# Communication Application

*Software Requirements Specification*

**1**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 9/28/2025 | 1.0 | Initial Version | Group 7 |
| 10/31/2025 | 2.0 | Design Phase | Group 7 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 4
   1. [SCOPE 4](#_TOC_250013)
   2. [Definitions, Acronyms, Abbreviations 4](#_TOC_250012)
   3. [References 4](#_TOC_250011)
   4. [Overview 4](#_TOC_250010)
2. Overall Description 5

2.1

. Product Perspective 5

* 1. [Product Architecture 5](#_TOC_250009)
  2. [Product Functionality/Features 5](#_TOC_250008)
  3. [Constraints 5](#_TOC_250007)
  4. [Assumptions and Dependencies 5](#_TOC_250006)

1. Specific Requirements 6
   1. [Functional Requirements 6](#_TOC_250005)
   2. [External Interface Requirements 7](#_TOC_250004)
   3. [Internal Interface Requirements 7](#_TOC_250003)
2. Non-Functional Requirements 8
   1. [Security and Privacy Requirements 8](#_TOC_250002)
   2. [Environmental Requirements 8](#_TOC_250001)
   3. [Performance Requirements 8](#_TOC_250000)
3. **Purpose**

This document outlines the requirements for the Communication application

### Scope

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

### Definitions, Acronyms, Abbreviations

### References

Use Case Specification Document – refer to pg. 9 UML Use Case Diagrams Document – refer to pg. 12 Class Diagrams – refer to pg. 13

Sequence Diagrams – refer to pg. 14

### Overview

The application is a client-server messaging system designed to be scalable to a large number of users. The system supports both direct messaging between two users and group messaging with three or more participants. To ensure accountability and security, the application maintains immutable chat logs that record message content, sender and recipient IDs, and timestamps.

1. **Overall Description**

### Product Perspective

### The system will serve as a messaging application for a collection of employees and IT members.

### Product Architecture

The system will be organized into 2 major modules: the Client module and the Server module.

### Product Functionality/Features

* + - **User Messaging**: Users can send and receive text-based messages in real time through the client application.
    - **User Management**: The system supports account creation, login, logout, and session handling for secure access.
    - **Notifications**: The client can display message alerts and status notifications to keep users informed of new activity.
    - **Logging and Monitoring**: The server maintains logs of communication events, errors, and system status for monitoring and troubleshooting.
    - **External Interfaces**: The system connects over TCP/IP, ensuring compatibility across different platforms and network environments.
    - **Scalability and Reliability**: The architecture supports multiple concurrent clients and ensures reliable message delivery.
    - **Security**: User data and communications are protected through authentication and encrypted transmission.

### Constraints

The program should not include any databases, libraries, frameworks, or other technologies.

### Assumptions and Dependencies

It is assumed that the server is running when a user wants to connect.

The User’s device is dependent on having a Java Runtime Environment to run the Program. It is assumed that the User’s device has a network interface card to connect to the server.

1. **Specific Requirements**

### Functional Requirements

* + 1. **Common Requirements:**
       1. Privacy will be minimized in the system design where all administrators will have read access to all chat logs but must not be able to modify the logs.
       2. The system should support both synchronous and asynchronous communication.
       3. The application should be scalable to a large number of users with little or no difference in runtime.
       4. All users should be able to interact with the application through a GUI implementation.

### User Module Requirements:

* + - 1. Users should be allowed to log in using their issued ID and PIN, both of which are alphanumeric strings between 6 and 20 characters in length.
      2. IT/Admins should be able to log in through the same system.
      3. IT users should be able to create new users, picking a unique username, password, and ID.
      4. Users should be notified of new messages.
      5. Users will be assigned roles (IT or general user) upon successful login using credentials provided by IT

### Server Module Requirements:

* + - 1. Chat logs should be immutable and cannot be deleted.
      2. Chat logs should be accessible asynchronously.
      3. Chat logs should only be accessible by users with IT/Admin privileges.

### Messages Module Requirements:

* + - 1. Messages should only contain text supported by the UTF-8 standard.
      2. Once added to a chat, users should not be able to leave the conversation.
      3. Users should be able to exchange direct messages between two users or participate in group chats with three or more users.
      4. Each message should be logged with a timestamp, sender ID, recipient ID(s), and message content.
      5. Messages should be limited to a maximum of 1000 characters.

### External Interface Requirements

3.2.1 The communication should happen over TCP/IP.

### Internal Interface Requirements

3.3.1 Error handling and retry mechanisms should be defined to prevent message loss or corruption.

1. **Non-Functional Requirements**

### Security and Privacy Requirements

Example:

* + 1. The System must encrypt data being transmitted over the Internet.

### Environmental Requirements

Example:

* + 1. System cannot require that any software other than a web browser be installed on user computers.
    2. System must make use of the University’s existing Oracle 9i implementation for its database.
    3. System must be deployed on existing Linux-based server infrastructure.

### Performance Requirements

Example:

* + 1. System must render all UI pages in no more than 9 seconds for dynamic pages. Static pages (HTML-only) must be rendered in less than 3 seconds.

**Use Case Specification**

## Use Case ID: UC-01

**Use Case Name:** Login

**Relevant Requirements:** Device is connected to wifi, and has typing/mouse capabilities. Minimal hardware requirements.

**Primary Actor:** General user (does not vary per level of staff) or IT

**Preconditions:** User account exists; system is online

**Postconditions:** User authenticated; role assigned (IT/regular user); chat dashboard displayed

#### Basic Flow:

1. User enters credentials.
2. System verifies credentials.
3. System assigns role (IT or regular user).
4. Dashboard is displayed. If IT role, additional abilities are given in addition. **Extensions/Alternate Flows:** Invalid credentials → error/warning shown; device lockout after 3 failed attempts. Can be manually allowed to login via IT, or accounts can be locked.

## Use Case ID: UC-02

**Use Case Name:** Viewing logs

**Relevant Requirements:** User was properly assigned IT role and access in UC-01

**Primary Actor:** IT, System

**Preconditions:** IT user exists and properly logged into system that is properly connected to device **Postconditions:** IT user gets *TEXT or System.out.println()?* Of chosen user’s logs, either with another user or in general?

**Basic Flow:** The user IT types into a prompt is chosen, system accesses the logs of the user, and then prints/displays logged chats.

1. The user IT types into a prompt is chosen
2. System accesses chat logs of user
3. Computer prints accessed chat logs

**Extensions/Alternate Flows:**

## Use Case ID: UC-03

**Use Case Name:** Create group

**Relevant Requirements:** User properly logged in and given user role, and has typing/mouse capabilities

**Primary Actor:** User, including IT

**Preconditions:** Properly able to interact with interface, and has other user ID’s of other group chat members

**Postconditions:** a chat system that includes and sends/receives messages to all members of the group chat

#### Basic Flow:

1. Click “create group” button
2. Types multiple users to add to group chat and click “create”
3. Group chat is properly initialized

**Extensions/Alternate Flows:** Invalid user→ error/user ID not found

## Use Case ID: UC-04

**Use Case Name:** Create new user

**Relevant Requirements:** IT user, and has enough info on new user to create them

#### Primary Actor: IT

**Preconditions:** Ready user id/password

**Postconditions:** New user with the given user ID/password, and potentially any metadata and whether they are also IT

#### Basic Flow:

1. IT creates user with information given
2. User is created along with metadata and roles given. 3.
3. ​

**Extensions/Alternate Flows:** Existing user credentials already, does not create a new user and warns IT that attempted creating user.

## Use Case ID: UC-05

**Use Case Name:** Send Message

**Relevant Requirements:** Only IT can create users. System supports role assignment and password.

**Primary Actor:** User, including IT

**Preconditions:** Sender logged in; recipient (user or group) resolvable. Sender has permission to message recipient.

**Postconditions:** Message persisted and timestamped. Delivery enqueued to recipient(s); sender sees “sent” status. Event available to logging subsystem (UC-07).

#### Basic Flow:

1. User types message and sends
2. Message is sent through system to targetted users and logged in UC-07

**Extensions/Alternate Flows:** Network/server error → “failed to send”, retry option

## Use Case ID: UC-06

**Use Case Name:** Receive message

**Relevant Requirements:** Real-time delivery (websocket/push). User is online or offline with queued delivery.

**Primary Actor:** User or IT

**Preconditions:** Recipient account exists and is addressable. Message previously sent (UC-05) and accepted by system.

**Postconditions:** Message appears in recipient’s chat view. Delivery/read receipts updated as applicable. Notification triggered (toast/badge/sound) per user settings.

#### Basic Flow:

1. Message is sent to server, and sent to recipient user
2. Recipient user receives notification that a new message from sender user ID
3. Recipient can read message

**Extensions/Alternate Flows:** Log store unavailable warning of system issue to user; retry later and alert IT. Storage near limit, throttle non-critical logs and alert IT.

## Use Case ID: UC-07

**Use Case Name:** Log message

**Relevant Requirements:** Device that UC-05 is connected to system and internet

**Primary Actor:** System, user (IT included)

**Preconditions:** A message is sent/received (UC-05/UC-06). Logging feature enabled and reachable. **Postconditions:** Message metadata written to log store with timestamps and IDs. Logs stored in an array for search (UC-02). Failures are captured and alerted to IT.

#### Basic Flow:

1. Message successfully sent and received, and sent to be logged
2. Message is appended to array to be later retrieved.

**Extensions/Alternate Flows:** No memory available, IT warned and future messaging suspended

## Use Case ID: UC-08

**Use Case Name:** Create GUI

**Relevant Requirements:** Used device has the Swing/JavaFX libraries in its java version installed

**Primary Actor:** Software, upon successful UC-01, with extra if given role is IT

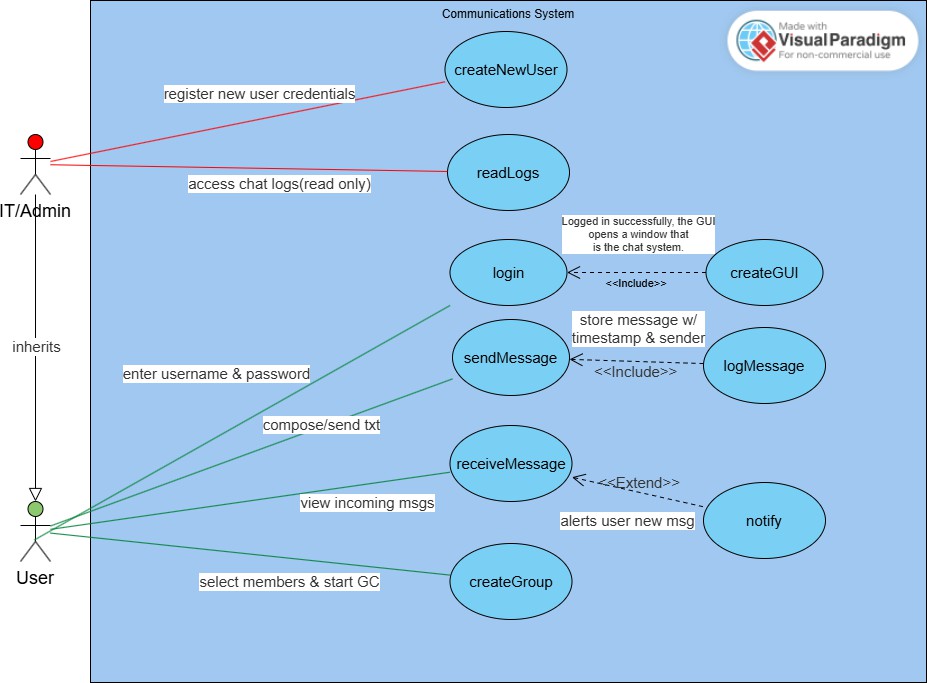
**Preconditions:** UC-01 properly logged in, and software properly starts up.

**Postconditions:** Interface to chat properly begins, and interface for IT is also given, to pull up a user’s cha

#### Basic Flow:

1. Logged in successfully, the GUI opens a window that is the chat system.
2. Depending on IT or not, allowed additional features for chat log access and user creation

**Use Case Diagrams**



**Class Diagrams**

A diagram of a computer flowchart

AI-generated content may be incorrect.

**Sequence Diagrams**

A diagram of a computer program

AI-generated content may be incorrect.A diagram of a chat system

AI-generated content may be incorrect.