<Online School>

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 project requires a great support.

For a great performance the operating system will be Ubuntu 18.04 and. The project will be done in

Visual Studio code IDE.

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

*[Define system quality attributes in terms of scenarios according to the following template:*

* *Quality attribute definition*
* *Source of stimulus: the entity (human or another system) that generated the stimulus or event*
* *Stimulus: a condition that determines a reaction of the system*
* *Environment: the current condition of the system when the stimulus arrives*
* *Artifact: is a component that reacts to the stimulus. It may be the whole system or some pieces of it*
* *Response: the activity determined by the arrival of the stimulus*
* *Response measure: the quantifiable indication of the response*
* *Tactics*

*]*

*---TO BE COMPLETED---*

## Availability

This section lists all reliability requirements.

* 1. **Availability**

The Registration System shall be available 24 hours a day, 7 days a week.

## Performance

The performance characteristics of the system are outlined in this section.

* 1. **Simultaneous Users**

The system shall support up to 2000 simultaneous users against the central database at any given time, and up to 500 simultaneous users against the local servers at any one time.

* 1. **Database Access Response Time**

The system shall provide access to the legacy course catalog database with no more than a 10 second latency.

## Security

## Testability

## Usability

This section lists all of those requirements that relate to, or affect, the usability of the system.

* 1. **Windows/Linux Compliance**

The desktop user-interface shall be Windows 10 and Linux distribution compliant.

* 1. **Design for Ease-of-Use**

The user interface of the C-Registration System shall be designed for ease-of-use and shall be appropriate for a computer-literate user community with no additional training on the System.

# Design Constraints

Some of the contraints:

* The language used will be python
* (--to be completed--)

[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.]