Commerce Procesboek

12 december

**Lecture sql**

Sql herkende ik nog van het vak datarepresentaties, in dit vak hebben we een aantal opdrachten gemaakt over dit onderwerp. Grotendeels was dit wel al weggezakt in mijn geheugen dus de lecture was wel nodig/interessant. Tijdens deze lecture probeer ik gelijk mee te schrijven om alle queries en arguments zo goed mogelijk te onthouden. De vraag die in het begin van de lecture bij me opkomt is hoe we deze tables gaan implementeren met Django in de websites. De uitgebreide SQL syntax blijkt voor het commerce-project minder nodig te zijn om te kennen, daar Django dit voor ons doet door middel van modellen. In models.py, in de map van onze app, kunnen we classes aanmaken voor elke tabel. Deze tabellen kunnen we dan met elkaar verbinden met Foreign keys en zo kunnen we een heel systeem van modellen van data maken. Na het maken van de modellen is het nodig de migrations aan te maken in python. Hiermee wordt de python code omgezet in een sql file, wat we daarna kunnen aanpassen wanneer we python manage.py shell gebruiken. Nu kunnen we met behulp van python onze databases aanpassen, updaten, dingen deleten etc. Met de Django admin app kunnen we hierna op een nog makkelijkere manier data toevoegen aan onze modellen. Verder leren we ook nog hoe we gebruikers aanmaken en login/logout functionaliteiten toevoegen aan de applicatie.

**13 december**

Na het doorlezen van de specificaties, en het documenteren van een vaag/geschetst plan heb ik een stuk meer inzicht in hoe de applicatie er ongeveer uit komt te zien en hoe hij hoort te werken. Het eerste waar ik mee begon was nadenken over hoe de verschillende modules op elkaar aansluiten, en welke velden elke module nodig heeft. De module die al aangemaakt is, User, is gemaakt met als input Abstractuser. Ik moest even op de Django site kijken hoe deze precies werkt. Waar ik achter kwam is dat dit het default user model is van django, maar dat je deze op de manier hoe hij er nu staat altijd later nog kan aanpassen als dat nodig is. Ook moet het model nog toegevoegd worden in de admin.py.

Wat ik nu heb gedaan is in models.py de modellen aangemaakt met de fields waarvan ik denk dat ik ze nodig heb, en de Foreignkeys waarmee ik denk dat het systeem uiteindelijk goed kan werken. Hierna heb ik mijn aanpassingen gepusht naar github en ben ik gaan kijken naar het onderdeel Create Listing. Hiervoor moest ik een aantal forms maken in de createnew.html, en met de post method deze informatie sturen naar een functie in views.py waar de informatie omgezet werd naar data die door django naar sql data gezet kan worden. Telkens als ik dit probeerde kreeg ik een error, maar na een half uur proberen toch maar even in de lecture gekeken. Blijkbaar was ik vergeten mijn veranderingen in modules.py te migraten….. zo kan je blijven proberen natuurlijk. Hierna lukte het wel om een list aan te maken en via de admin kon ik deze zien en ook weer verwijderen.