Functioneel Ontwerp



Medzo Seck,1661785

Rick de Jong, 1664503

Nawid Norouzi, 1641732

Jotte Smit, 1630766

[**Inleiding**](#_z1stzf7i2rxu) **4**

[**Life-Cycle**](#_ot3vz7su9tep) **5**

[**Use-cases**](#_qd0605m554ho) **6**

[**Volledigheid Analyse**](#_ayewwcfjqc91) **8**

[Attributen check](#_5qxxc4lp891i) 8

[Functionaliteiten check](#_ftzshj5nfyqx) 9

[Conclusie](#_wl4y41xw3o2a) 10

[**Actoren**](#_vmiaopx3l573) **11**

[**Sitemap**](#_gyiojx3x9ng) **12**

[**Use Case 1: Lening maken**](#_hpuh90x40qv5) **13**

[Lening maken](#_z3zqs68kmklj) 13

[Wireframe](#_n6wj4z2ouxrj) 14

[**Use Case 2: Inventaris bekijken**](#_rxxqlyohomnc) **15**

[Inventaris bekijken](#_jn24j6xxv8bh) 15

[Wireframe](#_pdd672zgpxya) 16

[**Use Case 3: Inventaris beheren**](#_4in2mpg0ft3l) **17**

[Inventaris beheren](#_wos7praalqxr) 17

[Wireframe](#_bj5q2r8k8dee) 18

[**Use Case 4: Product aanmaken**](#_sjzuccdjlt7k) **20**

[Product aanmaken](#_4d0etrywclkx) 20

[Wireframe](#_m5ab7uv1qj10) 21

[**Use Case 5: Product verwijderen**](#_oxidgdm5tj36) **22**

[Product verwijderen](#_59t8o4h15lo8) 22

[Wireframe](#_xjjuehpy6xnl) 23

[**Use Case 6: Eigen lening bekijken**](#_uniutvet5lkg) **24**

[Eigen lening bekijken](#_gijh4efar9ha) 24

[Wireframe](#_8kamkd25oul9) 24

[**Use Case 7: Verlopen lening**](#_vvsqc6phlkwh) **26**

[Verlopen lening](#_h4wlb22cae7j) 26

[Wireframe](#_d0zteqo1ei71) 26

[**Use Case 8: Lening bekijken**](#_7rrw8xu2zvid) **27**

[Lening bekijken](#_7jn8a6vjmaf0) 27

[Wireframe](#_6ry311estdbk) 28

[**Use Case 9: Lening afronden**](#_lxe1bbos22k3) **29**

[Leningen afronden](#_vktokyexg7x5) 29

[Wireframe](#_xqovt6fi9run) 30

[**Technische Requirements**](#_zfeyy44lrvu3) **32**

# Inleiding

In dit functioneel ontwerp wordt toegelicht hoe het leensysteem voor Tim wordt gebouwd. Allereerst zal er een stuk worden uitgelegd over de Life-Cycle van een leenproduct. Vervolgens zullen het use case diagram voor deze applicatie worden uitgelegd. Onder het diagram zal per genoemde use case een samenvatting staan die aan het einde van het functioneel ontwerp verder worden toegelicht.

Er wordt hierna geschreven over de volledigheids analyse met een attributen check, een functionaliteit check en een conclusie. Daarna zullen de verschillende actoren die dit systeem gaan gebruiken worden toegelicht en zal een sitemap getoond worden.

Het document sluit af met uitgebreid use case samenvattingen,wireframes en de technische requirements analyse.

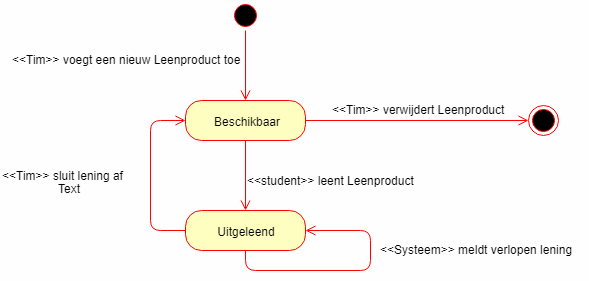
# Life-Cycle

Het is de taak van Tim om de leenproducten te beheren, hierdoor hebben de studenten de mogelijkheid deze producten te gebruiken.

Aan het eind van ons project zullen wij een systeem opgezet hebben die tim moet assisteren met het managen van de leenproducten.

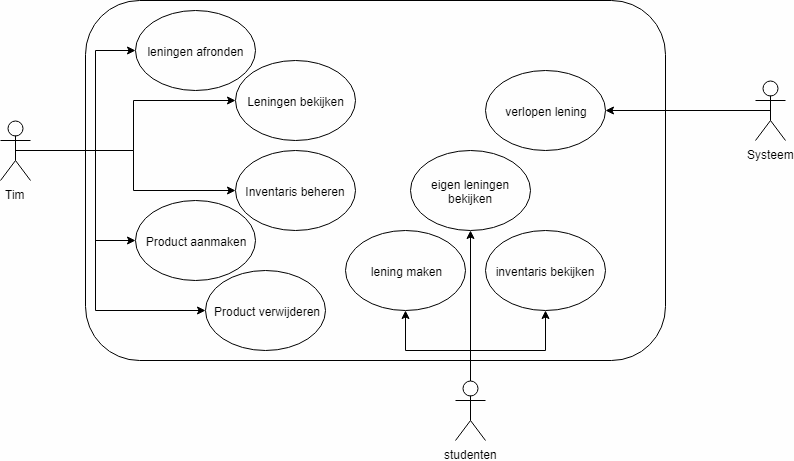
Dit systeem zal door tim en de studenten gebruikt worden om leenproducten makkelijker te lenen en beheren.

Als we dit vertalen naar een toestandsdiagram voor het kern-object van deze casus dan komen we tot het volgende model:



# Use-cases

Tim moet producten toevoegen ("Product aanmaken", create) en verwijderen ("Product verwijderen", delete). Een student moet de producten inzien ("Inventaris bekijken", read), producten lenen ("Lening maken", create&update), en deze leningen bekijken ("Eigen leningen bekijken", read). Tim moet leningen bekijken ("Leningen bekijken", read), afronden ("Leningen afronden", update), en producten bijwerken ("Inventaris beheren", update). Het systeem moet controleren of leningen verlopen, en de betreffende studenten erover melden ("Verlopen lening", create&update).



Een beknopte samenvatting voor elke use case:

|  |  |
| --- | --- |
| **Use case:** | **Samenvatting:** |
| Lening maken | Het Systeem toont de lening template met daarin de startdatum[in], de einddatum[in] en de leenproducten[o] die gekozen zijn. De Actor vult de gegevens in.Het Systeem checkt vervolgens de ingevulde gegevens. De Actor bevestigt hierna de lening. Het Systeem slaat de lening op en zet status uitgeleend op true van geleende leenproducten. |
| Inventaris bekijken | Het Systeem toont een scherm met daarin een lijst van alle aanwezig leenproducten. De Actor kiest een Leenproducten[in] die hij wilt lenen. Het Systeem toont de details van het Leenproduct[o]. De Actor bevestigt zijn keuze. Het Systeem voegt het leenproduct toe in lening maken pagina. Het Systeem toont het overzicht van alle leenproducten weer |
| Inventaris beheren | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o). De actor klikt(i) in de tabel op een Productnaam. Het systeem toont een formulier met de vooringevulde input fields Productnaam(o), Aantal(o), Beschrijving. Actor past in de input fields aan(i) wat de actor aan wilt passen en klikt(i) op de button ‘Aanpassingen opslaan’ als de actor klaar is. |
| Product aanmaken | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o) en een button ‘Maak product aan’(o). Actor klikt(i) op button ‘Maak product aan’. Het systeem toont een formulier met een button ‘Product aanmaken’ en de volgende input fields: Productnaam, Aantal en Beschrijving. Actor vult Productnaam(i), Aantal(i) en Beschrijving(i) in en slaat door op de button ‘Product aanmaken’ te klikken het nieuwe product op. Het systeem geeft het nieuwe product een ProductID(i). |
| Product verwijderen | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o). De actor klikt(i) in de tabel op een Productnaam. Het systeem toont een formulier met de voor ingevulde input fields Productnaam(o), Aantal(o) en een button ‘Product verwijderen’(o). Actor klikt(i) op de button ‘Product verwijderen’. Het Systeem toont een pop-up(o) met de tekst ‘Weet u zeker dat u dit product wilt verwijderen?’ en de keuzes ‘JA’ en ‘NEE’. Actor klikt(i) op ‘JA’. Systeem verwijdert product. |
| Eigen lening bekijken | De student wilt zijn openstaande leningen bekijken. De student navigeert naar de pagina waarop zijn leningen te zien zijn. De pagina geeft van de openstaande leningen de informatie weer. De informatie van de leningen zijn tegelijkertijd te zien op een enkele pagina.  De pagina geeft per lening  -De namen van de geleende object  -De hoeveelheid van geleende objecten  -De inleverdatum van de geleende objecten |
| Verlopen lening | Het systeem stuurt een email bericht dat een lening is verlopen. |
| Lening bekijken | Actor wil de leningen inzien. Actor vraagt de leningen op (lening.afgerond [in], lening.teLaatIngeleverd [in]), en Systeem weergeeft ze (student.studentnummer [out], lening.einddatum [out], leenproduct.naam [out]). |
| Leningen afronden | Actor wil leningen afronden. Actor vraagt de leningen op (lening.afgerond [in], lening.teLaatIngeleverd [in]), en Systeem weergeeft ze (student.studentnummer [out], lening.einddatum [out]). Actor kiest een lening , Systeem weergeeft de lening details (student.voornaam [out], student.achternaam [out], lening.startdatum [out], leenproduct.naam [out], leenproduct.beschrijving [out]), Actor kiest producten (leenproduct.aantal [out], leenproduct.functioneel [out]), Actor wijzigt deze (leenproduct.aantal [in], leenproduct.functioneel [in]), Actor bevestigt, en Systeem wijzigt de leningen naar afgerond (lening.afgerond [in]). |

# Volledigheid Analyse

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

## Attributen check

Door per attribuut dat we uit de samenvattingen van de use cases kunnen extraheren te onderzoeken in welke use case er een waarde aan het attribuut wordt toegekend, dan wel 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(n) van het attribuut toont. |
| lening.startdatum | lening maken | lening bekijken |
| lening.einddatum | lening maken | lening bekijken |
| lening.leenproducten | lening maken | lening bekijken |
| lening.afgerond | lening bekijken,  lening afronden | eigen lening bekijken |
| lening.teLaatIngeleverd | lening bekijken,  lening afronden | eigen lening bekijken,  lening afronden |
| leenproduct.productID | product aanmaken | product aanmaken,  inventaris beheren,  product verwijderen |
| leenproduct.productnaam | product aanmaken  inventaris beheren | inventaris beheren,  product aanmaken,  product verwijderen |
| leenproduct.beschrijving | product aanmaken,  inventaris beheren | inventaris beheren,  product aanmaken,  product verwijderen |
| leenproduct.aantal | product aanmaken,  inventaris beheren,  lening maken | inventaris beheren,  product aanmaken,  product verwijderen |
| leenproduct.functioneel | product aanmaken,  inventaris beheren,  lening maken | inventaris beheren,  product aanmaken,  product verwijderen |
| student.studentnummer | - | lening afronden |

De scope van dit project bevat niet het maken van een registratie voor student, dus zullen de gebruikers hardcoded in de database geschreven worden. Dit zijn de gebruikers waarmee we zullen testen.

## Functionaliteiten check

Uit ons Use Case diagram kunnen we ontnemen dat er drie actoren zijn. Waarbij verschillende actoren verschillende functionaliteiten zal hebben. De actor tim heeft de admin rol en zal dus recht hebben om leningen en leenproducten te bewerken en verwijderen. De actor student heeft de lener rol en zal dus het recht hebben om leningen aan te maken. De actor systeem heeft geen rol als gebruiker binnen het product. Deze verschillende rollen vereisen dat het te realiseren systeem weet door welke actor het bediend wordt.

Een oplossing voor dit issue is het realiseren van een dedicated systeem per onderkende actor, de andere oplossing voor dit issue is het realiseren van een autorisatiesysteem voor de actoren. We hebben gekozen voor het realiseren van een autorisatiesysteem.

Het autorisatiesysteem zal functionaliteiten verbergen aan de hand van het autorisatieniveau van de gebruiker. Als de gebruiker niet geautoriseerd is om een pagina te bezoeken zal deze niet te zien zijn. Mocht het gebeuren dat de gebruiker toch op de pagina komt zal het systeem de gebruiker terug sturen naar de homepage. Dit autorisatieniveau zal bepaald worden door een check de id van de gebruiker. Aangezien we maar één gebruiker is admin rechten heeft zullen we zijn id hardcoded checken.

## Conclusie

Het autorisatiesysteem zal worden geïmplementeerd om functionaliteit tussen Tim en student gescheiden te houden. De gebruikers zullen hardcoded in de database worden gezet met een rol.

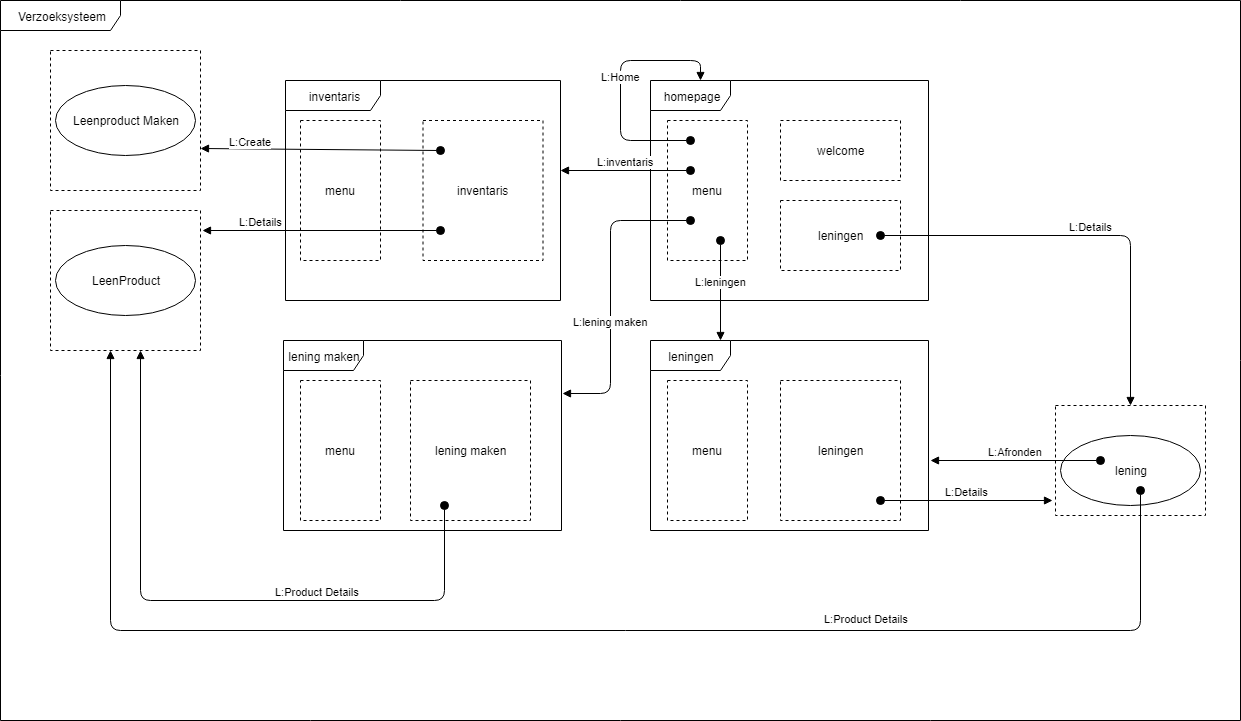
# Actoren

|  |  |
| --- | --- |
| Project | Verzoek Systeem |
| Rolnaam | Tim |
| Rolbeschrijving | De eigenaar is verantwoordelijk voor het beheren van het systeem. Dit omschrijft het toevoegen en weghalen van voorwerpen uit de inventaris en het afronden van leningen.  De eigenaar kan de huidige inventaris van het systeem en alle huidige leningen inzien.  De eigenaar is ook verantwoordelijk voor het beheer van het fysieke systeem wat aan de software gekoppeld wordt. |
| Personen | |  |  | | --- | --- | | Naam | Tim Le Febre | | Email | onbekend | | Functie | Docent | | Aantekeningen | - | |

|  |  |
| --- | --- |
| Project | Verzoek Systeem |
| Rolnaam | Student |
| Rolbeschrijving | De studenten zijn leerlingen die spullen via het systeem willen lenen. Studenten kunnen een lijst van beschikbare spullen bekijken en nieuwe leningen openen. Ook is het mogelijk om eigen leningen in te zien. |

# Sitemap

Uit de usecases kan de volgende sitemap worden gemaakt.



Het menu bevat links naar de pagina’s van leningen bekijken, lening maken en inventaris bekijken. Het menu is te zien op alle pagina’s.

Op de welkomstpagina is een welkomstbericht te zien. Hierin staan instructies over hoe het systeem gebruikt kan worden. Hier kunnen ook de eigen leningen bekeken worden.

Op de leningen pagina is door Tim te zien welke leningen de gebruikers hebben openstaan. Tim kan hier leningen afronden als spullen zijn teruggebracht.

Op de inventaris pagina is een lijst te zien van alle te lenen spullen. Studenten kunnen hier details bekijken van de spullen. Via de detailpagina van een object kan de student aangeven dat hij het object wilt lenen.

Tim kan via de inventaris pagina nieuwe objecten toevoegen. Dit opent een popup scherm waar de informatie ingevuld kan worden. Tim kan ook de eigenschappen van een object aanpassen of het object verwijderen.

Op de lening maken pagina kan de student zijn gekozen producten lenen Daarnaast kan hij de details van de gekozen leenproducten bekijken.

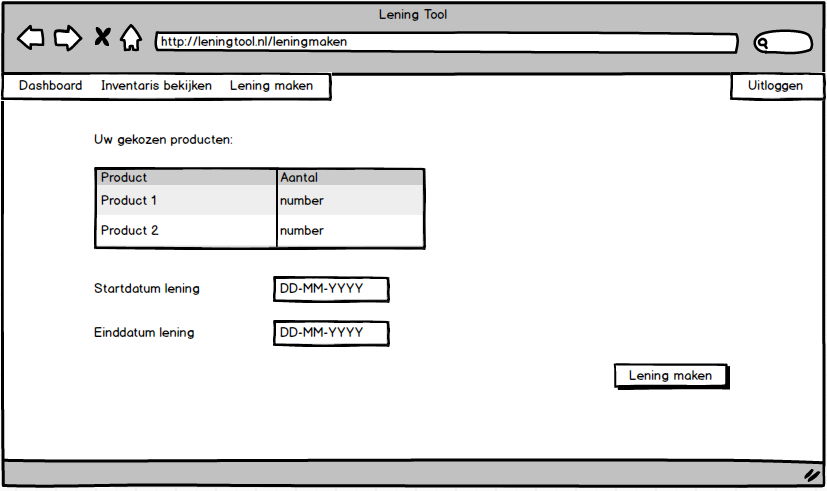
# Use Case 1: Lening maken

## Lening maken

|  |  |
| --- | --- |
| ID | 1 |
| Naam | Lening maken |
| Actor | Student |
| Samenvatting | Het Systeem toont de lening template met daarin de startdatum[in], de einddatum[in] en de leenproducten[o] die gekozen zijn. De Actor vult de gegevens in.Het Systeem checkt vervolgens de ingevulde gegevens. De Actor bevestigt hierna de lening. Het Systeem slaat de lening op en zet status uitgeleend op true van geleende leenproducten. |
| Pre-conditie | * Is ingelogd * Bezit minder dan 5 leningen * Er is minimaal één leenproduct geselecteerd |
| Main Scenario | 1. Het Systeem toont een scherm met geselecteerde leenproducten en de twee verplichte in te vullen velden:    1. Startdatum    2. einddatum 2. De Actor vult de verplichte velden in 3. Het Systeem checkt de ingevulde gegevens 4. De Actor bevestigd de ingevulde gegevens 5. Het Systeem slaat de lening op en zet status uitgeleend op true van geleende leenproducten |
| Post-Conditie Main Scenario | lening is aangemaakt |
| Alternatief scenario 1 | m.b.t stap 3   |  |  | | --- | --- | | Trigger | ingevulde gegevens kloppen niet | | Scenario | 1. Het Systeem heeft een fout gevonden in de opgegeven gegevens 2. Het Systeem geeft aan welke gegevens fouten veroorzaken. 3. Actor keer terug naar stap 2 van de use case | |
| Post-Conditie Alternatief Scenario 1 | Het Systeem heeft in de template aangegeven welke fout gemaakt is. |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en een lening te maken. We zijn in deze wireframe op de pagina lening maken. Deze wireframe laat zien hoe een student een lening aanmaakt. De student ziet in de tabel de gekozen producten die hij wil lenen. De startdatum van de lening is de huidige datum. De einddatum van de lening is maximaal 10 weken na de startdatum. Als de student de data heeft ingevoerd klikt hij op de button Lening maken om de lening te starten.



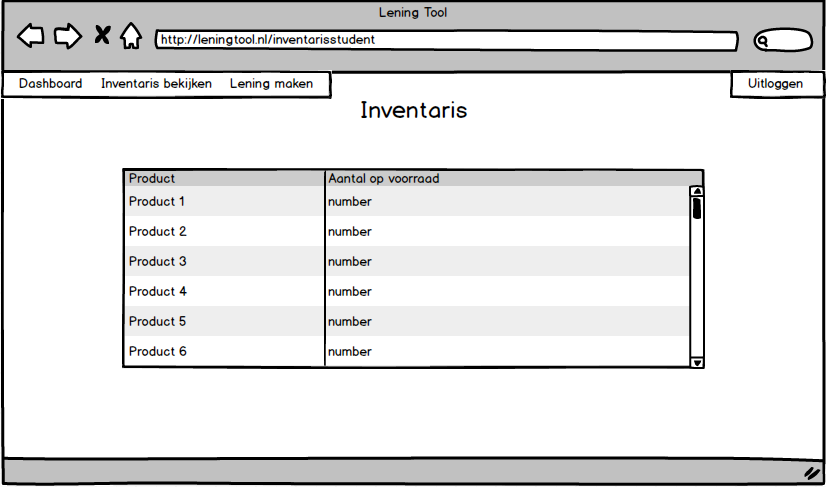
# Use Case 2: Inventaris bekijken

## Inventaris bekijken

|  |  |
| --- | --- |
| ID | 2 |
| Naam | Inventaris bekijken |
| Actor | Student |
| Samenvatting | Het Systeem toont een scherm met daarin een lijst van alle aanwezig leenproducten. De Actor kiest een Leenproducten[in] die hij wilt lenen. Het Systeem toont de details van het Leenproduct[o]. De Actor bevestigt zijn keuze. Het Systeem voegt het leenproduct toe in lening maken pagina. Het Systeem toont het overzicht van alle leenproducten weer. |
| Pre-conditie | * De actor moet ingelogd zijn. |
| Main Scenario | 1. Het Systeem toont een lijst van alle aanwezig leenproducten. 2. De Actor kiest een Leenproduct[in] die hij wilt lenen. 3. Het Systeem toont de details van gekozen Leenproducten[o]. 4. De Actor bevestigt zijn keuze. 5. Het Systeem voegt het leenproduct toe in lening maken pagina. 6. Het systeem toont overzicht van alle aanwezig leenproducten. |
| Post-Conditie Main Scenario | Actor wordt teruggestuurd naar de lening maken pagina. |
| Alternatief scenario 1 | m.b.t stap 4   |  |  | | --- | --- | | Trigger | Actor bevestigd zijn keuze niet | | Scenario | 1. Actor bevestigd zijn keuze niet 2. Actor keer terug naar stap 1 van de use case | |
| Post-Conditie Alternatief Scenario 1 | Systeem is ongewijzigd |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en een lening te maken. We zijn in deze wireframe op de pagina inventaris bekijken. In de tabel staat in de eerste kolom de productnaam en in de tweede kolom het aantal producten op voorraad.



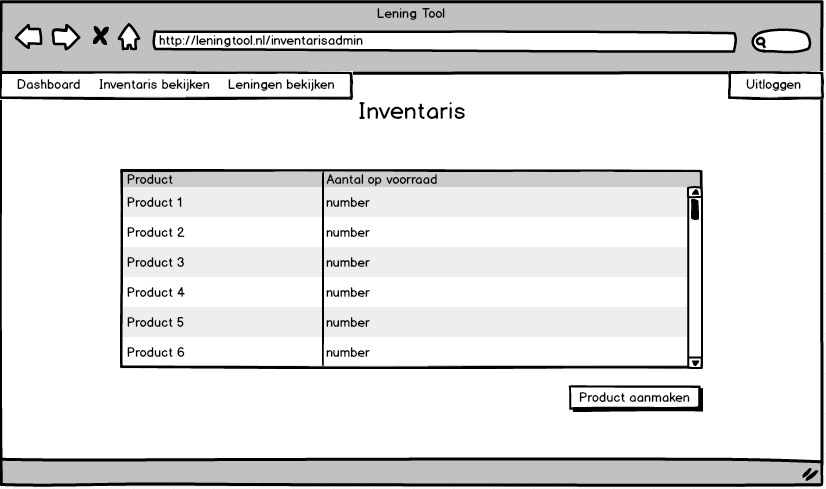
# Use Case 3: Inventaris beheren

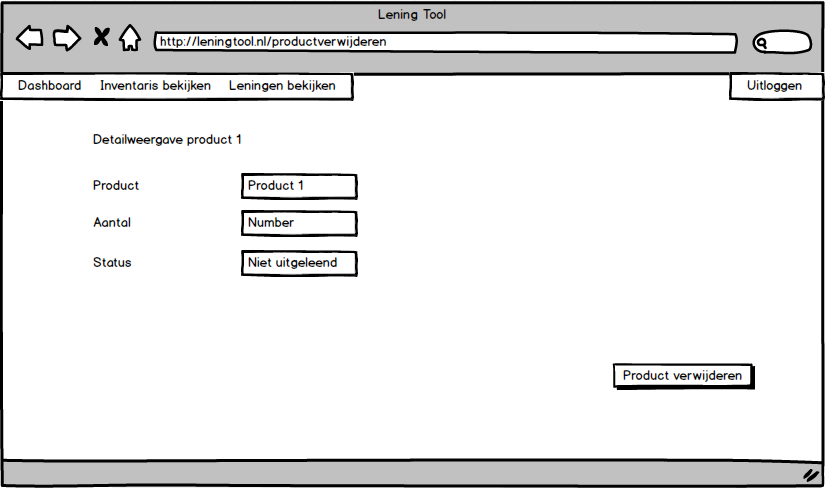
## Inventaris beheren

|  |  |
| --- | --- |
| ID | 3 |
| Naam | Inventaris beheren |
| Actor | Admin |
| Samenvatting | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o). De actor klikt(i) in de tabel op een Productnaam. Het systeem toont een formulier met de vooringevulde input fields Productnaam(o), Aantal(o), Beschrijving. Actor past in de input fields aan(i) wat de actor aan wilt passen en klikt(i) op de button ‘Aanpassingen opslaan’ als de actor klaar is. |
| Pre-conditie | * De actor moet ingelogd zijn. * Het systeem moet producten bevatten. |
| Main Scenario | 1. Systeem toont een pagina met de kolommen Productnaam en Aantal.  2. Actor klikt in de tabel op een Productnaam.  3. Systeem toont een formulier met ingevulde invoer velden Productnaam, Aantal en Beschrijving.  4. Actor past de nodige input velden aan.  5. Actor klikt op de button ‘Aanpassingen opslaan’.  6. Systeem slaat productaanpassingen op. |
| Post-Conditie Main Scenario | Er is een product aangepast in het systeem. |
| Alternatief scenario 1 | |  |  | | --- | --- | | Trigger | Main Scenario stap 6: de actor heeft een veld niet ingevuld | | Scenario | 1. Systeem toont een foutmelding dat er informatie mist.  2. Systeem gaat verder met stap 3. | |
| Post-Conditie Alternatief Scenario 1 | Systeem is ongewijzigd. |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en de huidige leningen te bekijken. We zijn in deze wireframe op de pagina inventaris bekijken. In de tabel ziet de admin in de eerste kolom de producten, in de tweede kolom de voorraad. Door op een product te klikken komt de admin op de detailpagina van een product om hier het product aan te passen. Rechts onderin staat een button om een nieuw product toe te voegen aan het systeem.



Na het klikken op een product is dit de detailpagina van het product. De admin kan hier de productnaam, het aantal producten en de status van het product aanpassen.

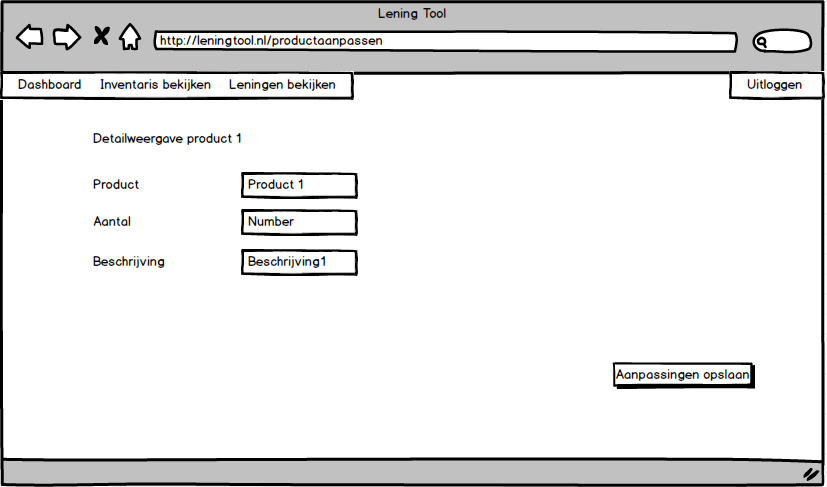
# Use Case 4: Product aanmaken

## Product aanmaken

|  |  |
| --- | --- |
| ID | 4 |
| Naam | Product aanmaken |
| Actor | Admin |
| Samenvatting | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o) en een button ‘Maak product aan’(o). Actor klikt(i) op button ‘Maak product aan’. Het systeem toont een formulier met een button ‘Product aanmaken’ en de volgende input fields: Productnaam, Aantal en Beschrijving. Actor vult Productnaam(i), Aantal(i) en Beschrijving(i) in en slaat door op de button ‘Product aanmaken’ te klikken het nieuwe product op. Het systeem geeft het nieuwe product een ProductID(i). |
| Pre-conditie | De actor moet ingelogd zijn. |
| Main Scenario | 1. Systeem toont een pagina met de kolommen Productnaam, Aantal, Uitleen status en een button ‘Product aanmaken’.  2. Actor klikt op de button ‘Product aanmaken’  3. Systeem toont een pagina met inputfields Productnaam, Aantal, Beschrijving en een button ‘Maak product’.  4. Actor verstrekt informatie.  5. Actor klikt op button ‘Product aanmaken’  6. Systeem voegt ProductID toe aan het product. |
| Post-Conditie Main Scenario | Er is een product toegevoegd aan het systeem. |
| Alternatief scenario 1 | |  |  | | --- | --- | | Trigger | Main Scenario stap 4: de actor heeft een veld niet ingevuld | | Scenario | 1. Systeem toont een foutmelding dat er informatie mist.  2. Systeem gaat verder met stap 3. | |
| Post-Conditie Alternatief Scenario 1 | Systeem is ongewijzigd. |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en de huidige leningen te bekijken. We zijn in deze wireframe op de pagina product aanmaken. De admin kan hier een productnaam invullen en het aantal producten dat in de inventaris moet. Vervolgens kan de admin op de button product aanmaken klikken om het product toe te voegen.



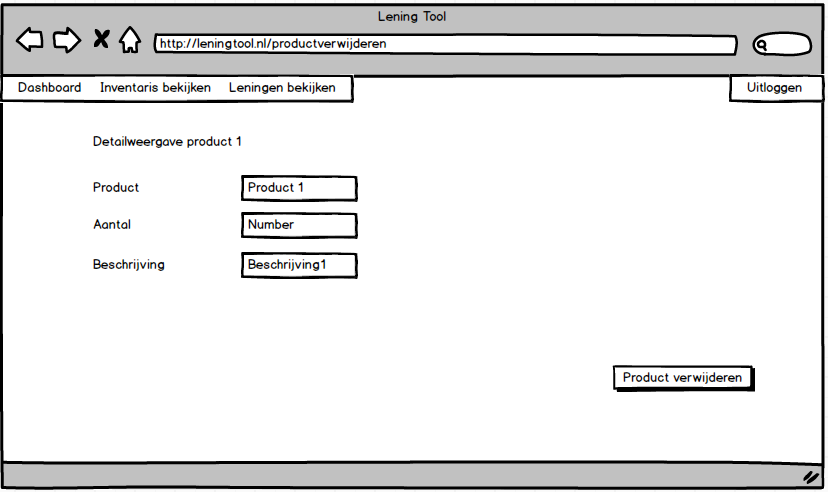
# Use Case 5: Product verwijderen

## Product verwijderen

|  |  |
| --- | --- |
| ID | 5 |
| Naam | Product verwijderen |
| Actor | Admin |
| Samenvatting | Het systeem toont een pagina waar een tabel met de volgende kolommen staat: Productnaam(o), Aantal(o). De actor klikt(i) in de tabel op een Productnaam. Het systeem toont een formulier met de voor ingevulde input fields Productnaam(o), Aantal(o) en een button ‘Product verwijderen’(o). Actor klikt(i) op de button ‘Product verwijderen’. Het Systeem toont een pop-up(o) met de tekst ‘Weet u zeker dat u dit product wilt verwijderen?’ en de keuzes ‘JA’ en ‘NEE’. Actor klikt(i) op ‘JA’. Systeem verwijdert product. |
| Pre-conditie | De actor moet ingelogd zijn.  Het systeem moet producten bevatten. |
| Main Scenario | 1. Systeem toont een pagina met de kolommen Productnaam, Aantal en Uitleen status.  2. Actor klikt in de tabel op een ProductID.  3. Systeem toont een formulier met voor ingevulde inputfields Productnaam, Aantal, Uitleen status en de button ‘Product verwijderen’.  4. Actor klikt op de button ‘Product verwijderen’.  5. Systeem toont een pop-up met de tekst ‘Weet u zeker dat u dit product wil verwijderen?’ en de opties ‘JA’ en ‘NEE’.  6. Actor klikt op ‘JA’.  7. Systeem verwijdert product |
| Post-Conditie Main Scenario | Er is een product verwijdert in het systeem. |
| Alternatief scenario 1 | |  |  | | --- | --- | | Trigger | Main Scenario stap 6: de actor klikt op ‘NEE’ | | Scenario | 1. Systeem gaat verder met stap 5. | |
| Post-Conditie Alternatief Scenario 1 | Systeem is ongewijzigd. |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken de huidige leningen te bekijken. We zijn in deze wireframe op de detailpagina van een product. Door links onderin op de button product verwijderen te klikken komt er een popup met de vraag of de admin zeker weet dat hij dit product wilt verwijderen. Als de admin op JA klikt kan de admin een product uit het systeem verwijderen.



# Use Case 6: Eigen lening bekijken

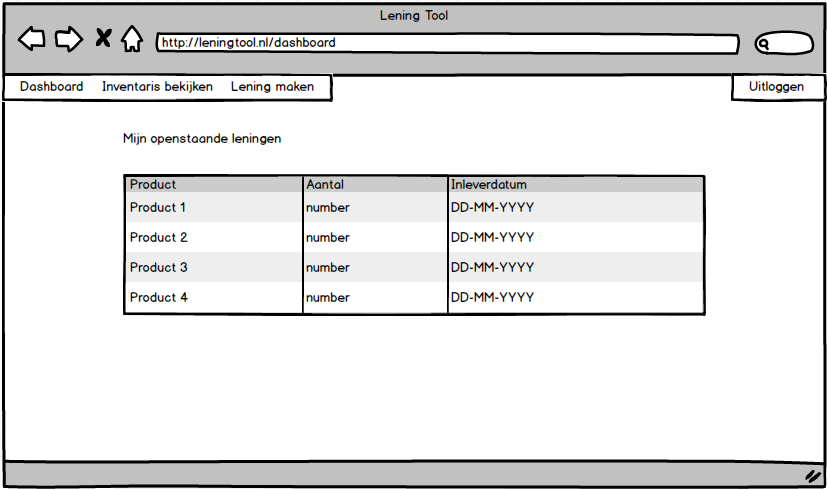
## Eigen lening bekijken

|  |  |
| --- | --- |
| ID | 6 |
| Naam | Eigen lening bekijken |
| Actor | Student |
| Samenvatting | De student wilt zijn openstaande leningen bekijken. De student navigeert naar de pagina waarop zijn leningen te zien zijn. De pagina geeft van de openstaande leningen de informatie weer. De informatie van de leningen zijn tegelijkertijd te zien op een enkele pagina.  De pagina geeft per lening  -De namen van de geleende object  -De hoeveelheid van geleende objecten  -De inleverdatum van de geleende objecten |
| Pre-conditie | De student heeft een lening open staan.  De gebruiker is ingelogd. |
| Main Scenario | 1. Student navigeert naar de pagina waarop leningen zijn weergegeven. 2. Systeem haalt informatie over leningen van deze student uit database. 3. Systeem geeft opgehaalde informatie weer. |
| Post-Conditie Main Scenario | Systeem toont de leningen van de student. |

## 

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om de inventaris te bekijken en een lening te maken. We zijn in deze wireframe op de pagina dashboard. Dit is de pagina die een student ziet als hij inlogt. In de tabel ziet hij zijn openstaande leningen staan.



# Use Case 7: Verlopen lening

## Verlopen lening

|  |  |
| --- | --- |
| ID | 7 |
| Naam | Verlopen lening |
| Actor | Systeem |
| Samenvatting | Het systeem stuurt een email bericht dat een lening is verlopen. |
| Pre-conditie | Er staat een lening open. Deze lening overschrijdt de inleverdatum. |
| Main Scenario | 1. Lening overschrijdt de inleverdatum. 2. Systeem stuurt een melding aan student dat de lening is verlopen. |
| Post-Conditie Main Scenario | Het systeem verstuurt een email bericht aan de leerling. |

## Wireframe

Omdat dit een automatisering van het systeem is, zit bij deze use case geen wireframe.

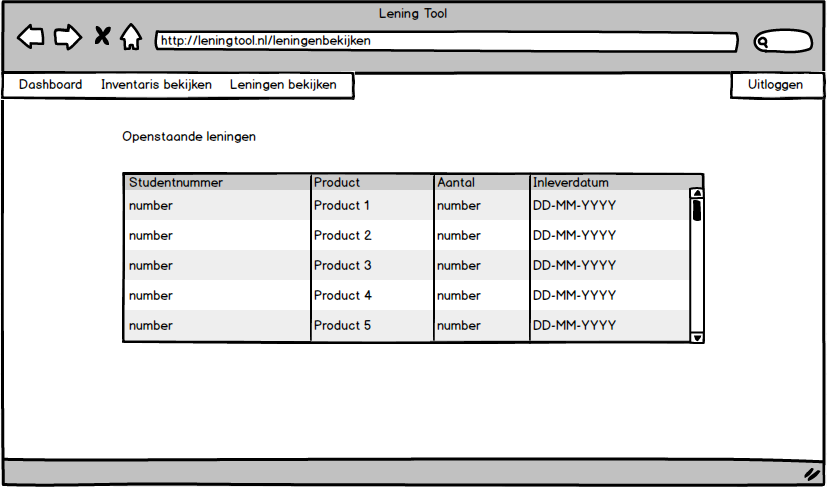
# Use Case 8: Lening bekijken

## Lening bekijken

|  |  |
| --- | --- |
| ID | 8 |
| Naam | Lening bekijken |
| Actor | Admin |
| Samenvatting | Actor wil de leningen inzien. Actor vraagt de leningen op (lening.afgerond [in], lening.teLaatIngeleverd [in]), en Systeem weergeeft ze (student.studentnummer [out], lening.einddatum [out], leenproduct.naam [out]). |
| Pre-conditie | Er moet een lening gemaakt zijn. |
| Main Scenario | 1. Systeem toont een menu met opties: "Openstaand", "Afgerond".  2. Actor kiest een optie.  3. Systeem toont een menu met opties: "Op tijd", "Te laat".  4. Actor kiest een optie.  5. Systeem weergeeft de leningen (lener (studentnummer), einddatum, uitgeleende producten (naam)), waarvan de waarden (afgerond, teLaatIngeleverd) overeenkomen met de weergave optie.  6. Actor kiest een lening.  7. Systeem weergeeft de lening details (lener (voornaam, achternaam, studentnummer), startdatum, einddatum, status, de uitgeleende producten (naam, korte beschrijving, aantal, uitgeleend, functioneel)). |
| Post-Conditie Main Scenario | Systeem is ongewijzigd. |
| Alternatief scenario 1 | |  |  | | --- | --- | | Trigger | Stap 5. Er zijn geen openstaande/te late/afgeronde leningen. | | Scenario | 1. Systeem toont in een informatiebericht dat er geen leningen zijn van dit type. | |
| Post-Conditie Alternatief Scenario 1 | Systeem is ongewijzigd. |

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en de huidige leningen te bekijken. We zijn in deze wireframe op de leningen bekijken pagina. De admin kan in de tabel in de eerste kolom het studentnummer zien, dus van wie de lening is. In de tweede kolom ziet de admin het leenproduct. In de derde kolom het aantal producten van dat product en in de derde kolom de inleverdatum van het product.



# Use Case 9: Lening afronden

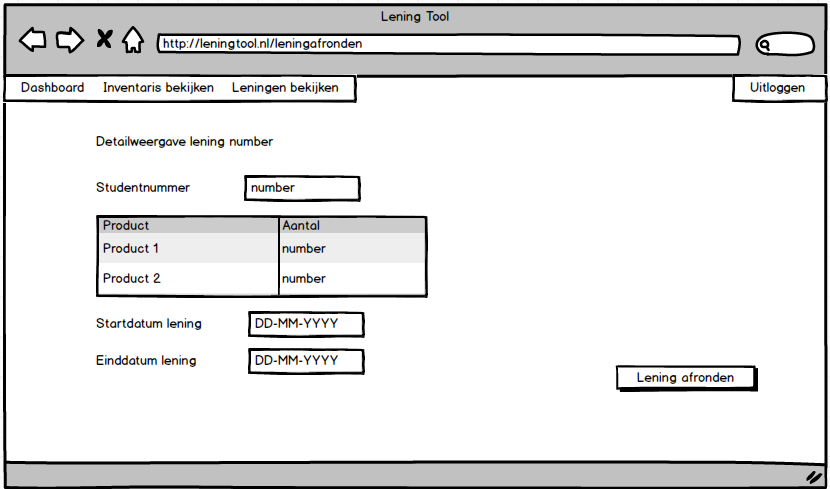
## Leningen afronden

|  |  |
| --- | --- |
| ID | 9 |
| Naam | Leningen afronden |
| Actor | Admin |
| Samenvatting | Actor wil leningen afronden. Actor vraagt de leningen op (lening.afgerond [in], lening.teLaatIngeleverd [in]), en Systeem weergeeft ze (student.studentnummer [out], lening.einddatum [out]). Actor kiest een lening , Systeem weergeeft de lening details (student.voornaam [out], student.achternaam [out], lening.startdatum [out], leenproduct.naam [out], leenproduct.beschrijving [out]), Actor kiest producten (leenproduct.aantal [out], leenproduct.functioneel [out]), Actor wijzigt deze (leenproduct.aantal [in], leenproduct.functioneel [in]), Actor bevestigt, en Systeem wijzigt de leningen naar afgerond (lening.afgerond [in]). |
| Pre-conditie | Er moet een lening gemaakt zijn. |
| Main Scenario | 1. Systeem toont een menu met opties: "Openstaand", "Afgerond".  2. Actor kiest "Openstaand".  3. Systeem toont een menu met opties: "Op tijd", "Te laat".  4. Actor kiest een optie.  5. Systeem weergeeft de leningen (lener (studentnummer), einddatum, uitgeleende producten (naam)), waarvan de waarden (afgerond, teLaatIngeleverd) overeenkomen met de weergave optie.  6. Actor kiest een lening.  7. Systeem weergeeft de lening details (lener (voornaam, achternaam, studentnummer), startdatum, einddatum, status, de uitgeleende producten (naam, korte beschrijving, aantal, functioneel).  8. Actor kiest producten, geeft per product een nieuw aantal (als het product ontbreekt), geeft aan of een product niet functioneel is, en keert terug naar de lening details.  8. Actor kiest "Afronden".  9. Systeem zet de geselecteerde leningen op "Afgerond". |
| Post-Conditie Main Scenario | De leningen zijn gewijzigd. Ze hebben de "Afgerond" status. Indien van toepassing, zijn product waarden gewijzigd (aantal en functioneel). |
| Alternatief scenario 1 | (Zie alternatief scenario 1 van "Lening bekijken".) |
| Post-Conditie Alternatief Scenario 1 | - |
| Alternatief Scenario 2 | |  |  | | --- | --- | | Trigger | Stap 0. Actor wil per lener en per lening afronden. | | Scenario | 1. Actor kiest een lener.  2. Systeem weergeeft de lener details (studentnummer, voornaam, achternaam, HU-mail, mobiel nummer), en zijn leningen (einddatum, uitgeleende producten (naam)).  3. Actor kiest een lening.  4. Systeem weergeeft de lening details (startdatum, einddatum, status, de uitgeleende producten (naam, korte beschrijving)).  5. Actor kiest "Afronden".  6. Systeem zet de lening op afgerond. | |
| Post-Conditie Alternatief Scenario 2 | De lening is gewijzigd. Het heeft de "Afgerond" status. |

## 

## Wireframe

Linksboven ziet u het menu met de navigatie naar het dashboard, om het inventaris te bekijken en de huidige leningen te bekijken. We zijn in deze wireframe op de leningen bekijken detailpagina. Als de admin in de leningen bekijken tabel op een product klikt komt hij in de detailweergave van die lening. Hier kan de admin vervolgens de lening afronden als hij de producten weer binnen heeft gekregen van de student. Om dit te doen klikt de admin op de button lening afronden.



# Technische Requirements

De applicatie dient gerealiseerd te worden in het front-end framework Angular. De data wordt opgeslagen in een Firebase Realtime Database. De applicatie wordt getest met de front-end testautomatie tool CyPress. De applicatie moet op zowel desktop als mobiele apparaten bruikbaar zijn. De broncode wordt beheerd in een Git repository (GitHub).