

Front End Programming

# Lening-app

## Functioneel Ontwerp

Roy Hendriks, Rick Schoeman, Nick Schouten en Lucas Bos

13-10-2017

FEP		
<b>Document datum</b>	13-10-2017	
<b>Document versie</b>	V1.0	
<b>Versiebeheer</b>	V1.0	Geen Verandering

## Inhoudsopgave

1.	Inleiding .....	2
2.	Toestand diagrammen .....	3
3.	Use Cases.....	5
3.1	Samenvattingen .....	8
4.	Volledigheidsanalyse.....	9
4.1	Attributen check.....	9
4.2	Functionaliteiten check.....	10
4.3	Conclusie .....	10
5.	Actoren .....	11
6.	Use Case 1: Product toevoegen .....	12
6.1	Use Case Template .....	12
6.2	Sitemap .....	13
6.3	Wireframes .....	14
7.	Use Case 2: Lening indienen .....	15
7.1	Use Case Template .....	15
7.2	Sitemap .....	17
7.3	Wireframes .....	18
8.	Use Case 3: Lening opgehaald administreren.....	20
8.1	Use Case Template .....	20
8.2	Sitemap .....	21
8.3	Wireframe.....	22
9.	Use Case 4: Lening retourneren administreren.....	23
9.1	Use Case Template .....	23
9.2	Sitemap .....	24
9.3	Wireframe.....	25
10.	Sitemap .....	26
11.	Technische requirements .....	27

## 1. Inleiding

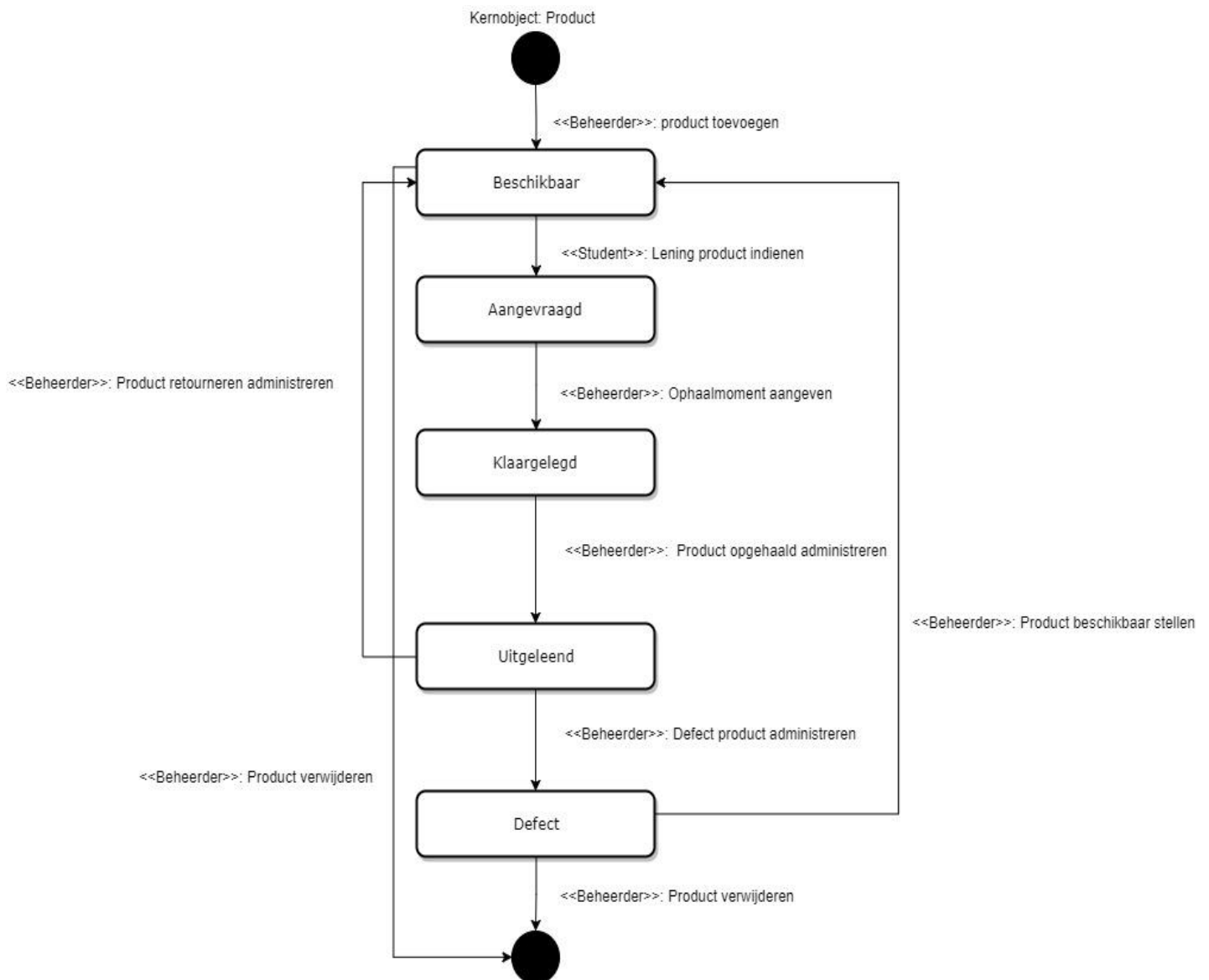
In het vak Front End Programming, is het de bedoeling dat er een app wordt gemaakt voor het bijhouden van de leningen in het IT-lab. Hier is op dit moment geen goed systeem voor en er moet dus een nieuw systeem komen. Het functioneel ontwerp (FO) in dit document is de basis voor het nieuwe systeem. Dit document zal dan ook gezien kunnen worden als een FO voor de eerste minimale versie van de te realiseren applicatie.

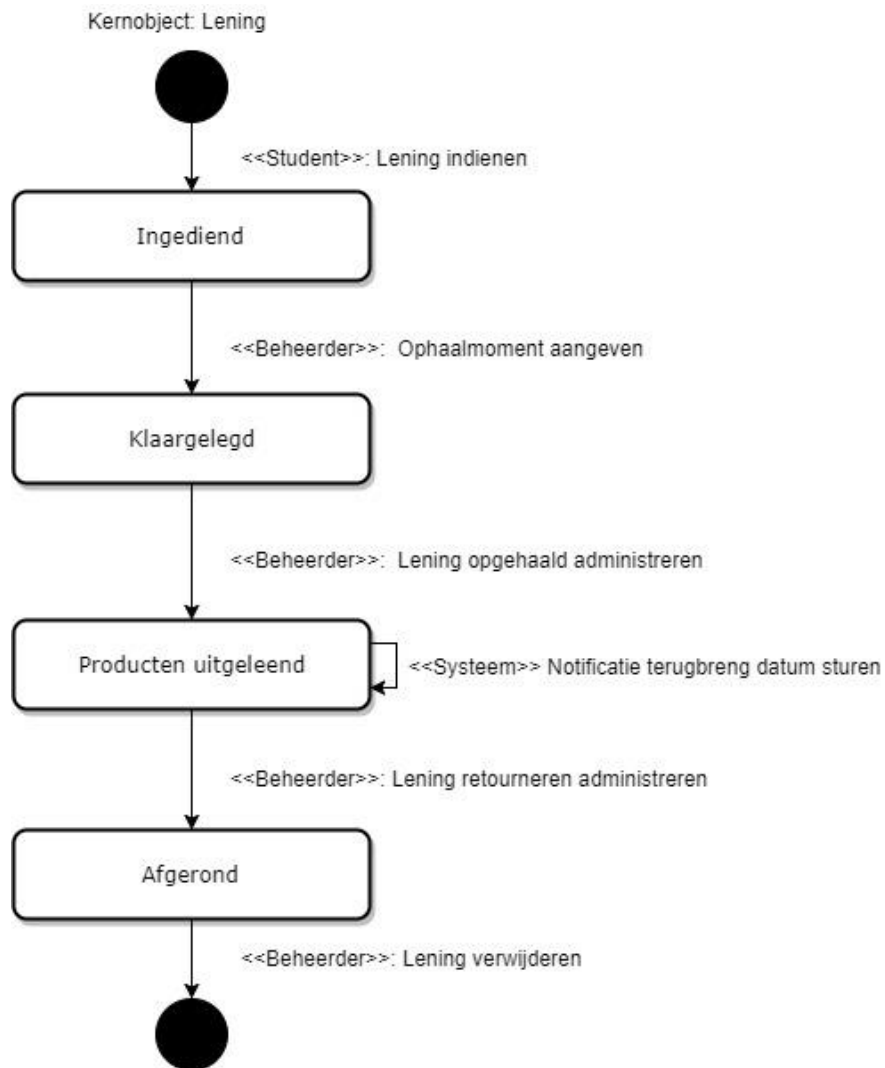
Dit FO beschrijft de gewenste werking van de app. Dit doen we door eerst de life-cycle (toestand diagram) van het kernobject in kaart te brengen en de Use Cases eruit af te leiden. Vervolgens zullen we de gevonden use cases een voor een beschrijven. Dit doen we aan de hand van een Use Case Template ondersteund door een of meerdere wireframes en een sitemap. De gegeven wireframes zijn van het niveau low-fidelity, wat inhoudt dat de schermen en hun dynamische content wel vast staan, de visualisatie nog open staat voor een verdere creatieve invulling.

## 2. Toestand diagrammen

De beheerder zal producten moeten toevoegen aan het systeem. Wanneer dit is gebeurd zullen studenten deze producten moeten kunnen lenen. Wanneer een product lening is ingediend zal de beheerder een moment aangeven waarop de student langs kan komen om het product op te halen. Als de student zijn lening heeft opgehaald of heeft teruggebracht zal de beheerder dit in het systeem aangeven.

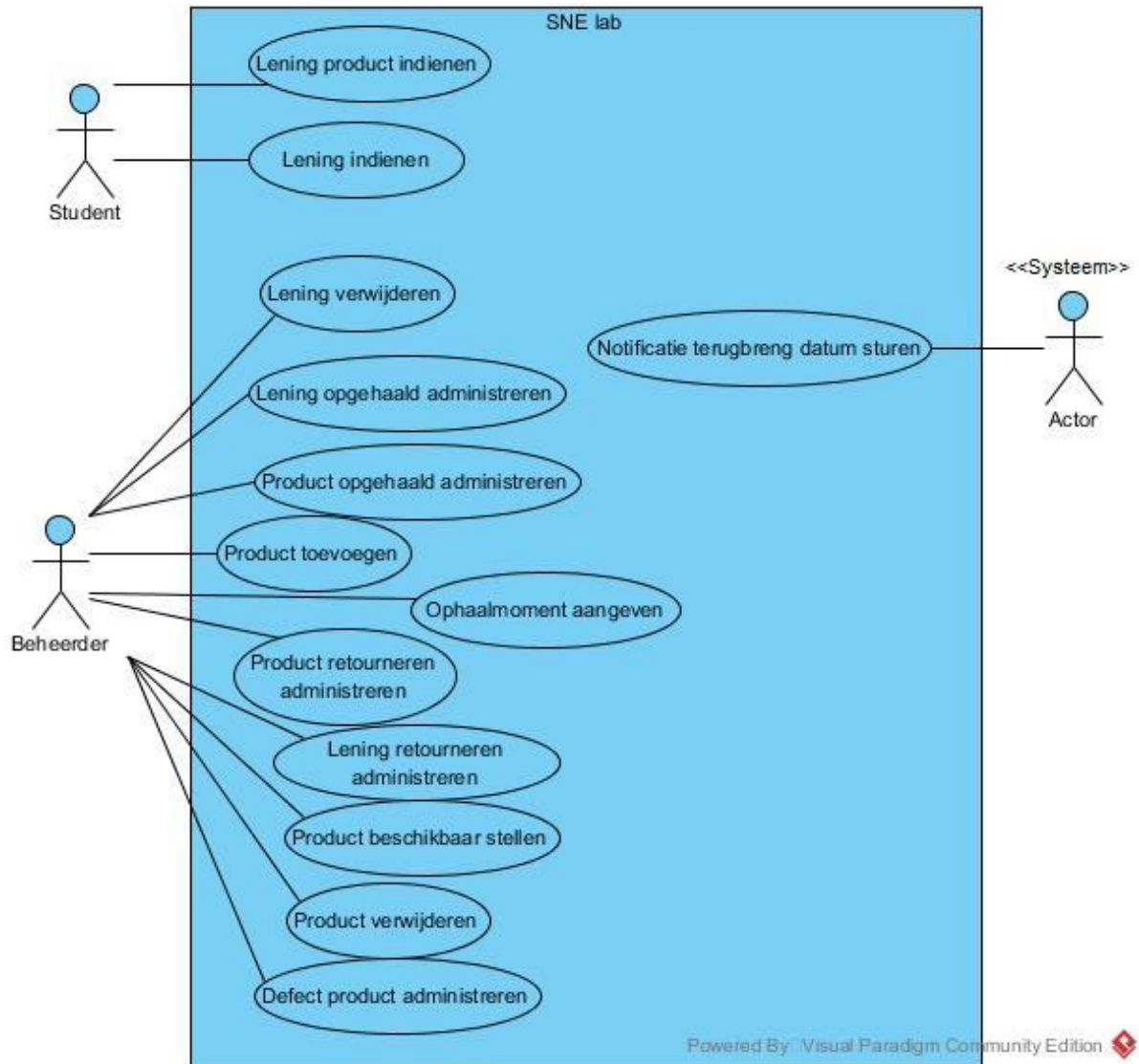
Als we dit vertalen naar een toestand diagram dan komen we tot de volgende modellen:



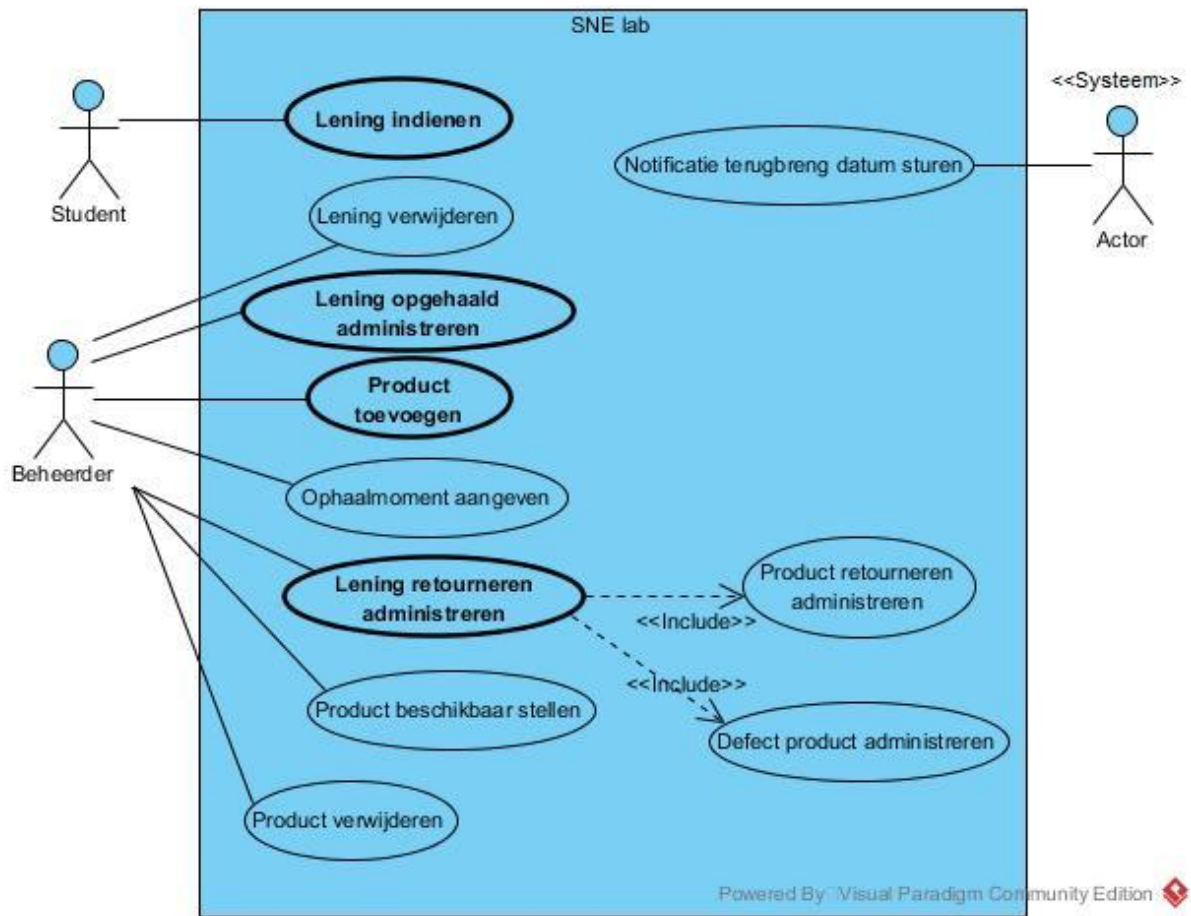


### 3. Use Cases

Op basis van de analyse van de toestandsdiagrammen kunnen we het volgende Use Case Diagram afleiden:

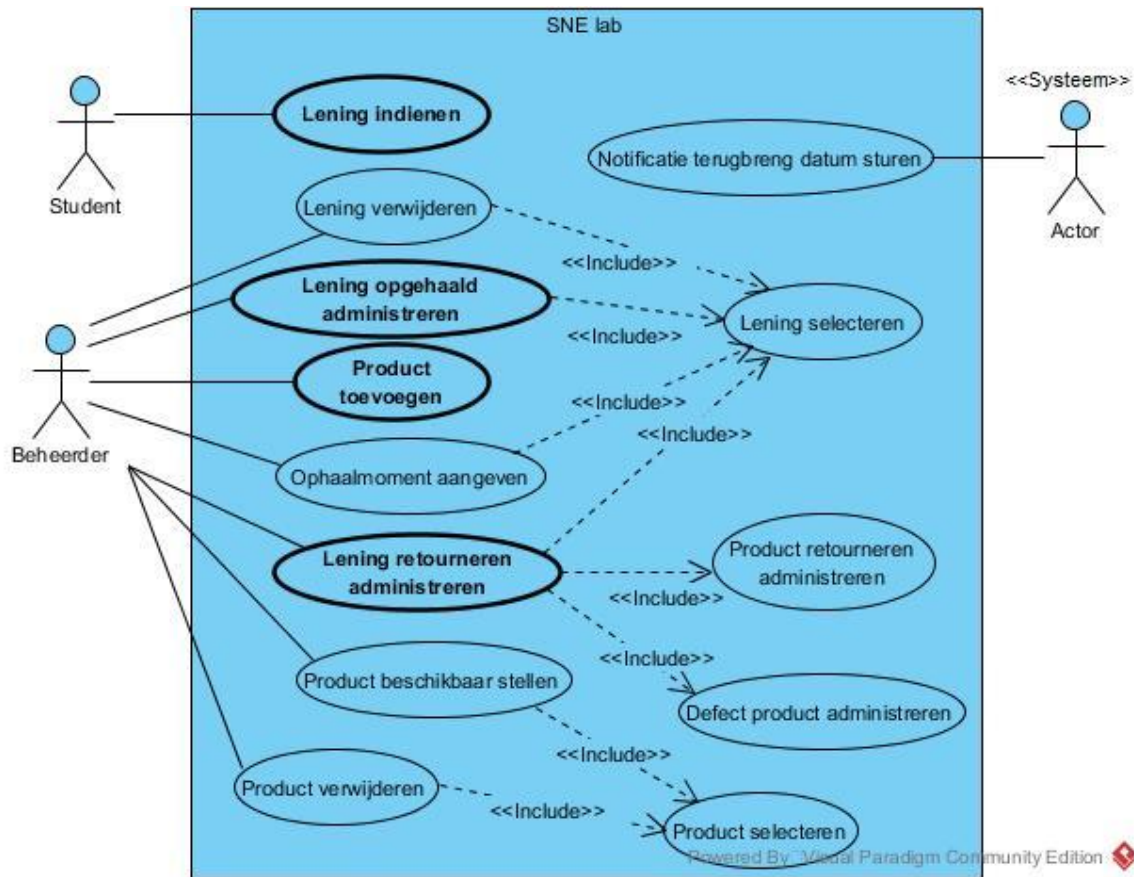


Omdat we twee toestand diagrammen hebben kunnen meerdere use cases worden samengevoegd omdat ze min of meer hetzelfde zijn alleen een andere naam hebben. Dit geeft het volgende use case diagram:



Wij zullen de dikgedrukte use cases verder uitwerken in dit document en ook als eerste realiseren, omdat hiermee een goede basis is gelegd voor de lening app.

Verder is het zo dat deze use cases gelden voor één lening of product dus moet voor sommige use cases eerst een product of lening worden geselecteerd. Daardoor ontstaat het volgende use case diagram:





### 3.1 Samenvattingen

Per use case kunnen we nu de samenvattingen opstellen:

Use Case	Samenvatting
<b>Lening indienen</b>	<p>De actor geeft aan d.m.v. <u>klascode [in]</u> voor welk vak hij producten wilt lenen. Vervolgens geeft hij aan uit een lijst met <u>productnaam[out]</u> met bijhorende <u>productbeschrijving[out]</u> en de <u>voorraad[out]</u> welke producten en het aantal per product hij wilt lenen.</p> <p>De actor kiest om de gekozen producten te lenen.</p> <p>Het systeem haalt a.d.h.v. het <u>studentnummer [out]</u> van de ingelogde actor de <u>voornaam[out]</u> en <u>achternaam[out]</u> op.</p> <p>Het systeem maakt de lening aan met <u>status[in]</u>'ingediend'. De lening bevat een <u>lening nummer[in]</u>, het <u>studentnummer [out]</u> van de actor, de gekozen <u>klascode[in]</u>, het huidige <u>blok [out]</u> en de <u>uitgeleende producten [in]</u>, per product het <u>productid[out]</u> en <u>aantal[in]</u>.</p> <p>Per <u>geleend product[in]</u> word de <u>status[in]</u> veranderd in 'Aangevraagd'.</p>
<b>Lening opgehaald administreren</b>	<p>Het systeem toont het <u>lening nummer[out]</u>, <u>studentnummer[out]</u>, de <u>voornaam[out]</u>, <u>achternaam[out]</u> en de lijst met uitgeleende <u>producten[out]</u> van de gekozen lening. Per product wordt de <u>productid[out]</u>, <u>productnaam[out]</u> en de <u>productomschrijving[out]</u> getoond.</p> <p>De actor geeft aan dat de lening is opgehaald.</p> <p>Het systeem verandert de <u>status[in]</u> van de lening in 'producten uitgeleend' en de <u>status[in]</u> van de producten naar 'uitgeleend'</p>
<b>Lening retourneren administreren</b>	<p>Het systeem toont het <u>lening nummer[out]</u>, <u>studentnummer[out]</u>, de <u>voornaam[out]</u>, <u>achternaam[out]</u> en de lijst met <u>uitgeleende producten[out]</u> van de gekozen lening. Per product wordt de <u>productid[out]</u>, <u>productnaam[out]</u> en de <u>productomschrijving[out]</u> getoond en er is een mogelijkheid om aan te geven of dit product <u>defect is[in]</u> of niet. De actor geeft aan per product of deze defect is of niet. Vervolgens kiest de actor om de wijzigingen op te slaan.</p> <p>Het systeem verandert de <u>status[in]</u> van de lening in 'Afgerond'.</p> <p>Per teruggebracht product word de <u>status[in]</u> veranderd in 'Beschikbaar', maar wanneer is aangegeven dat het product defect is word de <u>status[in]</u> 'Defect'.</p>
<b>Product toevoegen</b>	<p>De actor kan een nieuw product toevoegen, hiervoor moet hij een <u>productnaam[in]</u>, <u>productbeschrijving[in]</u> en aantal aan te maken producten geven. Het systeem maakt net zoveel producten aan als het gegeven aantal met <u>status[in]</u> 'Beschikbaar' en kent aan elk product een uniek <u>productID[in]</u> toe.</p>

## 4. Volledigheidsanalyse

Op basis van de bovenstaande informatie kunnen we een eerste check op de volledigheid van het beschreven systeem uitvoeren.

### 4.1 Attributen check

Uit de samenvattingen van de use cases kunnen we de attributen halen. Door te onderzoeken in welke use case er een waarde aan het attribuut wordt toegekend en in welke use case het attribuut wordt gebruikt, kunnen we achterhalen of onze beschrijving 'magic' en/of 'black hole' attributen bevat.

Attribuut	Use Case waar het attribuut gevuld wordt	Use Case die de waarde van het attribuut toont.
<b>Productnaam</b>	- Product toevoegen	- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>Productbeschrijving</b>	- Product toevoegen	- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>(product) Status</b>	- Product toevoegen - Lening indienen - Lening opgehaald administreren - Lening retourneren administreren	
<b>productID</b>	- Product toevoegen	- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>Klascode</b>	- Lening indienen	
<b>Voorraad</b>		- Lening indienen
<b>Studentnummer</b>		- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>Voornaam</b>		- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>Achternaam</b>		- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren
<b>(lening) status</b>	- Lening indienen - Lening opgehaald administreren - Lening retourneren administreren	
<b>Lening nummer</b>	- Lening indienen	- Lening opgehaald administreren

		- Lening retourneren administreren
<b>Blok</b>		- Lening indienen
<b>Uitgeleende producten</b>	- Lening indienen	- Lening opgehaald administreren - Lening retourneren administreren
<b>Product defect</b>	- Lening retourneren administreren	

Uit bovenstaande tabel kan worden afgeleid dat het systeem wel voorraad, studentnummer, voornaam en achternaam toont, maar er geen use cases zijn waarin deze attributen aan het systeem worden toegevoegd. We hebben dus 'magic' attributen gevonden.

Als oplossing hiervoor zullen we handmatig de studentnummer, voor- en achternaam in het systeem zetten. De voorraad zal een optelling van de nog beschikbare producten zijn.

Ook blijkt dat de productstatus, klascode, lening status en product defect wel een waarde toegewezen krijgen, maar dat deze waarde later niet opgehaald word. Dit zijn dus 'black hole' attributen. Omdat we hier maar vier use cases hebben uitgewerkt en we denken dat deze attributen wel nodig zijn voor de andere use cases zullen wij deze attributen gewoon blijven vullen.

## 4.2 Functionaliteiten check

Uit ons Use Case diagram kunnen we ontnemen dat er twee actoren zijn. Dit vereist dat het te realiseren systeem weet door welke actor het bediend wordt.

Een oplossing voor deze issue is het realiseren van een dedicated systeem per onderkende actor, de andere oplossing voor deze issue is het uitbreiden van het systeem met een use case 'inloggen'.

We kiezen hier voor het toevoegen van de use case 'inloggen'.

Een use case die deel uitmaakt van de superuser 'anonieme gebruiker', een user van wie alle users erven en die we expliciet niet in het Use Case Diagram hebben opgenomen. Het toevoegen van de use case inloggen heeft echter ten gevolge dat we niet weten welke rol een gebruiker heeft die ingelogd is. Deze issue lossen we voor nu op door de in het systeem bekende users en hun rol hardcoded in het systeem te implementeren.

## 4.3 Conclusie

De use case inloggen wordt aan het systeem toegevoegd en de users (met studentnummer, voor- en achternaam) en rollen worden voor nu hardcoded geïmplementeerd.

## 5. Actoren

In het use case diagram uit hoofdstuk 3 is te zien dat we drie actoren hebben: student, beheerder en systeem. De templates voor de student en beheerder zijn hieronder te vinden.

Rolnaam:	Student
Rol Beschrijving	De student wordt de hoofdgebruiker van het systeem. Hij zal de producten kunnen bekijken en deze lenen. Ook kan hij al zijn leningen bekijken.

Rolnaam:	Beheerder	
Rol Beschrijving	De beheerder gaat over het verloop van een door een student ingediende lening. Hij kan de leningen bekijken en administreren wanneer een lening wordt opgehaald en teruggebracht. Ook kan hij producten toevoegen of veranderen in het systeem.	
Personen:	Naam:	Tim le Febre
	Foto:	
	Email:	Tim.lefebvre@hu.nl
	Functie:	Docent

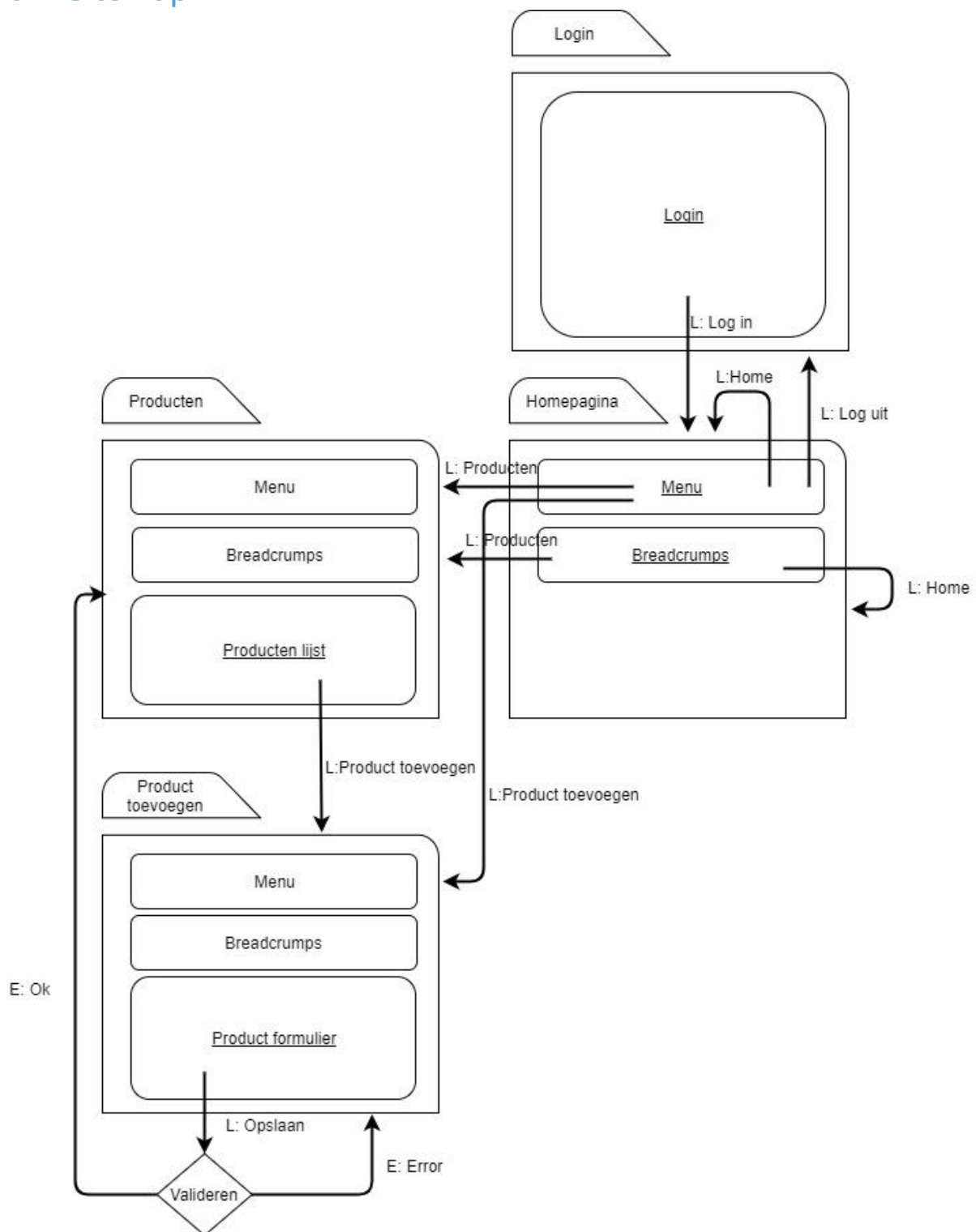
## 6. Use Case 1: Product toevoegen

In dit hoofdstuk wordt de use case om een product toe te voegen beschreven door eerst de use case template en sitemap te geven en vervolgens deze template te visualiseren met behulp van wireframes.

### 6.1 Use Case Template

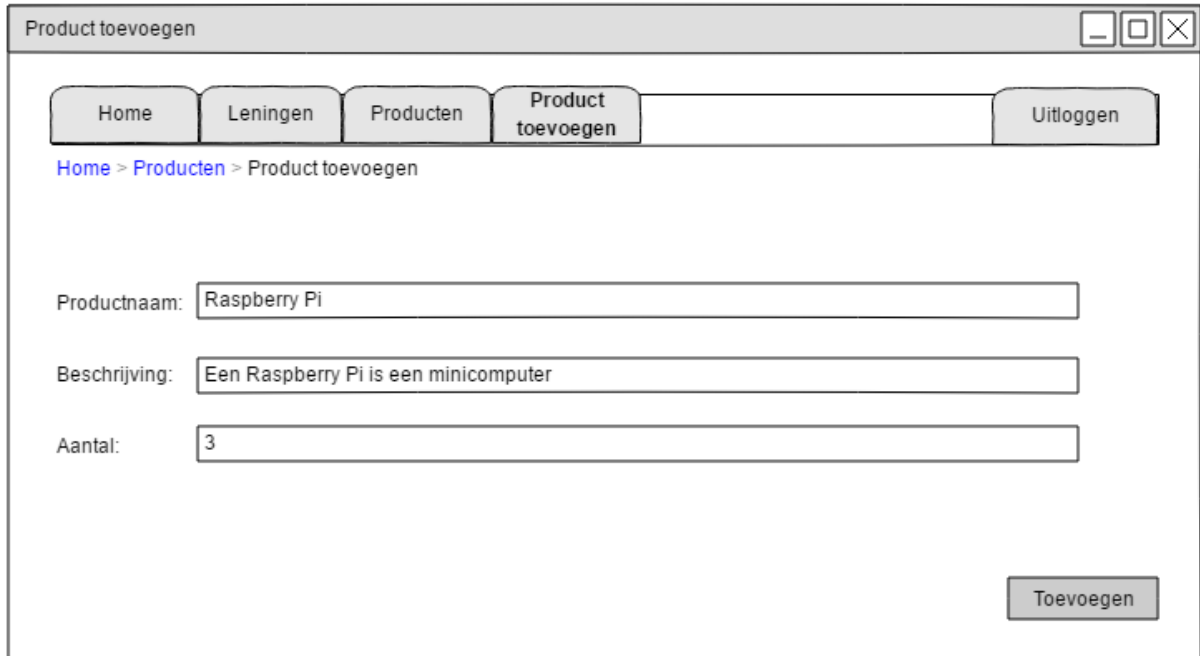
Gemaakt door	Rick Schoeman	
Use Case Nr	1	
Use Case Naam	Product toevoegen	
Actor	Beheerder	
Samenvatting	De actor kan een nieuw product toevoegen, hiervoor moet hij een <u>productnaam[in]</u> , <u>productbeschrijving[in]</u> en aantal aan te maken producten geven. Het systeem maakt net zoveel producten aan als het gegeven aantal met <u>status[in]</u> 'Beschikbaar' en kent aan elk product een uniek <u>productID[in]</u> toe.	
Pre Conditie	De actor is ingelogd en op de product toevoegen pagina	
Main Scenario	<ol style="list-style-type: none"> <li>1. Het systeem vraagt om een <u>productnaam[in]</u>, <u>productbeschrijving[in]</u> en het aantal producten om toe te voegen.</li> <li>2. A. De actor geeft de gevraagde informatie. B. De actor kiest om de producten toe te voegen</li> <li>3. A. Het systeem checkt of alle gevraagde informatie is gegeven B. Het systeem maakt net zoveel producten aan als het gegeven aantal met <u>status[in]</u> 'Beschikbaar' en kent aan elk product een uniek <u>productID[in]</u> toe.</li> </ol>	
Post conditie	Er zijn een of meerdere producten toegevoegd	
Alternatief Scenario 1	Trigger	Bij stap 3a – Niet alle benodigde informatie is gegeven
	Scenario	<ol style="list-style-type: none"> <li>1. Het systeem geeft een melding dat niet alle informatie is verstrekt.</li> <li>2. Systeem keert terug naar stap 1 van het main scenario.</li> </ol>
Alternatief Scenario 2	Trigger	Bij stap 2: Actor annuleert de Use Case
	Scenario	1. Systeem breekt de use case af
	Post conditie	Het systeem is ongewijzigd ten opzichte van voorafgaande aan de aanroep.

## 6.2 Sitemap



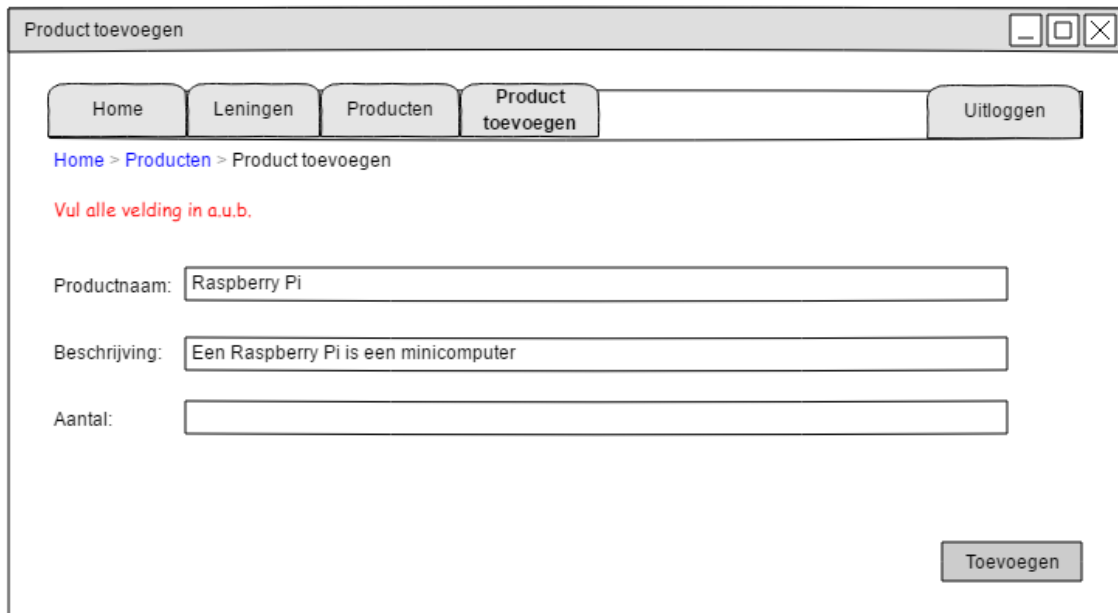
## 6.3 Wireframes

Hier zijn de drie componenten uit de sitemap te zien. Het wireframe laat stap 1 van het main scenario van het use case template zien. Hier kan de actor stap 2a uitvoeren door de gevraagde informatie in te vullen en 2b door op de knop 'toevoegen' te klikken. Hierna volgt stap 3 en maakt het systeem de producten aan.



The wireframe shows a web application window titled "Product toevoegen". At the top is a navigation bar with buttons: "Home", "Leningen", "Producten", "Product toevoegen" (highlighted), and "Uitloggen". Below the navigation bar is a breadcrumb trail: "Home > Producten > Product toevoegen". The main form contains three input fields: "Productnaam:" with the value "Raspberry Pi", "Beschrijving:" with the value "Een Raspberry Pi is een minicomputer", and "Aantal:" with the value "3". A "Toevoegen" button is located at the bottom right of the form.

Wanneer echter niet alle velden zijn ingevuld en er wordt op toevoegen geklikt zal het alternatief scenario 1 getriggerd worden en wordt er een error bericht getoond.



The wireframe shows the same "Product toevoegen" form, but with an error message displayed in red text: "Vul alle velding in a.u.b.". The "Aantal:" field is now empty, indicating it was not filled. The "Productnaam:" and "Beschrijving:" fields still contain their previous values. The "Toevoegen" button remains at the bottom right.

## 7. Use Case 2: Lening indienen

In dit hoofdstuk wordt de use case lening indienen beschreven door eerst de use case template en sitemap te geven en vervolgens deze template te visualiseren met behulp van wireframes.

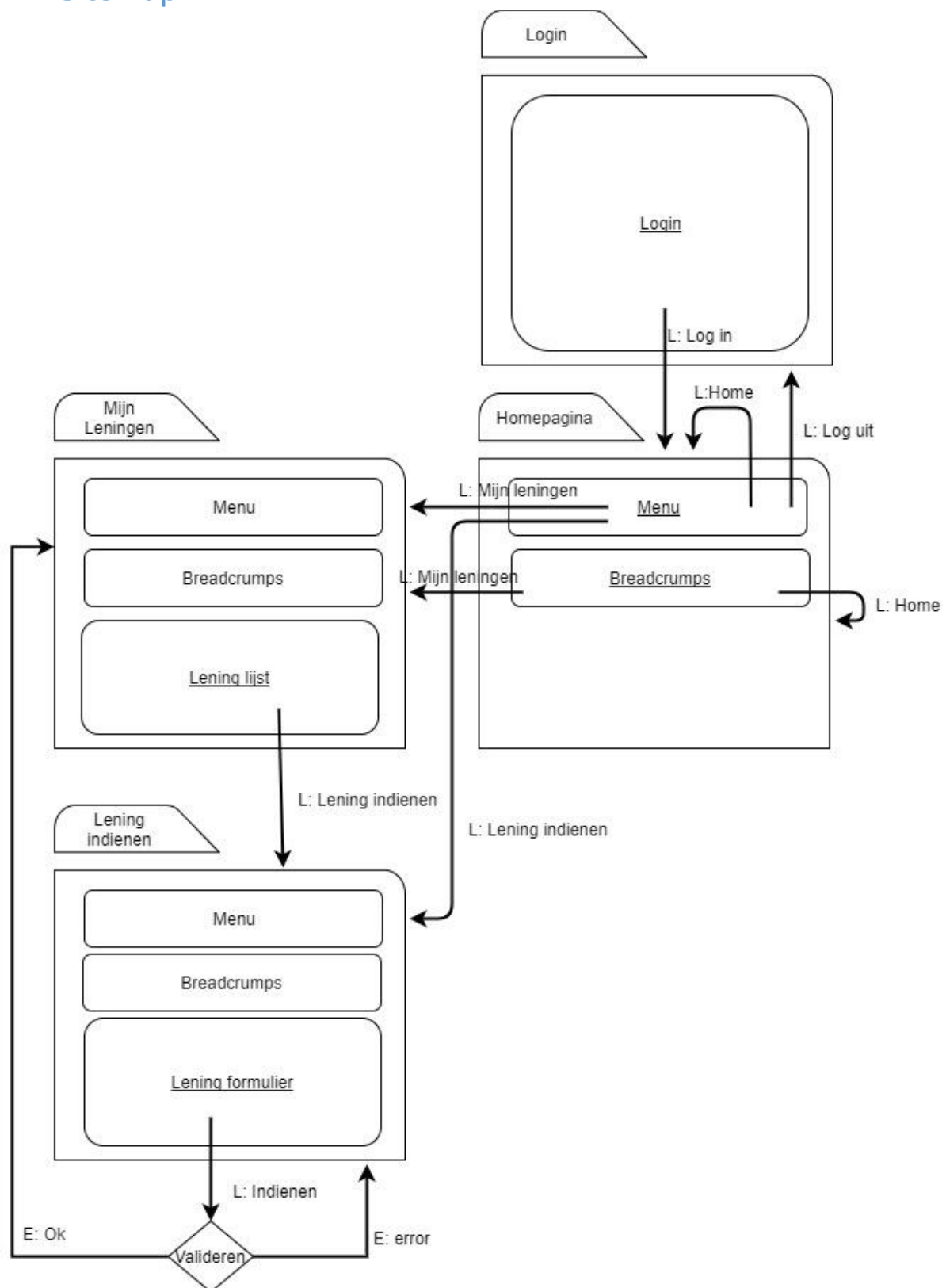
### 7.1 Use Case Template

Gemaakt door	Lucas Bos
Use Case Nr	2
Use Case Naam	Lening indienen
Actor	Student
Samenvatting	<p>De actor geeft aan d.m.v. <u>klascode</u> [in] voor welk vak hij producten wilt lenen. Vervolgens geeft hij aan uit een lijst met <u>productnaam</u>[out] met bijhorende <u>productbeschrijving</u>[out] en de <u>voorraad</u>[out] welke producten en het <u>aantal</u>[in] per product hij wilt lenen.</p> <p>De actor kiest om de gekozen producten te lenen.</p> <p>Het systeem haalt a.d.h.v. het <u>studentnummer</u> [out] van de ingelogde actor de <u>voornaam</u>[out] en <u>achternaam</u>[out] op.</p> <p>Het systeem maakt de lening aan met <u>status</u>[in]'ingediend'. De lening bevat een <u>lening nummer</u>[in], het <u>studentnummer</u> [out] van de actor, de gekozen <u>klascode</u>[in], het huidige <u>blok</u> [out] en de <u>uitgeleende producten</u> [in], per product het <u>productid</u>[out] en <u>aantal</u>[in].</p> <p>Per <u>geleend product</u>[in] word de <u>status</u>[in] veranderd in 'Aangevraagd'.</p>
Pre Conditie	De actor is ingelogd en op de lening indienen pagina.
Main Scenario	<ol style="list-style-type: none"> <li>1. Het systeem vraagt om de klascode waar de actor producten voor wilt lenen.</li> <li>2. De actor geeft klascode</li> <li>3. Het systeem toont een lijst met de producten die nog op voorraad zijn. Per product wordt de productnaam [out], productbeschrijving [out] en de voorraad [out] getoond en een invoerveld zodat de actor het aantal[in] producten kan aangeven dat hij wilt lenen.</li> <li>4. a) De actor geeft aan welke producten en hoeveel hij per product wilt lenen. b) De actor kiest om de lening te plaatsen</li> <li>5. a) Het systeem checkt of dat er een klascode is gegeven b) Het systeem checkt dat er per product niet meer worden uitgeleend dan op voorraad is c) Het systeem haalt a.d.h.v. het studentnummer[out] van de ingelogde actor de voornaam[out] en achternaam[out] op. d) Het systeem voegt de lening aan het system toe. e) Het systeem veranderd de status[in] van elk uitgeleend product[in] naar 'aangevraagd'</li> </ol>
Post conditie	Er is een nieuwe lening aan het systeem toegevoegd.



Alternatief Scenario 1	Trigger	Bij stap 5a: Er is geen klascode gegeven
	Scenario	<ol style="list-style-type: none"> <li>1. Systeem geeft een melding dat er klascode moet worden gegeven</li> <li>2. Systeem keert terug naar stap 1 van het main scenario</li> </ol>
Alternatief Scenario 2	Trigger	Bij stap 2 & 4: Actor annuleert de Use Case
	Scenario	1. Systeem breekt de use case af
	Post conditie	Het systeem is ongewijzigd ten opzichte van voorafgaande aan de aanroep.
Alternatief Scenario 3	Trigger	Bij stap 5b: De actor heeft aangegeven dat hij meer producten wilt lenen dan op voorraad is
	Scenario	<ol style="list-style-type: none"> <li>1. Systeem geeft een melding dat er niet genoeg van het gekozen product op voorraad is.</li> <li>2. Systeem keert terug naar stap 3 van het main scenario</li> </ol>

## 7.2 Sitemap



## 7.3 Wireframes

In deze wireframe zijn de drie componenten uit de sitemap te zien. Het systeem vraagt om een klascode bij stap 1 van het main scenario van de use case template. Bij stap 2 geeft de student zijn klascode op door dit in te vullen. Hierna volgt stap 3 waarbij het systeem een lijst met producten laat zien die nog op voorraad zijn. Vervolgens geeft de student aan welke producten en hoeveel hij per product wilt lenen door dit in te vullen in de tabel onder het kopje aantal. Hij kan ervoor kiezen om de lening te plaatsen door op indienen te klikken (stap 4). Zodra de student ervoor heeft gekozen om de lening te plaatsen volgt stap 5 waarbij het systeem een aantal checks uitvoert, het systeem de lening toevoegd en de status van elk uitgeleend product wordt veranderd.

Lening indienen

Home
Mijn leningen
Lening indienen
Producten
Uitloggen

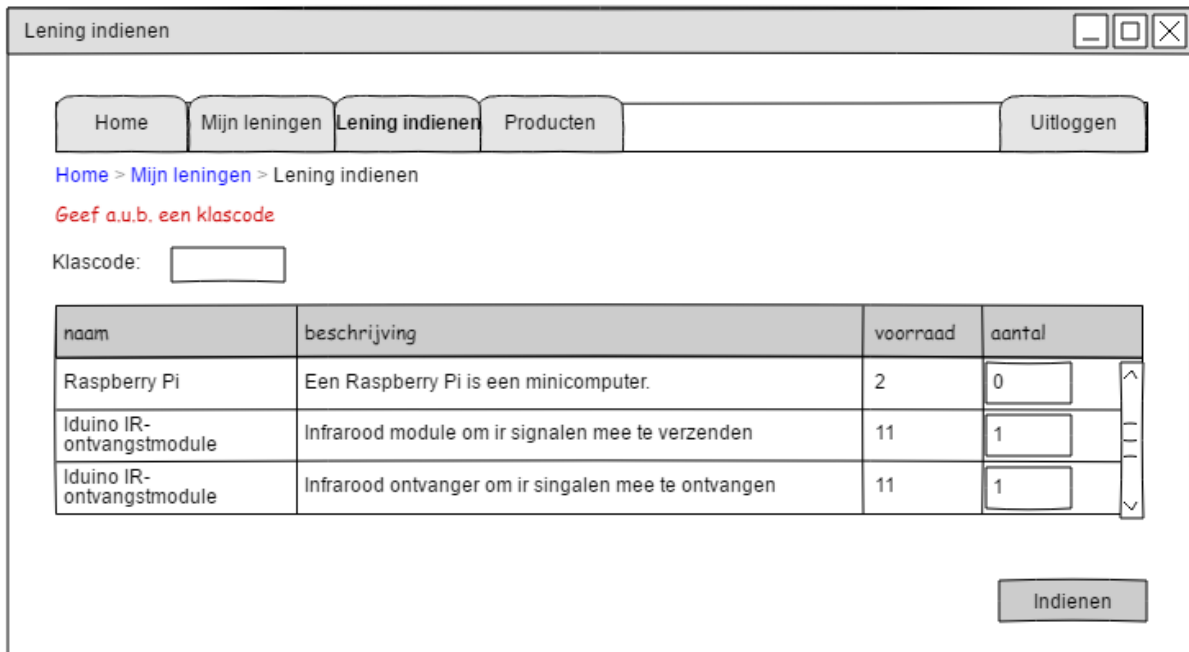
Home > Mijn leningen > Lening indienen

Klascode:

naam	beschrijving	voorraad	aantal
Raspberry Pi	Een Raspberry Pi is een minicomputer.	2	<input type="text" value="0"/>
Iduino IR-ontvangstmodule	Infrarood module om ir signalen mee te verzenden	11	<input type="text" value="1"/>
Iduino IR-ontvangstmodule	Infrarood ontvanger om ir singalen mee te ontvangen	11	<input type="text" value="1"/>

Indienen

Wanneer er geen klascode is gegeven en er op 'Indienen' wordt gedrukt zal alternatief scenario 1 worden getriggerd en wordt er een fout melding getoond:



Lening indienen

Home Mijn leningen **Lening indienen** Producten Uitloggen

Home > Mijn leningen > Lening indienen

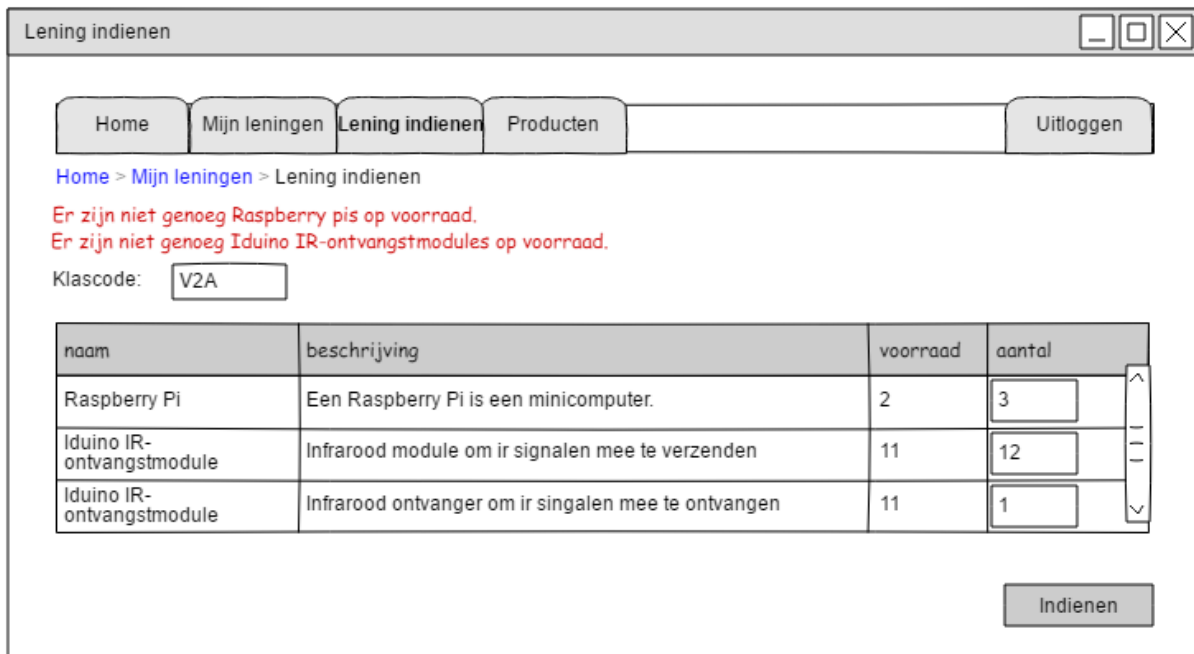
Geef a.u.b. een klascode

Klascode:

naam	beschrijving	voorraad	aantal
Raspberry Pi	Een Raspberry Pi is een minicomputer.	2	<input type="text" value="0"/>
Iduino IR-ontvangstmodule	Infrarood module om ir signalen mee te verzenden	11	<input type="text" value="1"/>
Iduino IR-ontvangstmodule	Infrarood ontvanger om ir singalen mee te ontvangen	11	<input type="text" value="1"/>

Indienen

Als er echter wel een klascode is gegeven maar het gegeven aantal producten om te lenen is hoger dan de voorraad en er word op 'Indienen' geklikt zal alternatief scenario 3 worden getriggerd en word er een foutmelding getoond.



Lening indienen

Home Mijn leningen **Lening indienen** Producten Uitloggen

Home > Mijn leningen > Lening indienen

Er zijn niet genoeg Raspberry pis op voorraad.  
Er zijn niet genoeg Iduino IR-ontvangstmodules op voorraad.

Klascode:

naam	beschrijving	voorraad	aantal
Raspberry Pi	Een Raspberry Pi is een minicomputer.	2	<input type="text" value="3"/>
Iduino IR-ontvangstmodule	Infrarood module om ir signalen mee te verzenden	11	<input type="text" value="12"/>
Iduino IR-ontvangstmodule	Infrarood ontvanger om ir singalen mee te ontvangen	11	<input type="text" value="1"/>

Indienen

Wanneer er geen klascode en teveel producten worden geleend zullen allebei de fout berichten worden getoond.

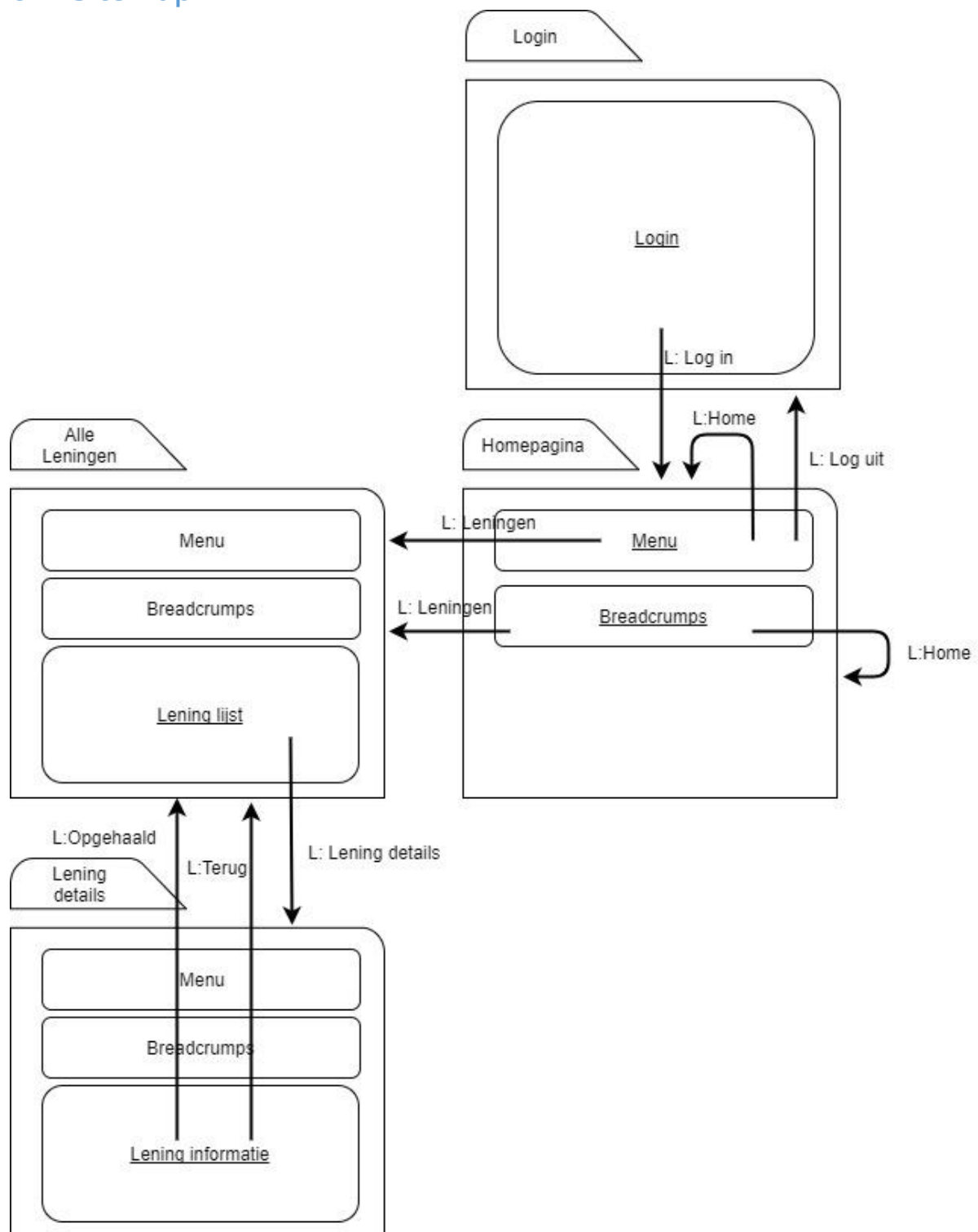
## 8. Use Case 3: Lening opgehaald administreren

In dit hoofdstuk wordt de use case lening opgehaald administreren beschreven door eerst de use case template en sitemap te geven en vervolgens deze template te visualiseren met behulp van wireframes.

### 8.1 Use Case Template

Gemaakt door	Roy Hendriks	
Use Case Nr	3	
Use Case Naam	Lening opgehaald administreren	
Actor	Beheerder	
Samenvatting	<p>Het systeem toont het <u>lening nummer[out]</u>, <u>studentnummer[out]</u>, de <u>voornaam[out]</u>, <u>achternaam[out]</u> en de lijst met uitgeleende <u>producten[out]</u> van de gekozen lening. Per product wordt de <u>productid[out]</u>, <u>productnaam[out]</u> en de <u>productomschrijving[out]</u> getoond.</p> <p>De actor geeft aan dat de lening is opgehaald.</p> <p>Het systeem verandert de <u>status[in]</u> van de lening in 'producten uitgeleend' en de <u>status[in]</u> van de producten naar 'uitgeleend'</p>	
Pre Conditie	De actor is ingelogd, op de lening opgehaald administreren pagina en heeft een lening geselecteerd.	
Main Scenario	<ol style="list-style-type: none"> <li>1. Het systeem toont het lening nummer[out], studentnummer[out], de voornaam[out], achternaam[out] en de lijst met uitgeleende producten[out] van de gekozen lening. Per product wordt de productid[out], productnaam[out] en de productomschrijving[out] getoond.</li> <li>2. De actor geeft aan dat de lening is opgehaald.</li> <li>3. Het systeem verandert de status[in] van de lening in 'producten uitgeleend' en de status[in] van de producten naar 'uitgeleend'</li> </ol>	
Post conditie	De status van de lening is verander naar 'producten uitgeleend' en de status van de producten is veranderd naar 'uitgeleend'	
Alternatief Scenario 1	Trigger	Bij stap 2: Actor annuleert de Use Case
	Scenario	1. Systeem breekt de use case af
	Post conditie	Het systeem is ongewijzigd ten opzichte van voorafgaande aan de aanroep.

## 8.2 Sitemap



## 8.3 Wireframe

Hier zijn de drie componenten uit de sitemap te zien. Het wireframe toont stap 1 van het main scenario uit het use case template. Hierin kan de beheerder stap 2 uitvoeren door aan te geven dat een lening is opgehaald door op de 'opgehaald' knop te klikken. Hierna volgt stap 3 en veranderen de statussen van de lening en producten.

Lening informatie

Home

Leningen

Producten

Product toevoegen

Uitloggen

Home > Leningen > Lening informatie

Lening nummer: 1  
Studentnummer: 1967045  
Naam: Rob Kramer

Id	Naam	Omschrijving
1	Raspberry Pi	Een Raspberry Pi is een minicomputer.
5	Iduino IR-ontvangstmodule	Infrarood module om ir signalen mee te verzenden.
7	Led lampje	Een led lampje geeft licht.

Terug

Opgehaald

## 9. Use Case 4: Lening retourneren administreren

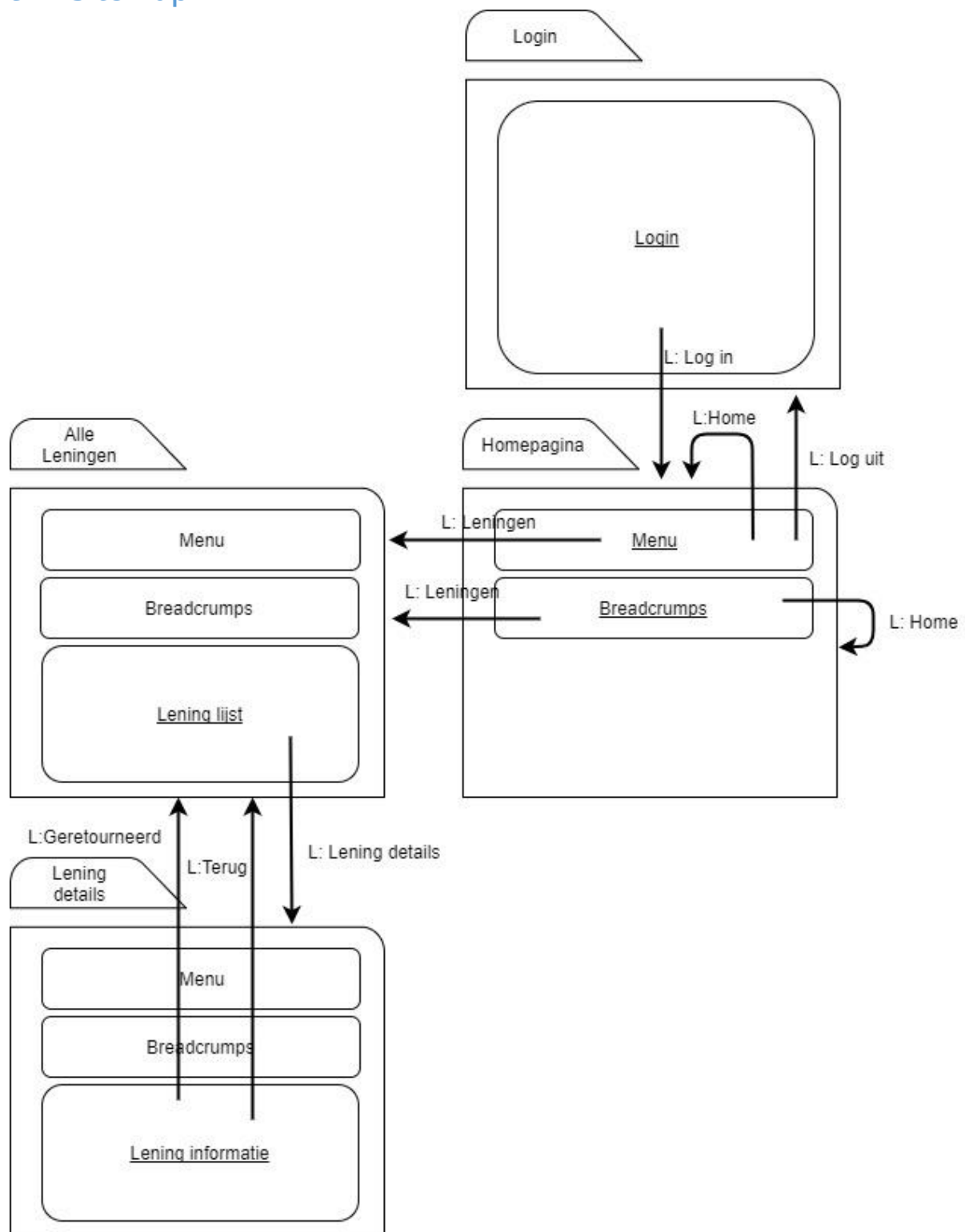
In dit hoofdstuk wordt de use case om een lening retourneren te administreren beschreven door eerst de use case template en sitemap te geven en vervolgens deze template te visualiseren met behulp van wireframes.

### 9.1 Use Case Template

Gemaakt door	Nick Schouten	
Use Case Nr	4	
Use Case Naam	Lening retourneren administreren	
Actor	Beheerder	
Samenvatting	Het systeem toont het <u>lening nummer[out]</u> , <u>studentnummer[out]</u> , de <u>voornaam[out]</u> , <u>achternaam[out]</u> en de lijst met <u>uitgeleende producten[out]</u> van de gekozen lening. Per product wordt de <u>productid[out]</u> , <u>productnaam[out]</u> en de <u>productomschrijving[out]</u> getoond en er is een mogelijkheid om aan te geven of dit product <u>defect is[in]</u> of niet. De actor geeft aan per product of deze defect is of niet. Vervolgens kiest de actor om de wijzigingen op te slaan. Het systeem verandert de <u>status[in]</u> van de lening in 'Afgerond'. Per teruggebracht product word de <u>status[in]</u> veranderd in 'Beschikbaar', maar wanneer is aangegeven dat het product defect is word de <u>status[in]</u> 'Defect'.	
Pre Conditie	De actor is ingelogd, op de lening retourneren pagina en heeft een lening geselecteerd	
Main Scenario	<ol style="list-style-type: none"> <li>1. Het systeem toont het lening nummer[out], studentnummer[out], de voornaam[out], achternaam[out] en de lijst met uitgeleende producten[out] van de gekozen lening. Per product wordt de productid[out], productnaam[out] en de productomschrijving[out] getoond en er is een mogelijkheid om aan te geven of dit product defect is[in] of niet.</li> <li>2. A. De actor geeft aan per product of deze defect is of niet. B. De actor kiest om de wijzigingen op te slaan</li> <li>3. A. Het systeem verandert de status[in] van de lening in 'Afgerond'. B. Per teruggebracht product word de status[in] veranderd in 'Beschikbaar', maar wanneer is aangegeven dat het product defect is word de status[in] 'Defect'.</li> </ol>	
Post conditie	De status van de lening is veranderd naar 'Afgerond' en de status van de uitgeleende producten is veranderd naar 'Beschikbaar' of 'Defect'	
Alternatief Scenario 1	Trigger	Bij stap 2: Actor annuleert de Use Case
	Scenario	1. Systeem breekt de use case af
	Post conditie	Het systeem is ongewijzigd ten opzichte van voorafgaande aan de aanroep.



## 9.2 Sitemap



## 9.3 Wireframe

Hier zijn de drie componenten uit de sitemap te zien. Het wireframe toont stap 1 van het main scenario uit de use case template. Hierin kan de beheerder stap 2a uitvoeren door aan te geven of een product defect is door dit aan te vinken in de tabel. Hij kan vervolgens stap 2b uitvoeren door op de knop 'Geretourneerd' te klikken. Hierna volgt stap 3 en veranderen de statussen van de lening en producten.

Lening informatie

Home

Leningen

Producten

Product toevoegen

Uitloggen

Home > Leningen > Lening informatie

Lening nummer: 1  
Studentnummer: 1967045  
Naam: Rob Kramer

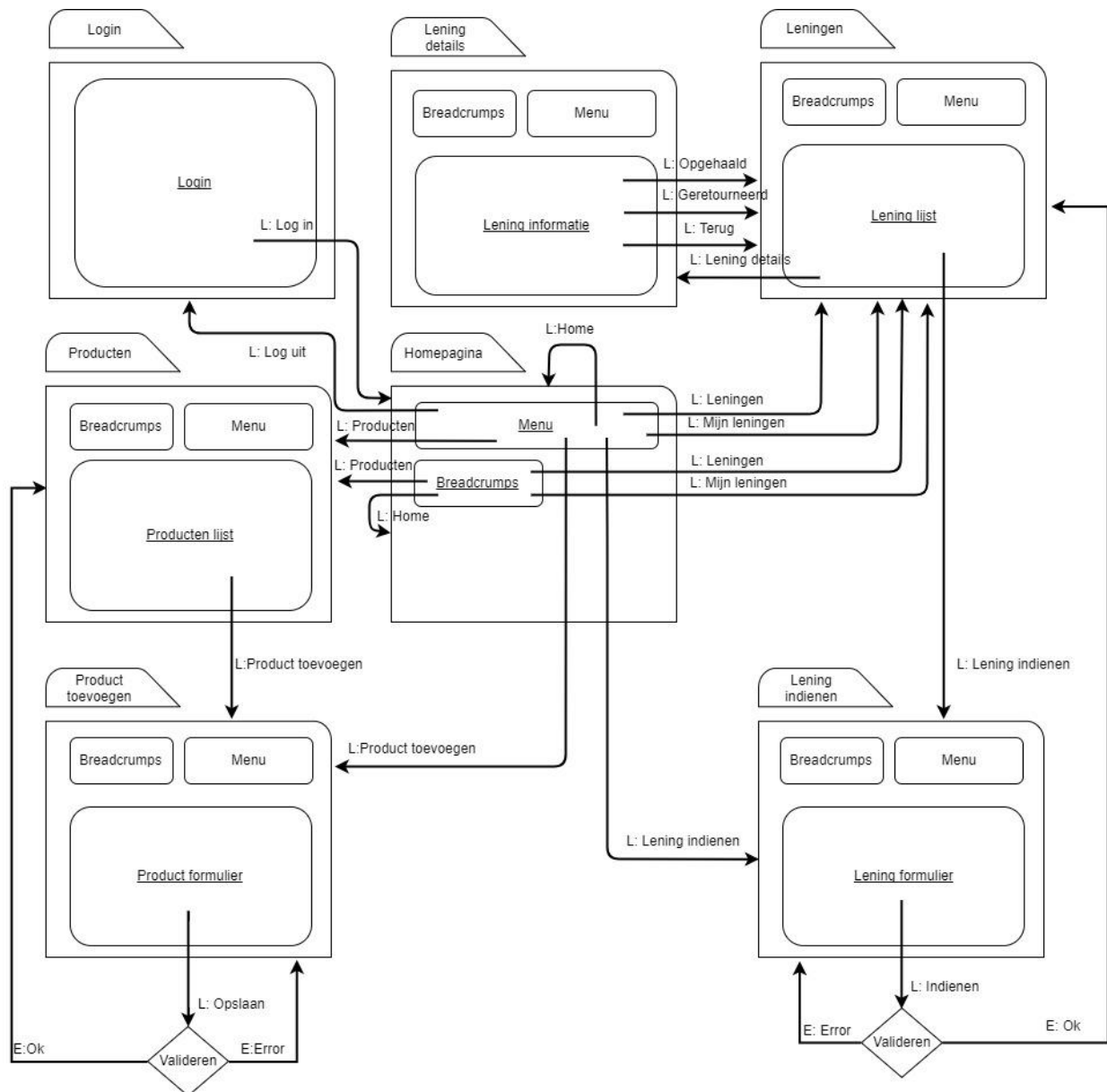
Id	Naam	Omschrijving	Defect	
1	Raspberry Pi	Een Raspberry Pi is een minicomputer.	<input type="checkbox"/>	^
5	Iduino IR-ontvangstmodule	Infrarood module om ir signalen mee te verzenden.	<input checked="" type="checkbox"/>	—
7	Led lampje	Een led lampje geeft licht.	<input type="checkbox"/>	v

Terug

Geretourneerd

## 10. Sitemap

Op basis van de Use Cases welke in de vorige hoofdstukken zijn behandeld, kan de volgende sitemap worden opgemaakt.



## 11. Technische requirements

De applicatie dient in eerste instantie te worden gerealiseerd als web app doormiddel van angular 2.

Voor de login use case, als mede voor de data opslag, dient de app gebruik te maken van Firebase (<https://firebase.google.com>).