Shopping list

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 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.]

Project’s name is Shopping list, this project’s purpose is to create an application when users can share their shopping list with other users that are members to the same group as him.

Requirements :

Design an implement an application for an online shopping list. Each user will be able to create an account and create or join a group, there they will be able to post new products to buy or mark a product as bought. Users are able to create some constraints for a product, like pick product producer, or offer a list of shops from where products should be bought. An user should create an account by providing username, password and email. Users will be able to:

* Login
* Update account data
* Create groups
* Join groups
* Leave groups
* Add products
* Configure product 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*

*]*

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, not his behavior. Non functional-requirements are designed in the system architecture.

The stimulus is represented by the user who logs in and uses the application. If he logs in successfully he will be able to do operations described before, and he will initialize a chain of operations that begins in GUI and and end in the database, where all the data is kept.

Artifact is represented by reacting to stimulus, and it will be represented by a GUI, more specific a website that will provide forms for user to insert new data, and also buttons to do specific actions, like add product or mark product as bought.

Response is a result given to a request received by controller and sent forward to service, database, etc. Application can handle this request with a simple database query or even a complex one, it also can load a full web page, it depends of the type of response

## Availability

Represents the probability that a system will work as required when required during the period of a mission. This aspect is assured mainly by testing the application, because if the application is tested bugs can be discovered much easily and fixed as well.

## Performance

Represents the amount of useful work accomplished by computer system.

It can be represented by:

* Short response time for a given piece of work
* High throughput (rate of processing work)
* Low utilization of computing resources
* High availability of computing system or application
* High bandwidth
* Short data transmission time

## Security

Is represented by the capability of a system to prevent attacks, data stealing, accidental actions and also to provide privacy.

In my application security will be implemented using spring security in the form of user accounts, so that each user will be able to execute operations on his behalf only.

## Testability

Software testability is the degree to which a software artifact (i.e. a software system, software module, requirements- or design document) supports testing in a given test context. If the testability of the software artifact is high, then finding faults in the system (if it has any) by means of testing is easier.

My application will be tested using unit and integration tests, of all possible flows including successful ones and the ones that lead to failure.

## Usability

Defines how well the application meets the requirements of an user, how user friendly it is (how easy it is to use). Operations that users will be able to do were described earlier.

# 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 application will by implemented using Java as programming language and Spring boot as main framework for developing and hibernate for database access, it could also use more framework, but that will be discovered when some jobs are needed (like placing shop address on map). As database I will use MySQL. As IDE I will use Intellij.