Online Library

Supplementary Specification

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

The **Supplementary Specification** captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

Legal and regulatory requirements, including application standards.

Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.

Other requirements such as operating systems and environments, compatibility requirements, and design constraints.

# Non-functional Requirements

## Availability

This is the proportion of time the system is in a functioning condition. Since the application is protected from database flooding attempts by the administrators or the anti-spam system and considering that the server’s processing power and storage size is constantly increasing as more users come, the system is 100% reliable, and the only factors which could affect this in the future would be the lack of funds for maintaining the system and upgrading the hardware(since this is a free application, funds come from donations or from users which want to publish more than their allocated amount of material).

## Performance

The response time of this application determines how fast the users access a book (search + open), post a review or upload a book. It is not a key factor in this application, but it should not be ignored. The response time is given by the speed/amount of processing units dedicated to the project and it should be kept under 5 seconds. Since this is a application which expects few users (not more than a hundred), this shouldn’t be difficult to achieve.

## Security

The online library only contains free books; no users are allowed to sell or buy material. Because of this, the security of the user accounts isn’t a top priority. Another factor to consider is the security of the database from overflowing attempts. This is guaranteed by the approval system where no book is published to the application without the administrator’s consent. This is a good security measure for a small size application like this, but in the future could become an overwhelming task. To solve that, an anti-spam system could be developed.

## Testability

Local network testing can be done in this application, because of the simple architecture(client-server). One person can be the user and maintainer of the server and database at the same time, while testing every functionality of the application.

## Usability

The application has book categories and uses easy-to-read fonts for reading. The interface is simple to use, but users can send messages to administrators for complaints.

# Design Constraints

The application will be built in Java using Spring, JavaFX libraries for the user interface and JDBC for communicating with the MySQL databases.