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

# Introduction

The tool used to develop this project is Java Spring Boot which focuses on developing web application and microservices with Spring Framework. The application should be compatible with the latest version of Java and Spring Boot.

Regarding the operating system, the application should be designed to run on both Windows and Linux.

The database used for this project should be MySQL or PostgreSQL.

Moreover, the application should be able to handle a large number of users and orders simultaneously.

# Non-functional Requirements

## Availability

**Quality attribute definition:** availability isconcerned with system failure and its associated consequences.

**Source of stimulus:** internal/external.

**Stimulus:** omission, crash, timing, response.

**Environment:** normal, degraded operation

**Artifact:** process, storage, processor, communication

**Response:** record, notify, disable, continue (normal/degraded) be unavailable

**Response measure:** repair, time, availability, available/degraded, time interval

**Tactics:** failover mechanisms, regular system maintenance, detect issues, measures to prevent downtime.

## Performance

**Quality attribute definition:** performance focuses on responding to the occurred events.

**Source of stimulus:** users.

**Stimulus:** periodic events.

**Environment:** normal mode.

**Artifact:** system.

**Response:** processes events and stimuli from users.

**Response measure:** latency, deadline, miss rate, data loss.

**Tactics:** caching, database optimization, parallelism, minimizing resource usage.

## Security

**Quality attribute definition:** system's ability to resist unauthorized usage while still providing its services to legitimate users.

**Source of stimulus:** system correctly identified, incorrectly identified, unknown external/internal identity, authorized/non-authorized access to limited resources

**Stimulus:** the attempt of modifying the data, accessing system services, reducing availability to system resources

**Environment:** online/offline, connected/disconnected, firewalled/open

**Artifact:** system services

**Response:** authenticates/hides users, blocks/allows access, grants/withdraws permission to access data, records access/modifications to access or modify data/services by identity

**Response measure:** security measures with probability of success; probability of detecting attack and its responsible individual; restore data/services.

**Tactics:** strong authentication mechanisms, encrypting sensitive data, implementing firewalls.

## Testability

**Quality attribute definition:** testability refers to the ease with which software can be made to demonstrate its faults through testing.

**Source of stimulus:** unit tester

**Stimulus:** performs unit tests

**Environment:** design time, developing time, compilation time, deployment time

**Artifact:** component of the system

**Response:** prepares test environment

**Response measure: p**robability of failure if fault exists; time to perform tests; time to prepare test environment

**Tactics:**

## Usability

**Quality attribute definition:** usability focuses on the accomplishment of a desired task and the kind of user support the system provides.

**Source of stimulus:** users

**Stimulus:** use system efficiently, minimize impact of errors, adapt system

**Environment:** runtime, configure time

**Artifact:** system

**Response:** support for efficient navigation within a screen; multiple simultaneous activities; re-use of already entered data and/or commands.

**Response measure:** user satisfaction, task time, gain of user knowledge, number of errors.

**Tactics:** user centered design, error prevention and handling, documentation, usability testing.

# Design Constraints

The application must be developed using Java as the primary programming language and using the Spring Boot framework, used for developing web applications.

It must use a specific database management system, such as MySQL or PostgreSQL in order to store information relevant to the cake shop, like users, customers or admin, ingredients, products, orders and so on.

The project must provide unit testing for the implemented operations and use the validations.

The application must be developed using IntelliJ IDEA or Eclipse.

Regarding the user interface, the application does not have any design constraints