Rapport Project RetailSonar

20 Augustus 2018

Sander Thierens

(Andres Maes)

(Henri Sioen)

# 

# Overzicht API’s

## Web en Android

Voor de bespreking van de API’s voor de website en de android applicatie verwijzen we naar de Javadoc die u vindt op de volgende websites:

Website javadoc: <https://henrisioen.github.io/HenriSioenWebsite/> .

Android javadoc: <https://henrisioen.github.io/HenriSioenAndroid/> .

## Libraries

* Website:

Voor de website gebruikten we meerdere bibliotheken voor zowel layout als voor de weergave van visuele elementen zoals tabellen en knoppen.

* 1. Primefaces versie 6.2.3: Deze bibliotheek werd gebruikt voor de weergave van visuele elementen zoals tabellen en knoppen, maar ook confirmatie dialogs. De reden hiervoor zijnde dat deze bibliotheek html tags voorziet die reeds de nodige javascript en css bevatten zodat deze niet zelf moet geschreven worden.
  2. Blitzer: Dit is een layout bibliotheek dat een uitbreiding is op de Primefaces bibliotheek.
  3. Commons: Deze bibliotheek werd gebruikt voor de verwerking van de bestandsnamen van geüploade bestanden, onder meer voor de naam en de extensie van het bestand.
  4. Omnifaces: Gebruikt voor het weergeven van afbeeldingen uit de database die verwerkt en doorgegeven worden naar de website als bytestreams.
  5. Google maps API: voor het visualiseren van kaarten met locaties van één of meerdere filialen. Deze werd geïmplementeerd met behulp van Javascript.
  6. Java EE
* App & REST service:

Voor de app en de REST service werd er gebruik gemaakt van de library “Java JWT: JSON Web Token for Java and Android”. Deze library zorgt ervoor dat we JWT’s kunnen aanmaken, en kunnen lezen. Er waren wel een aantal aanpassingen nodig omdat niet elke functionaliteit werkte wanneer we de code in ons project importeerden.

# SWOT Analyse

## Website

1. Sterktes
   1. Categorieën kunnen gedefinieerd worden.
   2. Algemene eigenschappen die gedefinieerd kunnen worden met categorie.
   3. Bedrijfs specifieke eigenschappen die die gedefinieerd kunnen worden met categorie.
   4. Verschillende soorten eigenschappen die gedefinieerd kunnen worden (normaal, dropdown..)
   5. Takenlijst per filiaal.
   6. Onderscheid tussen regio- en expansiemanager.
   7. Winkels in de buurt kunnen weergegeven worden.
   8. Visuele representatie van de locatie van een of meerdere filialen
   9. Strakke layout
   10. Afstanden van percelen kan als eigenschap worden gekozen
   11. Oppervlaktes van percelen kan als eigenschap worden gekozen
   12. Custom eigenschappen kunnen worden ingegeven
   13. Mogelijkheid tot verwijderen van filialen zonder beînvloeden van andere filialen
   14. Bedrijfgegevens beschikbaar voor de managers
2. Zwaktes
   1. Het gebruik van meerdere gebruikers op hetzelfde moment is theoretisch gezien ondersteund, maar er kunnen problemen voorkomen indien meerdere admins tegelijkertijd zijn ingelogd en dezelfde data proberen aanpassen.
   2. Categorieën kunnen niet verwijderd worden.
   3. Bedrijven kunnen niet verwijderd worden.
   4. Afstanden en oppervlakten eigenschappen kunnen niet verwijderd worden
   5. Afstand tot filialen van concurrentie kan niet berekend worden
3. Kansen
   1. Concept toepassen voor andere ondernemingen.
   2. Uitbreiden van huidige website, meer functionaliteit toevoegen.
   3. Website beter beveiligen.
4. Bedreigingen
   1. Reeds bestaande gelijkaardige software.
   2. Concurrenten met gelijkaardig project.
   3. Geen https verbinding met de website.
   4. Concurrency problemen.

Tabel SWOT Analyse Website (enkel de belangrijkste elementen zijn opgenomen)

|  |  |
| --- | --- |
| Interne analyse | Externe analyse |
| Sterktes | Kansen |
| Definiëren van categorieën | Concept toepasbaar voor andere ondernemingen |
| Definiëren van eigenschappen | Uitbreiden van huidige functionaliteit |
| Takenlijst per filiaal | Website beter beveiligen |
| Winkels in de buurt |  |
| strakke layout |  |
| Gebruiksvriendelijker |  |
| Zwaktes | Bedreigingen |
| Geen concurrency ondersteund | Reed bestaande gelijkaardige software |
| Geen verwijdering van categoriën mogelijk | Concurrerende bedrijven |
| Geen Afstand tot concurrentie | Geen https verbinding met de website |

## App & REST service

1. Sterktes
   1. De REST service is (zoals in de naam) stateless, elke functie kan op elk moment door elke gebruiker worden opgeroepen waardoor meerdere gebruikers tegelijk aangemeld kunnen zijn.
   2. Men kan de taken zien die aan een filiaal gekoppeld zijn.
   3. De verbinding tussen de server en de app is geëncrypteerd met JWT’s.
   4. Taken worden gemarkeerd als ingevuld na het invullen van alle eigenschappen van het filiaal in kwestie.
   5. Afstand meten van percelen
   6. Oppervlakte meten van percelen
   7. Mogelijkheid tot indienen van de parameters in meerdere keren
   8. Opnieuw ingeven van bepaalde parameters mogelijk
2. Zwaktes
   1. Afbeeldingen gekoppeld aan filialen kunnen niet bekeken worden.
   2. De app kan geen afbeeldingen toevoegen aan een filiaal.
   3. taken worden slechts als “voltooid” gemarkeerd als een gebruiker alle eigenschappen in één keer ingevuld, niet als ze allemaal ingevuld zijn in de database.
3. Kansen
   1. Concept toepassen voor andere ondernemingen.
   2. Fotos toevoegen aan filialen.
   3. notities/opmerkingen toevoegen aan ingevulde eigenschappen.
4. Bedreigingen
   1. Reeds bestaande gelijkaardige software.
   2. Concurrenten met gelijkaardig project.

Tabel SWOT Analyse app & REST service (enkel de belangrijkste elementen zijn opgenomen)

|  |  |
| --- | --- |
| Interne analyse | Externe analyse |
| Sterktes | Kansen |
| Meerdere gebruikers tegelijk aangemeld | Concept toepasbaar op andere ondernemingen |
| Taken zien per filiaal | Foto’s toevoegen aan filialen |
| Simpele layout en opbouw van de app | Notities/opmerkingen toevoegen aan ingevulde eigenschappen |
| Veilige communicatie door JWT’s |  |
| Afstanden en Oppervlaktes meten |  |
| Zwaktes | Bedreigingen |
| Geen afbeeldingen bekijken of posten | Reeds bestaande gelijkaardige software |
|  | Concurrerende bedrijven |
|  |  |

# Reflectie en Uitbreidingen

## Web

Bij de ontwikkeling van de webpagina’s en de nodige beans om de input te verwerken waren er enkele uitdagingen die overwonnen dienden te worden vooraleer we verder konden. Zo was de login eerst niet via Glassfish geprogrammeerd waardoor de webpagina’s niet beveiligd waren tegen niet ingelogde gebruikers. Later werd dit aangepast, maar dit bracht enige vertraging met zich mee. Hierna was de eerste stap de flow van de webpagina naar de databank in orde brengen. Helaas duurde dit weer even. Dit zijn elementen die in een volgend project zeker sneller moeten kunnen verlopen om in de echte ontwikkelingsfase voor de functionaliteit meer te kunnen doen. We hebben uiteindelijk webpagina’s ontwikkeld voor drie gebruikers, de admin (RetailSonar), de expansiemanager en de regiomanager. Elk van deze gebruikers heeft een ander dashboard waar deze verschillende zaken kunnen zien/aanpassen. We hebben dus de gevraagde functionaliteit voor de mogelijke gebruikers succesvol geïmplementeerd. We gaan nu iets dieper in op elk van deze gebruikers en kijken wat er beter kon en waar uitgebreid kan worden.

1. Admin

De werknemers van Retailsonar zijn in staat om bedrijven toe te voegen, aan die

bedrijven kunnen dan gebruikers toegevoegd worden. Deze gebruikers kunnen zowel regio- als expansiemanagers zijn. De admin is ook in staat om andere admins toe te voegen. Om er voor te zorgen dat er steeds een admin user is, is er een root gebruiker die niet via de website verwijderd kan worden. De admin kan algemene eigenschappen toevoegen en die toekennen aan een bedrijf. Per bedrijf kunnen dan ook nog eens eigenschappen toegevoegd worden die bedrijfsspecifiek zijn. Beide types eigenschappen kunnen zowel standaard (text input required) als met een voorgedefinieerd dropdownmenu aangemaakt worden vanuit de web. Bij de ontwikkeling van de functionaliteit van deze eigenschappen was het even lastig om juist te zien hoe we per filiaal konden opslaan en hoe we ze precies aan een bedrijf konden toekennen. Een mogelijke uitbreiding is dan ook een betere, vooral visueel meer aantrekkelijke manier vinden om deze eigenschappen toe te kennen aan een bedrijf. Alsook andere opties voor het definiëren van eigenschappen.

1. Expansiemanager

De expansie manager is in staat om filialen toe te voegen en de eigenschappen die de regiomanager invulde te bekijken. Verder kan deze ook nieuwe regio- en expansiemanagers aanmaken. Hij kan ook een takenlijst aanvullen met taken en deze markeren als klaar/niet klaar alsook verwijderen indien nodig. Op een kaart worden de filialen weergegeven. Een uitbreiding kan zijn om op de locatie van een filiaal te kunnen klikken en zo het adres, toegekende regiomanager en navigatie naar het filiaal te kunnen zien.

1. Regiomanager

De regiomanager kan de toegekende eigenschappen invullen, afbeeldingen toevoegen en de winkels in de buurt bekijken op een kaart. Deze winkels worden niet opgeslagen in de databank door een technische barrière die helaas niet overwonnen werd. Meer specifiek het terugkoppelen van informatie die gegenereerd wordt in javascript naar de backing bean. Verder zou een uitbreiding ook kunnen zijn om indien men op de locatie van een filiaal klikt op de kaart (dashboard) de navigatie te zien vanuit de huidige locatie van de persoon.

Als globale uitbreiding kan er nog gewerkt worden aan ingebouwde agenda waarin vooraf ingepland kan worden wanneer welke manager welk filiaal gaat opbouwen. De zichtbaarheid van de afspraken kan ook verschillende zijn afhankelijk van de user (expansiemanager vs regiomanager).

## App & REST service - reflectie

De app en de REST service werden samen ontwikkeld dus bespreken we ze hier ook samen. De eerste fase van de app was al doende leren werken in android en de http communicatie tussen de server en de app. Dit was vrij simpel op een paar dingen na die een groot deel van de ontwikkelingstijd innamen.

Het eerste obstakel was werken met AsyncTasks, Dit zijn klasses die een deel van hun vorm mee hebben van het commando-ontwerppatroon: elke task wordt namelijk in zijn eigen klasse geïmplementeerd en opgeroepen met task.execute().

daarnaast was een ander obstakel, namelijk het programmeren van een recyclerview, dit was even een moeilijkheid maar na de eerste keer werd het vanzelfsprekend.

Het belangrijkste struikelpunt bij de ontwikkeling van de app was dat alles nieuw was, als we meer tijd hadden gestopt in het leren van de programmeer conventies voor android en over de opties die android aanbiedt (bv recyclerview) dan hadden we de app sneller kunnen ontwikkelen en er meer features aan hebben toegevoegd.

App & REST service - uitbreiding

Bij de uitbreiding zullen we zoals bij de reflectie enkel de app bespreken, dit omdat de app en de REST service één geheel vormen aangezien een feature in de REST service altijd ook geïmplementeerd moet worden in de app (anders heeft het geen nut dat het bestaat).

Volgend is een opsomming van alle features die de app nog zou kunnen hebben:

1. Agenda feature

In onze database is een “taak” iets dat bij een filiaal hoort, nu kan deze enkel een

beschrijving bevatten (een korte string van letters) we zouden er ook voor kunnen zorgen dat de expansiemanager aan elke taak een datum kan toevoegen (die dan de deadline is tegen wanneer het filiaal moet bezocht worden) verder kunnen we er dan in de android app voor zorgen dat deze datum automatisch toegevoegd wordt aan de agenda van de telefoon van de gebruiker.

1. Google maps

Een andere uitbreidingsmogelijkheid zou navigatie zijn. Indien de user dit wenst, zou deze op een knop kunnen drukken om dan doorverwezen te worden naar de google maps app waar de navigatie reeds ingevoerd is en de gebruiker direct kan beginnen met navigeren.

Vergelijking met reeds bestaande apps

1. Offline Surveys

Hoewel deze app vooral bedoeld is om enquêtes mee te houden, zou het ook mogelijk zijn om deze te gebruiken om de verschillende parameters in te geven die nodig zijn voor Retail Sonar. Bij het openen van de app valt meteen de eentonigheid er van op. De layout bevat enkel grauwe kleuren die niet meteen aanspreken, dit in tegenstelling tot onze app waarbij we gekozen hebben voor opvallende kleuren (de kleuren van het logo van Retail Sonar). De invoer van de parameters daarentegen is wel aangevuld door een passend symbool, waardoor de gebruiker bijna in één oogopslag kan zien over welke parameter het gaat. Bij deze app is er de mogelijkheid om in meerdere pagina’s de vragen te zetten, maar niet de mogelijkheid om terug te keren naar de vorig pagina om een aanpassing te doen. Wij verkozen alle invoer op één view te houden, omdat het aantal in te vullen parameters beperkt genoeg is om het overzichtelijk te houden.

<https://play.google.com/store/apps/details?id=com.askoura.offlinesurveys>

1. Globe observer (NASA) globe clouds

Met behulp van deze app geef je informatie aan NASA over de wolken toestand boven je locatie. We kunnen deze app dus zeer goed vergelijken met onze eigen app die informatie vraagt over een mogelijke locatie van een nieuw filiaal. Wat meteen opvalt is de duidelijkheid van de app, men werkt met zeer grote knoppen met verschillende kleuren. Deze eigenschap is een zeer groot pluspunt aangezien hun app altijd buiten zal worden gebruikt waar er kans is op fel zonlicht waardoor het scherm minder zichtbaar wordt. Hiermee hebben we echter geen rekening gehouden bij onze eigen app, omdat de testen altijd binnen gebeurden. Onderaan het scherm vinden we verscheidene knoppen met symbolen terug die, in tegenstelling tot hun invoer, niet meteen duidelijk weergeven wat de precieze werking ervan is. Bij onze eigen app maakten we de keuze om elke activity zo simpel mogelijk te houden zodat er geen verwarring kan ontstaan.

<https://play.google.com/store/apps/details?id=gov.nasa.globe.observer>

1. Magpi

Magpi geeft de gebruiker de mogelijkheid om via een webservice zelf een enquête aan te maken. Hierdoor zou RetailSonar deze app kunnen gebruiken om al hun benodigde gegevens te kunnen verzamelen. Net als bij onze eigen app start Magpi met een eenvoudige, maar duidelijke login pagina. Ook de lay out van de andere activities zijn zeer simplistisch gehouden om op een snelle manier de app te kunnen gebruiken. Dit vonden we ook bij onze eigen applicatie zeer belangrijk. We wouden het aantal knopen op het scherm minimaal houden zodat alles overzichtelijk zou blijven. Wanneer we bij Magpi dan vervolgens naar de vragenlijst willen gaan, gebeurt de overgang van het menu naar deze lijst met een animatie effect. Dit geeft een amateuristisch effect en is vrij storend wanneer men snel wil verwisselen van activity. Onze app bevat geen animatie overgangen omdat deze in een professionele omgeving gaat gebruikt worden. Een animatie hierbij zou zeker geen meerwaarde geven en eerder een storend element zijn. De vragenlijst van Magpi is net als onze app ook een doorlopende lijst en men kan gemakkelijk de eerder ingevulde vragen aanpassen. <https://play.google.com/store/apps/details?id=org.magpi.android>

1. GoFormz

GoFormz is een vergelijkbare app als Magpi. Door online een vragenlijst op te stellen kan men deze laten invullen via de applicatie. GoFormz gebruikt in plaats van een doorlopende lijst, een pdf-achtig invulformulier. Het grote voordeel hieraan is dat men op deze manier de vragen veel beter kan ordenen. Men kan namelijk zowel de vragen onder elkaar als naast elkaar plaatsen. Hierdoor is het zeer gemakkelijk om een specifieke vraag te vinden. Omdat men ook vragen naast elkaar plaatst moeten de lettertypes veel kleiner worden gezet en waardoor de leesbaarheid daalt . Dit hebben we niet met onze app die door middel van de vragen onder elkaar op te sommen heel leesbaar blijft. Omdat RetailSonar de app buiten zal gebruiken is de leesbaarheid dus ook zeer belangrijk. Goformz maakt ook gebruik van een navigatie drawer die op elk moment kan geopend worden door een knop in de linkerbovenhoek. Wij hebben de keuze gemaakt om geen drawer te gebruiken omdat we vonden dat er niet genoeg situatie bestonden waarbij deze zou worden gebruikt. Dit komt natuurlijk omdat onze app gebouwd is voor één specifiek doel, in tegenstelling tot GoFormz waarbij zowel het invullen van een formulier kan gebeuren, als het navigeren tussen verschillende andere vragenlijsten die men kan invullen.

<https://play.google.com/store/apps/details?id=com.goformz>

1. Symptomate Symptom Checker

Deze app geeft de gebruiker de mogelijkheid om aan de hand van enkele vragen over zijn ziekt symptomen de mogelijke diagnoses te weten te komen. Net als bij onze Retail Sonar applicatie moeten er een aantal vragen worden ingevuld die vervolgens worden doorgestuurd naar een server. Deze app werkt met een apart tabblad per vraag, hierdoor blijft het scherm eenvoudig van opmaak. Het nadeel is dat wanneer men een bepaalde vraag wil aanpassen men alle voorgaande vragen moet afgaan voor men aan de vraag komt die men wil aanpassen. We zien dus hier duidelijk dat men hier mikt op een zo eenvoudig mogelijke layout. Men gebruikt maar één andere kleur (blauw) dan de achtergrond (grijs), om de titelbar en checkboxes te kleuren. Op deze manier kan men duidelijk zien welke checkbox men heeft aangeduid. Ook wij hebben gekozen voor rood en grijs waardoor de leesbaarheid zo optimaal mogelijk is.

<https://play.google.com/store/apps/details?id=com.symptomate.mobile>