

Android toepassing

Groep 6

Blog <http://mumetech.wordpress.com/>

Bram Truyaert, 3^{de} bachelor Informatica
Laurens Van Keer, 3^{de} bachelor Informatica

Abstract—Dit is het verslag van groep 6 voor de tweede multimedia opdracht. Wij hebben een Android-toepassing genaamd “GeoTrack” uitgewerkt. Deze applicatie maakt gebruik van de Android GPS functionaliteit om de geografische locaties van Last.fm gebruikers te verzamelen in een eigen database. In combinatie met de Last.fm API kunnen de gebruikers zo op de hoogte gesteld worden van de locaties en muziekinteresses van anderen in hun buurt. De belangrijkste uitdagingen waren het leren werken met de Android SDK in korte tijd en het aan de praat krijgen van de GPS functie in combinatie met de Google maps component.

Ingediend op: maandag 2 november 2009



1 IDEE

Het oorspronkelijk idee was om onze Flex-applicatie¹ te porten naar het Android platform. Het grote voordeel hiervan is dat we slechts minimale aanpassingen moeten doen aan het storyboard en de server-side code niet meer hoeven aan te passen. Echter, na het concept herbekeken te hebben, stelden we ons vragen over het praktisch nut van zo'n toepassing op een Android toestel.

We overwogen het idee om de applicatie uit te breiden met functionaliteit om – op basis van een geografische locatie – opkomende evenementen weer te geven. Dit nieuwe concept kwam ook in ons eerste storyboard terecht. Naderhand zagen we in dat deze toevoeging het geheel inconsistent maakt, dus zochten we naar een nieuw idee.

Het uiteindelijke concept hebben we pas erg laat bedacht (vrijdag 30 november), maar aangezien we al een groot deel van het eerste idee hadden geïmplementeerd en dus al behoorlijk goed met de Android SDK overweg konden, zijn we toch nog in geslaagd een werkende versie te realiseren.

Het doel van “GeoTrack” is nu het verzamelen en verspreiden van geografische locaties tussen Last.fm gebruikers. Wanneer een gebruiker inlogt met zijn Last.fm accountnaam op onze locatie, worden zijn geografische coördinaten m.b.v. de GPS-functie doorgezonden naar onze server. Vervolgens worden de coördinaten van andere, dichtbijgebruikers met een gelijkaardige muzieksmaak teruggegeven. De gebruiker kan dus zo te weten komen of er anderen in zijn of haar buurt zijn met een gelijke muziekinteresse.

2 STORYBOARD

Zie figuur 1 voor de finale versie van ons storyboard.

Wanneer de applicatie opstart, krijgt de gebruiker een eenvoudig “login” scherm te zien. Hiermee kan met iedere bestaande Last.fm username ingelogd worden.

¹ Onze Flex-applicatie visualiseert en vergelijkt de populariteit van muzieknnummers en -artiesten in verschillende landen.

Het voordeel van deze aanpak is dat er niet met een paswoord ingelogd moet worden over een potentieel onveilige verbinding. Uiteraard is het grote nadeel hiervan dat iedereen met om het even welke username kan inloggen. De huidige versie is dus helaas nog erg vatbaar voor misbruik.

Vervolgens worden de gebruikersnaam en de GPS coördinaten doorgestuurd naar een PHP script. Dit script slaat de gegevens op in een eigen MySQL database en zendt een XML-file met de gegevens van andere gebruikers terug naar de client.

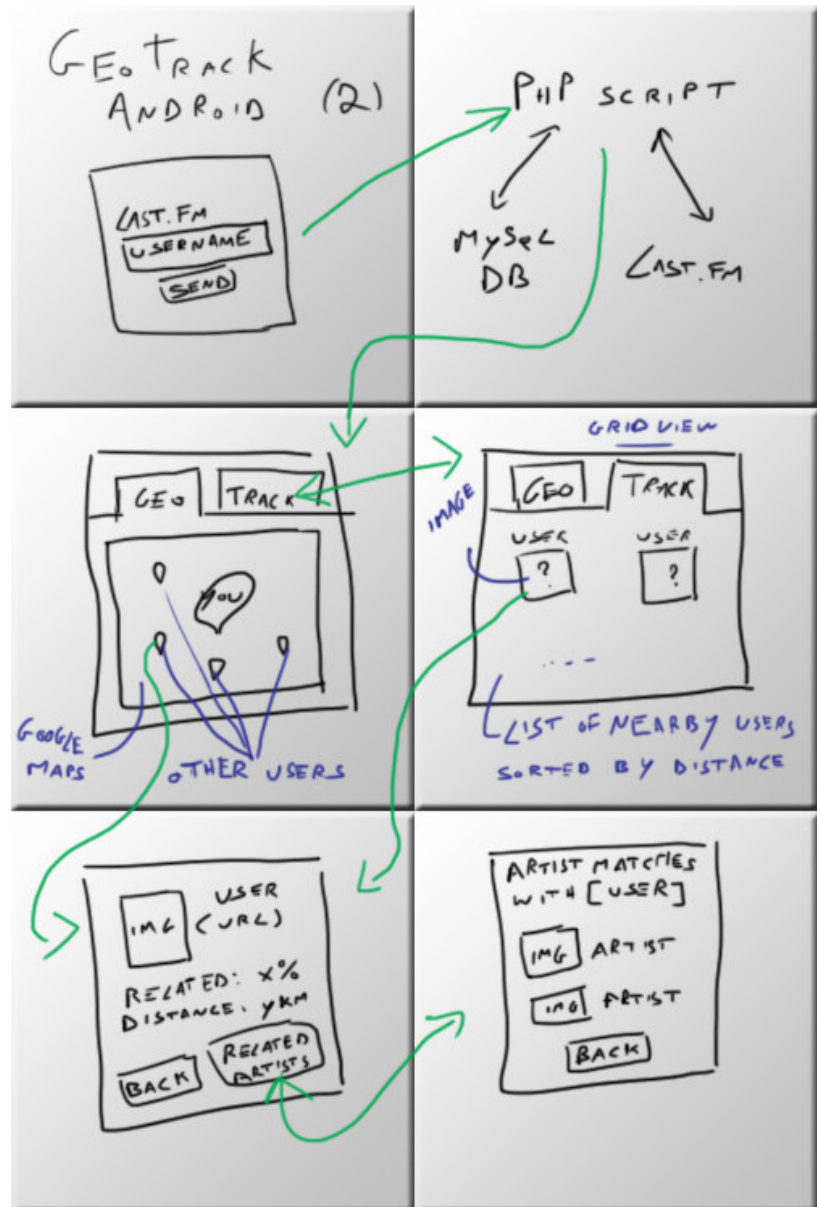
De gebruiker krijgt nu de keuze tussen twee tabs, met de titels "Geo" en "Track".

In de "Geo"-tab krijgt de gebruiker een Google map met de locaties van anderen erop aangeduid. Op zo'n locatie klikken (of aanraken, op een mobiel toestel) brengt hem/haar naar een detailvenster van de gebruiker in kwestie.

In de "Track"-tab worden dezelfde gegevens voorgesteld, maar in een *gridview*. Dit is dus een soort lijst van de afbeeldingen van dichtbijgebruikers. De gebruiker wordt ook naar het detailvenster gebracht, wanneer hij/zij op een afbeelding klikt.

Het detailvenster van een gebruiker geeft wat essentiële informatie weer: de *Tasteometer* score², de afstand en een "related artists"-knop. Deze knop brengt de gebruiker naar een venster dat een lijst van artiesten "matches" weergeeft. Dit zijn dus de artiesten die de twee gebruikers gemeenschappelijk hebben in hun Last.fm libraries.

Dit zijn alle features die de finale versie gehaald hebben. We hadden nog meer ideeën zoals de combinatie met een locatie-gebaseerde chatapplicatie³, het toevoegen van een detailvenster voor de artiesten, routebepaling naar de andere gebruikers, ... Uit tijdsgebrek was het helaas onmogelijk deze features nog te implementeren.



² De Tasteometer score geeft weer hoe goed de muzieksmaak van twee gebruikers overeenkomt (<http://www.last.fm/api/show?service=258>).

³ <http://code.google.com/p/androidchat/>

3 SOFTWARE-ONTWERP

SERVER-SIDE

Visual Paradigm for UML, Standard Edition (K.U. Leuven)		
geo_users		
+id	integer(11)	Nullable = false
user	varchar(15)	Nullable = false
lastip	varchar(15)	Nullable = false
time	integer(11)	Nullable = false
latitude	double(11)	Nullable = false
longitude	double(11)	Nullable = false

Figuur 2 - De SQL tabel "geo_users"

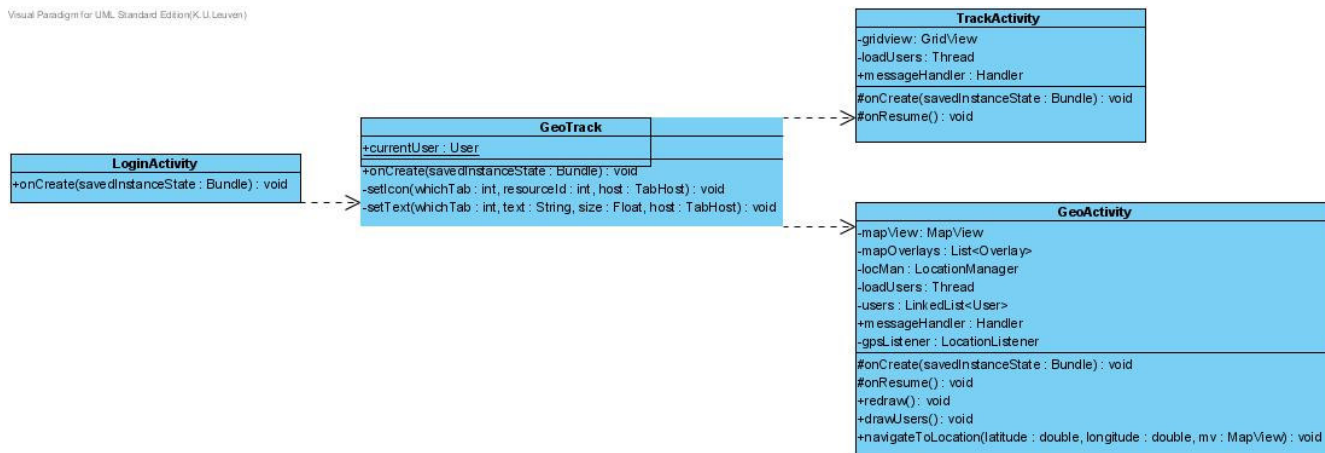
Het server-side gedeelte van onze toepassing is vrij eenvoudig. Dit bestaat uit slechts twee PHP scripts, één MySQL tabel en een PHP library⁴ voor de Last.fm API.

Het doel van de server is het bijhouden van alles gebruikers van de applicatie en het maken van Last.fm API oproepen.

De structuur van de SQL tabel "geo_users" is in figuur 2 te zien. De tabel houdt dus voor iedere gebruiker de geografische coördinaten (longitude, latitude), tezamen met het laatste tijdstip waarop de gebruiker zich heeft aangemeld en het IP-adres waarmee dit gebeurde (om misbruik tegen te gaan).

CLIENT-SIDE

Het client-side gedeelte bestaat uit zes Android activities. Elke Activity komt overeen met een XML bestand (met daarin de grafische componenten) uit de /res/layout/ map. Zoals in de meeste Android applicaties, worden activities opgeroepen d.m.v. tijdelijke intent-objecten.



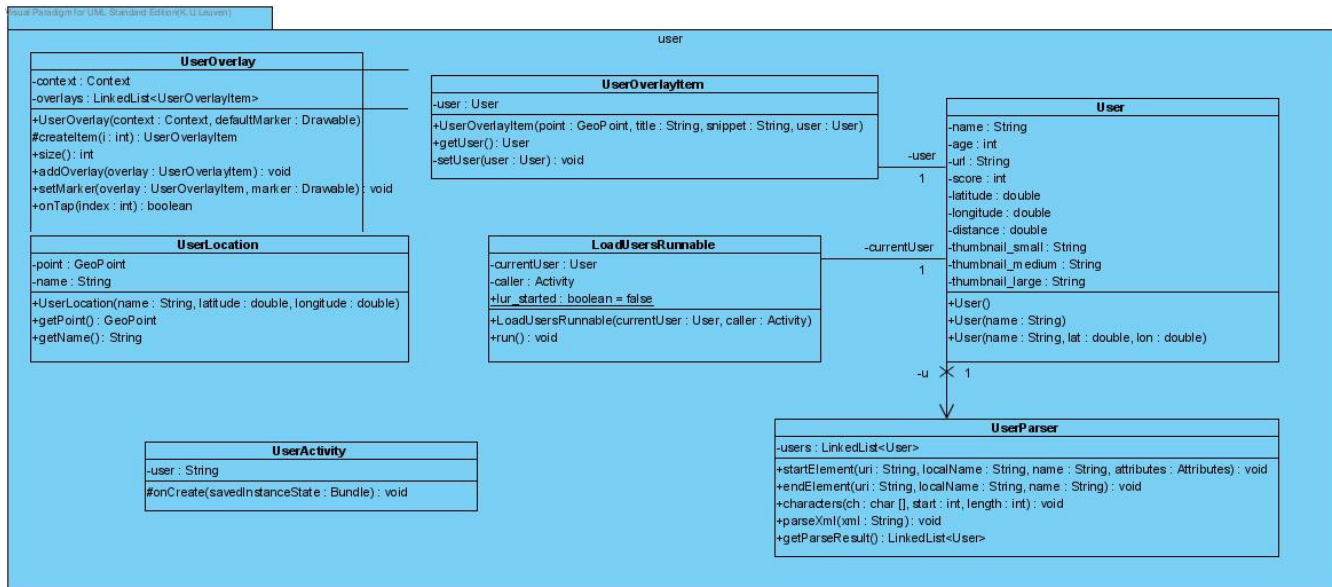
Figuur 3 – UML diagram van de belangrijkste activities bij het opstarten van de applicatie

In figuur 3 zijn de activities te zien die bij het opstarten van de "GeoTrack" applicatie belangrijk zijn. De LoginActivity komt bijvoorbeeld overeen met de login-view (/res/layout/login.xml). Zodra een username is ingevuld, wordt in de LoginActivity m.b.v. een intent de GeoTrack activity opgeroepen. Deze dient als een "container" voor de twee tabs ("Geo" en "Track"). De twee tabs komen elk ook met een activity en een view overeen.

In de "GeoActivity" wordt een Google map gegenereerd. Er wordt eveneens een thread gestart (die gebruikt maakt van de hulpklasse "LoadUsersRunnable" in de user package) die voor iedere gebruiker – gegenereerd door het PHP script – een klikbaar icoontje maakt.

Ook de "TrackActivity" maakt gebruik van de user package en een thread om een lijst van gebruikers te maken. Alle "user" hulpklassen worden in het UML diagram in figuur 4 weergegeven.

⁴ Last.fm API for PHP, door Matt Oakes, <http://www.matto1990.com/projects/phplastfmap/>



Figuur 4 - UML diagram van de "user" package

De "User" klasse stelt een Last.fm gebruiker voor. Met de "UserParser" wordt de XML van de server verwerkt. Deze wordt aangeroepen in een thread die de "LoadUsersRunnable" interface gebruikt. De "UserActivity" hoort bij het detailvenster van een gebruiker. Ook bij deze Activity hoort een XML bestand: /res/layout/user.xml.

Om de lijst van artiesten die twee gebruikers gemeenschappelijk hebben te laden, wordt een gelijkaardig systeem gebruikt. Enkel de weergave is verschillend: hiervoor wordt een *ListView* i.p.v. een *GridView* gebruikt, maar dat is slechts een verschil in implementatie. Het model van de "artist" package is dus analoog aan de "user" package.

4 IMPLEMENTATIE IN ANDROID

HET INLOGGEN EN OPSTARTEN

Wanneer de gebruiker een Last.fm username heeft ingevuld en op de "Send!" knop duwt, wordt m.b.v. een intent de username doorgegeven aan de "GeoTrack" Activity. Deze gaat de twee tabs inladen en met de ontvangen gebruikersnaam een statische "currentUser" variabele aanmaken.

We beseffen dat dit laatste niet echt strookt met de OOP-principes. We vinden dit echter wel de meest logische en eenvoudigste aanpak. De "currentUser" variabele moet namelijk van overal bereikbaar zijn. Het telkens doorgeven van deze variabele via een intent zodra een nieuwe activity aangemaakt moet worden, leek ons te omslachtig.

HET LADEN VAN DE GOOGLE MAP EN DE GPS-COÖRDINATEN

Het maken van tab met daarin een Google map is eenvoudig: dit is slechts het aanmaken van een subklasse van de Android klasse "MapActivity" (hier dus "GeoActivity").

Bij het inladen van de GPS-coördinaten ondervonden we wel moeilijkheden. Wij hebben een LocationListener object in de "GeoActivity" activity aangemaakt. Dit object "luistert" naar GPS updates. De moeilijkheden ontstonden tijdens het testen van deze feature: het updaten van de GPS coördinaten m.b.v. het "DDMS" perspectief in Eclipse werkte namelijk niet. Na wat zoeken vonden we een alternatieve testmethode: door via Telnet in Windows het lokale adres localhost:5554 te openen, kunnen bijvoorbeeld met het commando "geo fix 4.69765 50.8668" de GPS coördinaten geüpdatet worden.

HET LADEN VAN DE USERS

Het laden van dichtbij Last.fm gebruikers met een gelijkaardige smaak komt neer op het aanroepen van het “geo_users.php” script en het *parsen* van de resulterende XML.

Het PHP script aanvaardt drie input (`$_REQUEST`) variabelen: de gebruikersnaam (“user”), een breedtegraad (“lat”) en een lengtegraad (“lon”). Het script zal checken of deze gebruiker reeds in onze eigen database zit of niet en zal dan een rij in de tabel updaten of aanmaken. Vervolgens wordt een Last.fm API oproep gemaakt voor iedere andere gebruiker in onze database die in de afgelopen 15 minuten in de buurt (afstand < 10 km) van de huidige gebruiker zat. Het resultaat is een lijst van *Tasteometer* scores en extra informatie over de users. Hiervan worden enkel de gebruikers met een gelijkaardige muzieksmaak (score > 40%) in een XML bestand teruggeven.

De constanten zoals de 10 km, de 15 minuten en de minimale Tasteometer score zijn uiteraard allemaal erg arbitrair en vereisen testen. In de huidige “GeoTrack” versie worden deze constanten letterlijk gedefinieerd in het PHP script, maar een alternatieve (en betere) aanpak zou bijvoorbeeld een optie-menu in de Android applicatie kunnen zijn.

Het parsen van de XML gebeurt d.m.v. een eigen SAX parser: “UserParser”. Deze parser wordt in de loadUsers thread geladen, die gebruikt maakt van de LoadUsersRunnable interface. Deze interface beschrijft de functionaliteit van de thread: het inladen van de XML en het oproepen van de “UserParser” XML parser.

HET TEKENEN VAN DE “MARKERS”

Voor iedere gebruiker moet een zogenaamde “marker” in de map in de “Geo” tab getekend worden. Dit is een klikbaar icoontje dat naar het detailvenster voor de gebruiker in kwestie leidt. Hier hadden we ook moeilijkheden mee en dit voornamelijk door de beperkte en gebrekkige officiële documentatie. Dankzij mailinglists⁵ en het *anddev.org* forum⁶ vonden we de oplossingen voor onze problemen.

Voor de markers hebben we een eigen “ItemizedOverlay” klasse gemaakt: “UserOverlay”. Deze heeft een “onTap(int)” methode die een UserActivity activity start met een intent.

HET LADEN VAN DE ARTIESTEN

Om de artiesten die twee gebruikers in hun bibliotheek gemeenschappelijk hebben, hebben we een nog een PHP script geschreven: “related_artists.php”. Dit script aanvaardt twee parameters: “user1” en “user2”. Het resultaat is een XML bestand met daarin de artiestennamen en links naar afbeeldingen die erbij horen.

Het laden van deze XML file in onze View is bijna identiek aan het laden van de users, met het verschil dat er een ListView i.p.v. een GridView gebruikt wordt.



Figuur 5 - De “Geo”-tab en enkele “markers”

⁵ <http://code.google.com/p/android/issues/detail?id=2214>

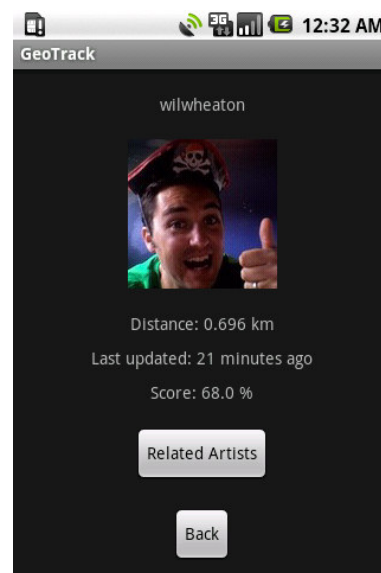
⁶ <http://www.anddev.org/how-to-create-a-clickable-overlay-t6871.html>

5 RESULTAAT

De applicatie kan zeker nog enkele toevoegingen gebruiken. Enkele voorbeelden zijn een uitgebreidere beveiliging van de PHP scripts en de database, een optiemenu in de Android applicatie, een combinatie met een chatapplicatie, het weergeven van naar welke nummers de gebruikers in je buurt naar aan het luisteren zijn, ...

Toch zijn we toch tevreden van het resultaat. We hebben op een relatief korte tijd een werkende Android mashup gemaakt en erg veel geleerd over de Android SDK.

De aanvankelijke bedoeling van het nieuwe concept was wel de applicatie “nuttiger” te maken, maar we beseffen ook wel dat het pas praktisch “nut” kan hebben als er een grote *userbase* is. We betwijfelen dus ook of een applicatie als “GeoTrack” in de praktijk zou aanslaan.



6 OVER ANDROID

Het grote voordeel van Android ligt voor de hand: het is een open framework en in tegenstelling tot iPhone en iTouch applicaties, is er geen lastig proces van goedkeuring nodig en kan de applicatie op verschillende hardware draaien. Bij dit laatste stellen we ons toch vragen. In de emulator lijkt onze mashup te werken, maar zal dat op elke mogelijke Android Phone ook zo zijn?

Het was aangenaam om in de vertrouwde Eclipse omgeving te kunnen werken. De Android debugger en het DDMS perspectief heeft ons talloze bugs helpen oplossen. Helaas is dit nog niet 100% stabiel, want de GPS updater lijkt bij ons niet te werken en de emulator wordt niet altijd correct gekoppeld aan de Eclipse omgeving.

Android maakt niet gebruik van een Swing of AWT UI, maar heeft een eigen UI systeem en hiërarchie. Dit maakt de leercurve iets steiler, maar wij vinden het wel een gemakkelijk te gebruiken systeem. De keuze aan Android UI componenten is groot en ze zijn bovendien aanpasbaar. De optie om strings af te splitsen van de views vinden we ook een voordeel. Zo was het voor ons bijvoorbeeld gemakkelijker om via SVN ons project te updaten, zonder telkens onze apikey te moeten wijzigen in de programmacode.

Het grootste nadeel van Android is volgens ons het gebrek aan uitgebreide documentatie. Zeker in vergelijking met Flex is dit erg teleurstellend. We vonden vaak de oplossingen voor onze problemen op onofficiële sites en forums.

7 EXTRA

Een video-opname van onze toepassing is te vinden via onze blogpost op <http://mumetech.wordpress.com/2009/11/02/geotrack-android/>.

8 BESLUIT

Voor deze opdracht hebben we goed nagedacht over hoe we onze oorspronkelijke Flex applicatie naar een praktische Android toepassingen konden porten. Uiteindelijk hebben we besloten het hele concept van onze mashup aan te passen. Het doel is nu het verzamelen en verspreiden van de geografische locaties van Last.fm gebruikers.

Na de feedback en zelfreflectie over onze Flex opdracht, hebben we onze aanpak drastisch veranderd. Het storyboard dat we nu tekenden komt veel meer overeen met het eindresultaat. We hebben op voorhand de (beperkte) Android API documentatie bekeken en onze mogelijkheden verkend. De implementatie ging vrij vlot, op enkele moeilijkheden na (zie deel 4).

We zijn behoorlijk tevreden van het resultaat, hoewel er nog een aantal tekortkomingen zijn. Voor de volgende opdracht gaan we proberen vroeger een definitief storyboard af te hebben, want we hebben gemerkt dat de implementatie erg vlot verliep nadat we exact wisten wat het resultaat moest zijn.

APPENDIX

	Laurens Van Keer	Bram Truyaert	Totaal
Storyboard	2	1	3
Ontwerp	2	1	3
Implementatie cliënt	31	28	59
Implementatie server	6	0	6
Verslag schrijven	4	1	5
Video maken	1	0	1
Totaal	46	31	77