KU LEUVEN kulak

Bachelor Informatica
Informaticawerktuigen
Prof P. De Causmaecker - J. Goemaere

Verslag - Practicum Informaticawerktuigen

Academiejaar 2020 – 2021

Thibaut Deliever - Faisal Ahktar - Leander Deweerdt

Inleiding

In dit verslag worden verschillende aspecten van het practicum besproken en verklaard. Het verslag is opgedeeld in meerdere delen, die elk ingaan op specifieke onderdelen van de opdracht:

- **Minimale vereisten**: Hier wordt nagetrokken dat alle minimale vereisten voldaan zijn, met eventuele bijhorende locaties/url-routes.
- **Struikelblokken** In deze sectie worden enige moeilijkheden, vertragingen en andere struikelblokken omtrent het maken van het practicum gepresenteerd.
- **Opdracht in detail**: Hier wordt één opdracht, samen met zijn werking, binnen het persoonlijk perspectief overlopen worden, alsook 1 opdracht binnen het groepsperspectief.
- **Extra implementatie's** Tot slot worden de extra functionaliteiten van de database opgelijst.

1 Minimale vereisten

1.1 Persoonlijk perspectief

- De gebruiker kan zich **registreren**, **aanmelden** en **uitloggen**.
  ⇒ ∼/Register & ∼/Login
- De gebruiker kan zijn **persoonlijke data** raadplegen.
  ⇒ ∼/Account
- De gebruiker kan een **activiteit toevoegen**. Deze activiteit kan zowel wandelen, fietsen als lopen zijn. De datavelden die elk van deze activiteiten bevatten, zijn: Afstand, tijdsduur, datum + tijdstip, calorieën en sportuitrusting.
  ⇒ ∼/Account/CreateActivity
- De gebruiker kan een **overzicht** van zijn meest **recente activiteiten** terugvinden. Deze staan zowel op de dashboard- (laatste maand) alsook op de activiteiten-pagina (volledig overzicht).
  ⇒ ∼/Account & ∼/Account/Activity
- De gebruiker kan een **maandelijkse evolutie** van elke sport afzonderlijk terugvinden op zijn dashboard-pagina. Het bereik van deze grafiek zal lopen van 1/jan/yyyy t.e.m. 31/dec/yyyy. Bijgevolg zal de gebruiker op 1 januari een 'reset' zien van deze data.
  ⇒ ∼/Account

1.2 Groepsperspectief

- De gebruiker kan zich **toevoegen aan een bestaande groep** of een **nieuwe groep aanmaken**. Wij zijn echter van het principe vertrokken dat een groep iets exclusief is en dat een gebruiker kan toegevoegd worden enkel en alleen door iemand intern uit een bestaande groep. Elke gebruiker binnen een bestaande groep heeft dus de mogelijkheid om een andere gebruiker toe te voegen.
  ⇒ ∼/Account & ∼/Account/Group
- De gebruiker kan zijn **sportprestaties vergelijken** binnen de groepen waarin hij/zij aangesloten is. Hierin zitten alle mijlpalen per sport verwerkt. Deze ranking is terug te vinden op de dashboard, alsook op de groepspagina van de gebruiker.
  ⇒ ∼/Account & ∼/Account/Group

2 Struikelblokken

Over het algemeen zijn er niet veel problemen opgedoken binnen de implementatie van het practicum. De enigste struikelblokken vormden zich bij de start van het MVC-systeem, de grafieken en de onderlinge samenwerking.

2.1 MVC-Systeem

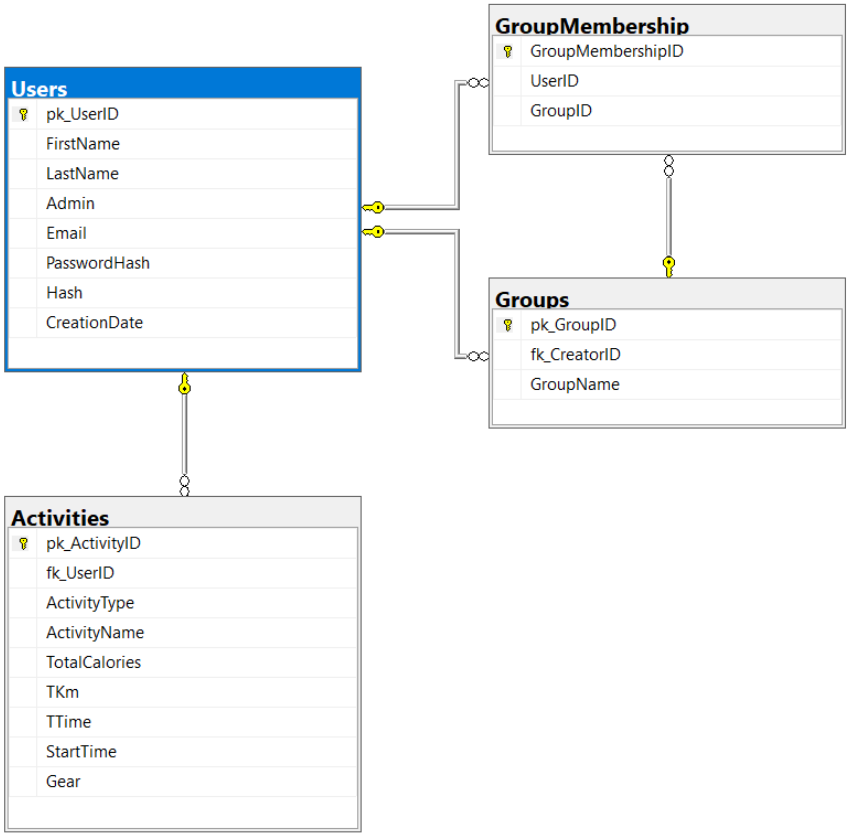
Wij waren allemaal onbekend met het MVC-systeem waardoor het practicum moeilijk uit de startblokken kon schieten. Na veelvuldig gebruik van de documentatie op de Microsoft-site alsook Stackoverflow vonden we onze weg hierin.

2.2 Grafieken

De grafieken van chart.js waren matig/onduidelijk gedocumenteerd naar onze mening. Hier hebben we dus ook even moeten proberen om alles te doen laten werken. Binnen het practicum hebben we ook geprobeerd om op dynamische manier grafieken te creëren wat na enige tijd ook werkte.

2.3 Database

De database was voor ons geen probleem. Hieronder kan u een overzicht van onze database-implementatie terugvinden. Echter bij het invoegen van een nieuwe groupmembership ondervonden we problemen bij het niet gebruiken van een primary key. Hierdoor hebben we de groupmembership-tabel geïmplementeerd met een primary key, zonder deze ergens te gebruiken.



2.4 Onderlinge Samenwerking

De grote meerderheid van de code werd geschreven door Thibaut. Wegens het maken van weinig afspraken en indeling van de taken, verwaterde de samenwerking behoorlijk. Hierdoor was de inbreng voor het practicum eenzijdig. Ook de communicatie tussen alle partijen was eentonig.

3 Opdracht in detail

3.1 Persoonlijk perspectief

De gebruiker zal na het inloggen op zijn dashboard terecht komen. Om deze pagina te verkrijgen, zal het volgende gebeuren binnen de applicatie:

1. Nadat de persoon op de inlog-knop duwt, zal zijn data (email, wachtwoord) gevalideerd worden binnen de 'validate()' functie, die zich bevindt in 'HomeController.cs'.
2. We gaan de persoon eerst zoeken binnen onze Sjette DB op basis van zijn uniek e-mailadres. Indien we een geldig e-mailadres hebben, zullen we op basis van het gevonden object de niet-geëncrypteerde versie van de gebruikers wachtwoord, encrypteren. Na de encryptie vergelijken we het berekende wachtwoord met het paswoord dat we terugvinden binnen het object. Als dit een match is verwijzen we door naar de Account-pagina waar het dashboard zich bevindt.
3. Deze redirect zal verwijzen naar de 'Index()' functie die zich bevindt in 'AccountController.cs'.
4. De eerste stap binnen 'Index()' is achterhalen, aan de hand van het token, welke gebruiker zich heeft ingelogd. Op basis van het gevonden ID kunnen we verschillende SQL-Query commando's uitvoeren om de nodige data te verkrijgen.
5. Deze data geven we mee aan een 'DashboardData.cs' klasse die enkel gebruikt wordt als Model binnen de dashboard-pagina. Binnen deze klasse zal ook alle data berekend worden; ranking, grafiekdata etc.
6. Deze klasse wordt meegegeven aan de View, waar het Model geïmplementeerd is en de data weergegeven wordt.

3.2 Groepsperspectief

Als de gebruiker sportprestaties wilt vergelijken met zijn vrienden binnen een bepaalde groep vanuit zijn dashboard, zal achterliggend het volgende gebeuren:

1. De gebruiker navigeert via de navigatiebalk bovenaan naar de groep-pagina.
2. Door op deze knop te duwen zal de gebruiker de functie 'Group()' geactiveerd hebben binnen de controller 'AccountController.cs'.
3. De eerste stap binnen 'Group()' is achterhalen aan de hand van het token welke gebruiker zich heeft ingelogd. Op basis van het gevonden ID kunnen we verschillende SQL-Query commando's uitvoeren om de nodige data te verkrijgen. We genereren alle mogelijke data in 1 maal, dit om vele POST en GET methodes te vermijden. Er kan dynamisch gewisseld worden tussen de data op de pagina zelf.
4. Deze data geven we uiteindelijk mee aan een 'GroupData.cs' klasse die enkel gebruikt wordt als Model binnen de groep-pagina. Binnen deze klasse zal ook alle data berekend worden zoals ranking, grafiekdata etc.
5. Deze klasse wordt dan meegegeven aan de View, waar het Model geïmplementeerd is en de data weergegeven wordt.

4 Extra implementatie's

Naast de minimumvoorwaarden en minimumfunctionaliteiten hebben wij ook wat extra implementaties gerealiseerd:

- Een gebruiker kan zijn eigen instellingen aanpassen. Dit gaat van persoonlijke info tot wachtwoorden.
  $\Rightarrow$ ∼/Account/Settings
- De wachtwoorden van onze gebruikers worden geëncrypteerd door middel van automatisch gegenereerde salts. Voor de demo-data hebben we echter wel eenzelfde salt gebruikt over de dummies, zodat we voor ieder account hetzelfde wachtwoord encrypteren.
- De site heeft het gedeelte met de gebruikersgegevens afgeschermd tegen 'unauthorized activities'. Concreet houden wij een cookie bij die gevalideerd wordt bij elke ingeladen pagina. Deze cookie zal van zichzelf vervallen. Indien een ongeldig token waargenomen wordt, zal de gebruiker een verwijzing krijgen naar de login. Via dit token houden wij ook bij welke persoon op dat moment ingelogd is om zo de correcte data weer te geven aan onze gebruikers.
- De oprichter van een sportgroep heeft de mogelijkheid om mensen uit de groep te zetten. Daarnaast kan hij ook de admin-rechten doorgeven aan een persoon intern binnen de groep. Indien de oprichten een groep wilt verlaten waar hij/zij admin-rechten op heeft, dient hij deze eerst door te geven aan een andere persoon.
- Op het dashboard en binnen de groeppagina hebben we verscheidene (al dan niet dynamische) grafieken. Door te klikken op het hoofdding van de grafiek kan de gebruiker andere data verkrijgen.
- Het grafisch aspect was voor ons een must zodat we de voor een kwalitatieve gebruikservaring kunnen zorgen.