Technisch Ontwerp



# Inhoudsopgave

[Inhoudsopgave 2](#_Toc133305164)

[Inleiding 4](#_Toc133305165)

[Database 4](#_Toc133305166)

[Projectmanagement system 4](#_Toc133305167)

[Kwaliteit van code bewaken 5](#_Toc133305168)

[Coding conventies 5](#_Toc133305169)

[Laravel conventies 5](#_Toc133305170)

[PHP conventies 5](#_Toc133305171)

[HTML conventies 5](#_Toc133305172)

[CSS conventies 5](#_Toc133305173)

[Prettier formatting 5](#_Toc133305174)

[JSON-conventies 6](#_Toc133305175)

[Performance bewaken 7](#_Toc133305176)

[Technologie 7](#_Toc133305177)

[Front-end 7](#_Toc133305178)

[Backend 7](#_Toc133305179)

[Database 7](#_Toc133305180)

[Datamodellen 10](#_Toc133305181)

[Security 10](#_Toc133305182)

# Inleiding

In het technisch Ontwerp wordt toegelicht hoe het project gemaakt gaat worden, hoe het project gemanaged wordt en hoe het project effect gaat hebben.

Voor dit project gaan wij gebruik maken van Laravel en VueJs.

De Front-end word geschreven in:   
Bron HTML 5 (<https://html.com/html5/> ),   
Bron CSS (<https://en.wikipedia.org/wiki/CSS> ),   
Bron VueJs (<https://vuejs.org/guide/introduction.html> )

De Back-end zal gebruik maken van het Laravel framework en het MVC ontwerppatroon.  
**Model-view-controller** (of MVC) is een ontwerppatroon dat het ontwerp van complexe toepassingen opdeelt in drie eenheden met verschillende verantwoordelijkheden: datamodel (model), datapresentatie (view) en applicatielogica (controller).

Bron PHP (<https://www.php.net/> ),  
Laravel (<https://laravel.com/> )   
Bron MVC (<https://nl.wikipedia.org/wiki/Model-view-controller-model> )  
Bron MVC (<https://fuelphp.com/docs/general/mvc.html> )

De database zal een SQL database worden.   
(<https://en.wikipedia.org/wiki/SQL> )

Wij hebben hiervoor gekozen omdat het Laravel framework een aantal voordelen biedt voor dit project. Laravel heeft namelijk een hoge applicatie snelheid en veel “Out of the box” functionaliteiten die handig zijn voor ons project en het realiseren makkelijker maken.

# Database

Wij hebben gekozen voor een SQL database omdat wij hier al ervaring mee hebben.

Wij maken gebruik van foreign key relationships zoals je in een SQL-database kan vinden, bijvoorbeeld elke leverancier heeft een klant id. Hierdoor kunnen wij makkelijk orders aan klanten koppelen.

Alle gegevens zullen op een veilige manier in de database opgeslagen worden.  
Voor meer informatie over de veiligheid verwijs ik je door naar het kopje “Security”

# Projectmanagement system

Wij gebruiken SCRUM met Jira voor het bijhouden van de progressie van het project.   
Door User stories op te breken in taken en in sprints te verdelen houden wij het overzichtelijk en kunnen wij de progressie goed bijhouden.

Onze sprints duren één werk week lang en beginnen op maandag 08:30 en eindigen vrijdag 17:00. Wij houden elke week een scrum meeting om de vooruitgang te bespreken en eventueel ideeën of problemen te bediscussiëren.

Indien een taak na afloop van de haar eerste sprint niet afgerond is, zal het probleem besproken worden. Als er hulp nodig is zal dit tijdens de meeting duidelijk worden en zal dit naar volgende sprint meegenomen worden.

# Kwaliteit van code bewaken

## Coding conventies

Tijdens het maken van dit project gebruiken wij de industriële standaard code conventies.

Coding conventions zijn **een set regels en voorschriften die developers en software programmeurs volgen om eenheid te krijgen in wat ze schrijven**. In plaats van elk in een eigen stijl en vorm te coderen, volgen ze de standaard die vooropgesteld wordt.

Om dit nog makkelijker te maken gebruiken wij “Prettier” om de code automatisch te formateren. Op deze manier blijft de code overzichtelijk en makkelijk te lezen.

### Laravel conventies

Het is belangrijk om coding practices aan te houden om code overzichtelijk te hebben.

Hiervoor gebruiken wij de documentatie van Laravel die kan je terugvinden hieronder:  
Bron: <https://laravel.com/docs/7.x>

### PHP conventies

De coding conventies die wij tijdens dit project gebruiken zijn te vinden op de bron hieronder:  
Bron:<https://flowframework.readthedocs.io/en/stable/TheDefinitiveGuide/PartV/CodingGuideLines/PHP.html>

### HTML conventies

De coding conventies die wij tijdens dit project gebruiken zijn te vinden op de bron hieronder:Bron: <https://www.w3schools.com/html/html5_syntax.asp>

### CSS conventies

De coding conventies die wij tijdens dit project gebruiken zijn te vinden op de bron hieronder:

Bron: <https://developer.wordpress.org/coding-standards/wordpress-coding-standards/css/>

### Prettier formatting

Alle code word automatisch door “Prettier” geformateerd. Meer informatie is op de onderstaande bron te vinden:

Bron: <https://www.jetbrains.com/help/phpstorm/prettier.html#ws_prettier_install>

### JSON-conventies

* De variabele moeten leesbaar zijn.
* Gebruik dubbel quotes.
* Laat een spaties na de dubbele punt
* Gebruik Snake Case
* Gebruik vier spaties voor inspringen

Voorbeeld JSON file:

{  
 "uuid": "d8e02d56-2648-49a3-bf97-6be8f1204f38",  
 "availability\_zone": "nova",  
 "hostname": "test.novalocal",  
 "launch\_index": 0,  
 "array0": [],  
 "array1": [  
 "low"  
 ],  
 "array3": [  
 "low",  
 "high",  
 "mid"  
 ],  
 "object0": {},  
 "object1": {  
 "value": "low",  
 "role": "some"  
 },  
 "name": "test"  
}  
Bron: <https://docs.openstack.org/doc-contrib-guide/json-conv.html>

## Performance bewaken

Wij bewaken de performance van onze website door de coding practices van het Laravel framework aan te houden. Deze coding practices zijn te vinden op de onderstaande bronnen:

Bron: <https://stackify.com/9-laravel-best-practices-for-building-better-websites/>   
Bron: <https://laravel.com/docs/7.x>

# Technologie

**Programs and coding languages used for development**

|  |  |
| --- | --- |
| **Program** | **Version** |
| PHP | 8.1.15 |
| CSS | 3 |
| HTML | 5 |
| Bootstrap | 5.3 |
| SQL | - |
| VueJS | 3 |

**Server**

|  |  |
| --- | --- |
| **Program** | **Version** |
| Samba | 4.15.13 |
| MySQL | 8.0.32 |
| phpMyAdmin | 8.0.32 |
| composer | 2.5.4 |
| PHP | 8.1.2 |
| Laravel | 9.0.2 |

## 

## Front-end

De front-end zal geschreven worden in Html, CSS en VueJS

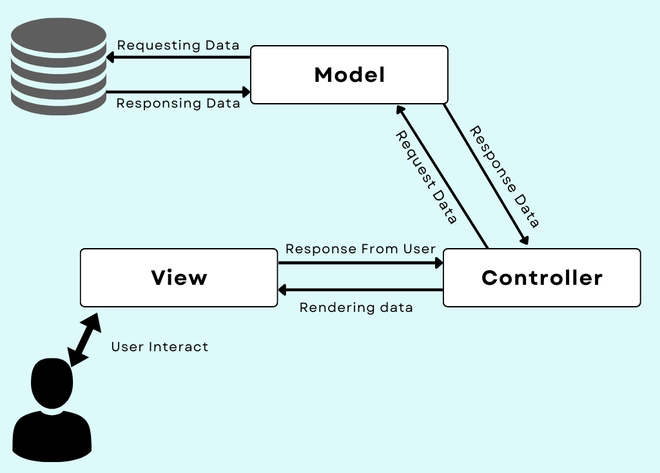
## Backend

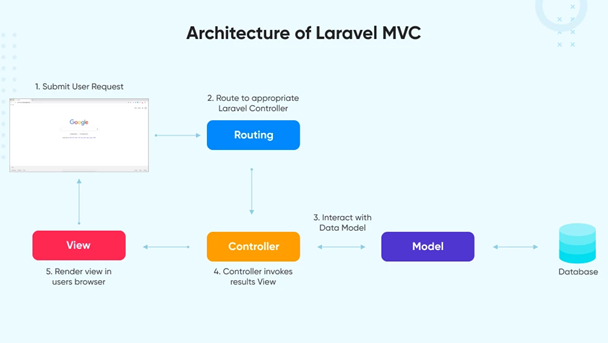
De back-end zal medemogelijk gemaakt worden door Laravel, Laravel is een web applicatie framework.

## Database

De database zal een SQL database zijn, wij gebruiken hiervoor dan PhpMyAdmin om wijzigingen te maken.

Hieronder twee afbeelding van de architectuur van Laravel MVC

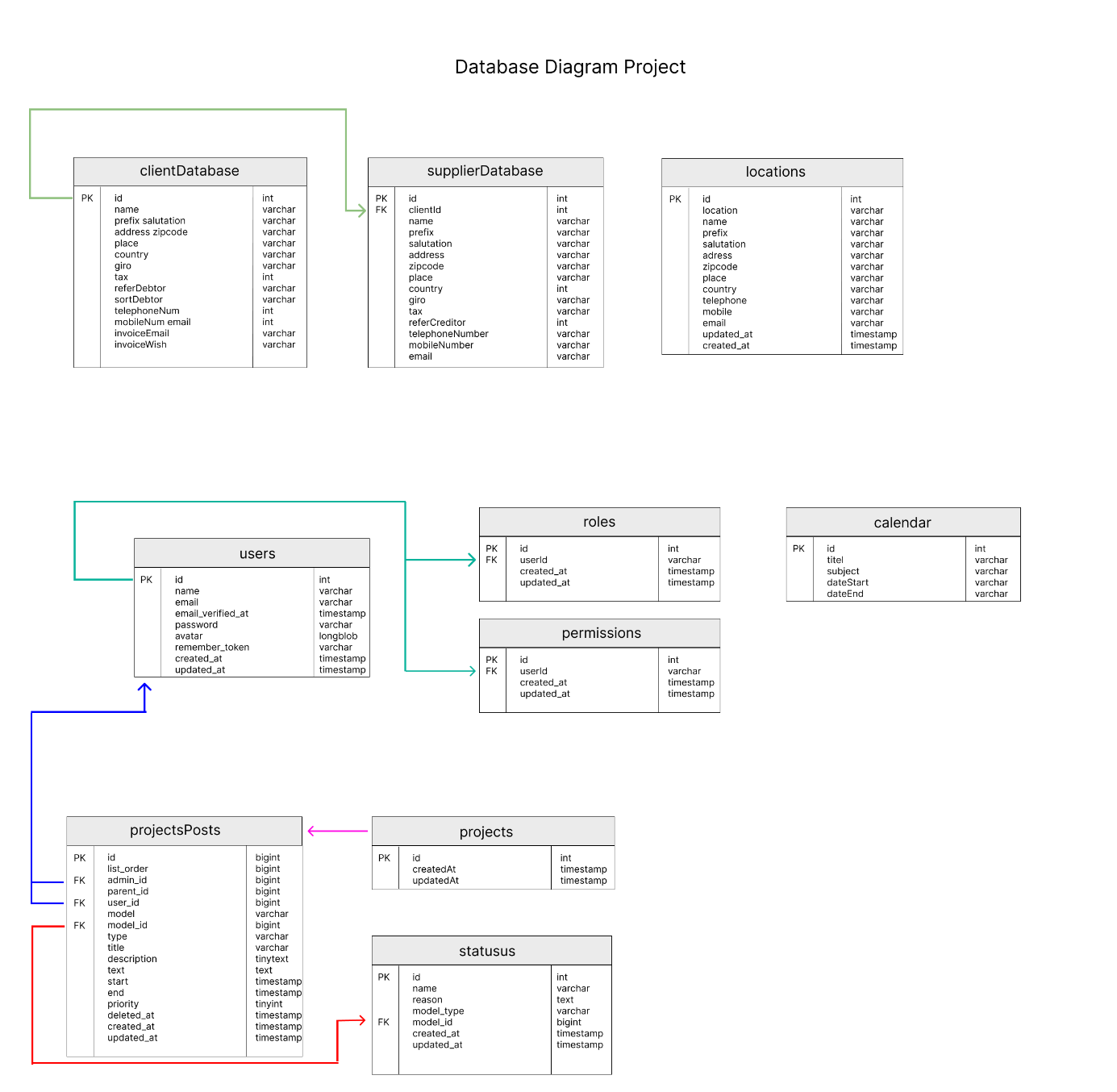




Meer informatie is terug te vinden in de bron hieronder:

<https://blog.pusher.com/laravel-mvc-use/>

# Datamodellen



# Security

De applicatie zal een inlog systeem hebben voor medewerkers. Uiteraard zullen er administratoren zijn.

De wachtwoorden van gebruikers zullen worden vergrendeld achter hashing. Dit is om de veiligheid van de gegevens te garanderen als de database gehackt wordt.

Ook zal er een twee stap authenticatie zijn waarbij medewerkers niet in het systeem kunnen komen zonder deze authenticatie succesvol te voltooien. Voor deze twee stap authenticatie zal de Google Autenticator app gebruikt worden.

Medewerkers zullen een eigen account krijgen om mee in te loggen maar de optie om er zelf één aan te maken is er ook. Accounts met het administrator rol kunnen niet gecreëerd worden door een simpelweg een account aan te maken in de sign up. Wel kan een administrator deze rol toevoegen aan het betreffende account op de rollen pagina.   
  
Ook zullen werknemers van bepaalde afdelingen alleen toegang krijgen tot informatie, leveranciers en klantgegevens met betrekking tot hun afdeling.