**Communication System**

Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| 09/25/2022 | 1.0 | Initial Version | Garrett Wong |
| 09/25/2022 | 1.1 | Extended Description, Requirements and added definitions. | Rylan Geer |
| 09/27/2022 | 1.2 | Extended Project Perspective | Garrett Wong |
| 09/28/2022 | 1.3 | Added Diagrams | Garrett Wong |
| 09/28/2022 | 1.4 | Revised the Requirements (3.1) section. Added several Use Cases (1.3) | Shane Cancilla, Rylan Geer |
| 11/3/2022 | 2.1 | Updated the Class Diagram | Everyone |
| 11/4/2022 | 2.2 | Added more Use Cases | Shane Cancilla |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

**1.** **Purpose 4**

1.1. Scope 4

1.2. Definitions, Acronyms, Abbreviations 4

1.3. References 4

1.4. Overview 4

**2.** **Overall Description 5**

2.1. Product Perspective 5

2.2. Product Architecture 5

2.3. Product Functionality/Features 5

2.4. Constraints 5

2.5. Assumptions and Dependencies 5

**3.** **Specific Requirements 6**

3.1. Functional Requirements 6

3.2. External Interface Requirements 6

3.3. Internal Interface Requirements 7

**4.** **Non-Functional Requirements 8**

4.1. Security and Privacy Requirements 8

4.2. Environmental Requirements 8

4.3. Performance Requirements 8

# Purpose

This document outlines the requirements for the Communication System.

## Scope

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

## Definitions, Acronyms, Abbreviations

Client - The part of the software that allows Users to connect to the Server in order to   
 send messages.

Server - The system that sets up a port and accepts connections from the client and   
 receives the messages in order to send them to other clients

User - An employee account that gets signed into through the client.

IT Client/user - A special user account that is allowed access to all of the message logs.

IT - Abbreviation for Information Technology.

## References

**Use Case ID: LoginCase1**

Use Case Name: User Login

Relevant Requirements: User connects to server and log into employee account

Primary Actor: User, Client, Server

Preconditions: Client is connected to server.

Postcondition: Client is logged into account and has access to messaging GUI.

Basic Flow:

1. User opens the terminal and connects to the server.
2. Users sign in into their employee account.
3. User is now in the messaging interface.

Alternate Flow:

1. User tries to sign in but fails.
2. System asks users some security questions to reset their password.
3. User changes the password.
4. User is now in the messaging interface.

Exceptions:

1. Databases to hold user information may be corrupted/modified and have missing information.
2. Users use invalid characters that raises an error.
3. User doesn't have an account yet.

Related Use Cases:

1. User signs up for the first time - LoginCase2
2. User wants to reset the password or change account information.

**Use Case ID: LoginCase2**

Use Case Name: First time user

Relevant Requirements: User creates an account in order to connect to the server and login.

Primary Actor: User, Client, Server

Preconditions: Client application is connected to server.

Postcondition: User created an account and can now login.

Basic Flow:

1. User opens the terminal and it connects to the server.
2. Users clicks sign up.
3. User enters their ID and preferred password.
4. Account is added to the server database.

Alternate Flow:

1. User tries to sign in but fails.
2. User gets prompted with a signup button and clicks it.
3. User enters their ID and preferred password.
4. Account is added to the server database

Exceptions:

1. Users used invalid characters that raises an error.
2. User ID is already taken.

Related Use Cases:

1. User logs in - LoginCase1
2. User wants to reset the password or change account information.

**Use Case ID: CreateMessageCase1**

Use Case Name: Messaging another user

Relevant Requirements: User messages another User

Primary Actor: User

Preconditions: Client is connected to server and is logged in

Postcondition: Both Users are able to see messages.

Basic Flow:

1. User A searches up User B using some unique identifier like an email or name.
2. User A types in messages and sends them to User B.
3. User A’s message should be shown to only User A and User B.

Alternate Flow:

1. User A already has a chat history with User B.
2. User A can directly select that message interface without explicitly searching User B’s information
3. User A and B begin messaging.

Exceptions:

1. User A is unable to find User B due to User B’s information being corrupted in the database.

Related Use Cases:

1. User can create a group chat and message multiple users

**Use Case ID: RemoveChannel**

Use Case Name: IT Removes Channel

Relevant Requirements: Channel is hidden/removed.

Primary Actor: IT User, Client, Server

Preconditions: IT User is logged in through the Client, and connected to Server.

Postcondition: The channel can know longer be written to, but the log is kept.

Basic Flow:

1. IT User selects chat/s to be hidden, and pushes to remove.
2. Channel is now invisible to all users, and cannot be contributed to.
3. Log of this chat being deleted is kept and stored in that chat's log.

Alternate Flow:

1. Channel is currently being accessed by users.
2. System returns the users within the channel to the list of chats with a notification that the chat has been removed.

Exceptions:

1. No chats exist yet

Related Use Cases:

1. IT User Login

**Use Case ID: RemoveMessage**

Use Case Name: User Removes Message

Relevant Requirements: Message is hidden/removed.

Primary Actor: User, Client, Server

Preconditions: User is logged in through the Client, and connected to Server. User is within a chat.

Postcondition: The message can no longer be viewed within the chat, but the record of it, as well as its removal, is kept in the chat log.

Basic Flow:

1. User selects their message/s to be hidden, and pushes to remove.
2. Message is now invisible to all users.
3. Log of this message being removed is kept and stored in that chat's log.

Alternate Flow:

1. The message/s they’ve selected is/are not their own.
2. System gives them a notification that they have selected message/s that are not their own, and does not remove it/them.

Exceptions:

1. No messages are selected.
2. The message/s they’ve selected is/are not their own.

Related Use Cases:

1. User Login
2. Log message

**Use Case ID: ITLogCheck1**

Use Case Name: IT member checks log

Relevant Requirements: IT User is allowed to view chat log history

Primary Actor: IT User

Preconditions: User is connected to server and is logged in, there has been chats and they

have a log

Postcondition: Unchanged (the logs have been viewed)

Basic Flow:

1. IT User logs into their account (IT account with special permissions).
2. IT User clicks on the logs button.
3. IT User finds the chat group they need to check and reads the log.

Alternate Flow:

1. IT User logs into their account (IT account with special permissions).
2. IT User clicks on the logs button.
3. There has not been any chats yet so the log is empty.

Exceptions:

1. Logs do not exist yet (no chats have been created)

Related Use Cases:

1. User messages another User
2. User Removes Message
3. IT Removes Chat

**Use Case ID: Create Channel**

Use Case Name: User Creates Channel

Relevant Requirements: User creates a new channel within the server

Primary Actor: User

Preconditions: User is connected to server and is logged in

Postcondition: A new channel is created that includes the selected users

Basic Flow:

1. User logs into their account.
2. User clicks the “+” icon in the channel section.
3. User selects the users included in the channel from a dropdown menu.
4. Channel is added, and user enters that channel.

Alternate Flow:

1. User clicks the “+” icon in the channel section.
2. User selects no users.
3. Window a prompts user to select.
4. User then selects users.

Exceptions:

1. There are no users in the database.

Related Use Cases:

1. User enters channel
2. User login

**Use Case ID: Enter Channel**

Use Case Name: User enters Channel

Relevant Requirements: User enters a channel within the server

Primary Actor: User

Preconditions: User is connected to server and is logged in, channel is created

Postcondition: User enters a channel to be able to interact with him

Basic Flow:

1. User logs into their account.
2. User clicks a channel in the channel section.
3. User enters that channel.

Alternate Flow:

1. N/A

Exceptions:

1. There are no channels in the database.
2. There are no channels that include the user.

Related Use Cases:

1. User creates channel
2. User login

**Use Case ID: Exit Channel**

Use Case Name: User exits Channel

Relevant Requirements: User exits a channel they are currently in

Primary Actor: User

Preconditions: User is connected to server and is logged in, and within a channel

Postcondition: User returns to main menu

Basic Flow:

1. User is within a channel
2. User clicks “return to main menu” button
3. User returns to main menu

Alternate Flow:

1. N/A

Exceptions:

1. There are no channels in the database.
2. There are no channels that include the user.

Related Use Cases:

1. User enters channel

**Use Case ID: Logout**

Use Case Name: User Logout

Relevant Requirements: User disconnects from server

Primary Actor: User

Preconditions: User is connected to server and is logged in, and not in a channel

Postcondition: User disconnects from the server and is brought back to login

Basic Flow:

1. User is at main menu
2. User clicks “disconnect from server” button
3. User returns to login

Alternate Flow:

1. N/A

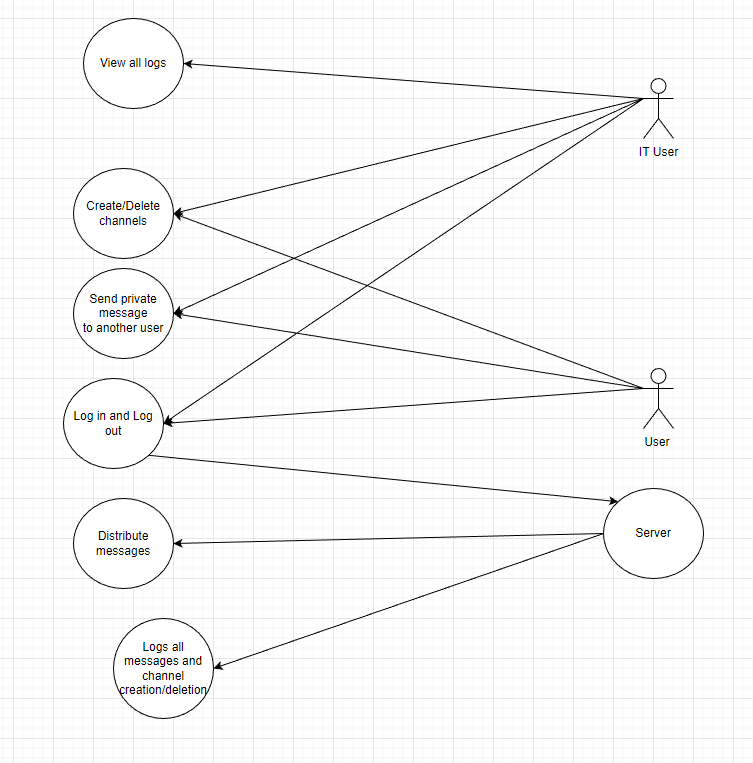
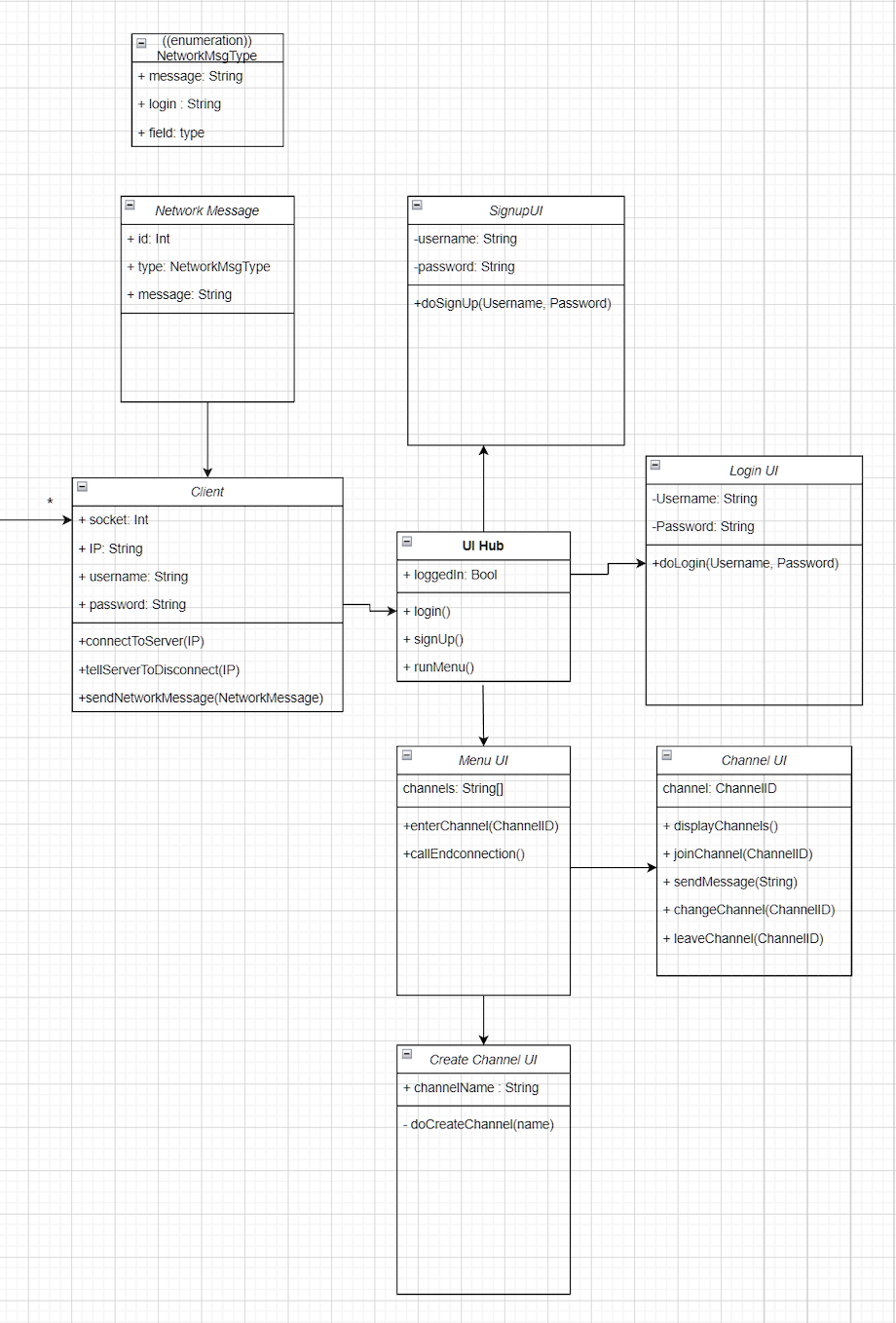
Exceptions:

1. N/A

Related Use Cases:

1. User login





Some use cases

## Overview

The communication system is designed to allow users to communicate with each other over the network using channels to separate discussions.

# Overall Description

## Product Perspective

The communication will be a Java based application hence it will require Java Runtime Environment. The server and client will run independently with clients’ ability to connect to the server for the communication system.

## Product Architecture

This system will be organized into server/client architecture. Server will hold all the messages. Server software will be organized into IT view, private messages, and public message channels. Client software will have a user interface throughout the whole program, a login, and logout system, and a way to send messages between the users.

The system will be organized into 3 major modules: the Client Module, the Server Module, and the Messaging Module. There will be additional modules for the Message Log module, IT User module(subclass of client that gives the IT users special privileges) and a module for chats.

Note: System architecture should follow standard OO design practices.

## Product Functionality/Features

The high-level features of the system are as follows

* + 1. Allow the user to connect to the server
    2. IT tracking of messages
    3. Hold separate conversation in channels
    4. Create channels and administer that channel
    5. Send private messages between two persons

## Constraints

Since the communication system including the client and server will be portable software, it must not use platform specific features. In addition, no databases, libraries, frameworks, or other technologies may be used without approval.

## Assumptions and Dependencies

It is assumed that two or more users will connect to the communication system at one time and communicate with each other in real time. There will be no external dependencies for this application. It is also assumed that the messages sent will only be text and not files or other media types.

# Specific Requirements

## Functional Requirements

### Common Requirements:

* + - 1. Users can have asynchronous/synchronous messaging between 2 or more users.
      2. IT and users login with a username (character length between 8-20 characters) and password (minimum character length 8; includes 1 capital letter; includes 1 special character).
      3. The program connects to a network via a Server and Client.

### User Module Requirements:

* + - 1. Users have the ability to login and logout of the server.
      2. Users can hide or modify their messages in a chat channel, but every published message is logged and remains logged even after modification.
      3. Requests to the server are made in the client module.
      4. Users have the ability to directly reply to specific messages in a chat channel.
      5. Users can create and name chat channels that include other users.
      6. Users are able to search in the list of current users, and add them to their chat channels.

### IT User Module Requirements:

* + - 1. IT users have all the same privileges as a user, but have additional special privileges.
      2. IT users have the ability to hide channels and messages(due to lack of use, abuse, or harassment), however the log of these messages and channels remains intact.
      3. IT users can view the logs of all chat channels.

### Server Module Requirements:

* + - 1. Server must distribute messages in channels to all users, there are no private channels.
      2. Server must be able to handle several clients at the same time.
    1. **Messaging Module Requirements:**
       1. Enables the ability to send the actual messages from the user to the chat.
       2. Creates the format for how the messages should show up on the chat: says who it is from and has a time stamp attached to it.

### Log Module Requirements:

* + - 1. Logs are created and updated for every channel of communication, and every message is recorded.
      2. The logs cannot be changed or deleted.
    1. **Chat Module Requirements:**
       1. The module that creates the different group chats.
       2. Holds the structure for the group messages.

### 

## External Interface Requirements

* + 1. The user must be able to log in and out of the communication system. The user must be able to send text messages.
    2. IT department must be able to view other users' messages including private messages

## Internal Interface Requirements

## The client and server must use TCP to communicate with each other.

## The message sent to the server must be a regular text string followed by the string length.

* + 1. The System must not use databases or libraries that were not previously approved.

# Non-Functional Requirements

## Security and Privacy Requirements

* + 1. There are no security requirements. However, privacy will be minimalized as the communication system will be tracked and monitored by the IT department.
    2. Privacy is also limited on the basis that there are no private channels. Every channel can be viewed by every client. The only limitation to channels are those that can contribute to them.

## Environmental Requirements

* + 1. Server and Client for the communication system must be run using the Java Runtime environment.

## Performance Requirements

## N/A