|  |
| --- |
| Robin van der Snoek  02/20/2023  Versie: 1.0 |

|  |
| --- |
| Technisch Ontwerp (TO) |
| Project: Cocktail Builder |
|  |

# Inhoud

[Inhoud 1](#_Toc124518219)

[Technische specificaties 3](#_Toc124518220)

[Codeertalen 3](#_Toc124518221)

[Database 3](#_Toc124518222)

[Frameworks 3](#_Toc124518223)

[Starter kit 3](#_Toc124518224)

[Package Manager 3](#_Toc124518225)

[Dependency Manager 3](#_Toc124518226)

[Presentation 4](#_Toc124518227)

[Layout 4](#_Toc124518228)

[Logo 4](#_Toc124518229)

[Details over de zoekbalk input veld: 4](#_Toc124518230)

[Hoofdpagina 4](#_Toc124518231)

[Featured dranken 4](#_Toc124518232)

[Willekeurige dranken 4](#_Toc124518233)

[Zoeken met Ingrediënten pagina 4](#_Toc124518234)

[Form 4](#_Toc124518235)

[Dranken selecteren 4](#_Toc124518236)

[Gezochte dranken tonen 5](#_Toc124518237)

[Hoeveelheid resultaten gevonden 5](#_Toc124518238)

[Dranken tonen 5](#_Toc124518239)

[Pagination 5](#_Toc124518240)

[Cocktail informatie tonen 5](#_Toc124518241)

[Informatie 5](#_Toc124518242)

[Tags 5](#_Toc124518243)

[Ingrediënten 5](#_Toc124518244)

[Ingrediënt informatie tonen 6](#_Toc124518245)

[Informatie 6](#_Toc124518246)

[Dranken 6](#_Toc124518247)

[Code 6](#_Toc124518248)

[Case style 6](#_Toc124518249)

[Functionaliteiten 6](#_Toc124518250)

[Navigeren naar cocktail of ingrediënt pagina 6](#_Toc124518251)

[Dranken zoeken via zoekbalk 6](#_Toc124518252)

[Dranken zoeken via ingrediënten 6](#_Toc124518253)

[Het toevoegen van dranken en ingrediënten 7](#_Toc124518254)

[Alle cocktail opnemen uit de API 7](#_Toc124518255)

[Controleren op nieuwe ingrediënten 7](#_Toc124518256)

[Alle nieuwe ingrediënten opnemen uit de API 7](#_Toc124518257)

[De measures 7](#_Toc124518258)

[Data toevoegen aan de database 7](#_Toc124518259)

[Task scheduling 7](#_Toc124518260)

[Het wisselen tussen metrisch en imperial 7](#_Toc124518261)

[Relationeel datamodel 8](#_Toc124518262)

[Entity Relationship Diagram 8](#_Toc124518263)

[Drinks table 8](#_Toc124518264)

[Ingredients table 8](#_Toc124518265)

[Drinks, ingredients pivot table 9](#_Toc124518266)

[Category table 9](#_Toc124518267)

[Glasses table 9](#_Toc124518268)

[Eloquent relationships 9](#_Toc124518269)

[Drinks 9](#_Toc124518270)

[Ingredients 9](#_Toc124518271)

[Glass 9](#_Toc124518272)

[Category 9](#_Toc124518273)

# Technische specificaties

## Codeertalen

HTML5

CSS3

PHP: v8.1.10

## Database

Database connection: SQL

MySQL: v8.0.31

## Frameworks

Laravel Framework: v10.0

Tailwind CSS: v3.0

Livewire: v2.x

## Starter kit

Breeze

## Package Manager

Npm: v8.19.2

## Dependency Manager

Composer: v2.4.1

## Code style

### Case style

We volgen de Laravel conventie en daarom zullen we voornamelijk met camelCasing werken in plaats van snake\_casing.

## Code veiligheid

### Queries

Bij queries word de functie ‘->whereRaw() voor de veiligheid ontweken. Aangezien we data van een API vandaan halen moeten we oppassen met de data dat we ontvangen.

### eval()

‘eval()’ is een functie waarbij je met een variable php code kunnen runnen, hier maken we geen gebruik van maken sinds dit ongewenste code vanuit de API kan betekenen.

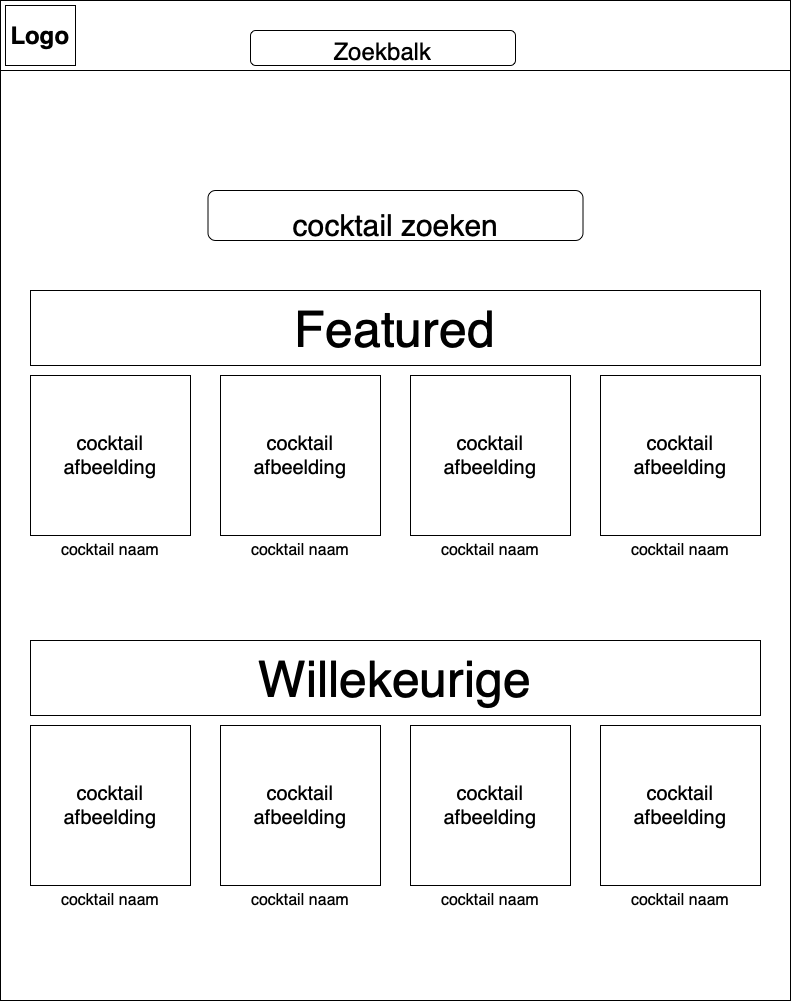
# Presentation

## Layout

### Logo

Op elke pagina wordt de footer met het logo, zoekbalk en account info getoond. Wanneer de gebruiker op de logo knop klikt, wordt hij genavigeerd naar het hoofdmenu.

### Details over de zoekbalk input veld:

EncType: text/plain

Input: verplicht

Query: zoekt alleen naar dranken

## Hoofdpagina

### Featured drinks

De acht eerst geselecteerde dranken die featured is volgens de featured boolean in de database.

De dranken worden getoond 4 per rij.

Note: Featured is in dit project random gegenereerd.

### Willekeurige dranken

Ook worden er 8 willekeurige dranken getoond die we ontvangen door middel van de ‘inRandomOrder’ scope.

Toon deze vervolgens ook met 4 op elke rij.

## Zoeken met Ingrediënten pagina

### Form

Alle Ingrediënten worden getoond met select box waarbij je ze kan selecteren.

In de controller wordt gevalideerd of er op zijn minst een ingrediënt is geselecteerd, als dit niet het geval is wordt de gebruiker niet doorgestuurd maar blijft hij op deze pagina

### Dranken selecteren

Via de cocktail controller selecteren we alle dranken waarbij alle ingrediënten in de cocktail voorkomen in de database.

## Afbeelding met tekst Automatisch gegenereerde beschrijvingGezochte dranken tonen

### Hoeveelheid resultaten gevonden

Om de hoeveelheid resultaten te ontvangen gebruik je de ‘->count()’ functie om gemakkelijk de hoeveelheid te krijgen dat in de array zit.

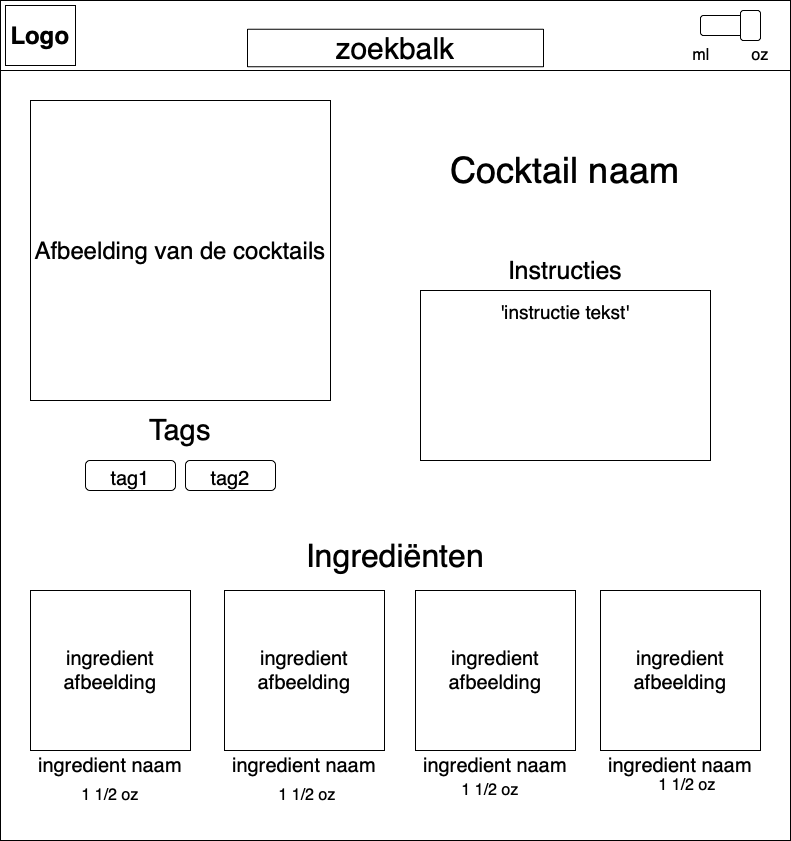
### Dranken tonen

Alle dranken worden getoond op alfabetische volgorde, met 4 dranken per rij.

Om alle dranken te tonen maken we gebruik van een foreach loop.

### Pagination

Per pagina worden er maximaal 16 dranken getoond. Om de volgende pagina te zien maken we gebruik van pagination.



## Cocktail informatie tonen

### Informatie

Toon de naam, afbeelding, instructies, categorieën (ook wel tags genoemd op de pagina) en ingrediënten.

### Tags

Wanneer de cocktail een categorie heeft, wordt die getoond als de eerste tag. Als de cocktail alcohol bevat, zal dat ook worden getoond als een tag.

### Ingrediënten

Onderaan de pagina worden er 4 ingrediënten per rij getoond die gebruikt worden in deze cocktail. Deze afbeeldingen leiden vervolgens naar de pagina van het ingrediënt zelf.

## Ingrediënt informatie tonen

### Informatie

Toon de naam en afbeelding, en dranken waar dit ingrediënt in voorkomt

### Dranken

Onderaan de pagina worden er 8 willekeurige dranken getoond die deze ingrediënt gebruiken. De afbeelding daarvan leidt vervolgens naar de pagina van die cocktail

# Functionaliteiten

## Navigeren naar cocktail of ingrediënt pagina

Wanneer je op de naam of afbeelding van zowel een cocktail of ingrediënt klikt. Word je genavigeerd naar een route die vervolgens in de controller alle informatie opzoekt via een query.

Wanneer de cocktail/ingrediënt is gevonden word je genavigeerd naar een view met de data van de query. Met deze data kan de volgende pagina vervolgens de naam, beschrijving enzovoort tonen.

## Dranken zoeken via zoekbalk

Wanneer je via de zoekbalk zoekt word je genavigeerd naar een route die vervolgens in de cocktail controller alle dranken opzoekt waarbij de naam overeenkomt met wat men heeft ingevuld in de zoekbalk. Vervolgens word je naar de view gestuurd met alle of geen gevonden dranken in een array.

## Dranken zoeken via ingrediënten

Wanneer je zoekt met geselecteerde ingrediënten, word je naar een route gestuurd die vervolgens in de cocktail controller alle ingrediënten ophaalt. Met deze ingrediënten is er een query die zoekt naar dranken waarbij al die ingrediënten in voorkomen.

## Het toevoegen van dranken, ingrediënten, categorieën en glazen

Om het uur worden alle dranken geselecteerd. Omdat het door de API niet zo gemakkelijk gaat, moeten we naar de dranken zoeken met de eerste letter van hun naam.

Voordat we dingen uit de API gaan halen, halen we eerst het volgende uit onze database.

* De namen van alle dranken
* De namen van alle ingrediënten

### Alle cocktail opnemen uit de API

Vervolgens doen we deze data in een aparte array, ook hebben we een string van A tot en met Z die we splitten in een alfabet array.

Met de alfabet array gebruiken we nu een foreach loop. In deze loop ontvangen we de dranken van de API die beginnen met deze letter. En kijken we bij elke cocktail of de naam al in onze dranken array staat. Alleen als het er nog niet in staat kan deze worden toegevoegd aan een nieuwe dranken data array.

### Controleren op nieuwe ingrediënten

Vervolgens moeten we bij elke nieuwe cocktail kijken naar de ‘strIngredient1’ tot en met ‘strIngredient15’ attributen om kijken of deze namen voorkomen met namen in onze ingrediënten array. Zo niet, dan voegen we die toe aan een nieuwe ingrediënten namen array. Hierbij hebben we alleen de naam van het ingrediënt, dus moeten alle informatie over het ingrediënt eerst vinden vanuit de API.

### Alle nieuwe ingrediënten opnemen uit de API

Hierbij gebruiken we weer een foreach loop die door alle nieuwe ingrediëntnamen loopt. In deze loop zoeken we de data van het ingrediënt vanuit de API. Deze data doen we vervolgens in een nieuwe ingrediënten data array.

### De measures

In de API wordt er gewerkt met oz en cups, dit moeten we standaard omrekenen naar alleen mL. Daarom kijken we eerst of de measure in cups of oz is berekend, en op basis daarvan gebruiken we de juiste berekening om het neer milliliter te berekenen.

### Data toevoegen aan de database

Nu we alles hebben kunnen we beginnen met het toevoegen van de data in de database. Eerst voegen we de nieuwe dranken toe, daarna de ingrediënten, dan de categorieën en tot slot de glazen.

Bij zowel de ingrediënten, categorieën en de glazen, word hierna alle dranken geüpdatet die het desbetreffende bevat. Bij de ingrediënten attachen we de cocktail, en bij de andere twee updaten de desbetreffende column. Die eerst de naam had, maar nu het id krijgt.

## Task scheduling

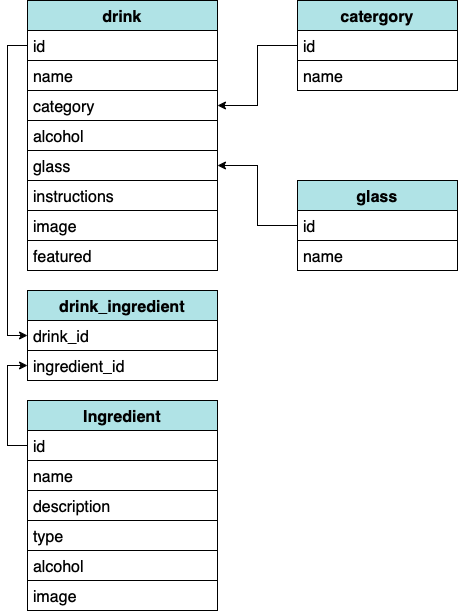
Om het toevoegen van dranken en ingrediënten om het uur te laten draaien gebruiken we Task Scheduling. Om het uur kunnen we dus een Artisan commando uitvoeren die vervolgens het toevoegen van de dranken en uitvoerd, en zodra er een cocktail wordt toegevoegd, gaat het ook kijken naar ingrediënten, het glass en de categorie.

## Het wisselen tussen metrisch en imperial

Bij het aanmaken van een cocktail wordt er gekeken of de volumes in ml of oz is ingevuld, als het in oz is ingevuld wordt het eerst naar ml berekent voordat het de database ingaat.

# Relationeel datamodel

## Entity Relationship Diagram



## Drinks table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Unsigned** | **Allow Null** | **Key** | **Default** | **Extra** |
| id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| name | VARCHAR | 255 |  |  |  |  |  |
| category | VARCHAR | 255 |  |  |  | NULL |  |
| alcohol | BOOLEAN |  |  |  |  | FALSE |  |
| glass | BIGINT | 255 |  |  | MUL |  |  |
| intruction | LONGTEXT | 255 |  | TRUE |  |  |  |
| image | VARCHAR | 255 |  | TRUE |  |  |  |
| featured | BOOLEAN |  |  |  |  | FALSE |  |

## Ingredients table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Unsigned** | **Allow Null** | **Key** | **Default** | **Extra** |
| id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| name | VARCHAR | 255 |  |  |  |  |  |
| description | LONGTEXT | 255 |  | TRUE |  | NULL |  |
| type | VARCHAR | 255 |  | TRUE |  | NULL |  |
| alcohol | BOOLEAN |  |  |  |  | FALSE |  |
| image | VARCHAR | 255 |  | TRUE |  |  |  |

## Drinks, ingredients pivot table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Unsigned** | **Allow Null** | **Key** | **Default** | **Extra** |
| id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| ingredient\_id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| Ingredient\_measure | VARCHAR | 255 |  |  |  | 0 mL |  |

## Category table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Unsigned** | **Allow Null** | **Key** | **Default** | **Extra** |
| id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| name | VARCHAR | 255 |  |  |  |  |  |

## Glasses table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Length** | **Unsigned** | **Allow Null** | **Key** | **Default** | **Extra** |
| id | BIGINT |  | TRUE |  | PRIMARY |  | AI |
| name | VARCHAR | 255 |  |  |  |  |  |

## Eloquent relationships

### Drinks

* Belongs to many ingredients
* Belongs to one glass
* Belongs to one category

### Ingredients

* Belongs to many drinks

### Glass

* Belongs to many drinks

### Category

* Belongs to many drinks