Initiatives Platform

Supplementary Specification

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/Mar/2018 | 1.0 | First version | Varadi Robert  Ieremias Viorel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 5

2.4 Reusability 5

2.5 Usability 5

3. Design Constraints 5

Supplementary Specification

# Introduction

The purpose of this document is to define requirements of the Initiatives Platform System. This Supplementary Specification lists the requirements that are not readily captured in the use cases of the use-case model. The Supplementary Specifications and the use-case model together capture a complete set of requirements on the system.

This specification defines the non-functional requirements of the system, such as reliability, usability, performance, and supportability as well as functional requirements that are common across a number of use cases. (The functional requirements are defined in the Use Case Specifications).

The non-functional requirements are prioritized based on ease of use and overall extendibility of the Initiatives Platform System, while respecting the required development and execution times.

# Non-functional Requirements

## Availability

The system must be available 100% of the time, except the planned periods of maintenance, when the system is under development and it is not available for the users.

## Accessibility

The platform must provide an intuitive user interface that is natural to use even for users with poor technological education. Also, it should be accessible from older or less performant hardware and software platforms.

## Capacity

The system must accommodate around 10,000 users must be able to process 500,000 requests per month.

## Performance

The system should be characterized by performance, that is:

* the response time for the operations (latency) must be less than 1s,
* the throughput (rate of processing work) of the system must be able to process 100 requests per second.

## Platform Compatibility

The system must be compatible with all platforms that can run a browser of version similar or newer than IE8.

## Security

The system must provide secure login and session management and must ensure data privacy especially on sensitive data. It also must follow the OWASP security practices.

# Design Constraints

The application must be implemented as a web-based platform with the compatibility constraints mentioned in Section 2 (Non-functional Requirements). It must be structured as a layered application in order to facilitate reuse and portability. Also, the application must implement the MVC pattern for easy integration of different UIs. The authentication module must be compliant with the OAuth2.0 authorization and authentication protocol. In terms of programming language Java will be used for its portability, robustness and multitude of extensions. Regarding the frameworks, Hibernate is the chosen JPA implementation and Spring is the chosen web framework. For the GUI, JavaScript library React will be used to create a responsive user interface.