

WORLD EXPLORER: EUROPE



INNOVATIVE MOBILE INTEGRATION

Gaspard Lammertyn

Inhoud

1	Introductie.....	3
2	Het doel.....	3
2.1	De App.....	5
2.1.1	De functionaliteit.....	5
2.1.2	De Smartphone functies	5
2.1.3	De User Interface	5
2.2	De admin website	9
2.2.1	De User Interface	9
2.2.2	Het profielbeheer.....	9
2.2.3	Het Databeheer	10
2.3	De API.....	13
2.4	De gebruikte technologieën	14
2.5	Huisstijl.....	15
2.6	Gebruikt in dit document.....	16

1 Introductie

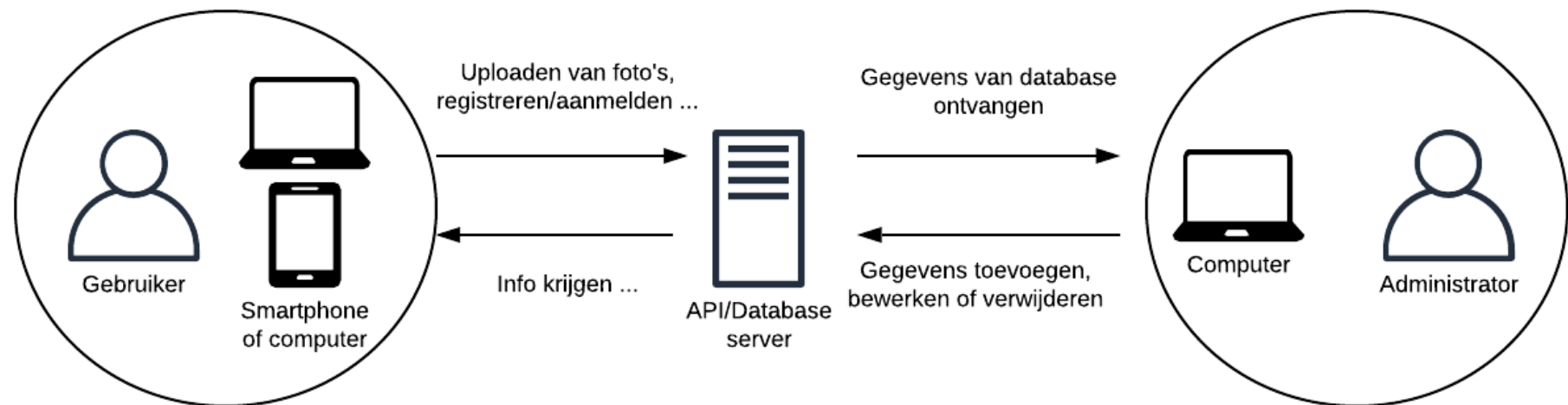
De mensheid heeft een heel lange geschiedenis achtergelaten. Soms kunnen we er vol schaamte naartoe kijken, maar 1 ding is zeker: Het zorgde voor de realiteit waarin wij nu leven. Het bracht onze waarden, onze cultuur... en door onze fouten in het verleden, weten we nu beter hoe we met elkaar moeten omgaan. Daardoor zijn er nu meer en meer mensen die overeenkomen met elkaar, ongeacht vanwaar ze vandaan komen. Hierdoor zijn wij ook nieuwsgierig naar elkaars afkomst en met de technologie van vandaag en een heleboel gegevens op het internet is deze info gemakkelijker te vinden dan ooit tevoren. Hierdoor kreeg ik het idee deze informatie eenvoudig weer te geven in één applicatie: World Explorer: Europe.

2 Het doel

World Explorer: Europe is een applicatie die info geeft over alle landen van Europa. Ik heb me beperkt tot Europa omdat ik tijdens de werkperiode tijd moet besteden aan de applicatie en alles erond. De app haalt zijn info via het internet van zowel een eigen database als externe API's. Die info bestaat uit het volgende:

- De naam van het gekozen land;
- Een foto met de vlag die hij van restcountries.eu haalt;
- Een korte beschrijving door Wikipedia of een eigen tekst;
- Enkele korte weetjes over het land zoals bv. lidmaatschap van de EU. Die info haalt hij ook via restcountries.eu;
- Een lijst met enkele typische nummers uit dat land via een afspeellijst op Spotify;
- Een collectie foto's die de gebruiker zelf kan toevoegen.

Het is ook de bedoeling dat de database beheerd wordt door een administrator via een externe website. De bewerkingen gaan dan via de API naar de database. Op de volgende pagina toon ik met een schema hoe dit proces in elkaar zit.



2.1 De App

De app zal ontwikkeld worden in de programmeertaal C# via Xamarin.Forms. Een framework speciaal gemaakt voor crossplatform app development in C#. Hij is beschikbaar voor Android en UWP. De app gaat ook de functies benutten van de laatste versie: [C# 8.0](#).

2.1.1 De functionaliteit

Als de eindgebruiker de app gebruikt, heeft hij de mogelijkheid om een land te kiezen en daarvan de info te lezen. Hij kan de landen opzoeken via een zoekbalk of via een knop op de pagina terechtkomen van het land waarin hij zich op dat moment ook fysiek bevindt. Het is ook mogelijk om te registreren/inloggen voor het uploaden van foto's en landen toe te voegen aan favorieten of een verlanglijst. Door de regelgeving van de GDPR kunnen deze foto's niet gezien worden door de medewerkers aan de applicatie.

2.1.2 De Smartphone functies

Als je een smartphone hebt is het natuurlijk ook nuttig om enkele functies ervan te benutten. Eén van die functies is de Geolocation. Hiermee gaat uw smartphone berekenen waar u zich bevindt. Dit gaat hij gebruiken om uw locatie te zoeken voor het land te vinden waar u zich momenteel bevindt (zie de functionaliteit hierboven).

Buiten dat heeft de app ook toegang tot de camera nodig.

2.1.3 De User Interface

De Interface van de app bestaat uit de volgende pagina's:

De MainPage:

Op deze pagina vindt de gebruiker een lijst van alle Europese landen op alfabetische volgorde. Per land staat links een foto met de vlag en de naam. Als de gebruiker op een land klikt/tikt komt hij op de infopagina terecht. Boven de lijst staat er ook een zoekbalk om een bepaald land te zoeken. Daarnaast is er ook een kleine knop om direct op de pagina terecht te komen van het land waarin men zich bevindt.

De InfoPage:

Bovenaan de infoPage Wordt de naam van het land weergegeven met de vlag erbij. Het eerste stuk info dat je krijgt bevat verschillende kleine weetjes over het land. Die weetjes zijn de hoofdstad, populatie en de munteenheid van het land. Daarnaast wordt er ook meegegeven of het land lid is van een regionaal blok zoals de EU.

De PhotoPage:

Dit is gewoon een carouselview met enkel de header van de pagina

De CameraPage:

Deze pagina bevat gewoon een scherm van de camera en een knop waarmee je foto's kan trekken. Het is ook mogelijk om te flitsen en van camera te veranderen.

De LoginPage:

Dit is de pagina met een simpel inlogscher. De gebruikers moeten m.b.v. hun mailadres en paswoord inloggen.. Als de gebruiker nog geen account heeft, moet hij onderaan het formulier op de link klikken/tikken naar de RegisterPage

De RegisterPage:

Op deze pagina kan een gebruiker een account aanmaken. Hiervoor moet hij de volgende gegevens invullen:

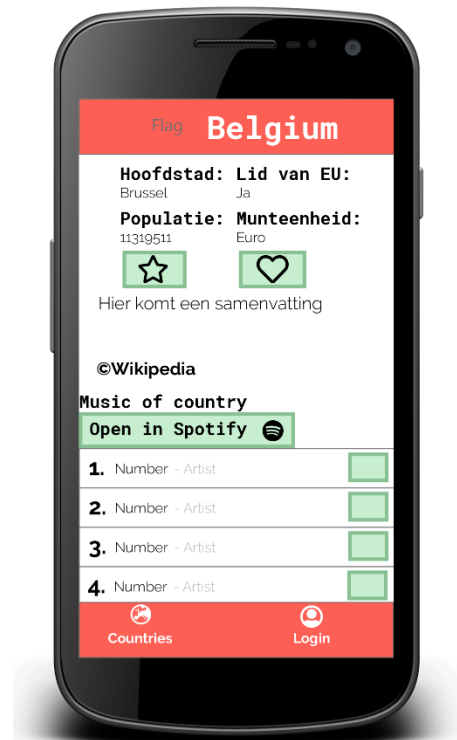
- Voornaam;
- Achternaam;
- Geboortedatum;
- E-mailadres;
- Nationaliteit;
- Wachtwoord;
- Herhaald wachtwoord.

De MyProfilePage

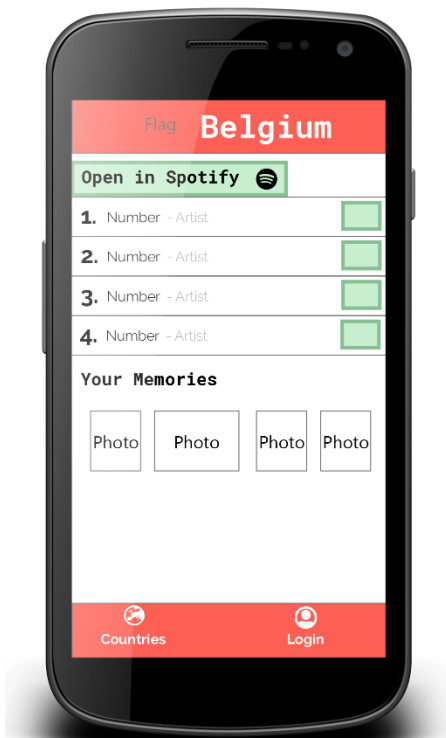
Tenslotte heb je onderaan op elke pagina een footer met de mogelijkheid om tussen de MainPage en de LoginPage te navigeren.



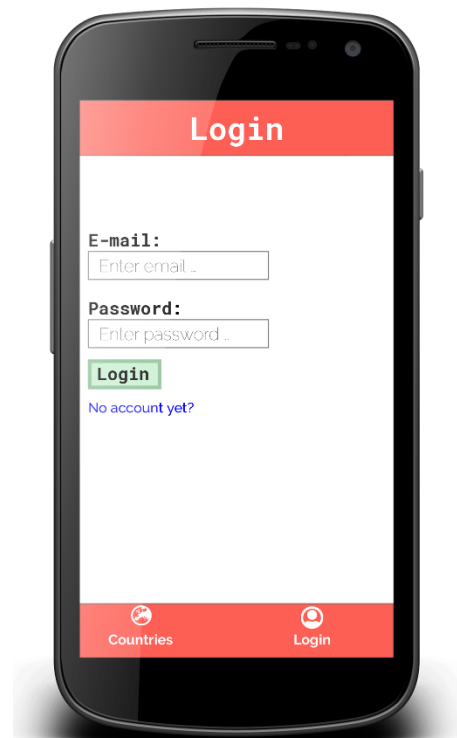
De MainPage



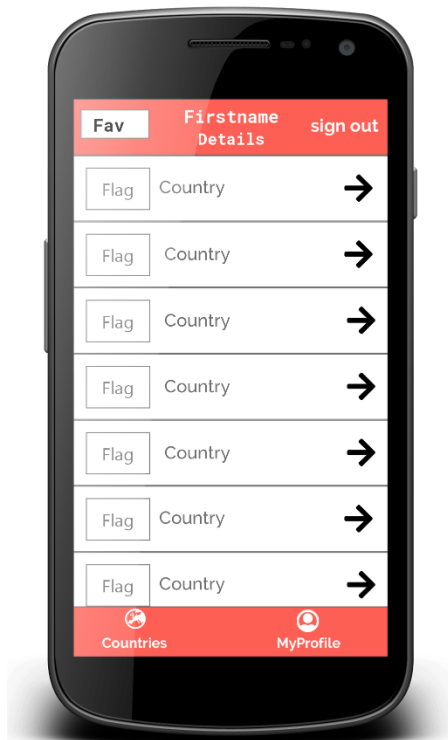
De InfoPage (eerste deel)



De InfoPage (tweede deel)



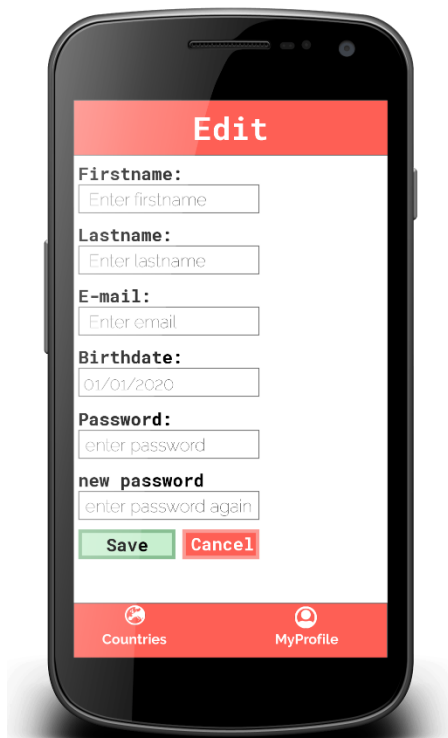
De LoginPage



De MyProfilePage



De DetailsPage



De EditPage

2.2 De admin website

Voor veiligheidsredenen kan de administrator geen bewerkingen doen op de app zelf. Daarom is het dus beter om de administratie op een aparte plaats te doen. Voor deze app gebeurt dat via een website.

De lay-out zal ik ontwerpen met [Ulkit](#). Een framework speciaal gemaakt voor een professioneel ontwerp in CSS. Om wat tijd te besparen zijn op het internet heel veel templates van voorgemaakte pagina's te vinden. De templates die ik koos zijn:

- De [Ulkit soda light admin template](#);
- De [Ulkit login](#) van de Ulkit layout examples.

Beide templates worden aangepast om zo een gelijkaardige stijl te hebben en ook om ongeveer dezelfde huisstijl als de app te krijgen.

Om de functionaliteiten van de website te programmeren, zal ik gebruik maken van Vue.js.

2.2.1 De User Interface

Elke pagina bestaat uit 2 onderdelen. Links vindt u een navigatievenster om tussen de verschillende pagina's te navigeren. De meeste pagina's zijn om de gegevens van de database te beheren, maar ook om de profielen van de administratoren te beheren.

2.2.2 Het profielbeheer

Hier kan de Administrator 4 taken uitvoeren: Afmelden, klanten beheren, , de rol van bestaande klanten en het eigen profiel bewerken. In het navigatievenster zal dit in de groepen 'Accounts' en 'Users' op deze volgorde worden weergegeven:

- Afmelden;
- Klantbeheer (om klanten te beheren en hun rol te veranderen);
- Bewerk profiel.

Als de administrator op één van deze keuzes klikt, komt hij op de gewenste pagina terecht.

ClientManagerPage:

Op deze pagina vindt u bovenaan een zoekbalk rechts ervan om klanten een knop om klanten op te zoeken. Daarnaast is er ook een knop die u naar de AddUserPage leidt. Onder beide vindt u een tabel met een lijst van alle klanten alfabetisch gesorteerd op achternaam. De tabel bevat de volgende onderdelen:

- Achternaam;
- Voornaam;
- E-mail;
- Nationaliteit;
- Geboortedatum;
- Is Admin;
- Leeg vak;

Het lege vak bevat de knop om naar de EditUserPage te gaan en een knop om klanten te verwijderen.

EditUserPage

Je kan op deze pagina terecht komen via de knop in de ClientManagerPage (zoals hierboven beschreven staat) en via de tab 'Bewerk gebruiker' in het navigatievenster. Via deze pagina kan je alle gebruikersgegevens bewerken (zie het onderdeel RegisterPage van de [User Interface](#) in de App). De Administrator heeft ook de optie om aan te kruisen of de gebruiker een Administrator is. Het enige dat de Administrator niet kan veranderen is het wachtwoord van de gebruiker.

AddUserPage

Is hetzelfde als de EditUserPage, maar om klanten toe te voegen.

2.2.3 Het Databeheer

Hier kan de Administrator 3 taken uitvoeren: landen bewerken, landen toevoegen en verwijderen. In het navigatievenster zal dit in de groep 'Countries' via de tab 'Manage' worden weergegeven

CountryManagerPage

Hier krijgt de administrator net dezelfde pagina als ClientManagerPage te zien, maar met andere opties/info. De zoekbalk dient deze keer om een land op te zoeken en de knop, die u op de ClientManagerPage naar de AddUserPage zou brengen, leidt u nu naar de AddCountryPage. In de tabel wordt nu de info van de landen (alfabetisch gesorteerd op naam) weergegeven. Dit bestaat uit de volgende onderdelen:

- Naam;
- Hoofdstad;
- Lid van: (Het lidmaatschap van een Unie zoals bv. de EU);
- Aantal favorietenlijsten met het land;
- Aantal verlanglijsten met het land.
- Leeg Veld

Deze keer bevat het lege veld een knop om het land te verwijderen en een knop om je door te verwijzen naar de EditCountryPage.

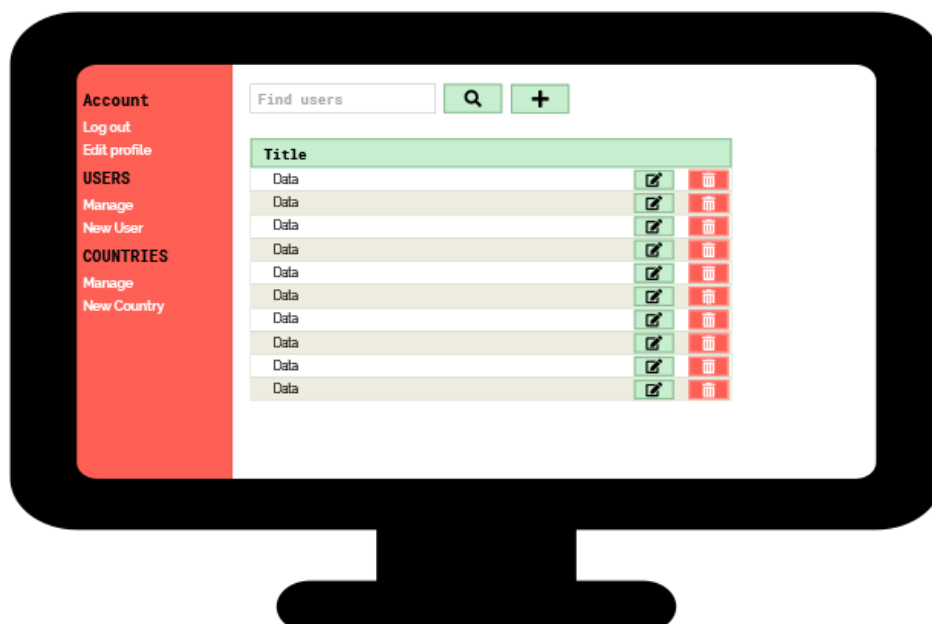
EditCountryPage

Op deze pagina kan de administrator alles bewerken over het land. Net zoals de EditUserPage kom je op deze pagina terecht via een knop op de CountryManagerPage.

AddCountryPage

Hier moet de Administrator via een formulier de informatie van het land ingeven. Die info bevat alles wat op de infopagina van het land terechtkomt.

Hieronder vindt u een overzicht met alle pagina's voor de website.



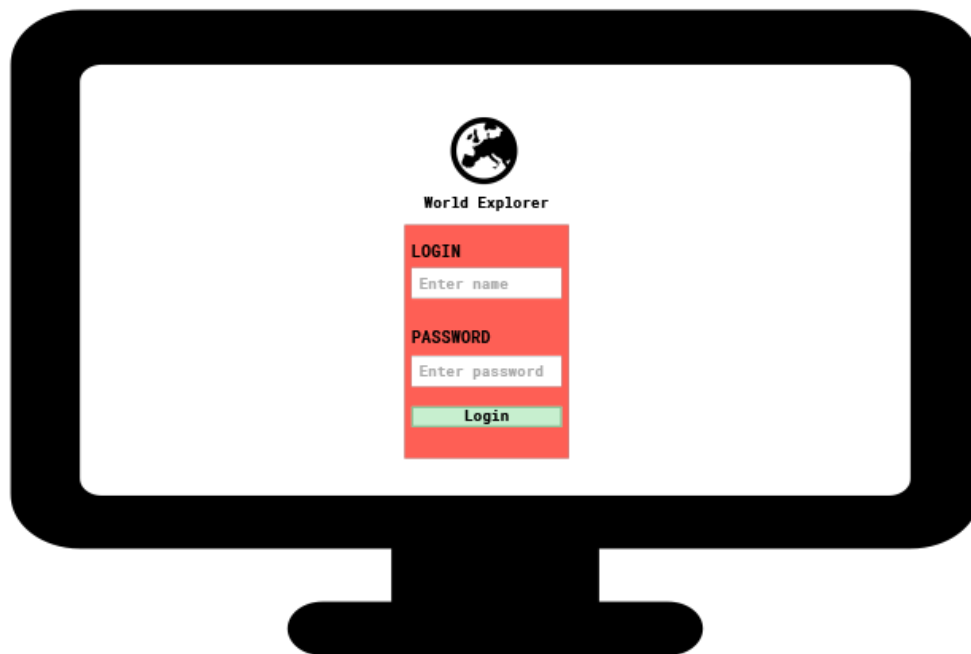
De EditPagina's van de website

The screenshot shows a web application interface on a monitor. On the left is a red sidebar with the following menu items: **Accounts & Info**, Log out, Edit profile, **USERS**, Manage, New User, **COUNTRIES**, Manage, and New Country. The main content area is white and contains a registration form. The form fields are: **Firstname:** (text input with placeholder 'Enter firstname'), **Lastname:** (text input with placeholder 'Enter lastname'), **E-mail:** (text input with placeholder 'Enter email'), **Birthdate:** (text input with placeholder '01/01/2020'), **Nationality:** (text input with placeholder 'Belgium'), **Password:** (text input with placeholder 'enter password'), **Repeat password** (text input with placeholder 'enter password again'), and **Is Admin** (checkbox). To the right of the form is a **Flag:** section with an 'Upload svg' button, a 'Choose' button, and a placeholder box labeled 'Flag'. A callout line points from the 'Flag' box to the text: 'Dit krijg je alleen op de AddCountryPage te zien.' At the bottom of the form are two buttons: 'Register' (green) and 'Cancel' (red).

De EditPage van de website

The screenshot shows a web application interface on a monitor. The red sidebar is identical to the previous one. The main content area is white and contains a details form. The form fields are: **Firstname:** (text input with placeholder 'Firstname'), **Lastname:** (text input with placeholder 'Enter lastname'), **E-mail:** (text input with placeholder 'Enter email'), **Birthdate:** (text input with placeholder '01/01/2020'), **Nationality** (text input with placeholder 'Belgium'), and **Is Admin** (checkbox). To the right is a **Flag:** section with a placeholder box labeled 'Flag'. A callout line points from the 'Flag' box to the text: 'Dit krijg je alleen op de EditCountryPage te zien.' At the bottom of the form are two buttons: 'Edit' (green) and 'Cancel' (red).

De detailsPage van de website



De LoginPage van de website

2.2.4 Extra Claim

Buiten is er ook een claim die gebaseerd is op basis van de populariteit per land. Die info wordt weergegeven in de CountryPopularityPage. Dit is net dezelfde pagina als iedere pagina met een tabel op deze website. De tabel bevat wel alleen het land, het aantal geregistreerde gebruikers per land en het aantal personen die het land aan hun wens- en favorietenlijst hebben toegevoegd. Ook is er een knop voorzien om de gegevens van het land te bewerken.

2.2.5 Vue.js

Voor de implementatie van Vue.js gaan eerst de gegevens gevalideert worden nog voor ze naar de server worden verstuurd. Daarmee bedoel ik dat alles waarvoor een formulier nodig is, eerst lokaal op de frontend wordt gevalideert en daarna op de backend via de server. Ook is het handig om hiermee tabellen op te stellen.

2.3 De API

Dit is de centrale plek waar alle code wordt uitgevoerd in verband met het ophalen van alle gegevens uit de database. Deze API gaat ook met andere API's communiceren om gegevens te verzamelen. Die API's zijn:

- De [restcountries API](#) (voor het verzamelen van landgegevens);
- De [spotify API](#) (voor de afspeellijst van het land te halen);
- De [Google maps API](#) (voor het vinden van het land waar de gebruiker zich bevindt);
- En de [Wikimedia API](#) (voor het verzamelen van een omschrijving per land).

Naast het verzamelen van de gegevens uit de database gaat hij ook nog via de Google maps API weergeven wat de naam van het land is waar de gebruiker zich bevindt en gaat hij deze gegevens terug naar de app of de website verzenden.

Daarnaast hebben ook andere ontwikkelaars toegang tot de API. Hiermee kunnen ze de gegevens vragen van het land met als extra properties: het aantal keren dat een land toegevoegd is aan een verlanglijst of lijst met favorieten, een lijst met de nummers van een spotify afspeellijst. Ze kunnen deze documentatie raadplegen via Swagger.

2.4 De gebruikte technologieën

Sommige technologieën werden al opgesomd, maar hieronder vindt u een volledige lijst:

De spotify/API-NET framework

Dit is een .net framework dat alle code om de Spotify api te kunnen aanspreken.

Info: <https://johnnycrazy.github.io/SpotifyAPI-NET/>

XamarinMediaManager

Dit is een plugin voor Xamarin die ervoor zorgt dat je maar één keer code moet schrijven om muziek op alle platformen af te spelen.

Info: <https://github.com/Baseflow/XamarinMediaManager>

Newtonsoft.json

Een nuget package voor .net dat het aanspreken van een json api vereenvoudigt.

FontAwesome

Om eenvoudige iconen op de website weer te geven zonder dat je afbeeldingen in een html image tag als afbeelding moet definiëren. De iconen zullen ook in Xamarin gebruikt worden, maar dan lokaal.

Info: <https://fontawesome.com/>

SyncFusion

Een UI Framework voor een professionele lay out.

Info: <https://www.syncfusion.com/>

Swagger

Een stukje software geïntegreerd in het project zelf waarmee ontwikkelaars hun api kunnen testen. Dit geeft een overzichtelijk resultaat weer.

<https://swagger.io/>

SignalR

Een websocket/library voor asp.net die onmiddellijk gegevens update zonder dat de gebruikers een pagina moeten refreshen. Dit ga ik gebruiker iedere keer aantallen te updaten bv. het aantal gebruikers dat één bepaald land toevoegen aan hun wenslijst. Het zal ook de lijst van de landen automatisch updaten. Zo kan de gebruiker geen fouten krijgen tijdens het gebruik van de app zoals bv. missende info van het land. Als dat niet goed genoeg is wil ik misschien een aparte chatruimte waar mijn gebruikers anoniem (enkel het land wordt zichtbaar) kunnen chatten.

<https://dotnet.microsoft.com/apps/aspnet/signalr>

2.5 Huisstijl

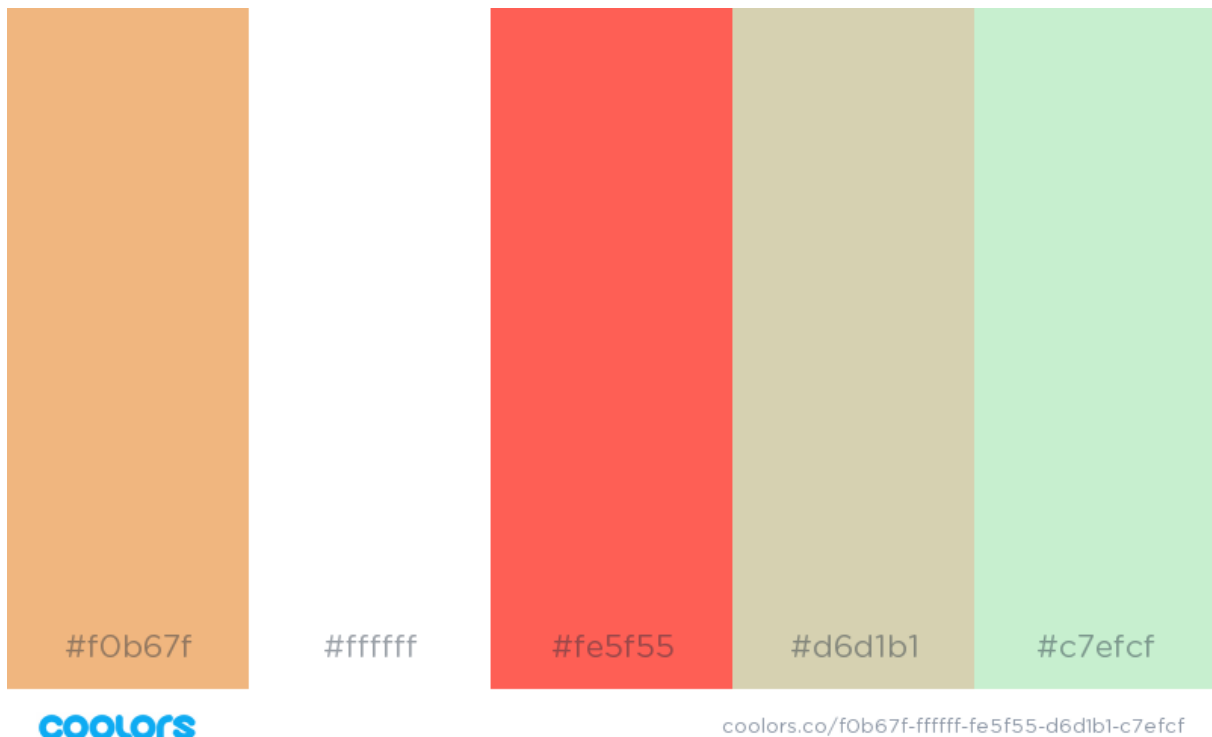
De lettertypes die ik in dit document gebruik zullen ook aanwezig zijn in de app en op de website zelf. Via de app gebeurt dit lokaal en via de website online via <https://fonts.google.com/>.

Die lettertypes zijn:

- Roboto Mono voor de titels;

- Raleway Light voor de tekst en subtitels(vetgedrukt).

Ik ga ook met een vast kleurenschema werken. Hieronder vind je een afbeelding met de bijhorende CMYK codes:



2.6 Gebruikt in dit document

- [Desktop Free Icon](#)
- [Arrow Right](#)
- [Map-marker-alt](#)
- [Star](#)
- [Spotify](#)
- [Heart](#)
- [User\(bewerkt\)](#)

Het logo en het icoon voor de login zijn zelf door mij met photoshop gemaakt. Hiervoor heb ik een map van Europa gebruikt, erover getekend en in een cirkel geplaatst. De gebruikersicoon heb ik zelf van fontawesome gehaald en in een cirkel geplaatst.