



Software Requirements Specification RedCars

Opdracht van RentIt

Merlijn Warps[621737] & Sandra ter Maat[620458]
Datum: 29-10-2020

Inhoud

1	Introduction	2
1.1	Overall Description	2
1.2	User Classes and Characteristics.....	2
1.3	Operating Environment	2
1.4	Design and Implementation Constraints	2
1.5	Product Functions	2
2	<i>Domain Model</i>	2
3	Use-case Descriptions	4
3.0.1	System sequence diagram	5
3.1	Brief Descriptions.....	6
3.1.1	Reserveren	6
3.1.2	Betaling van de huur (bij reservering)	6
3.1.3	Use Case: Terugbrengen van auto	6
3.1.4	Betaling van de huur (bij terugbrengen van de auto)	6
3.1.5	Ophalen van auto.....	6
3.1.6	Beheer auto's	6
3.1.7	Beheer account	6
3.2	Reserveren	7
3.2.1	Fully-dressed use case description	7
3.2.2	System Sequence Diagram (optional).....	8
3.3	Betaling van de huur (bij reservering)	9
3.3.1	Fully dressed use case description.....	9
3.3.2	System Sequence Diagram (optional).....	10
3.4	Use Case: Terugbrengen van auto	11
3.4.1	Fully dressed use case description.....	11
3.4.1	System Sequence Diagram (optional).....	13
3.5	Betaling van de huur (bij terugbrengen van de auto)	14
3.5.1	Fully dressed use case description.....	14
3.5.1	System Sequence Diagram (optional).....	15
3.6	Ophalen van auto.....	16
3.6.1	Fully dressed use case description.....	16
3.6.2	System Sequence Diagram (optional).....	17
4	Requirements.....	18
4.1	Functionality:	18
4.2	Usability:	19
4.3	Reliability:.....	19
4.4	Performance:	19
4.5	Supportability:.....	19

1 Introduction

1.1 Overall Description

RentIt heeft de opdracht gegeven tot het opzetten van een applicatie voor het sub-bedrijf RedCars. De applicatie moet het verhuren van leenauto's op centrale punten mogelijk maken.

1.2 User Classes and Characteristics

Binnen de applicatie speelt de klant de grootste rol als actor. Hij is namelijk de persoon die de auto's gaat huren, ophalen en terugbrengen. Dit zijn de kerninteracties met het systeem. De RentIt medewerkers zijn minder relevant in het designperspectief, zij kunnen klant-en autogegevens beheren.

1.3 Operating Environment

Voor het werken van de applicatie een centrale server nodig. De auto's krijgen elk hun eigen hardwaremodule die kan communiceren met de centrale server. Verder staat er op elke parkeerlocatie een paal. Deze paal heeft wederom een connectie met de centrale server.

1.4 Design and Implementation Constraints

De applicatie wordt object georiënteerd ontwikkeld in de programmeertaal C++. Verder is er geen keuze gemaakt over hardware of specifieke platformen aangezien dit buiten de opdrachtscope valt.

1.5 Product Functions

De functionaliteiten van het product worden uitgelegd in hoofdstuk 3. Hier staan o.a. de use case diagrammen en de bijbehorende brief descriptions die veel inzicht geven in de functionaliteiten van de applicatie.

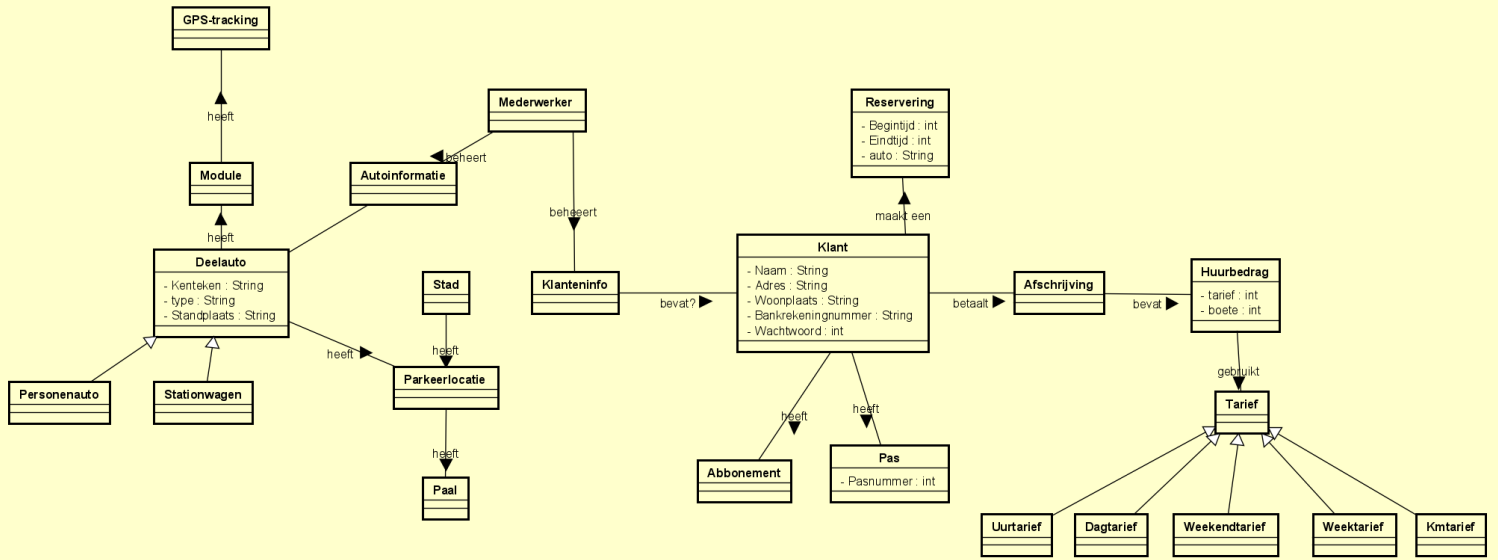
2 Domain Model

De centrale klasse hier is Klant. In het domeinmodel is te zien dat de klant een Reservering kan aanmaken, een Afschrijving kan betalen en beschikt over een Abonnement en een Pas. De Afschrijving bevat een bepaald Huurbedrag dat wordt opgebouwd vanuit een Tarief, zoals te zien is zijn er veel verschillende soorten tarieven.

Een andere belangrijke klasse is de Deelauto. Er zijn twee verschillende types auto's, personenauto en stationwagen. Elk van deze auto's heeft een eigen module die beschikt over GPS-tracking. Verder bevindt een auto zich standaard op een Parkeerlocatie. Elk van deze parkeerlocaties heeft een paal die een connectie heeft met de centrale server. In een stad kunnen meerdere parkeerlocaties zijn.

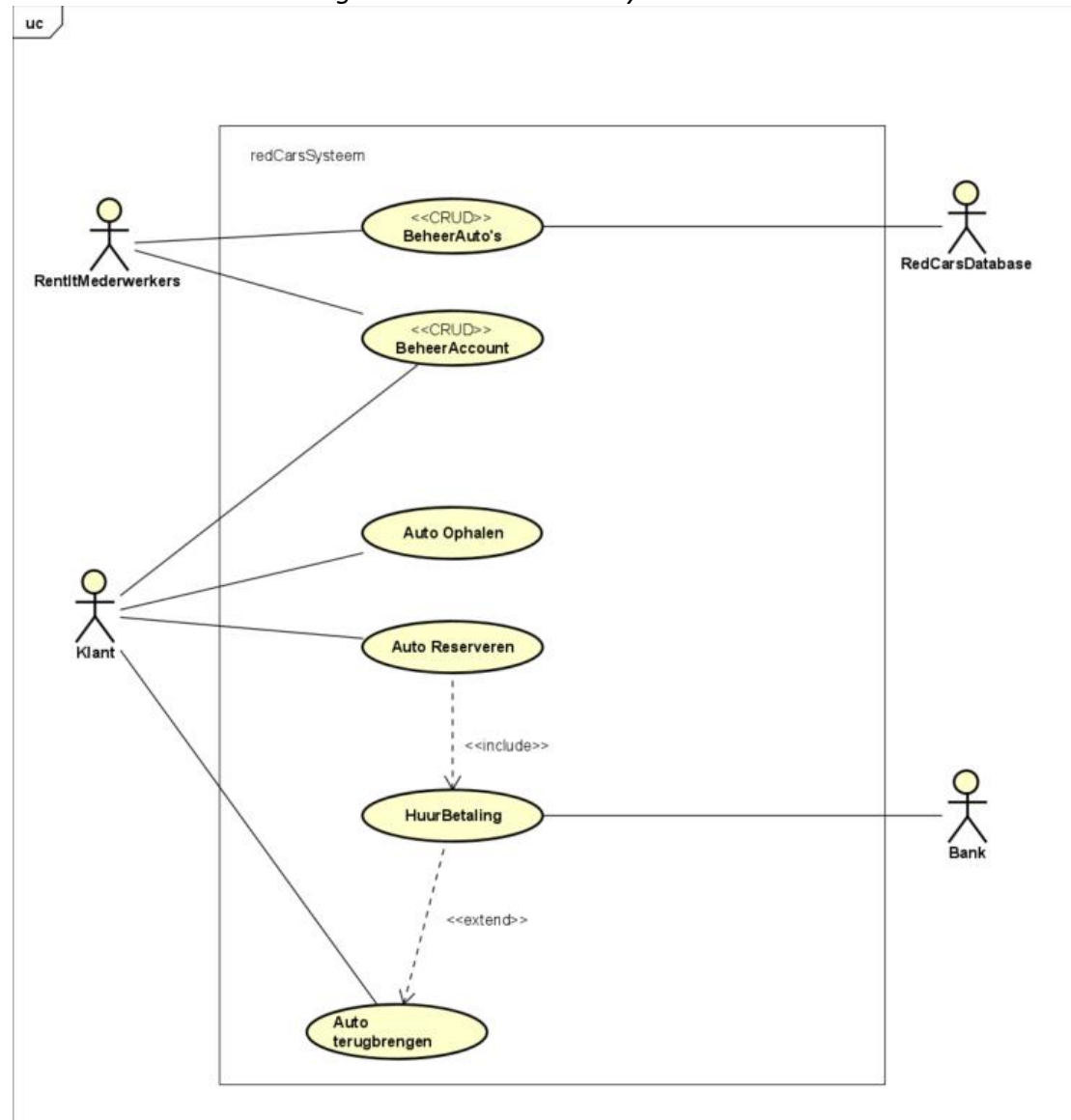
De RentItmederwerker heeft beschikking tot zowel de KlantenInformatie als de Autoinformatie en kan deze naar eigen inzicht beheren.

DomeinModel



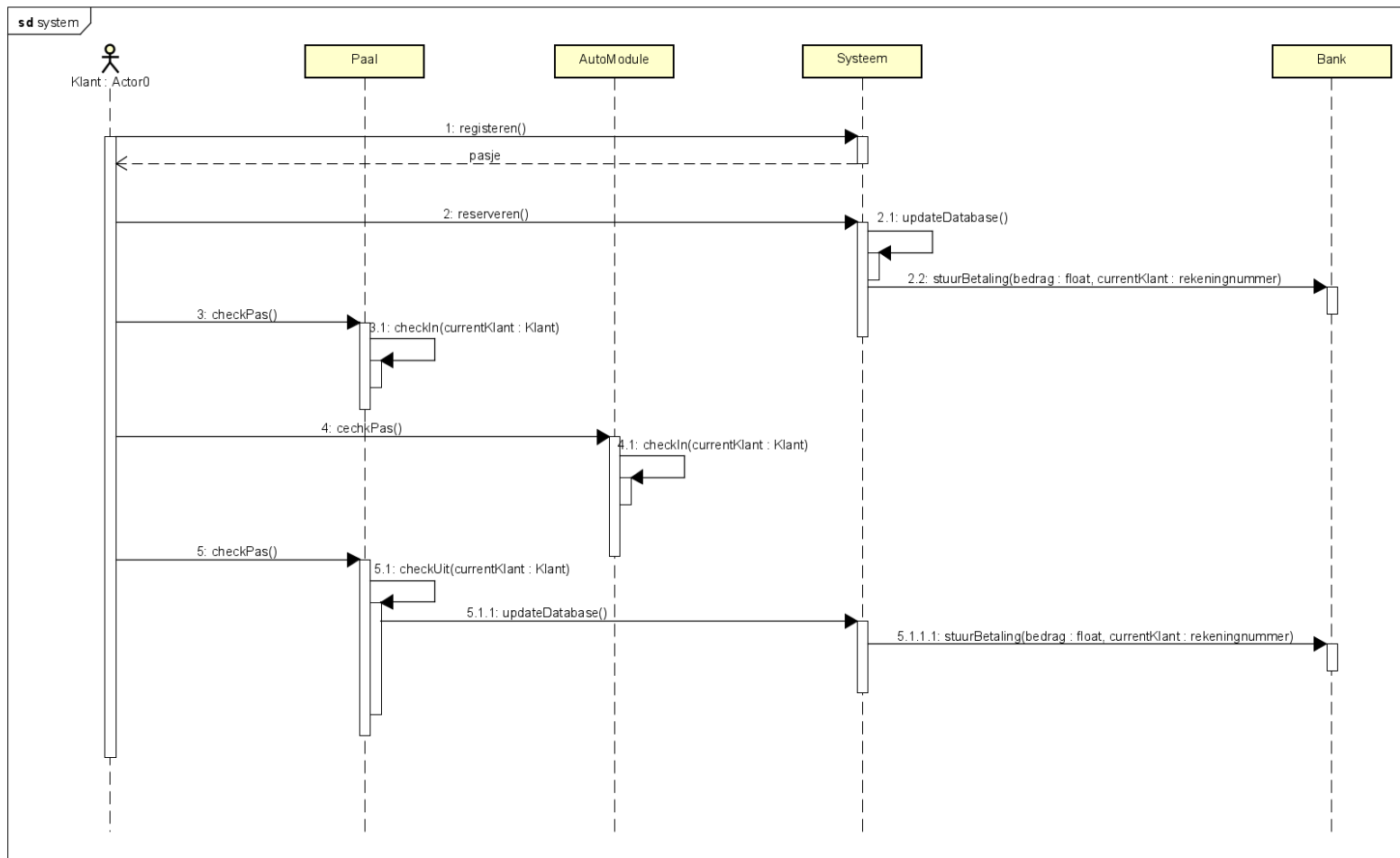
3 Use-case Descriptions

Hieronder het usecase diagram van het RedCarsSysteem.



3.0.1 System sequence diagram

Hieronder ziet u een sequence diagram van het gehele systeem. Onder de uitgebreidere uitleg van de usecases zijn de relevante interacties gemarkeerd. Niet alle interacties zijn in dit globale diagram opgenomen, deze zijn uitgewerkt in de component sequence diagrammen. Deze zijn te vinden in het SDD hoofdstuk 3.2.



3.1 Brief Descriptions

3.1.1 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.

3.1.2 Betaling van de huur (bij reservering)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.

3.1.3 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.

3.1.4 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

3.1.5 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 weggrijden.

3.1.6 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.

3.1.7 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 medewerker 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.

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.

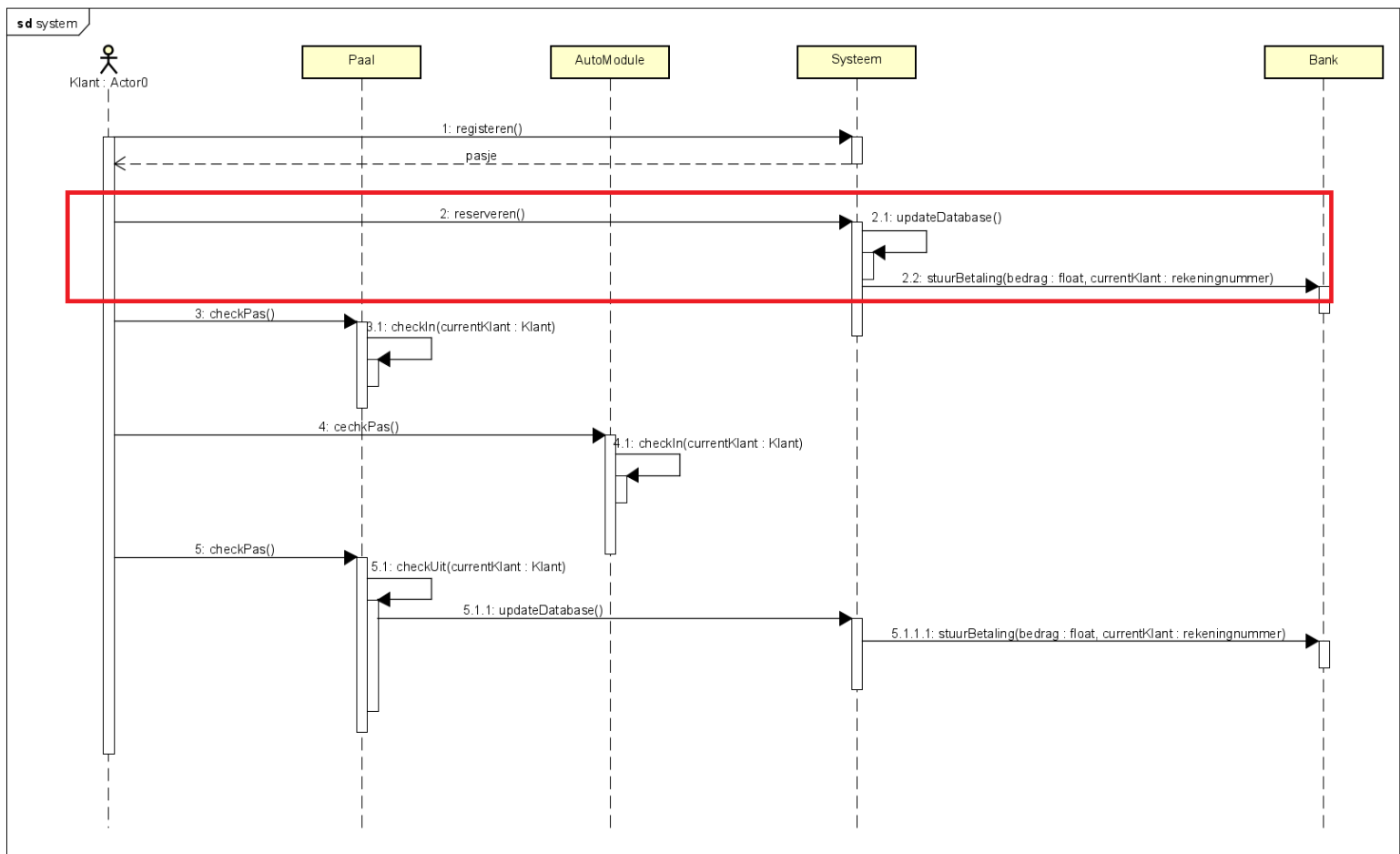
3.2 Reserveren

3.2.1 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 gereserveerde 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.	2. Het systeem laat de beschikbare auto's zien in de opgegeven stad.
3. De klant geeft het type auto aan.	4. Het systeem laat de beschikbare auto's zien van dat type.*
5. De klant geeft aan tijdens welke periode hij de auto wil huren.	6. Het systeem laat de tijdens de opgegeven tijdperiode beschikbare auto's zien.*
7. De klant kiest een auto die voldoet aan de opgegeven specificaties.	8. Het systeem berekend de prijs en toont deze.
9. De klant bevestigt de huurovereenkomst.	10. 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 opgegeven specificaties.

3.2.2 System Sequence Diagram (optional)



De relevante interacties passend bij deze usecase zijn te zien binnen het rode rechthoek.

3.3 Betaling van de huur (bij reservering)

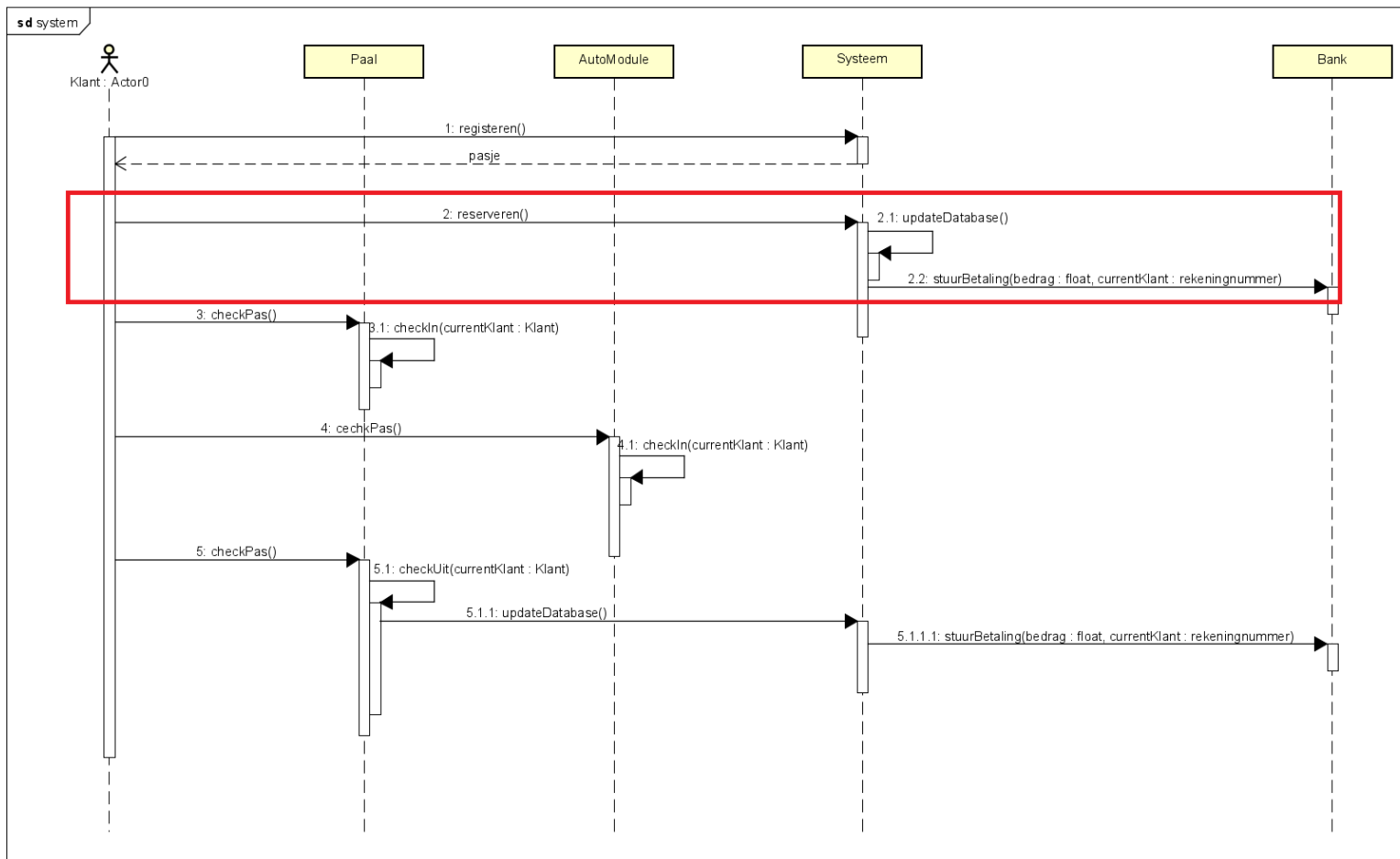
3.3.1 Fully dressed use case description

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.

3.3.2 System Sequence Diagram (optional)



De relevante interacties passend bij deze usecase zijn te zien binnen het rode rechthoek. Aangezien het betalen van de reservering erg nauw ligt met reserveren komen de afbeeldingen overeen.

3.4 Use Case: Terugbrengen van auto

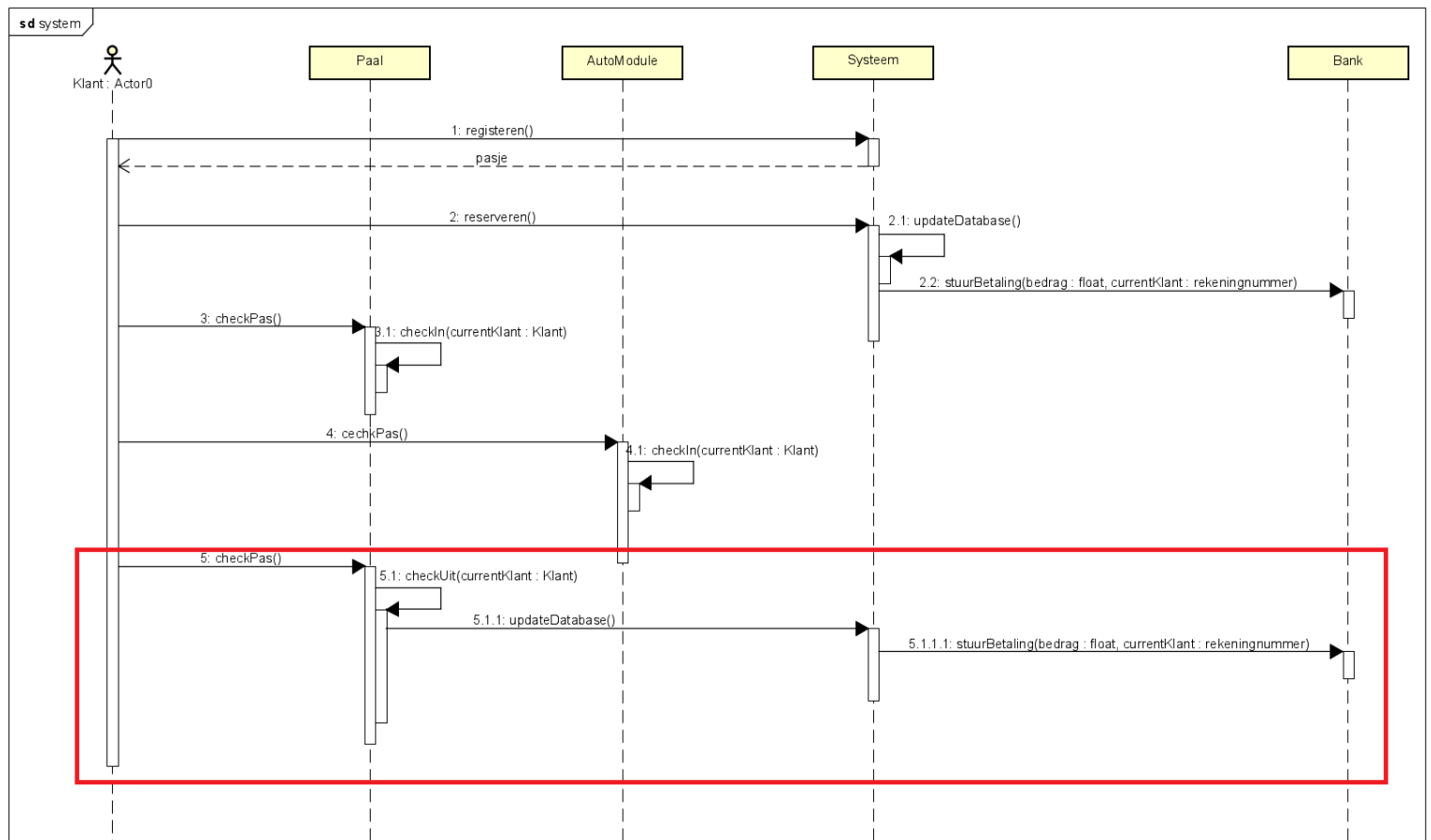
3.3.1 Fully dressed use case description

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.

3.4.1 System Sequence Diagram (optional)



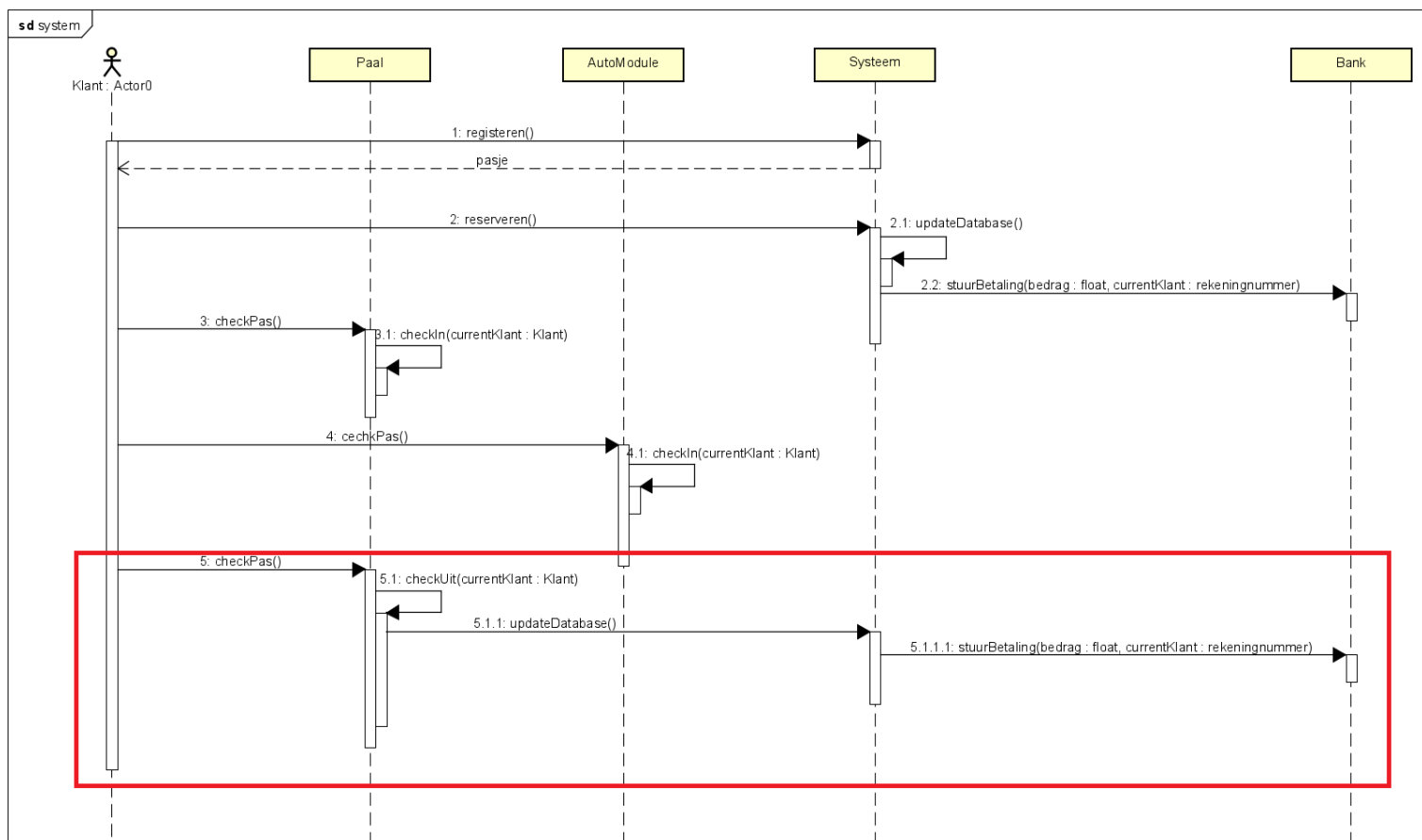
3.5 Betaling van de huur (bij terugbrengen van de auto)³

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.	

3.5.1 System Sequence Diagram (optional)



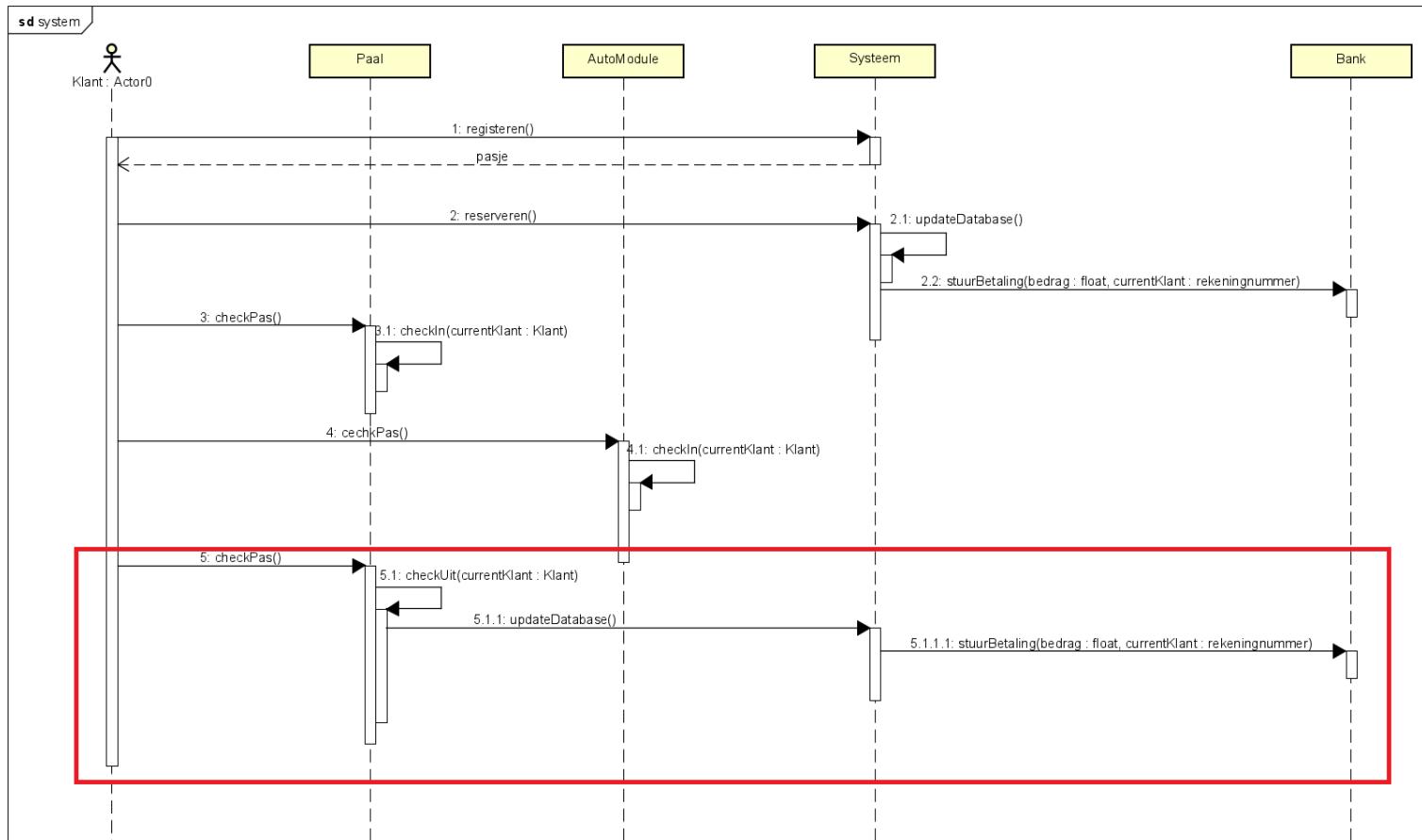
3.6 Ophalen van auto

3.6.1 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 hem 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.

3.6.2 System Sequence Diagram (optional)



4 Requirements

4.1 Functionality:

- F.1 Een klant kan zich registreren, dit doet hij door zijn naam, adres, woonplaats, e-mailadres en bankrekeningnummer op te geven.
- F.2 Na registreren krijgt de klant een persoonlijke pas thuis gestuurd.
- F.3 De klant kan inloggen met e-mailadres en pasnummer.
- F.4 De klant kan zijn klantgegevens beheren.
- F.5 RentIt-medewerkers kunnen klantgegevens beheren.
- F.6 RentIt-medewerkers kunnen klanten op inactief zetten.
- F.7 RentIt-medewerkers kunnen autogegevens beheren.
- F.8 Op de website kan de klant een overzicht zien van alle beschikbare auto's die voldoen aan de opgegeven specificaties (stad, tijdsduur, soort auto en specifieke parkeerlocatie).
- F.9 De klant heeft de mogelijkheid om te kiezen tussen een betaald of gratis abonnement.
- F.10 Een klant kan alleen een auto reserveren als de klant geen betalingsachterstand heeft.
- F.11 Een klant kan alleen een auto reserveren als hij niet op inactief is gezet vanwege misbruik.
- F.12 De klanten betalen na reservering via automatische afschrijving.
- F.13 Het is niet mogelijk voor klanten om meerdere auto's in dezelfde tijdperiode te reserveren.
- F.14 Bij het uitchecken worden eventuele boetes berekend. En toegevoegd aan de openstaande betalingsachterstanden.
- F.15 Bij het uitchecken wordt de prijs van het aantal gereden km's opgeteld bij eventuele betalingsachterstanden. Van dit bedrag wordt een automatische afschrijving naar de klant gestuurd.
- F.16 Een klant kan alleen de auto inchecken die hij heeft gereserveerd.
- F.17 Ten alle tijden is de GPS-tracking module in de auto ingeschakeld.

- F.18 Een klant kan alleen uitchecken bij dezelfde paal als waar hij de gehuurde auto heeft opgehaald.
- F.19 Bij het uitchecken van een auto worden autogegevens geüpdatet.
- F.20 Na het uitchecken wordt de eventueel resterende reserveringstijd voor de uitgecheckte auto verwijderd.
- F.21 De redcars auto en parkeerpaal geven lichtsignalen wanneer de auto stilstaat op een redcars parkeerplaats.

4.2 Usability:

- U.1 Wanneer de klant is vergeten uit te checken stuurt het systeem na 15 minuten een e-mail ter herinnering om uit te checken.
- U.2 Tijdens het reserveren krijgt de klant automatisch de goedkoopste optie die overeen komt met de door hem opgegeven specificaties.
- U.3 De website is responsief opgebouwd zodat de klant feedback krijgt op de acties die hij uitvoert.

4.3 Reliability:

- R.1 Bij het afronden van een reservering van een auto wordt de auto direct op niet beschikbaar gezet. Hij verschijnt niet langer op de website. Wanneer een andere klant de gereserveerde auto probeert te reserveren krijgt hij een melding dat deze auto reeds gereserveerd is.
- R.2 Wanneer een automatische betaling is mislukt wordt de klant hiervan op de hoogte gesteld door middel van een e-mail.
- R.3 De functionaliteit en betrouwbaarheid van de paal die buiten staat op de parkeerlocatie is niet gevoelig voor weersomstandigheden.

4.4 Performance:

- P.1 Het scannen van het pasje tijdens het in- en uitchecken moet binnen 1000ms worden afgehandeld door het systeem.
- P.2 De database server heeft genoeg opslag voor 5.000.000 gebruikers.

4.5 Supportability:

- S.1 De website wordt ondersteund op verschillende platvormen (computer / mobiel)

- S.2 Het systeem wordt modulair opgezet.
- S.3 De database wordt opgebouwd in SQL.