Ontwerp Document

Skippy

Luuk de Kinderen

# Inleiding

Skippy is een kassasysteem voor een dierenwinkel. Het helpt kassamedewerkers met het opstellen van een order. En het dient als overzicht tool voor de manager van de winkel.

In dit document staat beschreven hoe het systeem werkt en hoe het opgebouwd gaat worden.

Inhoud

[Inleiding 1](#_Toc41660264)

[Database ontwerp 3](#_Toc41660265)

[Klassen diagram 4](#_Toc41660266)

[Architectuur Diagram 5](#_Toc41660267)

# Database ontwerp

Je ziet hier een overzicht diagram van de database structuur.

Legenda:  
enter image description here Geel: Primary Key  
enter image description hereenter image description here Gevuld: NOT NULL  
enter image description hereenter image description here Leeg: NULLABLE  
enter image description hereenter image description here Rood: Foreign key  
enter image description hereenter image description here Blauw: Simple attribute (no key)

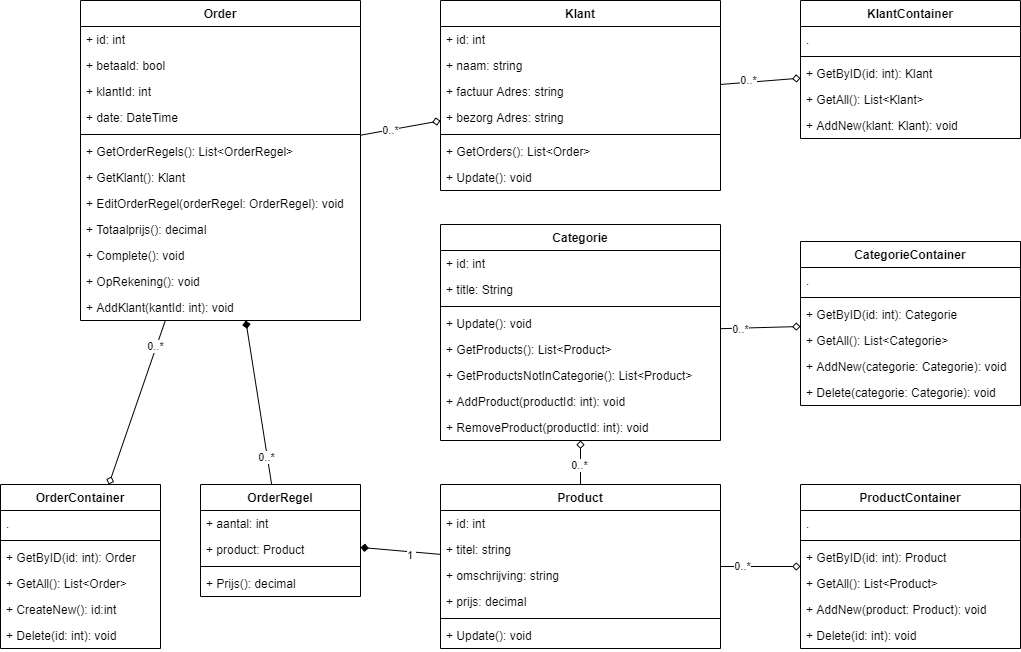
Binnen het systeem is het nodig om een product in meerdere categorieën te kunnen plaatsen. Daarom is hier gekozen om dit via een koppeltabel te doen. En niet via een enkele foreign key.

Ook is gekozen om een orderegel tabel te maken met een aantal kolom er in. Een product kan natuurlijk een aantal keer toegevoegd worden in een bepaalde order.

Omdat orders ook aangemaakt kunnen worden zonder dat deze voor een specifieke klant is. Is de Foreign key van KlantId op de order Nullable.

# Klassen diagram

Dit is het klassendiagram voor de Logica Laag van het systeem. Iedere klasse heeft een aantal properties en Functies. Relaties tussen bepaalde klassen zijn aangegeven met Lijnen.

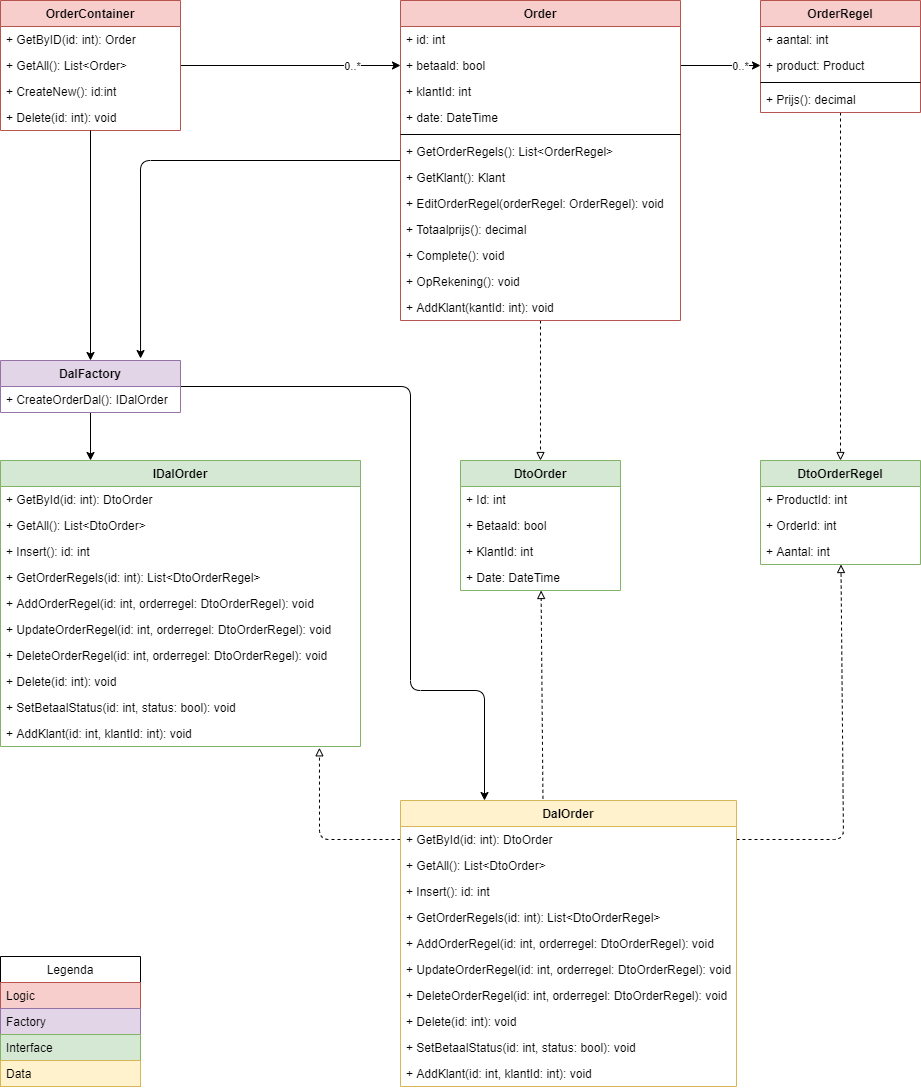


Voor elke entiteit (zoals die in het conceptueel model beschreven staan) Is een klasse aangemaakt met de bijbehorende functies. Voor bijna elke klasse is ook een container klasse gemaakt die weer verantwoordelijk is voor groepen van deze entiteit. Bij de Orderregel is dit niet het geval. De order klasse is namelijk de Containerklasse voor de orderregel. Een orderregel bestaat namelijk alleen als de order ook bestaat.

# Architectuur Diagram

Hier zie je een architectuur Diagram van het systeem. Dit diagram gaat over Orders en alles wat daar bij komt kijken. Iedere laag is verantwoordelijk voor een eigen onderdeel.

* Logic: kern van de applicatie Alle logica gebeurd hier
* Factory: Is de koppel laag die gebruikt word door Logica om Dal classes te bereiken.
* Interface: Is een soort blauwdruk/ contract laag zodat andere lagen weten wat bepaalde objecten kunnen.
* Data/DAL: Is de laag die direct in combinatie staat met de Database.



Door het gebruik van interfaces is het makkelijk om wijzigingen aan te brengen in een van de overige lagen zonder dat het hele systeem aangepast hoeft te worden.

Er is gekozen om geen losse Dal te maken voor de Order Regel. Omdat een Order Regel alleen binnen een order bestaat. Alle database interactie van de order en orderregel zijn daarom samengevoegd in een Dal klasse.