Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <18/3/20> | <1.0> | <general overview> | <Ioana Bozdog> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 5

2.5 Usability 5

3. Design Constraints 5

# Introduction

[The introduction of the **Supplementary Specification** provides an overview of the entire document.

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 availability is concerned with system failure

The stimulus is referring to crashes

The environment is the application

The response should be to display a message to notify the change

## Performance

The performance refers to timing, when events (interrupts, messages, requests from users etc.) occur and the system must respond to them

The source of stimulus are independent sources, users

In response, transactions are processed

## Security

The security is a measure of the system’s ability to resist unauthorized/illegal usage

The source of the stimulus is correctly identifying the individual

The stimulus is to resist the illegal usage

The environment is the application under normal operations

The response consists of the system blocking the access

## Testability

Testability refers to the ease with which software can be made to demonstrate its faults through testing

The source of the stimulus is the developer

The stimulus is to perform unit test

## Usability

Usability refers to how easy is for the user to accomplish a desired task

The source of the stimulus is the user

The stimulus is to try to be as user-friendly as possible

# Design Constraints

[This section needs to indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

* Language: Java or C#
* Efficient code
* Make use of the emotion API
* Make use of a database