

Analysedocument

Live performance

Michelle Broens | S21M | 08/05/2017

Inhoudsopgave

[Begrippen 2](#_Toc472607809)

[Inleiding 3](#_Toc472607810)

[Opdrachtgever 4](#_Toc472607811)

[Scenario 5](#_Toc472607812)

[Functionele eisen 6](#_Toc472607813)

[Niet-functionele eisen 8](#_Toc472607814)

[Use case 9](#_Toc472607815)

[Use case diagram 9](#_Toc472607816)

[Use case beschrijvingen 10](#_Toc472607817)

[User interface 15](#_Toc472607818)

[Bijlage GUI 18](#_Toc472607819)

[EER model 14](#_Toc472607820)

[EER bijlage 15](#_Toc472607821)

# Begrippen

|  |  |
| --- | --- |
| **Begrip** | **Uitleg** |
| Use case | In software engineering is een use case een lijst van acties die een persoon (actor) en het systeem uitvoert om een doel te bereiken, bijvoorbeeld het inloggen. |
| Userinterface | Hier praten de gebruiker en het systeem met elkaar. |
| Database | Een georganiseerde collectie van data in tabellen. |
| AFK | Een afkorting voor het woord afkorting. |
| MoSCoW | Een methode om aan te geven hoe belangrijk een requirement is, met de M voor must(moet), S voor shoulds(zou moeten), C voor could(zou kunnen) en W voor will not(wordt niet geïmplmenteerd). |
| Admin | De persoon in het systeem met de macht om bijvoorbeeld pizza’s uit de database te verwijderen. |
| Record | Dit is een rij uit een tabel van de database |
| Datagridview | Is een windows forms control die hier de tabellen uit de database laten zien |
| Primary key | Dit is een speciale kolom uit de database aangewezen om alle records uniek te identificeren |
| Foreign key | Dit is een externe kolom die verwijs naar de primary key in de originele tabel |

# Inleiding

In dit document komen er een aantal verschillende onderwerpen aanbod die te maken hebben met de analysefasen van het maken van een applicatie over … De onderwerpen die aan bod komen zijn:

* Een lijst met begrippen: in deze lijst staan woorden die de lezer misschien niet begrijpt of misschien niet helemaal begrijpt.
* Opdrachtgever: in dit stuk kom een beschrijving over wie de opdrachtgever is en wat de opdrachtgever algemeen verwacht.
* Het scenario: een korte beschrijving over een persoon die door de ontwikkelde applicatie loopt.
* De functionele eisen: de eisen die het programma moet kunnen, bijvoorbeeld een gebruiker kan inloggen.
* De niet-functionele eisen: de eisen die niet te maken hebben met het programmeren van code zoals de functionele eisen, bijvoorbeeld de code moet herbruikbaar zijn.
* De use cases: de functionele eisen worden in een diagram gezet zodat het duidelijk wordt door wie deze functie uitgevoerd wordt. Deze functies worden in de use case beschrijving van start tot eind doorgelopen.
* De userinterface: hier wordt een schets gemaakt van hoe de applicatie er ongeveer uit gaat zien.
* Het EER model: een schets van hoe de structuur van het programma er in de database uit gaat zien.
* De productdecompositie: een schets van de verschillende onderdelen die opgeleverd worden.
* Planning: de planning van hoe dit project uitgewerkt wordt. Per onderdeel staat een tijd van wanneer er een onderdeel begonnen wordt en wanneer deze af is.

# Opdrachtgever

De opdrachtgever van dit project is pizzeria La Crosta Insapore en zij willen een applicatie die voor hun het plaatsen van bestellingen afhandelt. Het op te leveren systeem bestaat uit een software applicatie en een database. De opdrachtgever heeft geen voorkeur uitgesproken ten aanzien van het platform waar de applicatie op draait.

De opdrachtgever wil een verschil tussen een klant en een medewerken (gebruiker en admin).

3 pizza vormen: rond, rechthoekig en driehoekig. 3 bodems: dun(krokant), dik (zacht) en glutenvrij. Een pizza kan een grootte hebben van: 50cm2 - 0.25m2. Er zijn ook salades en dranken beschikbaar. Er dient een kassabon gemaakt te worden voor elke bestelling met splitsing van BTW. Alleen alcohol is BTW 21%, de rest van de producten belasten 6% BTW.

BTW wordt pas op de kassabon erbij geteld, de bodem wordt berekend inclusief BTW.

De prijs van de bodems zijn:

* Dun = 0.80 cent per cm2.
* Dik = 1.20 cent per cm2.
* Glutenvrij = 2.00 cent per cm2.

Dee prijs van de pizza = bodem x ingrediënten.

Klanten geven de maat voor de pizza door. Per bodem soort geven ze deze maten door:

* Rond = diameter (d).
* Rechthoek = breedte (b) en lengte (l).
* Driehoek = (a, b, c).

Berekening voor de oppervlakte staat in de opdrachtomschrijving.

# Scenario

Marieke is een medewerker bij pizzeria La Crosta Insapore. Zij heeft een account waarmee ze in kan loggen op de applicatie. Als ze ingelogd heeft ziet ze het hoofdscherm op haar scherm heeft ze twee keuzes naar gebruikersmode of naar de database. De gebruikersmode geeft precies hetzelfde scherm als dat een normale gebruiker zou krijgen.

Ze kiest ervoor om naar de database te gaan. Dit is een apart scherm waar ze tabellen in de database kan aanpassen via de applicatie. Ze klikt dan op de knop toevoegen. Op het scherm wat ze ziet vult ze de velden in van het item dat ze in de database wil zetten. Als ze dit heeft gedaan klikt ze op de knop toevoegen. Het nieuwe item staat nu in de database. Ze komt er nu pas achter dat ze de naam verkeerd gespeld heeft. Ze klikt op de record in de datagridview en klikt vervolgens op aanpassen en ze wijzigt de naam. Ze krijgt na dit alles te horen dat het item niet toegevoegd hoefde te worden. Ze klikt nu op de knop verwijderen en verwijdert het item wat ze net heeft toegevoegd.

Pietje is een klat van pizzeria La Crosta Insapore. Hij ziet dat de pizza Margherita op nummer 1 staan en wil deze graag bestellen er staat ook nog dat deze veganistisch is, hij wil graag de grootte 50cm2 in een ronde vorm. Hij klikt op de pizza vanuit zijn hoofd scherm en klikt op volgende, hier krijgt hij nog twee schermen voor het bestellen van salade en drinken, maar hij wil alleen maar een pizza. Hij klikt op bestellen en vinkt “Mij onthouden” aan. Hij kiest dan de optie om de pizza zelf op te halen. Zijn bestelling is nu geplaatst in het systeem.

Marieke ziet Pietjes bestelling binnen komen en geeft deze door aan de koks in de pizzeria. Het is het einde van de dag dus ze gaat vanaf haar hoofdscherm weer naar het database scherm en kiest daar “Winst berekenen”. Deze optie berekent de winst die er per dag is gemaakt.

# Functionele eisen

Dit hoofdstuk gaat over de functionele requirements die geïmplementeerd gaan worden in de applicatie. Deze worden beschreven met een korte beschrijving, aangeduid hoe belangrijk deze requirement is met de MoSCoW analyse en een eventuele opmerking.

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| PR1.1 | Medewerkers kunnen ingrediënten**,** producten en standaardpizza’s toevoegen. | M |  |
| PR1.2 | Medewerkers kunnen ingrediënten, producten en standaardpizza’s verwijderen. | M |  |
| PR1.3 | Medewerkers kunnen ingrediënten, producten en standaardpizza’s aanpassen. | M |  |
| PR2 | Producten in de opdrachtomschrijving moet de klant kunnen bestellen. | M | Dit zijn o.a. de standaardpizza’s. |
| PR3 | Een klant kan een standaardpizza bestellen. | M |  |
| PR4 | Een klant kan zelf een pizza samenstellen. | M |  |
| PR5 | De bestelling wordt getoond na het plaatsen. | M |  |
| PR6 | Een bestelling bestaat uit een willekeurig aantal producten. | M | Een bestelling kan alleen uit een pizza bestaan, maar ook salade en drank met een pizza of alleen drank. |
| PR7 | De kassabon kan geëxporteerd worden naar een tekstbestand. | M | De template die aangehouden moet worden staat in de opdrachtomschrijving. |
| PR8 | De klant kan ervoor kiezen om zijn naam, adres, klantnummer en bestelhistorie op te slaan. | S | Dit wordt in de database opgeslagen |
| PR9 | Er moet een dagelijks overzicht komen van de omzet, met daarnaast de winst. | S | Winst = omzet - verkoopkosten |
| PR10 | De pizza’s mogen niet kleiner zijn dan 50cm2 en niet groter dan 2500cm2. | S |  |
| PR11 | De mogelijkheid om aan te geven dat een pizza veganistisch en/of halal is. | S |  |
| PR11 | Top 3 van meest verkochte pizza’s in een periode bijhouden. | C |  |
| PR12 | Aangeven of een bestelling bezorgd moet worden of deze opgehaald wordt. | C |  |
| PR13 | Pizza’s, ingrediënten en producten sorteren op naam of op prijs | C |  |

# Niet-functionele eisen

Deze requirements hebben niet te maken met het schrijven van code, maar die wel te maken hebben met de code.

|  |  |  |  |
| --- | --- | --- | --- |
| **AFK** | **Requirement** | **MoSCoW** | **Opmerking** |
| NF1 | De code moet herbruikbaar zijn. | M |  |
| NF2 | De prijs van een pizza wordt opgebouwd uit de prijs van de boden en de prijs van de ingrediënten op bases van de te beleggen oppervlakte. | M | Standaardpizza’s krijgen een korting van 25% op de berekende prijs. |
| NF3 | Er zit 6% BTW op de bestelde producten | M | Tenzij dit alcohol is dan zijn deze belast met 21%. |
| NF4 | In de oven kunnen in totaal 0,25 m² aan pizza’s tegelijkertijd gebakken worden. Iedere pizza moet 10 minuten in de oven. | M | Wanneer er geen pizza’s in de oven liggen, is de bestelling van een enkele pizza dus over 10 minuten klaar. Is de oven helemaal vol, dan kan de pizza pas gebakken worden als een andere pizza klaar is. Moet de pizza in de oven nog 5 minuten liggen en is er vervolgens voldoende ruimte beschikbaar, dan is de bestelling dus over 5 + 10 = 15 minuten klaar. |

# Use case

In dit hoofdstuk komt het use case diagram en de beschrijvingen die bij het diagram horen.

In deze use case worden de “M” van de MoSCoW lijst meegenomen en niet de “S” en “C”, deze zijn dan ook degene die uitgewerkt worden in de beschrijvingen.

## https://documents.lucidchart.com/documents/f1e8f685-27f7-425b-bc85-54ab07852991/pages/0_0?a=1088&x=41&y=-26&w=1298&h=1012&store=1&accept=image%2F*&auth=LCA%20a843d7ce664c2038fd9b09575539bc9edf790dbe-ts%3D1484762783 Use case diagram

## Use case beschrijvingen

|  |  |
| --- | --- |
| **Naam** | Actor kan ingrediënten, producten en standaardpizza’s toevoegen |
| **Samenvatting** | Een actor kan een van de drie genoemde toevoegen aan de database |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De actor klikt op zijn hoofdscherm op de knop “Database” 2. Het systeem stuurt de actor dan naar het “Database” scherm waarop een datagridview staat met de inhoud van de database. 3. De actor klikt dan op de knop “Toevoegen/Aanpassen” 4. Het systeem stuurt de actor naar het “Toevoegen/Aanpassen” scherm 5. De actor vult dan de velden op het scherm in en klikt dan op de knop “Toevoegen” 6. Het systeem stuur de ingevoerde data naar de database en wordt daar opgeslagen 7. Het systeem stuurt de actor terug naar het hoofdscherm |
| **Uitzondering** | [1] |
| **Resultaat** | De actor heeft een nieuw item toegevoegd aan de database |

|  |  |
| --- | --- |
| **Naam** | Actor kan ingrediënten, producten en standaardpizza’s verwijderen |
| **Samenvatting** | Een actor kan een record uit de database verwijderen |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De actor klikt op zijn hoofdscherm op de knop “Database” 2. Het systeem stuurt de actor dan naar het “Database” scherm waarop een datagridview staat met de inhoud van de database. 3. De actor klikt op de record die hij wil verwijderen en klikt dan op de knop “Verwijderen” 4. Het systeem geeft dan een pop-up scherm met de tekst “Weet u zeker dat u dit record wilt verwijderen?” 5. De actor klikt dan op “Ja” 6. Het systeem verwijdert het geselecteerde record uit de database 7. Het systeem stuurt de actor terug naar het hoofdscherm |
| **Uitzondering** | [1] |
| **Resultaat** | Het door de actor geselecteerde record is uit de database verwijderd |

|  |  |
| --- | --- |
| **Naam** | Actor kan ingrediënten, producten en standaardpizza’s aanpassen |
| **Samenvatting** | De actor kan een record in de database aanpassen |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De actor klikt op zijn hoofdscherm op de knop “Database” 2. Het systeem stuurt de actor dan naar het “Database” scherm waarop een datagridview staat met de inhoud van de database. 3. De actor klikt op de record die hij wil aanpassen en klikt dan op de knop “Aanpassen” 4. Het systeem geeft dan het “Toevoegen/Aanpassen” scherm 5. De actor vult dan in de getoonde velden de waardes in die hij aan wil passen en klikt dan op de knop “Opslaan” [1] 6. Het systeem geeft deze gegevens door aan de database 7. Het systeem stuurt de actor terug naar het hoofdscherm |
| **Uitzondering** | [1] Als de actor dezelfde waardes invult die er al staan wordt hij teruggestuurd naar het “Database” scherm |
| **Resultaat** | Het door de actor geselecteerde record is aangepast in de database |

|  |  |
| --- | --- |
| **Naam** | Producten bestellen |
| **Samenvatting** | De actor zorgt ervoor dat alle producten die de pizzeria heeft in de database staat |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd als medewerker |
| **Beschrijvingen** | 1. De actor klikt op de knop “Database” vanuit zijn hoofdscherm 2. Het systeem toont een scherm met daarop de mogelijkheid om nieuwe items toe te voegen aan de database 3. De actor klikt op “Toevoegen” 4. Het systeem toont een aantal velden die die actor in moet vullen 5. De actor vult de velden naar behoren in en klikt op de knop “Toevoegen” 6. Het systeem slaat het nieuwe item op in de database 7. De actor blijft dit doen totdat hij alle items van de opdrachtomschrijving heeft toegevoegd |
| **Uitzondering** |  |
| **Resultaat** | De klant kan de producten en pizza’s uit de opdrachtomschrijving bestellen |

|  |  |
| --- | --- |
| **Naam** | Bestelling tonen |
| **Samenvatting** | De actor zorgt ervoor dat na het bestellen de bestelling van de klant zichtbaar wordt |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd als medewerker |
| **Beschrijvingen** | 1. De actor klikt op de knop “Alle bestellingen tonen” 2. Het systeem toont een lijst van alle bestellingen met naam van de klant, alle items die de klant besteld heeft en de tijd waarop deze besteld zijn |
| **Uitzondering** |  |
| **Resultaat** | De actor kan alle bestellingen bekijken |

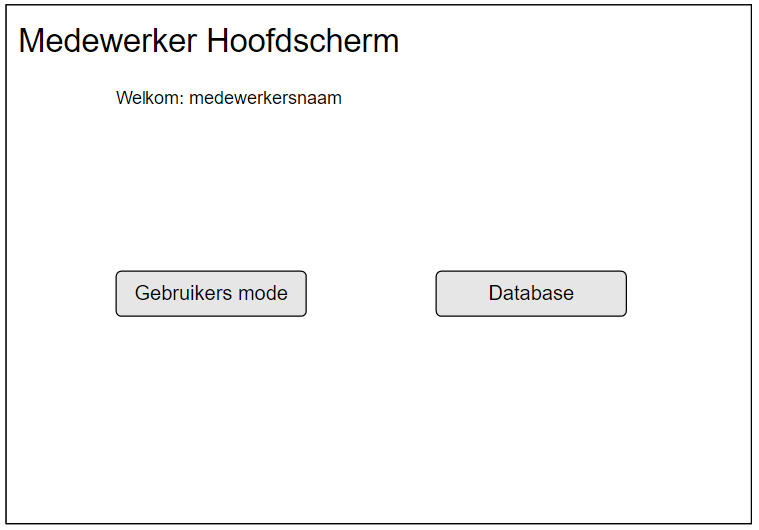
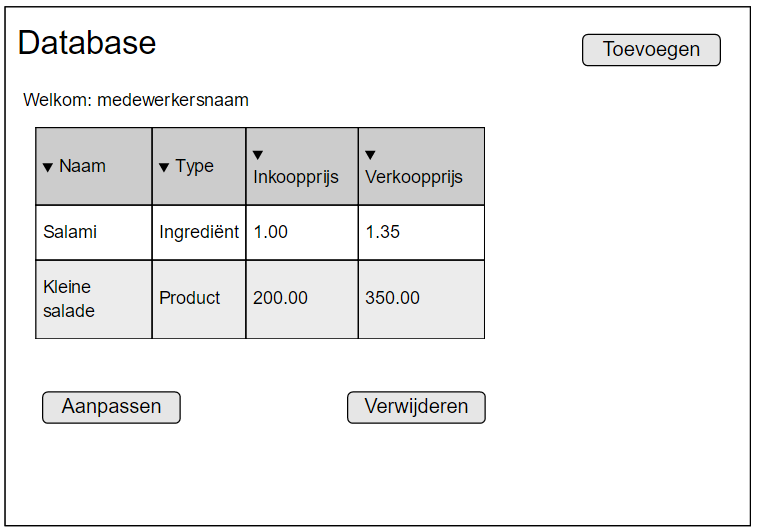
|  |  |
| --- | --- |
| **Naam** | Kassabon exporteren |
| **Samenvatting** | De actor zorgt ervoor dat er na de bestelling een kassabon op de pc van de actor komt de staan |
| **Actor** | Medewerker |
| **Aanname** | De actor is ingelogd als medewerker |
| **Beschrijvingen** | 1. De actor klikt op de knop “Alle bestellingen tonen” 2. Het systeem toont een lijst van alle bestellingen met naam van de klant, alle items die de klant besteld heeft en de tijd waarop deze besteld zijn 3. De actor klikt dan op een bestelling en nadat er een bestelling geselecteerd is op de knop “Kassabon exporteren” 4. Het systeem slaat een tekstbestand op met de waardes van de bestelling |
| **Uitzondering** |  |
| **Resultaat** | De actor heeft de kassabon op zijn pc staan |

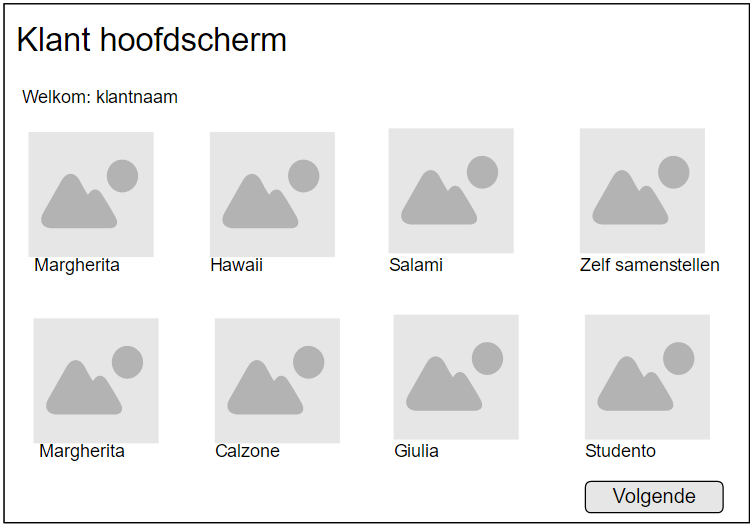
|  |  |
| --- | --- |
| **Naam** | Standaardpizza bestellen |
| **Samenvatting** | De actor kiest een standaardpizza uit en besteld deze |
| **Actor** | Klant |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De klant is na het inloggen naar zijn hoofdscherm gestuurd 2. Het systeem heeft alle standaardpizza’s ingeladen plus de optie “Zelf samenstellen” 3. Op het hoofdscherm ziet de actor alle pizza’s die hij kan bestellen hij klikt dan op de pizza “Margherita” en dan op de knop “Volgende” 4. Het systeem stuurt de actor naar het volgende scherm, “Salade” 5. De actor kan op dit scherm nog salade aan zijn bestelling toevoegen als hij dat wil. Als hij klaar is klikt de actor op de knop “Volgende” 6. Het systeem stuurt de actor door naar het scherm “Drinken” 7. Op dit scherm kan de actor eventueel nog drinken aan zijn bestelling toevoegen. Als de actor zijn keuze heeft gemaakt klikt hij op de knop “Bestellen” 8. Het systeem stuurt de actor dan naar het “Bestelling” scherm waar de bestelling van de actor weergegeven staat. 9. Als de actor tevreden is met zijn bestelling klikt hij op de knop “Bestellen” 10. Het systeem geeft dan een pop-up met de tekst “U bestelling is geplaatst” [1] |
| **Uitzondering** | [1] Als de actor niets aangeklikt heeft komt deze pop-up niet, maar daarvoor in de plaats een pop-up met de tekst “U heeft niets besteld” |
| **Resultaat** | De actor heeft een standaard pizza bestelt |

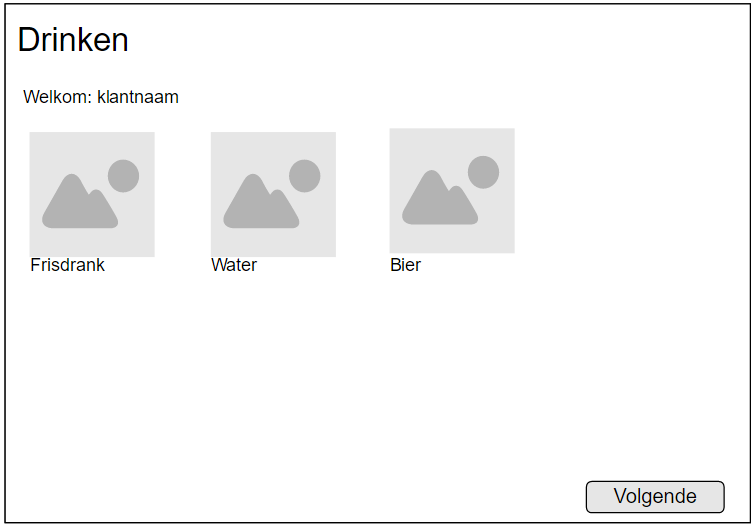
|  |  |
| --- | --- |
| **Naam** | Pizza samenstellen |
| **Samenvatting** | De actor stelt zelf een pizza samen |
| **Actor** | Klant |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De klant is na het inloggen naar zijn hoofdscherm gestuurd 2. Het systeem heeft alle standaardpizza’s ingeladen plus de optie “Zelf samenstellen” 3. Op het hoofdscherm ziet de actor alle pizza’s die hij kan bestellen hij klikt dan op ”Zelf samenstellen” en dan op de knop “Volgende” 4. Het systeem stuurt de actor naar een scherm waarop hij ingrediënten aan kan vinken om deze op zijn pizza te laten komen. Hij klikt de ingrediënten aan die hij op zijn pizza wilt en klikt “Volgende” 5. Het systeem stuurt de actor naar het volgende scherm, “Salade” 6. De actor kan op dit scherm nog salade aan zijn bestelling toevoegen als hij dat wil. Als hij klaar is klikt de actor op de knop “Volgende” 7. Het systeem stuurt de actor door naar het scherm “Drinken” 8. Op dit scherm kan de actor eventueel nog drinken aan zijn bestelling toevoegen. Als de actor zijn keuze heeft gemaakt klikt hij op de knop “Bestellen” 9. Het systeem stuurt de actor dan naar het “Bestelling” scherm waar de bestelling van de actor weergegeven staat. 10. Als de actor tevreden is met zijn bestelling klikt hij op de knop “Bestellen” 11. Het systeem geeft dan een pop-up met de tekst “U bestelling is geplaatst” [1] |
| **Uitzondering** | [1] Als de actor niets aangeklikt heeft komt deze pop-up niet, maar daarvoor in de plaats een pop-up met de tekst “U heeft niets besteld” |
| **Resultaat** | De actor heeft een standaard pizza bestelt |

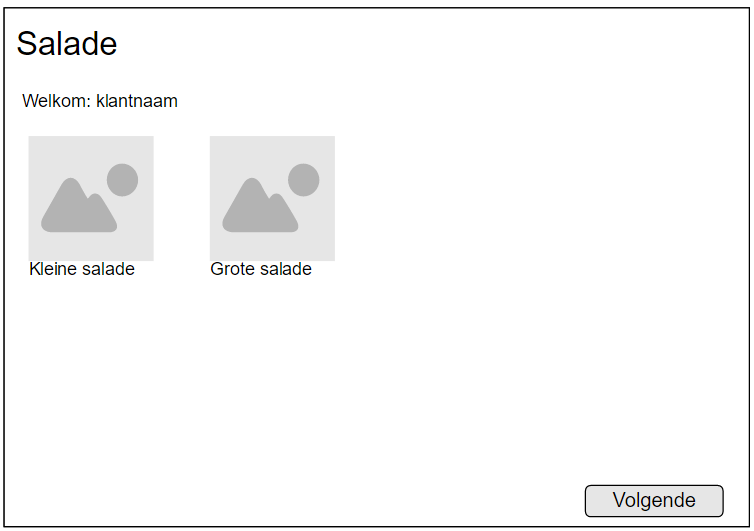
|  |  |
| --- | --- |
| **Naam** | Bestelling bepalen |
| **Samenvatting** | De actor kan zoveel producten als hij wil toevoegen aan zijn bestelling |
| **Actor** | Klant |
| **Aanname** | De actor is ingelogd |
| **Beschrijvingen** | 1. De actor klikt op een pizza die hij wil bestellen of hij kiest er geen en klikt op “Volgende” 2. Het systeem toont de actor een scherm met alle salades 3. De actor klikt de salade(s) aan die hij wil of hij kiest er geen en klikt op “Volgende” 4. Het systeem toont de actor een scherm met alle dranken 5. De actor klikt op de drank(en) die hij wil of hij kiest er geen en hij klikt op “Volgende” 6. Het systeem toont de actor een scherm met zijn complete bestelling 7. Als de actor tevreden is met zijn bestelling klikt hij op bestellen 8. Het systeem toont een pop-up scherm met de tekst “U heeft besteld” [1] |
| **Uitzondering** | [1] De actor heeft niets besteld, dan komt er een pop-up met de tekst “U bestelling is leeg” |
| **Resultaat** | De actor heeft een bestelling geplaats met zo veel items als hij wil |

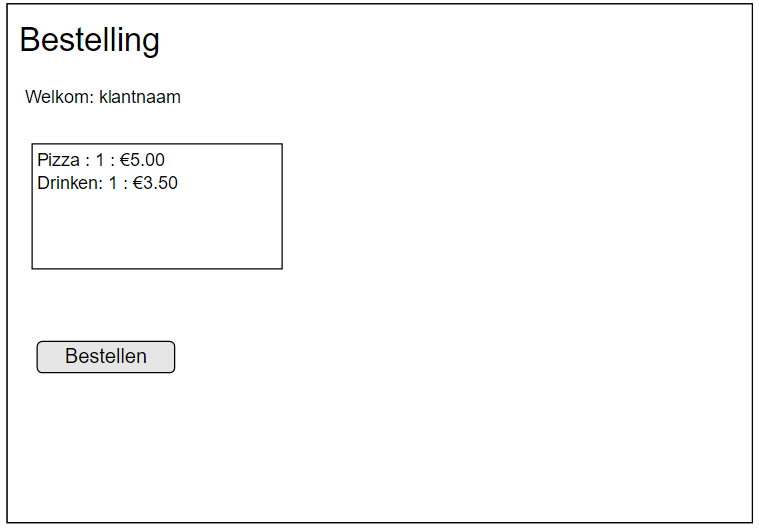
# User interface

In dit hoofdstuk komen een paar designs voor hoe de applicatie er van de buitenkant uit komt te zien.







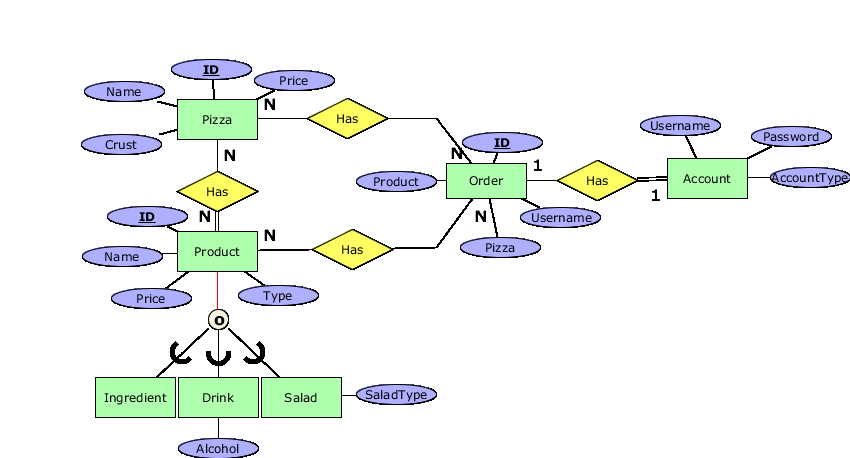


## Bijlage GUI

De grootte van mijn froms zijn 1020x500 pixels. De buttons zijn 125x30 pixels.

Ik heb er voor gekozen om de items te laten zien in listboxen in plaats van afbeeldingen en datagridviews omdat dit me veel tijd zal besparen.

# EER model

 In dit hoofdstuk komt het EER model samen met een bijlage om het model verder uit te leggen.  
Dit is een ruwe schets van hoe de database eruit gaat zien.

## EER bijlage