Communication Project

Group 4

Anh Truong

Nicholas Bui

Shih Tung Yapp

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 06/22/2022 | 1.0 | Initial Version | Anh Truong, Nicholas Bui, Shih Tung Yapp |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 3
   1. Scope 3
   2. Definitions, Acronyms, Abbreviations 4
   3. References 4
   4. Overview 6
2. Overall Description 6
   1. Product Perspective 6
   2. Product Architecture 6
   3. Product Functionality/Features 6
   4. Constraints 6
   5. Assumptions and Dependencies 6
3. Specific Requirements 7
   1. Functional Requirements 7
      1. Common Requirements: 7
      2. \_\_\_\_\_ Module Requirements: 7
      3. \_\_\_\_\_ Module Requirements: 7
      4. \_\_\_\_\_ Module Requirements: 7
   2. External Interface Requirements 7
   3. Internal Interface Requirements 8
4. Non-Functional Requirements 8
   1. Security and Privacy Requirements 8
   2. Environmental Requirements 8
   3. Performance Requirements 8

# Purpose

This document outlines the requirements for the Communication System.

## Scope

Create a communications system for a very large organization. This

system should allow employees to communicate over chat both

synchronously and asynchronously. Users should be able to chat

privately and in groups. All conversations should be logged and

viewable by the IT users. Privacy should be minimized. Only text is

required at this time..

## Definitions, Acronyms, Abbreviations

* Communication System- Client software to service communication within the company.
* Communications Server- Server to exchange data between users and store logs.
* System User interface: Register, Login, Recover, Delete Account
* Contacts- List of contacts
* Groups- Form groups from contacts to perform group messaging.
* Logs- View message history
* Inbox-  send a private message to another users
* Outbox- Delete the message or try to fix the issue causing the message not to be send.
* Status- A live/or instant updates, to let other users know that this users are on app.
* Read Receipts- messages/notifications showing that “received” users has received the message others sent.
* Search- Search tool used to locate messages in the log with a keyword.
* Delete- Delete messages in the user’s chat log
* Users - Entry level access to standard employees.
* Administrators - Access to system controls to manage users, view logs, and perform system maintenance.
* Use case starting with A indicates user accessible commands while use starting case with Z indicates commands reserve for administrator only.

## References

Use Case ID: A-01

Use Case Name: Users can view messages from others

Requirements: Other users sent messages

Primary Actor: Other users, server

Pre-conditions: Users should be active, and ready to view the message.

Post-Conditions: The other users should send the message, without deleting the messages. The object has been exchanged from the server and sent it back to the user

Basic Flow: 1. The user initiates an action to open the application, and systems responded by showing the messages from other users.

Use Case ID: A-02

Use Case Name: Users can send messages to others

Requirements: Users status was on

Primary Actor: users, server

Pre-conditions: Users should be active, and ready to send the message.

Post-Conditions: The user should send the message, without deleting the messages. The object was exchanged from the server and sent to the other users.

Basic Flow: 1. The user initiates an action to open the application, 2. Then send the message to other users, 3. and systems responded by showing the messages from other users.

Use Case ID: Z-01

Use Case Name: The administrator can create the chat group

Requirements: Admin Role

Primary Actor: Administrators, servers, users.

Pre-conditions: Admin should be active, and ready to create groups. Admin should have an Admin Role.

Post-Conditions: When the administrator has the admin role, he can add users to create a group chat, data can be exchanged from the server, then send notifications to users.

Basic Flow: 1. The administrator initiates an action to open the application, 2. Then add users to create groups, 3. and systems responded by put users in the same group.

Use Case ID: Z-02

Use Case Name: The administrator can remove a user from the chat group the chat group

Requirements: Admin Role

Primary Actor: Administrators, servers, users.

Pre-conditions: Admin should be active. Admin should have an Admin Role. There is a group.

Post-Conditions: When the administrator has the admin role, he can remove users from a group chat, data can be exchanged from the server, then send notifications to users.

Basic Flow: 1. The administrator initiates an action to open the application, 2. Then remove users to create groups, 3. and systems responded by removing users from that group.

Use Case ID: A-03

Use Case Name: The user can receive a notification from the system.

Requirements: User, System

Primary Actor: Servers, users.

Pre-conditions: Other users send a message to the user.

Post-Conditions:  The other users should send the message, without deleting the messages. The object has been exchanged from the server and sent it back to the user. Then the user can receive the message notification.

Basic Flow: 1. The user initiates an action to open the application, and 2. systems responded by showing the messages from other users. 3. User receives the message notification.

Use Case ID: A-04

Use Case Name: The user can delete contacts

Requirements: User, System

Primary Actor: Servers, users.

Pre-conditions: The user has sent some messages before.

Post-Conditions:  The user deletes the contacts

Basic Flow: 1. The user deletes the contacts, and 2. systems responded by removing contacts from users.

Use Case ID: A-05

Use Case Name: The user can search contacts

Requirements: User, System

Primary Actor: Servers, users.

Pre-conditions: The user has sent some messages before.

Post-Conditions:  The user searches the contacts, there will be a list that shows up

Basic Flow: 1. The user deletes the contacts, and 2. systems responded by a list showing up.

For use case and class UML diagram, refer to below.

## Overview

Communication systems allows user to communicate with one another through chat.

# Overall Description

## Product Perspective

## Product Architecture

The system will be organized into \_\_\_ major modules: the \_\_\_ module, the \_\_\_ module, and the \_\_\_\_\_ module.

Note: System architecture should follow standard OO design practices.

## 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):

## Constraints

List appropriate constraints.

Constraint example: Since users may use any web browser to access the system, no browser-specific code is to be used in the system.

## Assumptions and Dependencies

List appropriate assumptions

Assumption Example: It is assumed that the maximum number of users at a given time is 15,000.

# Specific Requirements

## Functional Requirements

### Common Requirements:

3.1.1.1 Every user will have a user id and password.

3.1.1.2 All users can send and receive message.

3.1.1.3 Notification system for unread message.

3.1.1.4 Read receipts for messages.

3.1.1.5 Administrator has the ability to add and remove user from a group.

3.1.1.6 All users has the ability to search logs according to specific keyword.

3.1.1.7 Administrator can read logs of all users.

3.1.1.8 Status information such as online, offline, busy and so on.

3.1.1.9 Users can edit message or delete message within a certain time frame.

3.1.1.10 Users can add or delete contacts.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 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.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 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.

### \_\_\_\_\_ Module Requirements:

Provide module specific requirements as appropriate.

Example:

3.1.2.1 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.

## External Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.2.1 The system must provide an interface to the University billing system administered by the Bursar’s office so that students can be automatically billed for the courses in which they have enrolled. The interface is to be in a comma-separated text file containing the following fields: student id, course id, term id, action. Where “action” is whether the student has added or dropped the course. The file will be exported nightly and will contain new transactions only.

## Internal Interface Requirements

Provide module specific requirements as appropriate.

Example:

3.3.1 The system must process a data-feed from the grading system such that student grades are stored along with the historical student course enrolments. Data feed will be in the form of a comma-separated interface file that is exported from the grading system nightly.

3.3.2 The system must process a data-feed from the University billing system that contains new student records. The feed will be in the form of a comma-separated text file and will be exported from the billing system nightly with new student records. The fields included in the file are student name, student id, and student pin number.

# Non-Functional Requirements

## Security and Privacy Requirements

Example:

4.1.1 The System must encrypt data being transmitted over the Internet.

## Environmental Requirements

Example:

4.2.1 System cannot require that any software other than a web browser be installed on user computers.

4.2.2 System must make use of the University’s existing Oracle 9i implementation for its database.

4.2.3 System must be deployed on existing Linux-based server infrastructure.

## Performance Requirements

Example:

4.3.1 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.

Team Github Repo: https://github.com/mingzyapp/CS401---Project/tree/main



