TI2730-D: Music Online

Verslag sprint 2

Groep 1:

Liesbeth van Alebeek

Bjorn van der Laan

Rogier Slag

Doel van systeem

Een website met een zoekbalkje voor Youtube-filmpjes. Hier kan een link naar een muziekfilmpje ingevoerd worden en vervolgens begint deze af te spelen. Ondertussen wordt er vanuit allerlei bronnen informatie opgehaald en gecombineerd. Het scherm vult zich met informatie over de artiest, concert-data, mogelijkheden om het bijbehorende album te kopen en nog meer.

Wanneer het nummer afgelopen is verschijnt er een lijst met suggesties voor andere muziek. Deze suggesties zijn gebaseerd op de informatie die de gebruiker op zijn of haar facebookprofiel heeft ingevuld, de luistergeschiedenis en de sociale en geografische gegevens van de gebruiker. Wanneer de gebruiker ervoor kiest om een suggestie te beluisteren, verschijnt ook de mogelijkheid om de kwaliteit van de suggestie te beoordelen. Deze informatie wordt vervolgens weer gebruikt om toekomstige suggesties te verbeteren.

Doelgroep

In ons project ligt de focus vooral op jongere mensen. Bij deze doelgroep is het belangrijk dat zij zelf niet al te veel moeite hoeven te doen. Jongeren zijn snel geneigd een website te verlaten als zij teveel stappen moeten maken om de informatie te krijgen die zij zoeken. Daarom hebben we gezorgd voor een simpel interface waarbij ze ons alleen de youtube link hoeven te geven. Op deze manier kan een filmpje dat iemand linkt direct ingevoerd worden, en hoeft de gebruiker niet eerst op te zoeken welk nummer het is, of wie de artiest is.

Schets van de oplossing en functionaliteit

Registreren en inloggen

Wanneer een gebruiker de site bezoekt komt deze als eerste terecht op een inlog-pagina. Nieuwe gebruikers kunnen hier naar de registratie-pagina gaan. Wanneer de gebruiker is ingelogd op facebook worden een aantal basisgegevens automatisch ingevuld. Anders moet dit handmatig gedaan worden. Na het registeren kan er voortaan ingelogd worden.

Muziek luisteren en informatie

Wanneer er ingelogd is, zijn er twee mogelijkheden voor het beluisteren van muziek. Er kan een linkje van een Youtubenummer ingevoerd worden, deze zal dan afgespeeld worden. Op de rest van de pagina verschijnt informatie over de artiest, het nummer, en het album waar het vanaf komt.

Als alternatief wordt er op het scherm een lijst van suggesties weergegeven. Hieruit kan er een nummer gekozen worden om te luisteren. Ook hiervan wordt de informatie weergegeven. Ook kan de gebruiker aangeven of het een goede suggestie was door middel van een thumbs up of down.

Recommendationsysteem

Luistersuggesties worden gebaseerd op hoeveel bepaalde nummers op elkaar lijken. Dit wordt bepaald aan de hand van artiest, bpm, genre, populariteit en rating. Per nummer worden deze eigenschappen vergeleken, en word een een waarde gegeven aan hoeveel twee nummers op elkaar lijken. Als een gebruiker een bepaald nummer luistert, wordt er in de database opgezocht welke nummers hier het meest op lijken, oftewel, de hoogste score hebben in relatie tot het oorspronkelijke nummer. In de toekomst

zullen deze suggesties ook vergeleken worden met eerder geluisterde muziek van de gebruiker. De nummers die ook hiermee vergeleken hoog scoren, zullen gesuggereerd worden.

Implementatie keuzes

Php

De keuze voor PHP is een die relatief snel is gedaan. PHP kent enkele grote voordelen op het gebied van webdevelopment, maar tegelijkertijd ook aan aantal nadelen. Na het afwegen van beide kanten is er gekozen deze taal als primaire ontwikkeltaal te gebruiken.

Allereest de voordelen. PHP is een taal die breed wordt ondersteund op webplatformen. Het is daardoor erg eenvoudig om hosting te regelen. Door de brede ondersteuning is de gemeenschap, die ontwikkelt en bijdraagt aan PHP, erg groot. Hierdoor is het weer eenvoudig om bestaande libraries te kunnen gebruiken, waardoor er geen code opnieuw hoeft te worden geschreven. Tijd die vrijkomt kan zo worden besteed aan het schrijven van code die daadwerkelijk functionaliteit toevoegt. Tenslotte is PHP een taal waarmee de hele groep enigszins bekend is en snel valt aan te leren. Ook dit bevordert de ontwikkelsnelheid.

Helaas heeft iedere keuze ook een aantal nadelen. PHP's voornaamste nadeel is de inconsistentie in de taal zelf en het gebrek aan afdwingen van een correcte manier van coden. De inconsistentie is eenvoudig op te lossen door gewoon regelmatig documentatie te gebruiken en zelf, waar nodig, een interface gebruiken welke wel consistent is. De correcte manier van coden is lastiger (try-catch, exceptions, classes), PHP heeft ondersteuning voor deze functionaliteiten pas sinds versie 5 in de taal opgenomen. Hierdoor is veel informatie op Internet niet geschreven voor deze features. Wat nog sterker meespeelt is dat ook binnen de TUDelft het onderwijs in PHP nog wordt gegeven op de "oude" manier. Hierdoor is het voor het team noodzakelijk te schakelen ten opzichte van wat men gewend is.

Over het totaal genomen is de keuze voor PHP sterk verdedigbaar. Het moeten aanleren van een correcte codingstijl kan ook juist als pluspunt worden gezien.

Youtube

De keuze voor YouTube is redelijk eenvoudig. Omdat er gebruik wordt gemaakt van de YouTube links voor user-input is het nodig om op enige wijze met YouTube te communiceren. Bovendien is de gratis API van YouTube erg uitgebreid wat betreft data die opgevraagd kan worden.

Ook is het bijna triviaal om de data van YouTube op te halen doordat de PHP API implementaties al gemaakt zijn en vrij beschikbaar zijn gemaakt. Het hergebruik van deze libraries is zeker aan te raden, aangezien dit heel veel programmeerwerk bespaart.

Door de grote hoeveelheid data die de API terug geeft kan er op veel verschillende kenmerken worden gematcht, wat de match en recommendation naar de gebruiker toe erg fijnmazig geregeld kan worden.

Facebook

De Facebook API is gebruikt om het registratie formulier te maken. Daardoor wordt, als men op Facebook is ingelogd, een deel van het formulier automatisch ingevuld. De API kan ook gebruikt worden om muziek interesse en de likes van een gebruiker ophalen, om zo een beter beeld te krijgen van wat deze gebruiker leuk vindt.

Database

Het gebruik van de database is tweeledig. Enerzijds wordt de informatie over de geregistreerde gebruikers opgeslagen; anderszijds verdient ook de informatie over de muzieknummers er een plekje.

Voor de gebruikersdata is een tabel opgezet. Deze bevat de inloggegevens van de gebruiker, de gebruikerskenmerken (geboortedatum, geslacht ed) en eventueel een Facebook authenticatietoken. Via dit token kan er later extra data van een gebruiker worden opgehaald. Door een aantal gebruikerskenmerken op te slaan bestaat de mogelijkheid om later deze data in te zetten voor het personaliseren van zoekresultaten en suggesties.

Voorts wordt er van ieder beluisterd nummer een informatieset opgeslagen in de database. Volgende keren hoeft deze data dan niet opgehaald te worden. Bovendien vormt deze dataset de basis van het algoritme welke gebruikers gerelateerde muzieknummers aanbiedt.

Van ieder nummer wordt vanuit verschillende bronnen data gecombineerd; voorbeelden zijn Spotify, Youtube en Echonest. Deze data wordt vervolgens weer opgeslagen in de database. Voorbeelden van de data zijn genres, ratings, populariteitstatistieken, tempo. Door het combineren van de verschillende kenmerken wordt de data daadwerkelijk interessant.

De data wordt in matrixvorm gerepresenteerd, maar in tabelvorm opgeslagen. Voor ieder item in database bestaat een x en een y veld, beide zijn identifiers voor muzieknummers. Het (x,y)-veld representeert dan ook in hoeverre y aan te raden is als x wordt beluisterd. Dit veld bestaat vervolgens uit twee delen. De eerste is daadwerkelijk het getal wat de match aangeeft, het tweede een link naar een andere tabel waar de puntenverdeling aangeeft (dus uitsplitsing per genre, artiest, album etc). Hoewel dit een dataredundantie is (welke vaak beter te vermijden is in een database), zorgt deze ervoor dat de hoeveelheid data welke doorlopen moet worden sterk slinkt. De waarde is namelijk al opgeslagen als ‘cache’ in het (x,y)-veld. Dit reduceert een O(nm) algoritme (x=aantal nummers, m=aantal kenmerken) tot een O(n) algoritme, een waardige winst.

Tenslotte wordt de data van beluisterde nummers ook gelinkt naar de individuele gebruikers. Door hen te volgen en zo direct informatie te kunnen krijgen over hun muzieksmaak wordt er een voordeel gegenereerd bij het personaliseren van resultaten. Deze data is vrij eenvoudig en bevat simpelweg een link tussen een nummer en een gebruiker, plus het aantal keer dat het betreffende nummer is beluisterd.

Sparql vs EchoNest

In eerste instantie werd met sparql de informatie opgehaald over de artiest en het nummer. Dit gebeurde door middel van een library voor php genaamd Sparqllib. Dit is een lichte doch krachtige library is, die alle benodigde functies bevat. Helaas bleek dat DBPedia.org niet snel genoeg resultaten terug te geven. Daarom is er overgestapt op EchoNest, hier kunnen JSON-objecten opgevraagd worden met alle gewenste informatie.

Uitvoering

Planning

Het project is opgedeeld in vier sprints, elke sprint bevat een aantal hoofdonderwerpen waar aan gewerkt zal worden. Op deze manier wordt er elke keer een volgend onderdeel van het eindproduct gerealiseerd.

1. De eerste sprint ligt de focus op het maken van een eenvoudig systeem, waarin de meeste basisfunctionaliteit gerealiseerd wordt. Er kan worden geregistreerd, ingelogd, video's worden afgespeeld

en informatie over de artiest wordt weergegeven. Hiermee is er een goede basis om volgende sprints uit te breiden.

2. Tijdens de tweede sprint wordt een begin gemaakt aan het recommendationsysteem. Informatie wordt gecombineerd en afgestemd op de gebruiker. De informatie behorend bij een nummer wordt uitgebreid en de site wordt opgemaakt en gebruiksvriendelijk gemaakt.

3. In de derde sprint worden de eerste tests gedaan op de werking van het recommendationsysteem. De webpagina die gebruikers te zien krijgen wordt verder gepersonaliseerd en het informatiesysteem wordt afgemaakt.

4. De laatste sprint is bedoeld om de puntjes op de i te zetten. Het systeem wordt getest, de resultaten van de tests worden gebruikt om verbeteringen door te voeren. Aan het eind van deze sprint is het systeem voltooid en kan het gepresenteerd worden.

Eerste sprint

De eerste spint van het project stond in het teken van het ontwikkelen van basisfunctionaliteit die het project ondersteunen. Denk hierbij vooral aan het uitzoeken van informatie over koppelingen en het opstarten van verschillende frameworks welke de ontwikkeling ondersteunen. Voorbeelden zijn het opzetten van actiepuntensystemen, documentatie, errorlogs, gebruikersaccount, gitHub en het framework zelf.

Allereerst is er begonnen met het kiezen van de databronnen die ingezet worden. De keuze hiervoor viel op Facebook, Youtube en DBPedia. Deze zijn vervolgens aan elkaar gelinkt in deze eerste sprint.

Voor het inloggen is er de mogelijkheid om dit via Facebook te doen. Indien een gebruiker ingelogd is binnen Facebook wordt het registratieformulier automatisch ingevuld met de al bekende parameters. Ook krijgen we in dat geval een authenticatietoken binnen waarmee we later extra informatie kunnen ophalen over de betreffende gebruiker.

Youtube wordt ingezet om de videokenmerken op te halen en om de video zelf af te spelen. Hiervoor biedt Youtube de gewenste functionaliteit binnen haar API's. Voor de verbindingen naar Youtube wordt gebruik gemaakt van de API van Zend Framework.

Tenslotte wordt de naam van de artiest die wordt ontvangen via de API van Youtube via een SparQL query doorgestuurd naar DBPedia, waar een korte samenvatting over die artiest ophaalt.

Qua layout wordt er op dit moment nog gebruik gemaakt van simpel ogende pagina's. Het is de bedoeling dit in de toekomst echter via het Twitter Framework Bootstrap te laten lopen, waarbij de data asynchroon wordt opgehaald voor de gebruiker.

Taakverdeling

Taak:	**Verantwoordelijk:**
Facebook-registratie	Bjorn
Querier RDF-data	Bjorn
Youtube API	Liesbeth
Bootstrap voor lay-out website	Liesbeth
Framework systeem	Rogier

Integreren onderdelen	Rogier

Tweede sprint

In de tweede sprint hebben we het ophalen van informatie verbeterd. Er is overgestapt van sparql naar EchoNest. Er is voor EchoNest gekozen omdat dit veel sneller werkt. Daarnaast is niet alle benodigde data beschikbaar in RDF, maar wel op te halen via EchoNest. Op dit moment worden BPM, rating en genres opgehaald van Echonest. Verder slaan we ook het Spotify-ID op, omdat dit een unieke waarde is, die gebruikt kan worden voor het ophalen van verdere informatie.

Wat betreft de website stond de eerste sprint puur in het teken van functionaliteit. In deze sprint is er aandacht besteed aan de opmaak van de website. Hiervoor hebben wij gebruik gemaakt van Twitter's Bootstrap. Met dit framework kan je in korte tijd een mooie opmaak bouwen, waar makkelijk extra's aan toegevoegd kunnen worden.

Daarnaast is er begonnen met het recommender systeem. Op dit moment nemen we 4 variabelen mee in de berekening: bpm, artiest, genre en populariteit. In de volgende sprint willen we het systeem uitbreiden om de zoekresultaten te verfijnen. Ook zullen dan de resultaten gepersonaliseerd worden.

Taakverdeling

Taak:	**Verantwoordelijk:**
Informatie EchoNest	Bjorn
Verslag	Bjorn
Recommender algoritme	Liesbeth
Bootstrap voor lay-out website	Liesbeth
Opbouwen database	Rogier
Integreren onderdelen	Rogier