Onderwijsgroep Professionele Opleidingen

Handelswetenschappen en bedrijfskunde

|  |
| --- |
|  |
| **Eilish Van Der Snickt** |
| Tweede bachelor - Toegepaste Informatica  Choose an item. |
|  |
|  |
| **Mobile & Internet**  Steven Ophalvens |
|  |
| Academiejaar 2018-2019 |

|  |  |
| --- | --- |
| Documentatie project | |
| Fitmap |

# Inhoudsopgave

Inhoudsopgave 3

Documentatie Fitmap 4

Project omschrijving 4

Over Fitmap 4

Functionaliteit 4

Project structuur 4

Databank 5

Klassen 5

Applicatie flow 6

# Documentatie Fitmap

## Project omschrijving

### Over Fitmap

Zoek je een applicatie waarop je gemakkelijk routes kan inplannen en bijhouden om deze later te bekijken of opnieuw af te leggen? Fitmap geeft je deze functionaliteiten.

Ben je sportief en hou je ervan om lange wandelingen of fietstochten te maken. Of wil je simpelweg weten welke route je hebt afgelegd om ergens te geraken. Dan kan je Fitmap gebruiken.

### Functionaliteit

Fitmap staat je toe om zowel op voorhand routes uit te stippelen en dan af te leggen als je te volgen terwijl je aangeeft wanneer de applicatie je moet volgen. Je kan 2 soorten routes maken:

Je kan ervoor kiezen om zelf op voorhand een route uit te stippelen eender waar ter wereld, je kan deze volgen terwijl je ze maakt of je kan ze opslaan en later opnieuw gebruiken.

Of je kan Fitmap aangeven wanneer de applicatie je moet volgen en dan zal deze op basis van locatie je route uittekenen wanneer je ze aflegt. Je kan op elk moment aangeven wanneer Fitmap moet stoppen met je te volgen en dan krijg je een overzicht van je afgelegde route.

Ook geeft Fitmap je een menu met al je opgeslagen routes, na het maken van een route kan je zelf kiezen of je deze wilt opslaan of niet. Wanneer je je route opslaat kan je deze op elk moment terug bekijken samen met andere info. Voor elke route worden er een paar gegevens berekent:

* Aantal kilometer afgelegd;
* Hoelang je onderweg bent geweest;
* aantal kilometer per uur gemiddeld;
* calorieën verbrandt gebaseerd op je gewicht.

## Project structuur

Het project is gemaakt in Android Studio. Ik gebruik Java als programmeertaal omdat ik vorig semester met Kotlin heb gewerkt. Ik wil zo veel mogelijk verschillende programmeertalen onder de knie krijgen om zo beter te kunnen programmeren.

Ik werk met Jetpack navigatie dat mij toelaat verschillende fragmenten aan te maken en elk van deze fragmenten heeft een ander lay-out scherm. Verder werk ik met een navigatie graph dat mij toelaat te relaties te leggen tussen bepaalde fragmenten om zo te kunnen navigeren tussen lay-out schermen.

Eerst en vooral heb ik een inlogscherm gemaakt en de optie om te registreren en een wachtwoord vergeten optie. Het aanmaken en het beheer van accounts wordt beheerd via Firebase authenticatie. Dan heb ik ervoor gezorgd dat de gebruiker al zijn gegevens kan veranderen; email aanpassen, wachtwoord aanpassen, account verwijderen, uitloggen, … .

Verder heb ik de map fragmenten van Google Maps toegevoegd in het project. Hiervoor moest ik licenties aanvragen om de map te mogen gebruiken.

En als laatste, wanneer ik al mijn verschillende fragmenten had en mijn project volledig gemaakt was qua lay-out en fragmenten. Dan heb ik een database aangemaakt gebruik makend van de Firebase Cloud Firestore. De Firestore houdt alle gegevens bij die de routes die verschillende gebruikers aanmaken genereren. Ook worden de gegevens opgehaald uit de Firestore wanneer deze moeten getoond worden in de applicatie.

## Databank

De databank is gemaakt in Firebase met de Cloud Firestore. Dit is een NoSQL database. Dat wil zeggen dat er geen ERD kan van gemaakt worden. Ik heb echter wel structuur in mijn database gestoken.

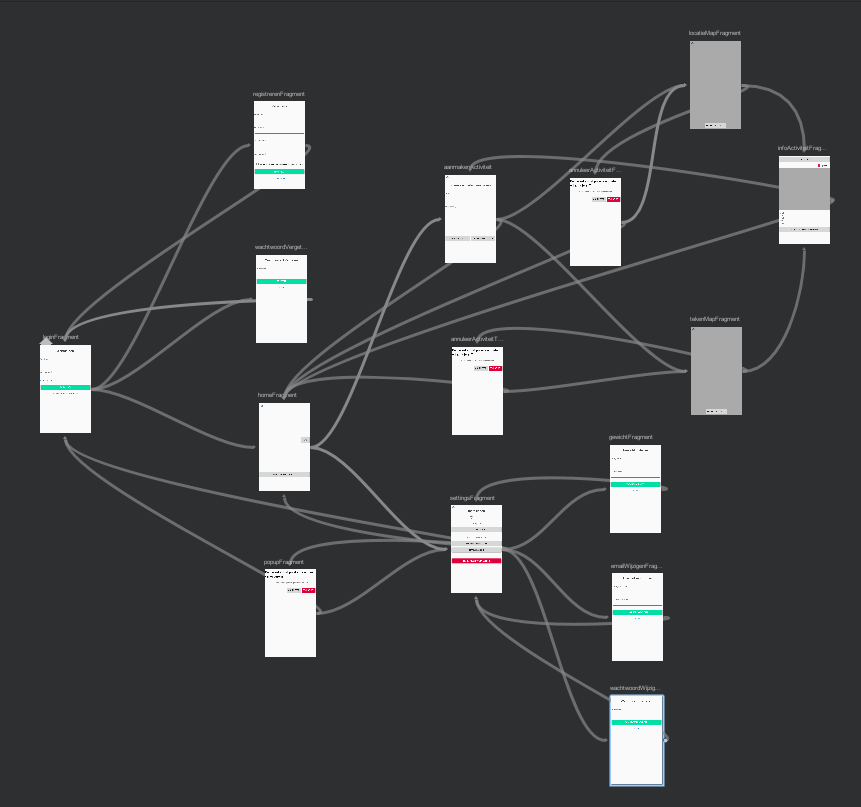
Eerst en vooral heb ik verschillende collecties gemaakt, wat je een beetje kunt vergelijken met tabellen. De collecties bevatten verschillende documenten. Documenten kan je wat vergelijken met kolommen en elk document bevat verschillende records.

Ik heb 4 verschillende collecties:

1. houdt alle titels en beschrijvingen van routes bij;
2. houdt alle geolocaties bij (een punt op de kaart; de combinatie van longitude en latitude);
3. houdt alle gegevens bij die berekend worden op basis van de verschillende routes;
4. houdt de gebruikersgegevens bij die moeten opgevraagd worden in de applicatie.

## Klassen

Aangezien ik Jetpack navigatie gebruik, maak ik veel meer gebruik van fragmenten in plaats van klassen. Hieronder vindt je een voorbeeld van het nav.graph van Fitmap. Deze toont de flow van alle fragmenten.



Ik werk ook wel met een paar klassen voor bepaalde mogelijkheden, zoals het werken met verschillende tabbladen in 1 scherm.

De ViewpagerAdapter klasse erft over van de reeds geïmplementeerde fragment FragmentPagerAdapter. Dit fragment is standaard geïmplementeerd in Java.

## Applicatie flow

Eerst en vooral moet de gebruiker zich aanmelden of registreren als deze nog geen account heeft. Dan krijgt de gebruiker een scherm te zien met alle routes die de gebruiker heeft opgeslagen in het verleden. De gebruiker kan een route kiezen en hierover meer informatie bekijken; titel, kilometers afgelegd, calorieën verbrandt, een map met de route uitgetekend, … .

Op dit algemeen menu kan de gebruiker ook naar zijn gebruikersinstellingen gaan of een nieuwe route aanmaken.

De gebruikersinstellingen staan toe dat de gebruiker zijn gegevens kan veranderen; email en wachtwoord aanpassen, … . Ook kan de gebruiker hier zijn account verwijderen en uitloggen uit de applicatie.

Wanneer de gebruiker ervoor kiest om een nieuwe route aan te maken, wordt hij naar een scherm gebracht dat hem toelaat een titel en beschrijving in te geven voor de nieuwe route en dan de nieuwe route te beginnen. Ook kan de gebruiker gewoon teruggaan naar het algemeen menu wanneer hij zich bedenkt en toch geen nieuwe route wilt maken.

De gebruiker kan kiezen om zelf een route te tekenen of de applicatie zich te laten volgen via locatie. Deze schermen hebben veel met elkaar gemeen. Ze bevatten allebei een kaart en een stop knop. Het enige verschil is dat de kaart voor de route te tekenen de functionaliteit bevat om markers te plaatsen op de kaart en zo routes getekend worden. De kaart met de locatie heeft deze functionaliteit niet. De kaart wordt telkens herladen met je locatie centraal en elke keer dat de kaart herladen wordt je huidige locatie weggeschreven naar de databank. Wanneer je op de stop knop drukt stopt de activiteit en kom je terecht op een scherm die je informatie over de juist afgelegde route geeft.

In dit scherm kan de route op een kleine kaar zijn route bekijken en ervoor kiezen om deze op te slaan of niet doormiddel van een check box. Default is deze aangevinkt. Ook kan je vanuit dit scherm ervoor kiezen om direct een nieuwe route te maken of terug te gaan naar het menu, waar je dan de info van alle routes kunt bekijken. Geïncludeerd van de nieuw gemaakte route.