# Reflectieverslag: Ledenadministratie

Naam: Gerard de Way

Studentnummer: 316295531

Datum: 28-10-2024

### Gebruikte tools

* Laravel
  + Eloquent ORM
  + Database: Migrations

### Database

* MySql
* Createscript.sql
* php artisan migrate:fresh --seed

### Beschrijving werking applicatie:

De applicatie is gestructureerd volgens de volgende logica.

Publiek beschikbare login en registratie pagina’s waar een gebruiker kan inloggen en een nieuwe account aanmaken. Ongeautoriseerde gebruikers worden verwezen naar de login pagina en acties binnen de pagina’s worden op basis van rol weergeven.

Een overzicht pagina voor de families waar families aangemaakt, bekeken of verwijderd kunnen worden. Bij het bekijken van een familie kan deze bewerkt worden en familieleden toegevoegd en verwijderd worden. Hier kan ook genavigeerd worden naar een van de familieleden waar deze bewerkt kan worden.

Alle geautoriseerde gebruikers kunnen families en familie leden bekijken. Als een gebruiker zich registreert in de applicatie krijg deze de rol user, welke alleen over lees rechten beschikt.

Als de gebruiker de rol ‘secretary’ heeft kan deze families en de onderliggende familie leden aanmaken, bewerken en verwijderen. Als de gebruiker de rol ‘treasurer’ heeft kan deze het totaalbedrag van contributies in het overzicht van families zien en contributies voor familieleden toevoegen in het familielid aanpas scherm.

Contributies worden op familie lid niveau toegevoegd, waarbij de ‘treasurer’ een boekjaar kan selecteren en een member type. Validatie wordt hier toegepast of het familie lid de juiste leeftijd en member type combinatie heeft en of er niet al contributie voor dit familie lid is toegevoegd voor dit boekjaar.

Aangezien contributies de waarde ontlenen aan de combinatie van boekjaar, leeftijd en member type, is het aanpassen van de contributies niet mogelijk. Het verwijderen van een contributie kan in het scherm van familie lid.

# Reflectieverslag:

## Inleiding

Tijdens het ontwikkelen van een ledenadministratie-applicatie in het veelgebruikte PHP-framework Laravel ben ik tegen verschillende uitdagingen en leermomenten aangelopen.

## Voorbereiding en Planning

In de beginfase had ik moeite met het doorgronden van de opdracht en had ik een overbodig complexe structuur opgezet voor het project. Het opzetten van factories, migrations en database seeders bleek uitdagend. De planning verliep verder relatief soepel, maar al snel realiseerde ik me dat de structuur, zoals ik die in mijn hoofd had, enkele problemen gaf met betrekking tot de eisen van de casus.

## Ontwikkelingsfase

De belangrijkste uitdaging in deze fase was het genereren van testdata. Ik stuitte al snel op problemen met het genereren van de data en de onderliggende verbanden.

Routing heb ik opgezet conform de routing die Laravel biedt. Hierbij heb ik publieke en geautoriseerde acties gescheiden door middel van het auth middleware-component. De ongeautoriseerde gebruiker wordt naar de loginpagina geleid. Aangezien gebruikers wel bij alle pagina's mogen komen, wordt de rolgebaseerde autorisatie op de acties zelf uitgevoerd.

Onderliggende verbanden tussen entiteiten worden gelegd door gebruik te maken van de object-relational mapper Eloquent van Laravel. Deze verbindt de relaties onderling en levert de verbanden uit via de belongsTo en hasMany methodes.

De businesslogica van de applicatie wordt uitgevoerd door verschillende modelklassen, waarbij deze publieke functies beschikbaar stellen voor berekeningen van contributies en leeftijd.

De CRUD-operaties van de families en de familieleden worden beheerd door controllers. Het saniteren van POST-data voor de verschillende acties wordt uitgevoerd door gebruik te maken van de validate-methode van het Request-object van Laravel. In de controllers worden ook de database-interactie-methodes van Eloquent aangeroepen om de verschillende CRUD-operaties veilig uit te voeren.

## Problemen en Oplossingen

Tijdens de ontwikkeling liep ik tegen enkele problemen aan:

Door mijn keuze om vroeg in het proces migraties en database seeders te implementeren, liep ik tegen problemen aan bij aanpassingen aan modelverbanden. Dit resulteerde in meerdere aanpassingen aan de datageneratiemethodes en bijbehorende foutmeldingen. Dit heb ik opgelost door eerst de modelverbanden recht te trekken en hierna pas migraties en seeding te implementeren.

Het implementeren van modellen volgens de standaarden van Laravel leverde een aantal problemen op. Door een verkeerde interpretatie van model-eigenschappen en de gerelateerde databasekolommen kwam ik verschillende lastige foutmeldingen tegen.

Ondanks de verkeerde implementatie van Eloquent-model eigenschappen leek de applicatie ogenschijnlijk correct te werken. Hierdoor kwamen tijdens het ontwikkelen van de models, controllers en views meerdere lastige problemen naar voren. Dit heb ik kunnen oplossen door de eigenschappen uit de modellen te halen, zoals conform is voor Eloquent-entities.

Het routeren van de requests met Laravels routing verliep, buiten enkele problemen met de parametrisering van de routes, redelijk goed.

## Leerpunten

Dit project heeft mij veel geleerd over de werking van Laravel en hoe dit framework concepten zoals object-relational mapping, authenticatie en migraties afhandelt. Hier zijn enkele van de belangrijkste leerpunten:

Dieper inzicht in object-relational mapping (ORM): Doordat ik een verkeerde interpretatie van het ORM had geïmplementeerd, kwam ik achter de vele verschillende manieren waarop een ORM verbonden zit in een MVC-applicatie.

Dieper inzicht in Laravel: Aangezien dit de eerste keer was dat ik een Laravel-applicatie moest opzetten, heb ik een goed inzicht ontwikkeld in hoe het framework in elkaar zit en de sterke en zwakke punten.

## Conclusie

Het ontwikkelen van de ledenadministratie-applicatie bracht veel nieuwe inzichten in bestaande kennis en leidde tot een beter begrip van het veelgebruikte framework Laravel en de bijbehorende technieken.