Software Requirements Specification

for

CS401 Final Project

Version 1.0

Prepared by

Group Name:

|  |  |  |
| --- | --- | --- |
| Luis Haro | Lhr004@gmail.com |  |
|  |  |  |

|  |  |
| --- | --- |
| Instructor: | Christopher Smith |
| Course: | CS 401 |
| Date: | 4/23/2020 |

Contents

Contents ii

Revisions ii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 1

2 Overall Description 2

2.1 Product Overview 2

2.2 Product Functionality 2

2.3 Design and Implementation Constraints 2

2.4 Assumptions and Dependencies 3

3 Specific Requirements 4

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 5

4 Other Non-functional Requirements 6

4.1 Performance Requirements 6

4.2 Safety and Security Requirements 6

4.3 Software Quality Attributes 6

5 Other Requirements 6

Appendix A – Data Dictionary 7

Appendix B - Group Log 7

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 

# Introduction

## Document Purpose

This document will focus and, on the scope, requirements and specifications of the final project. The document will inform us of the principals of the project so students or instructor can have an idea of the whole project; this way, the members of the group can follow the requirement or make changes to it, in the same way other students can have a picture of the project and take over if it is necessary.

## Product Scope

This software project will create a new chat program based on Java, users can use this program to create room, login, create new accounts, send messages to the room as public or private.

## Intended Audience and Document Overview

This document is intended to be use for developers as a guide for requirements as that develop the software; in addition, it’s intended to inform users of what they can do with this new software. It can also be used as a guideline for students whom might have to work in this project in the future.

## Definitions, Acronyms and Abbreviations

1. Client ‚ The software that runs on the end of the user machine, allowing them to connect to a chat room and send messages as well as receive messages.

2. Server, The software that runs on a remote machine and allows clients to communicate with each other in chat rooms and send direct/private messages between two people only.

3. Database ‚The software the stores information about users, chat rooms, and message histories.

4. User ‚ A representation of an end user within the application, including a username for identification in chat rooms and a password for authentication.

5. Message ‚ A representation of a text-based communication within the application, including a text body, a timestamp, and a chat room that the message will be a part of.

6. Chatroom ‚ A collection of Users who can send and receive messages that will be sent to all other Users that are in the same collection (room).

7. Direct Message (DM) ‚ A special chatroom that contains only two Users who can communicate by sending direct/private messages between each other.

## Document Conventions

Use Arial font size 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles use size 14 Arial bold.

## References and Acknowledgments

Most of the information in this document can be found in out GitHub page:

**https://github.com/luisHaroDev/FinalProject-2.git**

Template of this document can be found at CS.GMU.EDU

# Overall Description

## Product Overview



People want to communicate with each other since they are not able to see each other in person due to work or other reasons. Other people simply want to make new friends. The best way to communicate and make friends is through a chat application that allows them to interact with many people or just one person directly.

Our goal is to build a free, easy, and private chat room application that allows people to communicate with one another or a group of people in a chat room.

This is a group project for CS401 Software Engineering. We have been tasked with a semester-long software engineering project that requires us to implement a substantial piece of software using engineering principles and techniques that we have learned in class. We have selected a chat room application running over a network in a client-server model because it is a useful tool demonstrating network programming, information security, and object-oriented design.

## Product Functionality

This project will be a simple chat room / messaging system using a client-server model. The project will allow users on the client side to create an account, using a username and password, join a chat room, and send messages in that chat room.

The server will maintain each chat room and forward messages from clients in a chat room to all other clients in the same chat room. The server will use a database to maintain user accounts, chat rooms, and chat room message histories.

The client application will present controls to create an account, log in, select a chat room to chat with other people, view the chat room's message history once you have entered that specific chat room, and send messages. Also, be able to send private messages between two users. This chat will make a smoother communication for people. Broadcast general notifications, such as user logins and logouts.

## Design and Implementation Constraints

We will be using and array of classes to store data about users, rooms, messages, receiver and senders of messages and time. Assumptions and Dependencies

We will assume that our database is secure and not on site.

# Specific Requirements

## External Interface Requirements

### User Interfaces

This software will be user friendly and will use GUI interface to interact will the user

### Hardware Interfaces

The only hardware interface will be a computer and IDE with the available to run Java programs such as Eclipse, NetBeans, etc. and also a terminal

### Software Interfaces

Our programs will use swing as interface.

## Functional Requirements

#### Client

1. Connect to the server over a network automatically when it first gets launched
2. Allow the user to create a new account with a username and password
3. Allow the user to log in with a username and password
4. Get list of available chatrooms from server and display in the GUI
5. Allow a logged in user to select and join an available chatroom from the list
6. Get the chat log and users list for the specific chatroom and display it on screen when user selects a chatroom
7. Allow the user to send text messages to the chatroom
8. Receive messages from the server and asynchronously update the chat log displayed on screen to show the new messages
9. Allow the user to send text messages to other users in the chatroom by selecting them from the user list (Direct Message)
10. Display a system notification when:

* A user joins the chat room
* A user leaves the chat room
* The user receives a direct message

1. Allow the user to leave the chatroom and display the list of other available chatrooms to join
2. Allow the user to log out and exit the application, closing the server connection

#### Server

1. Allow connections from different clients over a network
2. Maintain client connections until the client decides to exit and close the connection
3. Connect to a database on the local system
4. Allow connected clients to create new accounts
5. Allow connected clients to log in using their proper usernames and passwords
6. Send list of available chatrooms to clients when it gets requested by the client
7. Manage users in each chatroom, adding and removing as needed based on client connection status
8. Broadcast messages from users in chatrooms to all other users in the same chatroom
9. Maintain a chat log for each chatroom and store the information in the database
10. Allow system administrator to create and delete chatrooms via a command line interface

#### Database

1. Store user account information –username, and encrypted password
2. Store chatroom message history for each chatroom

## Use Case Model

User run program.

Click create new account, enter username and password, this is saved in database.

Click create a new chatroom, user enter a name for the new chat room.

User type a string into a string box and click send

Message appeared to everyone in the chat to see.

User click log-off and the program is closed.

User run program

Enter credentials and click log-in

Notification of new messages will be display.

User click on existing chat room

User type a message and click send

Message appears on chat room window

User log-out

User A run program

Enter credentials and click log-in

Notification of new messages will be display.

User A click on existing chat room

User A type a message and select a Private and User B.

Click send, private message is received to the User B

User B click replay, to send a private message back

User A log-out

# Other Non-functional Requirements

## Performance Requirements

Our software will try to use our data structure as efficient as possible. We will delete user and add new one in the array in the same index to save memory. time complexity for search will be keep simple since it is a same data file.

## Safety and Security Requirements

encrypted password will be store in the database

## encrypted password Software Quality Attributes

The program should be easy to use and the code easy to read in order to allow team members to understand, it should have comments have explain variable, methods and attributes so other student can follow if it is necessary.

# Other Requirements

Appendix A- Group Log