**Groepswerk 2: Analyse**

*Geschreven door Hande*

**INTRODUCTIE:**

In het begin van het groepswerk zijn Grishian en ik samengekomen om eerst alles eens te analyseren. Wat hebben we nodig, hoe gaan we dit aanpakken, wat moet er allemaal in de applicatie zitten, hoe beginnen we hieraan? Om een antwoord te krijgen op deze vragen zijn we eerst begonnen met alles op te sommen om een duidelijk overzicht te krijgen, met als resultaat de volgende notities hieronder.

**ONDERDELEN VAN HET PROGRAMMA:**

* Klantenbestand of gebruikersbestand bestaande uit gebruikersnaam, e-mailadres (login), geencrypteerd wachtwoord, activiteit en of ze een administrator of gebruiker zijn.
* Boeken (producten) allemaal samen in een tabel en verschillende tabellen bestaande uit fysieke boeken, e readers en audioboeken (met overerving)
* Boeken gaan bestaan uit titel, ISBN-nummer, Genre, prijs, auteur, taal, reeks/series, pagina’s en lengte, duur of grootte
* Sorteerfunctie op alfabetische titel en prijs, filterfunctie op genre en boektype
* Wishlist linken met klantenbestand
* Klantenlogin
* Beheerderlogin
* Login/wachtwoord reset

**GEBRUIK:**

* Een GIT repository (<https://github.com/grishian/groepswerk2.git>)
* Flask WTForms ([WTForms — WTForms Documentation (3.0.x)](https://wtforms.readthedocs.io/en/3.0.x/))

**FAQ**:

In welke taal moet het programma staan? Engels of Nederlands?

* Maakt niet uit. Zorg wel dat alles in dezelfde taal is.

Moeten klanten login en beheer login ook in de databank gestoken worden?

* Daar zijn meerdere manieren voor. Je kan ze afhankelijk maken van je klantenbestand of ook niet. Dat wordt een klasse ‘user’. Meer wordt hierover gezien komende maandag (31 Januari 2022).

Moeten er aankopen geprogrammeerd worden?

* Nee

Geen shopping cart?

* Nee. De shopping cart fungeert eigenlijk een beetje als wishlist.

Verschillende rollen voor medewerker en manager?

* 1 profiel voor shopeigenaar is genoeg. Verschil tussen klant en beheerder is goed.

Moeten we iets van JSON voorzien?

* Nee

Hoeveel punten staan op analyse? Wat moet daar precies in?

* Als de code onduidelijk is, moet de analyse het wel verduidelijken. Daar mag een woordje uitleg bij.

De inhoud van de databank kunnen we dus niet delen. Dit zal dus enkel een db-design worden?

* Ja. De inhoud maakt niet uit, de werking en design wel.

Gaat de beheerder klanten kunnen verwijderen? Bijvoorbeeld klanten die inactief zijn, etc.

* Die functie mag, maar is niet per se nodig aangezien het ingewikkeld is.

Aangezien de deadline voor het project tegen eind mei is, is het de bedoeling dat we dan ons project constant gaan updaten met de nieuwe kennis en dingen die we nog gaan zien de komende maanden/weken? Of enkel met wat we ongeveer tot nu toe hebben gezien?

* Eigenlijk moet je het project kunnen maken met alles wat we tot nu toe hebben gezien.

**EXTRA TOEWIJZINGEN:**

Als klant moet je kunnen inloggen en browsen, filteren, opzoeken, boeken toevoegen aan je wishlist. Als beheerder moet je kunnen inloggen, boeken toevoegen, verwijderen, etc.

Er gaan klanten zijn die niet in staat zijn zelf hun wachtwoord te resetten, dus stel de optie in om zelf via de commandline als admin het wachtwoord te veranderen/resetten.

‘@with\_appcontext’-command is super belangrijk!

Cookies duration van 30 dagen is meer dan genoeg.

**UITLEG BIJ DIAGRAMMEN:**

Na het begin van de analyse op te starten en alles eens duidelijk op te sommen, onze vragen neer te pennen en de antwoorden hierop op te volgen, kwam Grishian met het idee om diagrammen te maken om nog een beter overzicht te krijgen op dit groepswerk. Tegen onze volgende samenkomst hadden we dit gedaan, dus in het mapje analyse kunt u verscheidene Use Case, Klassen- en Tabellendiagrammen terugvinden. Die zijn hierna ook aangepast doorheen de verloop en de ontwikkeling van de applicatie.

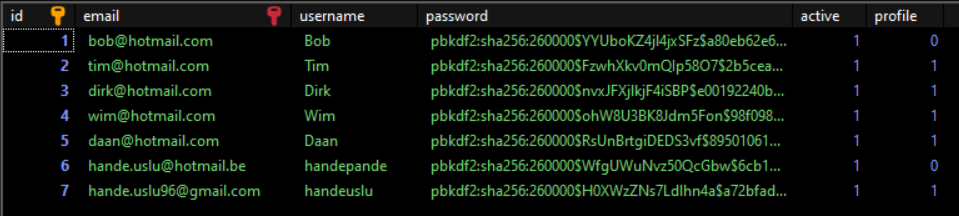
Eerst hadden we gekozen om in totaal 4 tabellen te maken voor de boeken. Drie tabellen voor elk boektype en dan 1 grote tabel waarin deze drie tabellen zouden samenkomen. Later hebben we dit aangepast naar alle boeken gewoon in 1 tabel samen te brengen omdat we het ons anders te moeilijk gingen maken. Simpel is beter.

**UITLEG BIJ APPLICATIE:**

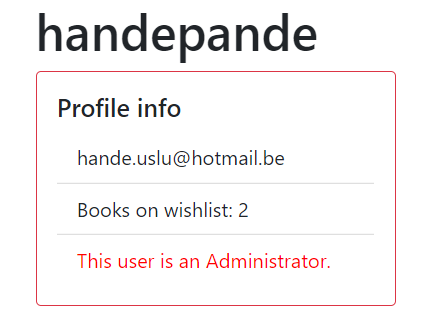
Op de GIT staat een zip-mapje met daarin de database data die je kan importeren, hierin staan al boeken, gebruikers, wishlists, etc.

Je kan als “gast” vrijwillig alle boeken bekijken en opzoeken, wel moet je je registreren om een wishlist aan te maken en die te gebruiken. Zodra je je hebt geregistreerd, kan je inloggen en je wishlist gebruiken. Je kan hier boeken aan toevoegen en verwijderen.

Elke gebruiker die zich registreert wordt automatisch in het systeem gezet als een normale gebruiker of klant. Dit wordt gekenmerkt in de databank in de ‘user’ tabel, in de ‘profile’ kolom met een ‘1’. Wil je dit aanpassen en iemand kenmerken als een “administrator”, dan verander je dit gewoonweg in de databank, de ‘1’ door een ‘0’.



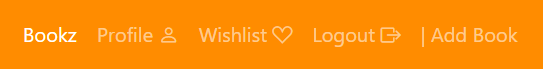
In de bovenstaande afbeelding zie je dus dat Bob en handepande administrators zijn. Hierna kan je dus ook op je profiel zien dat je een administrator bent. In het geval van een normale gebruiker, staat er niets.



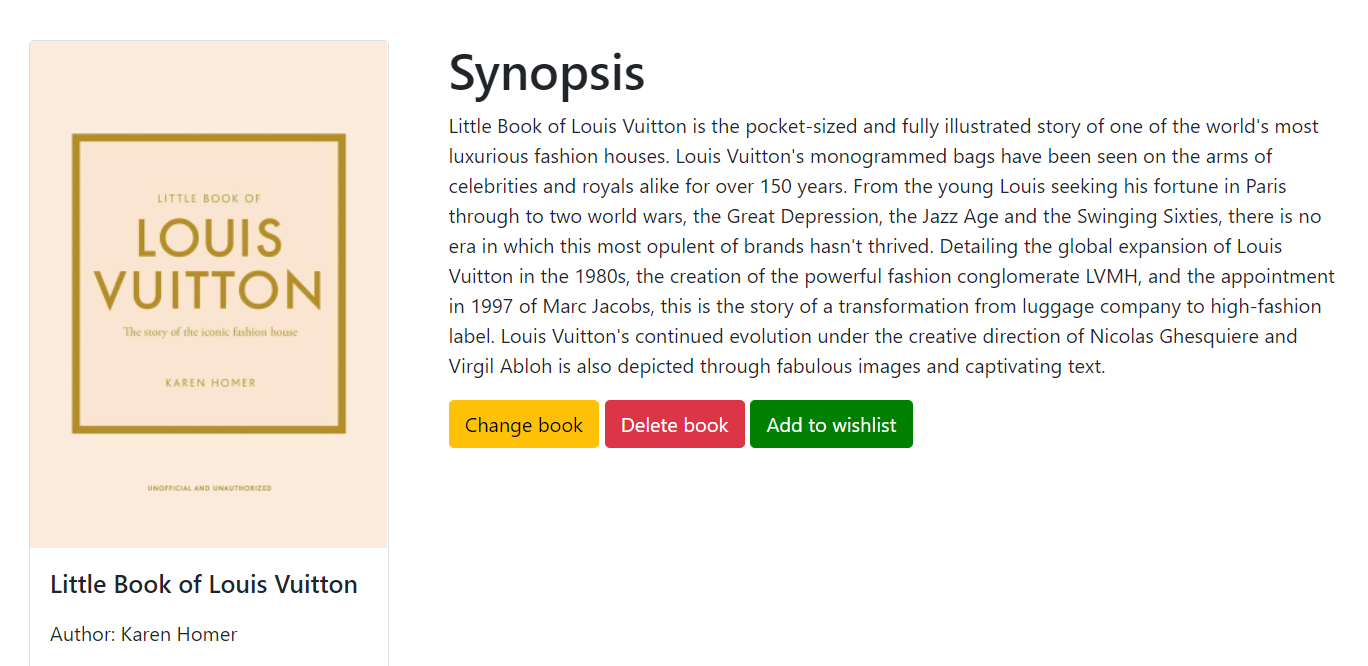
Op je profiel gaat ook een knop staan voor je wachtwoord te veranderen.

Op de hoofdpagina gaan alle boeken (3 boeken per pagina, met nummering voor naar een volgende of vorige pagina te gaan) staan, met de bijhorende sorteer-, filter- en opzoekfuncties. De zoekfunctie werkt op basis van titels. Zodra je een titel of meerdere hebt geselecteerd uit de lijst, klik je op het knopje met het vergrootglas ernaast en dan verschijnen de boeken met de titels die je hebt geselecteerd.

Als je inlogt als een administrator, verschijnt bovenaan de hoofdbalk een extra functie namelijk de knop om boeken toe te voegen. Deze optie verschijnt enkel voor administrators.



Als je als administrator ook boeken wilt verwijderen of aanpassen, selecteer of klik je simpelweg op een boek op de hoofdpagina, en dan verschijnen de verschillende opties naast het boek.



Hier kan je ook een boek toevoegen aan je wishlist.

**AGENDA:**

28 Januari 2022:

Eerste plan van aanpak besproken, document met korte analyse gemaakt van wat we nodig hebben en gaan maken. Dat hebben we dan opgesplitst in wat we gaan coderen in Python en wat we gaan steken in de databank. Ook zijn er vragen bedacht voor het vragenuurtje en hebben we andere interessante vragen van andere groepjes met de antwoorden toegevoegd aan de analyse.

Taak tegen 31 Januari 2022:

We gaan allebei use-case diagrammen maken voor alle klassen en tabellen om die dan te vergelijken. Dit doen we om een goed overzicht te krijgen op wat we gaan doen.

7 Februari 2022:

Diagrammen besproken, aangepast en gefinaliseerd. Besproken hoe we de login gaan aanpakken en verdere aanpassingen. Github opgestart en met behulp van Grishian leren werken met Sourcetree om code te delen.  
Begin pair-programming:

* basisklasse ‘customer’ gemaakt
* database aangemaakt en connectie vastgezet

Taak tegen 17 Februari 2022:

Grishian: klassen ‘customer, admin’ maken

Hande: klassen ‘books, ebooks, audiobooks and physical books’ maken

*…*

(Niet alle data van de bijeenkomsten en taken zijn gedocumenteerd.)