Statusverslag

(Features die wel/niet werden gerealiseerd)

Voor de back end :

* In de databank is een tabel ‘traveltimes’ geïmplementeerd geweest die niet gebruikt wordt. Deze tabel ging de ideale reistijden, berekend tijdens het lopen van de applicatie, moeten opslaan waardoor op een eenvoudige manier de reistijd van een traject vergeleken kan worden met de ideale reistijd. Een extra tabel hiervoor leek ons achteraf overbodig en er is gekozen geweest deze in de tabel van de routes op te slaan. Dit gebeurt niet meer onder de vorm van een ideale reistijd maar door een ‘speedlimit’ van het traject, die hetzelfde effect heeft (een ideale reistijd is uiteindelijk ongeveer hetzelfde als een traject aan de maximale snelheid afleggen). Naast vergelijken met de ideale snelheid van een traject is het nu ook mogelijk met de gemiddelde reistijd per dag te vergelijken, deze waarde bevindt zich in de trafficdata tabel.
* Dit was voordien niet gepland maar het is nu ook mogelijk eigenschappen van de applicatie te wijzigen terwijl de applicatie nog draait. Hiervoor zijn de properties bestanden (database.properties, app.properties, provider.properties) van de applicatie extern in een ‘config’ bestand geplaatst.
* De applicatie wordt afgesloten van 12u s’ nachts tot 6u s’ morgens, oorspronkelijk was het idee   
  om deze wel aan te laten s’ nachts maar met een groter pollinterval. Door onbetrouwbare data van de providers s’ nachts (te weinig weggebruikers) is er toch niet gekozen geweest voor deze optie.
* De REST API beveiliging is maar deels verwezenlijkt. Het plan voor de beveiliging was niet concreet vastgelegd maar de uiteindelijke beveiliging is ondermaats qua veiligheid, uitbreiding van dit onderdeel is dus een vereiste.

Features voor de front end :

* Het selecteren van meerdere providers in de menubalk van het dashboard om hen data te aggregeren (gemiddelde van hen data nemen) is niet verwezenlijkt geweest. Hier is bewust voor gekozen omdat je anders de data van een betere provider kan verzieken met data van een slechte provider.  
  Er is geopteerd geweest voor een alternatief waarbij je in de grafieken kan selecteren of je meerdere providers wil vergelijken met elkaar.
* De dag-functie van het dashboard ging origineel niet aanwezig zijn. Dit omdat deze data voor een bepaalde dag ook kan opgevraagd worden via de interval-functie maar om praktische redenen leek de dag-functie wel een handige tool.
* De zoekfunctie voor een route te zoeken op naam is ook een feature die maar tijdens het creatieproces van het dashboard is ontworpen. Uiteindelijk heeft dit een grote bijdrage tot de gebruiksvriendelijkheid van de applicatie.
* Pop-ups in de kaart zijn ook maar tijdens het verloop van sprint 2 ter sprake gekomen. Deze pop-ups maken het mogelijk nu ook data over een traject op te vragen door erop te klikken in de. Hierdoor wordt de keuze van hoe de applicatie gebruikt wordt overgelaten aan de gebruiker.
* De instellingen van het dashboard, zoals de opgeslagen intervallen en de kaart-coördinaten, worden nu meegegeven in een URL om collaboratie tussen verschillende gebruikers te vereenvoudigen (kopiëren en doorsturen van de URL). Dit is maar tijdens het ontwikkelingsproces van sprint 3 aan de orde gekomen.

Gekende problemen

* De pollthread maakte voor iedere route één thread aan, dus tot 30 threads, om de requests uit te voeren. Dit overmatige gebruik van threads zorgde voor een vertraagde werking van de applicatie waardoor nu één thread per provider, in totaal voorlopig 5 threads, wordt gebruikt.
* De query’s spelen een grote rol in de performantie van de applicatie. Deze moesten dan ook meerdere malen geoptimaliseerd worden voor toch een behoorlijke request tijd te verkrijgen bij query’s die veel worden uitgevoerd. Daarnaast leidt alle verwerking van data aan client-side (javascript) tot een vertraagd dashboard waardoor extra query’s en interfaces moesten toegevoegd worden.
* De productieomgeving was enorm traag en dit is deels opgelost geweest door geheugen te laten swappen bij het uitvoeren van de applicatie. Bij het invoeren van de nieuwe productieomgeving was het geheugenprobleem weg en werkte alles veel rapper.
* Glassfish zorgde voor enorm veel problemen. Sommige onverklaarbaar die opgelost werden met Glassfish te herstarten en andere die veroorzaakt waren door de recentste versie, Glassfish 4.1.1. Vervolgens leidde dit voordien vermelde gebrek aan geheugen ook nog eens tot het feit dat Glassfish enorm traag was. De REST API ontwerpen en uittesten was bijgevolg een moeizaam proces die meer tijd in beslag nam dan voorzien.
* De Coyote providerconnector, die geïmplementeerd is in Perl, zorgde voor heel wat problemen. Enkele voorbeelden hiervan zijn het parsen van de JSON die opgevraagd werd bij Coyote, de cookie voor de authenticatie bij hen API die niet werd onthouden en de interactie tussen Java en JSON tot stand brengen.
* In de databank was per ongeluk bij de waypoints de tabelnaam latitude en longitude omgewisseld. De routes werden daardoor getekend in de hoorn van Afrika, het heeft heel wat tijd gekost voor we dit hadden ontdekt.
* TomTom heeft een zeer beperkt aantal requests per dag. Dit probleem hebben we opgelost door 8 API-KEYS toe te voegen aan de applicatie en dus nu 8 accounts tegelijkertijd requests te laten uitvoeren.
* Op de kaart van het dashboard kwam boven water dat de waypoints niet correct samenvallen met de routes. Handmatige aanpassing van de waypoints voor een mooie kaartstructuur zorgde dus voor een lastig en tijdrovend proces.
* Bij het ontwerp van de tekstuele data en grafieken van het dashboard traden ook heel wat problemen op.

Verbeterpunten

* De fouten die door de REST API worden opgeworpen worden deels verwerkt in de REST API zelf, bijvoorbeeld bij gebrek aan een verplichte parameter, en deels door de ExceptionMapper. Er zou minder duplicate code optreden en de code zou in het algemeen meer leesbaar zijn indien alle opgeworpen fouten worden verwerkt in ExceptionMapper’s.
* De documentatie van de code is soms in het Nederlands en soms in het Engels, eenheid hierin creëren geeft een professioneler beeld van de applicatie.
* De TomTom provider die 8 API-KEYS nodig heeft is geen correcte oplossing voor het probleem. Een alternatief aanvragen aan TomTom, bijvoorbeeld uitbreiding van het aangeboden pakket, is waarschijnlijk een eenvoudigere oplossing.

Uitbreidingsmogelijkheden

* Doordat de properties bestanden nu buiten de applicatie staan, wordt het mogelijk om een instellingen optie te implementeren in het dashboard waarbij je bijvoorbeeld een provider kan toevoegen, het pollinterval kan aanpassen en eigenschappen van een provider kan veranderen.
* Qua beveiliging van de applicatie, het dashboard eventueel meegerekend, is het mogelijk login en register-scherm te implementeren die gebaseerd is op het API-KEY systeem die reeds aanwezig is. Een ingelogde gebruiker zou vervolgens dan een aanvraag voor een API-KEY kunnen indienen. Wat wel aanwezig is, is het controleren van iedere API request en kijken of een API-KEY header aanwezig en geldig is. Deze API-KEY hard gecodeerd in de applicatie.
* De REST API is volledig gedocumenteerd in het projectdossier maar deze is niet aanwezig in de applicatie zelf, wat het gebruik ervan vermoeilijkt voor derden die de API ook willen gebruiken. Deze in een overzichtelijke interface aanbieden is een uitstekende uitbreidingsmogelijkheid.
* Naast een login-scherm zou ook een pagina kunnen voorzien worden die de verschillende functionaliteiten aanbiedt van de applicatie (zoals de instellingen aanpassen, een interval toevoegen, de REST API documentatie, …). Dit zorgt dat de klant meer geleid wordt in het gebruik van de applicatie wat gebruikersvriendelijker is. Nadeel is echter dat er ook meer klikwerk wordt veroorzaakt.
* Opmerkelijke vertragingen (die wellicht veroorzaakt zijn door een ongeval of dergelijke) worden reeds voorgesteld door een traject rood te kleuren. Deze vertragingen worden berekend door reistijden te vergelijken met de gewoonlijke reistijd van het traject. Om de opvolging en de verwerking van deze gegevens te vereenvoudigen zou een oorzaak meegeven een goede aanvulling van de applicatie zijn. Deze oorzaken kunnen opgehaald worden bij enkele van de providers.