Version <1.0>



Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <18/03/2020> | <1.0> | Project deliverables 1 | Socaci Radu Andrei |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

* *Availability is concerned with system failure and its associated consequences*
* *Source of stimulus: external*
* *Stimulus: any user-application interaction*
* *Environment: normal operation (UP)*
* *Artifact: the whole system*
* *Response: notify action, try to perform it and in case failure occurred, retry.*
* *Response measure: no downtime*

## Performance

* *Performance refers to how quickly can the system respond to user generated evens*
* *Source of stimulus: external*
* *Stimulus: any user-application interaction*
* *Environment: normal operation*
* *Artifact: the whole system*
* *Response: perform action fast*
* *Response measure: low latency and a good response time*

## Security

* *The security aspect is primarily concerned with restricting unauthorized access and ensuring the system is not vulnerable to attacks.*
* *Source of stimulus: external*
* *Stimulus: modification attempt*
* *Environment: normal operation*
* *Artifact: system’s services*
* *Response: modification will be rejected with a 401/403 status code*
* *Response measure: the system is protected from malicious users*

## Testability

* *Testability refers to the ease of testing the application for defects.*
* *Source of stimulus: QA Tester*
* *Stimulus: testing is performed*
* *Environment: after component design is completed*
* *Artifact: the component*
* *Response: the system will respond with direct results of the tests performed*
* *Response measure: better test coverage*

## Usability

* *Usability refers to how easy is for the user to accomplish a desired task*
* *Source of stimulus: user*
* *Stimulus: user interaction*
* *Environment: system is deployed and usable*
* *Artifact: the whole system*
* *Response: the system should perform as the user requested*
* *Response measure: statistic with the amount of failures*

# Design Constraints

* The application should be developed using Node Express and React.
* The application should use the Model-View-Controller pattern client side and a 3-tier architecture server side.
* It should be a web application.
* The project should also have a client-server architectural style and the Observer design pattern (embedded through promises).
* The data will be stored in Firebase.
* Coding standards should be respected.