<Project Name>

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 16/03/2020 | 1.0 | Project Deliverable 1: Supplementary Specification | Mateiu Bianca |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 **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

## Availability

The flower shop shall be available 24 hours a day, 7 days a week.

## Performance

This section will give an overall description of the performance characteristics of the system. **Simultaneous Users:**

The system shall support multiple users against the central database at any given time.

**Database Access Response Time** :

The system shall provide access to the flower shop with no time or very little latency. **Transaction Response Time:**

The system must be able to complete 80% of all transactions within 2 minutes.

## Security

The application will be secured, and it will protect its users data.

## Testability

Unit testing and integration testing will be developed for the application.

## Usability

The desktop user-interface shall be Windows compliant.

# Design Constraints

The application will be designed using Java and Spring.

The data used in the application will be stored in a database, using an ORM (Object Relational Mapping) framework. I will be using Hibernate for this.

For testing the application, Junit will be used.