<Project Name>

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

Project’s name is Goal Score, this project’s purpose is to create an application when users can view rankings, top scorers and scores of a league.

Requirements:

Design and implement an application for football stats. Each user will be able to create an account, login, select favorites teams, view ranking of a league, view next matches for the favorite team and view the top scorers of a league. An user should create an account by providing username, password and email.

# Non-functional Requirements

A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

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 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

Availability of a system is typically measured as a factor of its reliability – as reliability increases, so does availability. Availability of a system may also be increased by the strategy of focusing on increasing testability, diagnostics and maintainability and not on reliability.

## Performance

Computer performance is the amount of useful work accomplished by a computer system. Outside of specific contexts, computer performance is estimated in terms of accuracy, efficiency and speed of executing computer program instructions.

## Security

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

## 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.

## Usability

Usability is the degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

# Design Constraints

The application will be implemented using Java as programming language and Spring boot as main framework for developing and hibernate for database access. As database I will use MySQL Workbench. As IDE I will use IntelliJ.