

# Verslag derde iteratie

December 23

# 2011

---

In dit verslag worden de vorderingen van het IKE project van groep 3 besproken. Dit verslag gaat in de op derde iteratie.

Gijs Kuijer

Mathijs Lagendijk

Roeland Oosterloo

## Inhoudsopgave

Inleiding.....	2
Programma van eisen.....	3
Ontwerp .....	4
Implementatie .....	10
Project evaluatie.....	14
Conclusies en aanbevelingen .....	15

## Inleiding

Zoveel mensen, zoveel wensen. Deze spreuk gaat voor bijna alles op, zo ook voor muziek. Altijd al is er een enorme diversiteit aan muziek beschikbaar geweest. Vandaag de dag is dat niet anders. Wat wel is veranderd, is dat de muziek veel toegankelijker is geworden. Doordat bijna alle muziek online te luisteren en kopen is, is de keuze overweldigend geworden. Om iets van orde aan te brengen in deze chaos zijn we een systeem aan het maken dat advies geeft over mogelijk interessante muziek aan de hand van de input van de gebruiker.

De onderzoeksvraag om dit probleem op te lossen is als volgt geformuleerd: Wat is de beste manier om een advies te doen over muziek gegeven een voorkeur voor een bepaalde artiest?

Om dit probleem op te lossen moest om te beginnen kennis worden opgedaan over zogenaamde recommender systemen, systemen die een advies geven. Omdat we zelf niet alle informatie bezitten om iets te doen met een naam van een artiest zijn de beschikbare online datasources onderzocht om te bepalen of deze bruikbaar zijn voor ons onderzoek.

Een van de randvoorwaarde is dat we er van uit gaan dat de gebruiker geen super nieuwe artiesten zoekt. Omdat ons systeem getraind is op een dataset en daarna niet meer bijleert kent deze alleen de artiesten die hem geleerd zijn.

In dit rapport zal als eerst het programma van eisen worden besproken. Daarna komt het ontwerp, gevolgd door de implementatie en als laatste de conclusies en aanbevelingen.

## Programma van eisen

Dit project moet ervoor gaan zorgen dat ook mensen die weinig muziek kennis hebben en niet vaak bij concerten komen, ook nieuwe muziek kunnen vinden die zij leuk vinden. Aangezien mensen er niet van houden om hele pagina's met vragen te gaan beantwoorden om vervolgens eindelijk een advies te krijgen, zal het product dat uit dit project naar voren komt ook simpel en duidelijk moeten zijn. Tevens zal er met weinig informatie van de gebruiker een hoop gedaan moeten worden, zodat de gebruiker altijd iets vindt wat hij of zij leuk vindt. Al deze kenmerken zijn opgenomen in het programma van eisen. Hieronder is een puntsgewijze lijst te vinden met eisen.

- Er moet een duidelijke interface zijn die makkelijk te bedienen is.
- De interface moet weinig informatie nodig hebben en toch goed kunnen functioneren.
- De interface moet snel zijn zodat mensen niet te lang hoeven te wachten en afhaken.
- De interface moet functionaliteit bieden die nu nog niet te vinden is op het web zodat mensen ook naar de website toe getrokken worden.
- De interface moet professionaliteit uitstralen waaruit naar voren komt dat er aandacht is besteed aan het bedenken van het geheel.
- Naast het advies moet er ook makkelijk meer informatie opgevraagd kunnen worden over de artiest, zodat er niet naar een andere website gegaan hoeft te worden om verder te zoeken.
- Ook oudere mensen moeten de website kunnen gebruiken, dit betekent dat er niet te gecompliceerde teksten en functies moeten zijn en alles in rustige stappen verloopt die ook voor ouderen nog te volgen zijn. Ook betekent dit dat de database waaruit advies gegeven wordt ook oudere muziek moet kunnen herkennen.

## Ontwerp

Bij het ontwerpen van ons recommender systeem hebben wij per iteratie een ontwerp van ons systeem gemaakt. Hierdoor hebben wij aan het einde van elke iteratie een volledig eindproduct wat het ontwerp van de desbetreffende iteratie weerspiegelt.

Voordat wij zijn begonnen met het ontwerpen van ons systeem hebben wij eerst overlegd hoe we het systeem zouden gaan maken. Na veel overleg zijn wij tot de conclusie gekomen dat wij het beste een web applicatie kunnen bouwen die gebruikers via een website kunnen gebruiken. Natuurlijk hebben wij ook andere opties overwogen. Bij het maken van bijvoorbeeld een desktop applicatie vonden wij echter het nadeel dat deze applicatie dan eerst gedownload zou moeten worden en vervolgens zou moeten worden geïnstalleerd voordat deze zou werken, terwijl de gebruiker juist snel muziek wil zoeken. Mede om deze reden denken wij dat een web applicatie de snelste manier is voor een gebruiker om muziek te ‘zoeken’.

In de eerste iteratie hebben wij nagedacht over hoe we ons systeem visueel zouden weergeven en hoe wij data zouden gaan vergaren. Wij hebben besloten om ons systeem simpel, maar krachtig in elkaar te zetten. Wij hebben dit gedaan door een zoekmachine te ontwerpen die vergelijkbaar is met de bestaande zoekmachines zoals Google en Yahoo, maar in plaats van webpagina's geeft ons systeem artiesten terug. Hier komt gelijk het tweede punt waar wij bij het ontwerp van de eerste iteratie over nagedacht hebben, namelijk wat wij nu eigenlijk gaan aanraden.

Wij hebben gekozen om artiesten terug te geven als zoekresultaten. De reden hiervoor is dat wij artiesten niet te specifiek, maar ook niet te algemeen vinden. Een muziekgenre kan bijvoorbeeld een groot bereik aan muziek zitten; er kunnen twee compleet verschillende artiesten uitkomen die hele andere muziek maken. Wanneer dit gebeurd is er natuurlijk geen sprake van een goed werkend recommender systeem. Aan de andere kant vonden wij de titels van individuele nummers te specifiek om als resultaat terug te geven. Wanneer wij nummers terug geven als zoekresultaat komt er een enorme dataset als resultaat terug. Hierdoor zal de zoekende gebruiker veel werk moeten verrichten om de data te doorkruisen. Het is dus veel gemakkelijker om artiesten als resultaat terug te geven en deze te doorzoeken. Artiesten zijn makkelijker te doorzoeken dan nummers, maar ook niet zo algemeen als een muziekgenre. Natuurlijk moet er uiteindelijk wel muziek gevonden worden, daarom hebben wij ons systeem zo ontworpen dat de albums van een uitgekozen artiest opgehaald kunnen worden en dat deze kunnen worden doorkruist zodat de gebruiker uiteindelijk toch muziek te horen krijgt.

Bij het ontwerpen hebben wij besloten om enkele bestaande muziekdatabases te gebruiken en deze aan elkaar te koppelen. Dat wil zeggen dat de informatie van de databases word samengevoegd waarbij de ene database de andere aanvult met data die nog niet is vergaard. Dit hebben wij gedaan omdat wij zoveel mogelijk informatie van een artiest willen hebben om deze in eerste instantie te kunnen gebruiken voor het produceren van adviezen en in tweede instantie om deze informatie aan de gebruiker weer te geven. Wij gebruiken de databases van LastFM en MusicBrainz omdat deze makkelijk zijn aan te spreken en omdat zij een grote up to date database hebben. Naast de data die bovenstaande databases bevatten (artiestnamen, geboortedata, albums, nummers) willen wij ook dat de gevonden muziek af gespeeld kan worden binnen ons systeem. Gelukkig hebben wij hiervoor ook een bestaande database gevonden genaamd GrooveShark. Dit is één van de dingen waar wij in onze tweede iteratie mee bezig zijn geweest.

In de tweede iteratie lag de focus van ons ontwerp op het verstevigen van ons recommender systeem, daarnaast wilden wij ook de mogelijkheid creëren om gevonden muziek binnen ons systeem af te spelen. Onze eerste gedachte lag bij het trainen van een neurale netwerk om op deze manier de goede resultaten terug te geven. Na het gesprek bij onze eerste iteratie, waar wij andere ideeën hebben gehoord en waar het bleek dat een neurale netwerk erg lastig zou gaan worden hebben wij besloten om eerst een andere manier te gebruiken en ons ontwerp bij te stellen.

De manier waarop wij nu ons recommender systeem hebben ontworpen is door het leggen van links tussen artiesten. Deze links worden versterkt door de data in de dataset van Yahoo. Hierover volgt later meer in het hoofdstuk implementatie.

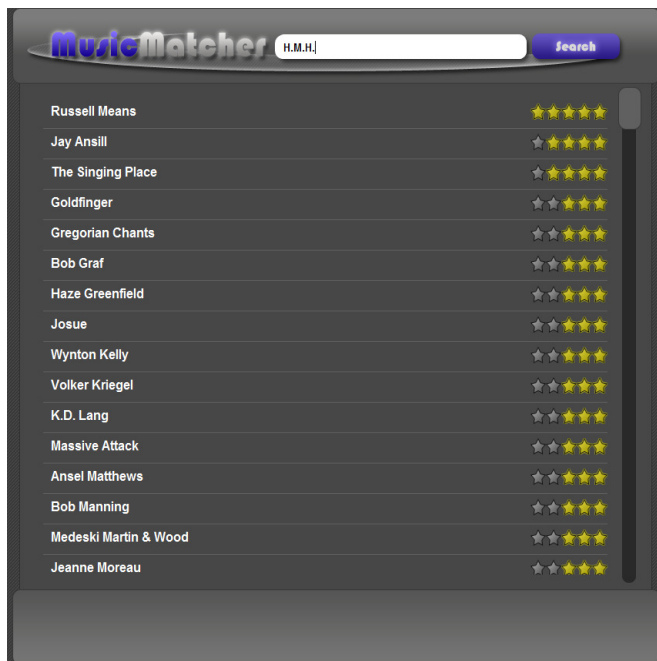
Het volgende punt waar ons ontwerp van de tweede iteratie verschilt van het ontwerp van de eerste iteratie is het afspelen van muziek. Bij het ontwerpen hebben wij besloten dat het toch erg handig is als de muziek gelijk kan worden afgespeeld. Hierdoor kan de gebruiker die aan het zoeken is de muziek direct beluisteren en belissen of hij/zij het leuk vindt of dat hij/zij verder moet zoeken.

Het ontwerp van de derde iteratie is zodanig in elkaar gezet dat we de punten uit de voorgaande iteraties willen verstevigen. Zo is aan het ontwerp van de derde iteratie een rating systeem toegevoegd. Dit systeem willen wij gaan gebruiken om de sterkte van de eerdergenoemde links tussen artiesten (die wij gebruiken om recommendations te maken) te kunnen beïnvloeden. Op deze manier kan ons systeem als het ware leren wat de juiste recommendations zijn en dus betere recommendation geven. Ook zijn op deze manier nieuwe artiesten veel makkelijker toe te voegen en te integreren in het systeem.

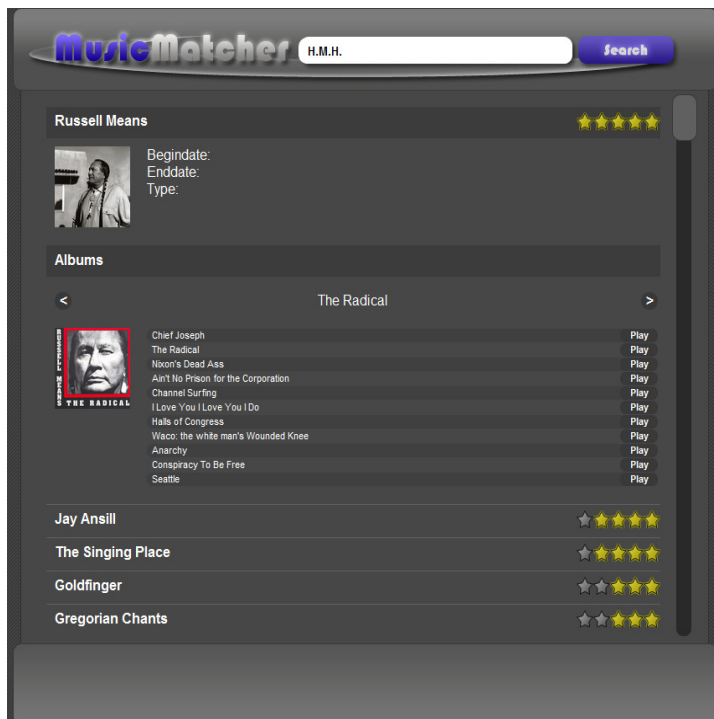
Het ontwerp van de site is natuurlijk ook zeer belangrijk. Dit is zeer dynamisch ontworpen zodat alles zonder te verversen kan uitklappen en zoekresultaten kan teruggeven. Op onderstaande afbeelding is te zien hoe het begin van de site er eerst uit zag.



Vervolgens kon je in het zoekvenster een artiest invoeren waarna er een lijst met advies artiesten uit kwamen die aanbevolen werden aan de gebruiker. Hierbij kon aan het aantal sterren gezien worden hoe goed het advies was. Dit is te zien in de volgende afbeelding.



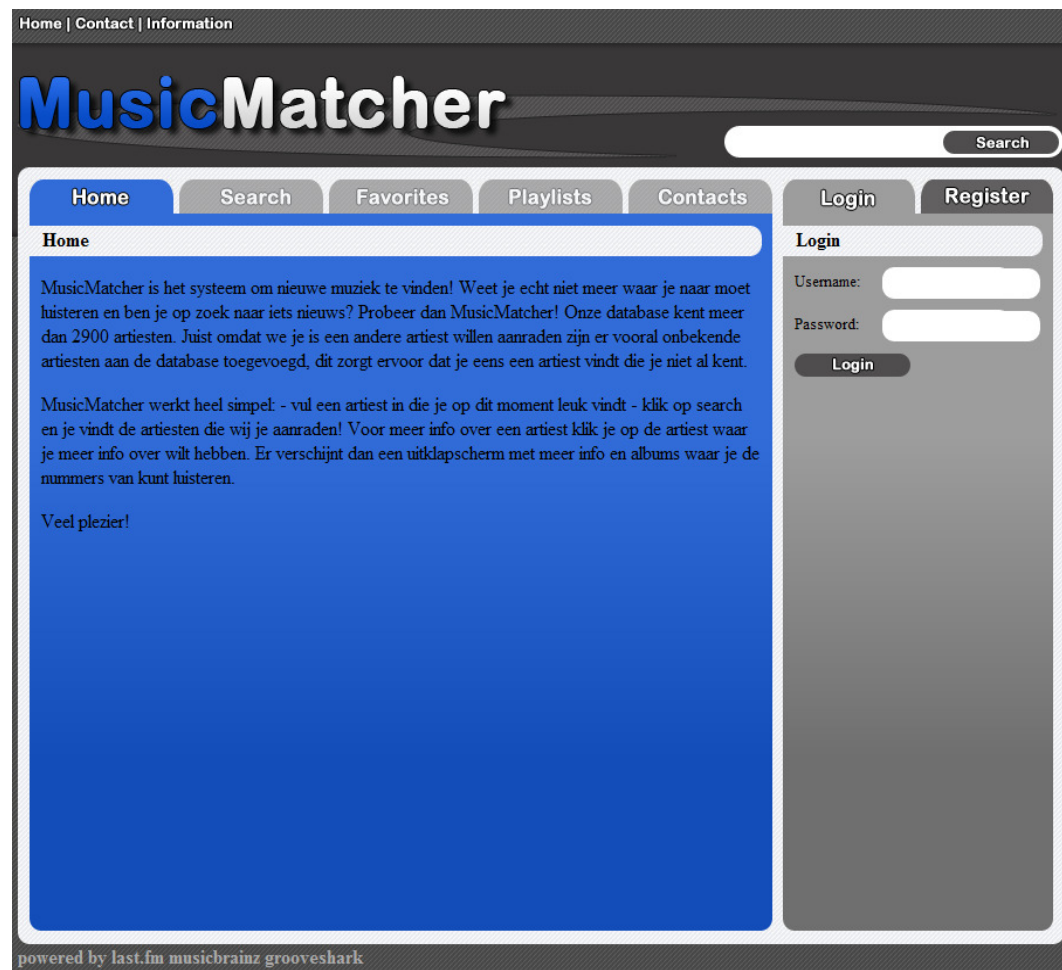
In dit scherm kon dan een artiest aangeklikt worden voor meer informatie. Hier kan je dus denken aan albums waar je doorheen kon scrollen om vervolgens een nummer te selecteren wat je af wilde spelen. Een voorbeeld is te zien in de afbeelding hieronder.



Door op de play knop te drukken ging er dan het nummer spelen van dat album. Hier wordt Grooveshark voor gebruikt zodat men een idee kreeg van het nummer.

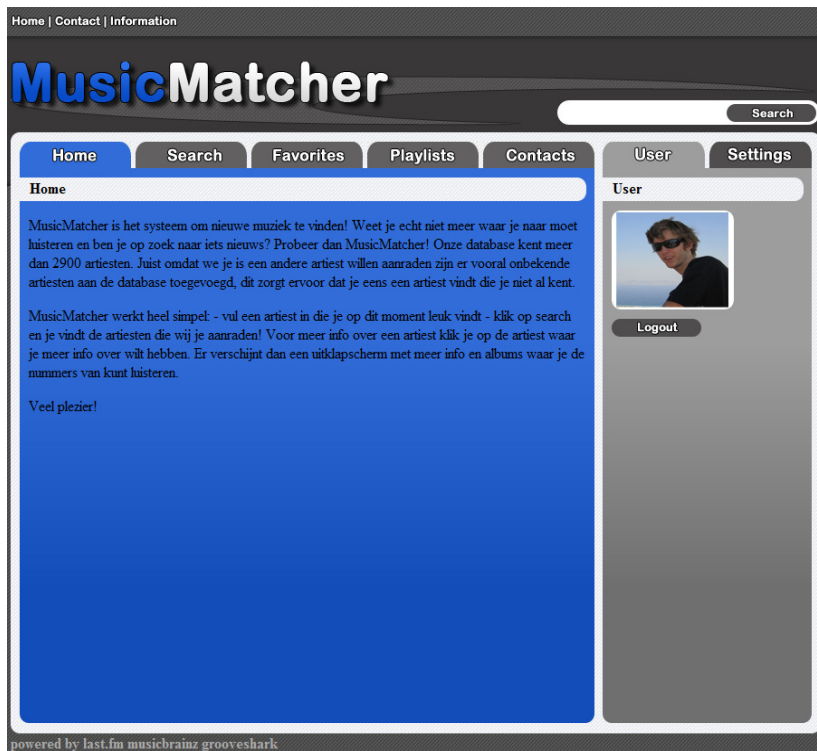
Ondertussen is er een ander ontwerp aangezien er nu een gebruikerssysteem geïmplementeerd is. Dit systeem had een andere lay-out nodig om het allemaal nog passend te maken. Het begint allemaal al bij een wat minder simpel begin scherm, waarbij de gebruiker kan inloggen of registreren als de gebruiker nog niet geregistreerd is.

Het eerste scherm is het home scherm. Hier kom je dus als je eerst de site laad. Op dit scherm is wordt er uitgelegd wat de site doet en kan je inloggen op registreren. Hieronder is te zien hoe dat eruitziet.



Vervolgens kan je inloggen. Dan krijg je een plaatje te zien wat je zelf kan uploaden een zogenaemde avatar. Hieronder is een voorbeeld te zien.





Je kan dan ook nog bij settings je wachtwoord, plaatje of emailadres wijzigen. Als je vervolgens zoekt op een artiest, krijg je de artiesten die ons recommender systeem je aanraden. Deze komen allemaal in een lijst tevoorschijn zoals hier onder is afgebeeld.



Vervolgens is er dan nog de mogelijkheid om op een artiest te klikken en meer info te krijgen. Hierbij kan je ook door de albums heen scrollen en de nummers van de albums luisteren via Grooveshark, tenminste als Grooveshark het nummer kent. Hieronder is afgebeeld hoe dat eruitziet.



Al met al lijkt ons dit een duidelijke interface om er voor te zorgen dat dingen makkelijk te vinden zijn. De screenshots van verdere pagina's zijn verder weggelaten omdat die zo ongeveer hetzelfde eruitzien en dit de belangrijkste pagina's zijn.

## Implementatie

Bij de implementatie van het product moeten er op vele dingen gelet worden. Aangezien dit een product van enige omvang is, is de planning een zeer belangrijk punt waar men niet zomaar overheen kan stappen. Zo hebben we een duidelijke planning over wat we in welke iteratie willen doen.

In de eerste iteratie hebben we veel tijd besteed aan het online krijgen van het product en het zorgen dat we een basis recommender systeem hadden. Hierbij hebben we eerste gebruikt gemaakt van LastFM en het recommender systeem dat daarin gebouwd is. De site is volledig gebouwd met HTML als lay-out. Hierdoor ziet de site er zeer strak uit, vooral omdat alles zeer dynamisch is door de AJAX functies waar het mee samenwerkt.

In de tweede iteratie hebben we vooral tijd besteed aan het koppelen van Groovespark aan ons systeem. Dit is een systeem waarmee elk nummer dat wij aanraden ook weer online beluisterd kan worden. Dit is een grote aanwinst voor ons systeem zodat mensen ook makkelijk kunnen luisteren of ze ons advies ook echt leuk vinden. Hierdoor kunnen we ook makkelijker erop vertrouwen dat de gebruikers ons gaan helpen het advies wat wij geven te beoordelen. Later in het verslag volgt meer over het beoordelen van de artiesten. Wat we ook in de tweede iteratie hebben gedaan is het bouwen van een tabel waarin alle artiesten uitgezet zijn tegen alle artiesten en er op de connecties tussen de artiesten een bepaalde rating wordt neergezet. Op deze manier kunnen we makkelijk zoeken naar artiesten die bij andere artiesten passen. Hiermee krijgen we dan ons eigen recommender systeem.

De tabel is samengesteld uit Yahoo data. De dataset die we ervoor hebben gebruikt bestond uit een regel met een identificatie van een gebruiker met de regels die erop volgden als artiesten en de beoordeling daarvan door die bewuste gebruiker. We hebben daarbij de verschillende artiesten van elke gebruiker apart op een rijtje gezet, en telkens per gebruiker gekeken welke beoordeling ze aan elke artiest gegeven hadden. Vervolgens hebben we via een Java programma gezorgd dat als de beoordelingen van twee artiesten allebei hoger waren dan 80 dan werd de waarde van de link tussen die artiesten met 3 opgehoogd. Was deze rating voor één artiest 80 en de ander 50 dan was dit met 2, en waren ze allebei 50 dan ging er 1 bij de link. Onder de 50 is het net als een onvoldoende voor een proefwerk en is besloten dat het dus ook onvoldoende is om de link nog maar iets op te hogen. Vervolgens toen we al deze links hadden in een tabel hebben we die geüpload, en gezorgd dat we toegang hadden tot deze tabel. Dit hebben we gedaan door in PHP een klasse te programmeren. In eerste instantie lieten we deze klasse de volledige tabel inlezen. Dit resulteerde in een memory overload die PHP niet aankon. De tabel was namelijk meer dan 17 MB in tekstbestand, PHP vond dit blijkbaar bij het omzetten naar een twee dimensionale array nog wat te groot. We hebben vervolgens bedacht dat bij het doorzoeken de tabel voor een resultaat van één artiest ook maar één lijntje gesplitst hoefde te worden. Deze lijn wordt dan verder op het maximum van de beoordelingen doorzocht en zo kan je de maximum 10 of 20 resultaten laten teruggeven. De 10 of 20 zijn hier maar voorbeelden, hiervoor kan in het systeem zelf van alles ingevoerd worden, mits er zoveel artiesten in de tabel staan.

Hoe we ons systeem hebben opgebouwd is niet heel makkelijk uit te leggen. De programmeertaal keuze was redelijk makkelijk. Aangezien we een recommender systeem moesten maken, wil je ook dat die makkelijk toegankelijk is voor gebruikers. En wat is er nu makkelijker dan het internet. Dus

aangezien we het systeem op internet gingen bouwen is er gekozen voor PHP als achterliggend systeem om alle informatie op te halen, als voorkant is er HTML en Javascript gebruikt om een mooie dynamische website te maken. Om alles met elkaar te verbinden is AJAX gebruikt zodat de pagina's nooit ververst hoeven te worden. Dit alles is zo geprogrammeerd om de site zo snel mogelijk te laten werken en de gebruikers het makkelijk te maken om de site door te werken.

De achterkant van de site waar alle informatie wordt opgehaald is gebouwd met PHP en haalt de informatie op van MusicBrainz en LastFM door middel van JSON. Deze acties duren het langst van alle acties op de website. Dit komt door MusicBrainz en LastFM omdat deze redelijk traag zijn in het geven van informatie. Hier is dus helaas niks aan te doen. De PHP is allemaal Object georiënteerd gemaakt zodat we heel makkelijk met klassen kunnen werken en geen werk dubbel hoeven te doen maar alles binnen de klassen wordt opgeslagen.

Van de achterkant van de site gaat de informatie in JSON objecten naar de voorkant, waar Javascript dan door middel van AJAX de informatie ophaalt en de HTML instuurt en alles zichtbaar maakt. Dit zorgt voor een totaal dynamische website waar geen enkele pagina ververst hoeft te worden.

Het recommender systeem wat we tot iteratie 2 gebruikte is een systeem van LastFM waar je voor een bepaald genre de beste artiesten kan opvragen. Dit werkt nog niet super goed en vandaar dat we zelf zijn gaan bouwen aan een recommender systeem. Bij dit systeem kwam vaak nog een artiest als Red Hot Chili Peppers terug als je zocht op Coldplay. Beide hebben namelijk als genre pop en dat wordt dus samen teruggegeven. Dit zijn echter heel verschillende artiesten en vonden wij dus nog niet goed genoeg.

In de derde iteratie hebben we de tabel die we bij de tweede iteratie hebben gebouwd nog wat uitgebreid met de tweede track van de dataset van Yahoo. Deze was nog wat groter zodat de beoordelingen nog wat signifikanter er uit zijn gaan schieten. Wat wel opvalt is dat er weinig bekende artiesten in de Yahoo dataset zitten. Hierdoor is het lastig om artiesten die mensen al kennen te gaan vergelijken met de artiesten in de tabel. We willen daarom ook nog beoordelingen van LastFM gaan halen om te zorgen dat er ook bekende artiesten in de tabel komen. Hiermee zouden we dan makkelijker nieuwe bekende artiesten kunnen vergelijken met de oude onbekende artiesten.

Wat we verder in iteratie drie hebben gedaan is het implementeren van een gebruikers systeem. Hiermee kunnen mensen zich registreren en daarna inloggen in de website. Hiermee kunnen we dan makkelijk een playlist functie bouwen zodat elke gebruiker een lijstje van nummers bij kan houden die hij of zij graag luistert. Ook hebben we favorieten ingebouwd. Deze favorieten zijn artiesten die door de gebruiker positief beoordeeld zijn. Deze kunnen ook weer makkelijk opgezocht worden door het tabblad favorieten. Deze favorieten worden dan ook weer in de tabel verwerkt waardoor links weer versterkt worden en de gebruikers ook andere gebruikers helpen met adviezen. Ook hebben we een simpele zoek functie geïmplementeerd die ervoor zorgt dat je makkelijk naar artiesten kan zoeken waarbij er onder de gezochte artiest dan adviezen gegeven worden over welke artiest erbij zou passen, ook meegenomen de favorieten van die gebruiker.

Uiteindelijk krijgen we dus een recommender systeem dat via PHP alle informatie ophaalt. Waarbij adviezen verstrekt worden op basis van een tabel die is samengesteld uit een flinke dataset. Verder wordt er meta informatie opgehaald van LastFM en MusicBrainz zodat we ook meer kunnen afbeelden dan alleen de artiest naam. Dit alles wordt dan verwerkt door Javascript met Ajax en

afgebeeld op de site zonder dat er een pagina verversst hoeft te worden. Dit alles zorgt voor een redelijk snelle site die makkelijk toegankelijk zou moeten zijn voor gebruikers via het internet.

## Product evaluatie

Een belangrijk element van het bouwen van een product is het testen ervan. Als het product niet goed getest is kan er niet ingestaan worden voor de kwaliteit van het product. Vandaar dat ook wij een plan hebben gemaakt voor het evalueren van de recommender die we gebouwd hebben.

Aangezien ons product een mix is van een user interface en een systeem wat erachter hangt is het automatisch testen van het product zeer lastig. Tevens is de Yahoo dataset niet geschikt om het product automatisch mee te testen. Dit komt omdat er veel dubbel in zit zodat een deel weglaten uit de dataset heel lastig is. We hebben er daarom voor gekozen om meerdere mede-studenten naar het product te laten kijken en te laten testen. Zij kunnen dan aan de hand van een evaluatie formulier aangeven hoe goed ze de verschillende onderdelen, en het geheel vinden. Via dit evaluatie formulier kunnen we dan ook achterhalen wat er nog verbeterd kan worden aan het product. Hier kunnen we dan weer aanbevelingen mee geven wat er nog meer aan het product gedaan kan worden en verbeterd moet worden.

We hebben natuurlijk wel zelf al een evaluatie gedaan. Wat opvalt op dit moment is dat er dus alleen nog onbekende artiesten gevonden kunnen worden. Deze artiesten zijn dan meestal ook niet bekend op MusicBrainz of LastFM zodat er weinig informatie van beschikbaar is. Ook Groovespark kent veel nummers van deze artiesten niet zodat de site nog niet optimaal benut kan worden. Het is wel al makkelijk om met de site te werken, het registreren en inloggen gaat zeer makkelijk zodat het ook aantrekkelijk is om dit te doen en gebruik te maken van de site.

## Project evaluatie

Tot nu toe loopt het project redelijk. Het gaat nog niet super vlot en snel. Dit heeft meerdere redenen. Één van de redenen is zeker dat het project nieuw is en we nog heel erg op ontdekkingstour zijn over het hoe en wat bij het project.

Een tweede is zeker dat het vakgebied waarover het project gaat nog in ontwikkeling is. Het interactief advies geven via websites is iets dat nog niet zo lang bestaat en er moet dus ook nog een perfecte manier gevonden worden om dit te bewerkstelligen. Hier doen we natuurlijk onze uiterste best voor, maar op een dusdanig kort tijdbestek is het lastig te realiseren.

Een derde factor die er zeker voor zorgt dat het nog niet heel soepel loopt is waarschijnlijk dat er naast het project nog Algoritmie loopt waar ook zeker meer dan 10 uur tijd in steekt per 2 weken zodat er weer minder tijd overblijft om na project tijd nog even verder te gaan en de finishing touch op dingen te geven. We hopen natuurlijk dat dit positief omslaat en ons tijd geeft om nog meer te bereiken in het project.

Ondanks deze drie factoren moeten we toch zeggen dat we redelijk positief kunnen zijn van het resultaat tot nu toe. Er staat een zeer nette website met nu nog een basis recommender systeem. We zijn echter zeer dichtbij een oplossing voor een geavanceerd recommender systeem waar nog niet alle opties van bekend zijn. In volgende iteraties is dit nog een prachtig onderwerp om verder op te bouwen.

## Conclusies en aanbevelingen

Het belangrijkste wat we deze iteratie hebben gedaan is het implementeren van het gebruikers systeem. Hierdoor krijgt de recommender een persoonlijk tintje wat belangrijk is om de adviezen persoonlijk te maken. Een aanbevelingen voor volgende iteraties is zeker nog om de tabel uit te breiden met bekende artiesten zodat het makkelijker wordt voor de gebruikers om populaire met minder populaire artiesten te vergelijken. Het is wel goed dat ook onbekende artiesten in de tabel zitten, hierdoor is het juist leuk te zien dat mensen artiesten vinden waar ze nog nooit van gehoord hebben.

De recommender werkt nu redelijk goed. Het systeem werkt, de verbetering kan nog gevonden worden in het vergaren van meer data. Dit zou er voor zorgen dat er betere adviezen uit de recommender komen rollen. Het vergaren van meer data is echter zeer lastig omdat het vaak in zeer lastige formaten aangeleverd wordt, en het maar op één manier verwerkt kan worden. Het is dus zaak om alle data constant te converteren en daarna te verwerken. De data ligt echter ook niet voor het oprapen, het is dus nog maar de vraag of er zoveel data vergaart kan worden dat de recommender degelijk genoeg gaat werken.



## **Bijlage A:**

### **Evaluatieformulier MusicMatcher**

Naam:

NetID:

Leeftijd:

Welke artiesten vind je leuk om naar te luisteren?

Probeer je te registreren op de MusicMatcher site.

Gaat dit makkelijk? Wat zijn de dingen die goed gaan wat gaat niet goed? Wat zijn de verbeterpunten?

Probeer je nu eens in te loggen. Gaat dit makkelijk zo ja waarom, zo nee waarom en hoe kan dit verbeterd worden?

Probeer nu is een advies te krijgen voor waar je naar moet luisteren.

Lukt dit makkelijk? Zo ja hoezo, zo nee hoezo niet en hoe kan het verbeterd worden?

Is het advies dat gegeven wordt ook iets wat je leuk vindt, weet je nog niet of je het leuk vindt, of lijkt het je helemaal niks?

Probeer eens extra info over een artiest te vinden.

Gaat dit makkelijk? Zo ja hoezo, zo nee hoezo niet en hoe kan het verbeterd worden?

Is na het bekijken van de info van de artiest je mening veranderd over het advies? Zo ja wat is je mening nu, zo nee hoezo niet?

Probeer eens een artiest aan te geven als favorite.

Lukt dit makkelijk? Zo ja hoezo, zo nee hoezo niet en wat kan er verbeterd worden zodat dit wel makkelijk gaat?

Probeer je favoriete artiest eens terug te vinden.

Gaat dit makkelijk? Zo ja hoezo, zo nee hoezo niet en wat kan er gedaan worden om dit te verbeteren?

Probeer eens een nummer toe te voegen aan de playlist.

Gaat dit makkelijk? Zo ja hoezo, zo nee hoezo niet en wat kan er verbeterd worden om dit makkelijker te maken.

Probeer de playlist eens af te spelen.

Gaat dit makkelijk? Zo ja hoezo, zo nee hoezo niet en wat kan er verbeterd worden om dit makkelijker te maken?

Probeer eens een contact toe te voegen in het systeem.

Gaat dit makkelijk? Zo ja hoezo, zo nee hoezo niet en wat kan er verbeterd worden om dit makkelijker te maken?