<Chat >

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <03/Apr/19> | <1.0> | Add first supplementary specifications | Ciontu Mihail & Pantea Paul |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

Supplementary Specification

# Introduction

This document contains information that was not entered in the use case diagram. The project is a windows chat application with mention that the application can be a web application as well if we change the technology in which is developed. For develop this application was required knowledge in Java , dada Structure and Fundamental Algorithms.

Login Menu function This functional requirement is for prompting the user with the option to register for the chat application, logging in, or exit the program. It will take the form of a GUI

Register function This aspect of the login menu will ask the user for the username, and password of the client. It will check if the username has been taken and will close if the username is not taken and will go back to the main login menu.

Login function This aspect will ask for the username and password. Errors will occur if a space is left blank, the username doesn’t exist, or the password doesn’t match with the username. If the username and password matches, you are online and able to message anyone else online.

Exit This aspect will close the chat application.

# Non-functional Requirements

These are the nonfunctional requirements of the chat application. This is basically the section that deals with the quality of the chat application rather than the functionalities of the application. User Friendly The chat application needs to be user friendly, when using its user interface. GUI By using GUI’s, it should make the application more user friendly and better to use instead of a command line. Buttons will be used.

## Availability

The system must be available all the time, for both types of user. The application will be always running, being available to everyone who has internet access and a device that supports a web page in browser.

## Performance

The application will be simple and fast, the client can enter and select the person with he want to chat, create a group of friends or modify his profile.

## Security

System integrity or security is sufficient to prevent unauthorized access to system functions, preventing information loss and protecting the privacy of data entered into the system through authorization and authentication. Every method throw an exception if the user is not authorized.

## Testability

Test the program and find any bugs or glitches in the chat application. Testing will be done as the application is being developed, but the more developed it gets, the more testing will be done.

## Usability

This can be measured in terms of ease of use. Application is user friendly, easy to learn. Navigation is simple and suggestive.

# Design Constraints

The application is implemented using Spring Boot, a framework for Java, with a Hibernate connection to the database. The layered architectural pattern with Model-View-Controller is used for separating the concerns.