

Internal Chat System

Software Requirements Specification

Revision History

Date	Revision	Description	Author
02/20/2022	1.0	Initial Version	Nadeem Mohammad

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

1. Purpose

This document outlines the requirements for the Internal Chat System.

1.1. Scope

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

1.2. Definitions, Acronyms, Abbreviations

Chat rooms are used synonymously with chat logs.

Users include Employees and System Admins (A special subset of employees)

1.3. References

Use Case Specification Document

UML Use Case Diagrams Document

Class Diagrams

1.4. Overview

The Internal Chat System is designed to handle messages between employees in a corporate environment. It will allow for group messaging, and it will keep a log of all the messages that get sent through the system.

2. Overall Description

2.1. Product Perspective

2.2. Product Architecture

The system will be organized into 2 major modules: the Server module, the Client module. Some minor modules that are in the server module, will be the Chatlog, User, and Messages Modules.

Note: System architecture should follow standard OO design practices.

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):

2.4. Constraints

To prevent server overload and spamming, there is a cap on the number of messages a user can send in a row in a given interval of time (no more than 1 message/second).

2.5. Assumptions and Dependencies

We'll assume that the user has access to the internet, since that is how they will connect to the server. The user needs a computer system with one of the following operating systems: Windows, Mac, or Linux.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Common Requirements:

3.1.1.1 Users should be allowed to log in using their issued username and password, both of which are alphanumeric strings between 6 and 20 characters in length.

3.1.2. Messages Module Requirements:

3.1.2.1 Every message has a content, sender, receiver, and a status.

3.1.2.2 When a message is sent to the server, it has a delivered status.

3.1.2.3 When a message is received, it stores the users who have read that message in order to display it to the rest of the users.

3.1.3. Chatlog Module Requirements:

3.1.3.1 A chat log logs all the messages sent by a user to another user or a group of users.

3.1.3.2 The server maintains a list of multiple chat logs -all of which can be accessed by the system admin

3.1.3.3 Every new message or creation of a chat log is sent to the server

3.1.4. User Module Requirements:

3.1.4.1 The System Admin can create new users (another employee or system admin), or delete an existing user.

3.1.4.2 Users have a username and password, as well as an identifier indicating whether they're currently logged in.

3.1.4.3 Every user has their own list of chat logs showing all the chat rooms they're in and the messages they have received. This list of chat logs is updated with the list of chat logs in the server when a user logs in (to make sure they receive their new messages).

3.1.4.4 Only one user with the same username/password can be logged in at the same time. If another user tries to log in with the same credentials, it will log out the previous user.

3.1.5. Client Module Requirements:

3.1.5.1 The client will handle the GUI that the user sees. It will provide the login page, where the users will login. After logging in, the users will see all the employees that are currently online. The user will also be able to see all the other employees that they messaged.

3.1.5.2 The client will handle messages/interactions from the User, and send them to the server.

3.1.5.3. The client will also be able to take information from the server, and display it to the user.

3.1.6. Server Module Requirements:

3.1.6.1 The server will be responsible for connecting to multiple users, at least up to 10,000.

3.1.6.2 It will send data to Client, where the Client will then display the data

3.1.6.3 The server will get data from the client, where it will process, and organize the data.

3.2. External Interface Requirements

3.2.1 The system must provide an interface that is simple and easy to use for users.

3.2.2 The system must provide an interface for the users to log in, see all their chat logs, and enter a chat log to view all it's messages.

3.2.3 The system must provide an interface for the users to create new chat logs.

3.2.4 The system must provide an interface for the system admin to create new users and delete existing ones.

3.3. Internal Interface Requirements

3.3.1 The system must track whether a message has been read by a user, and the meta data needs to be updated in the server.

3.3.2 The system must check messages in the client side and see it's up to date with messages on the server. If not, it must update it.

4. Non-Functional Requirements

4.1. Security and Privacy Requirements

4.1.1 Passwords should be 6-20 characters long in order to prevent unauthorized logins.

4.1.2 Login information will be created by an admin, which will be given to an employee, to ensure their username/passwords are not easily guessed.

4.1.3 Application must be written in Java.

4.1.4 The Application must be done by 5/2/2022

4.2. Environmental Requirements

4.2.1 System uses Java and JDK

4.3. Performance Requirements

4.3.1 System must be scalable up to 10,000 users without chat latency issues.