

# FUNCTIONEEL ONTWERP



Voor order-systeem sushi restaurant

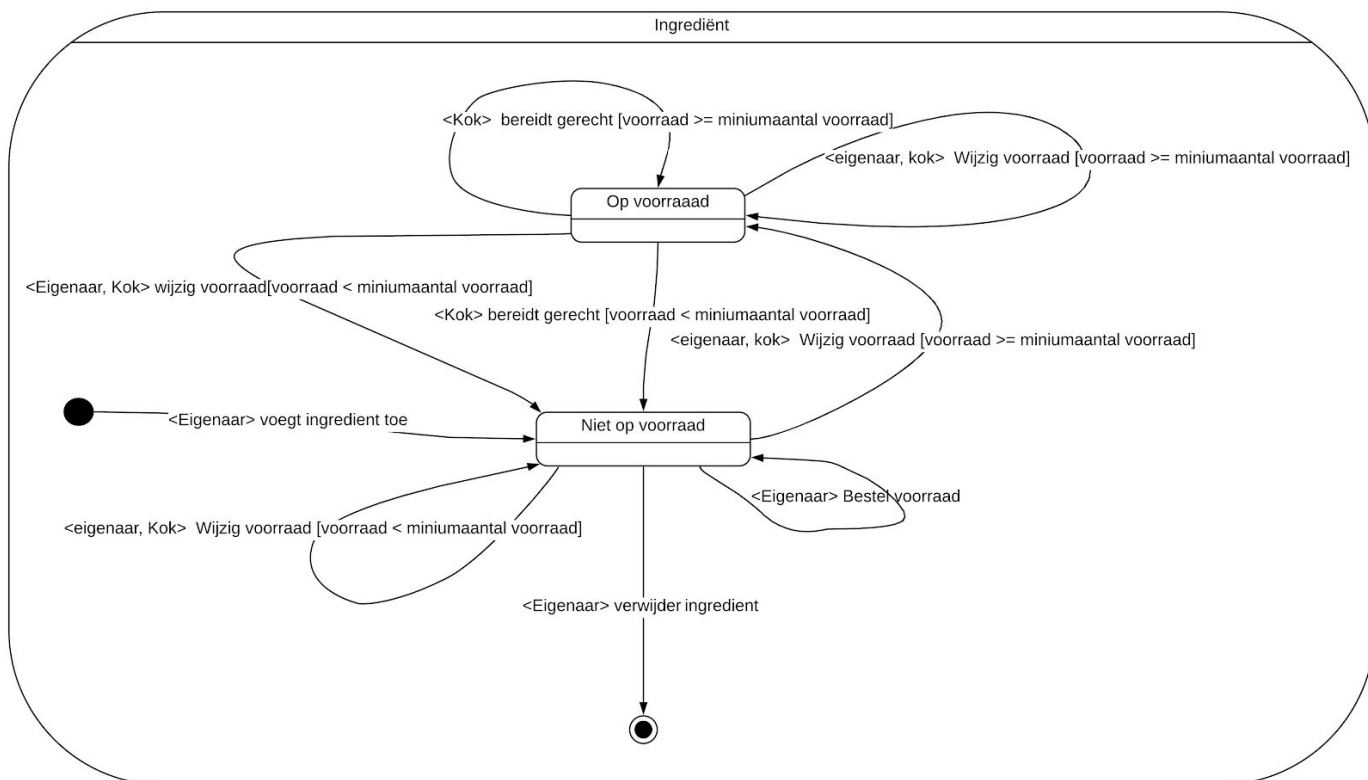
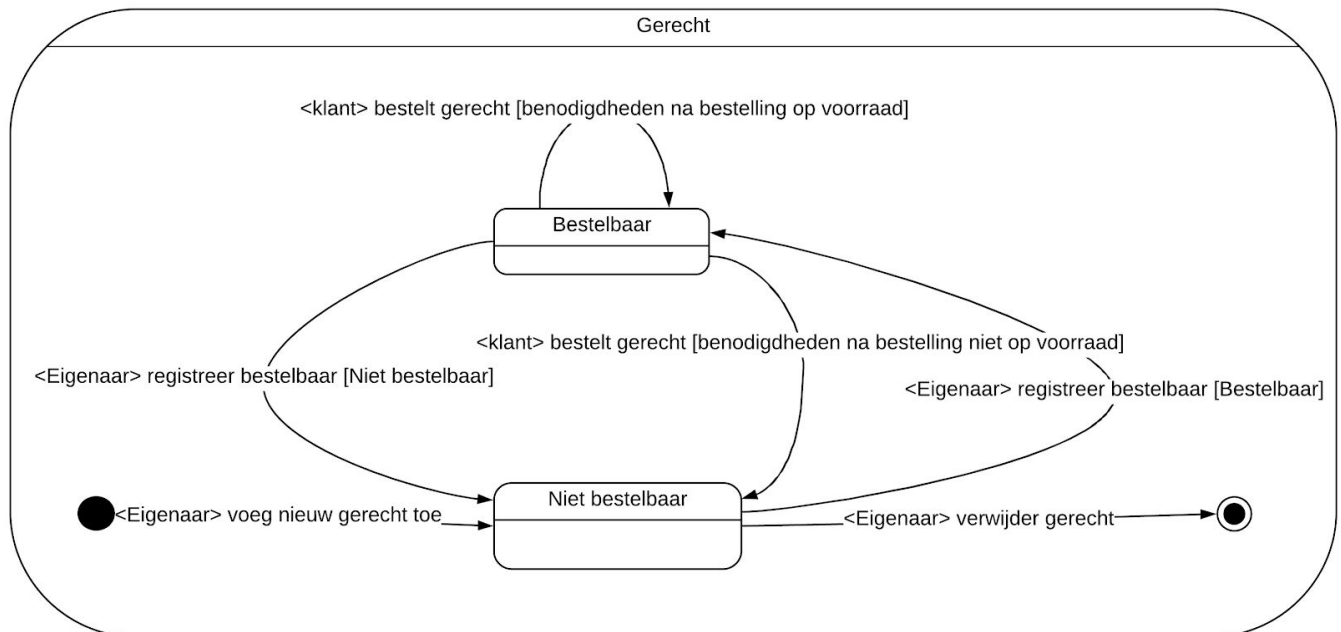
# Contents

<b>Inleiding</b>	<b>1</b>
Toestandsdiagram	2
Use Case Diagram	3
Use Case Samenvattingen	4
Use case Templates	5
Use Case - Gerecht Toevoegen	5
Use Case - Gerechten Verwijderen	5
Use Case - Gerecht bestelbaar/onbestelbaar Maken	7
Use Case - Gerecht Bestellen	8
Use Case – Bereid gerecht	9
Use Case - Wijzig voorraad	9
Use Case - Ingredient toevoegen	10
Use Case - Ingrediënt verwijderen	11
Use Case - Voorraad bestellen	12
<b>Bijlagen</b>	<b>13</b>
Volledigheid Analyse	13

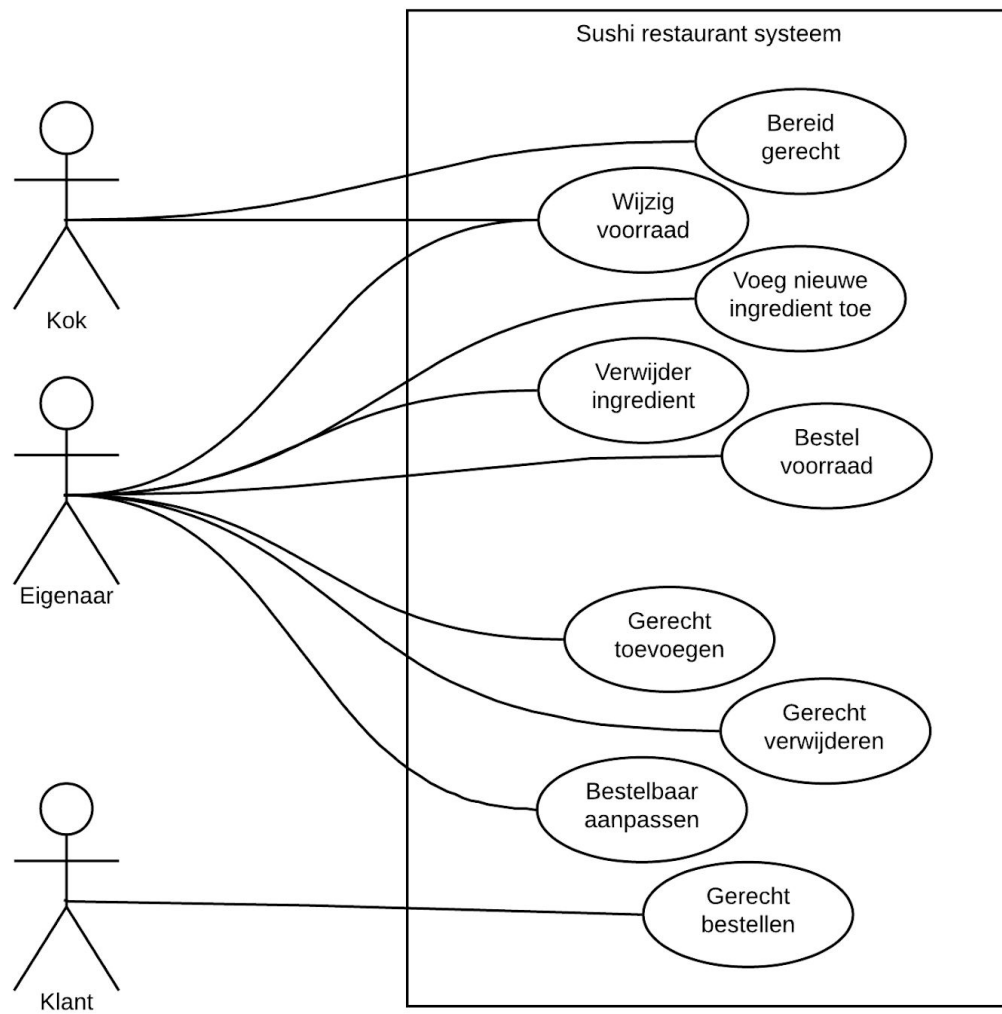
## Inleiding

Dit functioneel ontwerp gaat over de casus sushi restaurant.

## Toestandsdiagram



## Use Case Diagram



## Use Case Samenvattingen

Gerecht toevoegen	De actor voert de naam[in], omschrijving[in], seizoen[in] en prijs[in], en benodigde ingrediënten[in] van een gerecht in het systeem. Het systeem kent een uniek gerecht nummer[in] aan het gerecht toe.
Gerecht verwijderen	het systeem weergeeft een lijst van bestaande gerechten (gerecht nummers[out], naam [out]). Nadat de actor een gerecht heeft geselecteerd laat het systeem de informatie: naam[out], omschrijving[out], seizoen[out] en prijs[out] van een gerecht zien. actor gerecht uit het systeem.
Gerecht bestelbaar / onbestelbaar maken	Het systeem weergeeft een lijst van bestaande gerechten met gerecht nummer [out] en naam [out]. Nadat de actor een gerecht heeft geselecteerd laat het systeem de informatie: naam[out], omschrijving[out], seizoen[out] en prijs[out] van een gerecht zien. De actor geeft aan of het gerecht bestelbaar of onbestelbaar is en het systeem registreert het betreffende gerecht als zodanig.
Gerecht bestellen	Het systeem geeft een lijst van gerechten die besteld kunnen worden weer met de naam[out], omschrijving[out] en de prijs[out] en vraagt welke gerechten er besteld moeten worden en hoeveel. De actor geeft aan welke gerechten en hoeveel daarvan er besteld moeten worden. Het systeem voegt deze gerechten toe aan de lijst met gerechten die in bestelling zijn en de tijd[in] van de bestelling.
Bereid gerecht	Het systeem geeft een lijst met gerechten die in bestelling zijn weer, met de naam[out] en de tijd[out] van bestelling de lijst is gesorteerd op tijd van bestelling. De actor selecteert een besteld gerecht en geeft aan dat deze bereid wordt. In het systeem wordt de voorraad van ingrediënten aangepast op basis van welke ingrediënten[out] voor dit gerecht nodig zijn en de hoeveelheid[out] ervan.
Wijzig voorraad	Het systeem geeft de lijst met ingrediënten weer met de naam[out], hoeveelheid[out], houdbaarheidsdatum [out] en minimumvoorraad[out] van het ingrediënt. De actor verandert de hoeveelheid[in] van de ingrediënten.
Ingrediënt toevoegen	De actor voert in het systeem de houdbaarheidsdatum[in], naam[in], type[in] en het aantal[in] van het ingrediënt in. Vervolgens kent het systeem een uniek ingrediënt nummer[in] aan het ingrediënt toe.
Ingrediënt verwijderen	Het systeem weergeeft een lijst van ingrediënten namen[out] wanneer de actor een ingrediënt naam[in] selecteert dan geeft het systeem de informatie weer: ingrediënt nummer, naam[out], houdbaarheidsdatum[out] en het aantal[out] van het ingrediënt in. Als de actor het ingrediënt besluit te verwijderen, dan haalt het systeem deze informatie van het ingrediënt weg.
Voorraad bestellen	Het systeem geeft de lijst met ingrediënten weer met naam[out] en hoeveelheid[out] en minimumvoorraad[out]. De actor vult voor ingrediënten die onder (optioneel ook boven) de minimum voorraad zijn de nieuwe gewenste voorraad in en het systeem genereert een bestellijst in de vorm van een excel bestand met de naam[in], type[in] van elk ingrediënt en de hoeveelheid[in].

# Use case Templates

## Use Case - Gerecht Toevoegen

Use-case nummer	UC1-Gerecht toevoegen	
Use-case naam	Gerecht toevoegen	
Actor	Eigenaar	
Samenvatting	De actor voert de naam[in], omschrijving[in], seizoen[in] en prijs[in], en benodigde ingrediënten[in] van een gerecht in het systeem. Het systeem kent een uniek gerecht nummer[in] aan het gerecht toe.	
Preconditie	-	
Main scenario	<ol style="list-style-type: none"> <li>1. Het systeem vraagt om een naam, omschrijving</li> <li>2. Actor verstrekt deze informatie.</li> <li>3. Het systeem registreert het gerecht en toont een melding dat het gerecht is geregistreerd.</li> </ol>	
Postconditie	Een nieuw gerecht is geregistreerd in het systeem.	
Alternatief scenario 1	Trigger	Na stap 1 van het main scenario [Actor annuleert]
	Scenario	1. het systeem stopt de use case
	Postconditie	Het systeem is ongewijzigd
Alternatief scenario 2	Trigger	Na stap 2 van het main scenario [Niet alle velden zijn ingevuld]
	Scenario	<ol style="list-style-type: none"> <li>1. systeem meldt welke velden nog ingevuld moeten worden</li> <li>2. Systeem gaat terug naar stap 1 van het main scenario</li> </ol>

## Use Case - Gerechten Verwijderen

Use-case nummer	UC2-gerecht-verwijderen							
Use-case naam	Verwijder gerecht							
Actor	Eigenaar							
Samenvatting	Het systeem weergeeft een lijst van bestaande gerechten (gerecht nummers[out], naam [out]). Nadat de actor een gerecht heeft geselecteerd laat het systeem de informatie: naam[out], omschrijving[out], seizoen[out] en prijs[out] van een gerecht zien. actor gerecht uit het systeem.							
Preconditie	-							
Main scenario	<div>1. Het systeem toont een lijst met bestaande gerechten met gerecht nummer en naam en vraagt welke gerechten verwijderd moeten worden.</div> <div>2. De actor selecteert één of meer gerechten om te verwijderen.</div> <div>3. Het systeem vraagt of de gerechten echt verwijderd moeten worden.</div> <div>4. De actor keurt het verwijderen van de gerechten goed.</div> <div>5. Het systeem verwijderd de gerechten.</div>							
Alternatief scenario 1	<table><tr><td>Trigger</td><td>Na stap 3 van het main scenario [De actor keurt het verwijderen van het gerecht af]</td></tr><tr><td>Stappen</td><td>1. Het systeem gaat terug naar stap 1.</td></tr><tr><td>Postconditie</td><td>Het systeem is ongewijzigd</td></tr></table>		Trigger	Na stap 3 van het main scenario [De actor keurt het verwijderen van het gerecht af]	Stappen	1. Het systeem gaat terug naar stap 1.	Postconditie	Het systeem is ongewijzigd
Trigger	Na stap 3 van het main scenario [De actor keurt het verwijderen van het gerecht af]							
Stappen	1. Het systeem gaat terug naar stap 1.							
Postconditie	Het systeem is ongewijzigd							
Alternatief scenario 2	<table><tr><td>Trigger</td><td>Na stap 1 van het main scenario [Acteur cancelt scherm van verwijderen van gerechten.].</td></tr></table>		Trigger	Na stap 1 van het main scenario [Acteur cancelt scherm van verwijderen van gerechten.].				
Trigger	Na stap 1 van het main scenario [Acteur cancelt scherm van verwijderen van gerechten.].							

	Stappen	1. Use case wordt beëindigd.
	Postconditie	Het systeem is ongewijzigd
Postconditie	Één of meer rechten zijn uit het systeem verwijderd.	

## Use Case - Gerecht bestelbaar/onbestelbaar Maken

ID	UC3-wijzig_gerecht_bestelbaar							
Naam	Maak Gerecht Bestelbaar							
Actor	Systeem, Eigenaar							
Samenvatting	Het systeem weergeeft een lijst van bestaande gerechten met naam[out] en gerecht nummers[out]. Nadat de actor een gerecht heeft geselecteerd laat het systeem de informatie: naam[out], omschrijving[out], seizoen[out] en prijs[out] van een gerecht zien. De actor geeft aan of het gerecht bestelbaar of onbestelbaar is en het systeem registreert het betreffende gerecht als zodanig.							
Preconditie	Gerecht is niet bestelbaar. Gerecht is op voorraad.							
Main Scenario	<div>1. Systeem geeft alle bestaande gerechten weer met nummer en naam en vraagt welk gerecht bestelbaar of onbestelbaar moet worden.</div> <div>2. De actor selecteert een gerecht.</div> <div>3. Het systeem geeft het gerecht weer met meer gegevens: seizoen, omschrijving en prijs.</div> <div>4. De actor verandert of het gerecht bestelbaar is.</div> <div>5. Systeem wijzigt de status van het gerecht naar onbestelbaar of bestelbaar.</div>							
Postconditie	In het systeem is veranderd een gerecht bestelbaar is of niet							
Alternatief Scenario 1	<table><tr><td>Trigger</td><td>Na stap 1 van het main scenario [Actor annuleert]</td></tr><tr><td>Scenario</td><td>1. het systeem stopt de use case</td></tr><tr><td>Postconditie</td><td>Het systeem is ongewijzigd</td></tr></table>		Trigger	Na stap 1 van het main scenario [Actor annuleert]	Scenario	1. het systeem stopt de use case	Postconditie	Het systeem is ongewijzigd
Trigger	Na stap 1 van het main scenario [Actor annuleert]							
Scenario	1. het systeem stopt de use case							
Postconditie	Het systeem is ongewijzigd							
Alternatief Scenario 2	<table><tr><td>Trigger</td><td>Na stap 3 van het main scenario [Actor annuleert het</td></tr></table>		Trigger	Na stap 3 van het main scenario [Actor annuleert het				
Trigger	Na stap 3 van het main scenario [Actor annuleert het							



		veranderen van bestelbaar/niet bestelbaar]
	Scenario	1. Het systeem gaat terug naar stap 1 van het main scenario

## Use Case - Gerecht Bestellen

ID	UC4-bestel_gerecht								
Naam	Bestel Gerecht								
Actor	Klant								
Samenvatting	Het systeem geeft een lijst van gerechten die besteld kunnen worden weer met de naam[out], omschrijving[out] en de prijs[out] en vraagt welke gerechten er besteld moeten worden en hoeveel. De actor geeft aan welke gerechten en hoeveel daarvan er besteld moeten worden. Het systeem voegt deze gerechten toe aan de lijst met gerechten die in bestelling zijn en de tijd[in] van de bestelling.								
Preconditie									
Main Scenario	<div>1. Systeem geeft alle bestelbare gerechten weer met naam, omschrijving en de prijs en vraagt welke gerechten er besteld moeten worden.</div> <div>2.<div>a. Actor kiest één of meer gerechten.</div>b. Actor bevestigt bestelling.</div> <div>3.<div>a. Systeem koppelt de bestelling door naar de keuken.</div>b. Systeem laat aan de actor weten dat de bestelling succesvol is doorgekoppeld.</div> c. Systeem vermindert de voorraden van alle gebruikte ingrediënten in het gerecht.								
Postconditie	Actor heeft bestelling gemaakt.								
Alternatief scenario 1	<table><tr><td>Trigger</td><td>Na stap 1 van het main scenario [Actor verlaat het scherm]</td></tr><tr><td>Scenario</td><td>1. het systeem stopt de use case</td></tr><tr><td>Postconditie</td><td>Het systeem is ongewijzigd</td></tr></table>			Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]	Scenario	1. het systeem stopt de use case	Postconditie	Het systeem is ongewijzigd
Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]								
Scenario	1. het systeem stopt de use case								
Postconditie	Het systeem is ongewijzigd								

<b>Alternatief scenario 2</b>	<table> <tr> <td>Trigger</td><td>Na stap 2a van het main scenario [Actor verlaat het scherm]</td></tr> <tr> <td>Scenario</td><td>1. het systeem gaat terug naar stap 1 van het main scenario</td></tr> <tr> <td>Postconditie</td><td>Het systeem is ongewijzigd</td></tr> </table>	Trigger	Na stap 2a van het main scenario [Actor verlaat het scherm]	Scenario	1. het systeem gaat terug naar stap 1 van het main scenario	Postconditie	Het systeem is ongewijzigd
Trigger	Na stap 2a van het main scenario [Actor verlaat het scherm]						
Scenario	1. het systeem gaat terug naar stap 1 van het main scenario						
Postconditie	Het systeem is ongewijzigd						

## Use Case – Bereid gerecht

<b>Use-case nummer</b>	UC5-bereid_gerecht				
<b>Use-case naam</b>	Bereid gerecht				
<b>Actor</b>	Kok				
<b>Samenvatting</b>	Het systeem geeft een lijst met gerechten die in bestelling zijn weer, met de naam[ <b>out</b> ] en de tijd[ <b>out</b> ] van bestelling de lijst is gesorteerd op tijd van bestelling. De kok selecteert een besteld gerecht en geeft aan dat deze bereid wordt. In het systeem wordt de voorraad van ingrediënten aangepast op basis van welke ingrediënten[ <b>out</b> ] voor dit gerecht nodig zijn en hoeveel[ <b>out</b> ] er nog zijn.				
<b>Preconditie</b>	Er zijn ingrediënten op voorraad.				
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. Het systeem geeft een lijst (gesorteerd op tijd) met mogelijke gerechten met de naam en de tijd van bestelling.</li> <li>2. De kok kiest een gerecht die hij wil bereiden.</li> <li>3. <ol style="list-style-type: none"> <li>a. Het systeem verandert de voorraad van de gebruikte ingrediënten.</li> <li>b. Het systeem gaat terug naar stap 1.</li> </ol> </li> </ol>				
<b>Postconditie</b>	De voorraad van één of meer ingrediënten is veranderd in het systeem.				
<b>Alternatief scenario 1</b>	<table> <tr> <td>Trigger</td><td>Na stap 1 van het main scenario [Actor verlaat het scherm]</td></tr> <tr> <td>Scenario</td><td>1. het systeem stopt de use case</td></tr> </table>	Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]	Scenario	1. het systeem stopt de use case
Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]				
Scenario	1. het systeem stopt de use case				

## Use Case - Wijzig voorraad

Use-case nummer	UC6-wijzig_voorraad	
Use-case naam	Wijzig voorraad	
Actor	Kok	
Samenvatting	Het systeem geeft de lijst met ingrediënten weer met de naam[out], hoeveelheid[out], houdbaarheidsdatum [out] en minimumvoorraad[out] van het ingrediënt. De eigenaar/kok verandert de hoeveelheid[in] van de ingrediënten.	
Preconditie	geen	
Main scenario	<div><div>1.</div><div>Het systeem geeft een lijst met de beschikbare ingrediënten weer met hun naam, voorraad, houdbaarheidsdatum en minimumvoorraad.</div></div> <div><div>2.</div><div>De kok past de voorraad van de gewenste ingrediënten aan.</div></div> <div><div>3.</div><div>Het systeem slaat gewijzigde voorraad op.</div></div>	
Postconditie	De voorraad is veranderd in het systeem.	
Alternatief scenario 1		
	Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]
	Scenario	<div>1.</div> <div>het systeem stopt de use case</div>

## Use Case - Ingredient toevoegen

Use-case nummer	UC7-nieuwe_ingrediënt	
Use-case naam	Voeg nieuwe ingrediënten toe	
Actor	Eigenaar	
Samenvatting	De actor voert in het systeem de naam[in], type[in], houdbaarheidsdatum[in], minimumvoorraad[in], en het aantal[in] van het ingrediënt in. Vervolgens kent het systeem een uniek ingrediëntnummer[in] aan het ingrediënt toe.	
Pre-conditie	-	
Main scenario	<ol style="list-style-type: none"><li>1. Het systeem vraagt de naam, de eenheid waarmee de voorraad bijgehouden moet worden, de voorraad (aantal) en de minimumvoorraad voor het nieuwe ingrediënt.</li><li>2. De eigenaar voert de gevraagde gegevens in.</li><li>3. Het systeem registreert het nieuwe ingrediënt.</li></ol>	
Alternatief scenario 1	Trigger	Na stap 1 van het main scenario [Actor verlaat het scherm]
	Scenario	1. het systeem stopt de use case

<b>Alternatief scenario 2</b>	Trigger	Na stap 2 van het main scenario [Niet de nodige waarden gegeven.]
	Stappen	<ol style="list-style-type: none"> <li>1. Het systeem toont een melding dat niet alle waarden zijn gegeven.</li> <li>2. Het systeem gaat terug naar stap 1.</li> </ol>
<b>Postconditie</b>	Er is een nieuwe ingrediënt toegevoegd aan het systeem.	

## Use Case - Ingrediënt verwijderen

<b>Use-case nummer</b>	UC8-verwijder_ingrediënt	
<b>Use-case naam</b>	Verwijder ingrediënt	
<b>Actor</b>	Eigenaar	
<b>Samenvatting</b>	Het systeem weergeeft een lijst van ingrediënten namen[ <b>out</b> ] wanneer de actor een ingrediënt naam[ <b>in</b> ] selecteert dan geef het systeem de informatie weer: ingrediënt nummer[ <b>out</b> ], naam[ <b>out</b> ], houdbaarheidsdatum[ <b>out</b> ] en het aantal[ <b>out</b> ] van het ingrediënt in. Als de actor het ingrediënt besluit te verwijderen, dan haalt het systeem deze informatie van het ingrediënt weg.	
<b>Preconditie</b>	De voorraad van de ingrediënt is 0.	
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. Het systeem geeft de lijst met namen van bestaande ingrediënten weer.</li> <li>2. De actor geeft aan ingrediënt hij wil verwijderen.</li> <li>3. Het systeem geeft meer informatie over het ingrediënt weer (nummer, naam, aantal, houdbaarheidsdatum) en vraagt of het ingrediënt daadwerkelijk verwijderd moet worden.</li> <li>4. De actor geeft aan dat het ingrediënt verwijderd moet worden.</li> <li>5. Het systeem verwijdert het ingrediënt.</li> </ol>	
<b>Postconditie</b>	Het ingrediënt is uit het systeem verwijderd.	
<b>Alternatief scenario 1</b>	Trigger	Na stap 1 van het main scenario [Actor annuleert verwijderen van ingredient]
	Scenario	1. het systeem stopt de use case
<b>Alternatief scenario 2</b>	Trigger	Na stap 4 van het main scenario [Voorraad van ingredient is niet 0]
	Scenario	1. Het systeem geeft aan dat de ingrediënten niet verwijderd kan worden omdat er nog voorraad is.

		2. Het systeem gaat terug naar stap 1.
<b>Alternatief scenario 3</b>		
	Trigger	Na stap 3 van het main scenario [Actor annuleert verwijderen van ingrediënt]
	Scenario	1. Het systeem gaat terug naar stap 1.

## Use Case - Voorraad bestellen

<b>Use-case nummer</b>	UC9-bestel voorraad	
<b>Use-case naam</b>	Bestel voorraad	
<b>Actor</b>	Eigenaar	
<b>Samenvatting</b>	<p>Het systeem geeft de lijst met ingrediënten weer met naam[<b>out</b>] en hoeveelheid[<b>out</b>] en minimumvoorraad[<b>out</b>].</p> <p>De eigenaar vult voor ingrediënten die onder (optioneel ook boven) de minimum voorraad zijn de nieuwe gewenste voorraad in en het systeem genereert een bestellijst in de vorm van een excel bestand met de naam[<b>in</b>], type[<b>in</b>] van elk ingrediënt en de hoeveelheid[<b>in</b>].</p>	
<b>Preconditie</b>	-	
<b>Main scenario</b>	<ol style="list-style-type: none"> <li>1. Het systeem toont de lijst met ingrediënten met de naam, minimumvoorraad, hoeveelheid en mogelijkheid om de nieuwe hoeveelheid ervan in te voeren.</li> <li>2. A. De eigenaar vult de nieuwe hoeveelheden in. De eigenaar geeft aan hoe hij de bestellijst wil opslaan.</li> <li>3. Het systeem slaat de bestellijst op.</li> </ol>	
<b>Postconditie</b>	Er is een gegenereerde bestellijst opgeslagen.	
<b>Alternatief scenario 1</b>		
	Trigger	Na stap 1 van het main scenario [Actor annuleert het bestellen van voorraad]
	Scenario	1. het systeem stopt de use case
<b>Alternatief scenario 2</b>		
	Trigger	Na stap 2 van het main scenario [Actor vult invalide waarden in]
	Scenario	<ol style="list-style-type: none"> <li>1. Het systeem geeft aan dat de ingevulde waarden niet valide zijn.</li> <li>2. Het systeem gaat terug naar stap 1.</li> </ol>

--	--

## Bijlagen

### Volledigheid Analyse

Op basis van het toestandsdiagram:

Nr	Scenario	Vraag
1	Een gerecht is op voorraad, maar niet populair.	Wordt het gerecht dan gelijk onbestelbaar gemaakt?

Op basis van het Use Case Diagram:

Nr	Scenario	Vraag
1	Een gerecht is populair, in seizoen en op voorraad.	Mag de eigenaar het gerecht onbestelbaar maken? Wat zijn de gevolgen?

Op basis van de Use Case Templates:

Nr	Scenario	Vraag
1	Een klant bestelt een gerecht. De voorraad wordt automatisch door het systeem gewijzigd.	Is het niet logischer om het wijzigen van de voorraad aan de kok over te laten? Om fouten te voorkomen?

Attribuut	Bestel Gerecht	Maak gerecht beste lbaar	Maak gerecht niet bestelba ar	Voeg gerech t toe	Verwij der Gerech t	wijzig voorra ad	ingredi ent toevoe gen	ingredi ent verwijde ren	voorra ad bestell en	Conclu sie
Naam	OUT	OUT	OUT	IN	OUT	OUT	IN	IN/OUT	IN/OUT	OK
omschrijving	OUT	OUT	OUT	IN	OUT	-	-	-	-	OK
seizoen	-	OUT	OUT	IN	OUT	-	-	-	-	OK
prijs	OUT	OUT	OUT	IN	OUT	-	-	-	-	OK
ingrediënten	-	-	-	IN	-	-	-	-	-	BLACK HOLE
gerecht nummer	-	OUT	OUT	IN	OUT	-	-	-	-	OK
tijd	IN	-	-	-	-	-	-	-	-	BLACK HOLE
hoeveelheid	-	-	-	-	-	IN/OUT	-	-	IN/OUT	OK
houdbaar heidsdatum	-	-	-	-	-	OUT	IN	OUT	-	OK
minimumvo orraad	-	-	-	-	-	OUT	IN	-	OUT	OK
type	-	-	-	-	-	-	IN	-	IN	BLACK HOLE
aantal	-	-	-	-	-	-	IN	OUT	-	OK
ingrediënt nummer	-	-	-	-	-	-	IN	OUT	-	OK

NR	ISSUE	VRAAG
1	Bij use case ingredient toevoegen en voorraad bestellen wordt er een type ingevoerd maar die wordt vervolgens nooit meer opgehaald.	Is het dan niet beter om deze weg te laten?
2	bij use case bestel gerecht wordt er een tijd ingevoerd maar die wordt vervolgens nergens meer opgehaald.	Heeft het dan nog nut om deze op te slaan?



## Klassendiagram

