Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/03/18 | 1.0 | Setting the first non-functional requirements and design constraints | Biris Alexandra |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

# 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

The system is available 100% for the user and is used 24 hours a day and 365 days a

year. The system shall be operational 24 hours a day and 7 days a week.

## Performance

The system shall respond to the user in not less than several seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs, but responses to view information shall take no longer than 5 seconds to appear on the screen.

## Security

There are different categories of users, more exactly, client and administator. Each unique user should have access only to a limited number of actions, according to their rights. All system data must be backed up every 24 hours and the backup copies stored in a secure location. By this we can achieve a secure database for the project, but this functionality is beyond the course objective.

## Testability

The implemented system is tested using unit testing technique, by using test cases written by us or automatically generated ones.

## Usability

The system shall allow the users to access the system from the Internet as it uses a web browser as an interface. Since all users are familiar with the general usage of browsers, no special training is required. The system is user friendly.

# Design Constraints

## Hardware Constraints

The system requires a database in order to store persistent data. The database should have

backup capabilities. In some environments, RAM utilization and secondary storage (or lack of it)

are very real restrictions.

## Software Constraints

The development of the system will be constrained by the availability of required

software such as database and development tools. To develop the system we will use Java

language, Spring and Hibernate frameworks, Javafx for the GUI and MySQL for the database.