Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/03/2023 | 1.0 | First version | Ratoi Razvan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

This document will capture the system requirements that are not readily captures in the uses 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

Quality Attribute: Usability Source of Stimulus: The end-user Stimulus: The user wants to access a specific feature or function of the software system Environment: The user is using the software system in a typical operating environment Artifact: The user interface of the software system Response: The software system provides the user with a clear and easy-to-understand way to access the desired feature or function Response Measure: The time it takes for the user to find and access the desired feature or function Tactics: Use of clear and consistent labeling and organization of features, intuitive navigation, and user-friendly interfaces.

## Availability

This application will support the requirements specified, both functional and non-functional in a proper manner. The functionalities that will be developed for this application will be mainly designed for the requirements but will also be modeled according to some principles in order to achieve easy maintainability and modifiability.

## Performance

The application will be designed to run with a high performance, low complexity, so that the users can complete theirs tasks with the help of the application in the most efficient and fast way. Ideally, the longest period of any operation is represented by the access to the database, while performing any of the CRUD operations.

## Security

This product will be user-secured, meaning that each user of this application will have an account, password-protected and it will include authorization based on roles, so any activity on this application will require a registered user. The passwords will be encrypted in some way to secure the accounts from being accessed by any malicious users.

## Testability

The application should be easy to test, by using unit tests for each operation/function and finally by testing the final requests from the back end through an appropriate service such as Postman or Insomnia.

## Usability

The application should be easy to use, to be intuitive for the user such that there won’t be any time wasted on confusion with what has to be clicked or where one functionality may be. It should be organized per roles, such that, based on their role, the user will be redirected to a page where only the authorized operations would be found.

# Design Constraints

This application will be developed using C# language and .NET framework, which would imply to the respect of the OOP Paradigms. Also, the application will be design under the model of a Layered Architecture. The management of the database will be done with the help of the Entity Framework, which would also help in the development of other functionalities.