

# **Communications Project Group 4**

## *Software Requirements Specification*

## Revision History

Date	Revision	Description	Author
9/26/23	1.00	Initial Version	Benjamin Newton Aaron Nguyen Kosta Nikolaou Jagjot Nijjar
10/3/23	1.01	Section 1.3 - Added Use Case Specifications and Diagram	Aaron Nguyen
10/3/23	1.02	Section 1.3 - Added Class Diagram	Aaron Nguyen
10/3/23	1.03	Section 1.3 - Added Sequence Diagram	Aaron Nguyen
10/4/23	1.04	Section 1.3 - Modify Class Diagram	Aaron Nguyen
10/4/23	1.05	Section 3 - Added Specific Requirements	Benjamin Newton
10/5/23	1.06	Section 4 - Added Non-Functional Requirements	Benjamin Newton
<u>10/10/23</u>	1.07	Section 1.3 - Modify Class Diagram Section 3.1 - Add functional requirements	Aaron Nguyen Jagjot Nijjar Kosta Nikolaou
<u>10/17/23</u>	1.08	Section 2.1 - Add product perspective Section 3.2, 3.3, 4.2, 4.3 - Add additional requirements	Benjamin Newton Aaron Nguyen Jagjot Nijjar Kosta Nikolaou
<u>10/24/23</u>	1.09	Section 1.3 - Update UML class diagrams and sequence diagrams	Benjamin Newton Aaron Nguyen Jagjot Nijjar Kosta Nikolaou
<u>11/14/23</u>	1.10	Section 1.3 - Add missing ClientHandler, Client, GUI UML	Aaron Nguyen
<u>11/29/23</u>	1.11	Section 1.3 - Move UML Diagrams to Design Document	Benjamin Newton
<u>12/4/23</u>	1.12	Added use cases 5, 6, 7, 8	Kosta Nikolaou

# Table of Contents

<b>1. Purpose</b>	<b>4</b>
1.1. Scope	4
1.2. Definitions, Acronyms, Abbreviations	4
1.3. References	4
1.4. Overview	9
<b>2. Overall Description</b>	<b>10</b>
2.1. Product Perspective	10
2.2. Product Architecture	10
2.3. Product Functionality/Features	10
2.4. Constraints	10
2.5. Assumptions and Dependencies	10
<b>3. Specific Requirements</b>	<b>11</b>
3.1. Functional Requirements	11
3.1.1. Common Requirements:	11
3.1.2. Server Module Requirements:	11
3.1.3. Client Module Requirements:	11
3.1.4. Room Module Requirements:	11
3.2. External Interface Requirements	11
3.3. Internal Interface Requirements	11
<b>4. Non-Functional Requirements</b>	<b>12</b>
4.1. Security and Privacy Requirements	12
4.2. Environmental Requirements	12
4.3. Performance Requirements	12

# 1. Purpose

This document outlines the requirements for our Communication System (CS).

## 1.1. Scope

This document will catalog the user, system, and hardware requirements for the Communication System.. It will not, however, document how these requirements will be implemented.

## 1.2. Definitions, Acronyms, Abbreviations

CS - Communication System

## 1.3. References

### Use Case Specifications

**Use Case ID:** 1

**Use Case Name:** Login

**Relevant Requirements:** N/A

**Primary Actor:** User

**Pre-conditions:**

1. Server is running
2. Client is connected to the server
3. User has an account registered in the system

**Post-conditions:** User is logged into the messaging system client

**Basic Flow or Main Scenario:**

1. User enters their username and password in the client
2. Client sends a login request to the server
3. Server authenticates the user
4. On successful authentication, the server responds with a success indicator

**Extensions or Alternate Flows:**

4. On unsuccessful authentication, the server responds with failure indicator
5. User attempts to fix incorrect credentials and attempts login again

**Exceptions:** N/A

**Related Use Cases:** Creating a room (2), Selecting a room (3), Sending a message (4), Updating user status (5)

**Use Case ID: 2**

**Use Case Name:** Creating a room

**Relevant Requirements:** Login (1)

**Primary Actor:** User

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client

**Post-conditions:** A room is created where the user can message other user(s)

**Basic Flow or Main Scenario:**

1. User clicks on a button in the client to create a new room
2. User selects participants to be invited to the room
3. If only one participant is selected, no room name is required

**Extensions or Alternate Flows:**

3. If more than one participant is selected, the user must enter a name for the room

**Exceptions:** N/A

**Related Use Cases:** Sending a message (4)

**Use Case ID: 3**

**Use Case Name:** Selecting a room

**Relevant Requirements:** Login (1)

**Primary Actor:** User

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client

**Post-conditions:** The selected room is displayed in the user's client

**Basic Flow or Main Scenario:**

1. After logging into the client, the user is presented with a list of rooms they are a participant in
2. Clicking on a room displays the room in their client which includes messages from others and a list of participants

**Extensions or Alternate Flows:** N/A

**Exceptions:** If no room is available, the list of rooms will be empty

**Related Use Cases:** Sending a message (4)

**Use Case ID: 4**

**Use Case Name:** Sending a message

**Relevant Requirements:** Login (1)

**Primary Actor:** User

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client
3. User has a room selected

**Post-conditions:** A message is sent to the appropriate group

**Basic Flow or Main Scenario:**

1. After entering a room, the user is presented with a text field
2. The user can input text into this field and press 'Enter' to send the message
3. If the message is successfully delivered, they are presented with a success indicator

**Extensions or Alternate Flows:**

4. If the message fails to be delivered, they are presented with a failure indicator

**Exceptions:** N/A

**Related Use Cases:** Selecting a room (3)

**Use Case ID: 5**

**Use Case Name:** Updating user status

**Relevant Requirements:** Login (1)

**Primary Actor:** User

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client

**Post-conditions:** The status of the user is updated

**Basic Flow or Main Scenario:**

1. After logging into the client, the user can see a "Change Status" button
2. Clicking this button opens a dropdown menu with a list of statuses
3. Clicking on a status will update their status to their selection
4. Other users will be able to see the updated status

**Extensions or Alternate Flows:** N/A

**Exceptions:** The 'Offline' status is available internally but is not available for selection by the user

**Related Use Cases:** N/A

**Use Case ID: 6**

**Use Case Name:** Adding User to Room

**Relevant Requirements:** Login (1), Creating a room (2)

**Primary Actor:** User/IT

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client
3. User is in a room.

**Post-conditions:** The user is no longer in a room

**Basic Flow or Main Scenario:**

1. User selects the “Add User” button.
2. User types in name in field.
3. User selects the “Add User” button.
4. Added user appears in the room.
5. Message appears in the chat indicating a user has been added to the room.

**Extensions or Alternate Flows:** N/A

**Related Use Cases:** N/A

**Use Case ID:** 7

**Use Case Name:** Leaving a room

**Relevant Requirements:** Login (1), Creating a room (2)

**Primary Actor:** User/IT

**Pre-conditions:**

1. Client is connected to the server
2. User is logged into the client
3. User is in a room.

**Post-conditions:** The user is no longer in a room

**Basic Flow or Main Scenario:**

1. User selects the room in the room list
2. User selects the “Leave Room” button
3. Users remaining in the room are notified.

**Extensions or Alternate Flows:** N/A

**Related Use Cases:** N/A

**Use Case ID:** 8

**Use Case Name:** Reading chat logs

**Relevant Requirements:** Login (1), Creating a room (2)

**Primary Actor:** IT

**Pre-conditions:**

1. Client is connected to the server
2. IT user is logged into the client

**Post-conditions:** Chat logs are presented to the IT user

**Basic Flow or Main Scenario:**

1. IT user selects the “View Logs” button.
2. IT user types the name of the user to be audited in the “Username” field
3. IT user selects “Get User Logs” button.
4. That user’s logs are displayed with accompanying metadata.

**Extensions or Alternate Flows:**

- 4.1 IT user puts room uuid in “Room ID” field.
- 4.2 IT user selects “Get Room Logs” button
- 4.3 That room’s logs are displayed to the IT User

**Related Use Cases:** N/A

### **Use Case Diagram**

Please refer to pg. 9 of Design Document for Use Case Diagram

### **UML Class Diagrams**

Please refer to pg. 3 of Design Document for UML Class Diagrams

### **Sequence Diagram**

Please refer to pg. 8 of Design Document for Sequence Diagram

## **1.4. Overview**

The Communication system will allow users to message people directly or groups of people through creating rooms.



## **2. Overall Description**

### **2.1. Product Perspective**

The product consists of a communication system that permits asynchronous and synchronous messaging between clients. Users will be able to message directly as well as in groups and IT users will be able to view all logs.

### **2.2. Product Architecture**

This system will be broken into three main modules: The server, the client, and the room

### **2.3. Product Functionality/Features**

The high-level features of the system are as follows (see section 3 of this document for more detailed requirements that address these features): Users will be able to message other users either directly or in a group by creating a room. Rooms will handle synchronous and asynchronous messaging as well as having a history function.

### **2.4. Constraints**

Users will only be able to send other users text. Files will not be supported at this time.

### **2.5. Assumptions and Dependencies**

It is assumed that users will not spam other users in rooms as there will not be strict spam filtering on rooms.

Users are dependent on themselves to know other people to send messages too, there will not be a social network aspect of the communication system.

## **3. Specific Requirements**

### **3.1. Functional Requirements**

#### **3.1.1. Common Requirements:**

- 3.1.1.1 Users should be able to message other users.
- 3.1.1.2 Users should be able to see when a message was sent.
- 3.1.1.3 Users should be able to see who sent a message.
- 3.1.1.4 Users should be able to see a list of users available to message.
- 3.1.1.5 Users should be able to see a list of groups they are in.
- 3.1.1.6 Users should be able to create groups.
- 3.1.1.7 Users should be able to search for messages or content within messages.
- 3.1.1.8 Users should be able to set a status indicating their availability or presence.

#### **3.1.2. Server Module Requirements:**

- 3.1.2.1 The server should authenticate users trying to log in from the client.
- 3.1.2.2 The server should create rooms for users when a new message room is established
- 3.1.2.3 The server should handle delivering messages to a room.
- 3.1.2.4 The server should keep an array of all the currently created rooms
- 3.1.2.5 The server should allow multiple users to be online and send messages at the same time

#### **3.1.3. Client Module Requirements:**

- 3.1.3.1 Users should be allowed to log in using their issued id and password, both of which are alphanumeric strings between 6 and 20 characters in length.
- 3.1.3.2 In the client, users should be allowed to see a list of rooms that they have already sent messages in.
- 3.1.3.3 In the client the user should be allowed to see a user list of who is currently connected to the server.
- 3.1.3.4 Users should receive all of the messages that they missed while they were logged out of the system.

#### **3.1.4. Room Module Requirements:**

- 3.1.4.1 In a room the user should be able to see a history of all messages sent before in that room
- 3.1.4.2 The user should be able to add new messages to any room they are a part of.
- 3.1.4.3 The room should keep a history of every message for IT level users.

### **3.2. External Interface Requirements**

- 3.2.1 The system will need to be opened each time that the computer is turned on. It will not open on startup.
- 3.2.2 The system must alert users when a new message appears with a sound.

3.2.3 The client application should provide a text field for input.

3.2.4 The client should have a notification system.

### **3.3. Internal Interface Requirements**

3.3.1 The system must allow an IT user to see all of the history of all users.

3.3.2 The system must have access to the filesystem for storing files.

3.3.3 The server application should return a list of all rooms a user is a participant in upon login

## **4. Non-Functional Requirements**

### **4.1. Security and Privacy Requirements**

- 4.1.1 The system must always send data using TCP/IP.
- 4.1.2 Server securely stores user passwords.
- 4.1.3 Messages between users should not be encrypted.
- 4.1.4 Information between the server and client is not encrypted.

### **4.2. Environmental Requirements**

- 4.2.1 System cannot require any other software in order to run properly.
- 4.2.2 The server application must be running on a computer connected to the internet
- 4.2.3 Any client application(s) must be running a computer connected to the internet

### **4.3. Performance Requirements**

- 4.3.1 The client must be able to open up to the login screen in less than 5 seconds.
- 4.3.2 The system must route messages in less than 2 seconds.
- 4.3.3 The system must authenticate a user in less than 2 seconds.