

# Ontwerpdocument

Naam: Yori Scharnigg

Student nummer: 484512

Klas: DB01



## Table of Contents

<b>UML.....</b>	<b>3</b>
<b>ERD:.....</b>	<b>4</b>
<b>Meerlaagse architectuur .....</b>	<b>5</b>

## UML

In mijn UML-class diagram heb ik de klassen wat verder uitgewerkt.

Een user heeft 0 of meerdere orders.

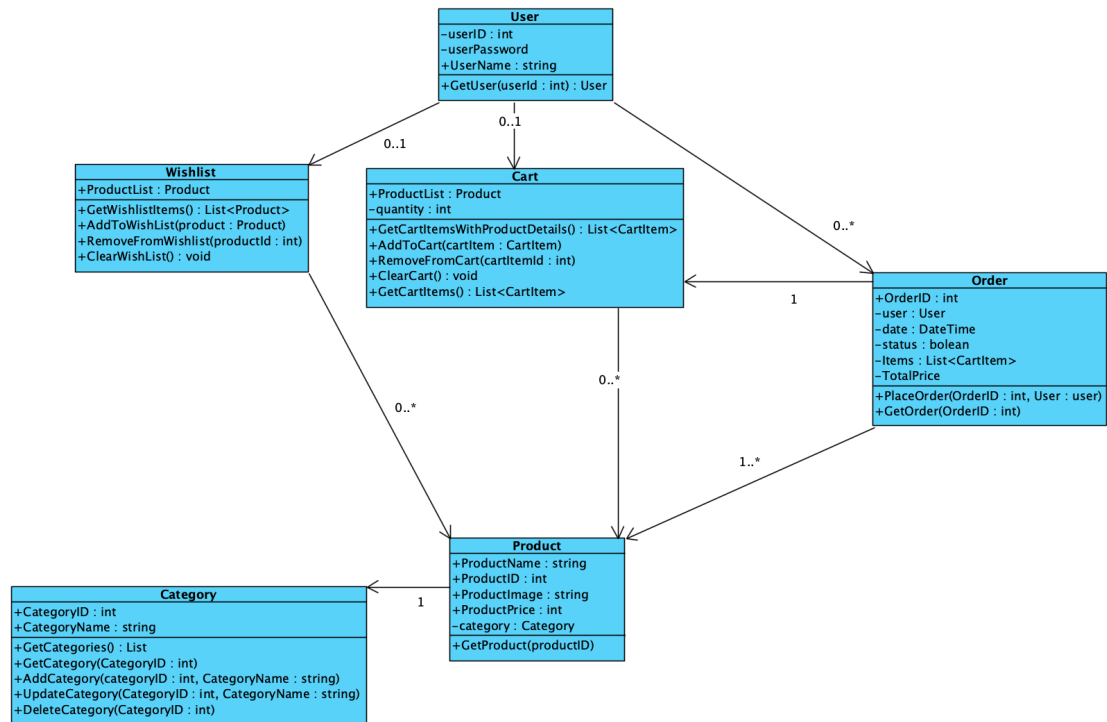
De user heeft ook 0 of 1 cart/wishlist.

Een order heeft 1 cart.

Een order heeft ook 1 of meerdere producten.

Een cart en wishlist hebben 0 of meerdere producten.

Een product heeft altijd 1 category.



## ERD:

Mijn ERD is voor een webshop.

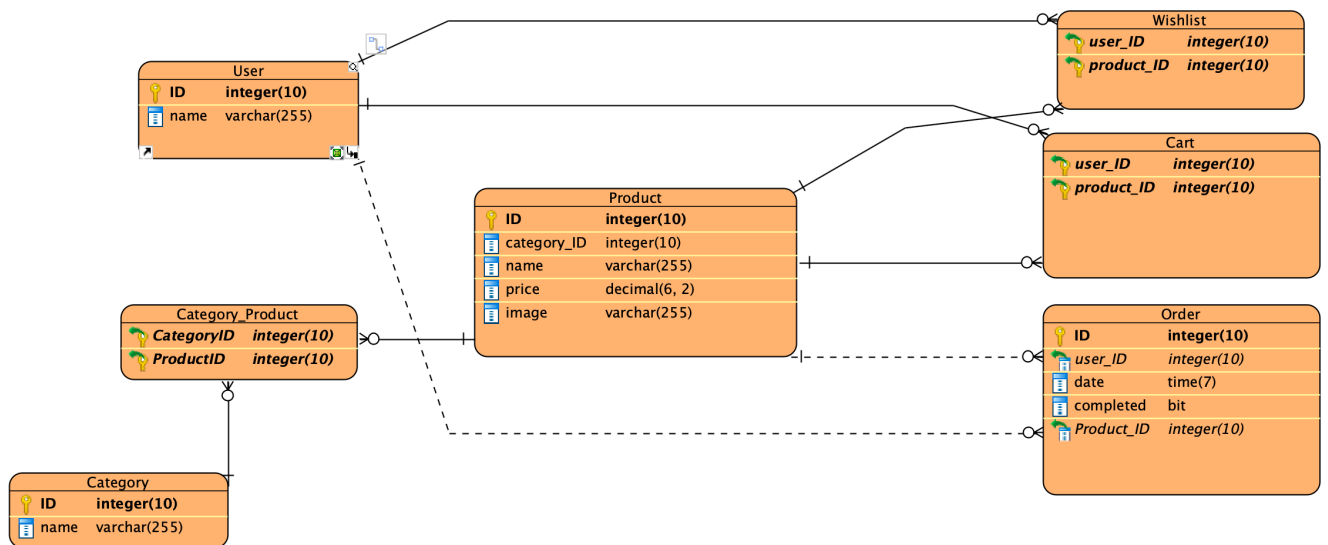
Een categorie heeft meerdere producten, maar een product kan ook meerdere categorieën hebben. Hierdoor heb ik een many-to-many relatie gebruikt. Door gebruik te maken van een junction table (category\_product) zorg ik ervoor dat deze relatie mogelijk is en voorkomt ook het toevoegen van duplicate entries.

Een wishlist en cart kunnen meerdere producten hebben.

Een gebruiker heeft maar 1 cart of wishlist.

Een gebruiker heeft meerdere orders, deze zijn completed of niet.

Een order heeft meerdere producten.



## Meerlaagse architectuur

Hierin is weergegeven welke communicatie de lagen met elkaar hebben.

De presentation layer maakt een instantie aan van een DAL classe, waardoor deze dus connectie heeft met de data layer. Hierna wordt deze instantie ggeïnjecteerd als een interface in de business logic layer. De logic layer gebruikt hierbij dan dependency inversion, waar het de interface mee aan maakt van de DAL.

De service classes in de business logic layer hebben contact met een DTO. Dit is een data transfer object en is de communicatie tussen de logic layer en de data layer. De logic layer maakt dan instanties van deze DTO aan en zet de data erin.

De interfaces maakt gebruik van deze DTO als een type of parameter, waardoor deze dus ook de DTO gebruikt.

De data layer maakt gebruik van de interface en de DTO. Iedere DAL class heeft een interface die deze implementeert. Dit zorgt ervoor dat de logica laag alleen maar weet welke methods er in deze DAL zijn, maar niet hoe deze uitgevoerd worden.

De DAL maakt ook nieuwe instanties van DTO aan waardoor het de data die uit de database gehaald wordt, in deze nieuwe instantie zet.

Verder zijn er nog unit tests. Deze testen de functionaliteiten van de business logic layer.

Iedere functionaliteit wordt dan getest.

