

ODDB

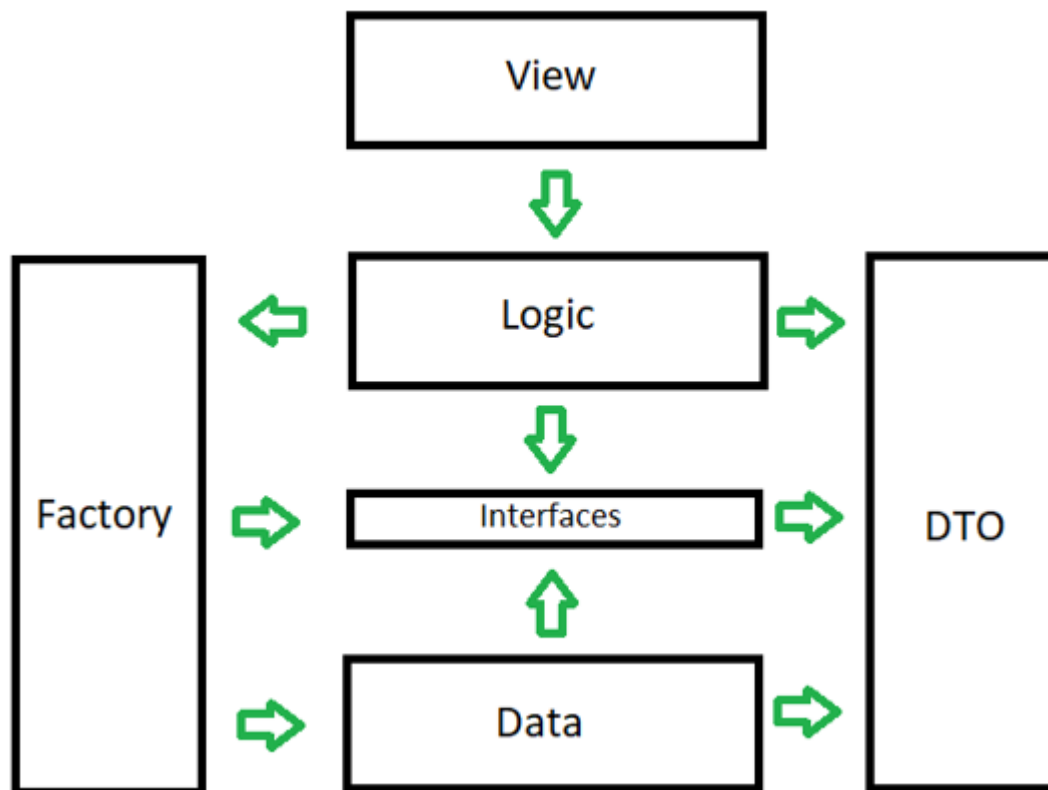
Ontwerp document



## Inhoud

Architectuur .....	3
Presentation Layer (View) .....	3
Business Logic Layer (Logic) .....	3
Interfaces, Factories en DTO's .....	4
Data Access Layer (Data): .....	4
Wat ga ik gebruiken: .....	4
Klasse diagrammen .....	5
Database ontwerp .....	6

## Architectuur



Voor het ODDB project word de 3-layer Architecture gebruikt. De 3-layer architecture is een vrij simpele architectuur, deze bestaat zoals de naam zegt uit 3 lagen: de presentation layer, de business logic layer en de data acces layer (DAL). Er zitten ook factories, DTO's en interfaces in deze architectuur zodat de lagen niet direct met elkaar communiceren.

### Presentation Layer (View)

De presentation layer is de laag die ervoor zorgt dat een gebruiker interactie kan hebben met de applicatie. Denk hierbij aan een UI. Deze laag communiceert met de logic laag, de logic laag staat wel los van de presentation layer. Hierdoor is je programma niet afhankelijk van UI.

### Business Logic Layer (Logic)

In deze laag staat je programma, hierin staan alle functies en methoden. De logic layer communiceert allen met inerfaces, factories en DTO's, hi. Hierdoor staat de kern van de applicatie los van alle andere onderdelen. Op deze manier kun je de applicatie dus runnen ongeachte van welke view of DAL laag er aan hangt.

### Interfaces, Factories en DTO's

De interfaces, factories en DTO's dienen als tussen 'lagen' zodat de logic laag toch met de DAL laag kan communiceren, maar op deze manier staat de logic laag overal los van. Hierdoor kan de logic laag met iedere view en DAL gebruikt kan worden.

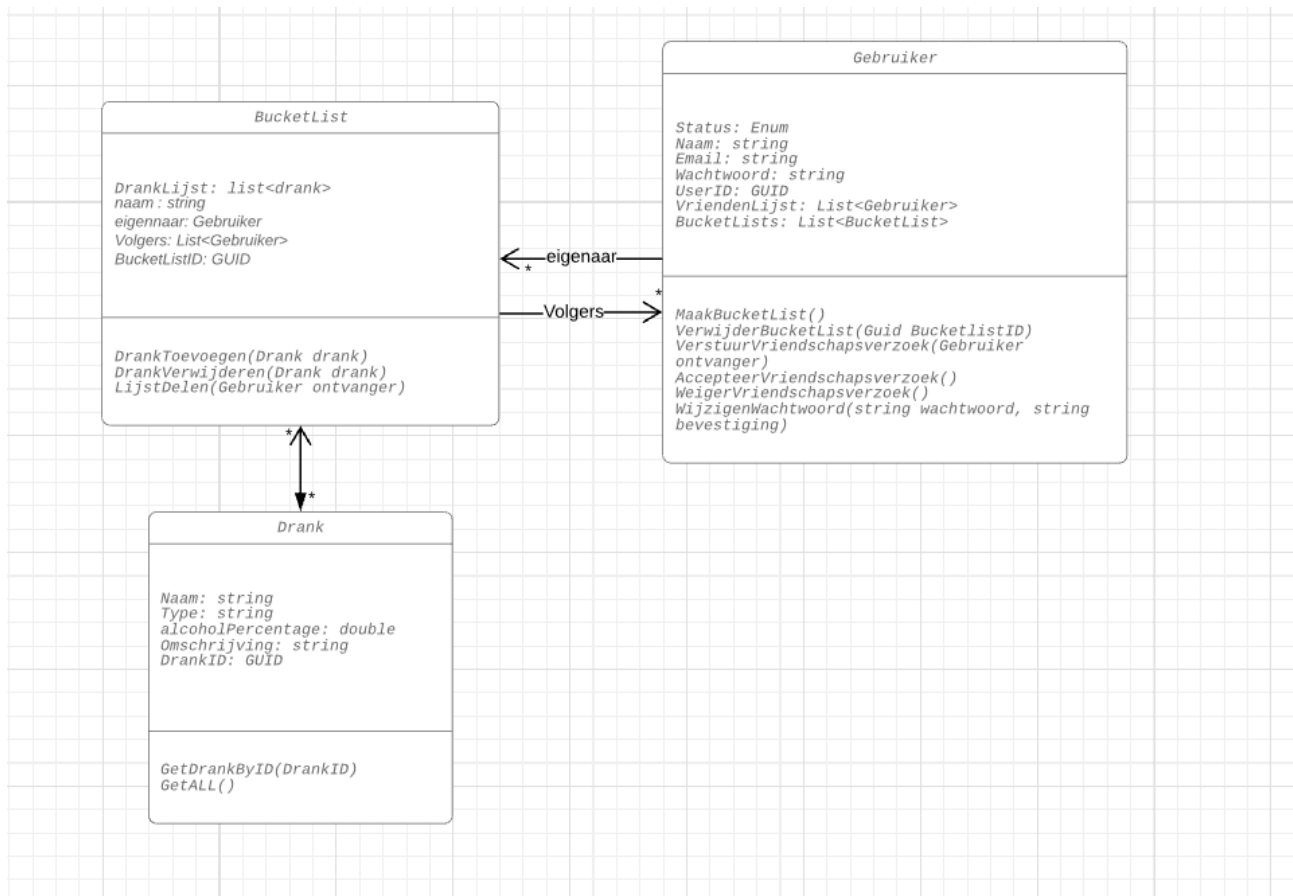
### Data Access Layer (Data):

In de Dal laag word de connectie met de database gemaakt en worden de queries aangemaakt die de database bevragen voor de benodigde informatie. Deze laag communiceert via interfaces ook met de Logic laag.

### Wat ga ik gebruiken:

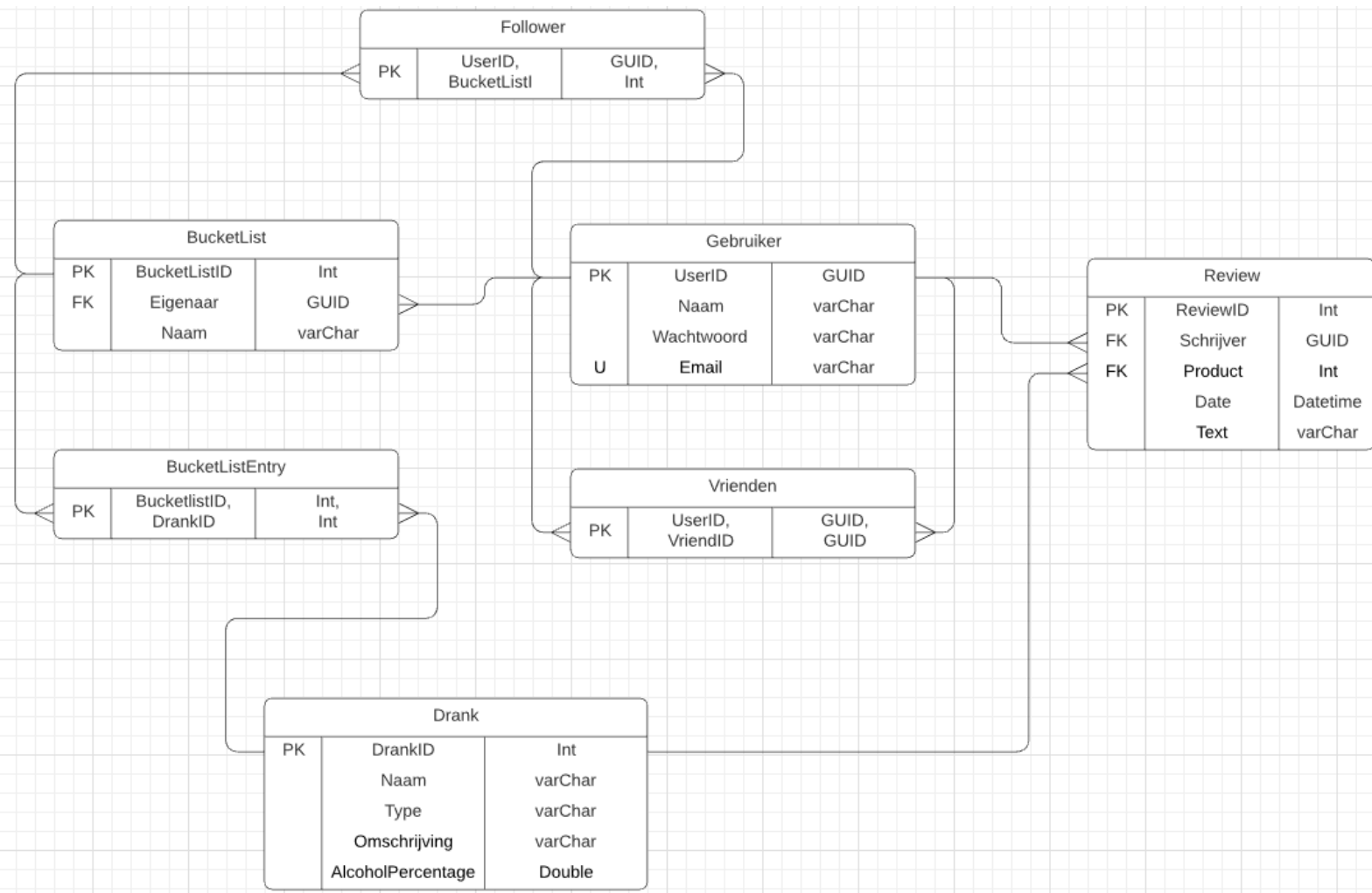
Ik heb ervoor gekozen om een iets andere structuur te gebruiken, wat ik heb gebruikt is een structuur waarbij je view via interfaces aan de data laag zit. Dit heb ik gedaan, omdat ik een Crud applicatie heb gemaakt en deze geen logica bevat. In dit geval is het dan handiger om je logica laag weg te laten. Verder gebruik ik ook geen factories. Ik maak gebruik van Dependency Injection.

## Klasse diagrammen



Hierboven staat de Klasse diagram voor mijn eigen project: “Online Drank Database” (ODDB). Er zijn 4 klasse betrokken in het programma, een gebruiker klasse, een bucketlist klasse en een drank klasse. De gebruiker klasse gaat de klasse zijn waarvan alle accounts afgeleid zullen worden. De bucketlist klasse is de blauwdruk voor alle bucketlists. De drank klasse zal de template zijn voor alle drank die in de database te vinden is. In de afbeelding is duidelijk te zien welke variabele alle klassen zullen hebben en welke functies terug te vinden zijn in de ze klassen.

## Database ontwerp



Dit is het database ontwerp, hierin staan alle verschillende tabellen die in mijn database opgenomen gaan worden. De tabellen met alleen een rij erin zijn koppel tabellen, deze worden gebruikt als er veel op veel relaties zijn.