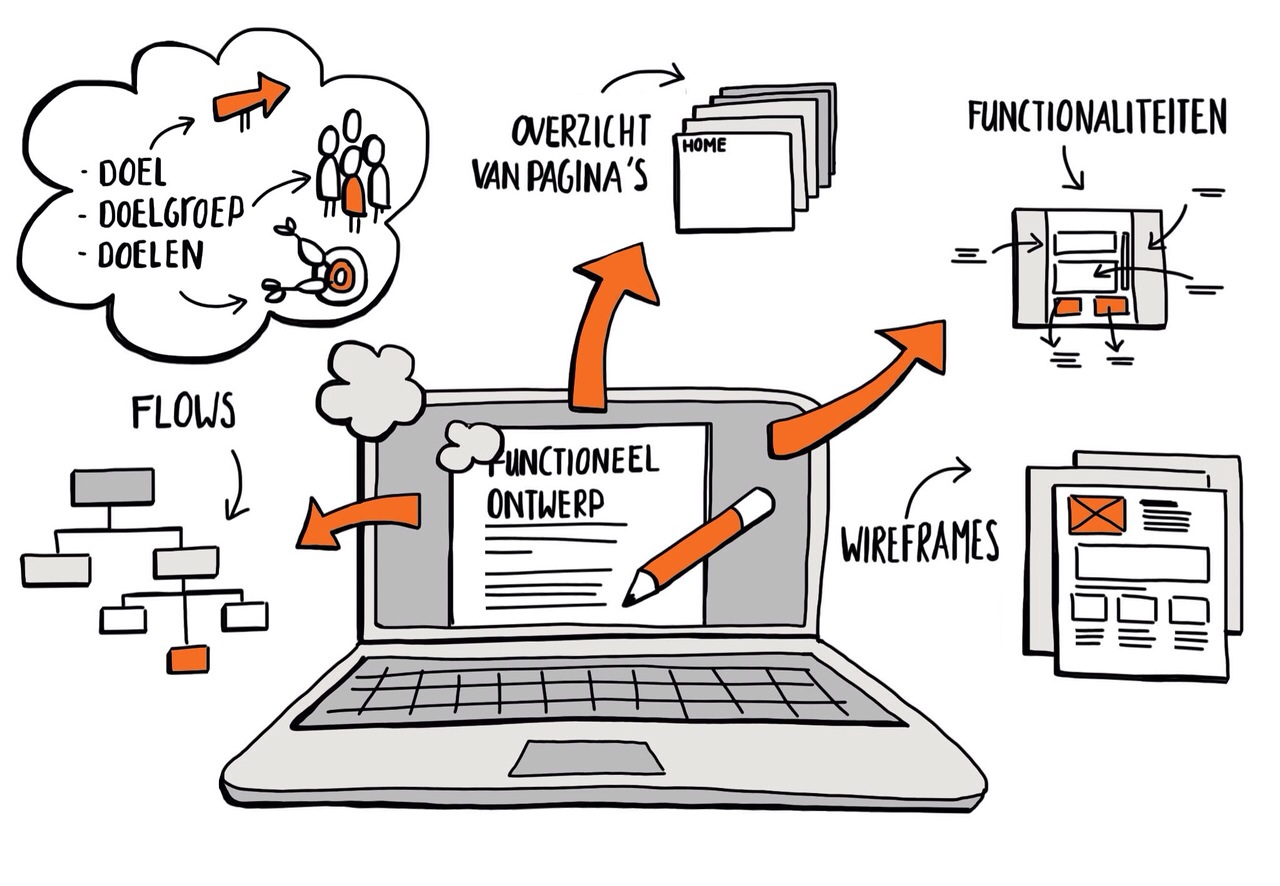
Functioneel ontwerp



**Dit Functioneel ontwerp is gemaakt door:**

Kadeem Vers, Hamza Kayouhi, Yennutiebat Noeti, Mohammed Atwa en Max Bosch.

**Project groepje:** 2

**Jaar van schrijven:** 2018

Inhoudsopgave

[Voorwoord 3](#_Toc510471201)

[Inleiding 4](#_Toc510471202)

[Use Case diagram 5](#_Toc510471203)

[Toestandsdiagram 6](#_Toc510471204)

[Bestelling invoeren 7](#_Toc510471205)

[Wireframes Use case 1 8](#_Toc510471206)

[Bestelling aanpassen 9](#_Toc510471207)

[Wireframes Use case 2 10](#_Toc510471208)

[Figuur 1 10](#_Toc510471209)

[Figuur 2 10](#_Toc510471210)

[Figuur 3 10](#_Toc510471211)

[Volgorde aanpassen 11](#_Toc510471212)

[Wireframes Use case 3 12](#_Toc510471213)

[Figuur 1 12](#_Toc510471214)

[Figuur 2 12](#_Toc510471215)

[Figuur 3 12](#_Toc510471216)

[Betaling uitvoeren 13](#_Toc510471217)

[Wireframes Use case 4 14](#_Toc510471218)

[Figuur 1 14](#_Toc510471219)

[Figuur 2 14](#_Toc510471220)

[Figuur 3 14](#_Toc510471221)

[Volledigheids check 15](#_Toc510471222)

# Voorwoord

Voor het vak Analyse en User Interfaces moeten we een functioneel ontwerp maken voor een gerecht systeem. In de les zijn we al voorbereid om daadwerkelijk een functioneel ontwerp te maken voor een applicatie. Het enigste wat meestal voor ons niet zo heel soepel gaat, is de toestand diagram. Om de toestand diagram toch goed te maken, hebben we er met z’n alle goed aandacht aan besteed. Uiteindelijk hebben we een mooie functioneel ontwerp gemaakt.

We wensen u veel leesplezier toe.

Kadeem Vers, Hamza Kayouhi, Yennutiebat Noeti, Mohammed Atwa en Max Bosch

Utrecht, 26 maart 2018

# Inleiding

Dit functioneel ontwerp is gemaakt om al een goed beeld te krijgen voor de applicatie van het project van het vak Analyse en User Interfaces.

In deze functioneel ontwerp zitten de use cases met wireframes en de bijhorende use case diagram verwerkt. Daarnaast bevindt er een toestand diagram en een volledigheidsanalyse in dit functioneel ontwerp.

# Use Case diagram



Figuur 1 Use case diagram

# Toestandsdiagram



Figuur 2 Toestandsdiagram

# Bestelling invoeren

|  |  |
| --- | --- |
| Use Case nummer | UC\_01 |
| Use Case naam | Bestelling invoeren |
| Actor | Ober, Klant |
| Samenvatting | Het systeem toont het voorgerecht, hoofdgerecht en nagerecht. Daarnaast toont het systeem per gerecht de naam en prijs van de gerechten.  De actor selecteert de gewenste **voorgerecht(i)**, **hoofdgerecht(i)** en **nagerecht(i)** in het systeem. Dan krijgt de actor in het systeem een totaalprijs(o) te zien. Daarna bevestigt de actor de bestelling in het systeem. |
| Pre-condities | Geen |
| Main scenario | 1. Systeem vraagt actor om een **voorgerecht(i),** **hoofdgerecht(i)** en **nagerecht(i)** te selecteren 2. Actor selecteert **voorgerecht(i),** **hoofdgerecht(i)** en **nagerecht(i).** 3. Systeem geeft het**totaalprijs(o).***(figuur1)* 4. Systeem vraagt actor om de bestelling te bevestigen.*(figuur2)* 5. Actor bevestigd de bestelling. |
| Post-condities (Main scenario) | 1. bestelling is ingevoerd |
| Alternative scenario (A1) | |  |  | | --- | --- | | **Trigger:** | Na stap 1; [Actor selecteert niet alle 3 de gerechten.] | | **Stappen:** | 1. Systeem geeft aan dat niet alle gerechten zijn geselecteerd.   1.1 Systeem gaat veder met stap 3. *(figuur3)* | |
| Post-condition (Alternative scenario A1) | Geen |

# Wireframes Use case 1

# Bestelling aanpassen

|  |  |
| --- | --- |
| Use Case nummer | UC\_2 |
| Use Case naam | Bestelling aanpassen |
| Actor | Klant |
| Samenvatting | Het Systeem laat de geselecteerde gerechten**(o)** zien met het **totaalbedrag(o)** de actor klikt op aanpassen en selecteert een ander **gerecht(i)**. |
| Pre-Conditie | De actor heeft een bestelling ingevoerd. |
| Main scenario | 1.Systeem laat een lijst met de geselecteerde gerechte zien met de totaalprijs.  2.Actor selecteert aanpassen. *(figuur 1)*  3.Systeem laat een nieuw scherm zien met een overzicht van **voorgerechten(o), hoofdgerechten(o)** en **nagerechten(o)**. *(figuur 2)*  4.Actor selecteert een nieuw **gerecht(i)**.  5.Systeem vraagt om een bevestiging.  6.Actor bevestigd. |
| Post-Conditie | De actor heeft de bestelling aangepast |
| Alternatief Scenario**(A1)** | |  |  | | --- | --- | | **Trigger** | Na stap 2[Actor selecteert verwijderen gerecht] | | **Stappen** | 2. Actor klikt op verwijderen en gaat verder naar stap 2. *(figuur 3)* | | **Post conditie** | De actor heeft het aanpassingsproces geannuleerd | |

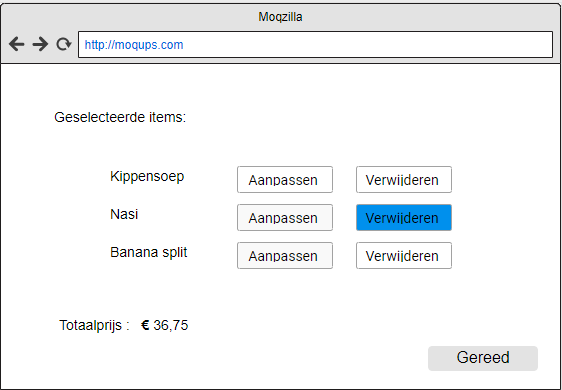
# 

# Wireframes Use case 2

## Figuur 1

## Figuur 2

## Figuur 3

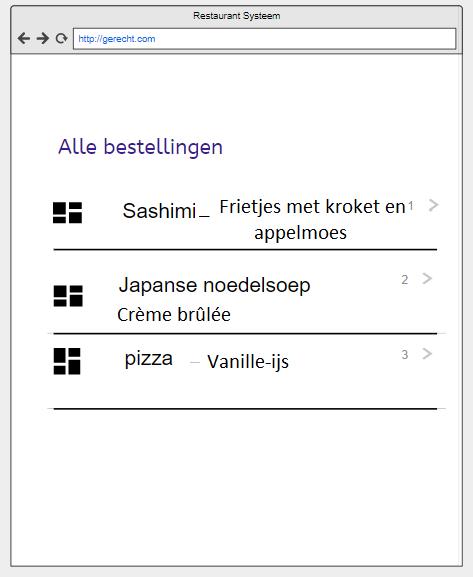


# Volgorde aanpassen

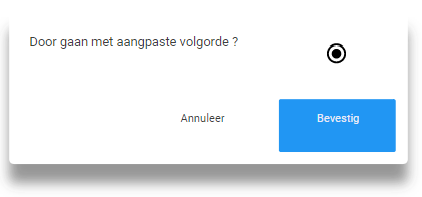
|  |  |
| --- | --- |
| Use case nummer | UC\_03 |
| Use case naam | Bestelling volgorde aanpassen |
| Actor | Chefkok |
| Samenvatting | Het systeem toont alle bestellingen op een scherm met de **bijbehorendenaam(o),** **volgorde (o)** van elke bestelling. Actor kan deze volgorde veranderen als hij dat nodig vindt door een aangepaste volgorde in te voeren bij een bepaalde besteling. |
| Pre conditie | Er zijn minimaal twee bestellingen in het Systeem. |
| Main scenario | 1. Systeem toont een lijst van bestellingen met de nummer van de **bestelling (I),** de **beschrijving(O)** en de volgorde **nummer (O)** van de bestelling. *(Zie figuur 1)* 2. Actor voert een aangepaste volgorde nummer in voor een bepaalde bestelling. 3. Systeem vraagt actor voor bevestiging. *(zie figuur 2)* 4. Actor accepteert de bevestiging. 5. Systeem toont lijst van bestellingen met de aangepaste volgorde. *(zie figuur 3)* |
| Post conditie | Volgorde aanpassen is gelukt. |
| Alternatieve scenario 1 | |  |  | | --- | --- | | **Trigger:** | Na stap 3 in de main scenario;  [Actor   accepteert de bevestiging niet.] | | **Stappen:** | 1. Systeem gaat verder met stap 1 van de main scenario. | |

# Wireframes Use case 3

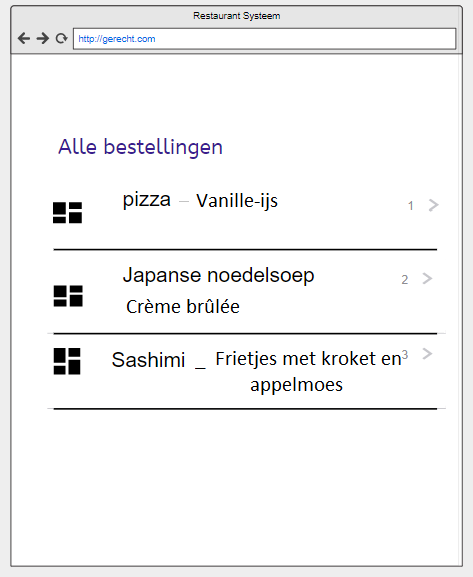
## Figuur 1



## Figuur 2



## Figuur 3



# Betaling uitvoeren

|  |  |
| --- | --- |
| ID | UC\_04 |
| Use case naam | Betaling uitvoeren |
| Actors | Klant, Ober |
| Samenvatting | Het systeem berekent het **totaalbedrag(o)** die klant moet betalen en vertoont het, vervolgens wordt het pinapparaat klaargemaakt, de betaling wordt daarna bevestigd door de ober en het systeem toont dat de **betaling status(o)** succesvol is. |
| Pre-conditie | Geen |
| Main Scenario | 1. Systeem berekent het **totaalbedrag(o)**die de klant moet betalen. *(Figuur1)* 2. Systeem maakt het pinapparaat klaar, en toont het **totaalbedrag(o).***(Figuur1)* 3. De actor voert de betaling uit met zijn pinpas. 4. Het pinapparaat vertoont dat de **betaling status(o)** succesvol is.*(Figuur2)* 5. De ober bevestigd de betaling. |
| Post-conditie | De klant ontvangt een bon als betalingsbewijs. |
| Alternatief Scenario 1 | |  |  | | --- | --- | | **Trigger:** | Na stap 3 van het Main scenario; [het pinapparaat vertoont dat de betaling is mislukt.] *(Figuur3)* | | **Stappen:** | 1. De ober reset de betaling. 2. De actor doet de betaling opnieuw. 3. Het systeem gaat verder naar stap 4 in de Main scenario. | |

# Wireframes Use case 4

## Figuur 1

## Figuur 2

## Figuur 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Bestelling invoeren | Bestelling aanpassen | Bestelling volgorde aanpassen | Betaling uitvoeren | Resultaat: |
| gerechtnaam | i | i | o | o | OK |
| bestelling volgorde nummer |  |  | i/o |  | OK |
| totaalprijs | o |  |  | o | Magic |
| betaling status |  |  |  | o | Magic |
| prijs | o | o |  | o | Magic |
| voorgerecht | i | i |  |  | Magic |
| hoofdgerecht | i | i |  |  | Magic |
| nagerecht | i | i |  |  | Magic |
| gerechtenlijst | o | o |  |  | Magic |
| betaling |  |  |  | i | Blackhole |

# Volledigheids check