4/30/2022 | 1.3 | Updated Diagrams and Document | Maxwell Wu |
| 5/1/2022 | 1.3.1 | Updated class diagram | Maxwell Wu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

**1.** **INTRODUCTION 4**

1.1. Goals and Objectives 4

1.2. Statement of Scope 4

1.3. Software Context 4

1.4. Major Constraints 4

**2.** **DAta DESIGN 5**

2.1. Client Side 5

2.2. Server Side 5

2.3. Chat Room Side 5

2.4. Message Side 5

**3.** **ARCHITECTURAL and COmPonent-level design 6**

3.1. Program Structure 6

3.2. Description of Client 6

3.3. Description of Server 6

3.4. Software Interface Description 6

**4.** **User Interface Design 7**

**5.** **Restrictions, LiMitations, and Constraints** [**8**](#_heading=h.1ksv4uv)

**6.** **Testing Issues** [**9**](#_heading=h.1ksv4uv)

# Introduction

## Goals and Objectives

This document describes important aspects of the implementation of the Communications System.

## Statement of Scope

Decisions in this document are made based on the following priorities (most important first): Maintainability, Usability, Portability, Efficiency

## Software Context

Server information will be maintained within the Server object. The Server object's various commands will handle the receiving and sending of all communications between the clients.

The Client will be responsible for sending and receiving data from the Server and present this data to the user with a GUI.

## Major Constraints

**Issue 1**: Where should we store the database?

**Option 1.1**: We can store the database in the Server object.

**Option 1.2**: We can store the database in a separate file that is properly sorted and accessible by the Server object. The Server object will sort in chronological order and within its respected chatroom.

**Decision:** We should go with 1.1. The database should only be on the “server” since only the Server accesses it because the clients only access the server.

**Issue 2**: Where should we store the information of usernames and passwords?

**Option 1.1**: We can store this information within the Server object.

**Option 1.2**: We can store the information in a separate file that the Server will search through.

**Decision:** They should be stored in the database which after Issue 1 I think should be part of the Server object.

# Data Design

## Client Side

The Client will connect to the server's socket to communicate with it. The Client will communicate with the Server to create chatrooms with the createChatroom() method. The user that starts the chatroom becomes its host and is the only one who may lock the Chatroom. A response from the Server will be displayed on the GUI confirming a successful creation of a Chatroom.

The Client will be able to send Messages to other clients through the server and using the sendMessages() method. The Client can be within multiple chatrooms and send messages to other users in it. The Client will be able to read messages from other users from the chatrooms they are in. A response from the Server will be displayed on the GUI confirming a Message was successfully sent.

Once a user successfully logs into the client, they are able to invoke the joinChatroom() or createChatroom() methods.

## Server Side

The Server has a server socket that multiple clients may connect to. Once a Client connects to the Server, it will create an individual thread and a ClientHandler object for that Client in order to send and receive Messages from them.

The Server will automatically receive Messages from the Client through their individual thread. The Server will then handle that Message and respond accordingly. Messages

Messages from the Chatroom will be distributed to the rest of the Clients inside it by the Server.

## Chat Room Side

Chatrooms can be created once a Client successfully logs into their User account. The Client that creates the Chatroom will be deemed the host and be able to lock the Chatroom so no one else can join.

Clients within Chatrooms can send Messages to all other Clients that are inside the Chatroom through the Server. The Chatrooms will log Messages within them that can be later read by a supervisor.

## Message Side

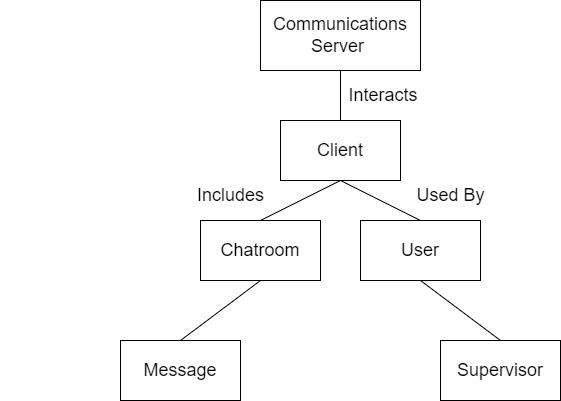
Messages will be used to communicate between Clients and the Server. Messages will have types that cannot be changed after their initialization. Messages will contain a status and text members that can be read by the Client or the Server.

# Architectural and Component-level Design

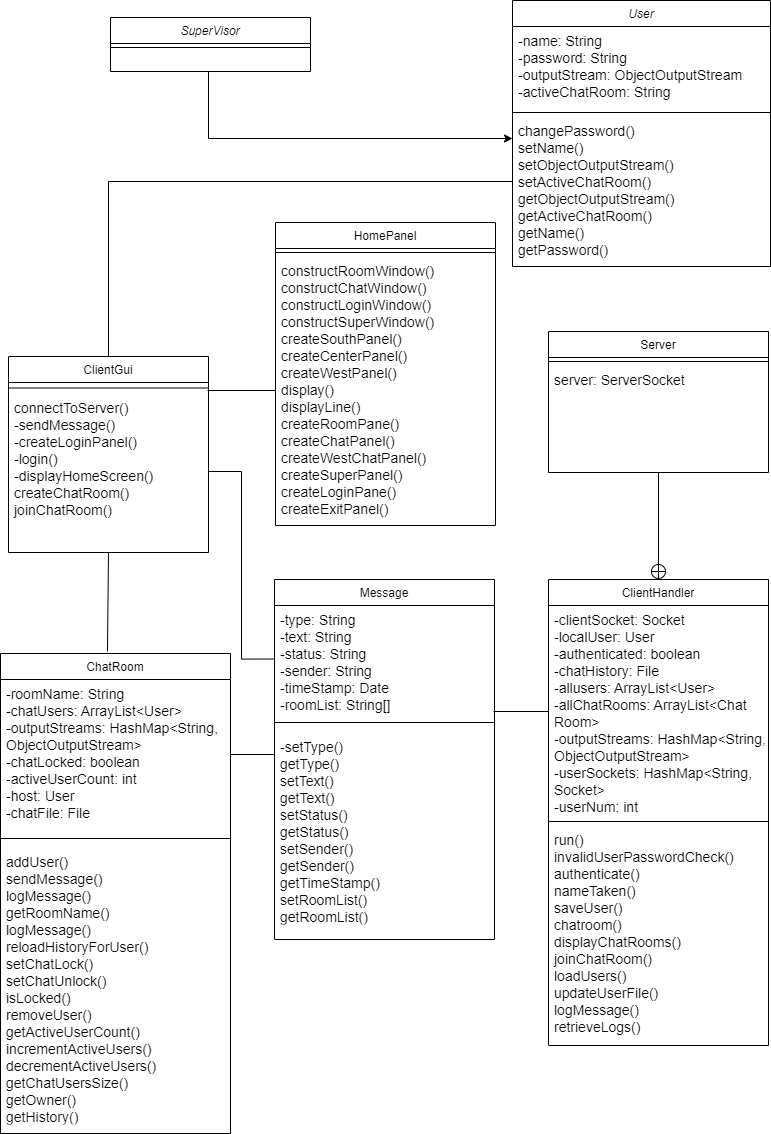
## Program Structure

The Communications System runs as a client-server application.

## Architecture Diagram:

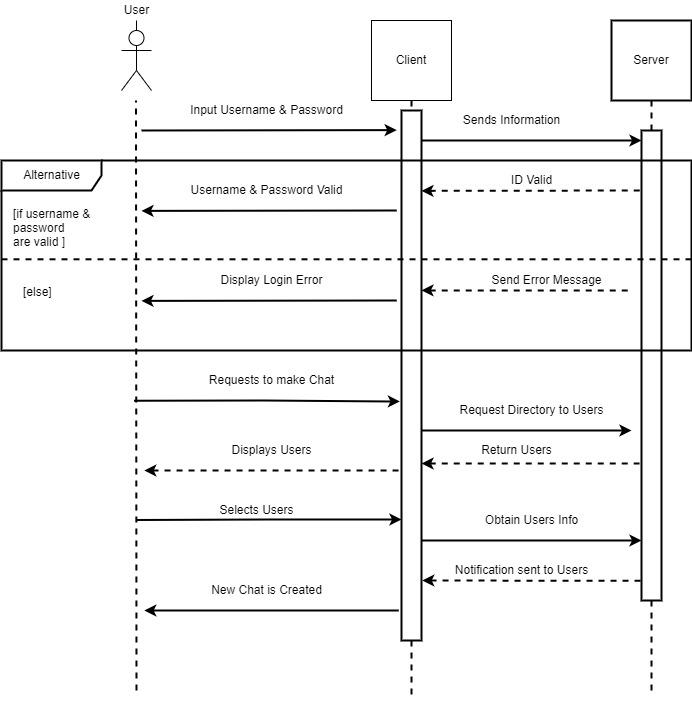


**Class Diagram**

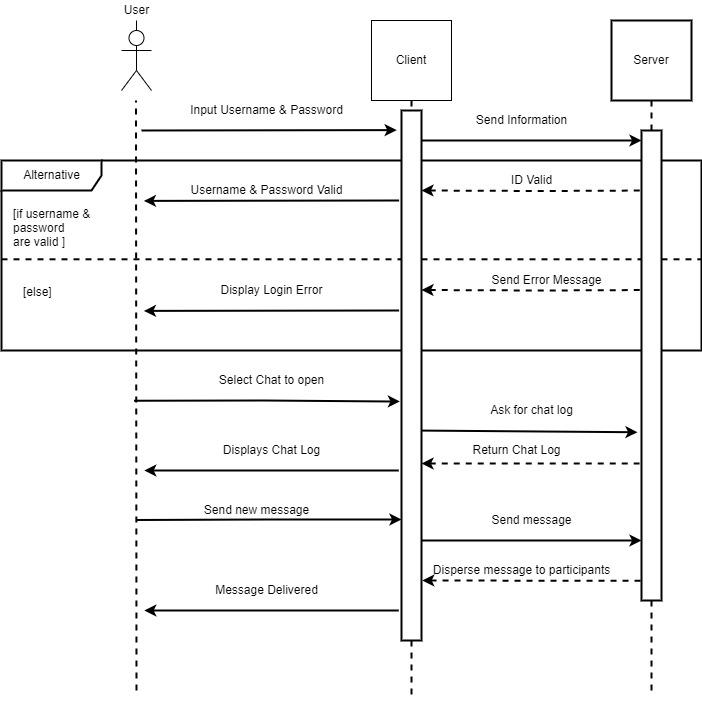


**Sequential Diagrams**

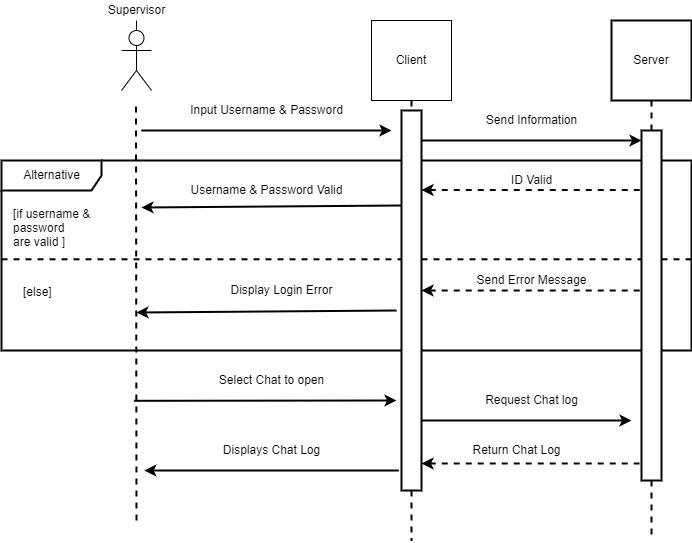
**Sequential Diagram Use Case #1000: Create a Chatroom**

****

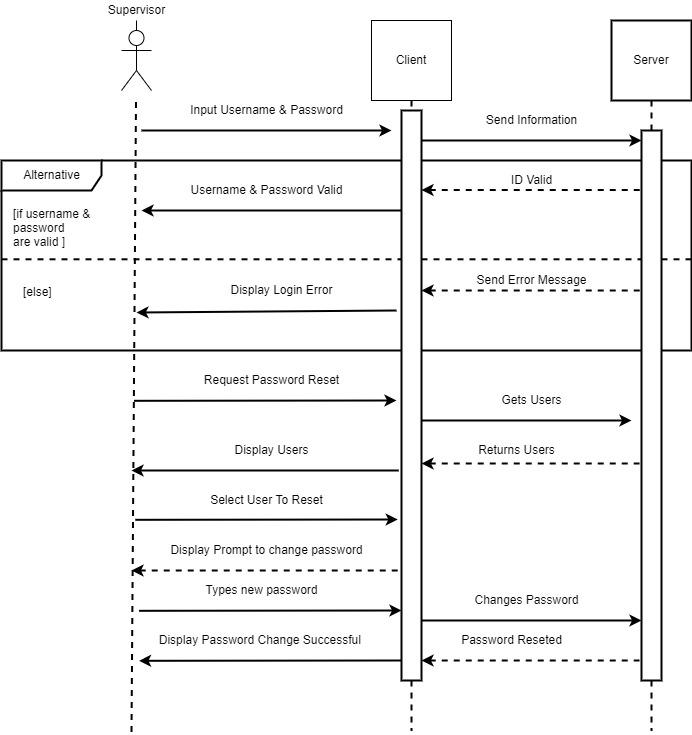
**Sequential Diagram Use Case #2000: Send a message**

****

**Sequential Diagram Use Case #3000: Access Chat History**

****

**Sequential Diagram Use Case #4000: Change Password**

****

## 

## Description of Client

## 3.2.1 Client processing narrative

connectToServer() - This method will allow the Client to connect to a server's socket and establish a connection between both.

sendMessage() - This method will allow the Client to communicate with the Server using a Message object.

createLoginPanel() - This method will use the JOptionPane to create a GUI that Client can use to login to their user account.

login() - This method creates a login Message that will be sent to the Server in order to be authenticated according to its internal database.

displayHomeScreen() - This method uses the JOptionPane to create a GUI where the Client may access Chatrooms or log out.

createChatroom() - This method will create a new Chatroom and the Client will be set as its host.

joinChatroom() - This method will allow the Client to join a preexisting Chatroom created by other Clients.

## 3.2.2 Client Interface description

The Client interface will be constructed with the use of JOptionPane library. The Client will begin with a login window. Once logged into the client there will be a menu with options to choose between making a chatroom, opening an existing chatroom, and logging off. Once in a chat room there will be various buttons to activate various methods including sending messages, locking/unlocking the chat, adding, and removing users and exiting the chat.

## 3.2.3 Client processing details

The Client will be implemented with the aid of the functions and processes given by the JOptionPane library. We will be using a do/while loop with an embedded switch statement to initiate the various methods that are needed for our main menu within our project. Each case will have a method that will accomplish a particular task. These methods will be activated when a button is clicked on within the Client. This process will be used within the possible selections of actions within the main menu and within the Chatrooms.

## Description of Server

**3.3.1 Server processing narrative**

authenticate() - This method will save a new user's information to the server's personal database file.

saveUser() - This method will save a new user's information to the server's personal database file.

loadUsers() - This method loads in all user information using the server's personal database file.

updateUserFile() - This method updates a user's information inside its personal database file for later use.

logMessage() - This method will log a sent Message's information and be put into a log within the server's database.

retrieveLogs() - This method retrieves the Server's stored logs that can be accessed by a supervisor.

**3.3.2 Server interface description**

The Server will receive all its data from the client and will record all the messages that pass through it. Furthermore, the Server will redistribute the message to the selected recipients, sending the data to their clients. The messages will be Asynchronous.

**3.3.3 Server processing details**

The Server will begin the processing and recording of Messages once one is sent from a client.

# User Interface Design

### 

A picture containing graphical user interface

Description automatically generated

The Client Interface will open with a User Login Page that will have two text areas asking for a Username and Password. Underneath them will be a Login button to submit the username and password. If the Login is unsuccessful the user will be brought back to the login to try again.

Diagram

Description automatically generated

Once the User has successfully logged in they have a decision between 3 different buttons. The first button will allow the user to create a new chatroom as the host. The second button will allow the user to open an existing chatroom. The third button will log them out of the system returning them to the login window.

Diagram

Description automatically generated

If the User wants to make a Chatroom, the user will be presented with a text area to give the chatroom a name. There will also be 2 buttons present. The first to confirm the creation of the chatroom and the second to cancel the operation and return to the main menu window.

Graphical user interface, application

Description automatically generated

If the User wants to open an existing Chatroom, a dropdown menu will appear with the existing Chatrooms to join. Once a preferred chatroom is selected the user can confirm with the help of a button. There will also be a button to cancel the operation which also returns them to the main menu window.

Graphical user interface, text, application, email

Description automatically generated

Once the User opens or creates a chatroom the client will transition to the chat page where the user can send and receive messages from other users. From this page the user can add or remove users with buttons. The host will be able to lock and unlock with buttons. Any Supervisor will be able to read the entire chat logs history with the help of a button. Lastly a button will be available to exit the chat room and return to the main menu.

# Restrictions, Limitations, and Constraints

There is only one Server instance that handles all the Client instances at once. Large amounts of concurrent Clients and Messages may reduce the speed and efficiency of the Server.

# Testing Issues

It will be difficult to test the full capabilities of the Communication System with only a single user logged into the system. Therefore, to solve this issue there will need to be test sessions where multiple users are logged into the Communication System at the same time from varying locations.