Functioneel Ontwerp

Use cases:

UC1 t/m UC19

Niveau: 2  
Versie: 1.5  
Datum: 13-10-2025

Team SE  
2025

Versiebeheer

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Wijzigingen |
| 0.1 | 9-6-2025 | Initiële versie |
| 1.0 | 12-9-2025 | UC1 t/m 3 |
| 1.1 | 16-9-2025 | UC4 t/m 6 |
| 1.2 | 23-9-2025 | UC7 t/m 9 |
| 1.3 | 29-9-2025 | UC10 t/m 13 |
| 1.4 | 07-10-2025 | UC13 t/m 15 |
| 1.5 | 13-10-2025 | UC15 t/m 19 |

Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Ontvangers |
| 1.0 | 12-9-2025 | John Brouwers |
| 1.1 | 16-9-2025 | John Brouwers en Freek de Jong |
| 1.2 | 23-9-2025 | John Brouwers en Freek de Jong |
| 1.3 | 29-9-2025 | John Brouwers en Freek de Jong |
| 1.4 | 07-10-2025 | John Brouwers en Freek de Jong |
| 1.5 | 13-10-2025 | John Brouwers en Freek de Jong |

Inhoud

[1 Inleiding 1](#_Toc202284050)

[2 Domeinanalyse 2](#_Toc202284051)

[2.1 Huidige situatie 2](#_Toc202284052)

[2.2 Gewenste situatie 2](#_Toc202284053)

[2.3 Use case diagram 3](#_Toc202284054)

[3 Use cases 4](#_Toc202284055)

[3.1 UC1 Tonen boodschappenlijsten 4](#_Toc202284056)

[3.2 UC2 Tonen boodschappenlijst 5](#_Toc202284057)

[3.3 UC3 Tonen producten 6](#_Toc202284058)

[3.4 UC4 Kiezen kleur boodschappenlijst 7](#_Toc202284059)

[3.5 UC5 Plaatsen product op boodschappenlijst 8](#_Toc202284060)

[3.6 UC6 Inloggen 9](#_Toc202284061)

[3.7 UC7 Delen boodschappenlijst 10](#_Toc202284062)

[3.8 UC8 Zoeken producten 12](#_Toc202284063)

[3.9 UC9 Filteren producten – ON HOLD 14](#_Toc202284064)

[3.10 UC10 Aanpassen product aantal 14](#_Toc202284065)

[3.11 UC11 Tonen meest verkochte producten 15](#_Toc202284066)

[3.12 UC12 Toevoegen productcategorieën 16](#_Toc202284067)

[3.13 UC13 Tonen klanten/boodschappenlijst per product 18](#_Toc202284068)

[3.14 UC14 Toevoegen prijzen 19](#_Toc202284069)

[3.15 UC15 Toevoegen THT datum 19](#_Toc202284070)

[3.16 UC16 Controleren leeftijd – ON HOLD 19](#_Toc202284071)

[3.17 UC17 Opslaan boodschappenlijst 19](#_Toc202284072)

[3.18 UC18 Opslaan boodschappenlijstitems 19](#_Toc202284073)

# Inleiding

Dit document bevat het functionele ontwerp en is gebaseerd op de requirementsanalyse.

Het productdoel van de applicatie is het proces van boodschappen picken in de supermarkt makkelijker maken.

In het volgende hoofdstuk is het domein vastgelegd en een use case diagram. In het hoofdstuk erna is per use case een uitwerking gemaakt met wireframe(s) aangevuld met aantekeningen. Indien nodig zijn er ondersteunende diagrammen toegevoegd. In Bijlage 1 Aanpak Functioneel Ontwerp is de aanpak van dit ontwerp beschreven.

De bron bestanden, zoals Visual Paradigm bestanden van de diagrammen en Balsamiq bestanden van de wireframes, zijn op GitHub beschikbaar. Heb je andere inzichten of voorkeuren, dan mag je het ontwerp aanpassen.

Succes met de use case!

Team SE

# Domeinanalyse

In dit hoofdstuk staat een beschrijving van het domein dat is voortgekomen uit een gesprek met de opdrachtgever. Het domein valt samen met de opgestelde requirements. Om het domein inzichtelijk te maken is gebruik gemaakt van een domeinmodel en een use case diagram.  
Het hoofdstuk begint met de huidige situatie, gevolgd door een beschrijving van de gewenste situatie. Daarna volgen de diagrammen met beschrijvingen van het domein.

## Huidige situatie

In de huidige situatie blijkt dat klanten moeite hebben om het overzicht te houden welke producten ze wel en niet hebben gepakt in de winkel. Dit geldt met name voor ouderen.

## Gewenste situatie

In de gewenste situatie kan een klant overzicht houden door een boodschappenlijst bij te houden. De boodschappenlijst kan vooraf worden samengesteld. Per product is aangegeven of het op voorraad is zodat de klant vooraf op zoek kan gaan naar een alternatief.

Hieronder staat het domeinmodel van de applicatie in de gewenste situatie. Een geïnteresseerde oriënteert zich op de vaardigheden van een student-webdeveloper. Als de vaardigheden gegroepeerd moeten worden of bestaan uit categorieën dan zal hiervoor een nieuwe functionele requirement moeten worden opgenomen. Op dit moment is nog niet duidelijk of dit noodzakelijk is.

A diagram of a company

AI-generated content may be incorrect.

Figuur 1 Domeinmodel van de gewenste situatie

## Use case diagram

Momenteel is dit use case diagram nog eenvoudig diagram en hebben we te maken met één actor, de klant.

A diagram of a product

AI-generated content may be incorrect.

Figuur 2 Use case diagram van de gewenste situatie

# Use cases

Dit hoofdstuk beschrijft de use cases vanuit functioneel perspectief.

## UC1 Tonen boodschappenlijsten

|  |  |
| --- | --- |
| Use case naam | Tonen boodschappenlijsten |
| Actoren | Klant |
| Beschrijving | Een klant bekijkt de gemaakte boodschappenlijsten |
| Precondities | Er zijn door de klant eerder boodschappenlijsten gemaakt, anders is de lijst leeg. |
| Hoofdscenario | 1. De klant bekijkt de boodschappenlijsten |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | Er zijn geen postcondities. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A white card with black text

AI-generated content may be incorrect.

Figuur 3 Wireframe tonen boodschappenlijsten

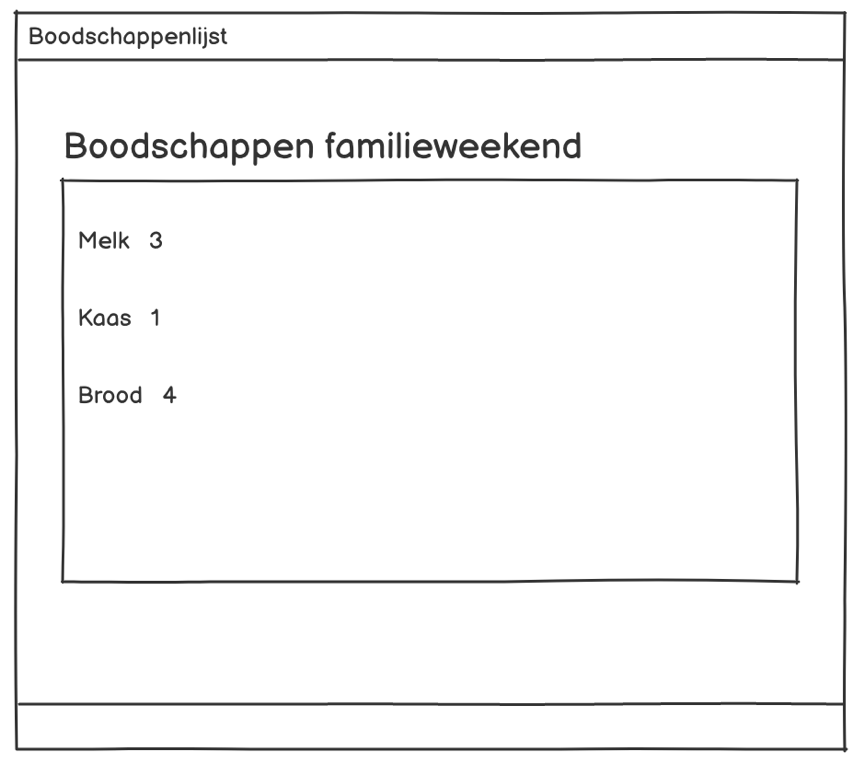
### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC2 Tonen boodschappenlijst

|  |  |
| --- | --- |
| Use case naam | Tonen boodschappenlijst |
| Actoren | Klant |
| Beschrijving | Een klant bekijkt een gemaakte boodschappenlijst |
| Precondities | De klant heeft de boodschappenlijst eerder samengesteld |
| Hoofdscenario | 1. De klant bekijkt de boodschappenlijst |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | Er zijn geen postcondities. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.



Figuur 4 Wireframe tonen boodschappenlijst

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC3 Tonen producten

|  |  |
| --- | --- |
| Use case naam | Tonen producten |
| Actoren | Klant |
| Beschrijving | Een klant bekijkt de producten lijst |
| Precondities | Geen |
| Hoofdscenario | 1. De klant opent de productenlijst 2. De klant bekijkt de productenlijst |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | Er zijn geen postcondities. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screen shot of a product

AI-generated content may be incorrect.

Figuur 5 Wireframe tonen producten

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC4 Kiezen kleur boodschappenlijst

|  |  |
| --- | --- |
| **Use case naam** | **Kiezen  kleur boodschappenlijst** |
| Actoren | Klant |
| Beschrijving | Een klant kiest de kleur van een boodschappenlijst |
| Precondities | Er is al een boodschappenlijst |
| Hoofdscenario | 1. De klant klikt op een bestaande boodschappenlijst 2. De klant klikt op de knop om naar de pagina te gaan om de kleur te wijzigen. 3. De klant kan een HEX kleurcode invoeren. 4. De klant klikt op bewaar knop om de kleur op te slaan. 5. De app gaat terug naar de boodschappenlijst pagina |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | Een klant kan zien dat de kleur van een boodschappenlijst is veranderd. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

*Figuur 6 Wireframe kiezen kleur boodschappenlijst*

1. Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC5 Plaatsen product op boodschappenlijst

|  |  |
| --- | --- |
| **Use case naam** | **Plaatsen product op boodschappenlijst** |
| Actoren | Klant |
| Beschrijving | Een klant kan een product op zijn eigen boodschappenlijst plaatsen |
| Precondities | Er zijn producten aanwezig  De klant heeft een boodschappenlijst, waarbij nog niet alle producten van de productenlijst aanwezig zijn op de boodschappenlijst. |
| Hoofdscenario | 1. De klant bekijkt een boodschappenlijst 2. Naast de boodschappen op die lijst ziet de klant het overzicht van producten. 3. De klant klikt op een product in de “Producten lijst” om deze toe te voegen aan de eigen boodschappenlijst 4. Het product verplaatst nu naar boodschappenlijst en is niet meer zichtbaar in de “Producten lijst” |
| Alternatieve scenario’s | 2a) Geen producten om toe te voegen   * Op de productenlijst komt een melding “Er zijn geen product meer om toe te voegen”. |
| Postcondities | De boodschappenlijst is aangevuld met het/de nieuwe product(en). |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer screen

AI-generated content may be incorrect.

*Figuur 7 Wireframe Plaatsen product op boodschappenlijst*

1. Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC6 Inloggen

|  |  |
| --- | --- |
| **Use case naam** | **Inloggen** |
| Actoren | Klant |
| Beschrijving | De klant logt in om zijn informatie (b.v. boodschappenlijsten) in te laden. |
| Precondities | De klant heeft een account en is nog niet ingelogd. |
| Hoofdscenario | 1. De klant opent de app. 2. Het loginscherm wordt weergegeven. 3. De klant voert gebruikersnaam en wachtwoord in. 4. De klant klikt op “Login”. 5. De app controleert de gegevens. 6. [De gegevens zijn juist] 7. De app gaat door naar de boodschappenlijsten pagina. 8. De app toont de naam van de ingelogde gebruiker op het scherm bovenaan. 9. [De gegevens zijn onjuist] 10. De app toon een melding “Ongeldige inloggegevens” en de klant wordt niet ingelogd. |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | De klant is ingelogd waardoor persoonlijke informatie en boodschappenlijsten van de klant opgehaald kunnen worden. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

*Figuur 8 Wireframe Inloggen*

1. Evil user stories

* Als hacker wil ik weten of een account bestaat of niet zodat ik gerichter kan proberen een account te hacken.
* Als hacker wil ik veelvoorkomende eenvoudige wachtwoorden proberen om toegang te krijgen tot een account

## UC7 Delen boodschappenlijst

|  |  |
| --- | --- |
| Use case naam | Delen boodschappenlijst |
| Actoren | Klant |
| Beschrijving | De klant kan een boodschappenlijst delen. Door bij een boodschappenlijst te klikken op “Delen boodschappenlijst” maakt de applicatie een export van de boodschappen naar een bestand. In een melding geeft de applicatie weer of het opslaan is gelukt en waar het bestand is geplaatst. |
| Precondities | 1. De klant heeft een boodschappenlijst 2. De klant is al op het overzicht van de boodschappenlijsten |
| Hoofdscenario | 1. De klant klikt op “Deel boodschappenlijst” 2. De app start de export naar het bestandssysteem    1. [opslaan gelukt]       1. De app toont een melding dat het opslaan is gelukt.       2. De app toont de locatie van het opgeslagen bestand.    2. [opslaan mislukt]       1. De app toont een melding dat het opslaan niet is gelukt.       2. De app toont informatie over de foutmelding (Exception.Message). 3. De klant kan de melding sluiten |
| Alternatieve scenario’s | Er zijn geen alternatieve scenario’s. |
| Postcondities | Als het opslaan is gelukt staat er op de aangegeven plaats een bestand met daarin de informatie van de opgeslagen boodschappenlijst. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

*Figuur 9 Wireframe delen boodschappenlijst*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC8 Zoeken producten

|  |  |
| --- | --- |
| Use case naam | Zoeken producten |
| Actoren | Klant |
| Beschrijving | Bij het vullen van de boodschappenlijst kan de klant zoeken in de beschikbare producten. Hierbij wordt alleen gezocht op de productnaam en niet op andere informatie zoals de naam van de categorie |
| Precondities | 1. De klant heeft een boodschappenlijst geopend |
| Hoofdscenario | 1. De klant ziet de producten op de lijst en tevens de beschikbare producten die nog niet op de lijst staan. 2. Het zoekveld is leeg    1. [zoeken op beschikbaar product]       1. De klant voert een zoekwaarde in die voor komt in de naam van een product en klikt op “zoeken”       2. De producten die voldoen aan de zoekopdracht worden weergegeven    2. [zoeken niet beschikbaar product]       1. De klant voert een zoekwaarde in die NIET voor komt in de naam van een product en klikt op “zoeken”       2. Er worden geen producten weergegeven       3. Er wordt een melding gegeven dat er geen producten zijn gevonden    3. [zoeken categorienaam]       1. De klant voert een zoekwaarde in die NIET voor komt in de naam van een product maar WEL in de naam van een categorie en klikt op “zoeken”       2. Er worden geen producten weergegeven       3. Er wordt een melding gegeven dat er geen producten zijn gevonden 3. De klant verwijderd de zoekwaarde en klikt op “zoeken” 4. De klant ziet weer alle beschikbare producten die nog niet op de lijst staan (zoals in stap 1) |
| Alternatieve scenario’s | Er is geen alternatief scenario |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

*Figuur 10: Wireframe zoeken producten*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC9 Registreren

|  |  |
| --- | --- |
| Use case naam | Zoeken producten |
| Actoren | Klant |
| Beschrijving | De klant kan bij de inlogpagina ervoor kiezen om een account aan te maken en zich dus te kunnen registreren, en met de nieuwe inloggegevens die ze hebben aangemaakt te kunnen inloggen. |
| Precondities | * + 1. De klant kan inloggen met de nieuwe gegevens |
| Hoofdscenario | 1. De klant ziet de inlogpagina met tekst onder de knop naar waar je je kan registreren. 2. De klant klikt erop en vult de mail en wachtwoord in die ze willen gebruiken om in te loggen. 3. De klant klikt op de registreren knop en word teruggestuurd naar de login pagina als alles succesvol was. 4. De klant logt in met de nieuwe gegevens en klikt op de login knop. 5. De klant zit nu succesvol in de boodschappen app. |
| Alternatieve scenario’s | 1. Als het niet is gelukt om een account aan te maken dan moet de klant het opnieuw proberen. |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.



*Figuur 11: Wireframe Registreren*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC10 Aanpassen product aantal

|  |  |
| --- | --- |
| Use case naam | Aanpassen product aantal |
| Actoren | Klant |
| Beschrijving | Voor de producten in een boodschappenlijst is het handig als de klant het aantal kan aanpassen. Dit kan door de + en – iconen naast de producten.  Het aantal mag niet lager zijn dan 0 en als een product op 0 staat wordt hij niet automatisch verwijderd. |
| Precondities | * De klant heeft een boodschappenlijst geopend * De boodschappenlijst heeft minimaal 1 product |
| Hoofdscenario | 1. De klant ziet de producten op de boodschappenlijst en het actuele aantal 2. [klik op de +]    1. Het aantal naast het product wordt verhoogd met 1 3. [klik op de -]    1. [het huidige aantal = 0]       1. Dan blijft het aantal 0    2. [het huidige aantal > 0]       1. Het aantal wordt verlaagd met 1 |
| Alternatieve scenario’s | 2a.1) Het aantal is gelijk aan de beschikbare voorraad van een product   * De teller wordt niet verder verhoogd. |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

Figuur 11: Wireframe Aanpassen product aantal

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC11 Tonen meest verkochte producten

|  |  |
| --- | --- |
| Use case naam | Tonen meest verkochte producten |
| Actoren | Klant |
| Beschrijving | In de app kan de klant zien welke 5 producten er het meest worden verkocht.  Het best verkocht wordt vastgesteld op basis van de kwantiteit van het aantal producten. |
| Precondities | Er staan producten op de boodschappenlijsten met een aantal > 0. |
| Hoofdscenario | 1. De klant opent de pagina “Meest verkocht”. 2. De klant kan de volgende informatie inzien:    * Orde nummer    * Naam    * Aantal verkocht    * Aantal in voorraad |
| Alternatieve scenario’s | 2a) Als er geen producten nog zijn verkocht   * Er wordt getoond dat er nog geen producten zijn verkocht. |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a computer

AI-generated content may be incorrect.

Figuur 12: Wireframes tonen meest verkochte producten

### Evil user stories

* Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC12 Toevoegen productcategorieën

|  |  |
| --- | --- |
| **Use case naam** | **Toevoegen productcategorie aan een product** |
| Actoren | Admin / Klant |
| Beschrijving | De actor voegt een productcategorie toe aan een product. De functionaliteit is beschikbaar op de pagina productcategorie.    Deze pagina is beschikbaar via de tabs onderin het scherm. Op het scherm worden de categorieën getoond. Na een klik op een categorie opent een scherm. Daarop zijn links de producten in de categorie te zien. Rechts op het scherm staan producten die na een klik worden toegevoegd aan de productcategorie. |
| Precondities | De gebruiker is ingelogd, waarbij de rol niet van belang is.  Je bent op de “productcategorie” weergave (via tab productcategorie) |
| Hoofdscenario | 1. Klik op een categorie 2. Er wordt een pagina geopend met daarin de productcategorieën. 3. Je kiest een productcategorie 4. Een detailpagina wordt geopend 5. [Er zijn geen producten in de productcategorie] 6. Je selecteert een product rechts in de keuzelijst. 7. Het product wordt getoond onder de productcategorie en verdwijnt uit de lijst. 8. [Alle producten zijn in de productcategorie] 9. [Een aantal producten zijn in de productcategorie] 10. [Een product wordt gezocht voordat deze wordt toegevoegd aan de productcategorielijst] |
| Alternatieve scenario’s | 4b [Alle producten zijn in de productcategorie]   1. Onder producten staat een melding dat alle producten zijn toegevoegd aan de productcategorie.   4c [Een aantal producten zijn in de productcategorie]   1. Je selecteert een product rechts in de keuzelijst. 2. Het product wordt getoond onder de productcategorie en verdwijnt uit de lijst.     4d [Zoeken producten]   1. Er wordt een pagina geopend met daarin de productcategorieën. 2. Je zoekt een product door in de zoekbalk een aantal karakters in te voeren. 3. De lijst toont de producten die het trefwoord in de naam bevatten. 4. Je selecteert een product rechts in de keuzelijst. |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

A screenshot of a screen

AI-generated content may be incorrect.

*Figuur 13: Wireframes toevoegen productcategorieën*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC13 Tonen klanten/boodschappenlijst per product

|  |  |
| --- | --- |
| Use case naam | Tonen klanten/boodschappenlijst per product |
| Actoren | Admin |
| Beschrijving | In de app kunnen de klanten worden getoond en welke boodschappenlijst per product zit. |
| Precondities | Er staan producten op de boodschappenlijsten met een aantal > 0. |
| Hoofdscenario | 1.De actor opent de klanten lijst en producten lijst  2. De actor ziet de volgende informatie:   * Klantnaam * Product * Aantal boodschappenlijsten |
| Alternatieve scenario’s | 1a) Als er geen klanten gevonden worden   * Er wordt getoond dat de data gefaald is om te laden. |
| Postcondities | Er zijn geen postcondities |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.



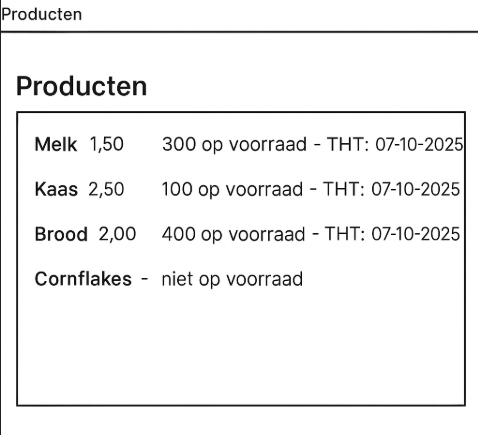
Figuur 13: Wireframes tonen klanten/boodschappenlijst per product

## UC 14 Toevoegen prijzen

|  |  |
| --- | --- |
| Use case naam | Toevoegen Prijzen |
| Actoren | Klant/Admin |
| Beschrijving | Het systeem moet prijzen kunnen toevoegen aan producten, opslaan en tonen. Wanneer producten worden geladen, moet de prijs automatisch worden opgehaald en zichtbaar zijn voor de gebruiker. |
| Precondities | * De Product class is aanwezig in het systeem. * De ProductRepository bevat productdata. * Er bestaat een ProductView waarin producten getoond worden. |
| Hoofdscenario | * De admin voegt een prijs toe bij het aanmaken van een product. * De ProductRepository slaat het product met prijs op. * Bij het laden van producten haalt het systeem ook de prijs op. * In de ProductView toont het systeem per product de bijbehorende prijs. |
| Alternatieve scenario’s | (n.v.t. – deze use case richt zich enkel op toevoegen, ophalen en tonen van prijzen) |
| Postcondities | * Elk product bevat een prijs. * De prijs wordt correct weergegeven in de productlijst. * De prijsinformatie is consistent geladen vanuit de repository. |

### Evil user stories

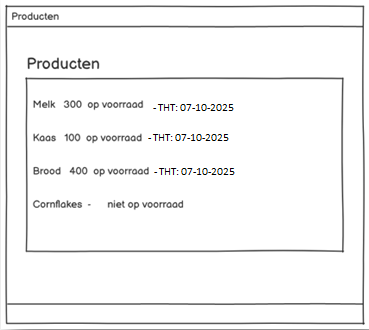
Geen evil user stories kunnen bepalen. Geen nadere actie nodig.



Figuur 14: Wireframe prijs met producten en THT

## UC15 Toevoegen THT datum

Dit is een uitbreiding op UC3 en daarom niet verder hier uitgewerkt. De grafische implementatie is vrij is te vullen door de developer.



Figuur 15: Wireframe tonen THT

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC17 en UC18

Leveren geen nieuwe functionaliteit vanuit gebruikersperspectief. Dus geen use case beschrijving en grafische implementatie.

## UC19

|  |  |
| --- | --- |
| Use case naam | Product in database en nieuw product aanmaken |
| Actoren | Admin |
| Beschrijving | Het systeem moet producten kunnen ophalen uit de database en het mogelijk maken dat een admin nieuwe producten toevoegt. Wanneer een nieuw product is aangemaakt, moet dit automatisch zichtbaar zijn in de productlijst, zonder dat de gebruiker de pagina hoeft te vernieuwen. |
| Precondities | * De databaseverbinding is actief. * De ProductRepository is aanwezig en heeft data * De ProductView toont bestaande producten * De gebruiker is ingelogd met de rol Admin |
| Hoofdscenario | * De admin opent de view en kiest “Nieuw product toevoegen” * Het systeem opent de view voor invoer van productgegevens * De admin voert de gegevens in (naam, prijs, voorraad) en bevestigt. * De ProductService slaat het nieuwe product op in de database * Het systeem laadt automatisch de bijgewerkte lijst met producten * Het nieuwe aangemaakte product word direcht zichtbaar in de View |
| Alternatieve scenario’s | • De gebruiker heeft geen adminrechten → het systeem toont een foutmelding of verbergt de optie. • De database is tijdelijk niet bereikbaar → het systeem toont een foutmelding en annuleert het toevoegen. |
| Postcondities | * Het nieuwe product is opgeslagen in de database. * Het product is direct zichtbaar in de productlijst. * De lijst in ProductView is bijgewerkt zonder herstart of handmatige verversing. * Enkel admins hebben toegang tot de functie voor nieuw product aanmaken. |

Afbeelding met tekst, schermopname, Lettertype, nummer

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 16: Wireframe nieuwe product toevoegen

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.