<Car Renting Aplication>

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <20/03/2017> | <1.0> |  | <Ionel Carmen> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

Supplementary Specification ofera o imagine de ansamblu a intregului document. Acesta include cerintele non-functionale si cerintele de sistem, care nu sunt usor de capturat in modelul de use case. Astfel de cerinte include:

-cerinte legale si de reglementare, incluzand standardele de aplicare

-atributele de calitate ale sistemului care urmeaza sa fie construit, inclusive gradul de utilizare, performanta si securitatea.

-alte cerinte, cum ar fi: sistemul de operare, cerintele de compatibilitate si constrangerile de proiectare.

# Non-functional Requirements

## Availability

Reprezinta probabilitatea ca un sistem sa functioneze corect chiar si dupa folosirea lui un anumit timp.

Atat datele cat si pagina web vor fi disponibile atat timp cat autorul sau administratorul nu le sterge.

## Performance

Una dintre cele mai importante cerinte non-functionale dorite de catre client este un sistem cu o rata mare de performanta.

Obiectivul de performanta este divizat in trei: timp, spatiu si receptivitate. Timpul este impartit in timp de raspuns si timp de procesare. Receptivitatea reprezinta cat de repede raspunde sistemul la interogari, iar spatiul reprezinta spatiul de stocare. Pentru o buna performanta, nu trebuie irosit spatiul.

## Security

Obiectivul de performanta este divizat in: confidentialitate, integritate si disponibilitate. Pentru a acoperi confidentialitatea este importanta autentificarea: utilizatorii trebuie sa de conecteze la sistem prin furnizarea unui username si a unei parole. Integritatea se refera la validarea datelor si asigura consistenta.

## Testability

Reprezinta gradul in care un sistem suporta testarea intr-un anumit context.

Pentru a testa un sistem trebuie sa avem in vedere:

-proprietatile metodelor de test folosite

-cerintele trebuie sa fie consistente, complete, neambigue, realizabile

-componentele software trebuie sa fie: controlabile, observabile, heterogene, documentate.

## Usability

Orice sistem creat pentru a fi utilizat de catre oameni trebuie sa fie usor de folosit, usor de invatat, usor de amintit (instructiunile) si folositor utilizatorului.

Principii:

-concentrat pe utilizatorii finali sip e nevoile acestora

-design iterative, acesta fiind imbunatatit in permanenta

# Design Constraints

Aplicatia CRA este limitata de catre serverul web si de catre serverul bazei de date.