Plane Tickets

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <02/04/2019> | <1.0> | <details> | Prata Mirela |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

This document clarifies functional and nonfunctional details of the application, so that the client can be informed about the system. The system’s requirements will be included in this file for the usability of the application, problems about the legally side of the application- for who is built, also the personal data which will be given for every client.

The concerns about quality, performance, usability, supportability, alongside the operating systems will be discussed and design constraints and compatibility requirements.

# Non-functional Requirements

Basically, Non-functional requirements describe how the system works, while functional requirements describe what the system should do. This does not mean the latter are more important, but most requirement gathering techniques focus on functional requirements, so large gaps in non-functional requirements are common.

## Availability

The system must be available all the time, for both types of user. The application will be always running, being available to everyone who has internet access and a device that supports a web page in browser.

## Performance

The application will be simple and fast, the client can enter and select the desired destination, the suitable offer and can buy the tickets in the cart.

## Security

Application must be secured, all fields must be validated for protection against SQL injection and all data should be protected by a password. For making sure about the security, a strong password will be required.

## Testability

The application should be easy to test, the modularity of the code improves this aspect.

## Usability

The system will offer a friendly page, as well as correct information. The application can be used by any type of person who is interested in purchasing a plane ticket.

# Design Constraints

The application is implemented using Spring Boot, a framework for Java, with a Hibernate connection to the database. The layered architectural pattern is used for separating the concerns.

The system has a logging part, simulating a real login for an application and it offers a real plane tickets market with different offers for destination.