Het ontwerp van de webwinkel bij Albert Heijn.





Door Kevin van Loon (2126940) en Rudwan Akhiat (2130818)

*Ingeleverd op 7 januari 2018*

Inhoud

[1. Inleiding 3](#_Toc503008029)

[2. Geschiedenis en achtergrond van Albert Heijn 4](#_Toc503008030)

[3. Wat zijn wireframes? 5](#_Toc503008031)

[3.1 Wanneer worden er wireframes gebruikt? 5](#_Toc503008032)

[3.2 Wireframes van AH. 5](#_Toc503008033)

[4. Wat zijn use cases? 6](#_Toc503008034)

[4.1 AH & Use Cases 8](#_Toc503008035)

[4.2 AH & Use Case Diagram 9](#_Toc503008036)

[5. Wat is een ERD? 10](#_Toc503008037)

[5.1 Het ERD van de webwinkel van Albert Heijn. 10](#_Toc503008038)

[6. Consistentie 11](#_Toc503008039)

[7. Korte samenvatting en toelichting 12](#_Toc503008040)

[Literatuurlijst 13](#_Toc503008041)

[Bijlage 1 : De wireframes 14](#_Toc503008042)

[Bijlage 2 : Alle use case specificaties 23](#_Toc503008043)

[Bijlage 3 : Use case diagram 29](#_Toc503008044)

1. Inleiding  
De betekenis van reverse engineering is het onderzoeken van een product om daaruit af te leiden wat de eisen zijn waaraan het product probeert te voldoen, om er zo de precieze interne werking van te achterhalen. Voor de vakken professionele vaardigheden en requirements engineering moet er een trio opdracht gemaakt worden. Er moet een webshop gekozen worden waar het begrip reverse engineering op toegepast moet worden.   
 De voorkeur gaat uit naar Albert Heijn, omdat alle participanten van de trio opdracht werken bij Albert Heijn. Er was dus sprake van voorkennis en een kijkje in het systeem van de Albert Heijn. Ook is er genoeg interesse aanwezig als er wordt gesproken over Albert Heijn, zo is iedereen tevreden over de Albert Heijn. Een breder onderzoek in de webwinkel is dan een leuke toevoeging. Als de doelstelling is behaald, kan het ook het geval zijn dat er meerdere onderdelen verbeterd kunnen worden. Functies zoals: processen die vastlopen, ontbrekende informatie, foutmeldingen tijdens het ophalen van gegevens of de webshop simpelweg vloeiender maken. Dit kan allemaal worden aangegeven bij de Albert Heijn, aangezien ieder hier werkt.

Het doel van dit verslag is om te weten te komen hoe de functionaliteiten, gegevens en lay-out van de webshop van Albert Heijn tot stand zijn gekomen. De gegevens in dit verslag zijn verzameld door bronnen op het internet en door de powerpoints van professionele vaardigheden en requirements engineering te benutten.

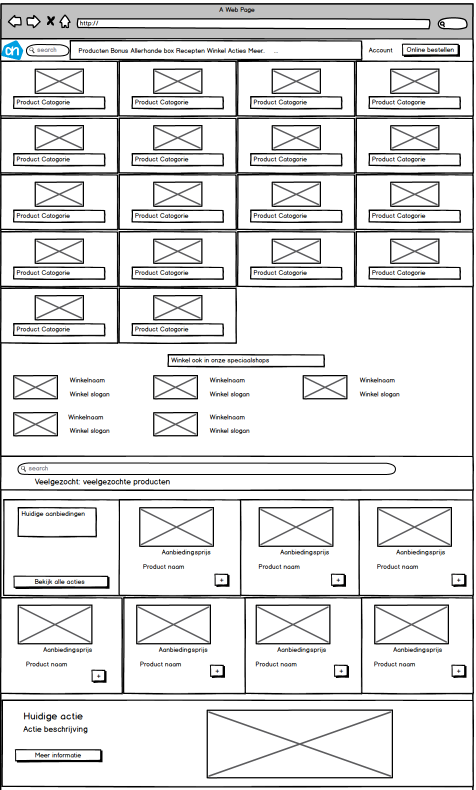
Om het doel van dit verslag te beantwoorden, is het verslag verdeelt in verschillende hoofdstukken. Eerst wordt er een korte beschrijving gegeven hoe de geschiedenis en achtergrond van Albert Heijn eruit heeft gezien. Dit wordt beschreven in hoofdstuk 2. Daarna wordt er uitgelegd wat wireframes zijn en waarom ze gebruikt worden. Dit gebeurt in hoofdstuk 3. In hoofdstuk 4 wordt uitgelegd wat use cases use specificaties zijn. Daarna wordt er uitgelegd wat een ERD is en wordt er toegelicht hoe het ERD van Albert Heijn er volgens ons uitziet. Dit gebeurt in hoofdstuk 5. In hoofdstuk 6 wordt de consistentie tussen de onderdelen vastgelegd. Hoofdstuk 7 geeft een korte samenvatting en toelichting over het verslag. Aan het verslag is ook een literatuurlijst toegevoegd. En ten slotte zijn er bijlage toegevoegd aan het verslag. Deze bijlage zijn alle wireframes en use case specificaties die door ons gemaakt zijn.

# 2. Geschiedenis en achtergrond van Albert Heijn

Albert Heijn(1865 - 1945) nam op 27 mei 1887 de kruidenierswinkel van zijn vader Jan Heijn in de Kerkbuurt van Oostzaan over. De winkel werd een groot succes waardoor meneer Heijn later een filiaal kon openen in Purmerend (Jonker, 2013). Een paar eeuwen later bezit Albert Heijn een Nederlands marktaandeel in de sector supermarkten van maar liefst 35%(Distrifood, 2016). Ook bevindt Albert Heijn zich nu onder Ahold Delhaize, een fusie van Ahold en Delhaize vond plaats in 2016. Een grotere droom zou meneer Albert Heijn zich niet kunnen voorstellen.  
 Als een grote winkel is het tegenwoordig bijna een must om een webshop te hebben, veel klanten hebben niet de mogelijkheid om constant langs te gaan bij een winkel. Vaak zijn producten online ook goedkoper dan in de winkel, omdat er minder kosten worden gemaakt bij een webshop. Een winkelpand inclusief gas, water en elektra kost natuurlijk veel om te onderhouden, daarnaast is er ook nog personeel dat rondloopt in een winkel. Meestal heeft een webshop alleen een warenhuis en/of een distributiecentrum, hier kan vaak ook nog eens meer worden opgeslagen dan in een pand voor een winkel. Kortom, de kosten van een webshop zijn vaak lager dan die van een winkel met meerdere panden. Het concept van ‘voor een bepaalde tijd besteld, morgen in huis’, verleidt klanten alleen maar meer om hun producten voortaan toch maar online te bestellen.   
 Lang beweerde men dat het onmogelijk was om supermarktproducten online te kunnen bestellen. Het argument dat vaak aan bod kwam is dat veel producten uit de supermarkt gekoeld bewaard moeten worden, denk hierbij aan dingen zoals: vleeswaren, zuivel, kaas, ijsjes en diepvriesmaaltijden. Maar ook producten zoals: brood, groente en fruit zijn niet lang houdbaar. De groei van de transportsector en de uitbreiding van grote namen zoals Jumbo en Ahold, maakt het mogelijk om te investeren in een nieuw concept, namelijk: een online supermarkt.

Online supermarkten zijn al een tijdje in de omloop. Albert Heijn kwam als eerste met albert.nl en was ook een lange tijd de enige supermarkt waar je online boodschappen kon doen. Dit is te danken aan de grootte van Ahold, Ahold heeft in 2000 het online boodschappenbedrijf Peapod gekocht in de Verenigde Staten. Hieruit heeft Ahold veel ervaring op kunnen doen, in het jaar 2000 lag de focus van supermarkten namelijk vooral op fysieke winkels. Pas in 2015 kwam Jumbo met de mogelijkheid om boodschappen online te doen, een andere koploper genaamd Picnic, meldde zich dat zelfde jaar nog als online boodschappenbedrijf.(Meinema, 2017)  
 Albert Heijn is op het moment de grootste in online boodschappen. Dit heeft Ahold wel vijftien jaar gekost, momenteel wordt er naar schatting een omzet van 400 miljoen euro bereikt. Jumbo zit in nog geen twee jaar op een omzet van 125 miljoen, maar heeft wel de ambitie om binnen vijf jaar marktleider te zijn. Picnic draaide in het eerste jaar een omzet van vijftien miljoen euro, maar inmiddels is dit gestegen naar 30 miljoen in alleen al het eerste kwartaal van 2017. Volgens verwachting zal het aan het eind van 2017 ongeveer oplopen tot 100 miljoen euro.

# 3. Wat zijn wireframes?

****Een wireframe is een visueel hulpmiddel bij het ontwikkelen van een website of applicatie. Wireframes kunnen gezien worden als een schets of bouwtekening van een website. In een wireframe worden zaken vastgelegd als navigatie, indeling en inhoud (Maltha, 2009). Wireframes focussen zich op de inhoud en dus niet op de grafische aspecten. Het maakt dus niet uit of een wireframe ‘mooi’ is of niet.

Figuur 1: Deel van wireframe homepage AH 1

## 3.1 Wanneer worden er wireframes gebruikt?

Wireframes kunnen gebruikt worden om ideeën een visueel beeld te geven. Dit visuele beeld kan inzicht geven aan de ontwerper van het product of aan de klant die het product wilt ontwikkelen.

Ook kan een ontwikkelaar een wireframe aan de klant laten zien hoe hij denkt dat het product eruit moet komen te zien. Doordat de klant inziet hoe de ontwikkelaar het product in zijn hoofd heeft, kan de klant zeggen of dit is wat hij in zijn gedachten had.

## 3.2 Wireframes van AH.

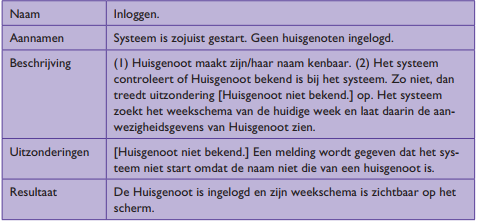
Van de website van AH zijn wireframes gemaakt. In figuur 1 is te zien hoe een wireframe van de hoofdpagina van de AH webshop eruit ziet.

De webshop van AH verandert vaak. Dit komt omdat de bonusproducten en aanbiedingen om de week veranderen. Ook verandert de webshop bij bepaalde feestdagen. Bijvoorbeeld bij kerstmis.

Hierdoor is het lastig om wireframes van de webshop te maken. Dit is opgelost door te kiezen voor algemene termen in de wireframes. Zo is in figuur 1 te zien dat de aanbiedingen geen naam hebben, maar een algemene naam.

4. Wat zijn use cases?  
Een use case kan gezien worden als een grafische scope van een systeem. Een systeem kan hier in principe van alles zijn, een automaat om een treinkaartje te kopen, de webshop van een website of een geldautomaat. Bij een use case wordt er duidelijk gemaakt wat de relaties van actoren zijn met de use cases. Bij actoren spreekt men vaak over de gebruikers zoals een klant, beheerder of medewerker. Er zijn verschillende manieren om een use case te maken, zo is het onder andere mogelijk om de specificaties van een use case te noteren in een tabel. In dat geval is het belangrijk dat er concrete informatie genoteerd wordt. Er moet in duidelijke stappen worden uitgelegd hoe de gebruiker en het systeem met elkaar samenwerken.

In *figuur 2* wordt zo kort duidelijk gemaakt hoe een use case in elkaar zit. De tabel gaat over een gebruiker die dient in te loggen op een systeem. Er zijn diverse onderdelen genoteerd die belangrijk zijn als er wordt gekeken naar een use case. Allereerst heeft de tabel een naam: inloggen. Het is belangrijk dat er een naam vermeld wordt bij een use case.

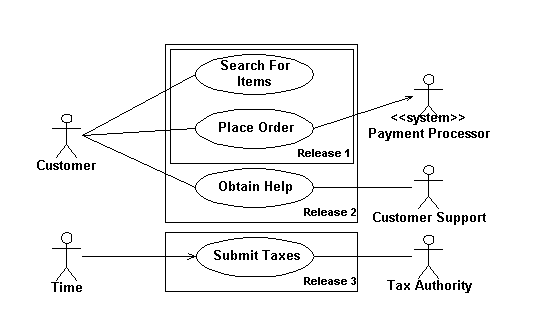


Figuur 2: Use Case Tabel 1

* Een naam zorgt voor een overzicht, men voegt hier vaak ook nog een cijfer aan toe. Het kan namelijk zo zijn dat er meerdere use cases zijn die over hetzelfde onderwerp gaan, maar net een ander scenario behandelen.
* Als tweede punt zien we de aannamen staan, hier staan de eisen in, er wordt verwacht dat de gebruiker deze stappen al heeft gemaakt. Als deze stappen niet zijn gemaakt, kan het scenario niet plaatsvinden.
* Hierna volgt een korte beschrijving, dit wordt ook wel een hoofdscenario genoemd. In het hoofdscenario worden de stappen van de gebruiker en de stappen van het systeem een voor een uitgelegd.
* Het onderdeel uitzonderingen gaat over situaties waar de use case niet mogelijk is. Als er geen juiste inlog gegevens worden ingevuld, kan het systeem niet werken.
* Aan het eind staat meestal het resultaat. Dit is de verwachte resultaat van een scenario, het is de bedoeling dat het systeem en de gebruiker samen voor dit resultaat zorgen.

Het is verstandig dat een use case wordt gevisualiseerd, het is namelijk nog altijd het belangrijkste element in een model van een geheel systeem. Een visualisatie hiervan wordt een use case diagram genoemd. Dit kan worden gezien als het onderdeel dat de requirements verbindt met elkaar in een systeem. Het is dan ook belangrijk dat een use case diagram niet door ontwikkelaars zelf wordt opgezet, maar door de belanghebbenden(Ambler, 2014).

Bij een use case wordt gekeken hoe een gebruiker met het systeem om kan gaan. Deze interacties worden zo concreet mogelijk vastgesteld, zodat het duidelijk si hoe de interacties tussen een gebruiker en het systeem zal verlopen. Een goed voorbeeld is een use case waar een gebruiker dient in te loggen. Voordat een inlogscherm ontwikkeld wordt is het verstandig om te kijken hoe een gebruiker het inlogscherm zal gebruiken. Het is mogelijk om dit te verwerken tot een diagram. Er komen dan wel meerdere UML termen bij kijken, maar daar wordt in dit verslag niet verder op ingegaan.

In *figuur 3* is een voorbeeld te zien. Het gaat hier om een use case waarbij het systeem moet helpen om te zoeken naar bepaalde functies, het plaatsen van orders, hulp kunnen doorgeven en belastingverzoeken kunnen doorvoeren.  


Figuur 3: Use Case Diagram 1

* Het diagram bestaat uit vier actoren: Customer, Time, Payment Processor, Customer Support en Tax Authority. De vier actoren hebben elk een interactie die samen met het systeem uitgevoerd moeten worden.
* In het diagram zijn vier use cases te vinden: ‘Search For Items’, ‘Place Order’, ‘Obtain Help’ en ‘Submit Taxes’. Er zijn drie scopes te vinden in *figuur 3*, maar om het kort en simpel te houden, is het nog niet belangrijk dat er naar gekeken wordt. In het diagram houdt elke scope een release in, meestal wordt dit in de praktijk niet zo vaak gebruikt en komt er maar één scope voor.
* Er zijn ook pijlen te zien in het diagram, pijlen kunnen verschillende functies hebben. De pijlen worden ook associations genoemd.

4.1 AH & Use Cases  
Ook bij de Albert Heijn is er sprake van use cases. Albert Heijn heeft een webshop, waar verschillende gebruikers gebruik willen maken van de webshop. Het is belangrijk dat het (sub)systeem van bepaalde functies op een correcte manier de aanvragen van users kan verwerken.

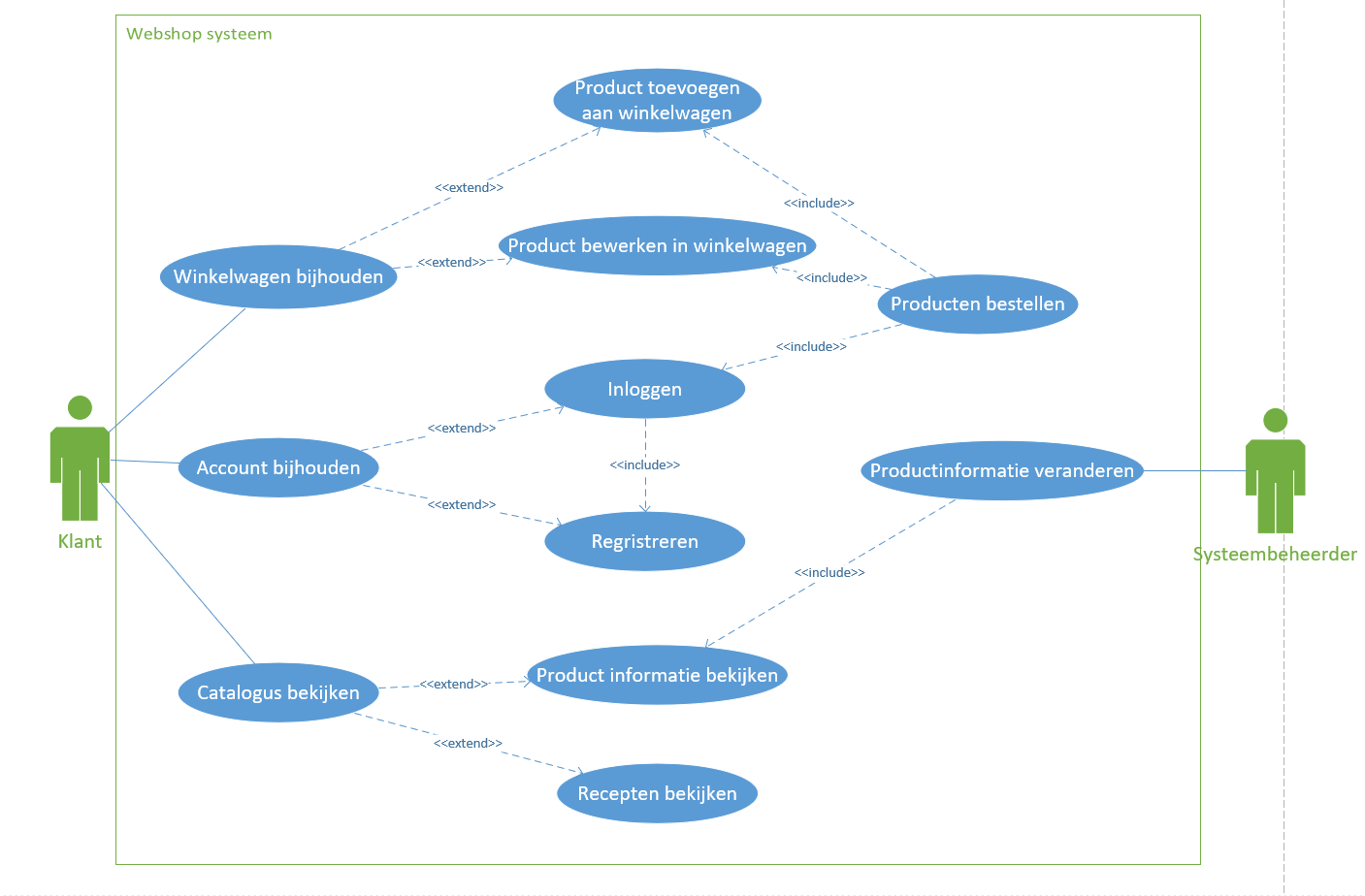
In *figuur 4* is een use case met use case specificaties te zien. Hier zijn wat meer gegevens af te lezen dan in de voorbeelden die eerder gegeven zijn. Dit keer gaat de use case wel over de webshop van Albert Heijn, het gaat namelijk over het registreren van de gegevens van een gebruiker, het systeem moet ervoor zorgen dat al de acties van de Klant goed worden doorgevoerd. Uiteindelijk moet het systeem namelijk slagen in het aanmaken van een nieuw account op basis van de ingevoerde gegevens. Ook zijn er nu alternatieve scenario’s te zien, een klant hoeft niet altijd via de opgenoemde scenario het resultaat te behalen. Het is bijvoorbeeld ook mogelijk om telefonisch een account aan te maken, met behulp van de klantenservice van Albert Heijn.



Figuur 4: Use case specificatie AH 1

4.2 AH & Use Case Diagram  
Zelfs van de Albert Heijn webshop kan een use case diagram worden gemaakt. Zoals eerder werd vermeld is het belangrijk dat er concreet wordt vastgelegd wat de gebruikers van het systeem terug moeten krijgen als zij bepaalde handelingen verrichten. Een systeem van een webshop is mogelijk om te onderverdelen in kleinere subsystemen, maar dat zorgt er ook voor dat er veel werk nodig gaat zijn, daarnaast moet er dan extra goed geanalyseerd worden hoe de vele subsystemen werken. In *Use Case Diagram 2* is een goed voorbeeld te zien van een webshop systeem met een algemene scope.  
 In *figuur 5* zijn er elf use cases te zien. Bepaalde gebruikers moeten dit soort scenario’s kunnen uitvoeren via het systeem. In *Use Case Diagram 2* is klant dit keer niet de enige actor, er is nu ook een systeembeheerder. Zoals staat vermeld, moet de systeembeheerder de informatie van producten kunnen vervangen. Een belangrijk onderdeel, veel webshops moeten vaak hun informatie of prijzen aanpassen. Met name de supermarkten, waar in het geval van Albert Heijn vaak aanbiedingen zijn, is het enorm van belang dat dit op een juiste manier aangepast kan worden. Ook staat er nu nieuwe informatie vermeld in de scope:

* Extends: betekent kort gezegd dat een use case een uitbreiding is van een andere use case, een main use case (waar de pijl vandaan komt), kan weer dieper op ingegaan worden. In het voorbeeld van ‘Winkelwagen Bijhouden’ bestaat de use case weer uit ‘Product bewerken in winkelwagen’ en ‘Product toevoegen aan winkelwagen’. Een uitbreiding betekent echter niet dat de main use case een los onderdeel is. ‘Winkelwagen bijhouden’ kan niet slagen als de twee extend use cases ook niet kunnen slagen.
* Includes: lijkt een beetje op de extend. Het is alleen wel anders, in het geval van de use case ‘Producten bestellen’ is een include pijl te zien die naar twee use cases loopt. De include pijl betekent dat de main use case afhankelijk is van een andere use case. Een klant kan geen producten bestellen als die niet is ingelogd, ook moet de klant bij zijn/haar winkelwagen kunnen tijdens het bestel-onderdeel.



Figuur 5: Use Case Diagram 2

# 5. Wat is een ERD?

Een ERD (Enitity Relationship Diagram) is een soort stroomdiagram dat illustreert hoe entiteiten zoals mensen voorwerpen of concepten binnen een systeem met elkaar verbonden zijn. ERD diagrammen worden vaak gebruikt om relationele databases te ontwerpen(Lucidchart Software, z.j). Bij het ontwerpen van ERD diagrammen worden bepaalde symbolen gebruikt. Elk symbool heeft een betekenis. Zo is bijvoorbeeld een rechthoek een entiteit en een geeft een ruit een relatie tussen entiteiten weer.

## 5.1 Het ERD van de webwinkel van Albert Heijn.

Figuur 6 : Het ERD van AH

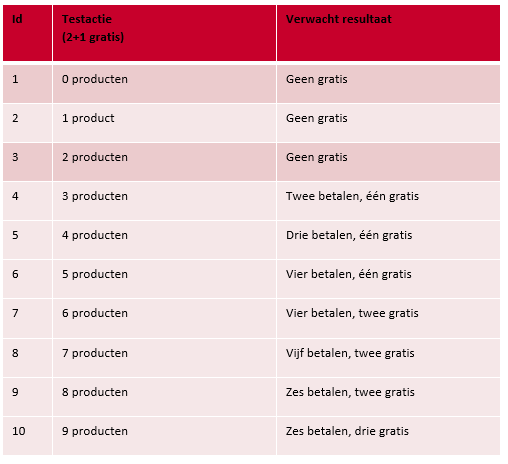
Voor de webwinkel van Albert Heijn is er een ERD gemaakt(Figuur 5). Het ERD heeft de entiteiten account, winkelmand, product, recept en bonusproduct.

Een account heeft een gebruikersnaam en andere gegevens zoals een postcode etc. Een account kan een winkelmand hebben, dit hoeft dus niet per se. Er wordt een winkelmand aangemaakt als een account een product toevoegt aan zijn of haar winkelmand. Een winkelmand bestaat uit producten. De producten hebben een prijs. Ook kan er in de winkelmand een aantal van het gekozen product gekozen worden.

Een product heeft een naam en categorie en prijs. Ook word er aangegeven of het product wel of niet op voorraad is. Een product kan ook een bonusproduct zijn. In dat geval heef teen bonusproduct een nieuwe prijs en heeft het een datum tot wanneer het bonusproduct in de bonus is. Een product kan ook een onderdeel zijn van een recept. Een recept bevat producten, de prijs en het type recept. Bijvoorbeeld dessert, tussendoortje etc.

# 6. Consistentie

Bij de use cases is het van belang dat de use cases foutloos uitgevoerd kunnen worden door het systeem. Het systeem moet kunnen registreren of een gebruiker de juiste of onjuiste gegevens invoert. In het geval de gebruiker een use case doorgeeft aan het systeem in het geval van bijvoorbeeld een korting berekenen, is het van belang dat de juist getallen getest worden voordat het product wordt afgerond.   
 Om het voorbeeld over de korting op een duidelijke manier uit te leggen, er zijn veel winkels die een korting meegeven bij een product wat vaker dan één keer gekocht kan worden door een klant. Denk hierbij aan dingen zoals: bioscoopkaarten, producten bij een supermarkt of videogames bij de Intertoys. Gebruikers kunnen korting krijgen als zij een minimum en maximum aantal producten kopen. Bij bioscoopkaarten wordt er meestal geen korting gegeven als er maar één kaart wordt gekocht, in het geval er tien of elf kaarten worden gekocht, is het bij de meeste bioscopen wel mogelijk om korting te ontvangen. Maar mocht de groep oplopen tot een grootte van 30 of zelfs 50 man, dan kan het de bioscoop juist geld kosten om zoveel korting te geven.  
 Al deze informatie heeft uiteindelijk met de consistentie van een use case te maken. Om een use case consistent te noemen, is het belangrijk dat zoals in het voorbeeld van de bioscoopkaarten, er goed wordt getest of het systeem de juiste handeling verricht in het geval een gebruiker het system gebruikt. Deze controles worden ook wel testen genoemd, waar grenswaardenanalyses plaatsvinden. Vooral grenswaardenanalyses zijn heel belangrijk bij webshops, in het geval er korting wordt meegegeven, is het wel bedoeling dat dit op de juiste manier gaat.

In het geval van de Albert Heijn webshop, is er ook sprake van consistentie. Albert Heijn heeft vaak acties die grenswaardes hebben. Zo is er soms een ‘2+1 gratis actie’ hierbij ontvangt de klant één gratis product als de klant twee van dezelfde producten koopt. Hier is wel van belang dat de klant drie producten afrekent, waarvan er dan uiteindelijk maar twee betaald worden. In figuur *figuur 7* wordt dit behandeld.  
In de tabel zijn de volgende drie kolommen af te lezen: een ID, Test actie en een verwacht resultaat.

Figuur 7: Use case test 1

* Een ID is een simpele teller die de verschillende test acties telt. Zo is er een overzicht, waardoor het duidelijk is wat er getest wordt.
* Een testactie houdt de inhoud in van de test, in het geval van de figuur, is er sprake van een grenswaarden die getest wordt.
* Als laatste kolom is er een verwacht resultaat. Dit spreekt voor zich en houdt het resultaat in dat er verwacht wordt.

# 7. Korte samenvatting en toelichting

Albert Heijn begon als een kleine winkel in 1887. Nu is Albert Heijn een van de grootste supermarkten in Nederland en andere landen. In dit verslag is onderzocht hoe het ontwikkelproces van de webshop van AH eruit heeft gezien. Dit is gedaan door het begrip reverse engineering toe te passen.

Van de webshop zijn wireframes gemaakt. Dit is gedaan om de functionaliteiten en de inhoud van de webshop duidelijk te laten zien, zonder de opmaak en kleuren. De wireframes in dit verslag zijn een schets van hoe volgens ons de eerste schetsen van de webwinkel er uit hebben gezien.

Ook zijn er use cases en use specificaties gemaakt. Dit is gedaan om de requirements te bepalen van zowel de klant en de beheerder van de website. In het verslag zijn in totaal elf use cases en zes use case scenario’s uitgewerkt.

Er is ook een ERD verwerkt in dit verslag. Het ERD geeft weer hoe wij denken dat de database achter de webshop van Albert Heijn eruit ziet. Dit kan echter op verschillende manieren gebeuren dus het is dus niet zeker of de echte database van Albert Heijn er ook zo uitziet.

De consistentie tussen de onderdelen is bepaald door middel van testen en een uitleg. De testen zijn gemaakt met een begin waarde en een uitkomst.

Ten slotte zijn er bijlage aan het verslag toegevoegd. Dit zijn alle wireframes en use specificaties die wij hebben gemaakt.

Met het maken van wireframes, use cases en een ERD hebben wij onze doelstelling gehaald. Het is namelijk duidelijk geworden hoe het reverse engineering proces eruit heeft gezien.

# Literatuurlijst

Maltha, H. (2009, 23 november). Wireframes, de bouwtekening van een website.. Geraadpleegd op 4 januari 2018, van https://www.frankwatching.com/archive/2009/11/23/wireframes-de-bouwtekening-van-een-website/

Ambler. (z.j.). UML 2: Use case diagrams. Geraadpleegd op 5 januari 2018, van http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

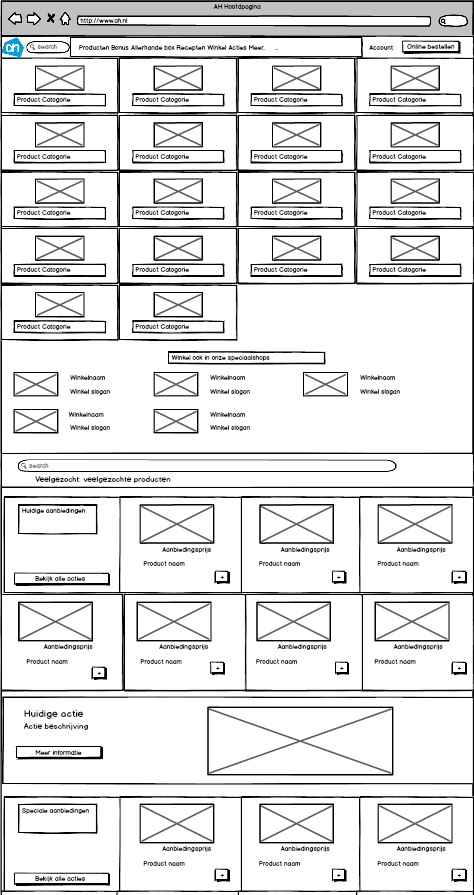
Lucid Software. (z.j.). Wat is een erd? Geraadpleegd op 5 januari 2018, van https://www.lucidchart.com/pages/nl/wat-is-een-entity-relationship-diagram

Lucid Software. (z.j.). Use case diagram tutorial. Geraadpleegd op 5 januari 2018, van https://www.lucidchart.com/pages/uml/use-case-diagram

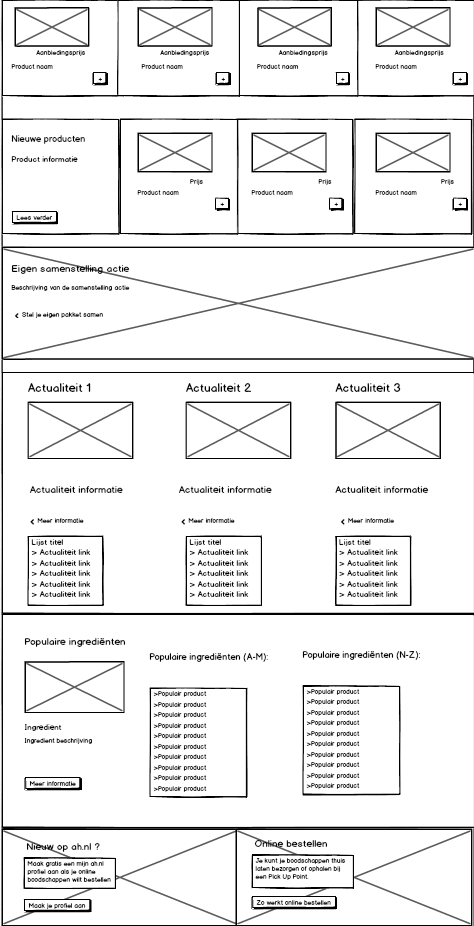
SmartDraw. (z.j.). Entity Relationship Diagram. Geraadpleegd op 5 januari 2018, van https://www.smartdraw.com/entity-relationship-diagram/

Van der Velden, K. (2014, 7 mei). Het voordeel van wireframes. Geraadpleegd op 4 januari 2018, van https://www.creativeict.nl/het-voordeel-van-wireframe

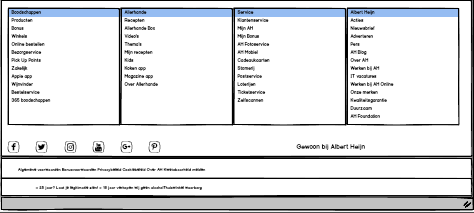
# Bijlage 1 : De wireframes



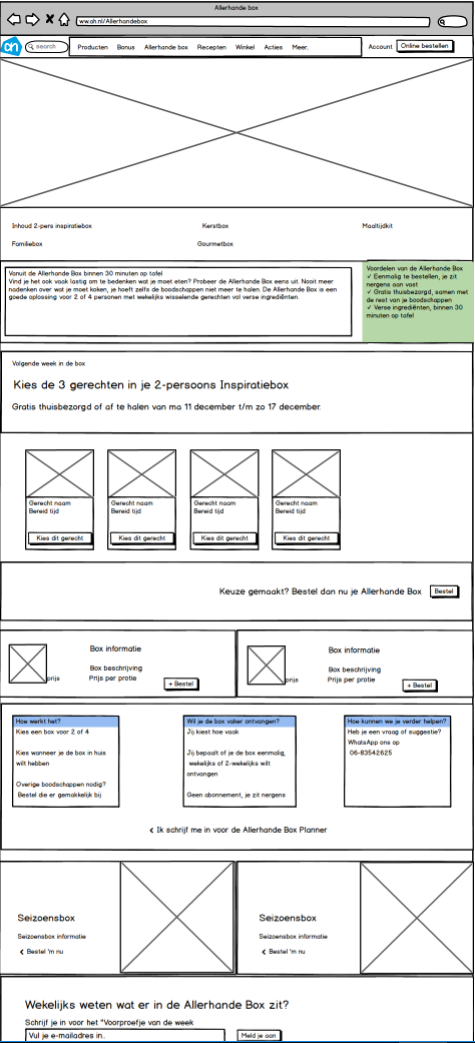
AH Hoofdpagina deel 1/3



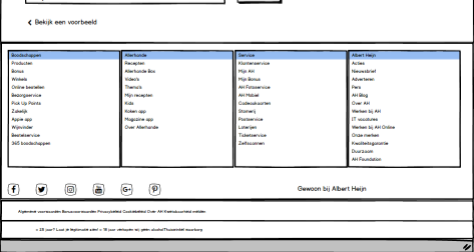
AH Hoofdpagina deel 2/3



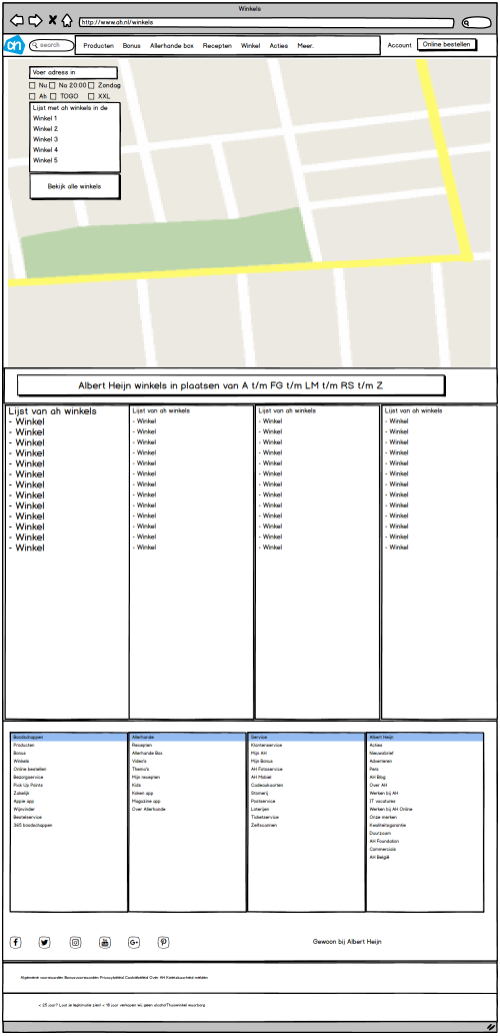
AH Hoofdpagina deel 3/3



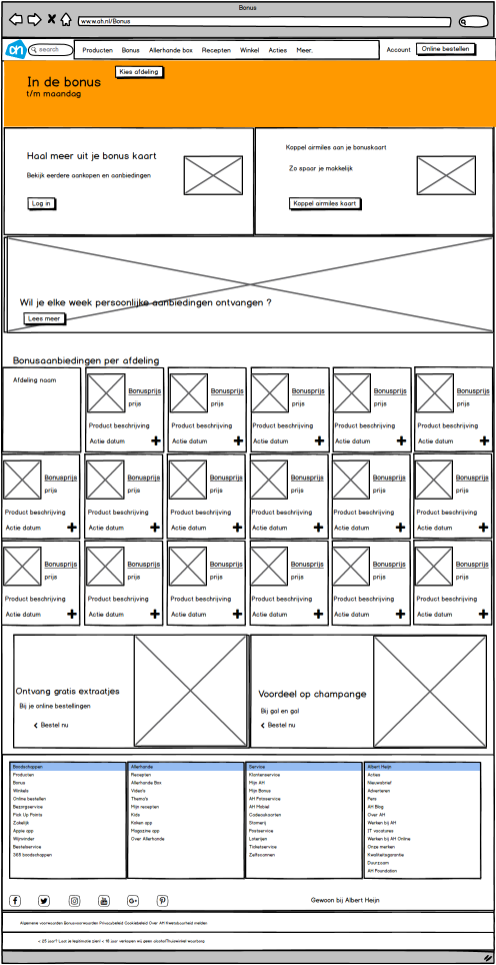
AH Allerhande box deel 1/2



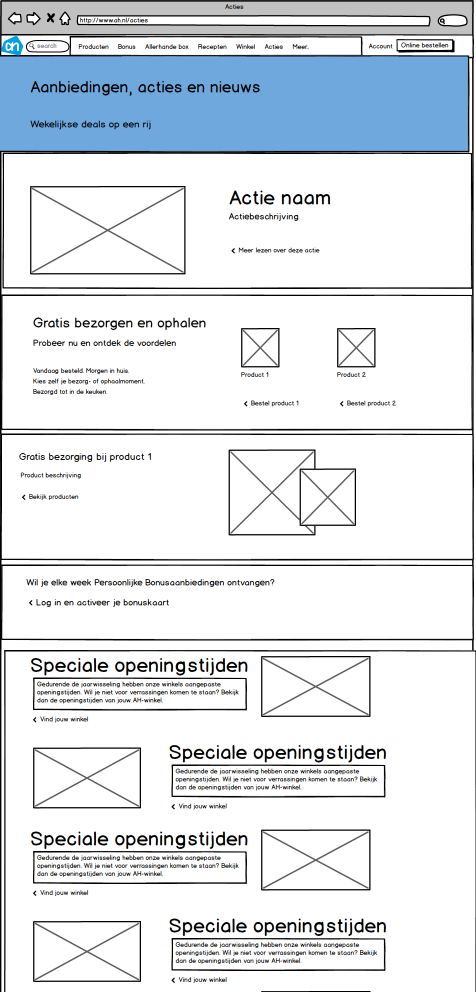
AH Allerhande box deel 2/2



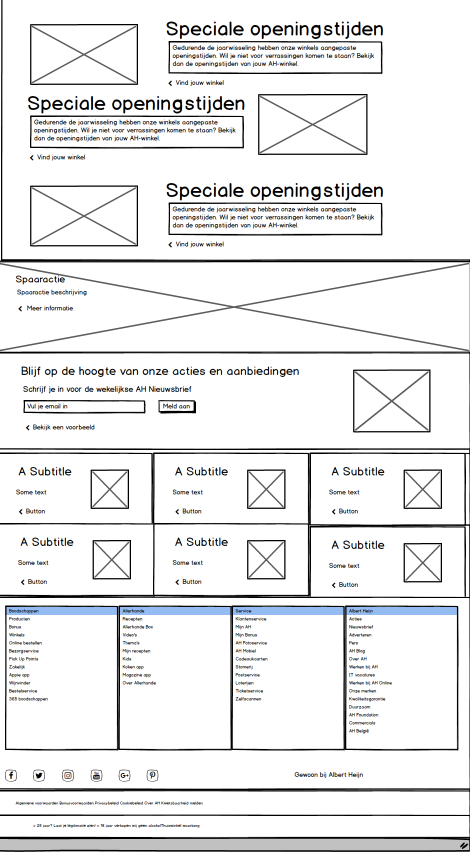
AH Winkels deel 1/1



AH Bonus deel 1/1



AH Acties deel 1/2



AH Acties deel 2/2

# Bijlage 2 : Alle use case specificaties

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | Klant registratie |
| Nummer use case | Registrering1 |
| Naam actor | Klant |
| Precondities | Systeem wacht tot de gebruiker zich op de pagina van registratie bevind |
| Hoofdscenario | 1. Klant gaat naar de homepagina van AH 2. Systeem toont de hoofdpagina van AH 3. Klant klikt op het icoontje van accounts 4. Het systeem toont een venster met een optie om een account te maken of in te loggen 5. Klant klikt op een nieuw account aanmaken 6. Systeem toont een nieuw venster waar gegevens ingevuld kunnen worden 7. Klant voert de benodigde gegevens in 8. Klant klikt op account aanmaken 9. Systeem controller of de gegevens kloppen 10. Systeem registreert gegevens in het systeem 11. Systeem stuurt een welkom mail 12. Klant logt in met inloggegevens 13. Systeem registreert het account als aangemaakt |
| Alternatief scenario 1 | Account wordt telefonisch geregistreerd door een medewerker van  bijvoorbeeld de klantenservice |
| Alternatief scenario 2 | Klant had al een account in het oude systeem waar alleen een bonuskaart  nodig was |
| Fout scenario 1 | E-mailadres komt al voor in de database |
| Resultaat | Klant heeft een account aangemaakt. |
| Postconditie | Systeem wacht tot de gebruiker zich op de pagina van registratie bevind |

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | Recept opzoeken |
| Nummer use case | Recept1 |
| Naam actor | Klant |
| Precondities | Klant bevindt zich op de recepten pagina |
| Hoofdscenario | 1. Klant vult de naam van het recept of een van de benodigde ingrediënten in 2. Systeem weergeeft meerdere suggesties aan de hand van de ingevulde informatie 3. Systeem weergeeft meerdere filtermogelijkheden 4. Klant filtert bepaalde recepten weg 5. Systeem weergeeft specifiekere recepten 6. Klant zoekt op de pagina naar het recept 7. Klant selecteert het gewenste recept 8. Systeem toont het recept |
| Alternatief scenario 1 | Klant vindt het recept direct na het invullen van de zoekbalk |
| Alternatief scenario 2 | Klant heeft de link van het gewenste recept al in de mail ontvangen |
| Fout scenario 1 | Het gewenste recept komt niet voor in de database van AH |
| Resultaat | Klant vindt het juiste recept |
| Postconditie | Klant bevindt zich op de recepten pagina |

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | Bonus product bekijken |
| Nummer use case | Bonus1 |
| Naam actor | Klant |
| Precondities | Klant bevindt zich op de bonus pagina |
| Hoofdscenario | 1. Klant klikt op de gewenste categorie op de bonus pagina 2. Systeem weergeeft een pagina van de aangeklikte categorie 3. Klant klikt op een specifiekere filter 4. Systeem filtert de niet aangevinkte voorkeuren weg 5. Systeem weergeeft een specifieker aanbod 6. Klant selecteert het gewenste product 7. Systeem weergeeft het gewenste product |
| Alternatief scenario 1 | De klant was al op de hoogte van de gewenste bonus artikelen via de post |
| Alternatief scenario 2 | De klant was al op de hoogte van de gewenste bonus artikelen via de mail |
| Fout scenario 1 | Het gewenste product is niet in de bonus |
| Resultaat | Klant vindt het gewenste bonus product |
| Postconditie | Klant bevindt zich op de bonus pagina |

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | E-mailadres wijzigen |
| Nummer use case | Account1 |
| Naam actor | Klant |
| Precondities | Klant bevindt zich op de dashboard pagina |
| Hoofdscenario | 1. Klant klikt op ‘wijzigen’ onder ‘mijn gegevens’ 2. Systeem weergeeft de klant zijn/haar huidig geregistreerde  gegevens 3. Klant klikt op de balk van e-mailadres 4. Klant vult het nieuwe e-mailadres in 5. Klant vult het wachtwoord in ter bevestiging 6. Klant klikt op ‘opslaan’ 7. Systeem weergeeft de dashboard pagina |
| Alternatief scenario 1 | Klant wijzigt het mailadres via de telefoon |
| Alternatief scenario 2 | Klant maakt een nieuw account aan met het gewenste mailadres |
| Fout scenario 1 | Klant vult een onjuist wachtwoord in |
| Resultaat | E-mailadres gewijzigd |
| Postconditie | Klant bevindt zich op de dashboard pagina |

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | Account verwijderen |
| Nummer use case | Account2 |
| Naam actor | Klant |
| Precondities | Klant bevindt zich op de dashboard pagina |
| Hoofdscenario | 1. Klant klikt op ‘wijzigen’ onder ‘mijn gegevens’ 2. Systeem toont de klant zijn/haar huidig geregistreerde  gegevens 3. Klant scroll naar beneden 4. Klant klikt op ‘Mijn ah.nl profiel verwijderen’ 5. Systeem toont een bevestiging scherm 6. Klant klikt op ‘doorgaan’ 7. Systeem toon teen scherm waar het wachtwoord van het account ingevuld moet worden 8. Klant voert wachtwoord in 9. Klant drukt op de knop met ‘Mijn ah.nl profiel verwijderen’ 10. Systeem toont een scherm met een bevestiging dat het  account is verwijderd 11. Klant klikt op de knop ‘verder’ 12. Systeem toont de homepagina AH |
| Alternatief scenario 1 | Account verwijderen via de telefonische klantenservice |
| Alternatief scenario 2 | Klant benaderd de klantenservice via mail om zijn/haar account uit te schakelen |
| Fout scenario 1 | Klant vult een onjuist wachtwoord in bij bevestiging |
| Resultaat | Account is verwijderd |
| Postconditie | Klant bevind zich op de homepagina van AH |

|  |  |
| --- | --- |
| **Item** | **Waarde** |
| Naam use case | Adres wijzigen |
| Nummer use case | Account3 |
| Naam actor | Klant |
| Precondities | Klant bevindt zich op de dashboard pagina |
| Hoofdscenario | 1. Klant klikt op ‘wijzigen’ onder ‘mijn gegevens’ 2. Systeem weergeeft de klant zijn/haar huidig geregistreerde  gegevens 3. Klant scrollt naar beneden 4. Klant klikt op het eerste vakje van ‘postcode, huisnummer en  toevoeging’ 5. Systeem vult automatisch de straatnaam en plaats in, dat wordt bepaald aan de hand van de ingevulde gegevens 6. Klant scrollt meer naar beneden 7. Klant vult het juiste wachtwoord in 8. Klant drukt op ‘opslaan’ 9. Systeem weergeeft de dashboard pagina |
| Alternatief scenario 1 | Adresgegevens worden gewijzigd door het contact nemen met  de klantenservice via de mail |
| Alternatief scenario 2 | - |
| Fout scenario 1 | Geen geldig adres ingevoerd |
| Resultaat | Adres gegevens gewijzigd |
| Postconditie | Klant bevindt zich op de dashboard pagina |

# Bijlage 3 : Use case diagram

