

# Analysedocument

Naam: Stijn Versluis

Klas: DB06T

Docent: Klaas Kox, Richard Stam

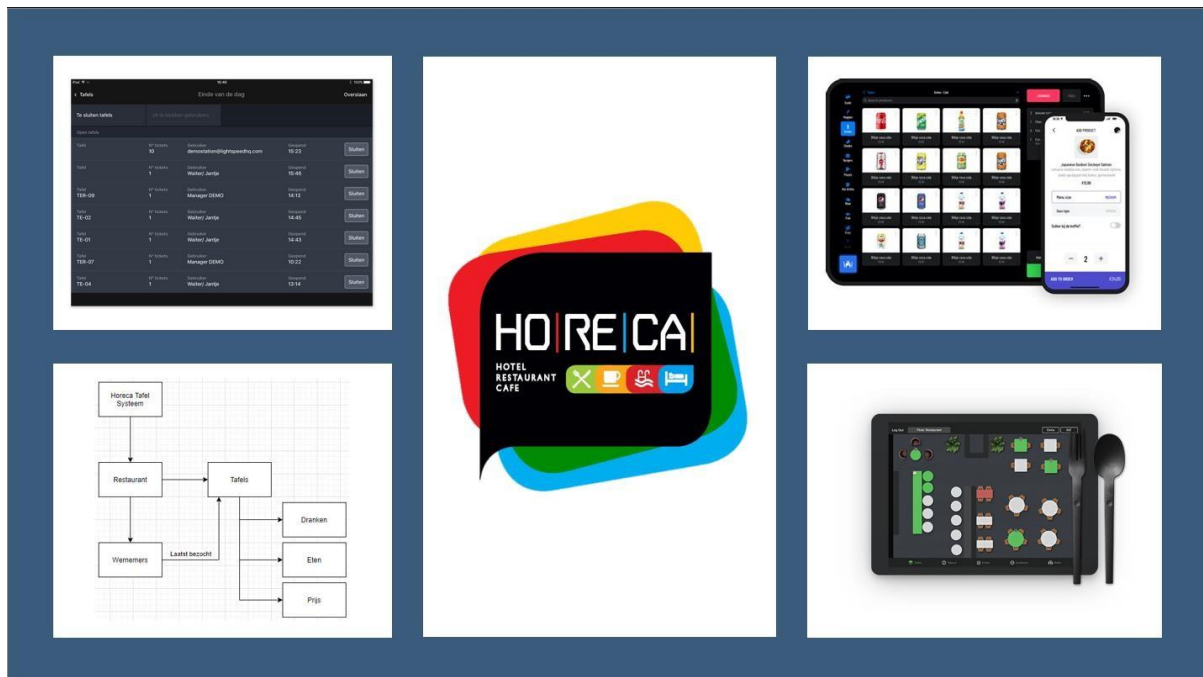
Semester coach: Manon Penning

## Contents

Introductie .....	3
One Slide idee.....	3
Context .....	3
Sprint Vormgeving.....	4
Sprint 1:.....	4
Sprint 2:.....	4
Sprint 3:.....	4
Requirements.....	5
Moscow methode .....	5
Functionele Requirements .....	6
Use-case.....	8
Use-cases Sprint 2 .....	8
Wireframes .....	12
Hoofdpagina:.....	12
Login pagina: .....	12
Het openen van een tafel: .....	13
Nieuwe tafel is toegevoegd aan te lijst: .....	13
Tafel Detail pagina: .....	14
Order aanmaken:.....	15
Diagrammen .....	16
EER-diagram .....	16
ER-diagram (DB-model) .....	17
Use-case diagram.....	18
Klassendiagram.....	19
Table: .....	19
Product:.....	20
Order:.....	21
Staff: .....	22
Testen.....	23
Test Data.....	23
Test Cases.....	24
Code.....	26

# Introductie

## One Slide idee



## Context

Het systeem wat gebouwd gaat worden is voor het bijhouden van tafels in een restaurant. Het restaurant heeft meerdere tafel waar gasten aan zitten. Deze tafels kunnen eten/drinken bestellen, dit kunnen ze uiteindelijk ook afrekenen. De tafels kunnen vaker bezet worden en vorige tafels moeten kunnen worden bekeken.

# Sprint Vormgeving

## Sprint 1:

Sprint 1 ben ik begonnen met het aanmaken van een One-slide idee en het Analysedocument. In dit document heb ik een het One-slide idee toegevoegd en een introductie geschreven. Het analysedocument heb ik opgemaakt met een voorpagina en inhoudsopgave.

Na het aanmaken van het analysedocument ben ik begonnen met het maken van Requirements. Ik had twaalf functionele Requirements opgesteld en hier ook beperkingen en kwaliteitseisen bij gevoegd.

Na deze opgesteld te hebben had ik een systeem in mijn hoofd wat ik meteen in Adobe XD heb gezet.

Hier kwamen al snel wireframes uit.

Na de wireframes had ik best veel tijd over en ben ik aan een aantal diagrammen gaan werken (EER, ERD, Use-Case en de klasse-diagrammen).

Hierna heb ik een project in visual studio aangemaakt.

## Sprint 2:

In overleg met klaas heb ik mijn Requirements in een andere volgorde in de MoSCoW gezet. En de Requirements wat beter onderhoudbaar gemaakt. Met nog twee extra Requirements toegevoegd.

Hierna ben ik mijn test-data gaan schrijven.

Nadat ik mijn testcases had gemaakt ben ik begonnen aan de code. Ik ben meteen in solid begonnen en heb alle DAL bestanden aangemaakt. Hier heb ik de functionaliteiten toegevoegd, maar nog niet verbonden met een Database.

Verder heb ik de link tussen BusinessLogic, in terface en DataLayer gelegd.

Daarna ben ik met het inloggen bezig geweest en heb ik wat aan het front-end gewerkt. Hierna heb ik een database aangemaakt en gekoppeld aan het systeem.

## Sprint 3:

Ik heb deze sprint grotendeels aan de code gewerkt en het systeem op orde gemaakt. Daarnaast heb ik de klasse-diagrammen wel aangepast met nieuwe functies die ik nodig heb. De aanpassingen die ik heb gemaakt zijn grotendeels front-end dingen. Zo heb ik een aantal knoppen aangepast en het wat opgeschoond door een aantal onnodige cards weg te halen zodra er niks in staat.

## Sprint 4:

De sprint heeft bij mij vooral gestaan voor het afmaken van het product (zonder administrator functies).

Ik heb de layout wat mooier gemaakt en wat duidelijker. Daarnaast heb ik sommige functionaliteiten een beetje veranderd. Ik kreeg vaak errors over het toevoegen van producten aan een order of het aanmaken van een order. Dit kwam waarschijnlijk omdat ik te veel Async functies aan het doen was in de vorm van Fetch. Verder met Klaas en Richard besproken: meer feedback vragen (elke week 2x) en een logboek bijhouden van wat ik ga doen en wat ik heb gedaan die dag.

# Requirements

## Moscow methode

Must	De gebruiker kan inloggen
Must	De gebruiker moet een tafel kunnen openen
Must	De gebruiker kan een order toevoegen aan een tafel
Must	De gebruiker kan producten toevoegen aan een order
Must	De gebruiker moet een product van een order af kunnen halen
Must	De gebruiker kan een order definitief maken
Must	De gebruiker moet een productenoverzicht kunnen zien van een tafel

Should	Een tafel moet kunnen worden afgeslagen/betaald
Should	Er is een administrator account
Should	Administrator kan producten aanmaken
Should	Administrator kan producten aanpassen
Should	Administrator kan producten verwijderen
Should	Administrator kan gebruikers toevoegen
Should	Administrator kan gebruikers aanpassen
Should	Administrator kan gebruikers verwijderen

Could	Een Administrator kan tafels toevoegen
Could	Een Administrator kan tafels verwijderen

Won't	Een gebruiker kan producten toevoegen
Won't	Een gebruiker kan producten verwijderen
Won't	Een gebruiker kan tafels toevoegen
Won't	Een gebruiker kan tafels verwijderen

## Functionele Requirements

FR-01	De gebruiker moet kunnen inloggen	M
B-01.1	Voor het inloggen gebruikt de gebruiker zijn naam en code	
B-01.2	Er kan alleen ingelogd worden wanneer er een account bestaat met de ingevulde gegevens	

FR-02	De gebruiker kan een order toevoegen aan een tafel	M
B-02.1	De gebruiker heeft een tafel geselecteerd	
K-02.1	Er is een overzicht van producten die besteld zijn (FR-05)	
K-02.2	Er is een overzicht van producten die besteld kunnen worden	

FR-03	De gebruiker kan producten aan een order toevoegen	M
B-03.1	De gebruiker heeft een tafel geselecteerd	
B-03.2	Er moet een order open staan/zijn toegevoegd (FR-02) op de geselecteerde tafel	
K-03.1	De gebruiker kan een product selecteren	
K-03.2	Er is een overzicht van producten die aan de order zijn toegevoegd	
K-03.3	Indien het product al aan de order is toegevoegd verander het aantal van dit product +1.	

FR-04	De gebruiker kan een product van een order afhalen	M
B-04.1	De gebruiker heeft een tafel geselecteerd	
B-04.2	Er moet een order open staan/zijn toegevoegd (FR-02) op de geselecteerde tafel	
K-04.1	De gebruiker kan een product selecteren uit de order lijst	
K-04.2	Er is een overzicht van producten die aan de order zijn toegevoegd	

FR-05	De gebruiker kan een order definitief maken	M
B-05.1	De gebruiker heeft een tafel geselecteerd	
K-05.2	De gebruiker selecteert een van de producten die op de lijst van bestelde producten staat	

FR-06	De gebruiker moet een productenoverzicht kunnen zien van een tafel	M
K-06.1	Er is een overzicht van de producten die besteld zijn	
B-06.1	De gebruiker moet een tafel geselecteerd hebben	
B-06.2	De tafel moet producten hebben	

FR-07	Een tafel moet kunnen betalen/worden afgeslagen	M
K-07.1	Er is een overzicht van de totale prijs van de producten	
B-07.1	De gebruiker moet een tafel geselecteerd hebben	
B-07.2	Er kan geen order meer open staan.	

FR-08	Een administrator kan gebruikers toevoegen	M
-------	--	---

B-08.1	De administrator gebruikt de naam van de gebruiker en een willekeurige code	
B-08.2	De administrator kan geen ongeldige tekens gebruiken	

FR-09	Een administrator kan gebruikers aanpassen	M
B-09.1	De administrator gebruikt de naam van de gebruiker en een willekeurige code	
B-09.2	De administrator kan geen ongeldige tekens gebruiken	

FR-10	Een administrator kan gebruikers verwijderen	M
B-10.1	De administrator moet een gebruiker geselecteerd hebben	
B-10.2	Een administrator kan zichzelf niet verwijderen	

FR-11	Een administrator kan producten toevoegen	M
B-11.1	De administrator gebruikt de naam van het product; het type product en de prijs	

FR-12	Een administrator kan producten aanpassen	M
B-12.1	De administrator kan geen ongeldige tekens gebruiken	

FR-13	Een administrator kan producten verwijderen	M
B-13.1	De administrator moet een product geselecteerd hebben	

FR-14	De gebruiker moet een tafel kunnen openen	M
B-14.1	De tafel moet gesloten zijn (niet al open staan).	

# Use-case

## Use-cases Sprint 2

<b>Naam:</b>	<b>UC-01: Inloggen (FR-01)</b>
Samenvatting:	De actor logt in zijn account in.
Actors	Gebruiker
Aannamen:	De gebruiker heeft een account
Scenario:	<ol style="list-style-type: none"><li>1. De actor wil inloggen.</li><li>2. Het systeem laat de inlog pagina zien.</li><li>3. De actor vult zijn/haar gegevens in.</li><li>4. Het systeem logt de actor in. [1][2]</li></ol>
Uitzondering:	<ol style="list-style-type: none"><li>1. De actor heeft niet alle gegevens correct ingevuld.</li><li>2. De actor heeft geen account.</li></ol>
Resultaat:	De actor heeft succesvol ingelogd.

<b>Naam:</b>	<b>UC-02: De gebruiker moet een tafel kunnen openen (FR-14)</b>
Samenvatting:	De actor opent een tafel wanneer gasten aan tafel gaan zitten.
Actors	Gebruiker
Aannamen:	De gebruiker is ingelogd (UC-01)
Scenario:	<ol style="list-style-type: none"><li>1. De actor selecteert een tafel.</li><li>2. Het systeem laat de details zien van de opening. [1]</li><li>3. De actor accepteert de opening van de tafel.</li><li>4. Het systeem zet de tafel open.</li></ol>
Uitzondering:	<ol style="list-style-type: none"><li>1. De tafel staat al open</li></ol>
Resultaat:	De actor heeft succesvol ingelogd.

<b>Naam:</b>	<b>UC-03: De gebruiker kan een productenoverzicht van een tafel inzien (FR-06)</b>
Samenvatting:	De gebruiker ziet de bestelde items van een tafel
Actors	Gebruiker
Aannamen:	De gebruiker is ingelogd (UC-01), een tafel is geopend (UC-02)
Scenario:	<ol style="list-style-type: none"><li>1. De actor selecteert een tafel.</li><li>2. Het systeem laat de tafel detail pagina zien met producten overzicht. [1]</li></ol>
Uitzondering:	<ol style="list-style-type: none"><li>1. Er zijn geen producten toegevoegd.</li></ol>
Resultaat:	De actor kan alle bestelde producten inzien

<b>Naam:</b>	<b>UC-04: De gebruiker kan een order aan een tafel toevoegen (FR-02)</b>
Samenvatting:	De actor voegt een order toe aan een tafel
Actors	Gebruiker
Aannamen:	De gebruiker is ingelogd (UC-01), een tafel is geopend (UC-02)
Scenario:	<ol style="list-style-type: none"><li>1. De actor geeft aan een order aan te willen maken. [1]</li><li>2. Het systeem voegt de order toe aan de tafel en laat deze zien.</li></ol>



Uitzondering:	1. De gebruiker heeft een tafel geselecteerd.
Resultaat:	De actor heeft een order aan een tafel toegevoegd

<b>Naam:</b>	<b>UC-05: De gebruiker kan een product aan een order toevoegen (FR-03)</b>
Samenvatting:	De actor voegt een product toe aan een order
Actors	Gebruiker
Aannamen:	De gebruiker is ingelogd (UC-01), de gebruiker heeft een order aan een tafel toegevoegd (UC-03).
Scenario:	<ol style="list-style-type: none"> <li>1. De actor geeft aan een product te willen toevoegen aan een order.</li> <li>2. Het systeem voegt het product toe aan de order en laat deze zien. [1][2]</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De gebruiker heeft geen tafel geselecteerd.</li> <li>2. Er is geen order geopend.</li> </ol>
Resultaat:	De actor heeft een product toegevoegd aan de order.

<b>Naam:</b>	<b>UC-06: De gebruiker kan een order definitief maken (FR-05)</b>
Samenvatting:	Een order die op een tafel staat kan definitief worden gemaakt. (gesloten worden en niet kunnen worden aangepast)
Actors	Gebruiker
Aannamen:	UC-01, UC-04 en UC-05 zijn uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor geeft aan de geopende order op te willen slaan.</li> <li>2. Het systeem slaat de order op [1][2]</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De gebruiker heeft geen tafel geselecteerd.</li> <li>2. De tafel heeft geen order open staan.</li> </ol>
Resultaat:	De order is opgeslagen en staat nu in de lijst van orders. Er is geen order meer die open is.

### Use-Cases Sprint 3:

<b>Naam:</b>	<b>UC-07: De gebruiker kan een product van een order afhalen (FR-04)</b>
Samenvatting:	Een product dat aan een order is toegevoegd kan worden verwijderd (als deze fout is toegevoegd bijv.).
Actors	Gebruiker
Aannamen:	UC-01, UC-02 en UC-05 zijn uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor geeft aan een product van de order te verwijderen.</li> <li>2. Het systeem verwijdert het product van de order. [1][2]</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. Er is geen order geopend.</li> <li>2. Er is geen product toegevoegd aan de open order.</li> </ol>
Resultaat:	Het aantal producten is met 1 naar beneden gegaan of is van de lijst wegegehaald.

<b>Naam:</b>	<b>UC-08: Een tafel moet kunnen worden afgeslagen (FR-07)</b>
Samenvatting:	Wanneer een tafel betaald heeft kan deze worden afgeslagen zodat de tafel opnieuw kan worden geopend.
Actors	Gebruiker
Aannamen:	UC-01, UC-02 zijn uitgevoerd
Scenario:	<ol style="list-style-type: none"> <li>1. De actor geeft aan de tafel te willen sluiten.</li> <li>2. Het systeem sluit de tafel. [1][2]</li> </ol>

Uitzondering:	<ol style="list-style-type: none"> <li>1. De tafel is geopend.</li> <li>2. Er is geen order die open staat.</li> </ol>
Resultaat:	

## Use-Cases Sprint 4 en 5 CRUD Gebruikers, Producten en Tafels:

<b>Naam:</b>	<b>UC-09: De administrator kan gebruikers toevoegen (FR-08)</b>
Samenvatting:	De administrator kan gebruikers toevoegen aan het systeem.
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd
Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de gebruikers pagina.[2]</li> <li>2. Het systeem laat een lijst zien van gebruikers.</li> <li>3. De actor geeft aan een nieuwe gebruiker te willen aanmaken.</li> <li>4. Het systeem geeft een formulier weer.</li> <li>5. De actor vult de gegevens van de nieuwe gebruiker in.</li> <li>6. Het systeem voegt de gebruiker toe.[1]</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De ingevulde gegevens zijn niet juist.</li> <li>2. De actor is geen administrator</li> </ol>
Resultaat:	Er is een nieuwe gebruiker aangemaakt met deze gegevens.

<b>Naam:</b>	<b>UC-10: De administrator kan gebruikers aanpassen (FR-09)</b>
Samenvatting:	De administrator kan bestaande gebruikers gegevens aanpassen.
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de gebruikers pagina. [2]</li> <li>2. Het systeem laat een lijst zien van gebruikers.</li> <li>3. De actor geeft aan een bestaande gebruiker te willen aanpassen.</li> <li>4. Het systeem geeft een formulier weer met de gegevens van de gebruiker.</li> <li>5. De actor past de gegevens van de gebruiker aan.</li> <li>6. Het systeem past de gebruiker aan.[1]</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De aangepaste gegevens zijn niet juist.</li> <li>2. De gebruiker is geen administrator.</li> </ol>
Resultaat:	De gebruikers gegevens zijn aangepast.

<b>Naam:</b>	<b>UC-11: de administrator kan gebruikers verwijderen (FR-10)</b>
Samenvatting:	De administrator kan bestaande gebruikers verwijderen
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de gebruikers pagina. [1]</li> <li>2. Het systeem laat een lijst zien van gebruikers.</li> <li>3. De actor selecteert de gebruiker die de actor wil verwijderen.</li> <li>4. Het systeem verwijdert de gebruiker.</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De actor is geen administrator.</li> </ol>
Resultaat:	De gebruiker is verwijderd en staat niet meer in de lijst van gebruikers.

<b>Naam:</b>	<b>UC-12: De administrator kan producten toevoegen (FR-11)</b>
Samenvatting:	De administrator kan producten toevoegen aan het systeem
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd.

Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de producten pagina. [1]</li> <li>2. Het systeem laat een lijst zien van producten.</li> <li>3. De actor geeft aan een nieuw product te willen toevoegen.</li> <li>4. Het systeem laat een formulier zien.</li> <li>5. De actor vult de gegevens in van het nieuwe product.</li> <li>6. Het systeem voegt het product toe.</li> </ol>
Uitzondering:	1. De actor is geen administrator
Resultaat:	Het product is toegevoegd aan het systeem.

<b>Naam:</b>	<b>UC-13: De administrator kan een product aanpassen (FR-12)</b>
Samenvatting:	De administrator kan een bestaand product aanpassen
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de producten pagina. [1]</li> <li>2. Het systeem laat een lijst zien van producten.</li> <li>3. De actor selecteert een product.</li> <li>4. Het systeem geeft een formulier weer met de gegevens van het product.</li> <li>5. De actor past de gegevens van het product aan.</li> <li>6. Het systeem past het product aan.</li> </ol>
Uitzondering:	<ol style="list-style-type: none"> <li>1. De actor is geen administrator</li> <li>2. De aangepaste gegevens zijn niet juist.</li> </ol>
Resultaat:	Het product is aangepast.


<b>Naam:</b>	<b>UC-14: De administrator kan producten verwijderen (FR-13)</b>
Samenvatting:	De administrator kan bestaande producten verwijderen uit het systeem.
Actors	Administrator
Aannamen:	<b>UC-01</b> is uitgevoerd.
Scenario:	<ol style="list-style-type: none"> <li>1. De actor navigeert naar de producten pagina. [1]</li> <li>2. Het systeem laat een lijst zien van producten.</li> <li>3. De actor selecteert het product die de actor wil verwijderen.</li> <li>4. Het systeem verwijdert het product.</li> </ol>
Uitzondering:	1. De actor is geen administrator.
Resultaat:	Het product is verwijderd.

# Wireframes

## Hoofdpagina:



Table nr.	Server - last opened	Price	Placed
1	Kimberley - 12:52	50,-	12:45

Table nr.	10	
	10	
	101	
	102	
	103	
	104	

## Login pagina:

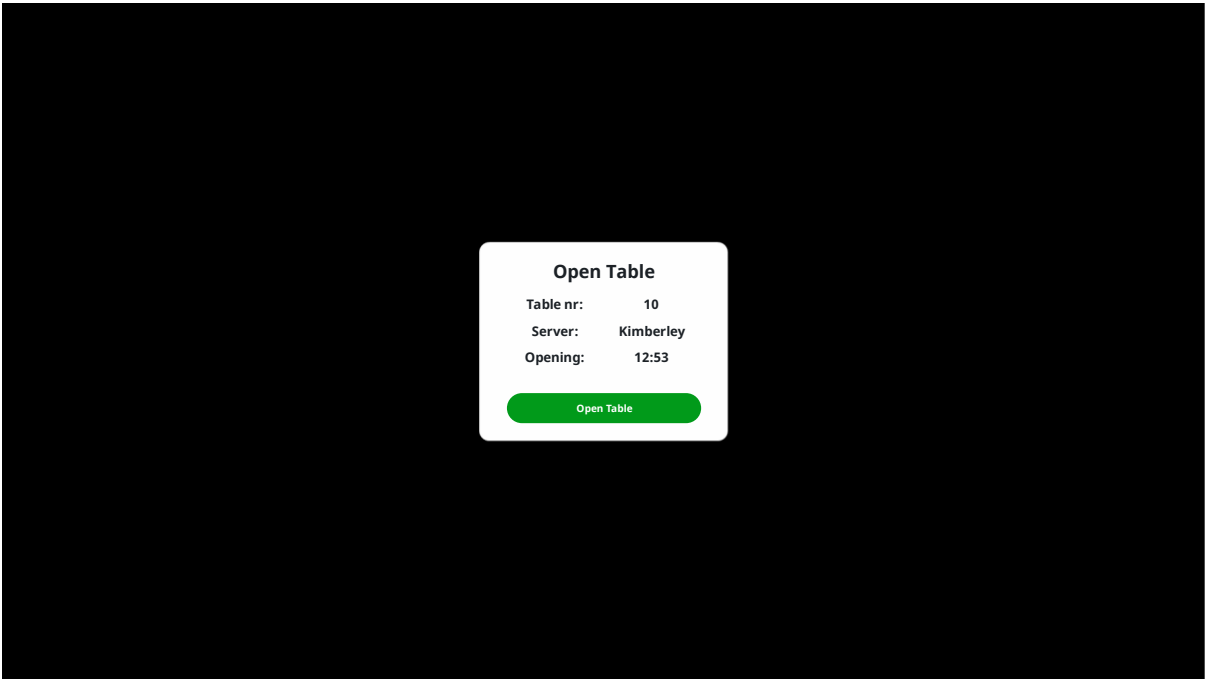
Login

Name

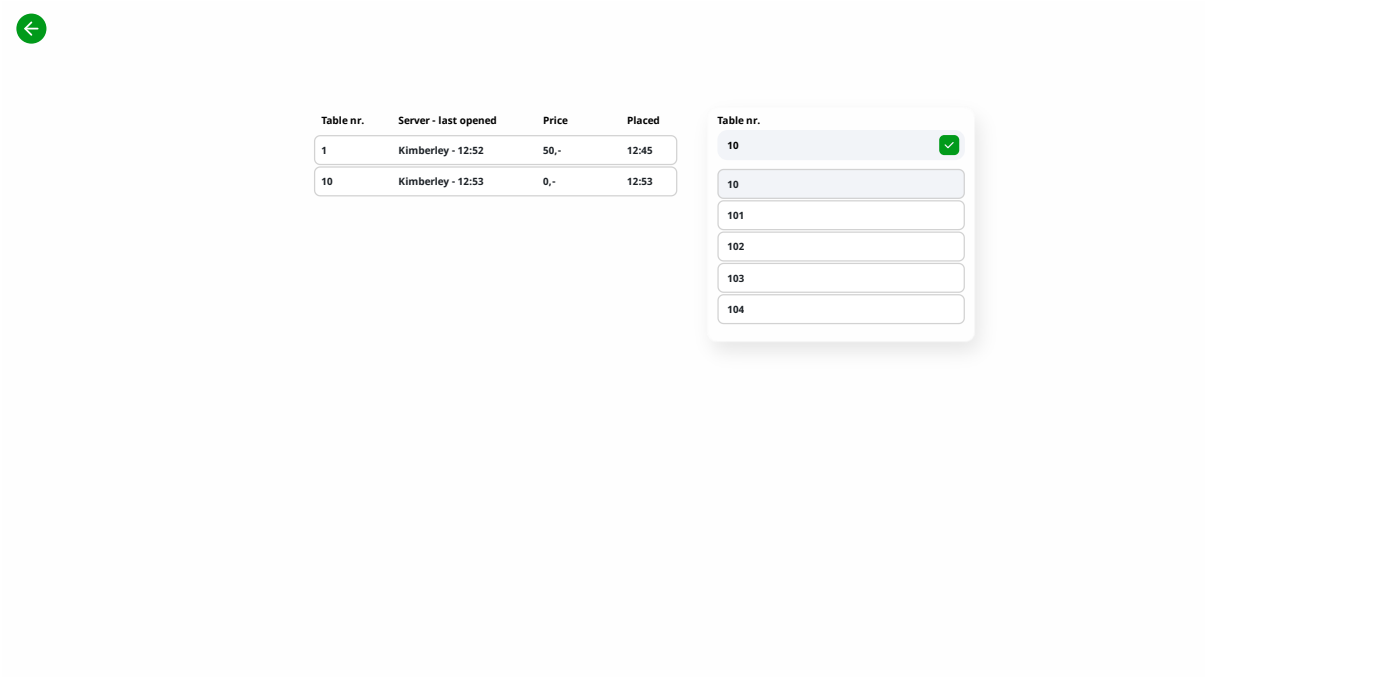
Code

Login

# Het openen van een tafel:



# Nieuwe tafel is toegevoegd aan te lijst:



## Tafel Detail pagina:



Table nr.	1
<b>Drinks</b>	
Coca Cola 2,80	2
Coca Cola Light 2,80	1
<b>Food</b>	
Bruger 14,-	1
Sandwich 12,-	2
<b>Extra</b>	
Fresh Salad 3,80	1
<b>Price:</b> 56,20	

Create Order

### Drinks

Coca Cola 2,80	+
Coca Cola Light 2,80	+
Coca Cola Zero 2,80	+
Fanta 2,80	+
Ice Tea 2,80	+

### Food

Full English BrFt 17,-	+
12 hour 12,50	+
Burger 14,-	+
Sandwich 12,-	+
Steak 18,-	+

### Extra's

French Fries 4,80	+
Patats 4,80	+
Fresh Salad 3,80	+
Karaf Water Free	+



Table nr.	10
<b>Drinks</b>	
Coca Cola 2,80	2
Coca Cola Light 2,80	1
Coca Cola Zero 2,80	1
Fanta 2,80	1
Ice Tea 2,80	1
<b>Food</b>	
Full English BrFt 17,-	1
12 hour 12,50	1
Burger 14,-	1
Sandwich 12,-	1
Steak 18,-	1
<b>Extra</b>	
French Fries 4,80	1
Patats 4,80	1
Fresh Salad 3,80	1
Karaf Water Free	1
<b>Price:</b> 0,-	

Create Order

### Drinks

Coca Cola 2,80	+
Coca Cola Light 2,80	+
Coca Cola Zero 2,80	+
Fanta 2,80	+
Ice Tea 2,80	+

### Food

Full English BrFt 17,-	+
12 hour 12,50	+
Burger 14,-	+
Sandwich 12,-	+
Steak 18,-	+

### Extra's

French Fries 4,80	+
Patats 4,80	+
Fresh Salad 3,80	+
Karaf Water Free	+

# Order aanmaken:

←

Table nr.1

Drinks

Coca Cola2,802

Coca Cola Light2,801

Food

Bruger14,-1

Sandwich12,-2

Extra

Fresh Salad3,801

Price:56,20

Order

Drinks

Coca Cola2,802

Coca Cola Light2,801

Save

Drinks

Coca Cola2,80+

Coca Cola Light2,80+

Coca Cola Zero2,80+

Fanta2,80+

Ice Tea2,80+

Food

Full English BrFt17,-+

12 hour12,50+

Burger14,-+

Sandwich12,-+

Steak18,-+

Extra's

French Fries4,80+

Patats4,80+

Fresh Salad3,80+

Karaf WaterFree+

←

Table nr.10

Drinks

Coca Cola2,802

Coca Cola Light2,801

Save

Drinks

Coca Cola2,80+

Coca Cola Light2,80+

Coca Cola Zero2,80+

Fanta2,80+

Ice Tea2,80+

Food

Full English BrFt17,-+

12 hour12,50+

Burger14,-+

Sandwich12,-+

Steak18,-+

Extra's

French Fries4,80+

Patats4,80+

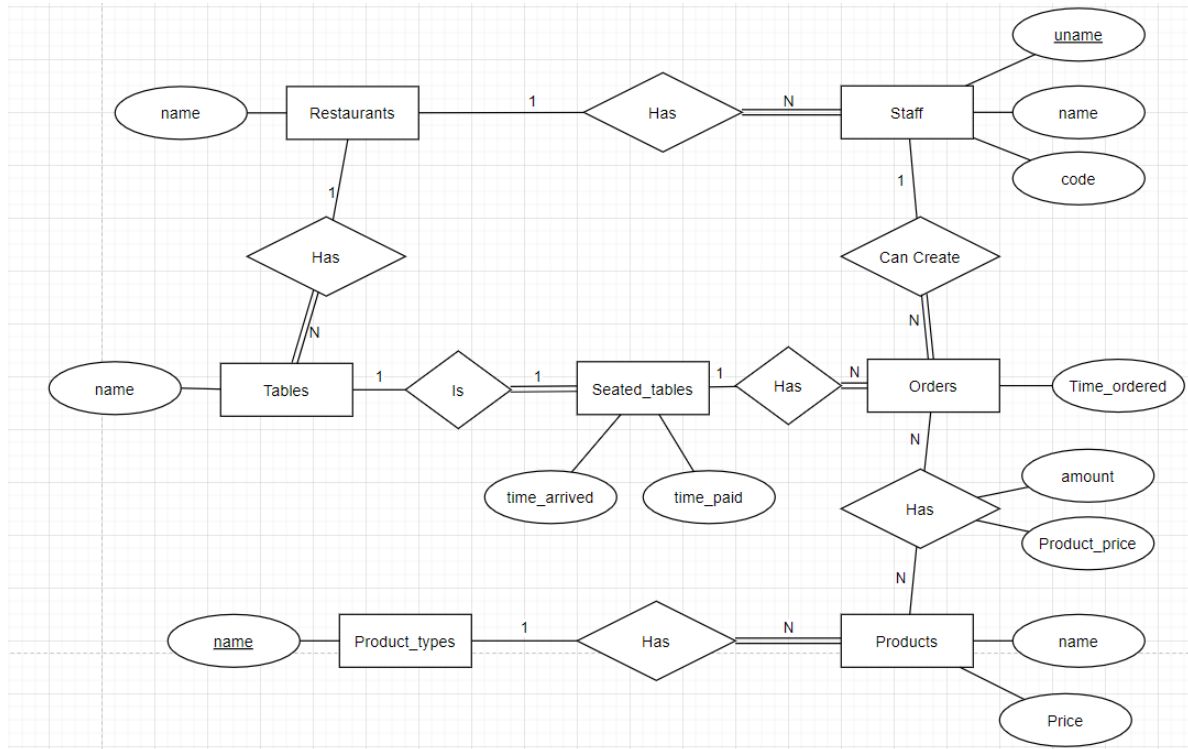
Fresh Salad3,80+

Karaf WaterFree+

Price:0,-

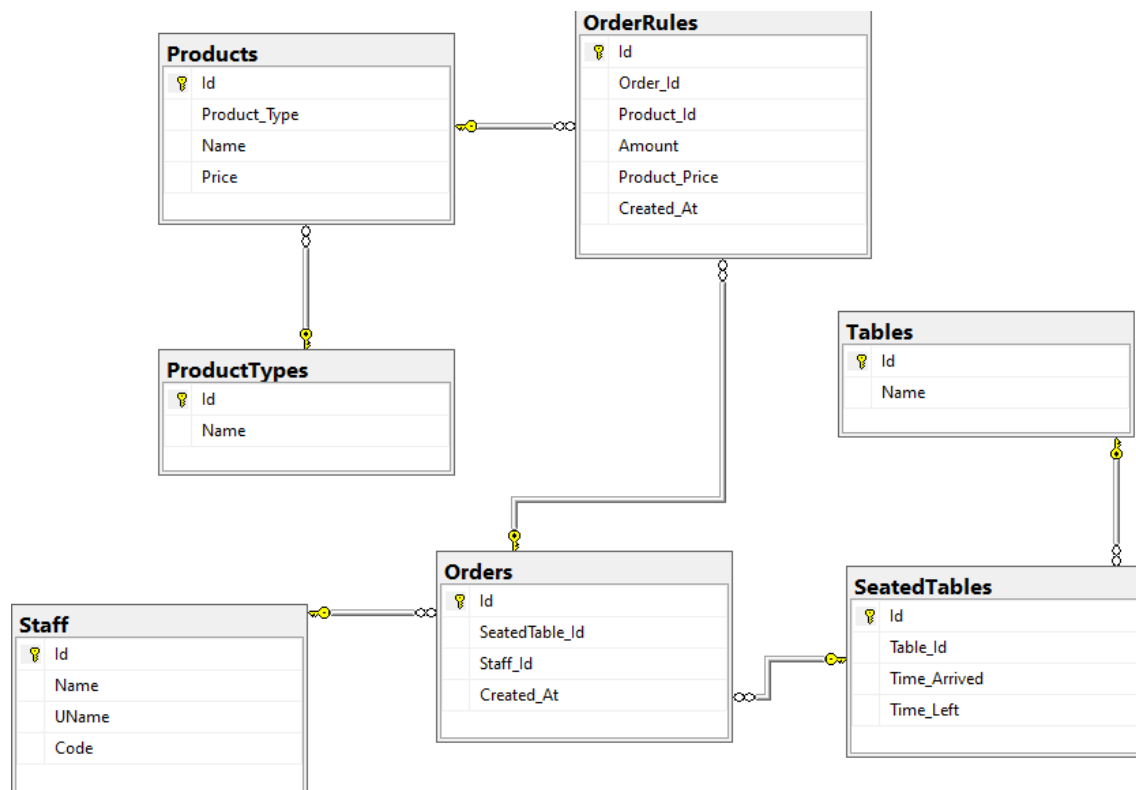
# Diagrammen

## EER-diagram

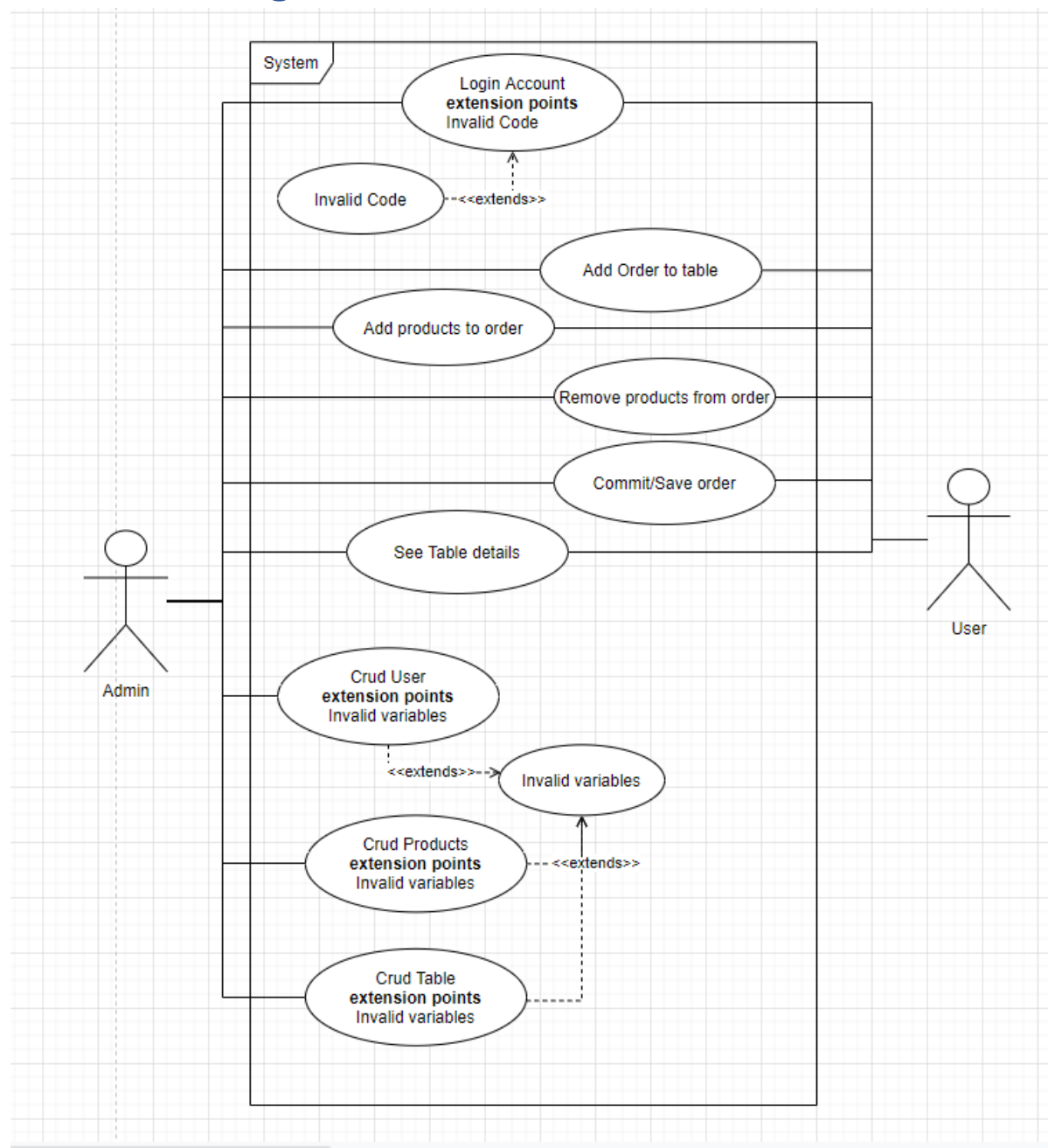




## ER-diagram (DB-model)

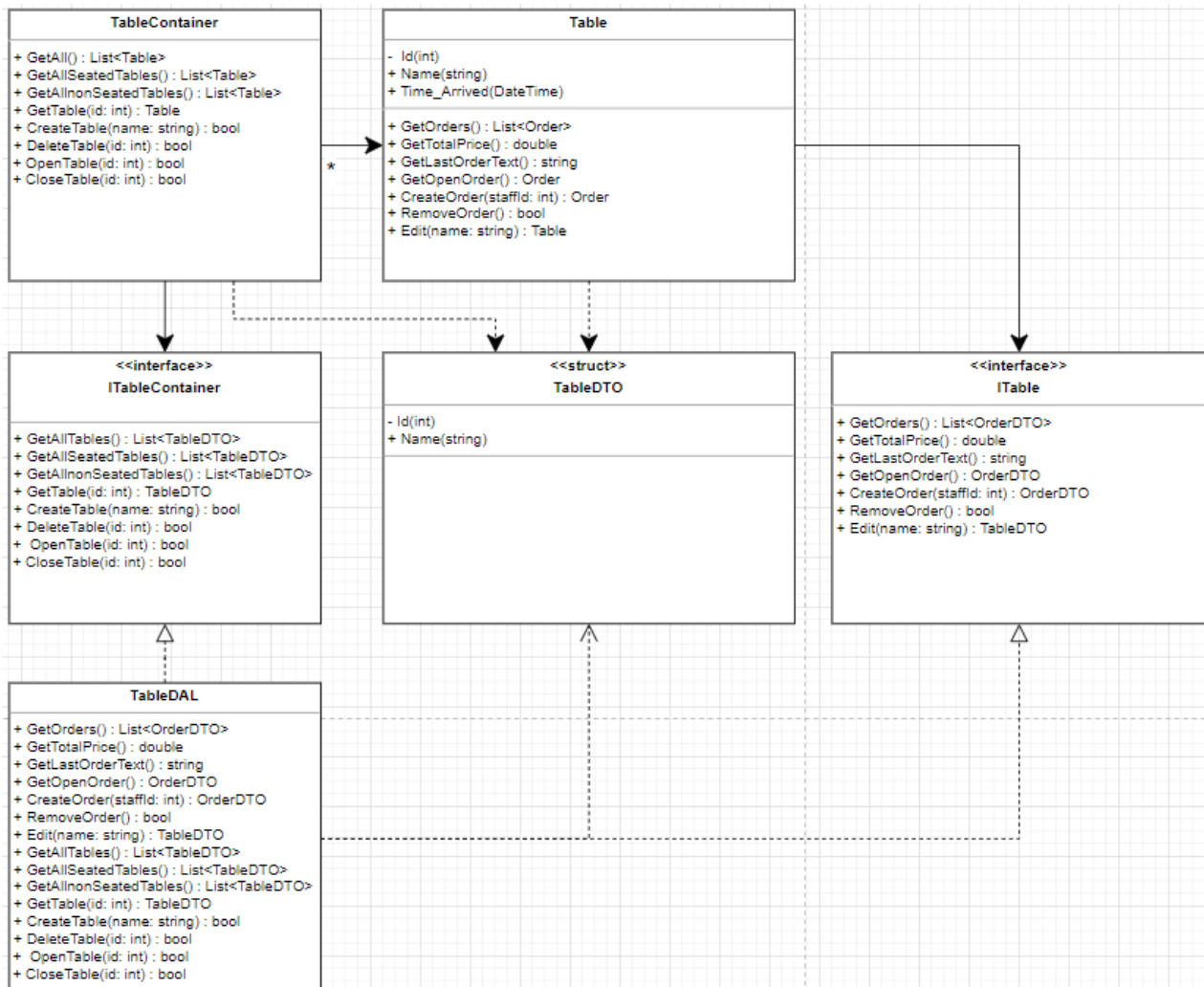


## Use-case diagram

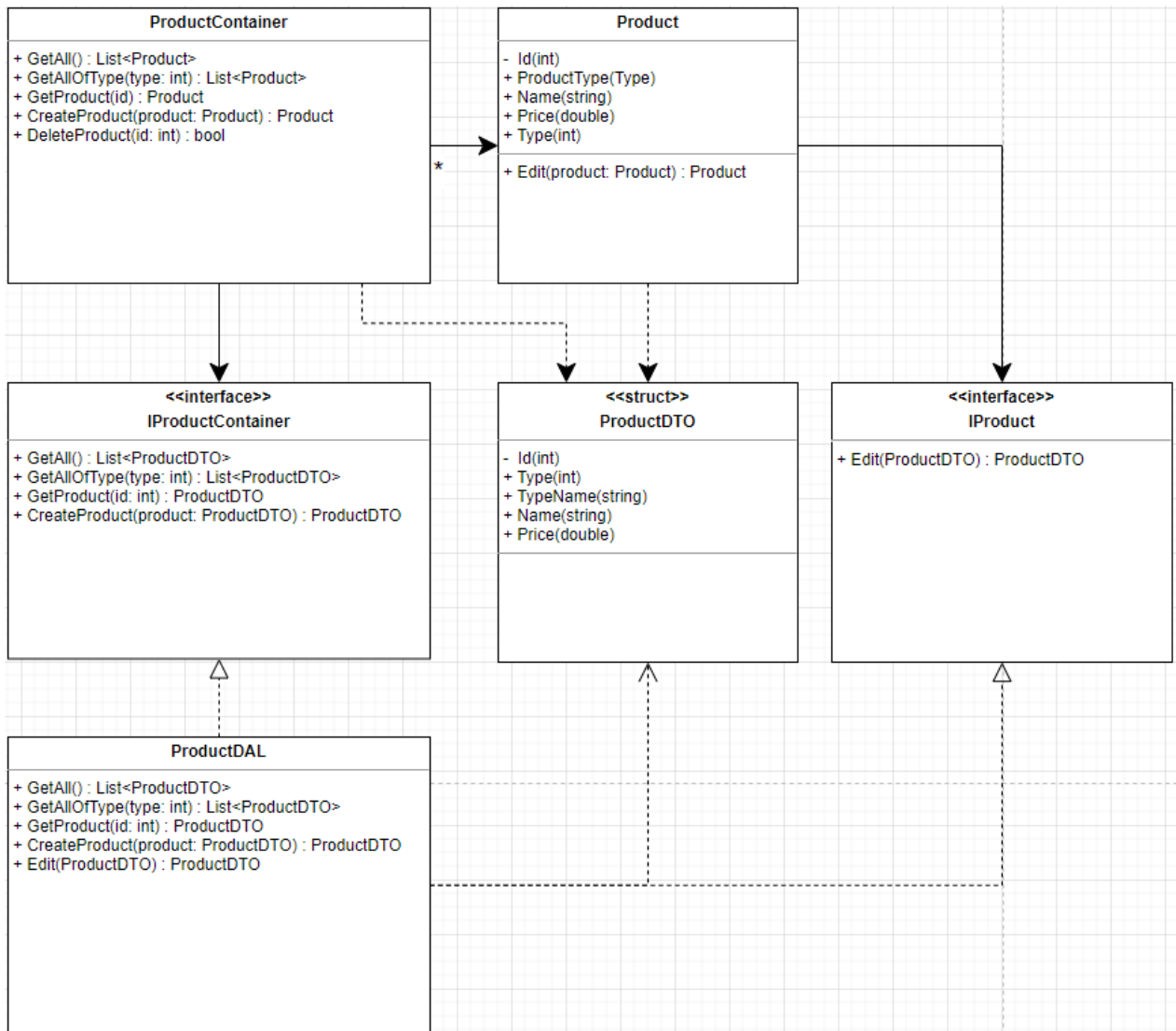


# Klassendiagram

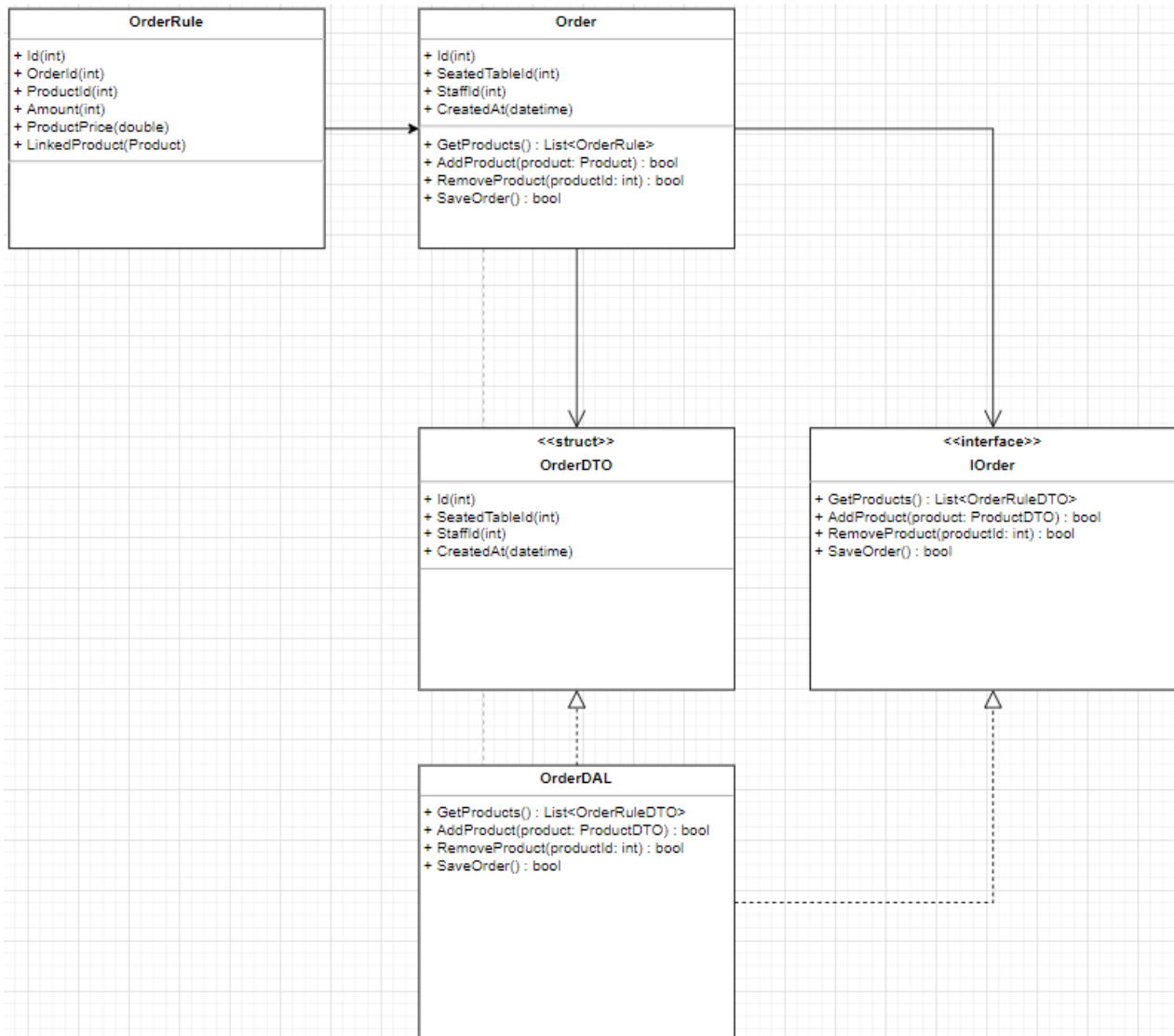
## Table:



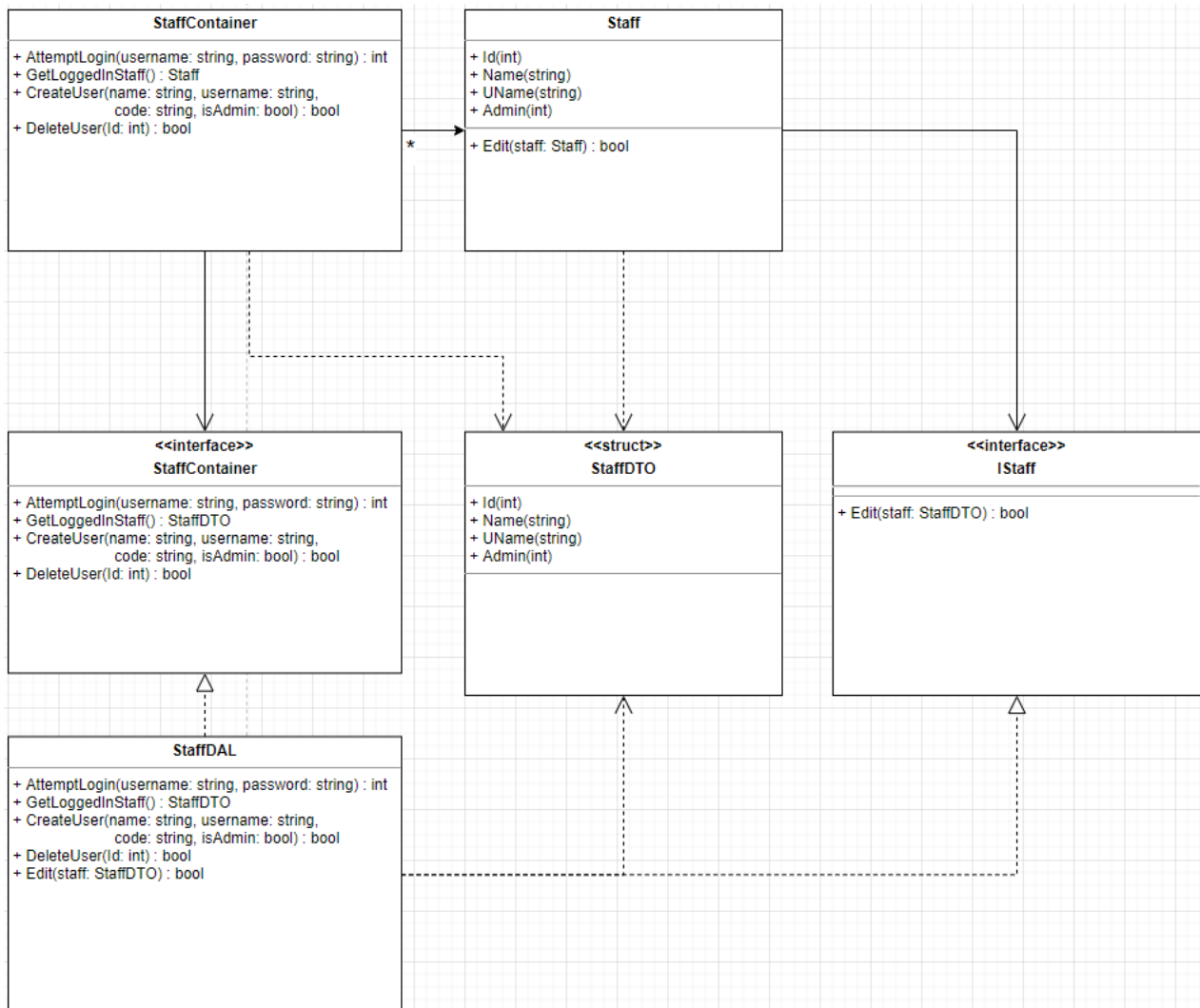
## Product:



## Order:



Staff:



# Testen

## Test Data

<b><i>Test-Data</i></b>	<b>Type</b>	<b>Data</b>
<i>TD-01</i>	Gebruikersnaam Code	Gebruiker1 101101
<i>TD-02</i>	Gebruikersnaam Code	Gebruiker3 101102
<i>TD-03</i>	Gebruikersnaam Code	Gebruiker1 101103

## Test Cases

Test Case	Use Case	Stappen	Verwachting	✓
TC-01	UC-01	1. Vul de inloggegevens in. (Maak gebruik van TD-01)	Er verschijnt een error "Ingevulde gegevens zijn niet correct".	
TC-02	UC-02	1. Vul de inloggegevens in. (Maak gebruik van TD-02)	Er verschijnt een error "Ingevulde gegevens zijn niet correct".	
TC-03	UC-01	1. Vul de inloggegevens in. (Maak gebruik van TD-03)	De gebruiker wordt verwezen naar de hoofdpagina.	
TC-04	UC-02	1. De gebruiker selecteert een tafel uit de lijst van beschikbare tafels (bijv. T-01). 2. De gebruiker accepteert de opening van de tafel.	Na stap 1 verschijnt er een pop-up met de details van de tafel en de opening. Na stap 2 staat de tafel in de lijst van geopende tafels.	
TC-05	UC-02	1. De gebruiker selecteert een tafel uit de lijst van beschikbare tafels (bijv. T-02). 2. De gebruiker accepteert de opening van de tafel <u>niet</u> .	Na stap 1 verschijnt er een pop-up met de details van de tafel en de opening. Na stap 2 staat de tafel niet in de lijst van geopende tafels.	
TC-06	UC-03	1. De gebruiker selecteert een tafel uit de lijst van geopende tafels (bijv. T-01).	De pagina wordt verwezen naar een pagina met een leeg overzicht van de bestelde producten en een overzicht van producten die besteld kunnen worden.	
TC-07	UC-04	1. De gebruiker maakt een order aan.	Op de pagina komt een extra kolom met een lege order.	
TC-08	UC-05	1. De gebruiker selecteert een product uit de producten lijst "Dranken" (Cola).	In de kolom van de order komt er een kopje bij van het type item (Dranken) en daaronder het item (Cola).	
TC-09	UC-05	1. De gebruiker selecteert hetzelfde (als TC-07) product uit de producten lijst "Dranken" (Cola).	In de kolom van de order verandert er onder het kopje Dranken van de cola de 1 in een 2.	
TC-10	UC-05	1. De gebruiker selecteert een product uit de producten lijst "Eten" (Soep).	In de kolom van de order komt een nieuw kopje Eten en daaronder komt het item "soep" te staan.	





# Code

Het project/de code is te vinden op:

<https://github.com/StijnVersluis/ServingSystem>