28/12/2013  
Er werd gestart met uit uitschrijven van de requirements uit de opgave-PDF’s, daar deze voor IOT studenten ietwat anders ligt.

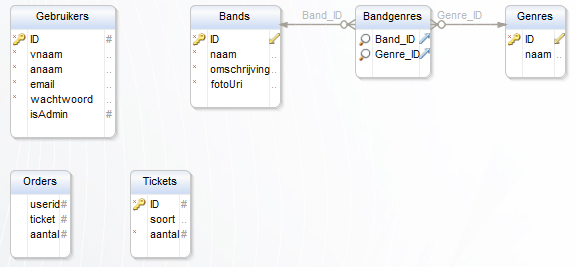
- RSS feed met enkele korte berichten  
- Verloop van het festival is vooraf vastgelegd, niet te wijzigen via de admin webinterface  
- Database best voorzien op podiums-podiumnaam + bands / hun tijdsslot  
- Bij te houden info van bands: naam, korte omschrijving, foto, genres  
- Genre  
- Database op MS SQL (geen local DB)

Gebruikers:  
- Overzicht van de optredens (lineup), foto groep en timing  
- Eenvoudig sorteren per dag / tijdslot van de optredens, details kunnen opvragen  
- Tickets bestellen: naam, voornaam, email, aantal tickets  
- Er is een maximum aantal tickets beschikbaar

Administrators:  
- overzicht van de reservaties te kunnen opvragen  
- username: docent – docentdocent

- Anonieme users?

Na het uitschrijven van de requirements maakte ik gebruik van DbSchema (<http://www.dbschema.com>) om een database diagram uit te werken (en ineens in SQL code te gieten om uit te voeren op m’n lokale database). De database voor deze site is niet gek gecompliceerd.



Mede omdat er in de opgave voor IOT studenten geen wijzigingen hoeven te gebeuren aan de podia/line-up en deze dus statisch op de website mag verschijnen.  
Eveneens genereerde ik ineens wat data om in de database te steken (manueel als test data en/of omdat het niet via de webinterface moet kunnen).

29/12/2013 – 1/1/2014

Er werd doorgewerkt aan een methode om gebruikers te authentiseren en eenvoudig mogelijk te maken een administrator van een niet-administrator te onderscheiden. Na veel pogingen dit met de isAdmin flag te doen voor de users en vergelijken van oplossingen uit MVC3 (waar user management iets makkelijker was), koos ik uiteindelijk voor een vergelijkbare oplossing als in het security labo. De standaard ASP.net template werd gebruikt en omgezet naar een MS SQL database en het toevoegen van extra fields die niet in de standaard template zaten maar wel vereist waren voor de opgave.  
Spijtig genoeg redelijk wat tijd in gestoken & verloren.

Verder werd een pagina met een overzicht van de bands aangemaakt die uit de database wordt opgehaald, een lijst van beschikbare tickets met hun prijs (anonieme gebruikers) en begon ik met uitzoeken hoe ik het probleem kon aanpakken gebruikers max 5 tickets te laten bestellen + tegelijkertijd het aantal beschikbare tickets in de gaten kon houden. Voor admins werd de ticket status pagina aangemaakt waarop het resterend aantal beschikbare tickets per dag te zien is.

De edits met de login leverden hoe dan ook een nieuw database schema op.

