



Vrije
Universiteit
Brussel

Bachelorproef AMuRate: Voortgangsrapport 3e Bachelor Computerwetenschappen

David Sverdlov
dsverdlo@vub.ac.be

January 24, 2013



Contents

1	Inleiding	3
1.1	Doel	3
1.2	Referenties	3
1.2.1	AMuRate homepagina	3
1.2.2	Android	3
1.2.3	Ice Cream Sandwich	3
1.2.4	Last.fm	3
1.2.5	Github	3
1.3	Afkortingen en definities	3
1.3.1	API	3
1.3.2	XML/JSON	4
1.3.3	HTTP	4
1.3.4	GUI	4
2	Achtergrond	4
2.1	Last.fm/api: REST	4
2.2	REST architectuur	4
3	Voorgrond	5
3.1	Website	5
3.2	Naam en logo	5
4	Vereisten	6
5	GUI	6
6	Implementatie	7
6.1	Packet onderverdeling	7
6.2	Database	8
6.2.1	Lokale Database	8
6.2.2	Externe Database	8
7	Voortgang	8
8	Bronnen / Tools	9
8.1	Eclipse	9
8.2	Android Development Toolkit (ADT)	9
8.3	Verschillende informatie	9

1 Inleiding

1.1 Doel

De uitdaging van dit bachelorproject is om een applicatie te ontwikkelen voor het mobiele platform Android. De applicatie moet gebruikers in staat stellen om muzieknummers op te zoeken en een beoordeling te kunnen geven. Die scores worden dan opgestuurd naar een data-collection server met een database, waar later recommender algoritmen op zullen werken. (De implementatie daarvan valt niet binnen de scope van dit project.) Dit project wordt gepromote door prof. Dr. Ann Nowé en begeleidt door Maarten Deville en Peter Vranckx.

1.2 Referenties

1.2.1 AMuRate homepage

<http://dsverdlo.github.com/AMuRate>

1.2.2 Android

<http://www.android.com/>

1.2.3 Ice Cream Sandwich

<http://www.android.com/about/ice-cream-sandwich/>

1.2.4 Last.fm

<http://www.last.fm/home>

1.2.5 Github

<https://github.com/>

1.3 Afkortingen en definities

1.3.1 API

API staat voor Application Programming Interface en zorgt voor de communicatie tussen programmas, door de scheiding te vormen tussen verschillende lagen van abstracties.

1.3.2 XML/JSON

XML staat voor Extensible Markup Language en is een van de meest gebruikte opmaaktalen, die gestructureerde gegevens kunnen omzetten in platte tekst. (Om het zo makkelijk(er) door te kunnen sturen.)

JSON is een afkorting van JavaScript Object Notation en is een alternatieve simpele manier om objecten voor te stellen als platte tekst.

1.3.3 HTTP

HTTP staat voor HyperText Transfer Protocol en is het medium tussen een webbrowser en een webserver. Die communicatie gebeurt door middel van URLs, die verwijzen naar 'iets' op een of andere webserver.

1.3.4 GUI

GUI staat voor Graphical User Interface en is een visuele vormgeving van een programma, dat door middel van knoppen, afbeeldingen, ... gebruikers toelaat om op een gebruiksvriendelijkere manier met de applicatie om te gaan .

2 Achtergrond

2.1 Last.fm/api: REST

Om informatie (over muziek in dit geval) op te kunnen zoeken, moet er gebruik gemaakt worden van een online database met een openbare en hanteerbare API. *Last.fm* is een muziek recommendation service met een enorme online muziek database, die een gratis (lees: voor geregistreerde gebruikers) API aanbiedt, die iedereen toelaat om mobiele/desktop programmas of web services te bouwen met hun data.

Die procedure verloopt als volgt: om data uit de database te halen moet er een call (oproep) gestuurd worden. Deze gebeurt via HTTP GET naar de Last.fm server die op zijn beurt antwoordt met een XML object (of JSON op aanvraag). De methode van operatie die hier gebruikt wordt heet 'REST'.

2.2 REST architectuur

REST staat voor Representational State Transfer en is een type van architectuur voor bepaalde gedistribueerde systemen (zoals het World Wide Web), waarin een duidelijke onderscheiding wordt gemaakt tussen client en server. Clienten kunnen oproepen doen naar de server, die de oproepen analyseert, een antwoord formuleert, en dat antwoord terug naar de klant stuurt. REST maakt gebruik van werkwoorden en zelfstandige naamwoorden om de oproepen ook leesbaar voor mensen te maken. De standaard werkwoorden

zijn GET, POST, PUT en DELETE, om data te kunnen verkrijgen, wijzigen of opsturen.

3 Voorgrond

3.1 Website

In dit project wordt er gebruik gemaakt van Git als versiebeheersysteem, omdat het een gratis, eenvoudige en betrouwbare manier is om de broncode te beheren. Om updates in het project naar de repository te sturen wordt er gebruikt gemaakt van Github voor Windows. Nog een voordeel van deze keuze is dat Github de mogelijkheid biedt om snel en simpel websites te maken(en te behoren) voor de bestaande projecten. Zo werd er een kleine website voor dit project opgesteld, die terug te vinden is op: <http://dsverdlo.github.com/AMuRate/>. Hier kan men de open-source broncode bekijken, het logboek of dit document raadplegen, en de applicatie (.apk) downloaden voor Android 4.0 toestellen.

3.2 Naam en logo

De naam van een project geeft meestal een kleine beschrijving van wat de applicatie doet/ waar hij voor dient. AMuRate komt van "Android Music Rating". Omdat het project vooral geïmplementeerd zou worden via een Android emulator, en simpelweg "AMR" te kort en onduidelijk zou zijn, werd er gekozen voor AMuRate. Het logo (zoals te zien op de voorpagina) is een compositie van een 5-ster in een bol die iets weg heeft van een "dragon ball" (uit de Japanse animatieserie DragonBall Z). In DBZ bestaan er maar 7 bollen op de hele wereld en wanneer die allemaal verzameld zijn, mag de beheerder een wens doen. Een beetje gelijkaardig aan hoe elke artiest hoopt 5 sterren (/ 7 dragon balls) te verzamelen. In de ster zelf zijn drie letters gekleurd ("AMR"), naar de naam van de applicatie.

4 Vereisten

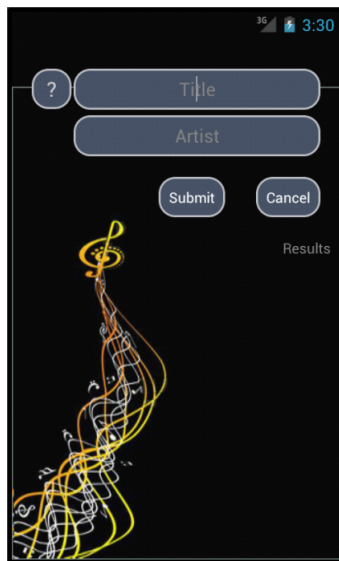
ID	Priority	Beschrijving	Status
1	necessity	Titel (en artiest) opzoeken	done
2	necessity	Track weergeven	done
3	necessity	Enkel op artiest opzoeken	in progress
4	optional	Album tonen	done
5	necessity	Score opslagen in database	done
6	necessity	Score ophalen in lokale database	done
7	necessity	Score opsturen naar remote db	NYI
8	optional	Zoek geschiedenis opslagen	in progress
9	optional	Stream/preview 30sec songs	in progress
10	optional	Caching implementeren	NYI
11	optional	Onderzoek uitbreidbaarheid (ios?)	NYI
12	optional	Informatie ophalen van media speler	NYI

Tabel 1: Vereisten van het project met uitleg en implementatiestatus.

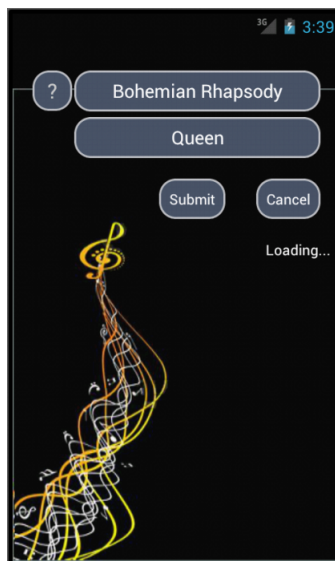
5 GUI

Hieronder volgen enkele schermopnames van de voorlopige GUI. Deze zal normaal gezien niet veel meer veranderen naar het verloop van de rest van het project, maar men kan nooit zeker weten.

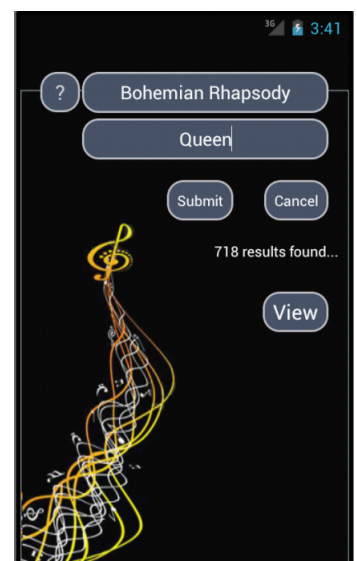
Voor het voorbeeld toon ik eerst het startscherm waar men mee begint wanneer de applicatie gestart wordt. Men kan kiezen



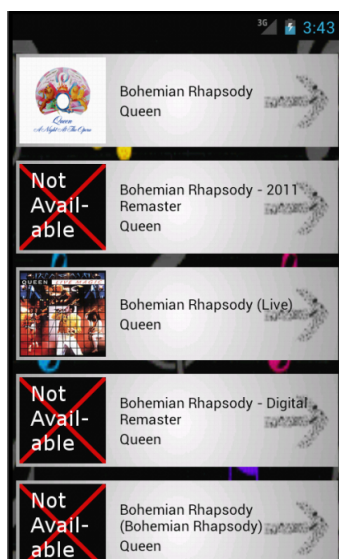
Start scherm



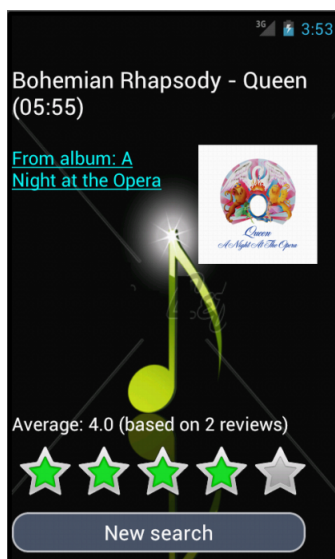
Zoek- en downloadproces



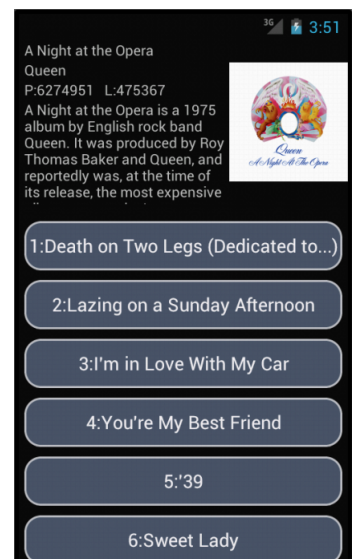
Klaar met downloaden



Geeft alle resultaten weer



Weergave van een track



Albumschermb

6 Implementatie

6.1 Packet onderverdeling

De broncode van het project is opgedeeld in 3 packets: *gui*, *objects* en *services*. In het *GUI* packet zitten alle activiteiten¹. Elk scherm heeft een aparte

¹Dit is de naam van een scherm in Android

activity nodig.

In het *objects* packet zitten de objecten, die abstractie brengen in het project. Bijvoorbeeld wanneer een track opgezocht werd en klaar is met downloaden, zal het omgezet worden in een klasse *Track*. Zo moet de GUI zich niets aantrekken van hoe de representatie eruit zag toen het van de server kwam. Packet *services* bevat de klassen die instaan voor verschillende diensten die moeten gebeuren in het project. Zo is er de *MyConnection* klasse, die de oproepen doet en antwoorden download van de Last.fm API. De database beheerder klasse is ook een dienst die verleend wordt aan het project, en bijgevolg ook in dit packet zit.

6.2 Database

6.2.1 Lokale Database

Ratings	
PK	_id
text	mbid
float	rating

Afbeelding 1: Dit schema stelt het model voor van de database.

De lokale database moet data bijhouden die offline geraadpleegd zou willen worden. Hierbij reken ik op een geschiedenis van de laatste zoekopdrachten en van de laatst gegeven scores. Aangezien deze applicatie vooral beroep doet op een actieve internetverbinding, zal er niet veel meer dan dat opgeslagen worden.

De implementatie van deze database gebeurt via SQLite.

6.2.2 Externe Database

Het design van de externe database zal ontworpen moeten worden met oog op de recommender algoritmen, die er later op 'losgelaten' gaan worden. Hier zal later meer informatie over komen.

7 Voortgang

Voor ik de voortgang van dit project begin te bespreken, eerst een klein overzicht van de hoogtepunten in het project tot nu toe.

Datum	Informatie
24-10	Informatie ontvangen i.v.m. opdracht
07-11	Afspraak met begeleiders en promotor, acceptatie bachelorproef
21-11	Android project aangemaakt met simpele GUI. Ook Last.fm account aangemaakt en unieke API sleutel aangevraagd.
05-12	Gebruikers kunnen echte calls maken en muziek opzoeken. Zitten nog bugs in en er kan nog niet gezocht worden naar artiesten
19-12	Er kan ook gezocht worden op artiesten, GUI werd herzien, abstractie in code toegevoegd.
03-01	Een score kan worden opgeslagen in de lokale database, en het totaal en gemiddelde kan uitgelezen worden.
xx-xx	Todo: Titel en artiest opslagen voor geschiedenis GUI. Ook nog op artiest kunnen zoeken. Caching implementeren. Ook alle bugs eruit halen.

Tabel 2: Enkele belangrijke highlights in de voortgang van het project.

Zoals in de sectie Vereisten gezien kan worden (met bewijs van sectie GUI), is de applicatie langs de client-kant bijna volledig af. Men kan al muziek nummers opzoeken, selecteren en beoordelingen geven die effectievelijk opgeslagen worden. Wat er aan deze kant nog ontbreekt is een manier om enkel op artiest te zoeken, geschiedenis te kunnen bekijken, en eventueel als er nog tijd is: caching implementeren om het verbruik en duratie van de internetverbinding op een minimum te houden.

Deze dingen staan vanboven op de TODO lijst. Daarna zal er aan de server-side gewerkt worden. Eens de werking van de recommendeer algoritmen beter bekend is, kan de externe database hiernaar opgezet worden, en moet er een aangepaste synchronisatie gebeuren tussen de twee databases.

8 Bronnen / Tools

8.1 Eclipse

Programmeerplatform Eclipse (Juno) werd gebruikt voor de implementatie en emulatie.

<http://www.eclipse.org/>

8.2 Android Development Toolkit (ADT)

Deze plug-in bevat alle benodigdheden om programma's voor Android in Eclipse te bouwen en testen.

<http://developer.android.com/sdk/index.html>

8.3 Verschillende informatie

En voor alle vragen klein tot groot op te lossen, rare errors/bugs te vinden of een diepere verstanding te krijgen van bepaalde termen:

<https://www.google.be/>

<http://www.wikipedia.org/>

<http://stackoverflow.com/>