

FUNCTIONEEL ONTWERP



Voor order-systeem sushi restaurant

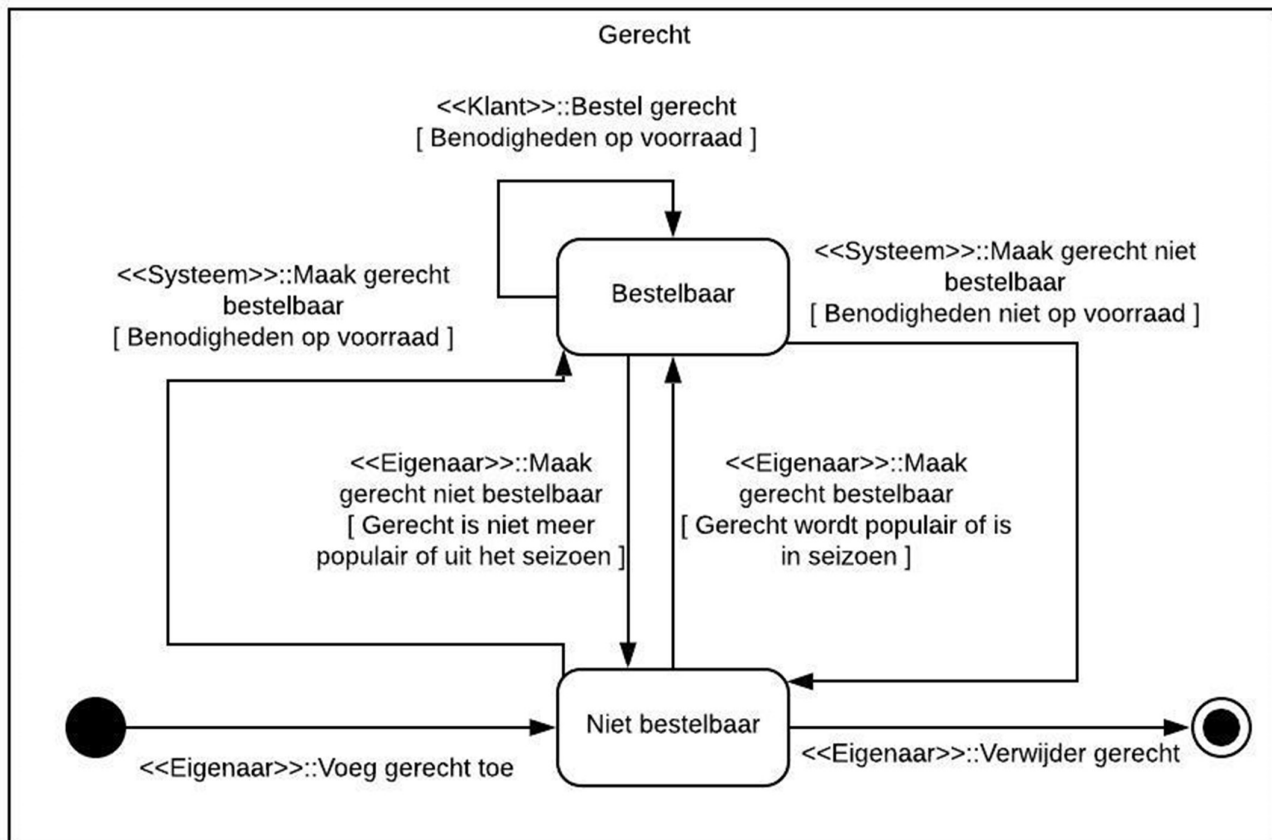
Contents

Inleiding	1
Toestandsdiagram	2
Use Case Diagram	3
Use Case Samenvattingen	4
Use Case - Gerecht Bestellen	4
Use Case - Gerecht Toevoegen	5
Use Case - Gerechten Verwijderen	6
Use Case - Gerecht Bestelbaar Maken	7
Use Case - Gerecht Niet Bestelbaar Maken	8
Bijlagen	9
Volledigheidsanalyse	9

Inleiding

Dit functioneel ontwerp gaat over de casus sushi restaurant.

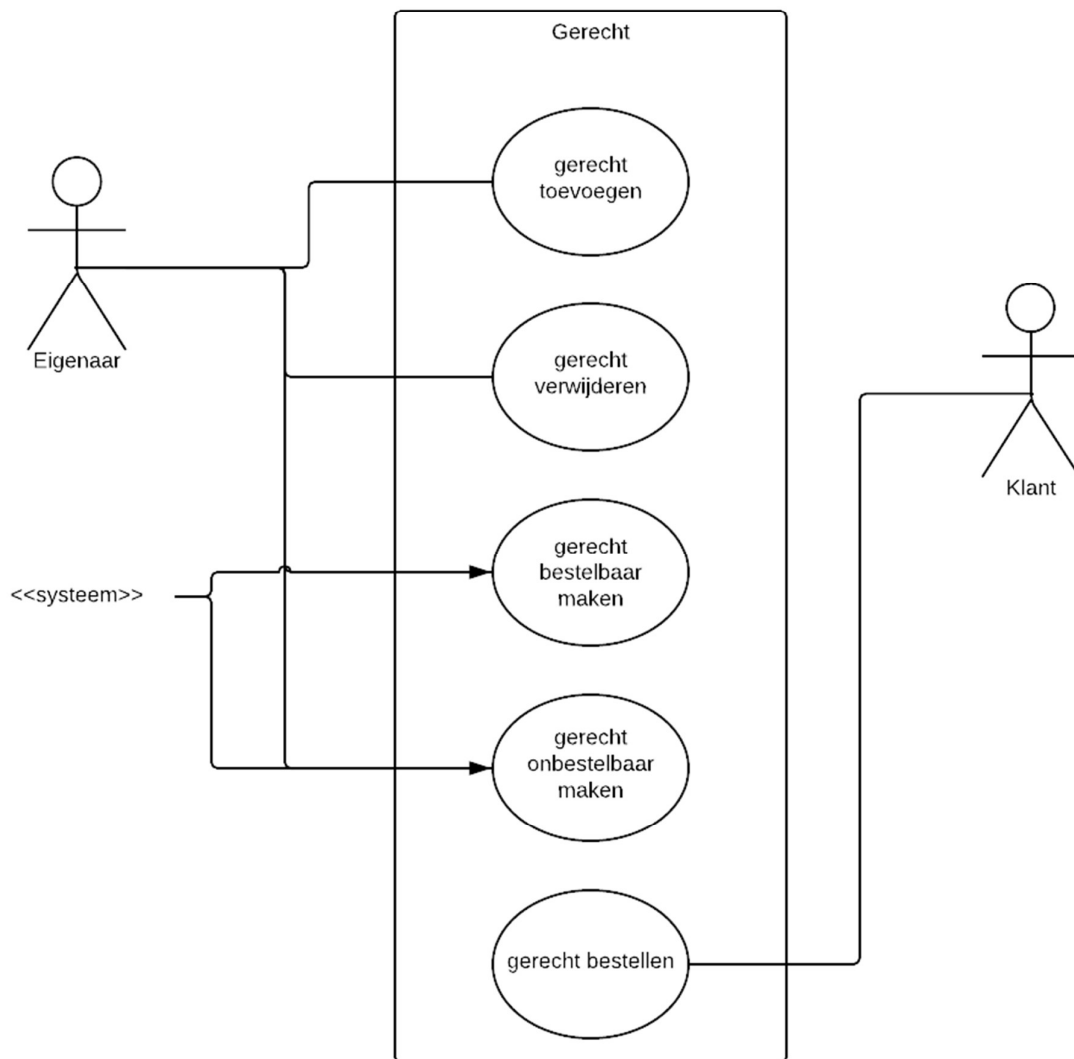
Toestandsdiagram



Use Case Diagram

Use Case Diagram gerecht

Arjan Rakké | March 8, 2019



Use Case Samenvattingen

gerecht toevoegen	De eigenaar voegt een nieuw gerecht toe aan het systeem en registreert welke ingrediënten er voor nodig zijn zodat de hoeveelheid ingrediënten bijgehouden kan worden.
gerecht verwijderen	De eigenaar verwijdert een gerecht, dus ook alle bijbehorende informatie uit het systeem.
gerecht bestelbaar maken	De eigenaar geeft in het systeem aan dat een gerecht besteld kan worden
gerecht onbestelbaar maken	De eigenaar geeft in het systeem aan dat een gerecht niet besteld kan worden.
gerecht bestellen	De klant selecteert via het systeem welke gerechten hij wilt bestellen en hoeveel hij ervan wilt.

Use Case - Gerecht Bestellen

ID	UC1-bestel_gerecht
Naam	Bestel Gerecht
Actor	Klant
Samenvatting	De klant selecteert via het systeem welke gerechten hij wilt bestellen en hoeveel hij ervan wilt.
Preconditie	De benodigdheden van het gerecht zijn op voorraad.
Main Scenario	<ol style="list-style-type: none"> 1. Systeem geeft alle bestelbare gerechten weer. 2. Actor kiest een gerecht. 3. Systeem koppelt de bestelling door naar de keuken. 4. Systeem laat aan de actor weten dat de bestelling succesvol is doorgesluisd. 5. Systeem vermindert de voorraden van alle gebruikte ingrediënten in het gerecht.
Postconditie	Actor heeft bestelling gemaakt.

Use Case - Gerecht Toevoegen

Use-case nummer	UC1-gerecht-toevoegen
Use-case naam	Voeg nieuw gerecht toe
Actor	Eigenaar
Samenvatting	De eigenaar voegt een nieuw gerecht toe aan het systeem en registreert welke ingrediënten er voor nodig zijn zodat de hoeveelheid ingrediënten bijgehouden kan worden.
Pre-conditie	-
Main scenario	<ol style="list-style-type: none">1. Het systeem vraagt om een naam, ingrediënten die bij het gerecht horen en een omschrijving van het gerecht.2. Actor verstrekt deze informatie.3. Het systeem registreert het gerecht en toont een melding dat het gerecht is geregistreerd.
Post-conditie	Een nieuw gerecht is geregistreerd.

Use Case - Gerechten Verwijderen

Use-case nummer	UC2-gerecht-verwijderen	
Use-case naam	Verwijder gerecht	
Actor	Eigenaar	
Samenvatting	De eigenaar verwijdert een gerecht, dus ook alle bijbehorende informatie uit het systeem	
Pre-conditie	-	
Main scenario	<ol style="list-style-type: none"> 1. Het systeem toont een lijst met bestaande gerechten. 2. De actor selecteert één of meer gerechten om te verwijderen. 3. Het systeem vraagt of de gerechten echt verwijderd moeten worden. 4. De actor keurt het verwijderen van de gerechten goed. 5. Het systeem verwijdert de gerechten. 	
Alternatief scenario 1	Trigger	Na stap 3 van het main scenario [De actor keurt het verwijderen van het gerecht.]
	Stappen	1. Het systeem gaat terug naar stap 1.
Alternatief scenario 2	Trigger	Na stap 1 van het main scenario [Acteur cancelt scherm van verwijderen van gerechten.].
	Stappen	1. Use case wordt beëindigd.

	Post-condite	-
Post-conditie	Één of meer rechten zijn uit het systeem verwijderd.	

Use Case - Gerecht Bestelbaar Maken

ID	UC4-maak_gerecht_bestelbaar
Naam	Maak Gerecht Bestelbaar
Actor	Systeem, Eigenaar
Samenvatting	De eigenaar geeft in het systeem aan dat een gerecht besteld kan worden.
Preconditie	Gerecht is niet bestelbaar. Gerecht is op voorraad.
Main Scenario	<ol style="list-style-type: none"> 1. Systeem haalt een lijst met alle niet bestelbare rechten op. 2. Actor bepaalt of een gerecht bestelbaar moet worden, op basis van populariteit en of het gerecht in seizoen is. 3. Systeem wijzigt de status van het gerecht naar "Bestelbaar".
Alternatief Scenario	<ol style="list-style-type: none"> 1. Systeem haalt een lijst met alle niet bestelbare rechten op. 2. Systeem checkt voor elk gerecht wat de voorraad is. 3. Systeem wijzigt de statussen van de rechten die op voorraad zijn naar "Bestelbaar".
Postconditie	Gerecht is bestelbaar.

Use Case - Gerecht Niet Bestelbaar Maken

ID	UC5-maak_gerecht_niet_bestelbaar
Naam	Maak Gerecht Niet Bestelbaar
Actor	Systeem, Actor
Samenvatting	De eigenaar geeft in het systeem aan dat een gerecht niet besteld kan worden.
Preconditie	Gerecht is bestelbaar.
Main Scenario	<ol style="list-style-type: none">1. Systeem haalt een lijst met alle bestelbare gerechten op.2. Actor bepaalt of een gerecht bestelbaar moet blijven, op basis van de populariteit en of het gerecht in seizoen is.3. Systeem wijzigt de status van het gerecht naar "Niet bestelbaar".
Alternatief Scenario	<ol style="list-style-type: none">1. Systeem haalt een lijst met alle bestelbare gerechten op.2. Systeem checkt voor elk gerecht wat de voorraad is.3. Systeem wijzigt de statussen van de gerechten die niet meer op voorraad zijn naar "Niet bestelbaar".
Postconditie	Gerecht is niet bestelbaar.

Bijlagen

Volledigheidsanalyse

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 bestelbaar	Maak gerecht niet bestelbaar	Voeg gerecht toe	Verwijder Gerecht	Conclusie
Naam	IN	IN	IN	IN	IN	OK
Bestelbaarheid	IN	IN/OUT	IN/OUT			OK
Benodigdheden	OUT	IN	IN	IN/OUT	IN/OUT	OK

Gerecht
<ul style="list-style-type: none"> - naam : String - bestelbaar : boolean - benodigdheden : ArrayList<Ingredient>
<ul style="list-style-type: none"> + Gerecht(naam : String, bestelbaar : boolean) + getNaam() : String + setBestelbaarheid(bestelbaar : boolean) : void + genoegVoorraad(benodigdheden : ArrayList<Ingredient>) : boolean + bestelGerecht(naam : String) : void