Software Requirements Specification

Revision History

| **Date** | **Revision** | **Description** | **Author** |
| --- | --- | --- | --- |
| mm/dd/yyyy | 1.0 | Initial Version | Your Name |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Purpose 4**](#_30j0zll)

[1.1. Scope 4](#_1fob9te)

[1.2. Definitions, Acronyms, Abbreviations 4](#_3znysh7)

[1.3. References 4](#_2et92p0)

[1.4. Overview 4](#_tyjcwt)

[**2. Overall Description 5**](#_3dy6vkm)

[2.1. Product Perspective 5](#_1t3h5sf)

[2.2. Product Architecture 5](#_4d34og8)

[2.3. Product Functionality/Features 5](#_2s8eyo1)

[2.4. Constraints 5](#_17dp8vu)

[2.5. Assumptions and Dependencies 5](#_26in1rg)

[**3. Specific Requirements 6**](#_lnxbz9)

[3.1. Functional Requirements 6](#_35nkun2)

[3.1.1. Common Requirements: 6](#_1ksv4uv)

[3.1.2. Direct Message Module Requirements: 6](#_z337ya)

[3.1.3. Group Chat Module Requirements: 7](#_1y810tw)

[3.1.4. Admin Module Requirements: 7](#_2xcytpi)

[3.2. External Interface Requirements 7](#_3whwml4)

[3.3. Internal Interface Requirements 8](#_qsh70q)

[**4. Non-Functional Requirements 9**](#_49x2ik5)

[4.1. Security and Privacy Requirements 9](#_2p2csry)

[4.2. Environmental Requirements 9](#_3o7alnk)

[4.3. Performance Requirements 9](#_1hmsyys)

# Purpose

This document outlines the requirements for the Communication System.

## 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.

## Definitions, Acronyms, Abbreviations

List any acronyms, terms etc. that need to be defined.  
 Define synchronous and asynchronous.  
 Define IT user and normal user.  
 Define Group Administrator in the group chat.  
 Define online status (online or offline) and availability status (idle or busy).

## References

Use Case Specification Document – Step 2 in assignment description

UML Use Case Diagrams Document – Step 3 in assignment description

Class Diagrams – Step 5 in assignment description

Sequence Diagrams – Step 6 in assignment description

## Overview

The Mine Pump Control System (MPC), is designed to monitor and pump flood water out of mine shafts. As underground mining operations take place far below the water table, flooding into mine galleries and shafts is an ever-present danger.

# 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:

Provide requirements that apply to all components as appropriate.

Example:

3.1.1.1 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.

3.1.1.2 The system should provide HTML-based help pages on each screen that describe the purpose of each function within the system.

3.1.1.1 All messages in the system, including direct messages, group chats, and blocked messages, should be logged in the server.

3.1.1.2 Users should be able to view all messages history that are stored in the local machine.

3.1.1.3 Users should be able to delete messages, including direct messages and group chats, such that deleted messages will not show up in the history but the messages logged in the server will not be deleted.

3.1.1.4 Users should be able to receive messages that are sent synchronously.

3.1.1.5 Users should be able to receive messages that are sent asynchronously.

3.1.1.6 Users should be able to send messages to other users (if they are not blocked by other users) regardless of whether other users are online or offline.

3.1.1.7 Users should be able to pin messages in both direct messages and group chats, allowing them to quickly access important messages without scrolling through the entire chat history.

3.1.1.8 Users should be able to unpin messages in both direct messages and group chats.

3.1.1.9 Users should be able to mute direct message conversations and group chats, in a way that they will receive the messages but not get notifications.

3.1.1.10 Users should be able to unmute direct message conversations and group chats if they muted them before, and start getting message notifications whenever they receive a new message from those conversations.

3.1.1.11 Users should be able to view the directory of all users in the system.

3.1.1.12 Users should be able to view other users’ online statuses (online or offline) and availability status (idle or busy).

3.1.1.13 Users should be able to change their availability statuses (idle or busy) whenever they want.

3.1.1.14 Users’ online statuses are only based on whether they are logged in the system or not and cannot be changed by users.

### Direct Message Module Requirements:

3.1.2.1 Users should be able to initiate a new conversation with any employees in the system.

3.1.2.2 Users should be able to block other users such that they will not be able to receive any messages from those blocked users.

3.1.2.3 Users should be able to unblock other users if they block them before and start receiving messages from them sent after the unblock event.

### Group Chat Module Requirements:

3.1.3.1 Users should be able to accept invitations to join group chats.

3.1.3.2 Users should be able to send requests to join group chats.

3.1.3.3 Users should be able to create new group chats and they will automatically become the Group Administrators of the new group chats.

3.3.3.4 Users who are members of the group chat should be able to invite other users to join.

3.3.3.5 Only users who are Group Administrators of the group chat can decide whether to let certain users join.

3.3.3.6 Users who are Group Administrators of the group chat should be able to invite other users to be the Group Administrators of the same group.

3.3.3.7 Only users who are Group Administrators of the group chat can remove members from the group chat.

3.3.3.8 Only users who are Group Administrators of the group chat can pin messages in the group chat that are visible to all members of the group.

3.3.3.9 Only users who are Group Administrators of the group chat can unpin messages that are pinned in the group chat before.

3.3.3.9 Users should be able to leave any group chats if they want.

### Admin 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.

3.1.4.1 IT users (admins) should be able to view all the logged messages stored in the server.

3.1.4.2 IT admins should be able to create new users.

3.1.4.3 IT admins should be able to delete existing users and automatically remove them from all the group chats they belong to. Any attempts to send new messages to these deleted users will be rejected and those users will receive an error message.

## 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.

3.2.1 The system must provide an interface for the client (the users).

3.2.2 The client application will save all the received and sent messages locally before exiting.

3.2.3 The server application will save all the logged files and user information locally before and can restore after restart.

## 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.

Server <-> application

3.3.1 The client application should be able to write new messages to the local file and retrieve chat history from it.

3.3.2 The server application should be able to communicate with the database and authenticate users’ login credentials.

3.3.3 The server application should be able to communicate with the client application for receiving and sending messages, user login and logout.

# Non-Functional Requirements

## Security and Privacy Requirements

4.1.1 The system must implement strong password policies for user authentication.

4.1.2 The system only allows one account to log in from one device at a time. Any requests to log in the same account from different devices will be rejected.

4.1.3 The system must maintain comprehensive logs for all user activities, ensuring traceability for security and auditing purposes.

4.1.4 Only IT users can access the system logs.

## 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.

4.2.1 Both server and user applications should run on a computer with Windows, MacOS, or Linux, with at least 4GB of RAM and enough hard drive space to store messages locally.

4.2.2 Both server and user applications require JRE 20 or above.

4.2.3 Both server and user applications are required to connect to the Internet with minimum an average of 1Mbps (Down/Up).

## Performance Requirements

4.3.1 The system must respond to user requests within 1 second for real-time chat interactions.

4.3.2 The system should not have unnecessary limits for the number of concurrent users online.