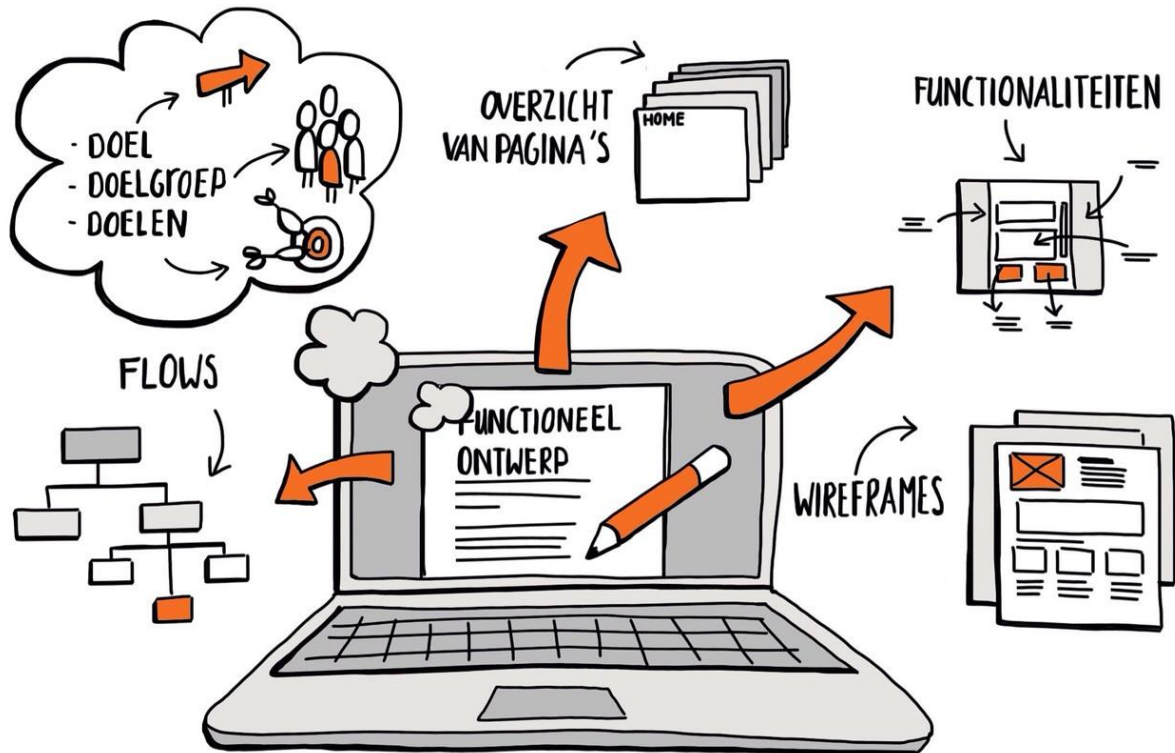


Functioneel ontwerp



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
Inleiding	4
Use Case diagram	5
Toestandsdiagram	6
Bestelling invoeren	7
Wireframes Use case 1	8
Bestelling aanpassen	9
Wireframes Use case 2	10
Figuur 1	10
Figuur 2	10
Figuur 3	10
Volgorde aanpassen	11
Wireframes Use case 3	12
Figuur 1	12
Figuur 2	12
Figuur 3	12
Betaling uitvoeren	13
Wireframes Use case 4	14
Figuur 1	14
Figuur 2	14
Figuur 3	14
Volledigheids check	15

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

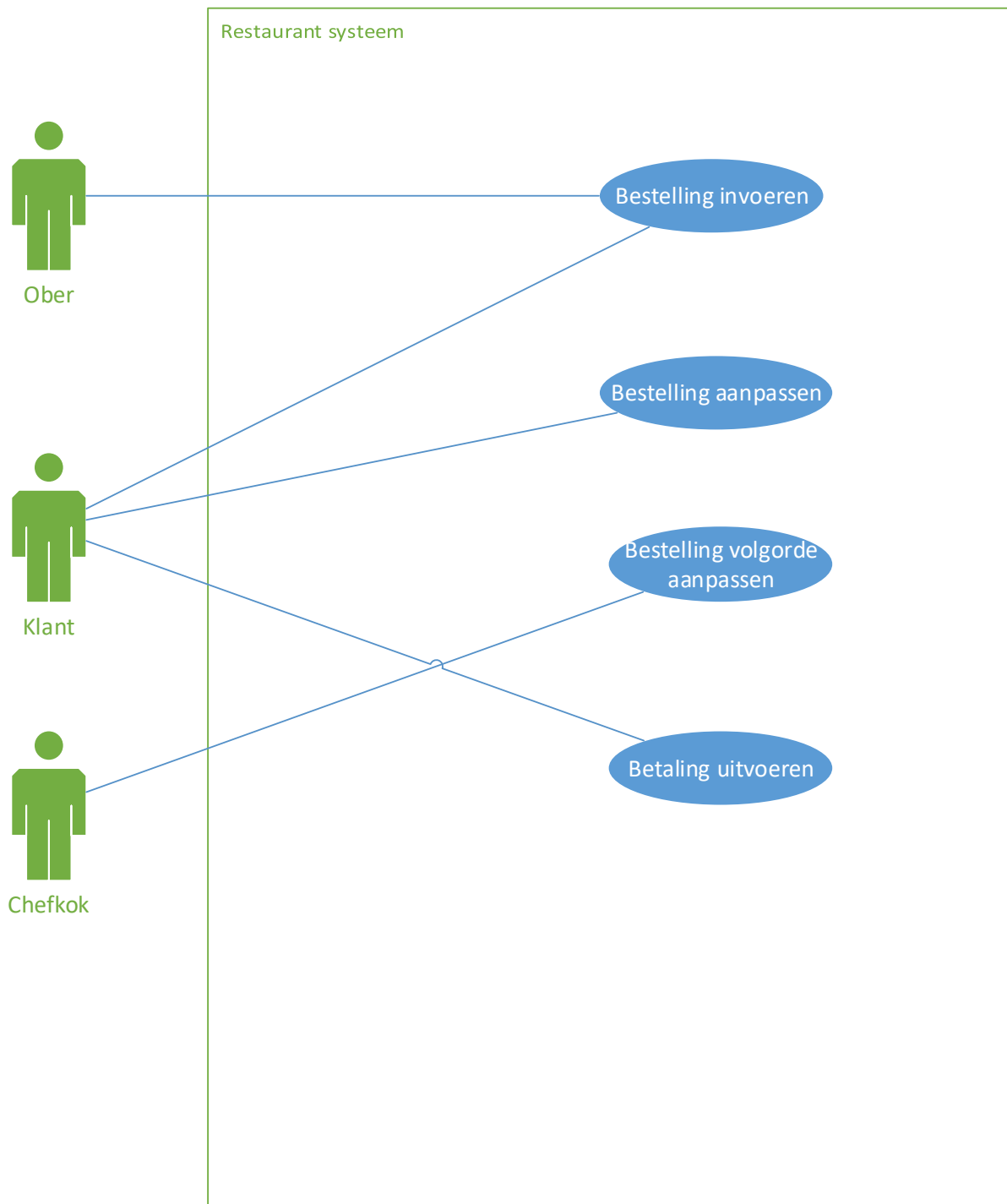
Utrecht, 25 maart 2018

Inleiding

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

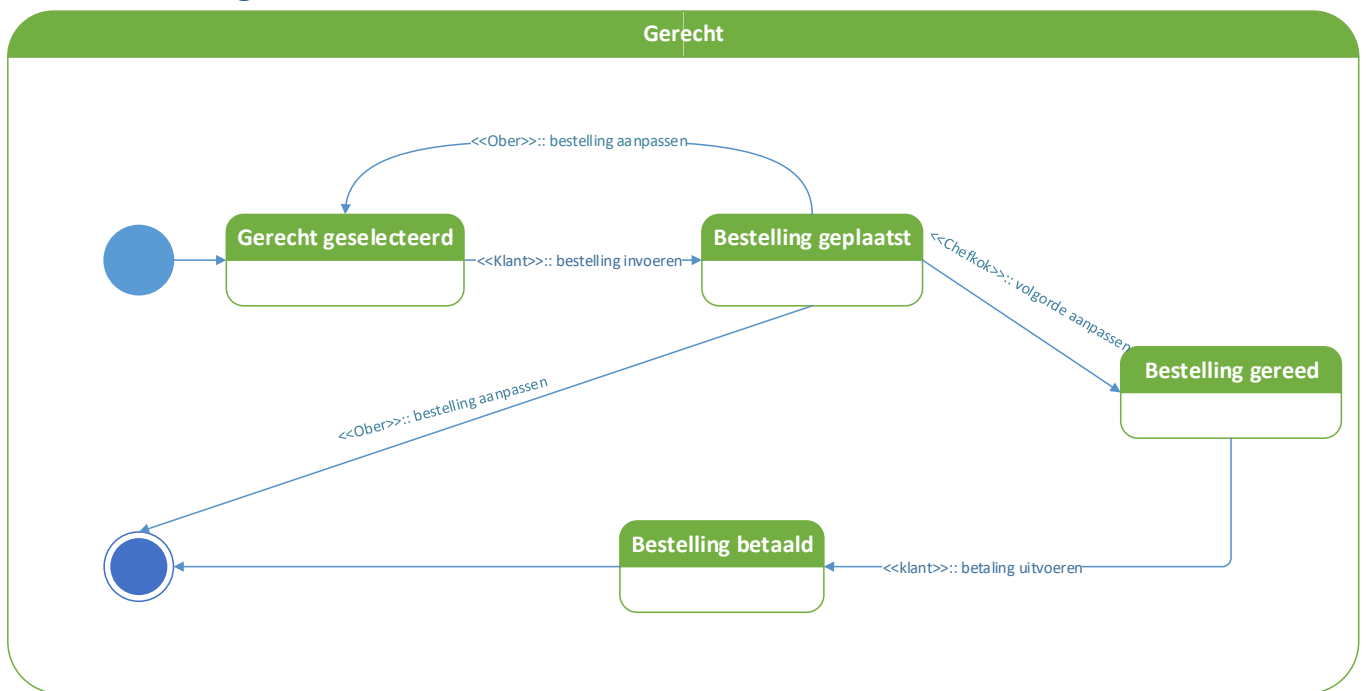
In deze functioneel ontwerp zitten de use cases met wireframes en de bijhorende use case diagram verwerkt. Daarnaast bevindt er een toestand diagram en een volledighedsanalyse in dit functioneel ontwerp.

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	<p>Het systeem toont het voorgerecht, hoofdgerecht en nagerecht. Daarnaast toont het systeem per gerecht de naam en prijs van de gerechten.</p> <p>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.</p>	
Pre-condities	Geen	
Main scenario	<ol style="list-style-type: none"> 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). (<i>figuur1</i>) 4. Systeem vraagt actor om de bestelling te bevestigen. (<i>figuur2</i>) 5. Actor bevestigt 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:	<ol style="list-style-type: none"> 1. Systeem geeft aan dat niet alle gerechten zijn geselecteerd. 1.1 Systeem gaat verder met stap 3. (<i>figuur3</i>)
Post-condition (Alternative scenario A1)	Geen	

Wireframes Use case 1

Polymer applicatie

← → ↺

Voorgerecht	Hoofdgerecht	Nagerecht
Soep €8,-	Biefstuk €40,-	Taart € 9,-
Broodje €3,-	Stampot €35,-	ijs €7,-
Nasi €22,-	Kip geheel €15,-	Toetje €8,-

= Button/knop

Totaalprijs € 95,-

Figuur 2

Figuur 1

✖ Niet alle 3 de gerechten zijn geselecteerd ✖

Figuur 3

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. <i>(figuur 1)</i> 3.Systeem laat een nieuw scherm zien met een overzicht van voorgerechten(o) , hoofdgerechten(o) en nagerechten(o) . <i>(figuur 2)</i> 4.Actor selecteert een nieuw gerecht(i) . 5.Systeem vraagt om een bevestiging. 6.Actor bevestigt.	
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. <i>(figuur 3)</i>
	Post conditie	De actor heeft het aanpassingsproces geannuleerd

Wireframes Use case 2

Figuur 1

Moqzilla

← → ↻ <http://moqups.com>

Geselecteerde items:

Kippensoep	Aanpassen	Verwijderen
Nasi	Aanpassen	Verwijderen
Banana split	Aanpassen	Verwijderen

Totaalprijs : € 36,75

Gereed

Figuur 2

Moqzilla

← → ↻ <http://moqups.com>

▼ Voorgerecht	▼ Hoofdgerecht	▼ Nagerecht
Nasi	Bami	Noodles
Spaghetti	Lasagne	Softijs
Soep	Kipnuggets	smurfenijs
Gehaktbal	Empanadas	Aardappelkartoffels

Gereed

Figuur 3

Moqzilla

← → ↻ <http://moqups.com>

Geselecteerde items:

Kippensoep	Aanpassen	Verwijderen
Nasi	Aanpassen	Verwijderen
Banana split	Aanpassen	Verwijderen

Totaalprijs : € 36,75

Gereed

Volgorde aanpassen

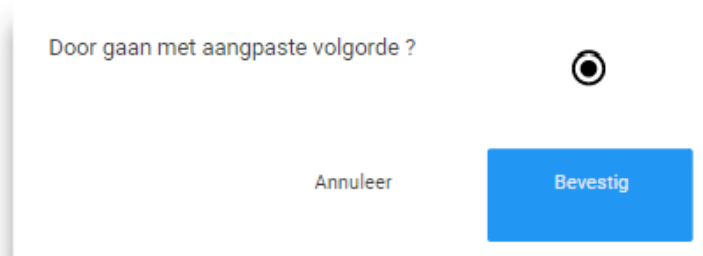
Use case nummer	UC_03	
Use case naam	Bestelling volgorde aanpassen	
Actor	Chefkok	
Samenvatting	Het systeem toont alle bestellingen op een scherm met de bijbehorendenaam(o) , volgorde (o) van elke bestelling. Actor kan deze volgorde veranderen als hij dat nodig vindt door een aangepaste volgorde in te voeren bij een bepaalde bestelling.	
Pre conditie	Er zijn minimaal twee bestellingen in het Systeem.	
Main scenario	<ol style="list-style-type: none"> 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 <i>figuur 1</i>) 2. Actor voert een aangepaste volgorde nummer in voor een bepaalde bestelling. 3. Systeem vraagt actor voor bevestiging. (zie <i>figuur 2</i>) 4. Actor accepteert de bevestiging. 5. Systeem toont lijst van bestellingen met de aangepaste volgorde. (zie <i>figuur 3</i>) 	
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

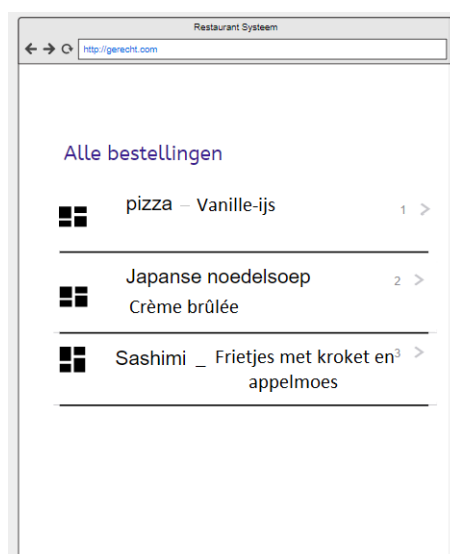
Figuur 1



Figuur 2



Figuur 3

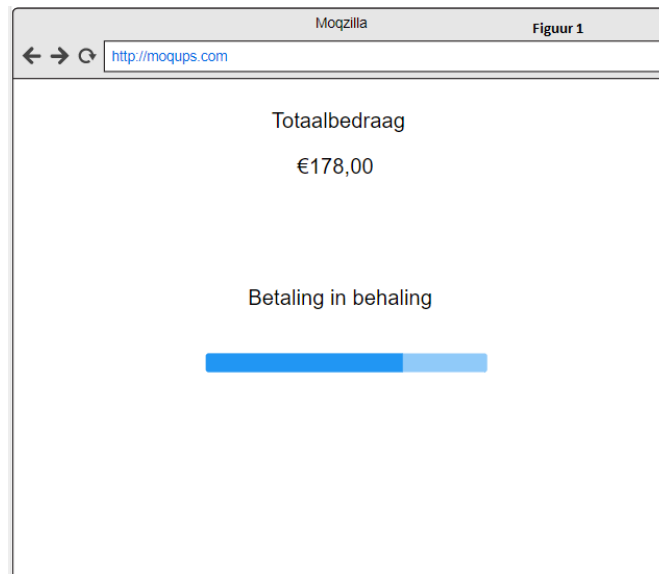


Betaling uitvoeren

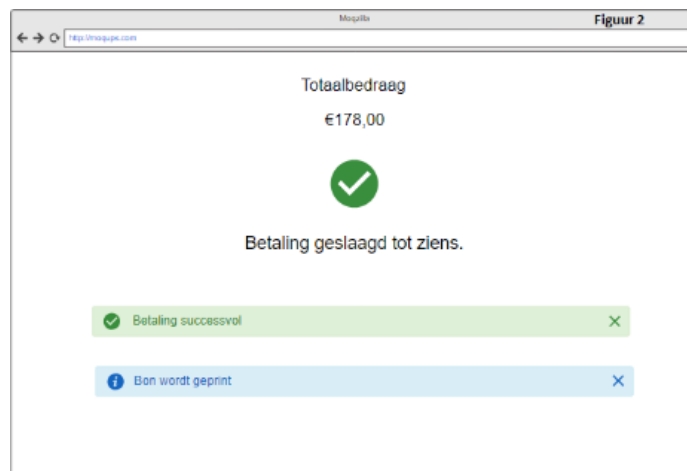
ID	UC_04	
Use case naam	Betaling uitvoeren	
Actors	Klant, Ober	
Samenvatting	Het systeem berekent het totaalbedrag(o) die klant moet betalen en vertoont het, vervolgens wordt het pinapparaat klaargemaakt, de betaling wordt daarna bevestigd door de ober en het systeem toont dat de betaling status(o) succesvol is.	
Pre-conditie	Geen	
Main Scenario	<ol style="list-style-type: none"> 1. Systeem berekent het totaalbedrag(o) die de klant moet betalen. <i>(Figuur1)</i> 2. Systeem maakt het pinapparaat klaar, en toont het totaalbedrag(o). <i>(Figuur1)</i> 3. De actor voert de betaling uit met zijn pinpas. 4. Het pinapparaat vertoont dat de betaling status(o) succesvol is. <i>(Figuur2)</i> 5. De ober bevestigt 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.] <i>(Figuur3)</i>
	Stappen:	<ol style="list-style-type: none"> 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

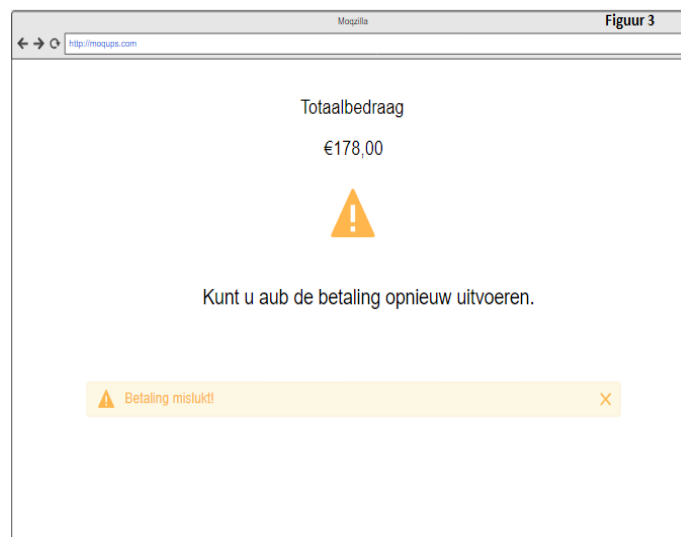
Figuur 1



Figuur 2



Figuur 3



Volledigheids check

	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