<Art gallery>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <22/03/2023> | <1.0> | <Initial specification> | <Tusa Laurentiu> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 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: User View interaction*
* *Source of stimulus: User sends a request to the application*
* *Stimulus: The application receives a new request*
* *Environment: the system is waiting for the user’s input*
* *Artifact: interface, system logic, database*
* *Response: the interface reacts to the user’s input, which takes the data, processes it and start searching for products in the database.*
* *Response measure: the products are displayed.*
* *Tactics: Use adequate system design for facilitating data flow.*

## Availability

The application is available to anyone that has access to internet and has an account created.

## Performance

The performance is assured by efficient algorithms and simple and basic tasks to be executed.

## Security

Security of the system will be controlled by an encryption of the user’s credentials and by further verifications on the input data provided by the user.

## Testability

The application will provide multiple tests to ensure that the data is stored correctly and that data is fetched in accordance with the provided requests.

## Usability

The usability of the application covers the quick access to the products and specific filters to be applied on the whole data set of products.

# 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.]

The art gallery application will be developed in .NET, following an object oriented paradigm. The design of the application will feature multiple menus, for each special use case, where the user can input data required for the specific operation.