ONTWERPdocument

Live Performance

**Naam:** Sven Henderickx

**Klas:** S27

**Assessor:**

**Datum:** 22-9-2016

Inhoud

[Inleiding 3](#_Toc423074826)

[Database ontwerp 5](#_Toc423074830)

[Toelichting 5](#_Toc423074831)

[Infrastructuur 7](#_Toc423074833)

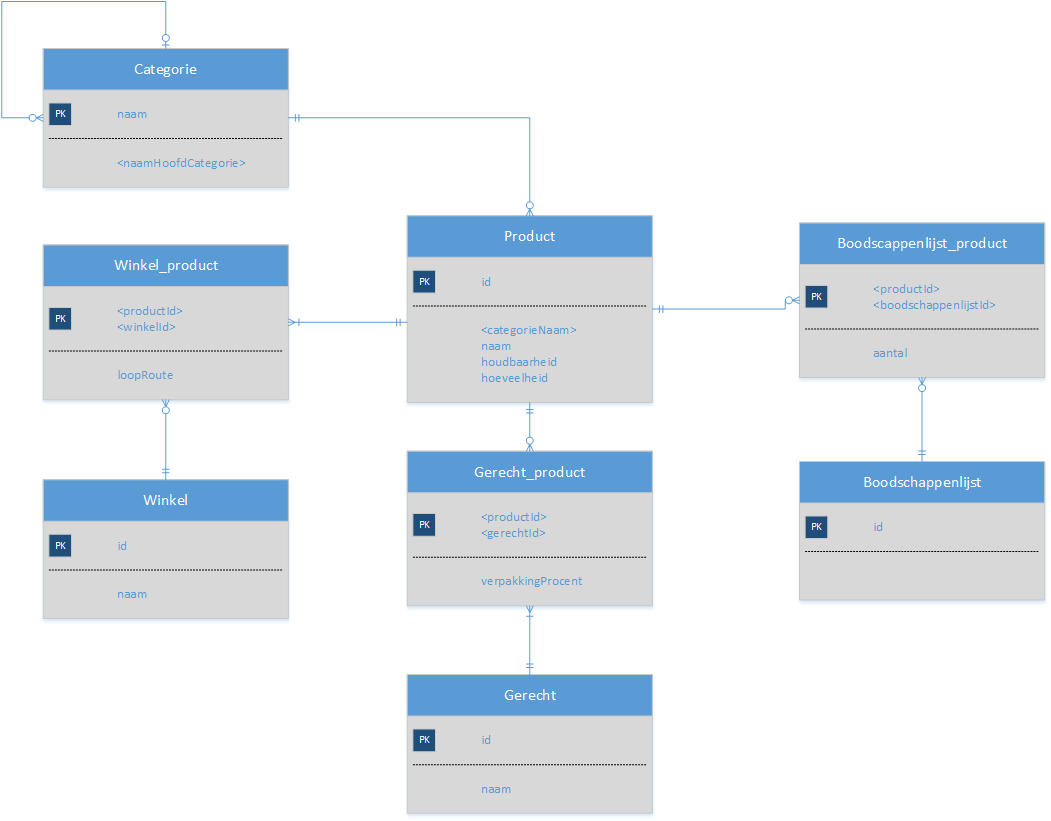
[Klassendiagram 8](#_Toc423074834)

[Toelichting 8](#_Toc423074835)

# Inleiding

In dit document zullen alle ontwerpen staan die ik zal gebruiken voor het implementeren van het programma.

# Database ontwerp

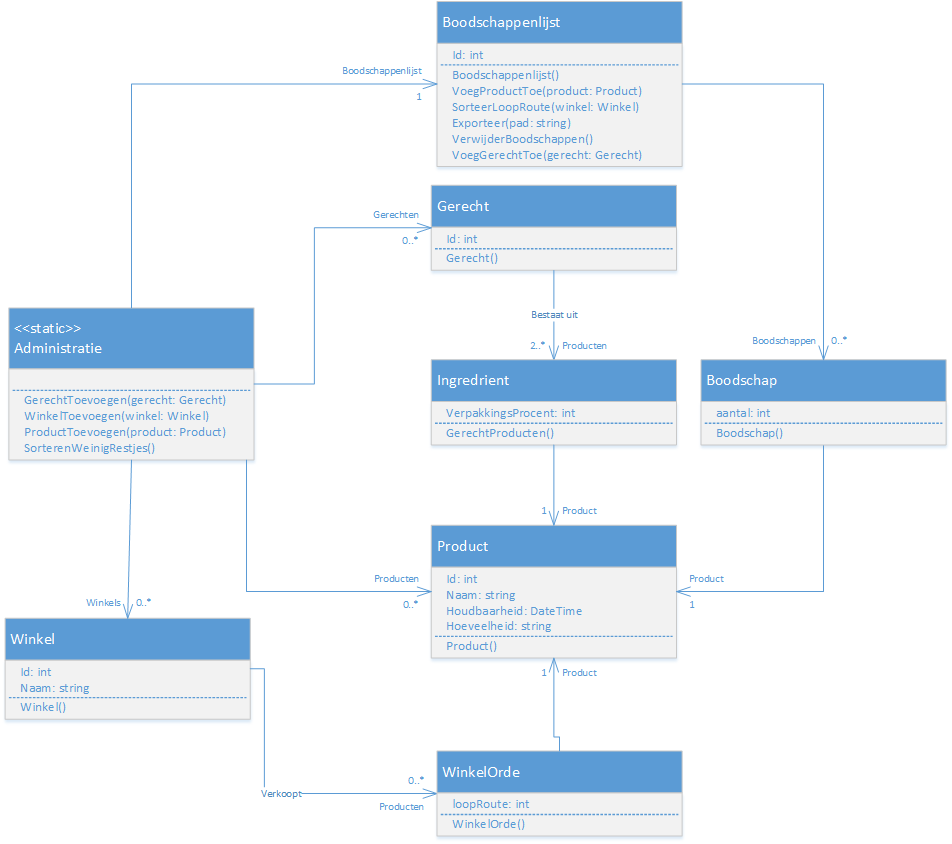


### Toelichting

Zoals je hier in het databaseontwerp ziet, zitten er veel verschillen in met mijn erd. Ik heb alle koppeltabellen toegevoegd en in de koppeltabellen nog de kenmerken loopRoute, verpakkingsProcent en het aantal. Dit komt omdat deze allemaal verschillen per product.

### Klassendiagram

Hieronder staat het klassendiagram. Dit is een schematische weergaven van hoe de applicatie eruit moet komen te zien. Het klassendiagram is opgesteld volgens de UML notatie.



### Toelichting

Ik heb een static klasse Administratie erbij gedaan hierin worden alle lijsten opgeslagen zoals producten etc. Verder zit er een klasse GerechtProduct in die ervoor zorgt dat in het gerecht kan worden aangegeven hoeveel verpakking er word verbruikt.

### Aanpassingen

Tijdens het implementeren is er een klasse gekomen tussen product en winkel deze klassen zorgt ervoor dat de looproute kan worden gemaakt. Ik had hiervoor gekozen omdat elk product in elke winkel een andere positie heeft.

Verder heb ik nog 2 klasses aangemaakt met de naam ZuinigeGerechten en ZuinigeIngredrient. Deze klasses heb ik aangemaakt om data op te slaan voor het berekenen en sorteren van de meeste zuinige combinatie van gerechten. In zuinigeGerechten word opgeslagen hoeveel vermindering er is en welke gerechten zijn gecombineerd. In ZuinigeIngredrient word gebruikt om het restafval te kunnen berekenen omdat dit per ingrediënt moet berekend per gerecht.

# Schetsen

