

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	1.0

Inhoud

Versiebeheer.....	2
ERD	3
Klassendiagram.....	4

Versiebeheer

Versie	Datum	Verandering
1.0	15-04-2021	Eerste versie.
1.1	16-04-2021	Document gevuld met userstories, requirements, MoSCoW, Visuele voorbeelden.

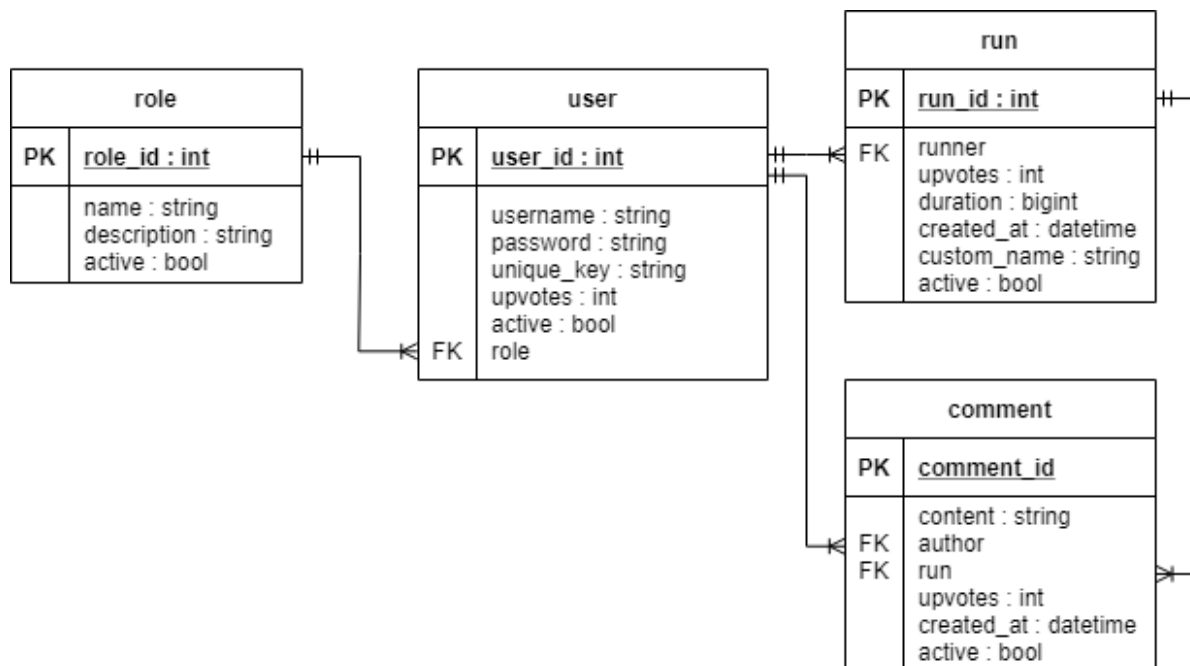
ERD

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.

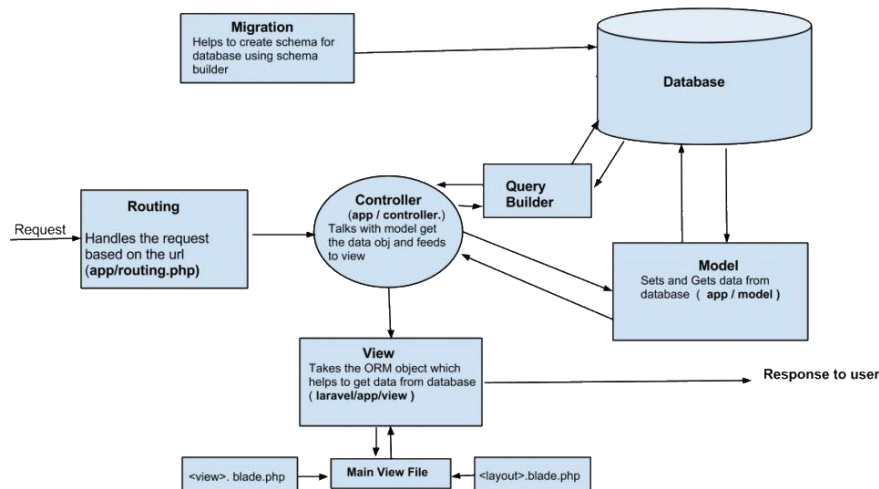
Verder heeft iedere user een role, deze role zal aangeven wat voor user het is bijvoorbeeld: Admin of gebruiker. De role zal in het web gecontroleerd worden voor bepaalde mogelijkheden zoals naar de Data Management tab te mogen gaan.

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.

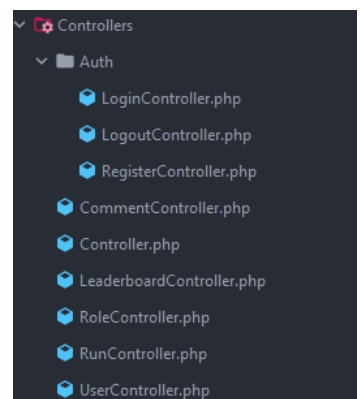


Klassendiagram

Voor mijn web systeem 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 web systeem 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 de model te zetten welke waardes massaal aanpasbaar zijn en welke velden verplicht zijn bij het aanmaken van een 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 in ingeroepen bij de pagina.

