<rentIT>

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <18/03/2021> | <1.0> | <details> | <Rus Rares> |
| <25/03/2021> | <1.1> | <details> | <Rus Rares> |
| <28/04/2021> | <2.0> | <finals> | <Rus Rares> |
| <26/05/2021> | <2.1> | <finals> | <Rus Rares> |

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 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

*Well,* [*non-functional requirements*](https://en.wikipedia.org/wiki/Non-functional_requirement) *are* [*requirements*](https://en.wikipedia.org/wiki/Requirement) *that define the operation of the system under test rather than the behavior of the system under test, or the* [*functional requirements*](https://en.wikipedia.org/wiki/Functional_requirement) *as these are known. A synchronous database request must respond with 1000 milliseconds. By using a renowned database MySQL Workbench, it will be able to support thousands of users, admin and matches entries with respect to the load time.*

## Availability

The application should run with no down time.

## Performance

Quick data base access and response, rapid communication between user and application, more than one user can use the app at once.

## Security

The security soft goal is divided into confidentially, integrity, and availability. To cover confidentiality, it’s important to achieve authentication, users are required to log into the system by providing username and with strong password. Integrity covers completeness such as data validation, accuracy, and consistency.

## Testability

Hard tests at the beta stage of the app, then some test regularly every 2 weeks.

## Usability

The app should be used by any one who want to rent or to find a quick way to rent an apartment.

# Design Constraints

The application should be client-server and the data on the server will be stored in a database. All the inputs of the application will be validated against invalid data before submitting the data and saving it. Should be able to run on any web browser.