

Functioneel Ontwerp

Loan System

TCIF-VKFEP-17, Front-End Programming

Geschreven door

Luuk van houdt, Sjard Willems en Matthijs Kemp

Inhoudsopgave

1 Inleiding	2
2 Backlog	3
2.1 Sprints	3
2.2 Onderverdeling	3
2.2.1 Bijdrage van Luuk	3
2.2.2 Bijdrage van Sjar	3
2.2.3 Bijdrage van Matthijs	4
3 Life-cycle	5
4 Architectonische significantie	6
4.1 Architectonische significante use cases	6
4.2 Architectonische insignificante use cases	7
5 Use case diagram	8
6 Volledigheid analyse	9
6.1 Attribuut check	9
6.2 Functionaliteiten check	10
6.3 conclusie	10
7 Actoren	11
8 Sitemap	12
9 Use case templates	13
9.1 Aanmaken	13
9.2 Reserveren	13
9.3 Uitlenen	14
9.4 Innemen	16
9.5 Niet ophalen	17
9.6 Repareren	18
10 Technische requirements	19

1 Inleiding

Bij het vak Front End Development werd ons gevraagd om een uitleensysteem te digitaliseren en verbeteren d.m.v. het analyseren van het bedrijfsproces. Het Functioneel Ontwerp (hierna FO) is de basis voor de applicatie. Tijdens het analyseren van de casus hebben we vastgesteld dat het leen-object het kern-object is.

Zoals uit het bovenstaande diagram opgemaakt kan worden bestaat het hoofdprocess in de casus uit: het beschikbaar stellen, reserveren en uitlenen van het leen object.

2 Backlog

2.1 Sprints

Aangezien de ontwikkelperiode 3 weken is hebben we besloten om 3 sprints te definiëren.

In de tabel hieronder staan de drie sprints die we hebben gedefinieerd.

Nummer	Weeknr.	Beschrijving
1	42	Basis code
2	43	Architectonisch significante use cases
3	44	Architectonisch insignificante use cases

2.2 Onderverdeling

2.2.1 Bijdrage van Luuk

Onderdeel	Bijdrage
Use cases	<ul style="list-style-type: none">- Innemen- Niet ophalen
Diagrammen	<ul style="list-style-type: none">- State diagram- Use case diagram- Sitemap
Ontwikkelstraat	<ul style="list-style-type: none">- Firebase- Github- Trello

2.2.2 Bijdrage van Sjard

Onderdeel	Bijdrage
Use cases	<ul style="list-style-type: none">- Repareren- Reserveren
Diagrammen	<ul style="list-style-type: none">- State diagram

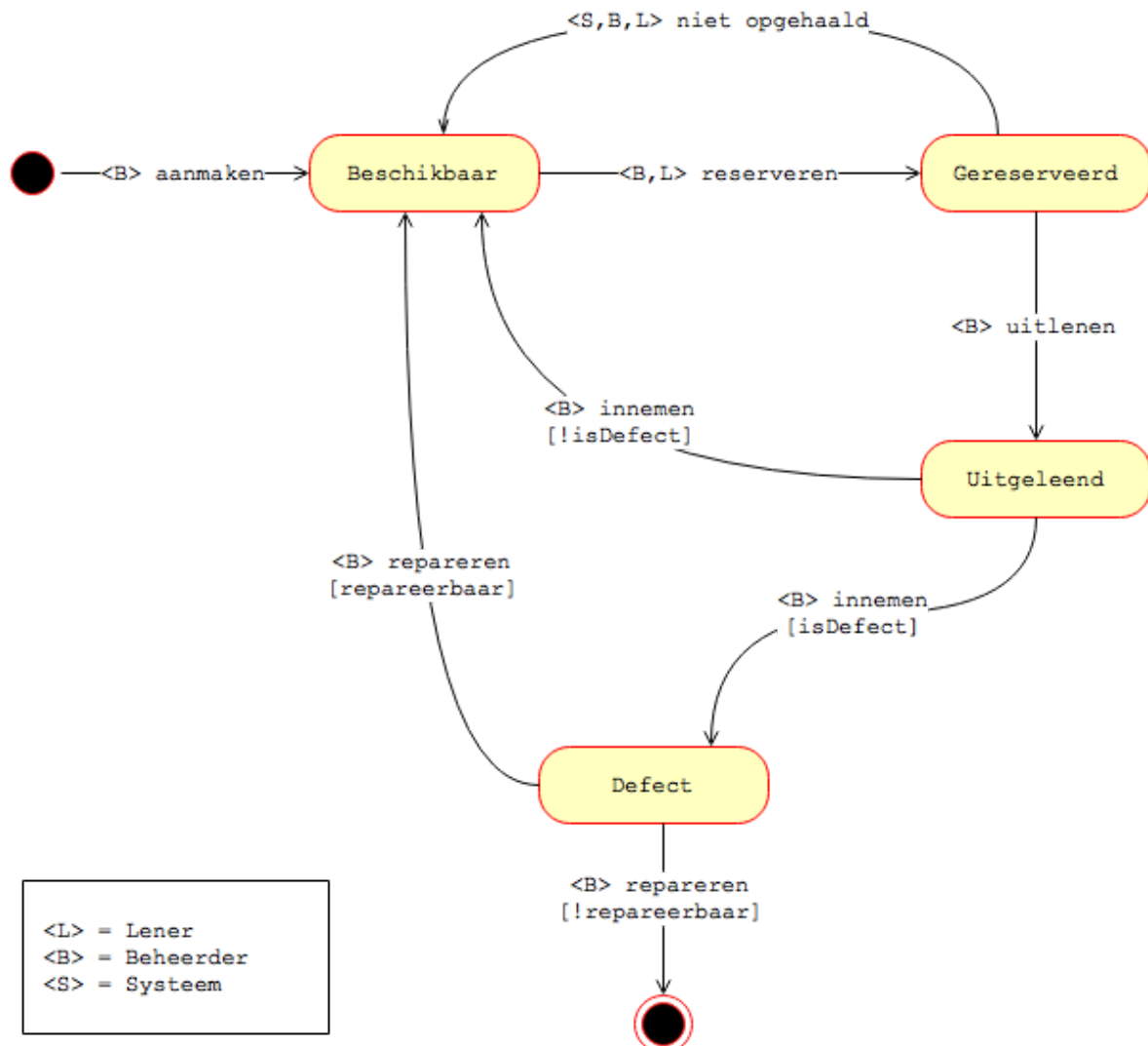
	- Use case diagram
Functioneel ontwerp	- Attriboot check

2.2.3 Bijdrage van Matthijs

Onderdeel	Bijdrage
Use cases	- Aanmaken - Uitlenen
Diagrammen	- State diagram - Use case diagram
Functioneel ontwerp	- Attriboot check - Actor templates

3 Life-cycle

Het is de taak van de beheerder om leen-objecten toe te voegen of te verwijderen, daarnaast kan de beheerder objecten uitlenen en innemen. Leners kunnen de leenobjecten reserveren en reserveringen annuleren. Als we dit vertalen naar een toestandsdiagram krijg je het volgende:



4 Architectonische significantie

Tijdens het opstellen van de use cases hebben we gekeken naar welke use case wel en welke niet architectonische significant waren.

4.1 Architectonische significante use cases

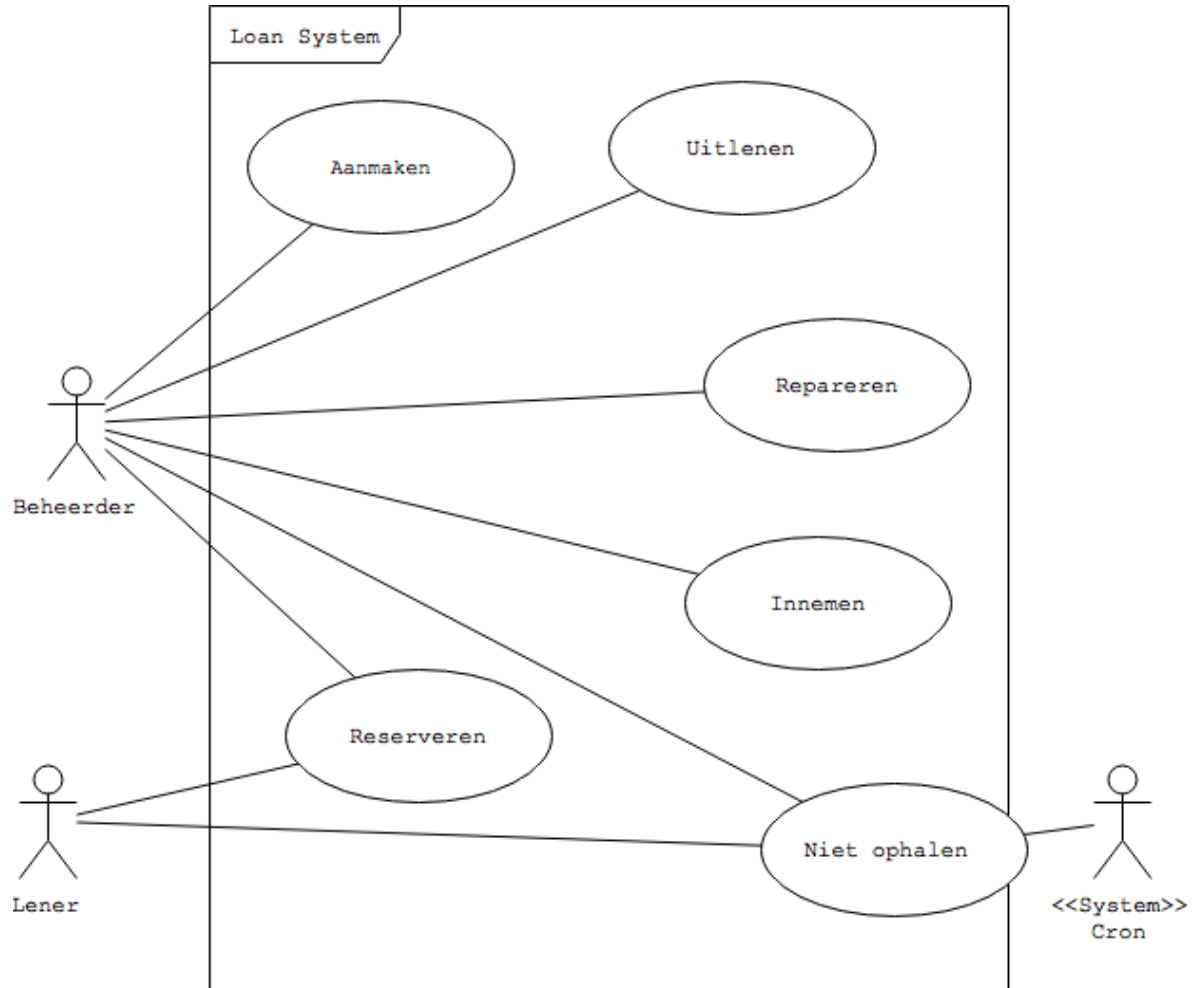
Use Case	Samenvatting
Aanmaken	Elk item moet voordat het kan worden uitgeleend eerst worden aangemaakt in het systeem. De actor voert het <u>item type</u> [in] en het aantal [in].
Reserveren	<p>De reserverings pagina toont de volgende velden:</p> <ul style="list-style-type: none">- Ophaal code [out]- Lenernaam [out]- Item type [in]- Ophaal datum [in]- Inlever datum [in]- Notes [in] <p>De actor vult deze in en verzend het formulier en het systeem voegt de reservering toe. Het systeem stuurt een mail met reservatie informatie. De actor wordt verwezen naar de reserverings detailpagina.</p>
Uitlenen	De lener komt aan met zijn <u>ophaal code</u> [in]. De actor vult deze in en er verschijnt een reservering detailpagina. De actor pakt het item en voert de <u>code</u> [in]
Innemen	De actor ontvangt het item van de lener in met de unieke <u>item code</u> [in]. Met de code geeft de actor in het systeem aan dat het geleende item (reserveringsnummer [in]) weer is ingenomen.
Niet ophalen	<p>De actor bewerkt een item (<u>item code</u> [in]):</p> <ul style="list-style-type: none">• Wat de actor gereserveerd heeft (reserveringsnummer [in]) maar achteraf besluit dat de actor het item niet meer komt ophalen.• Wat een lener gereserveerd heeft (reserveringsnummer [in]) en waarvan de actor heeft begrepen dat het item niet meer zal worden opgehaald.• Wat een lener gereserveerd heeft (reserveringsnummer [in]) en waarvan de <u>inlever datum</u> [out] van de reservering de volgende dag valt. Dit doet de actor elke dag op de vastgestelde tijdstippen.
Repareren	Het systeem toont item detailpagina de actor veranderd de <u>status</u> [in] van het item naar "onbeschikbaar".

4.2 Architectonische insignificante use cases

Use Case	Samenvatting
Inloggen	Een anonieme gebruiker van de applicatie geeft zijn/haar <u>gebruikersnaam</u> [in] en <u>wachtwoord</u> [in] op, waarmee het systeem controleert of de gegevens aan een <u>geregistreerde gebruiker</u> [out] toebehoren.
Registreren	Een anonieme gebruiker van de applicatie geeft zijn/haar <u>e-mailaccount naam</u> [in], <u>e-mailaccount domeinnaam</u> [in], <u>studentnummer</u> [in], <u>voornaam</u> [in], <u>achternaam</u> [in], <u>wachtwoord</u> [in] en een <u>wachtwoord bevestiging</u> [in] op, waarmee het systeem de gegevens valideert en gebruikt voor het aanmaken van een <u>nieuwe gebruiker</u> [out].
Wachtwoord herstellen	Een anonieme gebruiker van de applicatie geeft zijn/haar <u>e-mailaccount naam</u> [in] en <u>e-mailaccount domeinnaam</u> [in] op, waarmee het systeem controleert of de gegevens toebehoren aan een <u>gebruiker</u> [out]. Vervolgens maakt het systeem een <u>token</u> [out] aan waarmee de gebruiker zijn wachtwoord mee kan bewerken. De actor ontvangt de token per mail en activeert de token met de <u>activatie code</u> [in]. De gebruiker geeft zijn <u>nieuwe wachtwoord</u> [in] en een <u>wachtwoord bevestiging</u> [in] op. Het systeem past het wachtwoord van de gebruiker aan en authenticceert de gebruiker.
Niet ingenomen	De actor annuleert de reservering en verandert de <u>status</u> [in] van het geleende item naar onbeschikbaar.

5 Use case diagram

Op basis van de analyse van de lifecycle kun je use cases afleiden. Deze hebben wij in een use case diagram gezet dat er zo uit ziet:



6 Volledigheid analyse

Op basis van de informatie die we in bovenstaande hoofdstukken hebben ontdekt kunnen we de eerste check op volledigheid gaan uitvoeren.

6.1 Attribuut check

We gaan hier elk attribuut dat we uit de use case samenvattingen halen onderzoeken. In dit onderzoek willen we kijken in welke use case er een waarde aan het attribuut wordt toegekend en in welke use case het attribuut wordt gebruikt. Zo kunnen we achterhalen of ons systeem 'magic' en/of 'black hole' attributen bevat.

Attribuut	Use Case waar het attribuut een waarde krijgt	Use case die de waarde(n) van het attribuut toont
Item type	<ul style="list-style-type: none">- Aanmaken- Reserveren	<ul style="list-style-type: none">- Uitlenen
Aantal	<ul style="list-style-type: none">- Aanmaken	
Item code	<ul style="list-style-type: none">- Aanmaken	<ul style="list-style-type: none">- Uitlenen- Niet ophalen- Innemen
Ophaal code	<ul style="list-style-type: none">- Uitlenen	<ul style="list-style-type: none">- Reserveren
Lenernaam	<ul style="list-style-type: none">- Reserveren	<ul style="list-style-type: none">- Reserveren
Ophaaldatum	<ul style="list-style-type: none">- Reserveren	<ul style="list-style-type: none">- Uitlenen
Inlever datum	<ul style="list-style-type: none">- Reserveren	<ul style="list-style-type: none">- Niet ophalen
Notes	<ul style="list-style-type: none">- Reserveren	<ul style="list-style-type: none">- Uitlenen
Reserveringsnummer	<ul style="list-style-type: none">- Reserveren	<ul style="list-style-type: none">- Niet ophalen- Uitlenen- Innemen

6.2 Functionaliteiten check

Uit ons Use Case diagram zien we dat er 3 actoren zijn, waarvan 1 het systeem zelf is. Dit vereist dat het systeem moet weten door welke actor het bediend wordt.

Hier zijn 2 oplossingen voor; 2 aparte dedicated systemen of een authenticatie. Omdat authenticatie de makkelijkere implementatie is, zullen we deze oplossing kiezen.

Authenticatie wordt geregeld door de use case 'inloggen'.

We kiezen dus voor het toevoegen van een use case genaamd 'inloggen'. Deze use case maakt deel uit van de superuser 'anonieme gebruiker'. Deze use case hebben we bewust niet opgenomen in het use case diagram.

Door het toevoegen van de use case inloggen weten we nog niet welke rol een gebruiker heeft. Dit is op te lossen door voor elke bekende gebruiker de rol hardcoded in het systeem te implementeren.

6.3 conclusie

De use case 'inloggen' wordt aan het systeem toegevoegd en de users alsmede hun rollen worden nu hardcoded geïmplementeerd.

7 Actoren

Uit het Use Case Diagram van het vorige hoofdstuk kunnen de volgende actoren worden afgeleid: Lener en Beheerder.

Ons systeem heeft ook een 3e actor genaamd systeem, deze regelt het bij het niet ophalen de meldingen naar de beheerder en het terugplaatsen naar beschikbaar van het leen-object. Omdat deze actor een systeem is en geen persoon zal hiervoor geen template worden uitgewerkt.

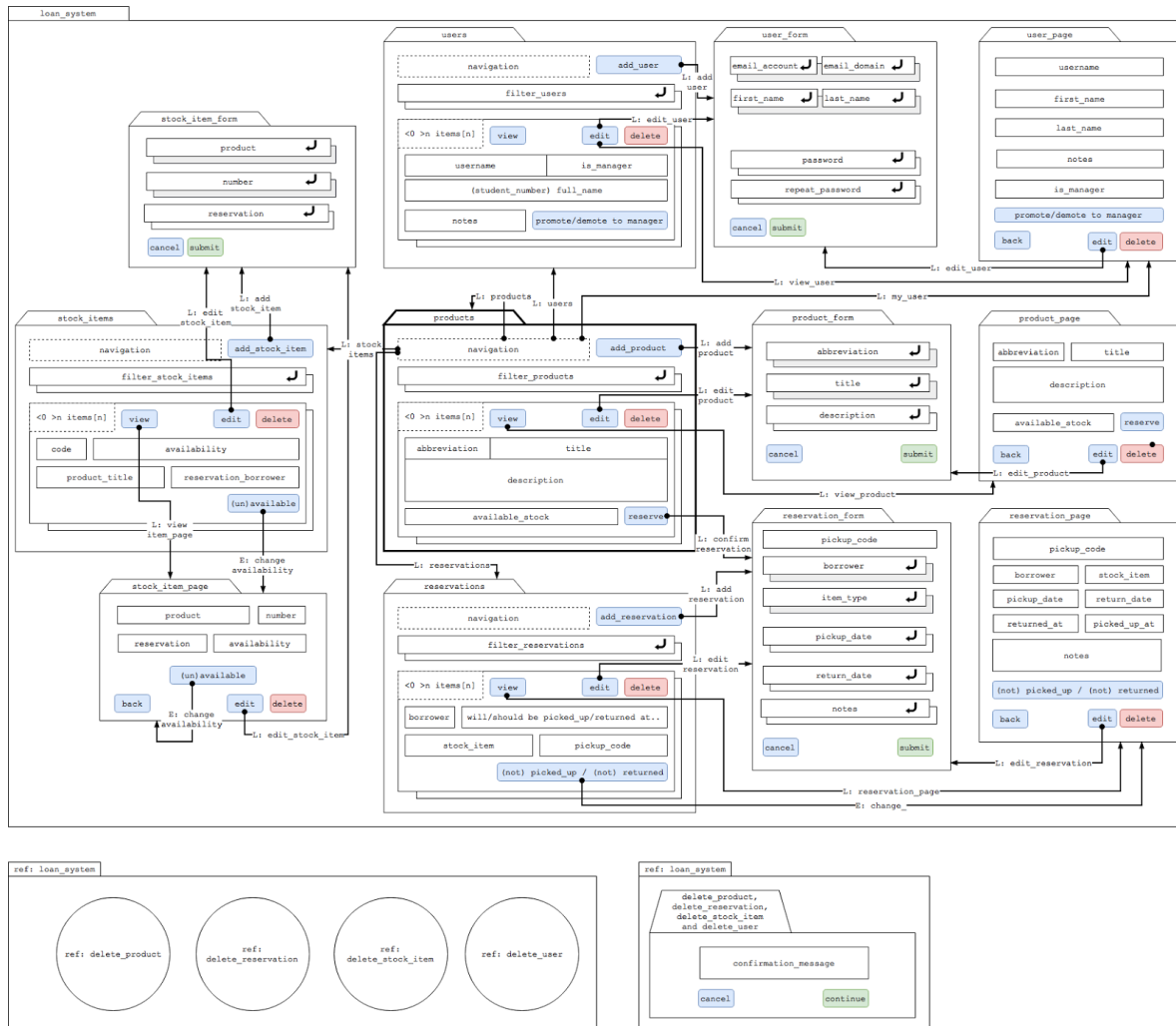
De templates voor de actoren zijn als volgt:

Project	Loansystem
Rolnaam	Lener
Rolbeschrijving	De lener is de persoon die een leen-object wilt lenen. De lener zal een reservering plaatsen, waarna hij kan beslissen om deze reservering op te halen of te annuleren.
Personen	Alle HU studenten en Docenten zijn potentiële leners

Project	Loansystem	
Rolnaam	Beheerder	
Rolbeschrijving	De beheerder is de persoon die de leen-objecten toe kan voegen of kan verwijderen uit het systeem. Tevens is dit de persoon die de leen-objecten uitgeeft/inneemt aan/van leners. De beheerder is ook de persoon die het laatste woord heeft in zaken betreffende schade en reparatie.	
Personen		
	Naam:	Tim le Febre
	Foto:	
	Email:	Tim.leFebre@hu.nl
	Functie:	SNE lab beheerder
	Aantekeningen:	

8 Sitemap

Op basis van de Use Cases welke in de volgende hoofdstukken worden behandeld, hebben wij een sitemap opgesteld. De sitemap die wij hebben gemaakt is conform de richtlijnen van UiaML (zie: <http://www.uiaml.org/>)



9 Use case templates

9.1 Aanmaken

Use Case Nummer	1
Use Case Naam	Aanmaken
Actor	Beheerder
Samenvatting	Elk item moet voordat het kan worden uitgeleend eerst worden aangemaakt in het systeem. De actor voert het <u>item type</u> [in] en het aantal [in].
Preconditie	Er is een item type
Main Scenario	<ol style="list-style-type: none">a. De actor selecteert het <u>item type</u> [in] b. De actor voert het <u>aantal</u> [in]De actor verstuurd de invoerHet systeem bevestigt geslaagde aanmaak en geeft voor elk item een <u>code</u> [out]
Postconditie (MS)	Nieuw(e) item(s) toegevoegd aan het systeem

9.2 Reserveren

Use Case Nummer	2
Use Case Naam	Reserveren
Actor	Lener, Beheerder
Samenvatting	<p>De reserverings pagina toont de volgende velden:</p> <ul style="list-style-type: none">- Ophaal code [out]- Lenernaam [out]- Item type [in]- Ophaal datum [in]- Inlever datum [in]- Notes [in] <p>De actor verzendt het formulier en het systeem voegt de reservering toe. Het systeem stuurt een mail met reservatie informatie. De actor wordt verwezen naar de reserverings detailpagina.</p>
Preconditie	-

Main Scenario	<ol style="list-style-type: none"> 1) Het systeem toont: <ol style="list-style-type: none"> a) Ophaal code [out] b) Lenernaam [out] c) Item type [in] d) Ophaal datum [in] e) Inlever datum [in] f) Notes [in] 2) De actor submit de form 3) Systeem stuurt een mail met reserverings informatie 4) Actor wordt doorverwezen naar de reserverings detailpagina 5) Er word een reservering toegevoegd 				
Postconditie (MS)	Er is een reservering toegevoegd aan het systeem				
Alternatief Scenario (A1)	<table border="1"> <tr> <td>Trigger</td><td>De actor is een beheerder</td></tr> <tr> <td>Scenario</td><td> <ol style="list-style-type: none"> 1) Het systeem toont: <ol style="list-style-type: none"> a) Ophaal code [out] b) Lenernaam [in] c) Item type [in] d) Ophaal datum [in] e) Inlever datum [in] f) Notes [in] 2) De actor submit de form 3) Systeem stuurt een mail met reserverings informatie 4) Actor word doorverwezen naar de reserverings detailpagina 5) Er word een reservering toegevoegd </td></tr> </table>	Trigger	De actor is een beheerder	Scenario	<ol style="list-style-type: none"> 1) Het systeem toont: <ol style="list-style-type: none"> a) Ophaal code [out] b) Lenernaam [in] c) Item type [in] d) Ophaal datum [in] e) Inlever datum [in] f) Notes [in] 2) De actor submit de form 3) Systeem stuurt een mail met reserverings informatie 4) Actor word doorverwezen naar de reserverings detailpagina 5) Er word een reservering toegevoegd
Trigger	De actor is een beheerder				
Scenario	<ol style="list-style-type: none"> 1) Het systeem toont: <ol style="list-style-type: none"> a) Ophaal code [out] b) Lenernaam [in] c) Item type [in] d) Ophaal datum [in] e) Inlever datum [in] f) Notes [in] 2) De actor submit de form 3) Systeem stuurt een mail met reserverings informatie 4) Actor word doorverwezen naar de reserverings detailpagina 5) Er word een reservering toegevoegd 				
Postconditie (A1)	Er is een reservering toegevoegd in het systeem.				

9.3 Uitlenen

Use Case Nummer	3
Use Case Naam	Uitlenen
Actor	Beheerder
Samenvatting	De lener komt aan met zijn <u>ophaal code</u> [in]. De actor vult deze in en er verschijnt een reservering detailpagina. De actor pakt het item en voert de <u>code</u> [in]
Preconditie	De lener heeft een ophaal code.
Main Scenario	<ol style="list-style-type: none"> 1. De actor voert de <u>ophaal code</u> [in]

	<ol style="list-style-type: none"> 2. Het systeem toont de reservering detailpagina die bij de ophaal code hoort. Deze pagina bevat de <u>lener naam</u> [out], <u>item type</u> [out], <u>ophaaldatum</u> [out], <u>inleverdatum</u> [out] en <u>notes</u> [out]. 3. De actor voert de <u>code</u> [in] van het item dat hij gaat meegeven aan de lener in.
Postconditie (MS)	Het item heeft als status "uitgeleend" gekregen

9.4 Innemen

Use Case Nummer	4				
Use Case Naam	Innemen				
Actor	Beheerder				
Samenvatting	De actor ontvangt het item van de lener in met de unieke <u>item code</u> [in]. Met de code geeft de actor in het systeem aan dat het geleende item (<u>reserveringsnummer</u> [out]) weer is ingenomen.				
Preconditie	De actor heeft over de unieke code.				
Main Scenario	<ol style="list-style-type: none"> 1. Het systeem toont de detailpagina van de reservering van het item wat relevant is voor de unieke code. 2. De actor geeft op de detailpagina van de reservering aan dat het item is ingenomen. 3. De actor controleert het item en geeft aan het systeem door dat het item na behoren functioneert. 				
Postconditie (MS)	Het systeem verhoogt het aantal beschikbare items van het relevante item type.				
Alternatief Scenario (A1)	<table border="1"> <tr> <td>Trigger</td><td>Het item is defect.</td></tr> <tr> <td>Scenario</td><td><i>Als vervolg van UC4.MS2:</i> De actor controleert het item en geeft aan het systeem door dat het item defect is. De actor neemt eventueel contact op met de lener.</td></tr> </table>	Trigger	Het item is defect.	Scenario	<i>Als vervolg van UC4.MS2:</i> De actor controleert het item en geeft aan het systeem door dat het item defect is. De actor neemt eventueel contact op met de lener.
Trigger	Het item is defect.				
Scenario	<i>Als vervolg van UC4.MS2:</i> De actor controleert het item en geeft aan het systeem door dat het item defect is. De actor neemt eventueel contact op met de lener.				
Postconditie (A1)	Het systeem verhoogt het aantal defecte items van het relevante item type.				

9.5 Niet ophalen

Use Case Nummer	5					
Use Case Naam	Niet ophalen					
Actor	Lener, Beheerder, Cron					
Samenvatting	<p>De actor bewerkt een item (<u>item code</u> [in]):</p> <ul style="list-style-type: none">• Wat de actor gereserveerd heeft (reserveringsnummer [in]) maar achteraf besluit dat de actor het item niet meer komt ophalen.• Wat een lener gereserveerd heeft (reserveringsnummer [in]) en waarvan de actor heeft begrepen dat het item niet meer zal worden opgehaald.• Wat een lener gereserveerd heeft (reserveringsnummer [in]) en waarvan de <u>inlever datum</u> [out] van de reservering de volgende dag valt. Dit doet de actor elke dag op de vastgestelde tijdstippen.					
Preconditie	De actor heeft een reservering die de actor.					
Main Scenario (MS)	<ol style="list-style-type: none">1. Het systeem toont de detailpagina van de reservering van de actor.2. De actor geeft aan dat de reservering geannuleerd kan worden.					
Postconditie (MS)	Het systeem verhoogt het aantal beschikbare items van het relevante item type.					
Alternatief Scenario (A1)	<table><tr><td>Trigger</td><td>De actor is een beheerder</td></tr><tr><td>Scenario</td><td>Het systeem toont de detailpagina van de reservering van een lener. Ga verder met UC5.MS2</td></tr></table>		Trigger	De actor is een beheerder	Scenario	Het systeem toont de detailpagina van de reservering van een lener. Ga verder met UC5.MS2
Trigger	De actor is een beheerder					
Scenario	Het systeem toont de detailpagina van de reservering van een lener. Ga verder met UC5.MS2					
Postconditie (A1)	Hetzelfde als postconditie UC4.MS					
Alternatief Scenario (A2)	<table><tr><td>Trigger</td><td>De actor is een cron systeem</td></tr><tr><td>Scenario</td><td>De actor voert aan het eind en aan het begin van de dag een commando uit waarmee gecontroleerd wordt of er nog reserveringen zijn die niet zijn opgehaald. Ofterwijl de actor controleert of er items zijn waarvan de periode</td></tr></table>		Trigger	De actor is een cron systeem	Scenario	De actor voert aan het eind en aan het begin van de dag een commando uit waarmee gecontroleerd wordt of er nog reserveringen zijn die niet zijn opgehaald. Ofterwijl de actor controleert of er items zijn waarvan de periode
Trigger	De actor is een cron systeem					
Scenario	De actor voert aan het eind en aan het begin van de dag een commando uit waarmee gecontroleerd wordt of er nog reserveringen zijn die niet zijn opgehaald. Ofterwijl de actor controleert of er items zijn waarvan de periode					

	<table border="1"> <tr> <td></td><td>waarin de lener het item kon ophalen is verstreken.</td></tr> </table>		waarin de lener het item kon ophalen is verstreken.
	waarin de lener het item kon ophalen is verstreken.		
Postconditie (A2)	Hetzelfde als postconditie UC4.MS		

9.6 Repareren

Use Case Nummer	6				
Use Case Naam	Repareren				
Actor	Beheerder				
Samenvatting	Het systeem toont item detailpagina de actor verandert de status van het item naar "onbeschikbaar".				
Preconditie	Het item kan gerepareerd worden				
Main Scenario (MS)	1) Het systeem toont de item detailpagina 2) De actor verandert de status van het item naar "onbeschikbaar"				
Postconditie (MS)	Het item is weer beschikbaar				
Alternatief Scenario (A1)	<table border="1"> <tr> <td>Trigger</td><td>Het item is niet repareerbaar</td></tr> <tr> <td>Scenario</td><td>1) Item wordt verwijderd uit het systeem</td></tr> </table>	Trigger	Het item is niet repareerbaar	Scenario	1) Item wordt verwijderd uit het systeem
Trigger	Het item is niet repareerbaar				
Scenario	1) Item wordt verwijderd uit het systeem				
Postconditie (A1)	Item is verwijderd uit het systeem				

10 Technische requirements

De applicatie dient te worden gerealiseerd als progressive web app. Voor data persistentie en authenticatie wordt gebruik gemaakt van een NoSQL database genaamd Firebase (<https://www.firebase.google.com>).

