Software Requirements Specification Template

Author: Uwe van Heesch

Version: 1 (21/09/16)

1 Introduction 1

1.1 Overall Description 1

1.2 User Classes and Characteristics 1

1.3 Operating Environment 1

1.4 Design and Implementation Constraints 2

1.5 Product Functions 2

*2* *Domain Model* 2

*3* Use-case Descriptions 2

3.1 Use case 1 2

3.1.1 Fully-dressed use case description 2

3.1.2 System Sequence Diagram (optional) 2

3.1.3 Operation Contracts (optional) 2

3.2 Use case 2 ( and so on) 2

4 Other functional requirements (optional) 3

5 Non-functional Requirements 3

5.1 Performance Efficiency 3

5.2 Security 3

5.3 Reliability (and so on) 3

6 User interface sketches (optional) 3

# Introduction

## Overall Description

*<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. If a separate description of the product scope is available (e.g. in the PvA), refer to it rather than duplicating its contents here.>*

## User Classes and Characteristics

*<Identify the various user classes (actors) that you anticipate will use this product. User classes may be differentiated based on the subset of product functions used. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. >*

## Operating Environment

*<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>*

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: hardware (e.g. specific mobile platforms), specific technologies, tools, and databases to be used; interfaces to other applications; programming language required; or communications protocols>*

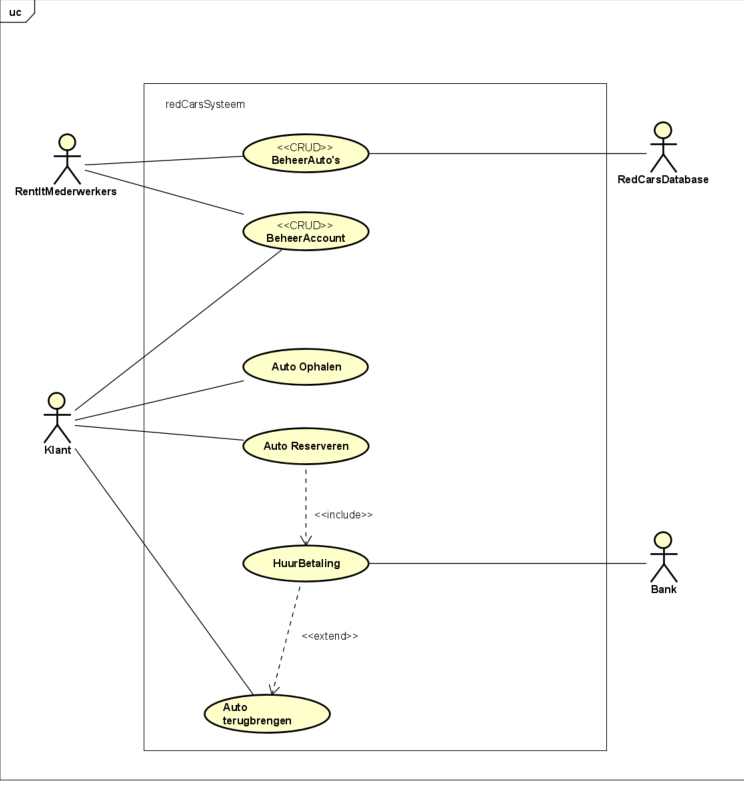
## Product Functions

*<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. In most cases, this section will primarily contain a use case diagram and brief use case descriptions >*

# *Domain Model*

<Provide a diagram showing important real-situation conceptual classes in the application domain. Do NOT include software classes. Describe each of the conceptual classes in a glossary.>

# Use-case Descriptions

*Hieronder het usecase diagram van het RedCarsSysteem.*

## Brief Descriptions

### Reserveren

De klant bekijkt welke auto’s beschikbaar zijn in de stad waar hij een auto wil huren. Hier kiest hij een auto uit. De klant geeft aan welk type auto hij wil gebruiken. Hij geeft ook aan hoe lang hij deze auto wil huren.

### Betaling van de huur (bij reservering)[[1]](#footnote-1)

De klant bekijkt welke auto’s beschikbaar zijn in de stad waar hij een auto wil huren. Hier kiest hij een auto uit. De klant geeft aan welk type auto hij wil gebruiken. Hij geeft ook aan hoe lang hij deze auto wil huren.

### Use Case: Terugbrengen van auto

De klant komt de RedCar terugbrengen waar hij de auto heeft opgehaald. Hij parkeert hier de auto en checkt uit bij de paal.

### Betaling van de huur (bij terugbrengen van de auto)

Het systeem berekent de kosten van de rit aan de hand van het type abonnement van de klant en het aantal gereden kilometers. Deze kosten worden meegenomen in de automatische incasso die de bank naar de klant stuurt.[[2]](#footnote-2)

### Ophalen van auto

De klant gaat naar de parkeerlocatie waar hij heeft aangegeven de auto op te halen. Hij checkt in bij de paal met zijn RedCars-pas. Hierna loopt hij naar de auto en opent deze met zijn pas. De klant kan nu wegrijden.

### Beheer auto’s

De RentItMedewerker bekijkt de data van de auto’s in de Redcars database. Mochten er waarden zijn veranderd, dan bewerkt de medewerker deze.

### Beheer account

De klant logt in op zijn Redcars account. Hier kan hij wijzigingen doorgeven in abonnement-type, woonplaats. Ook kan hij zijn account deactiveren.

De mederwerker heeft toegang tot de Redcars database. Hier staan de klantgegevens. De medewerker kan alle gegevens aanpassen. Bij gevallen van misbruik ( slecht betalen, auto’s achterlaten met nieuwe gebreken) maakt de medewerker de klant inactief.

## Reserveren

### Fully-dressed use case description

**Use Case:** Reserveren

|  |  |
| --- | --- |
| **Primary actor:** Klant | |
| **Stakeholders and Interests:** Klant, RentIt | |
| **Brief description:**  De klant bekijkt welke auto’s beschikbaar zijn in de stad waar hij een auto wil huren. Hier kiest hij een auto uit.  De klant geeft aan welk type auto hij wil gebruiken. Hij geeft ook aan hoe lang hij deze auto wil huren. | |
| **Preconditions:**  De klant heeft geen betalingsachterstand.  De klant is ingelogd. | |
| **Postconditions (Success Guarantee):**  De gereserveerde auto verschijnt niet langer als beschikbaar op de RedCarswebsite.  Het is voor de klant mogelijk om op de gekozen tijd in de geserveerde auto in te checken.  De klant heeft betaald. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De klant typt de stad in waar hij een auto wil huren. | 1. Het systeem laat de beschikbare auto’s zien in de opgegeven stad. |
| 1. De klant geeft het type auto aan. | 1. Het systeem laat de beschikbare auto’s zien van dat type.\* |
| 1. De klant geeft aan tijdens welke periode hij de auto wil huren. | 1. Het systeem laat de tijdens de opgegeven tijdperiode beschikbare auto’s zien.\* |
| 1. De klant kiest een auto die voldoet aan de opgegeven specificaties. | 1. Het systeem berekend de prijs en toont deze. |
| 1. De klant bevestigd de huurovereenkomst. | 1. Het systeem verwerkt de registratie in de database. |
| **Extensions (Alternative Flow): De klant gaat niet akkoord met de berekende prijs** | |
| 9. De klant gaat niet akkoord. | 10. Het systeem toont de beschikbare auto’s die voldoen aan de eerder opgegeven specificaties. |
| **Extensions (Alternative Flow): Er zijn geen auto’s aanwezig die voldoen aan de opgegeven specificaties** | |
|  | 2/4/6. Het systeem laat zien dat er geen auto’s beschikbaar zijn die voldoen aan de opgeven specificaties. |

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

## Betaling van de huur (bij reservering)

### Fully dressed use case description

**Use Case:** Betaling van de huur (bij reservering)

|  |  |
| --- | --- |
| **Primary actor:** Systeem | |
| **Stakeholders and Interests:** Klant, RentIt, Bank | |
| **Brief description:**  De klant heeft aangegeven hoe lang hij welk type wil huren. Op basis van deze gegevens berekent het systeem de kosten. Wanneer de klant hiermee akkoord gaat wordt een automatische incasso verstuurd. | |
| **Preconditions:**  De klant is ingelogd.  De klant heeft een auto gereserveerd. | |
| **Postconditions (Success Guarantee):**  De automatische incasso is verstuurd.  Het geld is door RentIt ontvangen. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1: Het systeem haalt de abonnement gegevens van de klant op uit de database. |  |
| 2: Het systeem berekent de prijs op basis van het type auto, het soort abonnement van de klant, de opgegeven periode en per opgegeven tijdseenheid (uur/dag/weekend/week). |  |
| 3. In de database wordt het openstaande bedrag van de klant opgehoogd met de zojuist berekende waarde. |  |
| 4. Het systeem vraagt aan de bank een automatische incasso te versturen naar de klant. | 5. De bank stuurt een automatische incasso naar de klant. |
| 7. Het systeem registreert dat de automatische incasso betaald is. | 6. De bank ontvangt het geld. |
| 8. In de database wordt het openstaande bedrag van de klant verlaagd met de zojuist betaalde waarde. |  |
| **Extensions (Alternative Flow):. Betaling mislukt** | |
|  | 5.De bank stuurt een automatische incasso naar de klant. |
| 7. Het systeem registreert dat de automatische incasso niet betaald is. | 6. De bank registreert dat de afschrijving mislukt. |
| 8. In de database wordt het openstaande bedrag van de klant niet bijgewerkt. |  |
| 9. Het systeem stuurt een bericht naar de klant waarin staat dat de incasso mislukt is. |  |

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

### Operation Contracts (optional)

*<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>*

## Use Case: Terugbrengen van auto

### 3.3.1 Fully dressed use case description

**Use Case:** Terugbrengen van auto

|  |  |
| --- | --- |
| **Primary actor:** Klant | |
| **Stakeholders and Interests:** Klant, RentIt | |
| **Brief description:** De klant komt de RedCar terugbrengen waar hij de auto heeft opgehaald. Hij parkeert hier de auto en checkt uit bij de paal. | |
| **Preconditions:**  De klant heeft een auto ingecheckt. | |
| **Postconditions (Success Guarantee):**  De auto staat op de juiste parkeerplaats.  De auto is weer als beschikbaar te vinden op de website.  De resterende tijd van de reservering van de klant is geannuleerd. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1: De klant parkeert zijn RedCars auto op de parkeerlocatie waar hij deze heeft opgehaald. |  |
| 2: De klant stapt uit. | 3: Het systeem registreert dat er een RedCars auto geparkeerd staat maar dat deze nog niet is uitgecheckt. De auto en de paal geven lichtsignalen. |
| 4: De klant checkt uit met zijn RedCars-pas bij de paal. | 5: Het systeem controleert of de klant een auto kan uitchecken op deze locatie. |
|  | 6. Het systeem controleert of de auto in goede staat is opgeleverd. |
|  | 7. De RedCarsDatabase wordt bijgewerkt met de ritgegevens (denk aan km gereden, benzine) |
|  | 8. Het systeem geeft aan de klant uitgecheckt is. |
|  | 9. Het systeem zet de auto weer op beschikbaar. |
|  | 10: De resterende tijd van de reservering van de klant wordt geannuleerd. |
|  | 11. De paal en de auto stoppen met het geven van lichtsignalen. |
|  | 12. De auto gaat op slot. |
| **Extensions (Alternative Flow):. Verkeerde parkeerplaats** | |
| 1: De klant parkeert zijn Redcars auto op een andere parkeerplaats dan waar hij deze heeft opgehaald. |  |
| 2: De klant stapt uit. | 3: Het systeem registreert dat er een RedCars auto geparkeerd staat maar dat deze nog niet is uitgecheckt. De auto en de paal geven lichtsignalen. |
| 4: De klant checkt uit met zijn RedCars-pas bij de paal. | 5: Het systeem controleert of de klant een auto kan uitchecken op deze locatie. |
|  | 6. Het systeem geeft aan dat de klant op deze locatie geen auto kan uitchecken. |
| **Extensions (Alternative Flow): Auto is slechte staat ingeleverd** | |
|  | 6. Het systeem controleert of de auto in goede staat is opgeleverd |
|  | 7. Het systeem merkt dat de auto in beschadigde stand is opgeleverd. |
|  | 8. De RedCarsDatabase wordt bijgewerkt met de ritgegevens (denk aan km gereden, benzine) |
|  | 9. Een RentIt medewerker wordt door het systeem op de hoogte gebracht dat de ingeleverde auto onderhoud nodig heeft. |
|  | 8. Het systeem geeft aan de klant uitgecheckt is. |
|  | 9: De resterende tijd van de reservering van de klant wordt geannuleerd. |
|  | 10 De paal en de auto stoppen met het geven van lichtsignalen. |
|  | 11. De auto gaat op slot. |
| **Extensions (Alternative Flow): Vergeten uit te checken** | |
| 4: De klant loopt zonder uit te checken weg van de parkeerlocatie. | 5. Het systeem registreert dat er een RedCars auto geparkeerd staat maar dat deze nog niet is uitgecheckt. De auto en de paal geven lichtsignalen. |
|  | 6. Na 15 minuten stuurt het systeem de klant een mail met een herinnering om uit te checken. Dit herhaalt zich ieder kwartier tot de klant is uitgecheckt. |
|  | 7. Twee uur nadat de auto geparkeerd is maar niet uitgecheckt stelt het systeem een RentIt medewerker op de hoogte. |

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

### Operation Contracts (optional)

*<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>*

## Betaling van de huur (bij terugbrengen van de auto)[[3]](#footnote-3)

### 3.4.1 Fully dressed use case description

|  |  |
| --- | --- |
| **Primary actor:** Systeem | |
| **Stakeholders and Interests:** Klant, RentIt, Bank | |
| **Brief description:** Het systeem berekent de kosten van de rit aan de hand van het type abonnement van de klant en het aantal gereden kilometers. Deze kosten worden meegenomen in de automatische incasso die de bank naar de klant stuurt. | |
| **Preconditions:**  De RedCars auto heeft het aantal gereden kilometers juist berekent.  De klant heeft de auto uitgecheckt.. | |
| **Postconditions (Success Guarantee):**  De automatische incasso is verstuurd.  Het geld is door RentIt ontvangen. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1: Het systeem haalt de abonnement gegevens van de klant op uit de database. |  |
| 2: Het systeem berekent de prijs op basis van het soort abonnement van de klant en het aantal gereden kilometers. |  |
| 3..Het systeem controleert op eventuele overschrijding van de gereserveerde periode. Bij overschrijding van de gereserveerde periode wordt het huurbedrag als boete berekend en worden de extra uren vervolgens nog apart per uur verrekend. |  |
| 4. In de database wordt het openstaande bedrag van de klant opgehoogd met de zojuist berekende waarde. |  |
| 5. Het systeem vraagt aan de bank een automatische incasso voor het gehele openstaande bedrag van de klant te versturen naar de klant. | 6. De bank stuurt een automatische incasso naar de klant. |
| 8. Het systeem registreert dat de automatische incasso betaald is. | 7. De bank ontvangt het geld. |
| 9. In de database wordt het openstaande bedrag van de klant verlaagd met de zojuist betaalde waarde. |  |
| **Extensions (Alternative Flow): Betaling mislukt** | |
|  | 5.De bank stuurt een automatische incasso naar de klant. |
| 7. Het systeem registreert dat de automatische incasso niet betaald is. | 6. De bank registreert dat de afschrijving mislukt. |
| 8. In de database wordt het openstaande bedrag van de klant niet bijgewerkt. |  |
| 9. Het systeem stuurt een bericht naar de klant waarin staat dat de incasso mislukt is. |  |

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

### Operation Contracts (optional)

*<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>*

## Ophalen van auto

### Fully dressed use case description

**Use Case:** Ophalen van auto

|  |  |
| --- | --- |
| **Primary actor:** Klant | |
| **Stakeholders and Interests:** Klant, RentIt | |
| **Brief description:**  De klant gaat naar de parkeerlocatie waar hij heeft aangegeven de auto op te halen. Hij checkt in bij de paal met zijn RedCars-pas. Hierna loopt hij naar de auto en opent deze met zijn pas. De klant kan nu wegrijden. | |
| **Preconditions:**  De klant heeft de op te halen auto gereserveerd.  De auto is door de vorige gebruiker op tijd naar de juiste plek teruggebracht. | |
| **Postconditions (Success Guarantee):**  De klant heeft ingecheckt bij de juiste paal.  De auto staat niet langer op de RedCar parkeerplek.  In de RedCarsDatabase is opgenomen dat de auto is ingecheckt. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1: De klant gaat naar de parkeerlocatie waar hij heeft aangegeven de auto op te halen. |  |
| 2: De klant checkt in bij de paal met zijn RedCars-pas. | 3. Het systeem controleert of de klant op deze parkeerlocatie een auto heeft gereserveerd. |
|  | 4. Het systeem geeft aan dat de klant nu is ingecheckt. |
| 5. De klant loopt naar de door heb gereserveerde auto en opent deze met zijn RedCars-pas. | 6. Het systeem controleert of de klant is ingecheckt voor dit type auto. |
| 8. De klant rijdt weg. | 7. Het systeem opent de auto, |
| **Extensions (Alternative Flow): Klant probeert de verkeerde auto te openen.** | |
|  | 7.. Het systeem geeft aan dat de klant bij de verkeerde auto is. |
| **Extensions (Alternative Flow): Klant heeft geen auto gereserveerd op deze parkeerlocatie.** | |
|  | 4. Het systeem geeft aan dat de klant geen auto gereserveerd heeft op deze locatie. |

### System Sequence Diagram (optional)

*<In case the use-case entails complex scenarios, you may decide to create a system sequence diagram showing events generated by external actors, the order of events and inter-system events. All systems are treated as a black box>*

### Operation Contracts (optional)

*<If the use case contains complex manipulations of domain objects, you may decide to specify operation contracts for all system operations included in the use case/ SSD.>*

# Other functional requirements (optional)

*<Use this section to describe functional requirements that cannot be expressed in the shape of use cases, for instance because they do not concern goal-oriented interactions of an actor with the system.>*

|  |  |
| --- | --- |
| Code | Description |
| FR1 | The system shall maintain an audit trail. |
| FR2 | .. |

# Non-functional Requirements

*<Describe non-functional requirements in this section. Please refer to existing software quality models like ISO\_IEC\_IEEE\_25010 or FURPS+.>*

## Performance Efficiency

|  |  |
| --- | --- |
| Code | Description |
| NFR1 | Responses to all user-initiated actions in the web-interface need to be rendered in less than 1 second. |
| NFR2 | .. |

## Security

|  |  |
| --- | --- |
| Code | Description |
| NFR3 | Personal user information needs to remain confidential to all parties other than system administrators. |
| … |  |

## Reliability (and so on)

# User interface sketches (optional)

*<Provide low-fidelity user interface sketches. Map the sketches to use cases and other requirements if applicable.>*

1. *Deze use case lijkt heel erg op de use case Use Case: Betaling van de huur (bij reservering). We hebben er voor gekozen om dit toch twee verschillende use cases te maken aangezien ze verschillende effecten op het systeem. Ook verschillen de pre condities.* [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)