|  |
| --- |
| Op excursie met android! |
| Mobiele Apps 3 |
| Tom Boelen |

[1 Projectomschrijving 2](#_Toc440860582)

[2 Een klein overzicht 2](#_Toc440860583)

[2.1 De Database 2](#_Toc440860584)

[2.1.1 Questions 3](#_Toc440860585)

[2.1.2 ScoreBoard 3](#_Toc440860586)

[2.2 De Activity’s 3](#_Toc440860587)

[3 Het AndroidManifest 4](#_Toc440860588)

[3.1 Android.Permissions 4](#_Toc440860589)

[3.2 SDK & Google services 4](#_Toc440860590)

[4 De activities 5](#_Toc440860591)

[4.1 HomeActivity 5](#_Toc440860592)

[4.2 MapsActivity 7](#_Toc440860593)

[4.2.1 Wat is geofencing? 7](#_Toc440860594)

[4.2.2 OnCreate 8](#_Toc440860595)

[4.2.3 OnResume 8](#_Toc440860596)

[4.2.4 OnConnected 9](#_Toc440860597)

[4.2.5 OnPause 9](#_Toc440860598)

[4.2.6 OnStop 9](#_Toc440860599)

[4.3 QuestionActivity 9](#_Toc440860600)

[4.4 Alle vragen opgelost? 11](#_Toc440860601)

[4.5 ScoreBoardActivity 12](#_Toc440860602)

[5 Mocking 13](#_Toc440860603)

[5.1 Wat is het? 13](#_Toc440860604)

[5.2 Voordelen? 13](#_Toc440860605)

[6 Conclusie 14](#_Toc440860606)

## Projectomschrijving

Een excursie met de GSM! Dat is iets waar in het onderwijs veel over nagedacht wordt. Maar de kennis en de beperkte tijd , zijn een probleem. Er bestaan (bijna) geen programma’s die leerkrachten kunnen gebruiken om makkelijk applicaties te maken. En al zeker geen waar geofencing in gebruikt wordt.

De bedoeling van mijn applicatie is dus om een aantal plaatsen op de kaarten aan te duiden. De leerlingen moeten zelf naar deze plaatsen navigeren , en op het moment dat ze daar aankomen moet er een vraag (of meerdere) verschijnen die ze moeten oplossen. Er is ook een mogelijkheid om dan gewoon informatie te verschaffen. De groepsnaam , klasnaam , en tijd wordt op dat moment bijgehouden. Ook worden er scores toegewezen tijdens het maken van de oefeningen. Een juist antwoord is +200 terwijl een fout antwoord -100 is. Je kan meerdere maal fout antwoorden dus ze moeten er toch goed over nadenken.

Na deze excursie , kunnen ze hun scores vergelijken met andere groepen die voordien op deze GSM gewerkt hebben.

## Een klein overzicht

Voor dat we beginnen aan de uitleg van mijn project , is het misschien handig om eerst een klein overzicht te hebben van het volledig programma. Het bestaat uit 2 grote delen , die constant met elkaar in verbinding zijn.

### De Database

#### Questions

De database is opgedeeld in 2 tabellen. Ten eerste hebben we de QUESTIONS tabel.  
Elk Question bezit een uniek ID , een naam , de vraagstelling zelf , enkele antwoorden , het juist antwoord en een veld wat aangeeft als deze vraag reeds beantwoord is of niet.

#### ScoreBoard

De SCOREBOARD tabel bestaat uit een kolom ID , Groupname , classname en de score.  
Zoals wel duidelijk wordt bij het vernoemen van deze kolommen, is dit een tabel waar de gegevens van de deelnemende mensen wordt bijgehouden. Het is belangrijk om te weten dat deze groupsID over heel het programma moet bijgehouden worden. Met deze variabele kunnen we veel query’s uitvoeren om gegevens op te halen.

### De Activity’s

Dit is natuurlijk een iets makkelijker overzichtje dan de Database. In deze applicatie maakt men gebruik van 4 activity’s. De HomeActivity om de excursie te starten en persoonlijke gegevens in te vullen , de MapsActivity om de map met bijbehorende fences te tonen , de QuestionActivity om bijbehorende vragen te stellen en de scoreboardactivity om een overzicht van de scores te geven.

De MapsActivity maakt gebruik van een GeofenceTransitionReciever om geofencing mogelijk te maken en MyPlaces om de nodige excursiepunten ter beschikking te hebben.

## Het AndroidManifest

Het androidManifest beschikt over essentiele informatie over de applicatie voor het android systeem. Deze informatie moet het android systeem hebben voor het enige code kan uitvoeren

### Android.Permissions

Om gebruik te maken van enkele ingebouwde functies op je android-gsm moet je permissies aanvragen. Hieronder zie je bijvoorbeeld permissies om gebruik te maken van de locatie , wifi , opslagbeheer , …

**<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" /><uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.ACCES\_GPS" />  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.ACCESS\_MOCK\_LOCATION" />**

### SDK & Google services

Ook de SDK-versie die je wil gebruiken moet je hier instellen. Dit kan belangrijk zijn voor nieuwe of oudere versies.

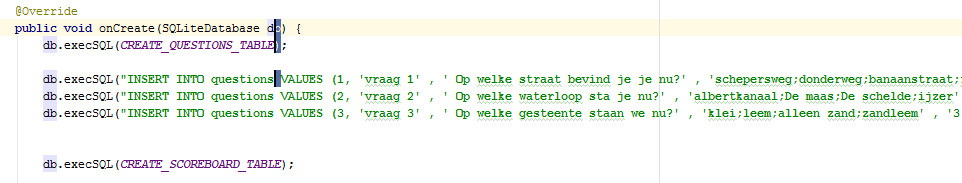
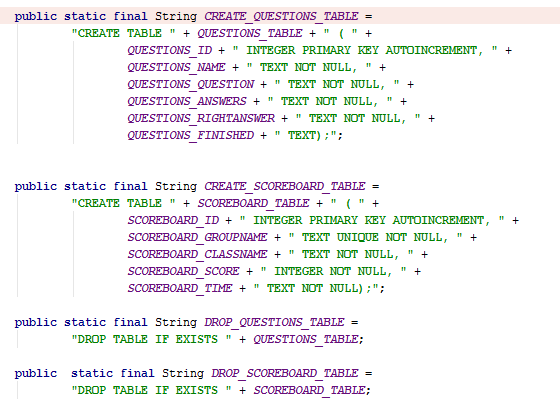
<**uses-sdk  
 android:maxSdkVersion="23"  
 android:minSdkVersion="17"  
 android:targetSdkVersion="17"** />

Hieronder zie je meta-data. Deze meta-data is nodig om gebruik te kunnen maken van Google Services en de Google API. Zo zend je een API\_KEY door naar Google.

**<meta-data  
 android:name="com.google.android.gms.version"  
 android:value="@integer/google\_play\_services\_version" />  
 <meta-data  
 android:name="com.google.android.maps.v2.API\_KEY"  
 android:value="AIzaSyDPXkI0evQLNNk9e0idMEQg1iM\_nDgrvWI" />**

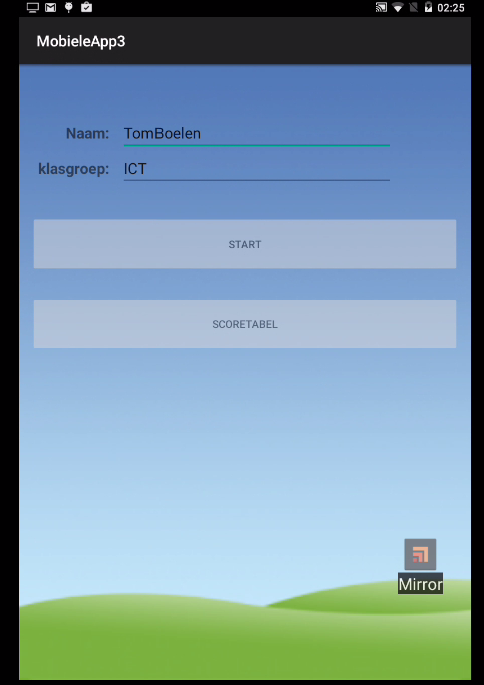
## De activities

### HomeActivity

Voordat de applicatie op de eerste activity-layout terecht komt, is er in tussentijd al heel wat gebeurt. Als er nog geen database bestond , dan is deze in tussentijd aangemaakt. Samen met de tabellen en mogelijk al ingevoegde waardes. Hieronder zie je voorbeelden van code die dan reeds uitgevoerd zijn.

Na deze heel integratie van de database , komen we uiteindelijk op het HomeScherm terecht.  
Hier kan je voor 2 opties kiezen. Je kan kiezen om je gegevens in te voeren en te starten , of eerst naar de scoretabel te kijken. Bij het klikken op de scoreTabel , opent een nieuwe activity om alle je scores te doorlopen en te tonen.

Op het moment dat je op de startbutton klikt , gebeuren er 3 zaken.

1. Er wordt gecontroleerd als de input (Naam & klasgroep) correct is. Dit wil zeggen : geen spaties , niet korter dan 3 letters , en de naam is niet eerder teruggevonden in de database.
2. Als bovenstaande controle geen fouten heeft gevonden , kan men deze data in de Scoreboard-table gooien. Je bent vanaf dan een groep die meedoet aan de excursie
3. Als bovenstaande correct gebeurd is , start je de nieuwe MapsActivity

### MapsActivity

De mapsactivity is de meest moeilijke activity. Dat is eigenlijk meer dan normaal , want het grootste deel van het programma gebeurd tijdens deze activity. Om die reden maak ik tijdens de uitleg van deze activity gebruik van het life-cycle-event. Niet alle methodes komen aan bod , maar wel de meeste belangrijke. Maar voor dat ik met de methodes verder ga , is het belangrijk om eerst wat te vertellen over Geofencing.

#### Wat is geofencing?

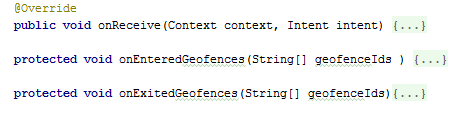
Bij geofencing wordt constant je locatie in de gaten gehouden om bepaalde acties in je applicatie uit te voeren.Zover zelfs , dat er een bepaalde actie wordt uitgevoerd terwijl je de ‘fence’ binnentreed , er in rond dwaalt en de fence buitengaat. Een fence is een gebied dat gecreëerd / aangeduid wordt door de maker op de map. Bij onze applicatie zit er een fence rond onze campus , rond mijn huis en rond plopsaland.

#### OnCreate

Op het moment dat de activity aangeroepen wordt , gaat men eerste de OnCreate() – functie aanroepen. Deze zorgt ervoor dat de map opgebouwd wordt en de mogelijke SetOnClickListeners geconfigureerd worden.

Tijdens het opbouwen van deze map gebeuren er meerdere zaken.   
Als eerste gaat men de google services gebruiken om de GoogleMap aan te roepen en in onze bijbehorende fragment te steken. Als tweede gaat men dan de “fences” opbouwen die men nodig heeft om de excursie te laten slagen. Deze fences zijn gebieden waar leerlingen later naar toe moeten wandelen om bijbehorende vragen ter beschikking te krijgen.  
Deze fences worden opgebouwd met een bepaalde icoontje en een radius rond dit icoontje.

Op het moment dat de map is opgebouwd , gaat men reeds de MonitorFences() – methode oproepen. Het is natuurlijk heel erg belangrijk om , op het moment dat de applicatie actief is , te bekijken als de gebruiker misschien wel de fence binnenwandeld. Want moest dit zo zijn, dan is het de bedoeling dat er natuurlijk iets gebeurd. Om die reden maken we in ons project gebruik van een klasse GeofenceTransitionReciever die wordt opgeroepen door de Geofence - googleApiClient ( herinner de API-key die we moesten invullen in het android manifest om hiervan gebruik te maken) . Als je de methodes van deze klasse nauw gaat bekijken,



dan zie je dat men duidelijk een methode heeft voor als men een geofence binnentreed of er uit weggaat. In deze klasse kan je dus bepalen wat je dus moet doen als iemand bijvoorbeeld de geofence van de campus binnenwandeld. Het is dus ook belangrijk dat deze klasse constant communiceert met de MapsActivity om door te geven wat er moet gebeuren.

#### OnResume

In OnResume moet er niet zoveel gebeuren. Er moet even gecontroleerd worden als de map nog beschikbaar is. Die kan immers in de onStop() – methode verwijderd zijn. Ook worden hier flags & providers toegewezen. Deze providers heb je nodig om gebruik te kunnen maken van je GPS , je network of zelfs voor de MOCK location.

#### OnConnected

OnConnected is natuurlijk een heel erg belangrijke methode. Deze wordt uitgevoerd zolang men actief is en zolang de map dus ook visueel beschikbaar is voor de gebruikers. Tijdens deze methode moet men natuurlijk constant in contact staan met de eerder uitgelegde GeofenceTransitionReciever. Men moet constant ‘penden’ met deze klassen om te bekijken wat er gebeurt terwijl je geconnecteerd bent.

Ook de locatieUpdates worden tijdens deze methode uitgevoerd. Verander je van locatie? Dan krijg je dat direct door van de StartLocationUpdates()-functie en kan dit direct aangepast worden op de map zelf.

#### OnPause

Tijdens de OnPause – functie gaan we niet zo heel veel doen.  
Hier gaan we de eerder toegewezen FLAGS en providers weer verwijderen zodat er tijdens de inactiviteit niet meer gebruik kan gemaakt worden van de GPS of het netwerk.

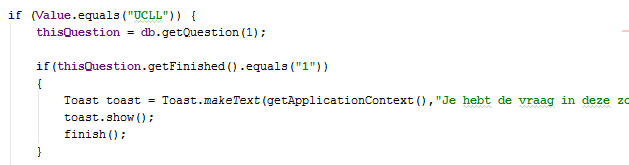
#### OnStop

Op het moment dat de applicatie de OnStop() – functie uitvoert , gaat we volledig de connectie met onze GoogleApiClient verbreken.Onze applicatie vanaf nu alleen nog maar werken als die terug vanvoor begint aan de cyclus. Bij OnCreate dus.

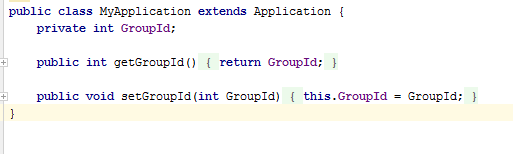
### QuestionActivity

De QuestionActivity wordt getriggerd op het moment dat je één van de geofences binnenloopt.   
Op dat moment krijg je een bepaalde vraag die je moet oplossen om verder te kunnen.

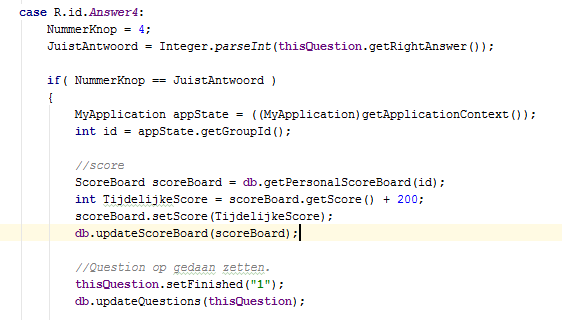
Het speciale aan deze activity is eigenlijk dat er altijd dezelfde activity gebruik wordt bij het binnenlopen van eender welke geofence. Het enige wat varieert is de vraag die wordt opgehaald uit de dabatase. Bij het binnenlopen van een geofence , krijg de questionActivity een ID mee. Dit ID wordt gebruikt om de bepaalde vraag in de database op te halen.



De backbutton wordt geblokeerd zodat men niet terug naar de mapsactivity kan als men de vraag niet weet. Het moeilijke aan deze activity was eigenlijk het feit dat je niet wist welke groep deze vragen aan het oplossen was. Je moet natuurlijk op één of andere manier weten als deze groep deze vraag wel of niet gehad heeft? Het heeft geen zin om één groep dezelfde vraag keer op keer te stellen als ze de zelfde geofence inlopen. Om die reden wordt er bij het creëren van de groep (de HomeActivity) , het GroupsID bijgehouden in een speciaal object zodat deze over heel de applicatie beschikbaar is.



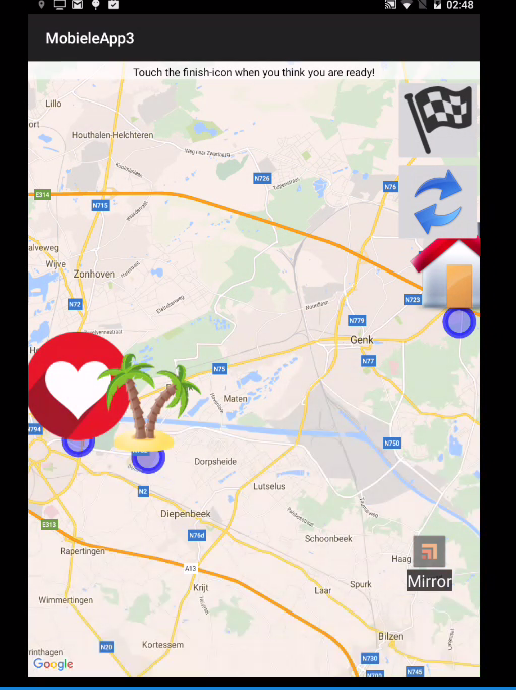
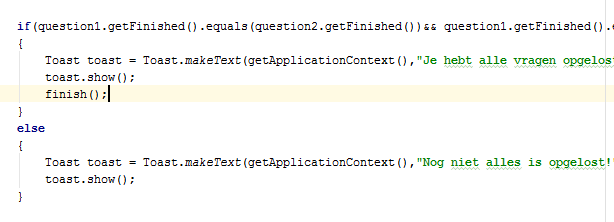
Dit object wordt ook gebruikt in de questionActivity om te bekijken als de gebruikers deze vraag al opgelost hebben of niet. Maar ook voor het toewijzen van scores bij het oplossen van de vragen.



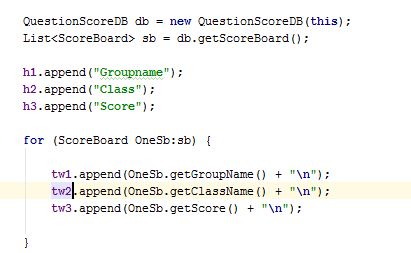
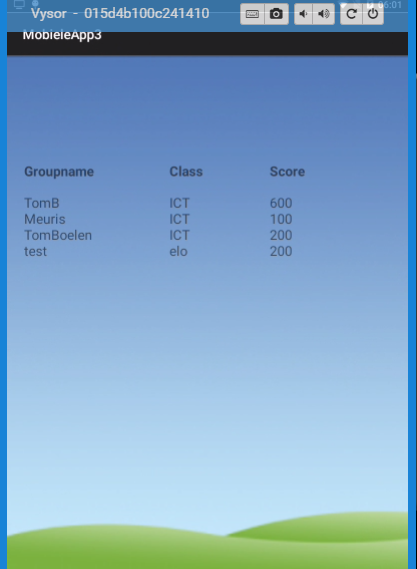
Op het moment dat een gebruiker juist klikt , krijgt hij +200 punten. Op het moment dat hij fout klikt krijgt hij -100. Ook wordt de question op ‘finished’ gezet dat hij niet nog eens dezelfde vraag moet oplossen.

### Alle vragen opgelost?

Heb je alle geofences bezocht en alle vragen opgelost?  
Klik dan op het ‘finish’-vlaggetje en er wordt bekeken als dit ook wel daadwerkelijk zo is.

Klopt dit? Dan wordt je teruggestuurd naar de HomePagina en kan je daar je score in de scoretabel gaan bekijken.

### ScoreBoardActivity

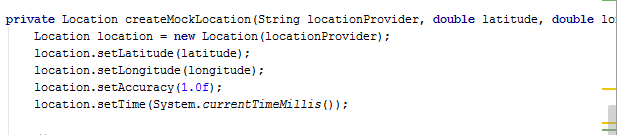
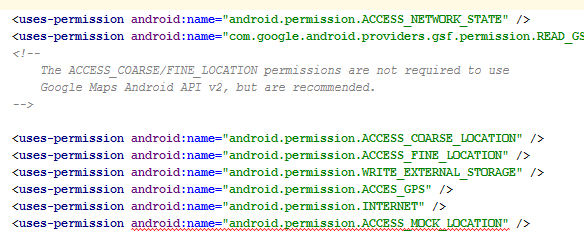
In de scoreboard activity krijgen we een overzicht van alle groepen die meegedaan hebben , hun klasnaam en de bijbehorende score. Ook dit kan je uit de database halen met de getScoreBoard() – functie. Het enige wat je dan nog moet doen , is deze door een foreach functie halen en zichtbaar maken.

## Mocking

### Wat is het?

Mocking betekent eigenlijk ‘nabootsen’. Ook in dit geval ga je iets nabootsen, namelijk je locatie.  
Je kan door gebruik te maken van Mocking gewoon je locatie aanpassen door op het scherm te klikken.

### Voordelen?

In principe wordt dat nooit gebruikt en wordt het ook ten strengste afgeraden.   
Bij mij staat het momenteel nog actief omdat ik geen emulators had die mij konden helpen om de locatie aan te passen.  
Leuk kan het wel zijn als je leerlingen steden/landen van de wereld of europa wil laten leren. Laat ze dan door mocking te gebruiken , deze landen aanduiden en het is al direct veel fijner!

## Conclusie

Voor mij was het programmeren in android een enorme aanpassing maar aan de andere kant ook wel een verrijking. Het programmeren was misschien niet makkelijker omdat we tot nu toe meer in .NET hebben geprogrammeerd , maar de mogelijkheden waren wel veel uitgebreider dan bij windows phone. Alleen al het gebruik van Google Services geeft je al veel meer mogelijkheden.

Voor mij is dit project zeker geslaagd en alle aspecten van onze lessen zijn zeker nog eens uitgediept geweest. Mobiele applicaties is iets wat altijd , en nog steeds , interesse zal opwekken voor mij.