

# Software Documentatie Web

Thomas van der Molen

Filters

Containing the word:

	Table	Action	Rows	Type	Collation	Size
<input type="checkbox"/>	comment	Browse  Structure  Search  Insert  Empty  Drop	0	MyISAM	utf8mb4_unicode_ci	1.0 KiB
<input type="checkbox"/>	migrations	Browse  Structure  Search  Insert  Empty  Drop	4	MyISAM	utf8mb4_unicode_ci	2.2 KiB
<input type="checkbox"/>	role	Browse  Structure  Search  Insert  Empty  Drop	0	MyISAM	utf8mb4_unicode_ci	1.0 KiB
<input type="checkbox"/>	run	Browse  Structure  Search  Insert  Empty  Drop	5	MyISAM	utf8mb4_unicode_ci	3.3 KiB
<input type="checkbox"/>	user	Browse  Structure  Search  Insert  Empty  Drop	2	MyISAM	utf8mb4_unicode_ci	2.2 KiB
	5 tables	Sum	11	MyISAM	latin1_swedish_ci	9.7 KiB

Project gegevens	
Projectleden	Thomas van der Molen (456905)
Projectnaam	PlatformSpeedrunner
Versie	1.3

## Inhoud

Versiebeheer .....	2
ERD .....	3
Klassen diagram .....	4
Laravel .....	5
structuur .....	5
Query Builder .....	6
Vulnerabilities .....	6
Limitaties .....	6

## Versiebeheer

Versie	Datum	Verandering
1.0	15-04-2021	Eerste versie.
1.1	16-04-2021	Document gevuld met userstories, requirements, MoSCoW, Visuele voorbeelden.
1.2	01-06-2021	Meer Laravel documentatie geschreven.
1.3	04-06-2021	ERD aangepast en klassendiagram toegevoegd.

## ERD

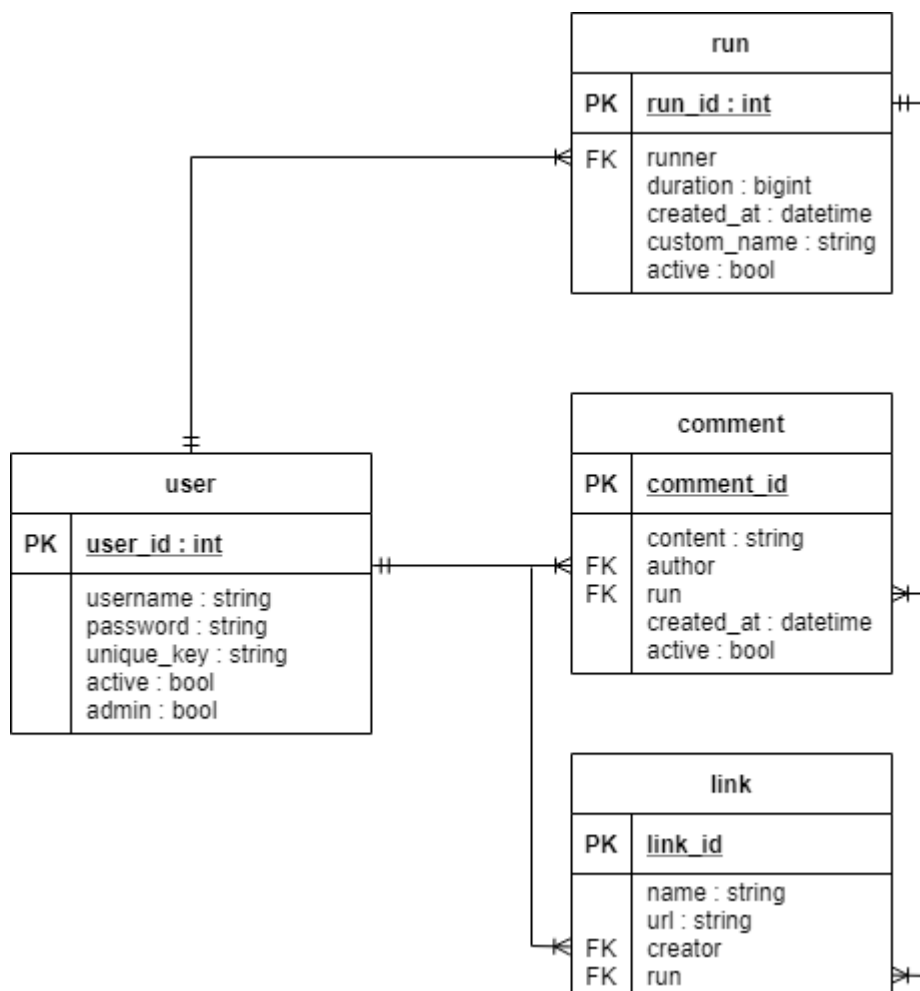
Voor mijn web gedeelte zal ik een database nodig hebben, deze database houdt alle informatie bij die mijn systeem nodig heeft om te functioneren met de requirements die mijn systeem heeft.

Deze database bestaat uit een centrale user, deze user heeft met namen een unique\_key, deze unique\_key is meegegeven vanuit de game. Deze key is hetzelfde bij de user als bij een json file in het spel, hierdoor hoeft de speler niet bij de game in te loggen en wordt deze key als een link gebruikt tussen het account en het spel.

Verder heeft iedere user een admin boolean, deze boolean zal aangeven wat voor user het is (admin of gebruiker). De role zal in het web gecontroleerd worden voor bepaalde mogelijkheden zoals naar de Data Management tab te mogen gaan.

Verder kan een run veel comments bevatten, deze comments worden weergegeven door de schrijver's username en de user die het geschreven heeft zal ook altijd zijn comment kunnen aanpassen of verwijderen.

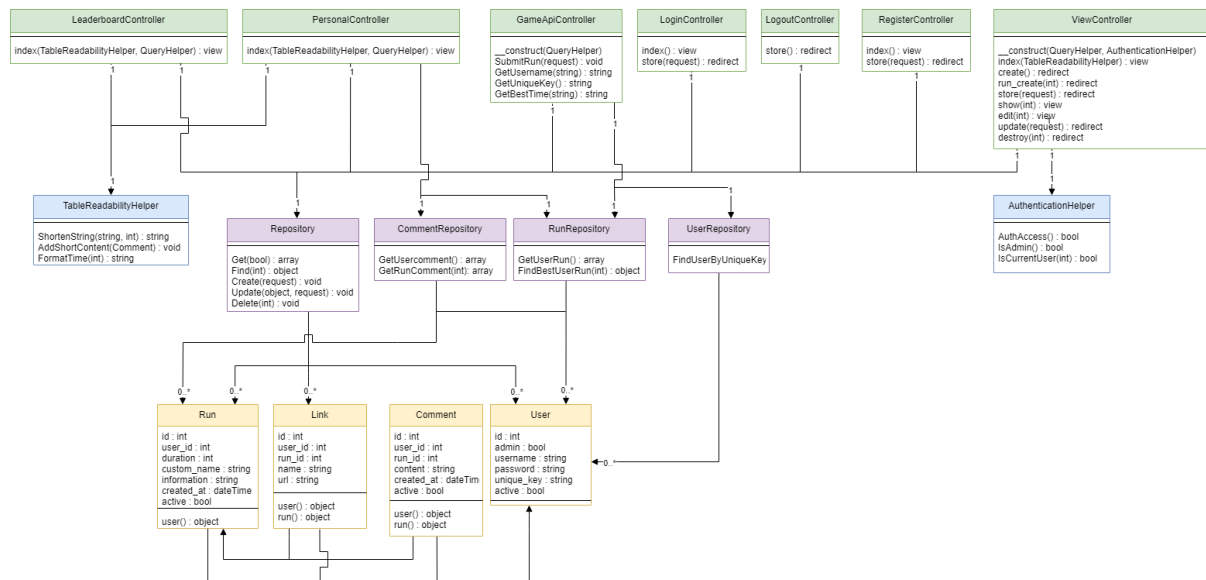
Verder houd een run ook andere handige informatie bij die gebruikers graag zouden willen hebben, deze informatie kan door de runner toegevoegd worden zoals extra informatie of links naar bijvoorbeeld videos van de run zelf.



## Klassen diagram

Binnen mijn web project maak ik gebruik van het framework Laravel en voornamelijk CRUD's voor datamanagement, hierdoor heb ik verschillende controllers die de CRUD functionaliteit bijhouden.

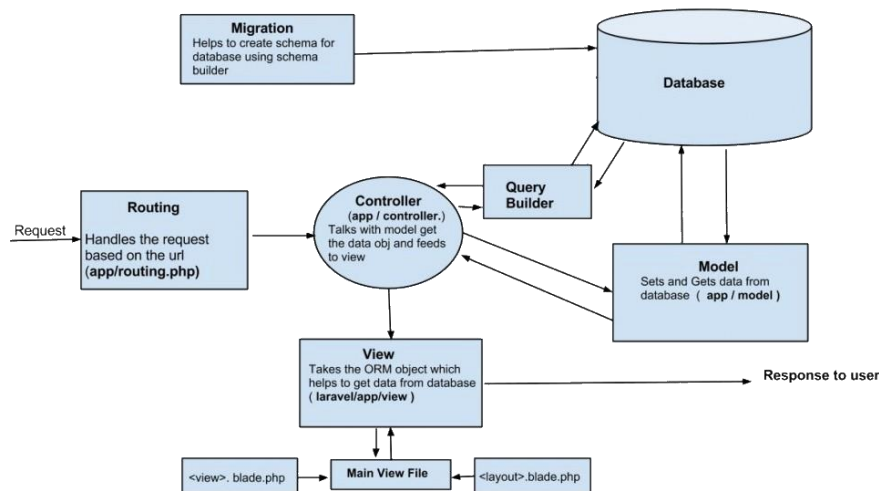
Om met mijn database te communiceren heb ik een abstractie laag toegevoegd tussen de controllers en mijn models, Hierbij kunnen de normale database queries die voor alle models van toepassing zijn gevonden worden in de Repository class, terwijl meer specifieke queries zich bevinden in andere repository classes zoals de UserRepository.



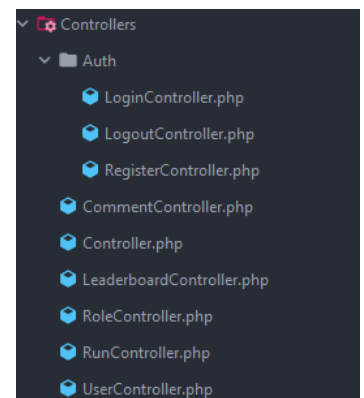
# Laravel

## structuur

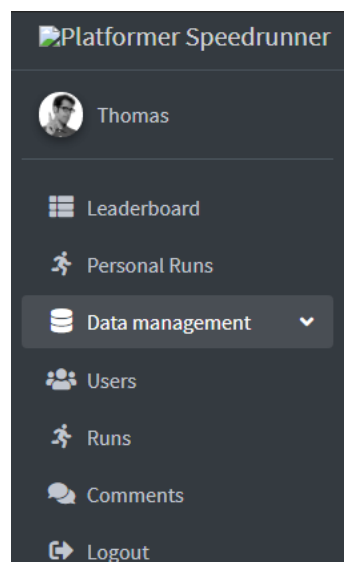
Voor mijn websysteem maak ik gebruik van Laravel als mijn framework en CRUD's voor de data visueel bij te kunnen houden. Laravel maakt gebruik van controllers, models en views voor het bijhouden en tonen van data. Hieronder is een diagram te zien van hoe het websysteem hier mee omgaat.



Migrations worden gebruikt om de tabellen in de database zelf aan te maken. Verder zal nadat een tabel is aangemaakt een Model gebruikt worden om deze informatie bij te houden, door bijvoorbeeld in een model te zetten welke waardes massaal aanpasbaar zijn en welke velden verplicht zijn bij het aanmaken van data. Hierna heb je de Controllers, een controller is het logica gedeelte van iedere pagina en CRUD. In de controller worden functies uitgevoerd zoals het verwijderen van data, of bijvoorbeeld het sorteren van een lijst van runs voor op een leaderboard. In mijn project heb ik voor iedere CRUD een controller, ook zijn er controllers voor het aanmaken en onderhouden van users en is er een aparte controller voor speciale logica bij de leaderboard.



Verder worden views gebruikt voor het tonen van de pagina's. Iedere pagina heeft zijn eigen views, zoals een edit pagina voor users. Veel van deze pagina's hebben wel dingen met elkaar gemeen zoals de sidebar, deze worden daarom in een aparte blade bijgehouden en ingeroepen bij de pagina.



## Query Builder

Binnen Laravel heb je de mogelijkheid om de Query Builder te gebruiken. De Laravel Query Builder geeft je de mogelijkheid om gemakkelijk database queries te maken en uit te voeren. Verder werkt de Query Builder ook snel, aangezien het alleen een query string aanmaakt en deze zal uitvoeren, hierdoor zal de Query Builder bijna net zo snel zijn als direct een database te bevragen via SQL.

## Vulnerabilities

De Query Builder maakt gebruik van PDO-parameter binding, dit helpt tegen mogelijke SQL injections. Laravel bereikt dit onder anderen door prepared statements. Als bijvoorbeeld een extra veld wordt toegevoegd aan een form voor een mogelijke SQL-query, zal dit niet werken omdat de query zelf niet deze SQL zal uitvoeren en zal dit gewoon als tekst worden bijgehouden.

Wel is het belangrijk aan te merken dat de raw queries binnen Laravel niet sanitized worden en zal dit dus niet door Laravel beschermd worden, hier zal het dus aan de programmeur liggen om het veilig te houden.

Mass Assignment is nog een plek waar de veiligheid van Laravel zwakker wordt, Binnen de ORM kun je gemakkelijk massaal de waarde van een table aanpassen. Dit kan problemen veroorzaken wanneer je bijvoorbeeld admin rechten bijhoudt binnen een user table. Je kunt jezelf hier wel tegen beschermen door in de Model dit goed op te stellen, hierdoor blijft het wel aan de programmeur om het systeem veilig te houden.

Verder zijn er momenteel op vulnerability databases zoals [snyk.io](https://snyk.io) geen bekende vulnerabilities gevonden binnen de Laravel Query Builder.

## Limitaties

Binnen de Laravel Query Builder kun je jammergenoeg niet directe relaties opzetten tussen tables, dit zorgt ervoor dat je meerdere queries moet doen om de juiste data te krijgen. Deze limitatie is opgelost in Laravel's Eloquent hier kun je wel relaties opzetten tussen tables.

Er zijn verschillende limitaties die door Laravel's Eloquent wordt opgelost, dit betekent wel dat er meer gebruik gemaakt zal moeten worden van dit systeem en dit zal met een performance verlies komen aangezien Eloquent niet zo snel kan zijn als een SQL query of de query builder op zichzelf.