Coach4Health

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

This document captures specifies the system requirements that are not readily captured in the use cases of the use-case model. These system requirements include quality attributes (availability, performance, security, testability, usability) and design constraints (software languages, software process requirements, operating systems and environments and so on). For each system requirements categories there will be a separate section.

# Non-functional Requirements

## Availability

Availability represents the degree to which a [system](https://en.wikipedia.org/wiki/System), [subsystem](https://en.wikipedia.org/wiki/Subsystem) or equipment is in a specified operable and committable state at the start of a specific task/service. Simply, it can be thought of as the proportion of time in which the system is found in a functioning condition. The application should be available to its users at any given time and different failures (such as database server failure) should not affect it’s using process.

## Performance

Performance represents the amount of useful work accomplished by the application. It can be estimated in terms of accuracy, efficiency and response time. Performance is implied when a user wants to pay for a workout/meal plan. The payment system used inside the “Coach4Health” must be fast and secure. Also the response time of the application must be kept to the minimum all the time.

## Security

Security represents the protection of hardware, software, data, people and also the procedures by which an application is accessed. Maintaining this attribute in case of “Coach4Health” means protecting all the personal data belonging to all the users. Also special attention must be directed to the payment system, since a user’s credit card information must be kept secure and hidden from other malicious users.

## Testability

Testability represents the degree to which a software artifact (software system, software module etc.) supports testing in a given test context. If the testability of the software system is high, then finding faults in the system (if it has any) by different means of testing is easier. Initially, the application will be released in a beta version so a few potential users can help the development team with ratings and identifying different security and performance faults. After the beta testing, the application will be available to the general public and made available for use on mobile devices as well.

## Usability

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), 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. As long as “Coach4Health” will implement all the functional requirements described in the “Project Analysis and Design” document, it will maintain a high grade of usability.

# Design Constraints

The application “Coach4Health” must meet the following requirements:

* Designed using a layered architecture
* Implemented using Java11 (different frameworks are allowed for creating the user interface)
* Operating system: Windows8 / Windows 10
* Must implement all the functional requirements described in the “Project Analysis and Design” document.
* Must have an associated MySQL database for storing information such as: user personal data, workout meals, workout plans etc.
* Flexible design and incremental implementation process (if we want to modify certain functionalities already implemented, we can do this after analyzing if this is appropriate and would not affect other parts of the application)