<Application for Sports Bookings>

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <17/03/2020> | <1.0> | <details> | <George Baraian > |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 sports bookings platform is a easy to use and relative interactive platform, with low system requirements that can operate on any operating system.

# Non-functional Requirements

The application should provide an easy interface for sport facility companies to interact with their clients and to make the process of choosing a court simple and intuitive. The system should be easy to use, but to offer enough functionalities and features.

## Availability

The application should be available all the time and the integration of a new feature should not make the application unusable.

## Performance

The queries to the database should be executed in the least amount of time possible. To ensure this, concurrent access to the database should be implemented, so that the application, if installed on a multi-threaded system, to enable a better performance.

## Security

The users and their approved operations should be clearly defined, so for example no client can access the operations of an admin and vice-versa. The passwords used for logging in should be encrypted with a one-way encryption algorithm.

## Testability

JUnit tests will be developed in order to stress test each component in isolation and to ensure that its functionalities are 100% correct. Integration tests would be also a valuable feature, but since this is only a small system, at this stage they are not necessary.

## Usability

Easy to use by the end user and by the administrator.

# Design Constraints

The programming language used will be Java for the backend and for the frontend Angular (or React or JQuery) will be used. Regarding the database, data will be stored in a MySQL DB and the connection from the application to the DB will be through the Hibernate framework. I will use a layered architecture alongside the MVC pattern, all of which work really well with the Spring framework for the development of web apps.