



Schakeljaar Master of Science in de
industriële wetenschappen: informatica
Academiejaar 2015-2016

Projectdossier VOP: Verkeerscentrum

In opdracht van mobiliteitscenter Gent

Ingediend op 17 mei 2016

Studenten verkeer 4

Aaron Mousavi
Dwight Kerkhove
Niels Verbeeck
Thomas Clauwaert
Tomas Bolckmans

Professoren

H. Naessens
V. Ongenae
P. Maenhaut

Inhoudsopgave

1	Inleiding	5
2	Statusverslag	6
3	Tekortkomingen	6
4	Taakverdeling	7
5	Analyse: Use cases	9
5.1	Globale domeinregels	9
5.2	Verzamel reistijdgegevens	10
5.3	Bekijk routeoverzicht	11
5.4	Bekijk routedetail	12
5.5	Bekijk routemap	13
5.6	Vergelijk providerdata	14
5.7	Wijzig route	15
5.8	Bekijk logpagina	16
5.9	Bekijk dashboard	17
5.10	Aanbieden routegegevens met reistijden	18
5.11	Verzamelen POI-gegevens	19
5.12	Verzamelen weergegevens	20
5.13	Vergelijk routes	21
5.14	Aanpassingen use cases na sprint 1	22
5.15	Look & Feel Requirements	22
5.16	Usability & Humanity Requirements	23
5.17	Operationele & omgevingsrequirements	23
5.18	Wettelijke requirements	23
6	Analyse: Mockups	24
6.1	Initiële Mockup Dashboard	24
6.2	Initiële Mockup Logging	24

7	Analyse: Diagrammen	25
7.1	Use case Diagram (Final)	25
7.2	Use case Diagram - Einde Sprint 1	25
7.3	Use case Diagram - Einde Sprint 2	26
7.4	Klassendiagram - Projectstructuur	27
7.5	ERD	28
7.6	Sequentiediagram - Bepalen File en Oorzaak	29
7.7	Sequentiediagram - Polling verkeersgegevens	30
7.8	Sequentiediagram - Overview Routes	31
8	Feedback	32
8.1	Vragen Team aan Klant	32
8.2	Vragen Klant aan Team	33
8.3	Afspraken Team	33
9	Kwaliteitscontrole	34
9.1	Overzicht Testplan	34
9.2	Unit Tests	34
9.3	Integration Tests (via Selenium)	34
9.4	Usability Tests (Use case testing)	35
9.5	Usability Tests (scenario)	37
9.5.1	Template	38
9.5.2	Persoon 1: IT-deskundige	39
9.5.3	Persoon 2: niet-IT-deskundige (1)	40
9.5.4	Persoon 3: niet-IT-deskundige (2)	41
9.6	Load Tests	42
10	Technische handleiding	43
10.1	GUI layer	43
10.2	Domain layer	43
10.2.1	Domain	43
10.2.2	Domain Objects	44
10.3	DAL layer	45
10.4	Poll service	45

11 Gebruikershandleiding	47
11.1 Inleiding	47
11.2 Het Dashboard	47
11.3 Overview	48
11.3.1 Summary	48
11.3.2 Detail per traject	49
11.3.3 Edit	51
11.3.4 For each provider	51
11.3.5 By Route	52
11.4 Map View	52
11.5 Logs	53
11.6 Compare Route	54
11.7 Compare Routes	55
12 Installatiehandleiding	57
12.1 Software installeren (via install.sh)	57
12.1.1 Perl installeren	57
12.1.2 Mysql database verkeer in orde brengen	58
12.1.3 Poll service in een scriptje	58
12.1.4 Instellen backup	58
12.2 Software manueel installeren	59
12.2.1 Datum/uur/tijdzone instellen	59
12.2.2 Java 8 installeren	59
12.2.3 Mariadb installeren	59
12.2.4 Extra tools installeren	60
12.2.5 Glassfish installeren	60
12.2.6 Perl installeren	60
12.2.7 Mysql database verkeer in orde brengen	60
12.2.8 Scrapers testen	60
12.2.9 Poll service in een scriptje	60
12.2.10 Instellen backup	61
12.2.11 Uploaden Verkeerweb	61

12.3 Documentatie Scrapers	63
13 Productbacklog	64
13.1 Productbacklog - Einde sprint 1	64
13.2 Productbacklog - Einde sprint 2	65
13.3 Productbacklog - Einde sprint 3	66

1 Inleiding

Het beleid omtrent de mobiliteit in en rond stad Gent is de laatste jaren flink veranderd: doorgaand autoverkeer wordt uit de stad geweerd, fietsers krijgen meer plaats in het centrum... Daarom is er sinds 2014 een nieuw regionaal verkeerscentrum opgericht voor alles wat met mobiliteit te maken heeft in en rond de stad. Bij nieuwe mobiliteitsplannen moet er steeds uitgebreid getest worden wat voor effect deze op de reistijden hebben. Daarom hebben vier teams van Universiteit Gent in opdracht van het Mobiliteitsbedrijf stad Gent een webapplicatie gemaakt die aan de hand van verschillende dataproviders (Google, HereMaps, Waze, TomTom, BingMaps en BeMobile) voor belangrijke routes de reistijden berekend.

Dit team (verkeer 4 genaamd) heeft voornamelijk de nadruk gelegd op het implementeren van zoveel mogelijk dataproviders, de vergelijking van deze data tussen de verschillende providers, het vergelijken van de reistijden bij verschillende routes, gelijke routes op verschillende tijdstippen, het detecteren van verkeersopstoppen en een oorzaak (slecht weer, accidenten, wegenwerken, ...) zoeken bij elke verkeersopstopping. In het onderstaande dossier kan u in meer detail lezen hoe wij deze ideeën hebben uitgewerkt.

2 Statusverslag

In sprint 1 werden de essentiële zaken reeds behandeld. We hebben ons gefocussed op het ophalen van de data van de verschillende providers en vervolgens op het zo goed mogelijk weergeven van deze data in de vorm van een webapplicatie. Vervolgens zijn hebben we besloten om ook “Points-of-interest” (POI’s) bij te houden. Deze plaatsen vertellen ons waar er onregelmatigheden zijn op of rondom de verschillende trajecten. Door uitschieters op de reistijden te detecteren hebben we periodes van files kunnen achterhalen. De oorzaken van deze files hebben we proberen achterhalen via de verzamelde POI’s en weergegevens. In de laatste stappen van het project hebben we geprobeerd om nog meer interessante gegevens visueel weer te geven. Zo kan je verschillende trajecten vergelijken op één bepaald tijdstip of één specifiek traject gelijken op verschillende tijdstippen.

Er zijn echter een aantal zaken die ook niet gerealiseerd zijn. Een voorbeeld hiervan is het ophalen van alle data op 1 specifiek moment en dit visueel tonen op de kaart. Een andere optie die we lieten vallen is het nog meer (of beter) kunnen filteren op pagina’s waar we trajecten vergelijken. We hebben ook nog onderzocht of we aan de hand van uitschieters en POI’s meldingen konden genereren die via twitter of andere social media konden verspreid worden maar hier hebben we niet genoeg tijd meer gehad. Tenslotte was het ook nog de bedoeling om alles behalve de database configuratie in een aparte settings pagina te kunnen beheren, aangezien dit gebruiksvriendelijker is dan het application.conf aan te passen en de war nadien opnieuw te deployen maar ook hier zijn we niet meer toe gekomen.

3 Tekortkomingen

1. Poll service: Er zit een bug in de poll service waarbij na verloop van tijd de thread die het pollen uitvoert stopt terwijl de andere threads om vb POI’s te matchen of weather gegevens op te vragen wel verder blijven lopen. Het gevolg is dat er geen gegevens meer worden opgeslagen tot de service herstart wordt. In eerste plek dachten we aan een memory leak die na verloop van tijd een OutOfMemory exception geeft en daardoor de thread afbreekt, hier hebben we verschillende aanpassingen voor gedaan om dit te vermijden maar dit heeft het probleem niet opgelost. Via de Netbeans profiler hebben we één van de oorzaken (Retrofit) gevonden en vervangen door een standaard URLConnection van java om REST gegeven op te vragen. We hebben ook de service in een bash script gestoken die ervoor zorgt dat het proces elke nacht gekilled en heropgestart wordt om de buildup van memory te vermijden maar we hebben gemerkt dat soms na een uur na de herstart hetzelfde probleem opdook. We vermoedden dat één van de providers errors gaf en zo alles plat trok maar zelfs na overal de exceptions op te vangen blijft dit probleem voorkomen. Aangezien dit ook maar sporadisch voorkomt is dit zeer lastig te debuggen. De logs die we opslagen gaf ook geen indicatie tot de oorzaak van het probleem.

Helaas hebben we geen tijd meer gehad om hier verder op te zoeken. Mogelijks kan het probleem opgelost worden door de polling volledig sequentieel uit te voeren in plaats van per route in parallel met een ExecutorService.

2. Admin login: We hebben ervoor gekozen om geen user beheer in ons applicatie te voorzien en ons toe te spitsen op het verzamelen en interpreteren van gegevens. Momenteel is het niet mogelijk om een deel van de website publiek te stellen en de administrator acties af te schermen. Wel kan de algemene login voor de website beheerd worden via de glassfish admin console.
3. Unit tests: Het was de bedoeling om alle services via dependency injection op te bouwen, hierdoor konden we dan gemakkelijk een mock service layer maken in de domain laag om vaste test data terug te geven waarop we unit tests konden uitvoeren. Helaas is door tijdsgebrek van de mocking niets in huis gekomen, waardoor we ook geen unit tests hebben kunnen schrijven. We hebben ons dan toegespitst op andere tests zoals o.a. load tests.
4. Routes wijzigen: Routes in ons systeem kunnen enkel gewijzigd worden. Verwijderen of toevoegen kan enkel via de databank gebeuren. Verder worden de waypoints van de route ook bepaald via

de TomTom API, waarvoor een geldige API key nodig is. Het wijzigen van de route is iets dat we voorzien hebben om gemakkelijker routes te kunnen corrigeren maar is eigenlijk niet echt bruikbaar voor eindgebruikers.

5. User interface: Sommige UI elementen zijn niet altijd optimaal in gebruiksgemak. Niet alle tabelhoofdingen scrollen mee met de pagina en ook al hebben we bootstrap gebruikt als framework, sommige elementen werken niet altijd even goed op kleinere schermen zoals tablets of gsms. Hierdoor is het aangeraden om de web applicatie te gebruiken op een scherm met hoge resolutie.

4 Taakverdeling

In de tabellen hieronder staan vooral de nieuwe features per sprint. De triviale dingen worden niet vermeld.

Omschrijving	UC / algemeen	Tomas	Thomas	Aaron	Dwight	Niels
Providers implementeren	Verzamel reistijdgegevens	x			x	x
Trajectoverzichtpagina	Bekijk trajectoverzicht		x		x	
Trajectdetailpagina	Bekijk trajectdetail			x		
Use cases	Analyse		x			
Polling service	Verzamel reistijdgegevens				x	
Trajecten weergeven op kaart	Algemeen				x	
Traject wijzigen	Traject wijzigen				x	
Verbeteringen, testen, documentatie en bug fixes		x	x	x	x	x
Projectdossier	Analyse		x	x		
Presentatie	Algemeen		x		x	
Repository opzetten	Algemeen			x		
Project aanmaken	Algemeen				x	
Wiki aanvullen	Algemeen		x	x	x	

Tabel 1: Taakverdeling sprint 1

Omschrijving	UC / algemeen	Tomas	Thomas	Aaron	Dwight	Niels
Logging	Bekijk logpagina	x				
POI integreren	Verzamel POI-gegevens				x	
Weer integreren	Verzamel Weergegevens					x
Trajectdetail	Bekijk trajectdetail			x		
Routes vergelijken	Routes vergelijken			x		
Vertalen naar Engels	Algemeen	x	x	x	x	x
Dashboard	Bekijk dashboard	x				
Diagrammen bijwerken	Analyse	x	x	x	x	
Testing (usability)	Testen		x			x
Testing (smoke en load)	Testen		x			
File detectie	Filen detecteren				x	
Verbeteringen, testen, documentatie en bug fixes		x	x	x	x	x
Projectdossier	Analyse	x	x	x	x	x
Presentatie	Algemeen	x		x		
Wiki aanvullen	Algemeen	x	x	x	x	x

Tabel 2: Taakverdeling sprint 2

Omschrijving	UC / algemeen	Tomas	Thomas	Aaron	Dwight	Niels
Vergelijk route	Route vergelijken			x		
Logging afwerken	Bekijk logpagina	x				
REST service	Routegegevens met reistijden					x
Verbeteringen, testen, documentatie en bug fixes		x	x	x	x	x
Projectdossier	Analyse	x	x	x	x	x
Presentatie	Algemeen	x	x			

Tabel 3: Taakverdeling sprint 3

5 Analyse: Use cases

5.1 Globale domeinregels

Wanneer er sprake is van “*providers*” dan wordt hiermee de verzameling van geïmplementeerde providers bedoeld. Deze bestaan momenteel uit TomTom, Coyote, HereMaps, Google Maps, Bing Maps, Be-Mobile, ViaMichelin en Waze. Indien er een providerafhankelijke beslissing werd genomen dan zal dit gespecificeerd worden.

Een operator is altijd een gebruiker. De verschillen zijn weggewerkt en in gebruik van de applicatie zijn beide dezelfde. Uit analytisch standpunt is echter een operator degene die wijzigingen kan aanbrengen aan het systeem (zoals bv. het wijzigen van een route), terwijl een gebruiker enkel observeert (en data afhaalt).

DR Route-informatie: Een route heeft een naam, een afstand, een normale reistijd, een huidige reistijd en bijgevolg een vertraging. Deze laatste 3 kunnen afwijken per provider.

DR Providers: Een provider heeft een naam en providerspecifieke eigenschappen.

DR Detail Filters: Op de detailpagina kunnen volgende filters ingesteld worden: startdatum (en tijdstip), einddatum (en tijdstip).

DR Vergelijk Filters: Op de vergelijkroutes-pagina kunnen volgende filters ingesteld worden: twee routes, een startdatum (en tijdstip), een einddatum (en tijdstip) en providers.

DR Dashboards: Op de homepage bevinden zich zogenaamde “dashboards”. Dit zijn de verschillende “panels” of aspecten op deze pagina. Momenteel is dit een samenvatting van de logs, een overzicht van de POI's, een minikaart met de huidige status, het weer en een overzicht van de laatste tweets van VerkeerGentB.

5.2 Verzamel reistijdgegevens

Use Case	Verzamel reistijdgegevens
Primaire actor	Polling Service
Stakeholders	
Preconditie	Er zijn minimaal 5 minuten verstreken sinds de laatste aanvraag van reisdata gegevens. Er zijn routes beschikbaar in de database.
Postconditie	De reistijden van elke route zijn voor elke provider geüpdatet.
Normaal verloop	
1.	De polling service wenst recente reistijd gegevens op te halen.
2.	Het systeem haalt alle beschikbare routes op.
3.	Het systeem scrapet de reistijden en vertragingen van hun respectievelijke websites.
5.	De polling service verwerkt de verzamelde gegevens en slaat deze op in de databank.
6.	De polling service vraagt (parallel) aan alle per-route providers de recentste reistijden per route.
7.	Het systeem scrapet de reistijden en vertragingen van hun respectievelijke website.
8.	De polling service wacht tot alle resultaten bekend zijn, verwerkt de gegevens en slaat deze op in de databank.
Alternatief verloop	
	Voor TomTom, Google en Here maps:
6A.	De reistijd gegevens kunnen niet opgevraagd worden via scraping.
6A1.	De provider vraagt de reistijd gegevens op aan de API van de provider met de opgeslagen API keys
6A2.	Ga naar stap 7
Domeinspecifieke regels	
Op te klaren punten	

Tabel 4: UC: Verzamel reistijdgegevens

5.3 Bekijk routeoverzicht

Use Case	Bekijk routeoverzicht
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	Routes zitten in de database.
Postconditie	Een overzicht van de routes wordt getoond.
Normaal verloop	
1.	De gebruiker wenst het routeoverzicht te bekijken.
2.	Het systeem haalt de routes op en geeft een overzicht.
Alternatief verloop	
2A.	Het systeem kan de routes niet op halen.
2A1.	Het systeem toont een correcte boodschap.
2A2.	Postconditie wordt niet bereikt.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 5: UC: Bekijk routeoverzicht

5.4 Bekijk routedetail

Use Case	Bekijk routedetail
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	Er bestaat een route waarvan men de details kan opvragen. Gebruiker bevindt zich op route-overzichtpagina
Postconditie	De trajectdetails zijn beschikbaar.
Normaal verloop	
1.	De gebruiker wenst de details van een route op te vragen.
2.	Het systeem geeft een lijst van routes.
3.	De gebruiker kiest een route.
4.	Het systeem toont de routedetails.
Alternatief verloop	
4A.	Het systeem kan de routedetails niet tonen.
4A1.	Het systeem geeft een correcte boodschap.
4A2.	Terug naar stap 3.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 6: UC: Bekijk routedetail

5.5 Bekijk routemap

Use Case	Bekijk routemap
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	De data van de routemap is beschikbaar.
Postconditie	De routemap wordt getoond waarop de routes worden weergegeven op een kaart met indicatie van hun reistijd.
Normaal verloop	
1.	De gebruiker wenst de routemap op te vragen.
2.	Het systeem haalt de recentste gegevens op om de routemap te vormen.
3.	Het systeem toont de routemap.
Alternatief verloop	
3A.	Het systeem kan de routemap niet tonen.
3A1.	Het systeem geeft een correcte boodschap.
3A2.	Terug naar stap 1.
Domeinspecifieke regels	
Op te klaren punten	De filters worden pas achteraf toegepast. Daarom zit dit niet in deze use case.

Tabel 7: UC: Bekijk routemap

5.6 Vergelijk providerdata

Use Case	Vergelijk providerdata
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	Er is providerdata aanwezig. Gebruiker bevindt zich op een routedetailpagina.
Postconditie	Een overzicht om providerdata te vergelijken wordt getoond.
Normaal verloop	
1.	De gebruiker wenst providerdata te vergelijken.
2.	Het systeem geeft een lijst met providers waartussen men kan vergelijken.
3.	De gebruiker kiest welke providers en welke filters hij wil vergelijken (DR Providers en DR Detail Filters).
4.	Het systeem toont een overzicht met de toegepaste filters op de gekozen providers.
Alternatief verloop	
4A.	Het systeem kan het overzicht met toegepaste filters en providers niet weergeven.
4A1.	Het systeem geeft een correcte boodschap.
4A2.	Terug naar stap 3.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 8: UC: Vergelijk providerdata

5.7 Wijzig route

Use Case	Wijzig route
Primaire actor	Operator
Stakeholders	
Preconditie	Er bestaat een route om te wijzigen.
Postconditie	Het systeem heeft een route gewijzigd.
Normaal verloop	
1.	De operator wenst een route te wijzigen.
2.	Het systeem geeft een overzicht van alle bestaande routes.
3.	De operator kiest een route.
4.	Het systeem vraagt de nodige informatie voor het wijzigen van een route.
5.	De operator geeft de nodige informatie (DR Routeinformatie).
6.	Het systeem valideert.
7.	Het systeem wijzigt de route.
Alternatief verloop	
6A.	De gegevens zijn incorrect.
6A1.	Het systeem geeft een correcte melding.
6A2.	Terug naar stap 5.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 9: UC: Wijzig route

5.8 Bekijk logpagina

Use Case	Bekijk logpagina
Primaire actor	Operator
Stakeholders	
Preconditie	/
Postconditie	De logpagina wordt weergegeven.
Normaal verloop	
1.	De operator wenst de logpagina te bekijken.
2.	Het systeem haalt logbestanden op.
3.	Het systeem toont de logpagina.
Alternatief verloop	
2A.	Het systeem kon de logbestanden niet ophalen.
2A1.	Het systeem geeft een correcte melding.
2A2.	De postconditie is bereikt.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 10: UC: Bekijk logpagina

5.9 Bekijk dashboard

Use Case	Bekijk dashboard
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	Dashboardpagina moet beschikbaar zijn.
Postconditie	De dashboardpagina wordt weergegeven.
Normaal verloop	
1.	De gebruiker wenst het dashboardpagina te bekijken.
2.	Het systeem toont de dashboardpagina met correcte dashboards.
Alternatief verloop	
2A.	Het systeem kan niet alle informatie van de dashboards ophalen.
2A1.	Het systeem geeft een correcte melding.
2A2.	Het systeem toont de dashboardpagina zonder correcte dashboards. Postconditie is bereikt.
Domeinspecifieke regels	(DR: Dashboards)
Op te klaren punten	

Tabel 11: UC: Bekijk dashboard

5.10 Aanbieden routegegevens met reistijden

Use Case	Aanbieden routegegevens met reistijden
Primaire actor	Gebruiker
Stakeholders	Operator
Preconditie	De trajectgegevens zijn beschikbaar.
Postconditie	De trajectgegevens zijn correct verkregen.
Normaal verloop	
1.	De gebruiker wenst routegegevens op te halen.
2.	Het systeem zorgt voor ophaalbare gegevens.
3.	De gebruiker maakt een specifieke request.
4.	Het systeem voldoet aan de request en geeft correcte routegegevens terug.
Alternatief verloop	
4A.	Het systeem kan niet aan de request voldoen.
4A1.	Het systeem toont een correcte melding.
4A2.	Terug naar stap 3.
Domeinspecifieke regels	Afhankelijk per provider.
Op te klaren punten	(DR waze)

Tabel 12: UC: Aanbieden routegegevens met reistijden

5.11 Verzamelen POI-gegevens

Use Case	Verzamelen POI-gegevens
Primaire actor	Polling Service
Stakeholders	
Preconditie	De POI gegevens moeten weggeschreven kunnen worden naar de databank.
Postconditie	De POI gegevens zijn weggeschreven naar de database.
Normaal verloop	
1.	De polling service wenst POI gegevens op te slaan.
2.	De polling service haalt POI gegevens op.
3.	Het systeem slaat de opgehaalde POI gegevens op.
4.	De POI gegevens zijn weggeschreven naar de database.
Alternatief verloop	
2A.	De polling service kan de POI gegevens niet ophalen.
2A1.	Het systeem toont een correcte melding.
2A2.	(Terug naar stap 1)(Postconditie niet bereikt).
3A.	Het systeem kan de opgehaalde POI gegevens niet wegschrijven naar de databank.
3A1.	Het systeem toont een correcte melding.
3A2.	(Terug naar stap 2)(Postconditie niet bereikt).
Domeinspecifieke regels	
Op te klaren punten	Bij falen opnieuw proberen of stoppen?

Tabel 13: UC: Verzamelen POI-gegevens

5.12 Verzamelen weergegevens

Use Case	Verzamelen weergegevens
Primaire actor	Polling Service
Stakeholders	
Preconditie	De weergegevens moeten weggeschreven kunnen worden naar de databank.
Postconditie	De weergegevens zijn weggeschreven naar de database.
Normaal verloop	
1.	De polling service wenst weergegevens op te slaan.
2.	De polling service haalt weergegevens op.
3.	Het systeem slaat de opgehaalde weergegevens op.
4.	De weergegevens zijn weggeschreven naar de database.
Alternatief verloop	
2A.	De polling service kan de weergegevens niet ophalen.
2A1.	Het systeem toont een correcte melding.
2A2.	(Terug naar stap 1)(Postconditie niet bereikt).
3A.	Het systeem kan de opgehaalde weergegevens niet wegschrijven naar de databank.
3A1.	Het systeem toont een correcte melding.
3A2.	(Terug naar stap 2)(Postconditie niet bereikt).
Domeinspecifieke regels	
Op te klaren punten	Bij falen opnieuw proberen of stoppen?

Tabel 14: UC: Verzamelen weergegevens

5.13 Vergelijk routes

Use Case	Vergelijk routes
Primaire actor	Gebruiker
Stakeholders	
Preconditie	Vergelijk Routes moet beschikbaar zijn.
Postconditie	Twee routes kunnen met elkaar vergeleken worden in een grafiek.
Normaal verloop	
1.	De gebruiker wenst het twee routes te vergelijken.
2.	Het systeem toont de vergelijk-routepagina met correcte filter-opties. (DR: Filteropties vergelijk route)
3.	De gebruiker stelt alle filteropties naar behoren in.
4.	Het systeem toont een grafiek waarin de twee routes vergeleken worden.
Alternatief verloop	
3A.	Het systeem kan de filteropties niet correct toepassen op de op te halen data.
3A1.	Het systeem geeft een correcte melding.
3A2.	Het systeem toont de vergelijk-routepagina opnieuw en de use case wordt afgesloten zonder het bereiken van de postconditie.
Domeinspecifieke regels	
Op te klaren punten	

Tabel 15: UC: Vergelijk routes

5.14 Aanpassingen use cases na sprint 1

Na feedback bij sprint 1 werd besloten om volgende use cases te verwijderen:

- “Verwijder route”: kan momenteel enkel door rechtstreeks de databank te wijzigen.
- “Nieuwe route toevoegen”: kan momenteel enkel door rechtstreeks de databank te wijzigen.
- “Infopagina’s bekijken”: het project blijft intern dus er is geen nood aan.
- “Parkeer- en bordgegevens verzamelen”: valt buiten de opdracht en is minder relevant.

Opmerking: Het wijzigen van een route is mogelijk via een grafische interface. Het verwijderen van routes kan met behulp van een SQL-query in de databank. Het toevoegen van routes kan eveneens met een dergelijke SQL-query aangemaakt worden en geperfectioneerd worden via de grafische interface.

Enkele use cases werden wat aangepast of verder verduidelijkt (zoals bijvoorbeeld “Dashboard bekijken”) en er is één use case bijgekomen namelijk: “Routes vergelijken”. “Statuspagina bekijken” is hernoemd naar “Logpagina bekijken”.

5.15 Look & Feel Requirements

NFR	Look & Feel Requirements
Indicator	Huisstijl
Meetvoorschrift	De stijl moet consistent zijn.
Norm	De stijl moet overzichtelijk en fris zijn (bootstrap is in orde).

NFR	Look & Feel Requirements
Indicator	Visueel overzichtelijk (snel tot benodigde informatie komen)
Meetvoorschrift	Men moet duidelijk kunnen afleiden waarvoor alles dient.
Norm	Men moet tussen gerelateerde pagina’s/secties snel kunnen browsen.

5.16 Usability & Humanity Requirements

NFR	Usability & Humanity Requirements
Indicator	Responsive voor verschillende apparaten
Meetvoorschrift	De applicatie moet beschikbaar zijn op alle moderne toestellen.
Norm	Op moderne toestellen moet elke feature beschikbaar en bruikbaar zijn.

5.17 Operationele & omgevingsrequirements

NFR	Operationele & omgevingsrequirements
Indicator	Productieomgeving en databasevoorzieningen voor de applicatie
Meetvoorschrift	De webapplicatie moet vlot draaien.
Norm	De applicatie moet de nodige gebruikers en operaties perfect kunnen uitvoeren zonder vertragingen.

5.18 Wettelijke requirements

NFR	Wettelijke Requirements
Indicator	Strikte afscherming en overeenkomst met providerinfo.
Meetvoorschrift	Alle data en publieke toegang moeten een geldige licentie hebben.
Norm	Er moeten concrete afspraken gemaakt worden met betrekking op de data van de providers.

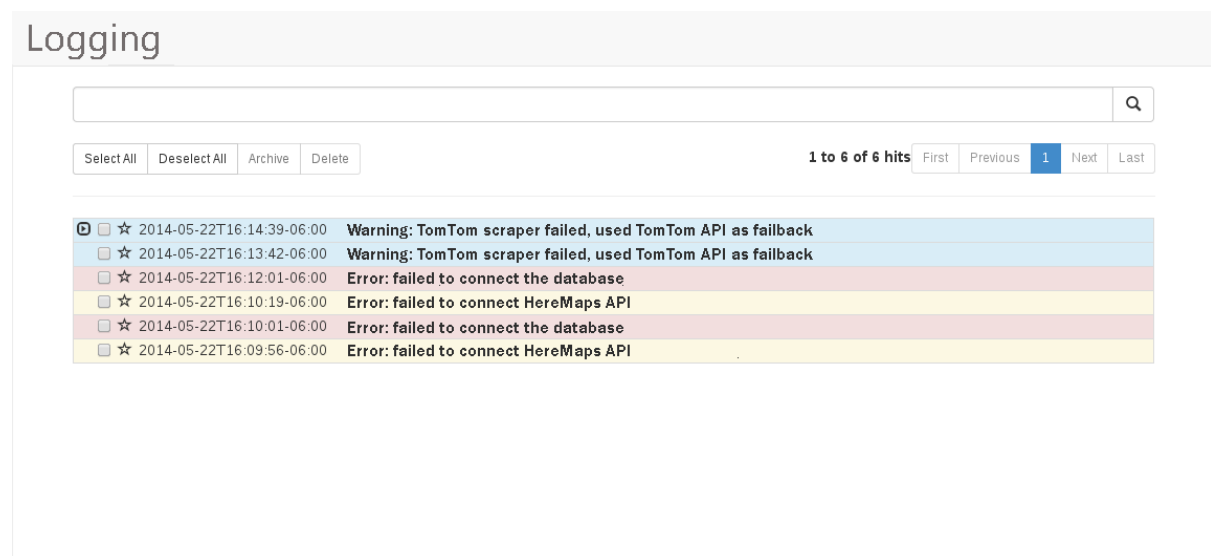
6 Analyse: Mockups

6.1 Initiële Mockup Dashboard



Figuur 1: Mockup Dashboard

6.2 Initiële Mockup Logging

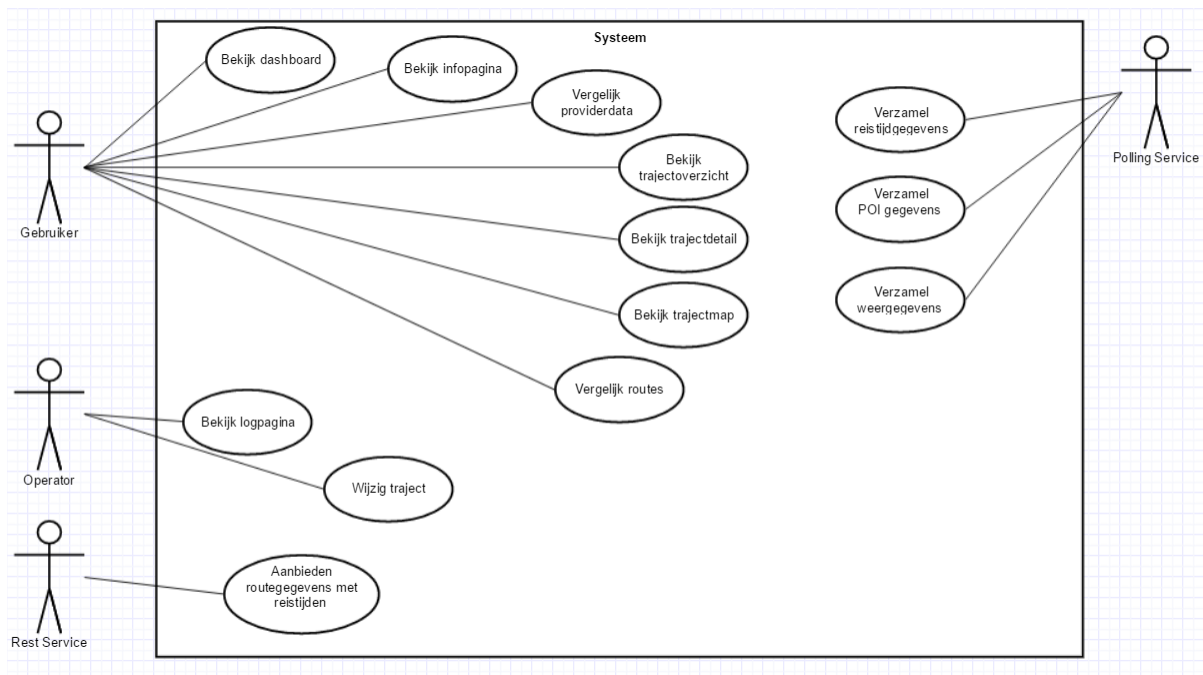


Figuur 2: Mockup Logging

Voor overige visuele ondersteuningen (in de vorm van screenshots) verwijzen we graag naar de gebruikershandleiding.

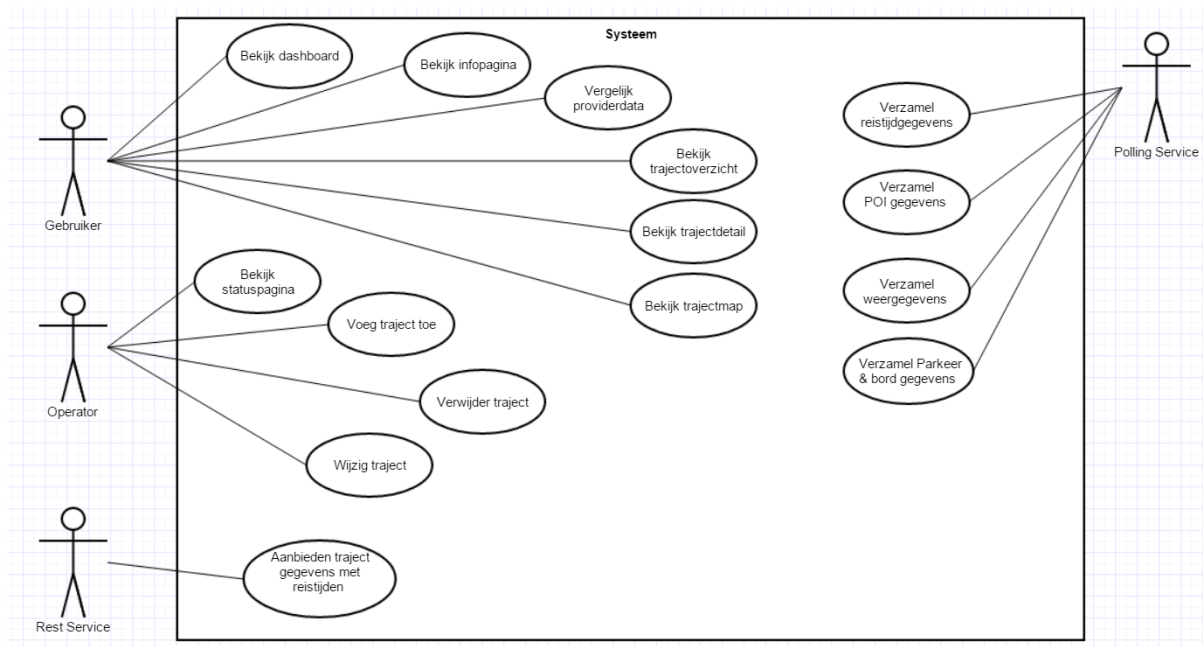
7 Analyse: Diagrammen

7.1 Use case Diagram (Final)



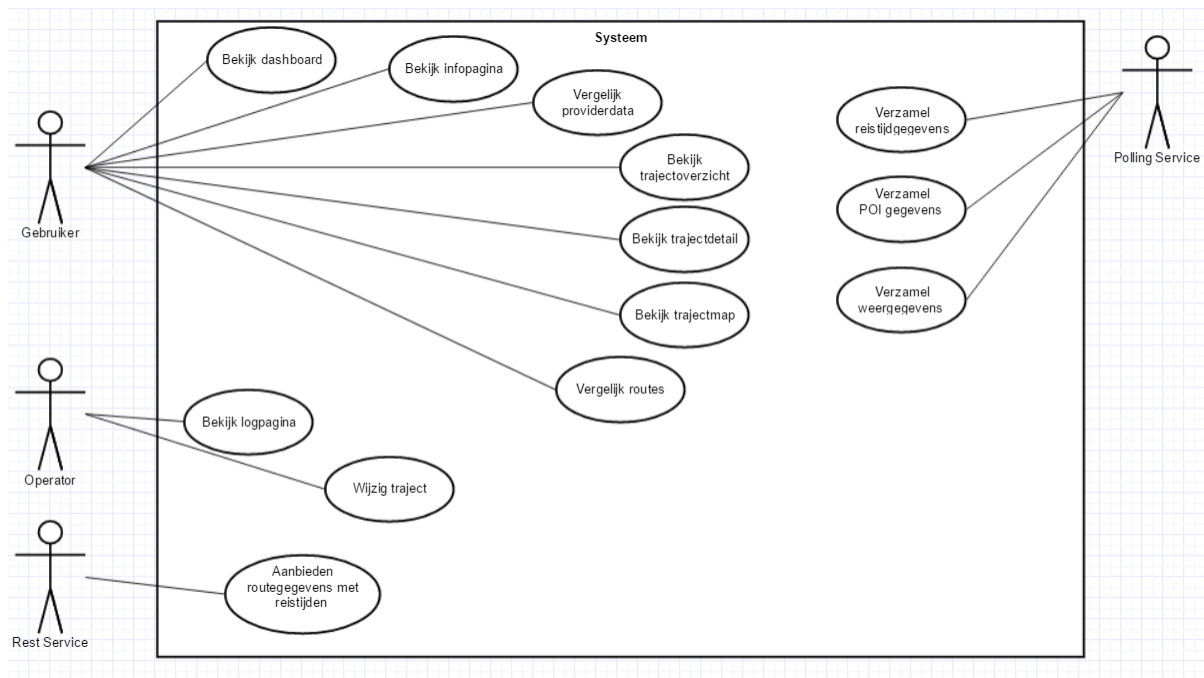
Figuur 3: Use Case Diagram

7.2 Use case Diagram - Einde Sprint 1



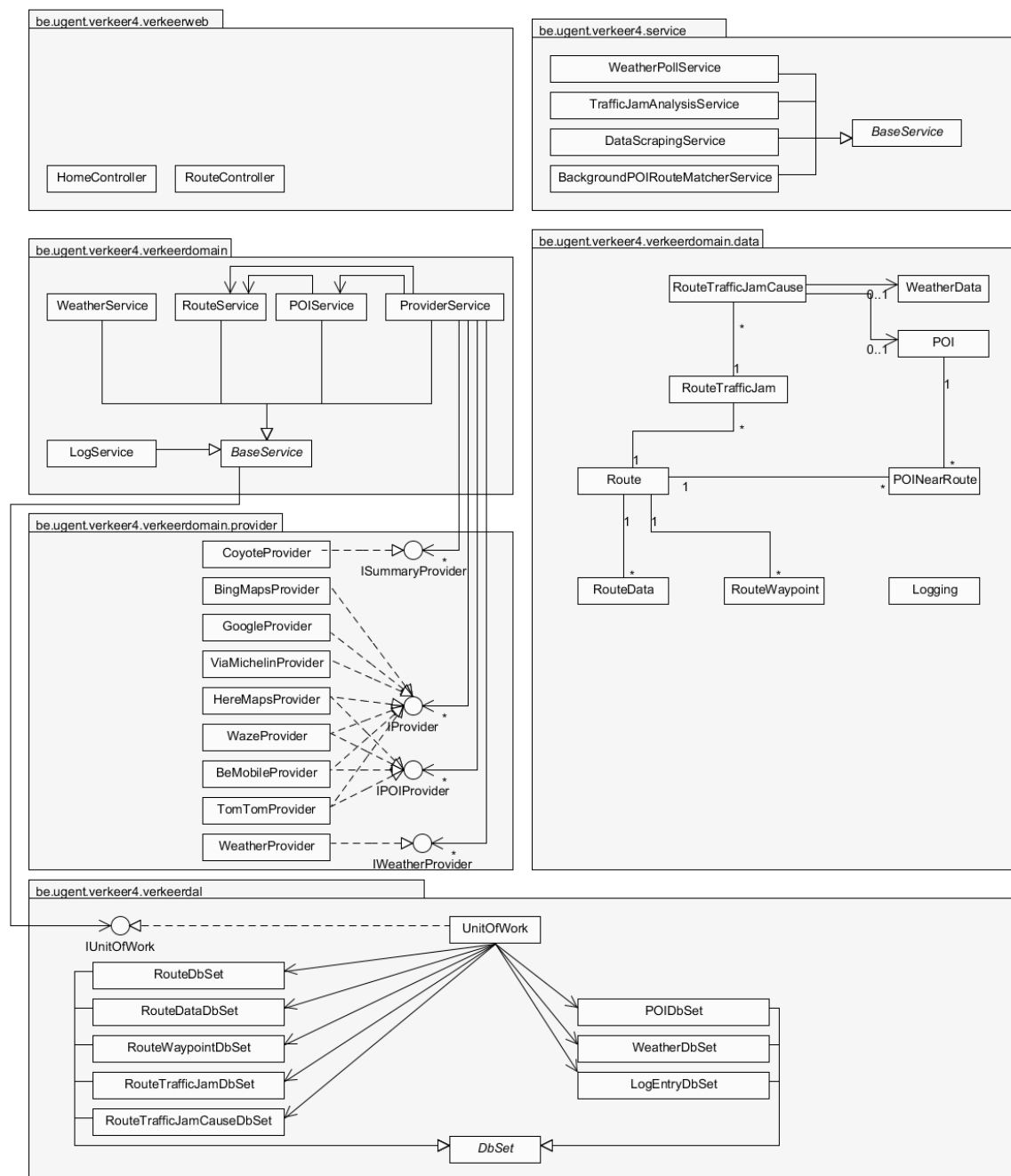
Figuur 4: Use Case Diagram - Einde Sprint 1

7.3 Use case Diagram - Einde Sprint 2



Figuur 5: Use Case Diagram - Einde Sprint 2

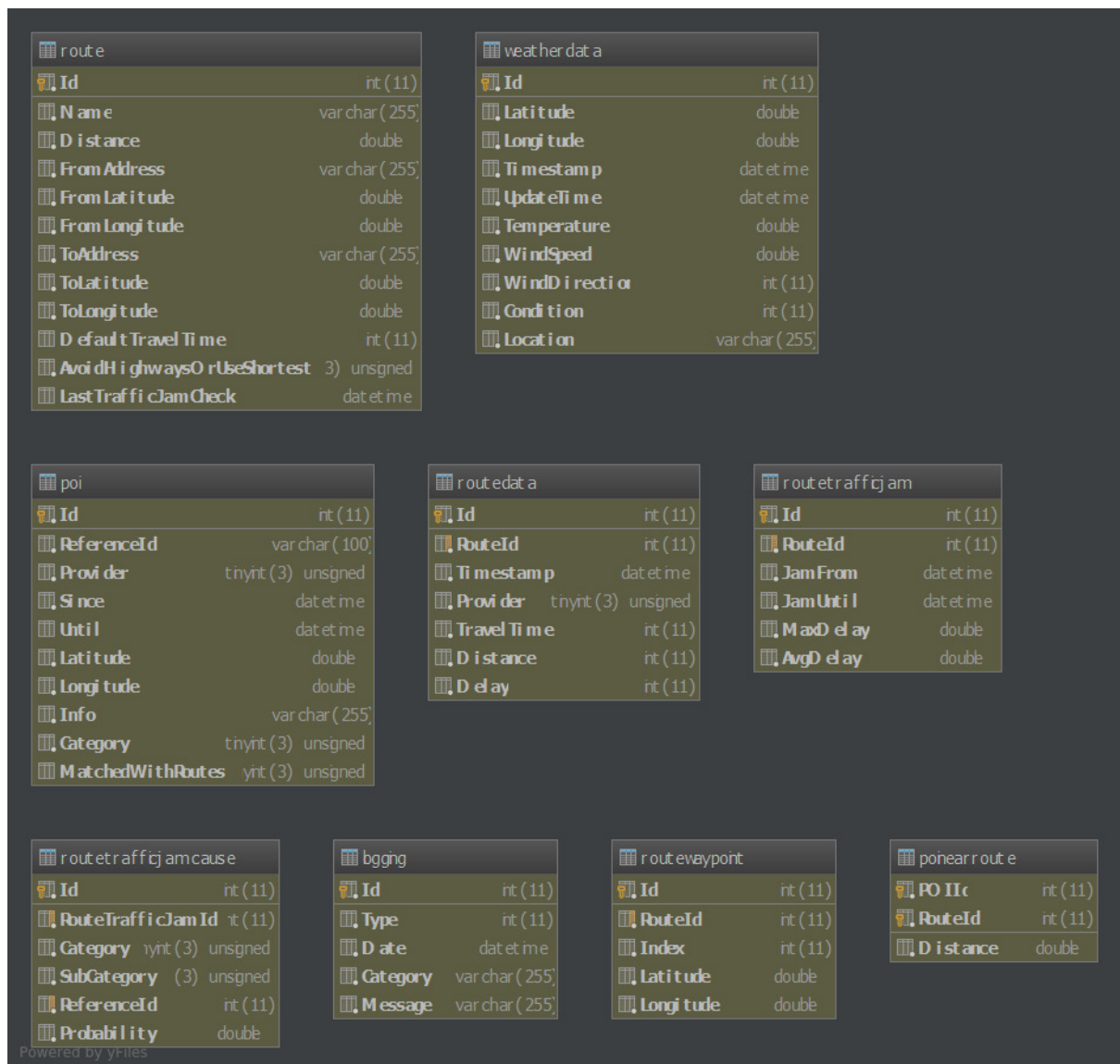
7.4 Klassendiagram - Projectstructuur



Figuur 6: Diagram - Projectstructuur

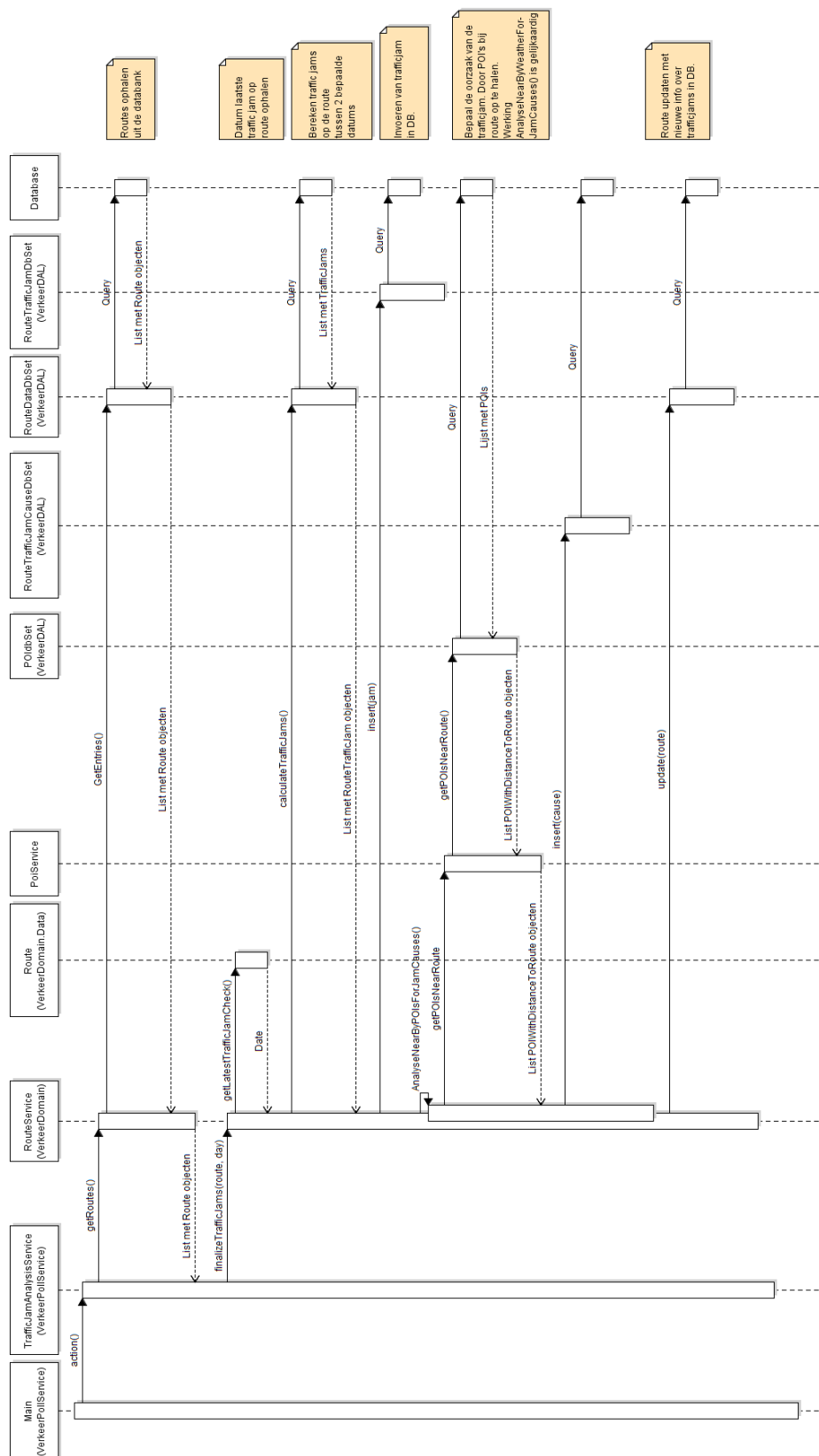
Zie technische handleiding voor een gedetailleerde uitleg van de verschillende componenten.

7.5 ERD



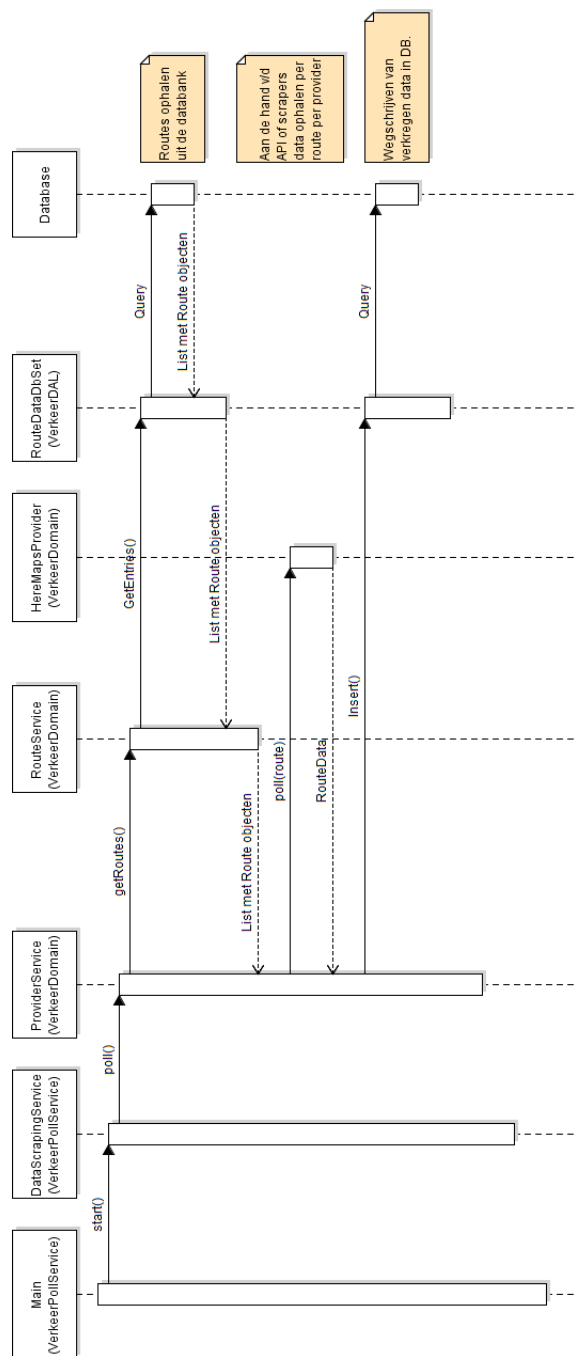
Figuur 7: Diagram - ERD

Projectdossier VOP



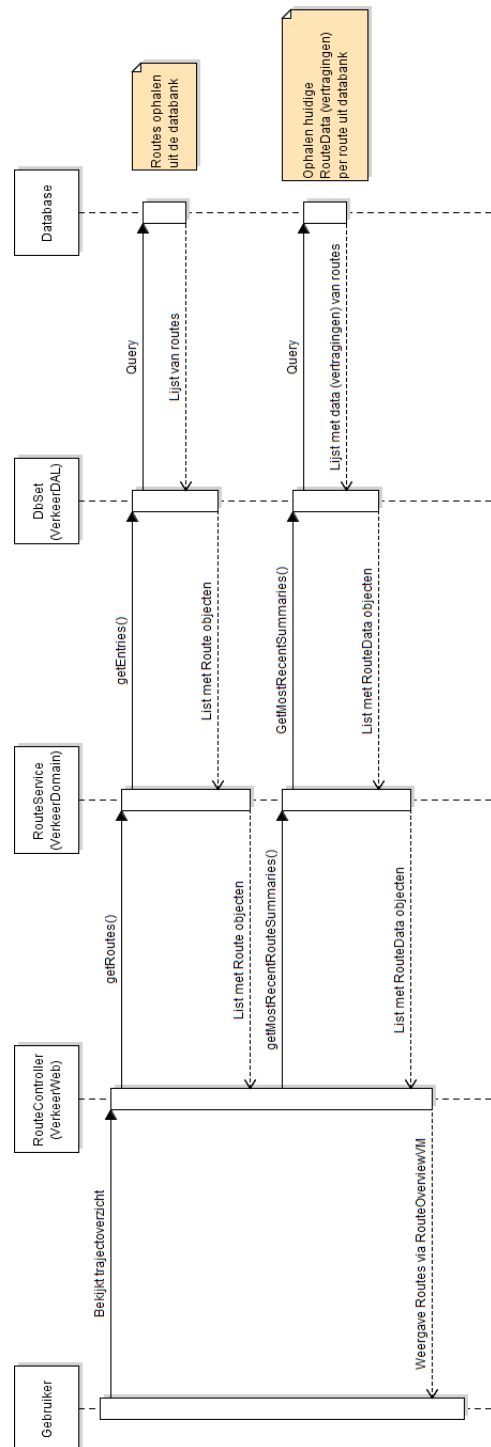
Figuur 8: Diagram - Bepalen File en Oorzaak

7.7 Sequentiediagram - Polling verkeersgegevens



Figuur 9: Diagram - Polling verkeersgegevens

7.8 Sequentiedagram - Overview Routes



Figuur 10: Diagram - Overview Routes

8 Feedback

In deze sectie staat een samenvatting van de communicatie met de klant. Deze punten verklaren een aantal beslissingen die genomen werden voor dit project.

8.1 Vragen Team aan Klant

Hoe kunnen we best een betrouwbaarheid aan een provider toekennen op basis van de gegevens die we van verschillende providers opgeslagen hebben?

Dit wordt het best door het team zelf bepaald.

Hoever terug moet de gegevens bewaard worden? Weken, maanden, jaren, altijd?

Liefst minstens 2 jaar.

Moet er 's nachts ook even vaak gepolled worden? Met andere woorden zijn er momenten waarop we het systeem minder moeten belasten?

Liefst consistent indien mogelijk, indien niet kan er eventueel tussen middernacht en 6u 's ochtends niet gepolled worden om requests op de API-calls te besparen.

Hoe moeten we de kleuren voorstellen van routes? Vb. Op basis van de vertraging?

Groen, oranje, rood naarmate de vertraging oploopt is in orde.

Moeten er nog wijzigingen gebeuren op de trajectdetailpagina?

Exporteren naar excel zou handig zijn.

Welke filters moeten toegepast kunnen worden op de trajectdetailpagina?

Onbeantwoord. (Aanvulling/wijziging van huidige filters..)

Welke grafieken moeten zeker aanwezig zijn?

Onbeantwoord. (Aanvulling/wijziging van huidige grafieken..)

Moet de applicatie meertalig zijn?

Engels is voldoende.

8.2 Vragen Klant aan Team

Is het mogelijk waarden tussen verschillende providers die sterk afwijken aan te duiden?

Deze worden nu in het rood aangeduid.

Is het mogelijk trajecten met de grootste vertragingen bovenaan te zetten?

Ja, is aangepast.

Kunnen de tijden worden weergegeven in de vorm van minuten en seconden i.p.v. seconden.

Ja, is aangepast.

8.3 Afspraken Team

Technische details?

- | | |
|--------------------------|------------------|
| • Java 8 | • Glassfish 4.11 |
| • Spring MVC | • Bootstrap |
| • IDE: Netbeans/IntelliJ | • Highcharts |
| • MariaDb | • Leaflet |

Eén gezamenlijke database of ieder lokaal afzonderlijk?

Iedereen werkt lokaal op zijn eigen database-instantie.

Wat is de projectstructuur

3-lagenstructuur. Enerzijds VerkeerWeb (frontend) en anderzijds VerkeerDomain, VerkeerDomainData en VerkeerDal (backend).

9 Kwaliteitscontrole

In deze kwaliteitscontrole worden enkele testen beschreven. Deze rubriek somt als het ware de verschillende testplannen op. Er wordt telkens wat uitleg gegeven over de gebruikte aanpak gevolgd door een beschrijving of een uitvoering van dergelijke test.

9.1 Overzicht Testplan

De unit testen zullen vooral de business logica van de verschillende use cases testen op een mock data laag met testgegevens. Hierdoor zijn de testgegevens steeds hetzelfde en kan bij aanpassingen aan de services gemakkelijk geverifieerd worden of er geen bugs geïntroduceerd zijn.

De integration tests testen de verbinding met de database en de stabiele werking van het systeem in het algemeen. Ook hier wordt gebruik gemaakt van de main flows van de belangrijke use cases.

Tenslotte testen de usability testen de algemene gebruiksvriendelijkheid van de applicatie, o.a of het voor een leek die niet vertrouwd is met het systeem vlug duidelijk is waar alles te vinden is en hoe gemakkelijk hij/zij het vindt om bepaalde gegevens op te vragen. Verder wordt er ook gekeken naar de performantie en responsiviteit van de applicatie door heel veel data in de databank te steken en te kijken of er geen vertragingen naar boven komen.

9.2 Unit Tests

Om zeker te zijn dat bepaalde algoritmes van het project correct werken, worden er unit testen aangeemaakt. Om maximaal rendement uit de unit testen te halen, worden er geen triviale testen gemaakt voor bijvoorbeeld setters zonder logica. Er zal gebruik gemaakt worden van het framework Mockito om dependencies te mocken en JUnit voor de testen.

De belangrijkste algoritmes die getest moeten worden zijn:

- Het detecteren van extreme providers op de detailpagina (door Aaron).
- Het gemiddelde berekenen van een route over een bepaalde periode en enkel voor bepaalde providers (door Aaron).
- Het detecteren van files (door Dwight).

9.3 Integration Tests (via Selenium)

Het is een discussiepunt om de functionele acceptatie-testen van selenium nu al dan niet te categoriseren onder integratietesten. Omdat de testen toch wel data van backend ophalen via de frontend alsof ze een gebruiker zijn hebben we toch gekozen om deze hier te plaatsen. Een correctere term is waarschijnlijk smoke of sanity test.

Selenium IDE is een plugin waarmee je browseractiviteiten kan opnemen. Men kan elementen op de webapplicatie selecteren en deze controleren. Na het opnemen kan zo'n test case (of suite) geëxporteerd worden naar testen die uitgevoerd kunnen worden in bijvoorbeeld JUnit.

Een voorbeeld van zo'n test is het ophalen en controleren (via regex momenteel) of elke route wel een acceptabele normale reistijd, huidige reistijd en vertraging heeft.

Een demo en voorbeeld zijn te zien op <https://www.youtube.com/watch?v=8HOH1TDGwdo>.

9.4 Usability Tests (Use case testing)

Met use case testing bedoelt men soms het “het maken van” use cases in een Agile of TDD (Test Driven Development)-omgeving. In deze rubriek hebben we echter de use cases genomen en deze omgezet in een stappenplan. Het doornemen van deze stappen test zo onze applicatie aan onze use cases. Op die manier worden de gewenste doelstelling getest en worden ook de use cases deel van de documentatie van het eindproduct.

Bij het maken van het stappenplan hebben we geprobeerd alles te schrijven vanuit het standpunt van de gebruiker. Om die reden zijn er enkele use cases die niet (rechtstreeks) getest kunnen worden door de gebruiker. Men zou alternatieven kunnen voorzien maar dit is niet echt nodig. In deze use cases worden alle gegevens verzameld en weggeschreven naar de databank. Deze use cases zijn dan ook niet echt bedoeld om te testen in een usability test. SQL-queries of inloggen op de server en onze testscripts runnen zijn workarounds. Het is echter zo dat als de andere use case die getest worden aan de hand van usability tests (ook) gaan falen als deze use cases niet correct zijn. Daarom mogen we concluderen dat deze use cases correct doorloopbaar zijn zonder expliciet een stappenplan te volgen.

- Uc: Verzamelen POI-gegevens
- Uc: Verzamelen weergevens
- Uc: Verzamel reistijdgegevens

Volgende 2 tabellen toont het stappenplan die opgemaakt werd op basis van de use cases. We doorlopen het stap voor stap een ✓ toont aan dat alles geslaagd is terwijl een ✗ aantoont dat er nog iets niet in orde is.

UC: Bekijk routeoverzicht		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op routeoverzicht klikken.	✓
3.	Een overzicht van de routes wordt getoond.	✓
4.	Een overzicht van de routes wordt getoond.	✓
UC: Bekijk routedetail		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op routeoverzicht klikken.	✓
3.	Gebruiker kan op een route klikken voor de details.	✓
4.	De routedetails worden getoond.	✓
UC: Bekijk routemap		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op routemap klikken.	✓
3.	De routemap wordt getoond.	✓
UC: Vergelijk providerdata		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op routeoverzicht klikken.	✓
3.	Gebruiker kan op een route klikken voor de details.	✓
4.	Gebruiker kan providerdata vergelijken.	✓
UC: Wijzig route		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op routeoverzicht klikken.	✓
3.	Gebruiker kan op wijzig klikken.	✓
4.	Gebruiker kan de details van de route wijzigen.	✓
UC: Logpagina bekijken		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan op logpagina klikken.	✓

Tabel 21: Usability Tests (use case testing (1/2))

UC: Dashboard bekijken		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker kan dashboard(s) bekijken.	✓
UC: Aanbieden routegegevens met reistijden.		
Stappenplan:		
1.	Gebruiker maakt een specifieke request.	✓
2.	De gebruiker krijgt correcte gegevens naar gelang de request.	✓
UC: Vergelijk routes		
Stappenplan:		
1.	Gebruiker gaat naar de website.	✓
2.	Gebruiker klikt op vergelijk routes.	✓
3.	Gebruiker geeft routedetails op.	✓
4.	Een grafiek wordt getoond waarin de routes vergeleken kunnen worden.	✓

Tabel 22: Usability Tests (use case testing (2/2))

Bovenstaande stappen werden overlopen op 16 april. Alle zaken die falen zijn nog geplande, niet-geïmplementeerde features.

9.5 Usability Tests (scenario)

Deze manueel uitgevoerde scenarios dienen de usability and user experience (UX) te testen. Er werd aan 3 personen gevraagd om een specifiek scenario door te nemen. De testpersonen moeten aanduiden of ze de verschillende stappen van het scenario goed kunnen uitvoeren of niet, alsook het loggen van de gespendeerde tijd. Het uitvoeren van de verschillende stappen test als gevolg het gebruik van de applicatie. Tot slot werd er gevraagd aan de testpersonen of ze nog opmerkingen, vragen of onduidelijkheden hadden.

Er werd geopteerd om 3 profielen te benaderen. Onze keuze is uitgegaan naar een IT-deskundige en 2 niet-IT-deskundigen. Deze laatste groep verschilt dan nog eens in leeftijd (enederzijds ± 20 jaar en anderzijds ± 50 jaar).

Er is 1 use case die we zelf uitvoerig gebruikt (en dus getest hebben) en dat is “route wijzigen”. Het spreekt ook voor zich dat we op de productieomgeving deze use case liever niet laten derden doornemen door derden.

9.5.1 Template

Het scenario (of stappenplan) dat werd gegeven aan de testpersonen ziet er als volgt uit:

<i>Stap</i>	<i>Beschrijving</i>	<i>Ok</i>	<i>Ind.</i>	<i>Cum.</i>
1	Ik kan naar de website: http://verkeer-4.vop.tiwi.be/ surfen en inloggen met de credentials guest/1RRBpmM0KC.			
2	Ik kan het weer aflezen van de website.			
3	Ik kan de laatste tweets van “VerkeerGentB” zien.			
4	Ik kan een overzicht van alle routes zien met hun afstanden, standaardtijden, huidige reistijden en huidige vertragingen.			
5	Ik kan naar een route gaan en zijn weg visueel zien op een kaart.			
6	Ik kan de vertraging van Bing op Paryssteen (R4) northbound te weten komen.			
7	Ik kan een visueel overzicht krijgen op de kaart waar er momenteel vertragingen zijn.			
8	Ik kan visueel zien of er zich ergens probleempunten vertonen in de buurt van Gent.			
9	Ik kan te weten komen wat de huidige reistijd was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.			
10	Ik kan te weten komen wat de vertraging was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.			
11	Ik kan de data van vorige test opslaan als in een csv-bestand (te openen met excel).			
12	Ik kan te weten komen welke route volgens google het meeste vertraging heeft.			
13	Ik kan te weten komen welke route de grootste afstand heeft.			

Tabel 23: Scenario Usability Test Template

9.5.2 Persoon 1: IT-deskundige

<i>Stap</i>	<i>Beschrijving</i>	<i>Ok</i>	<i>Ind.</i>	<i>Cum.</i>
1	Ik kan naar de website: http://verkeer-4.vop.tiwi.be/ surfen en inloggen met de credentials guest/1RRBpmM0KC.	✓	00:30	00:30
2	Ik kan het weer aflezen van de website.	✓	00:05	00:35
3	Ik kan de laatste tweets van “VerkeerGentB” zien.	✓	00:00	00:35
4	Ik kan een overzicht van alle routes zien met hun afstanden, standaardtijden, huidige reistijden en huidige vertragingen.	✓	01:00	01:35
5	Ik kan naar een route gaan en zijn weg visueel zien op een kaart.	✓	01:00	02:35
6	Ik kan de vertraging van Bing op Paryssteen (R4) northbound te weten komen.	✓	02:00	04:35
7	Ik kan een visueel overzicht krijgen op de kaart waar er momenteel vertragingen zijn.	✓	01:00	05:35
8	Ik kan visueel zien of er zich ergens probleempunten vertonen in de buurt van Gent.	✓	00:00	05:35
9	Ik kan te weten komen wat de huidige reistijd was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✓	02:00	07:35
10	Ik kan te weten komen wat de vertraging was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✓	01:00	08:35
11	Ik kan de data van vorige test opslaan als in een csv-bestand (te openen met excel).	✓	00:05	08:40
12	Ik kan te weten komen welke route volgens google het meeste vertraging heeft.	✓	02:00	10:40
13	Ik kan te weten komen welke route de grootste afstand heeft.	✓	00:10	10:50

Tabel 24: Scenario Usability Test Persoon 1: IT-deskundige

Opmerkingen:

- Onduidelijk navigatiemenu.
- In het algemeen niet zo duidelijk als je niet doorklikt.
- Zaken die nog niet geïmplementeerd zijn moeten er best nog niet op.

9.5.3 Persoon 2: niet-IT-deskundige (1)

<i>Stap</i>	<i>Beschrijving</i>	<i>Ok</i>	<i>Ind.</i>	<i>Cum.</i>
1	Ik kan naar de website: http://verkeer-4.vop.tiwi.be/ surfen en inloggen met de credentials guest/1RRBpmM0KC.	✓	00:40	00:40
2	Ik kan het weer aflezen van de website.	✓	00:05	00:45
3	Ik kan de laatste tweets van “VerkeerGentB” zien.	✓	00:00	00:45
4	Ik kan een overzicht van alle routes zien met hun afstanden, standaardtijden, huidige reistijden en huidige vertragingen.	✓	01:00	01:45
5	Ik kan naar een route gaan en zijn weg visueel zien op een kaart.	✓	00:30	02:15
6	Ik kan de vertraging van Bing op Paryssteen (R4) northbound te weten komen.	✓	01:00	03:15
7	Ik kan een visueel overzicht krijgen op de kaart waar er momenteel vertragingen zijn.	✓	01:00	04:15
8	Ik kan visueel zien of er zich ergens probleempunten vertonen in de buurt van Gent.	✓	00:00	04:15
9	Ik kan te weten komen wat de huidige reistijd was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✓	01:30	05:45
10	Ik kan te weten komen wat de vertraging was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✗	02:00	07:45
11	Ik kan de data van vorige test opslaan als in een csv-bestand (te openen met excel).	✓	00:30	08:15
12	Ik kan te weten komen welke route volgens google het meeste vertraging heeft.	✗	02:00	10:15
13	Ik kan te weten komen welke route de grootste afstand heeft.	✓	00:30	10:45

Tabel 25: Scenario Usability Test Persoon 2: niet-IT-deskundige (1)

Opmerkingen:

- Bij het weer wordt niet aangegeven dat dit van Gent is.
- CTT en D is niet duidelijk voor personen die er niet mee bezig zijn.
- Er zou een duidelijkere scheiding moeten zijn tussen de providers vanboven omdat het niet duidelijk is welke CTT en D bij welke provider horen.
- Sorteren niet duidelijk.

9.5.4 Persoon 3: niet-IT-deskundige (2)

<i>Stap</i>	<i>Beschrijving</i>	<i>Ok</i>	<i>Ind.</i>	<i>Cum.</i>
1	Ik kan naar de website: http://verkeer-4.vop.tiwi.be/ surfen en inloggen met de credentials guest/1RRBpmM0KC.	✓	01:40	01:40
2	Ik kan het weer aflezen van de website.	✓	00:10	01:50
3	Ik kan de laatste tweets van “VerkeerGentB” zien.	✓	00:10	02:00
4	Ik kan een overzicht van alle routes zien met hun afstanden, standaardtijden, huidige reistijden en huidige vertragingen.	✓	01:00	03:00
5	Ik kan naar een route gaan en zijn weg visueel zien op een kaart.	✓	00:30	03:30
6	Ik kan de vertraging van Bing op Paryssteen (R4) northbound te weten komen.	✓	02:00	05:30
7	Ik kan een visueel overzicht krijgen op de kaart waar er momenteel vertragingen zijn.	✓	01:00	06:30
8	Ik kan visueel zien of er zich ergens probleempunten vertonen in de buurt van Gent.	✓	00:10	06:40
9	Ik kan te weten komen wat de huidige reistijd was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✓	02:30	09:10
10	Ik kan te weten komen wat de vertraging was per provider op 17 maart tussen 6:00 en 11:00 in de Rooigemlaan (R40) northbound.	✗	02:30	11:40
11	Ik kan de data van vorige test opslaan als in een csv-bestand (te openen met excel).	✓	01:00	12:40
12	Ik kan te weten komen welke route volgens google het meeste vertraging heeft.	✗	02:30	15:10
13	Ik kan te weten komen welke route de grootste afstand heeft.	✓	01:00	16:10

Tabel 26: Scenario Usability Test Persoon 3: niet-IT-deskundige (2)

Opmerkingen:

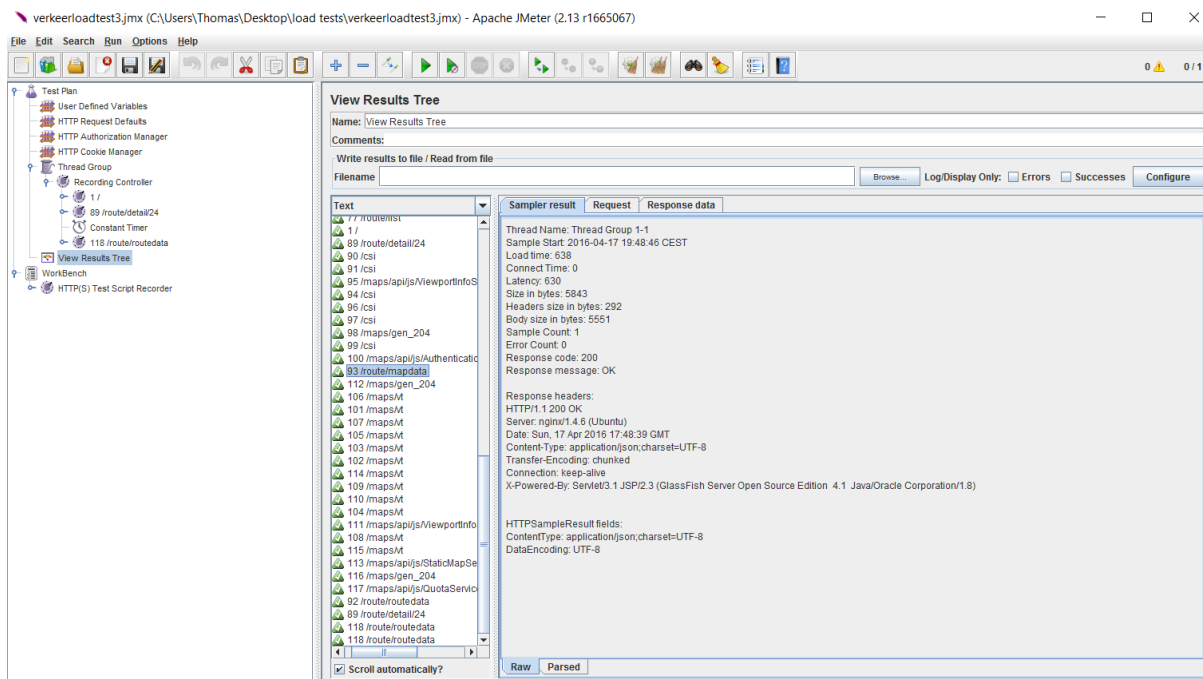
- CTT en D is niet duidelijk voor personen die er niet mee bezig zijn.
- Er zou een duidelijkere scheiding moeten zijn tussen de providers vanboven omdat het niet duidelijk is welke CTT en D bij welke provider horen.
- Sorteren niet duidelijk.
- Opslaan is niet duidelijk aangegeven.

9.6 Load Tests

Apache JMeter is een load testing tool waarin je users kan simuleren op een webapplicatie. In de vorm van “threads” worden HTTP requests uitgevoerd naar de verschillende pagina’s. Hiermee wordt er geverifieerd dat de server een bepaald aantal concurrente users aankan.

Er zijn drie testen voorzien. De eerste bezoekt enkel de homepage (en vult de basic auth correct in). De tweede gaat vervolgens nog naar de map view en de derde simuleert een gebruiker die van het dashboard naar de overview surft om nadien een route te selecteren die hij in detail wil bekijken. De testen worden lokaal uitgevoerd, maar de requests gaan naar de productieserver.

Er is een video om een demo van deze testen te bekijken: <https://www.youtube.com/watch?v=h2HiMyBYgCI>.



Figuur 11: Apache JMeter demo

De eerste server, die al verzadigd was, kon de testen niet goed aan. Na de migratie naar Digital Ocean werd de applicatie al stukken sneller. De testen toonden aan dat de applicatie vlot gebruikt kan worden zonder merkbare vertragingen tot 20 threads (users). De complexiteit van de gevraagde acties bepaalt vooral de vertraging (dit is waarschijnlijk te wijten aan de I/O-operaties van de databank). De applicatie zal dus perfect beheerbaar en vlot werken voor een klein tot middelgroot team op de huidige Digital Ocean droplet.

Op een tweede testserver, die dubbel zo krachtig is als de huidige productieomgeving (2GB geheugen en 2 CPU's) hebben we de derde stresstest ook uitgevoerd met 100 threads (gesimuleerde gebruikers). De webapplicatie vertoonde een vertraging, maar was nog steeds operationeel. Sommige threads kregen niet altijd een correct antwoord maar de “live” test was wel positief. Dezelfde test met 50 threads gaf geen enkel probleem!

10 Technische handleiding

De applicatie is onderverdeeld in 3 lagen:

10.1 GUI layer

Dit is de volledige MVC website met glassfish als server en bevat

- **Controllers:** de controllers beheren de HTTP acties, valideren de user input en roepen de overeenkomstige services uit het domain aan.
- **Views:** de views bestaan uit standaard JSP pages die met behulp van een model dat eraan meegegeven wordt in de controllers de html kan genereren.
- **ViewModels:** de gegevens die de controllers verzamelen van de domain services worden omgezet en indien nodig gecombineerd tot viewmodels die alle gegevens bevatten dat moet getoond worden in de pagina.
- **DataObjects:** voor ajax calls wordt er JSON teruggegeven die in de javascript code kan verwerkt worden. De json wordt automatisch gegenereerd van de data objecten die de controller teruggeeft. In feite zijn de view models en data objects gelijkaardig, de view models worden verwerkt aan server side, terwijl de data objects aan client side verwerkt worden.
- **Shared Views:** er zijn een aantal shared views voorzien zodat er geen html moet gecopy paste worden tussen meerdere views, zoals de inhoud van de head tag. Zodra html op meerdere views wordt gebruikt is het interessant om hiervoor een shared view te voorzien (met eventueel op te geven parameters zoals de title parameter in de head shared view).
- **Javascript/Typescript:** de code die aan client side uitgevoerd moet worden is deels geschreven in Typescript¹ en deels in plain javascript. De scripts zitten per pagina elk afzonderlijk in aparte bestanden die met script tags geinclude worden in de views.
- **CSS:** Er wordt gebruik gemaakt van Bootstrap als layout framework. Wanneer extra CSS moet toegepast worden op een pagina is hiervoor een apart stylesheet bestand voorzien die met een link tag wordt geinclude in de view.

10.2 Domain layer

De domain layer is opgesplitst in 2 delen:

10.2.1 Domain

Hierin zitten alle services die de business logic van de applicatie verwerken. De services maken gebruik van de domain objects. Voor elke service is een interface voorzien en hun bijhorende implementatie, hierdoor kan later nog gebruik gemaakt worden van dependency injection om de services automatisch te construeren. Elke service erft over van de BaseService die de UnitOfWork instantie voorziet om de gegevens uit de DAL op te vragen. De services zijn ongeveer ingedeeld volgens business unit:

- **RouteService:** de Route service beheert alles omtrent de routes zelf en zorgt dat routes kunnen opgevraagd worden met hun onderliggende data, waypoints en gedetecteerde files, kunnen geupdate worden

¹Typescript is een superset van javascript met een strong type system en transpiled naar javascript

- **ProviderService:** de Provider service voorziet alle nodige implementatie om de route data van alle routes efficiënt te verzamelen van verschillende providers. Zowel de route data (vertraging, huidige reistijd, ..) als de POI's (incident, road closed, ...) worden met behulp van de verschillende geregistreerde providers opgevraagd. Alle providers zijn apart geïmplementeerd in de sub package "provideren implementeren naargelang de beschikbare functionaliteit de interfaces IProvider (route data van 1 route opvragen), ISummaryProvider (route data van alle routes in 1x opvragen), IPOIProvider (poi's opvragen binnen een bounding box), IWeatherProvider (weergegevens opvragen rondom de routes). Sommige providers maken gebruik van een API die aangesproken wordt, andere maken gebruik van perl scrape scripts die de gevraagde gegevens teruggeven. Als beide beschikbaar zijn wordt steeds de scrape scripts eerst uitgetest, hierdoor wordt er geen verbruik om de API keys geregistreerd.
- **POIService:** de POI Service zorgt voornamelijk voor het opslaan en opvragen van POI's binnen een regio en bepaalde periode. De service kan ook de POI's matchen met nearby routes zodat POI's steeds voor één bepaalde route kan opgevraagd worden.
- **WeatherService:** de Weather service voorziet methodes om de opvraagde weergegevens op te slaan in de databank en terug op te vragen.
- **LogService:** de log service is een singleton dat doorheen de volledige applicatie kan worden aangesproken om logging naar de databank weg te schrijven.

10.2.2 Domain Objects

Dit zijn alle data objecten die gebruikt worden en zijn ook de objecten die in de DAL layer worden gemapped naar tabellen in de database. De volgende objecten worden momenteel gebruikt:

- **Route:** stelt een route voor met basisgegevens als naam en start en eindpunt (in latitude en longitude). Verder is er nog een flag dat aangeeft of routes moeten berekend worden als snelste of eerder als kortste (of avoid highways). Dit is nodig voor bepaalde routes die anders de autostrade zouden kiezen terwijl dat niet gewenst is.
- **RouteData:** route data is een data punt dat via de polling van verschillende providers is verzameld en opgeslagen? Het datapunt is van een specifieke provider op een bepaald tijdstip en bevat de reistijd en vertraging. De afstand van de route wordt ook mee bijgehouden zodat kan gedetecteerd worden of de provider een alternatieve route heeft voorgesteld i.p.v. de gewenste route.
- **RouteWaypoint:** elke route is van de start tot einde onderverdeeld in segmenten. Een route waypoint is zo een tussenliggend punt en bevat enkel de geografische coördinaten. Waypoints worden berekend bij het wijzigen van de start/eind positie van een route en worden gebruikt bij het tonen van de route op een kaart en bij de controle of POI dicht bij een route ligt.
- **POI:** stelt een gebeurtenis voor op een bepaald punt en heeft een tijdsperiode wanneer de POI actief door de provider werd voorgesteld, de Until wordt pas ingevuld wanneer de POI niet meer wordt teruggegeven door de provider, d.w.z. dat alle actieve POI's degene zijn met Until = NULL. POI's hebben ook een vaste categorielijst waartoe ze behoren, die op de website vertaald wordt naar de overeenkomstige icoontjes. De reference id van een POI is de unieke ID die de provider aan de POI gegeven heeft en wordt gebruikt om te detecteren bij opeenvolgende polls of het over dezelfde POI gaat of niet. Tenslotte bevat een POI ook nog een flag dat aangeeft of de POI reeds verwerkt is door de service die POI's matched met nearby routes.
- **POINearRoute:** POI's worden periodiek gematched met naburige routes. Dit object stelt zo een matching tussen een POI en een route voor.

- **RouteTrafficJam:** elke dag wordt de data van alle routes geanalyseerd en worden file periodes opgespoord en deze periodes worden opgeslagen als dit object in de databank. Hierdoor moet de zware berekening maar éénmaal gebeuren en kan men nadien gewoon deze objecten raadplegen. Buiten de start en eind van de periode bevat het object ook nog een gemiddelde en maximum van de vertraging over de periode.
- **RouteTrafficJamCause:** voor elke file probeert de service ook oorzaken te zoeken wat de aanleiding zou geweest zijn, vb. accidenten die actief waren aan de start van de file zijn een goede kandidaat. Elke oorzaak wordt opgeslagen als een **ROuteTrafficJamCause** object met een waarschijnlijkheidskans. Een oorzaak heeft een categorie (POI, Weer,...) en een subcategorie dat de categorie van bvb de POI zelf mee opslaat. Hierdoor kan er efficiënt een overzicht getrokken worden van alle files met hun oorzaken.
- **WeatherData:** dit object bevat de weergegevens op een bepaald tijdstip en kan zo mee opgenomen worden als oorzaak bij files (vb mist of zware regenbui).
- **LogEntry:** Alle warnings en errors die zowel de applicatie als de poll service produceert worden opgeslagen in de databank. Het object is een voorstelling van zo'n log lijn en bevat buiten de omschrijving ook nog waar het is opgetreden en over welk type (warning/error) het gaat.

De domain data bevat ook nog een aantal “composite” of samengestelde objecten die door JOINs in de database werden gecombineerd, vb **POIWithDistanceToRoute** is een POI object maar bevat nog een extra veld afstand dat door een join met **POINearRoute** is gecombineerd.

10.3 DAL layer

De DAL layer bevat de repositories (**DbSet<T>** klassen) van alle domain objecten om en voorzien standaard methodes voor de CRUD van de objecten. Als er specifieke gegevens moeten worden opgevraagd of extra filters moeten toegepast worden kan de repository van een object overgeërft worden en kan daarin de extra functionaliteit in worden voorzien. Om de applicatie efficiënt te houden is het nuttig om de joins reeds op de database uit te voeren en de gecombineerde gegevens terug te geven naar de domain services ipv de gegevens manueel in code te combineren.

Alle repositories zijn opvraagbaar via een overkoepeld **UnitOfWork** object dat ook opgesplitst is in een interface en implementatie. Hierdoor is het eenvoudiger de repositories te mocken voor unit testen en kan ook dependency injection gebruikt worden om in één keer de volledige DAL te initialiseren.

De database wordt aangesproken met een micro ORM **Sql2o**. Deze library zorgt ervoor dat objecten kunnen vertaald worden naar parameters in de SQL queries en dat de dataset dat teruggegeven wordt automatisch vertaald wordt naar objecten. Aangezien we na verloop van tijd met veel data werken (miljoenen rijen) hebben we de overhead van de ORM zoveel mogelijk beperkt. Hibernate is gemakkelijk 10x trager dan **Sql2o**, dat maar een fractie trager is dan het manueel mappen van gegevens op de objecten.

10.4 Poll service

Buiten de standaard 3-tier structuur is er ook nog een extra console applicatie voorzien dat periodiek verschillende acties uitvoert. Elke actie wordt in een aparte service geïmplementeerd en erft over van de **BaseService** klasse die de scheduling van de actie op zich neemt. Momenteel zijn de volgende acties gedefiniëerd:

- **DataScrapingService:** (elke 5min) deze service polled voor alle routes de **RouteData** gegevens aan alle providers. Per route worden de poll acties in parallel uitgevoerd zodat er zo weinig mogelijk verschil in tijd zit tussen de verkregen waarden. De service verzamelt ook alle POI gegevens voor elke provider.

- BackgroundPOIRouteMatcherService: (elke 5min) deze service gaat na welke POI's nog niet gematched zijn met naburige routes en zal deze gaan matchen. Hierdoor worden POINearbyRoute objecten aangemaakt en wordt nadien de POI matching status geupdate.
- TrafficJamAnalysisService: (elke 24u) deze service bekijkt de gegevens van alle routes dag per dag en zal de file periodes met hun oorzaken achterhalen. Alleen dagen die reeds volledig in het verleden liggen komen in aanmerking, aangezien voor die dagen geen gegevens meer zullen wijzigen.
- WeatherPollService: (elke 5min) deze service polled alle weer providers om de recentste weergegevens op te slaan.

De console applicatie draait momenteel in een screen sessie op de server, waardoor het mogelijk is de output gemakkelijk te bekijken.

11 Gebruikershandleiding

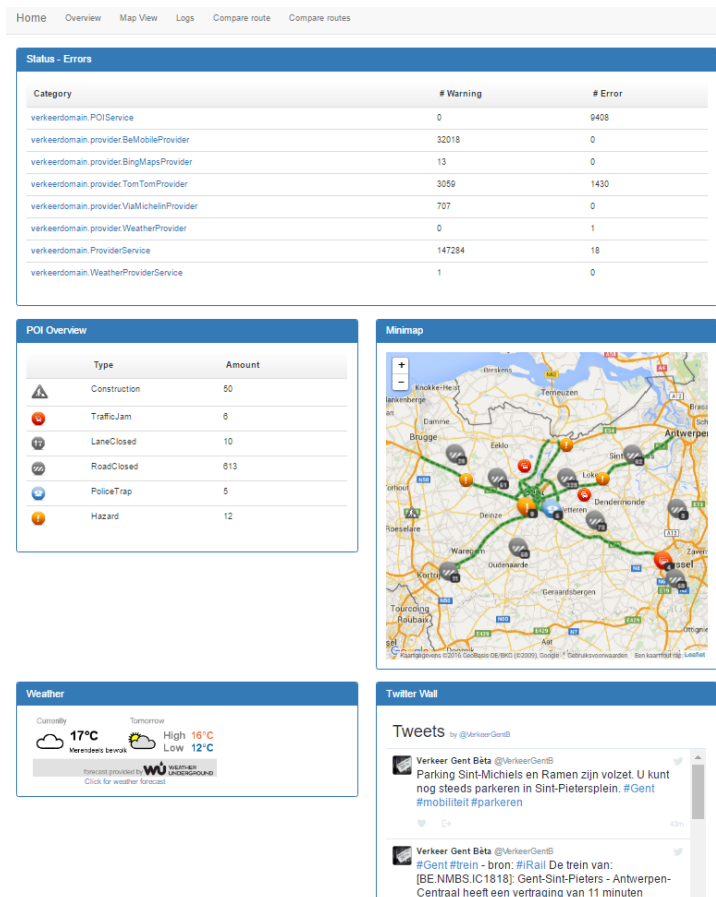
11.1 Inleiding

Dit is een handleiding geschreven als informatieve bron bij het gebruik van de website “Verkeer-4.vop.tiwi.be”. Het beoogt een overzicht te geven bij het opzoeken van verkeersinformatie in en rond Gent. Dit zowel realtime als in het verleden.

11.2 Het Dashboard

Bij het surfen naar “Verkeer-4.vop.tiwi.be” zal u verwezen worden naar het dashboard van de website. Hier kunt u een korte samenvatting vinden van de belangrijkste informatie rond verkeer.

- **Status - Errors** geeft een overzicht van het aantal gebeurtenissen per categorie rond de applicatie. Het geeft een beeld van als er een fout optreed.
- **POI Overview** toont het aantal van de huidige “Points Of Interest”. Hier worden onder andere ongevallen, trajectcontroles en belangrijke evenementen aangehaald.
- De **Minimap** zorgt voor een visuele weergave van de verkeerssituatie in en rond Gent. Zowel de files als de POI's worden weergegeven.
- De **Weather widget** wordt gebruikt om een korte schets van het weer in Gent te weergeven. Dit kan het verkeer namelijk sterk beïnvloeden.
- De **Twitter Wall** toont de recente gebeurtenissen in en rond Gent die werden gemeld door de stad Gent.

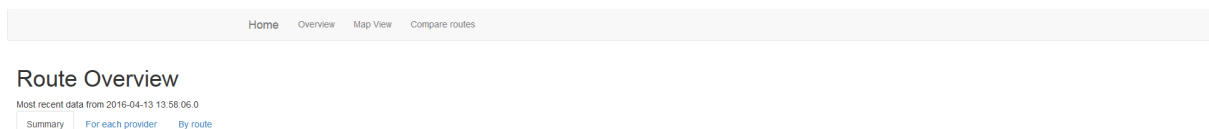


Figuur 12: Handleiding - Dashboard

Om verder te gaan naar de detail pagina's kunt u gebruik maken van de links aan de bovenkant van de pagina. Zo is het mogelijk om een overzicht te krijgen van de verschillende routes in de stad Gent. Ook een weergave op kaart en vergelijking tussen verschillende routes is mogelijk.

11.3 Overview

Bovenaan de pagina “Overview” kan u kiezen uit de tabbladen: “Summary”, “For each provider” en “By route”.



Figuur 13: Handleiding - Overview

11.3.1 Summary

Het tabblad “Summary” geeft een samenvatting van de verschillende routes in beide richtingen. De belangrijkste data zoals naam (Name), afstand (Distance), standaard reisduur (Standard Travel Time),

huidige tijdsduur (Current Travel Time) en vertraging (Delay) worden weergegeven. Het is mogelijk te sorteren op een bepaald onderwerp door op de hoofding van de tabel te klikken.

Afhankelijk van de opgelopen vertraging wordt een kleur toegekend aan elk traject. De kleur aanduiding varieert tussen groen (geen file) en rood (zwarte file).

De trajecten worden in beide richtingen weergegeven. Dit wordt aangegeven door de windrichting waarin deze gelegen zijn. (East - West - North - South bound). Bij het klikken op een traject komt u terecht op de detail pagina voor dat traject.

Summary For each provider By route					
Name	Distance	Standard Travel Time	Current Travel Time	Delay	
Rooigemlaan (R40) northbound	1355.0 m	03' 02"	03' 04"	00' 34"	Edit
Rooigemlaan (R40) southbound	1346.0 m	02' 57"	02' 42"	00' 16"	Edit
Gasmeterlaan (R40) eastbound	2229.0 m	04' 31"	05' 03"	00' 48"	Edit
Gasmeterlaan (R40) westbound	2191.0 m	03' 52"	04' 31"	00' 30"	Edit
Dok-Noord (R40) southbound	1796.0 m	04' 49"	04' 07"	00' 31"	Edit
Dok-Noord (R40) northbound	1876.0 m	03' 35"	04' 14"	00' 41"	Edit
Blaisantvest (N430) eastbound	2672.0 m	06' 03"	09' 32"	03' 08"	Edit
Blaisantvest (N430) westbound	2693.0 m	05' 12"	06' 18"	00' 41"	Edit
Keizer Karelstraat northbound	4269.0 m	08' 57"	10' 31"	02' 44"	Edit
Keizer Karelstraat southbound	4117.0 m	08' 54"	09' 19"	01' 21"	Edit
Kennedylaan (R4) southbound	15810.0 m	24' 47"	20' 59"	06' 47"	Edit

Figuur 14: Handleiding - Summary

Wijzigingen aan de routes kunnen gemaakt worden door bij een bepaalde route op Edit te klikken. Dit wordt hieronder verder beschreven.

Summary For each provider By route					
Name	Distance	Standard Travel Time	Current Travel Time	Delay	
Rooigemlaan (R40) southbound	1346.0 m	02' 57"	02' 42"	00' 16"	Edit

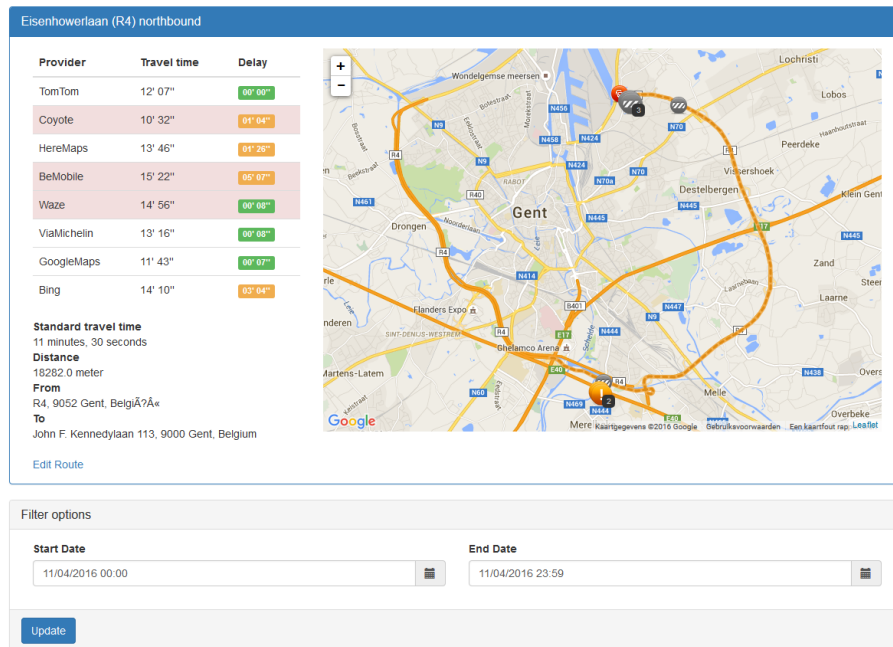
Figuur 15: Handleiding - Zoom in op Edit

11.3.2 Detail per traject

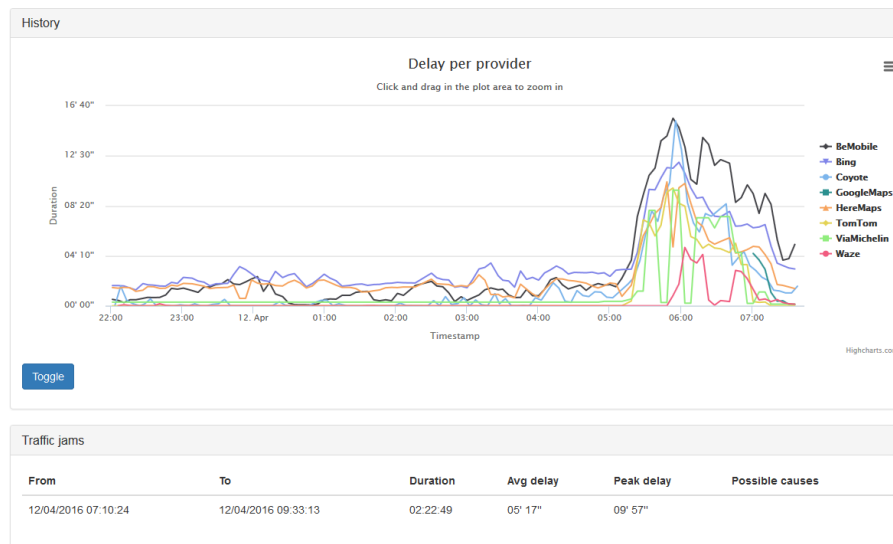
Op deze pagina vindt u de gedetailleerde informatie over het gekozen traject.

- Bovenaan worden de reistijden en vertragingen per provider weergegeven. Aan de rechterkant staat het traject afgebeeld op een kaart met een visuele weergave van hoe het verkeer verloopt op dit traject.
- Bij **Filteroptions** kan de begin- en einddatum geselecteerd worden als parameters voor het genereren van een grafiek.

- In **History** wordt de grafiek gegenereerd die de tijdsduur, om dit traject af te leggen, grafisch voorstelt tussen het interval ingesteld bij filteroptions.
- **Traffic jams** geeft alle files die zijn opgevangen in dat bepaald interval. Zo wordt er weergegeven hoe lang de file heeft geduurd (**Duration**), wat de gemiddelde vertraging (**Avg delay**) en piek vertraging (**Peak delay**) was en de mogelijke oorzaken (**Possible causes**) van de file. De mogelijke oorzaken kunnen invloeden zijn van ongevallen, werken of het weer.



Figuur 16: Handleiding - Detail (1/2)



Figuur 17: Handleiding - Detail (2/2)

11.3.3 Edit

Op het tabblad "**Summary**" kunnen de trajecten aangepast worden via de knop **Edit**. Hier kunnen de **name**, **from**- en **to** positions van het traject aangepast worden. De start- en eindpositie kunnen op 2 manieren aangepast worden, door de positie in te geven in het tekstveld of door op het kaartje de rode pijl te verslepen via de rechtermuisknop. Klik na de aanpassingen onderaan op **Save**.

Figuur 18: Handleiding - Aanpassen Route

Opgelet: Als je een route wenst te wijzigen dan **MOET** je beide kaarten eens beklikken. Ook als je enkel de “to” wenst te wijzigen dan nog moet je de “from” op de kaart eens wijzigen. Het veiligste is om je aanduidpunt eens te verplaatsen en opnieuw te zetten. Enkel dan worden de waypoints correct opgeslagen.

11.3.4 For each provider









Op het tabblad “**For each provider**” worden de reistijd (**CTT, Current Travel Time**) en de vertraging (**D, Delay**) per traject en per verkeersprovider weergegeven. Via het tandwiel kan er worden om bepaalde providers tijdelijk uit te schakelen in het overzicht.

Betekenis van de kolommen

- **Distance** geeft de afstand van het traject.
- **Current Travel Time** wordt berekend door het gemiddelde te nemen over de huidige reistijden van de verschillende providers.
- **Avg. Delay** geeft de gemiddelde vertraging voor alle providers.
- **CCT** geeft de huidige reistijd per provider weer.
- **D** geeft de huidige vertraging per provider weer.

Route Overview

Most recent data from 2016-04-13 13:58:06.0

Summary	For each provider	By route																	
<div><div><div></div><div></div></div><div>Name</div></div>																			
Distance	Current Travel Time	Avg. Delay	CTT		D		D		D		D		D		D		D		D
Roosijemlaan (R40) northbound	1355.0 m	02:56"	00:23"	02:56"	00:00"	02:33"	00:00"	02:31"	00:04"	02:58"	00:40"	02:15"	00:38"	02:39"	00:55"	02:58"	00:50"	04:38"	00:00"
Roosijemlaan (R40) southbound	1346.0 m	02:52"	00:23"	03:21"	00:00"	02:32"	00:00"	02:53"	00:15"	03:36"	01:23"	00:44"	00:00"	01:47"	00:00"	03:36"	01:31"	04:31"	00:00"
Gasmeterlaan (R40) eastbound	2229.0 m	04:59"	00:46"	05:14"	00:00"	04:28"	00:11"	05:36"	01:04"	06:07"	01:39"	01:24"	00:00"	03:53"	01:10"	06:17"	02:09"	06:59"	00:00"
Gasmeterlaan (R40) westbound	2191.0 m	05:13"	00:57"	04:48"	00:00"	04:17"	00:00"	05:48"	01:09"	05:09"	01:24"	02:51"	00:00"	04:22"	01:44"	06:12"	01:41"	08:24"	01:44"
Dok-Noord (R40) southbound	1796.0 m	04:46"	00:59"	04:27"	00:00"	04:27"	00:00"	05:22"	01:11"	05:19"	01:51"	02:57"	00:52"	04:06"	01:54"	05:51"	02:05"	05:43"	00:00"
Dok-Noord (R40) northbound	1876.0 m	04:23"	00:43"	04:05"	00:02"	03:11"	00:00"	04:08"	00:07"	04:44"	01:06"	03:21"	01:24"	04:16"	01:41"	05:36"	01:27"	05:44"	00:00"
Blaisantvest (N430) eastbound	2672.0 m	06:26"	00:56"	05:48"	00:00"	04:51"	00:00"	07:33"	00:33"	05:34"	00:02"	04:18"	01:25"	06:21"	02:54"	08:06"	02:41"	08:58"	00:00"
Blaisantvest (N430) westbound	2693.0 m	06:32"	00:44"	06:09"	00:00"	06:12"	00:00"	08:04"	00:18"	05:31"	00:00"	03:19"	01:10"	05:12"	01:51"	08:16"	02:34"	09:34"	00:00"
Keizer Karelstraat northbound	4269.0 m	10:35"	02:43"	10:08"	02:06"	09:01"	00:00"	11:36"	02:16"	10:47"	02:53"	05:31"	02:34"	13:11"	08:28"	11:47"	03:33"	12:39"	00:00"
Keizer Karelstraat southbound	4117.0 m	09:47"	01:35"	10:29"	00:00"	08:43"	00:00"	12:57"	01:45"	10:49"	02:56"	01:56"	00:00"	09:06"	04:06"	11:29"	03:54"	12:54"	00:00"
Kennedylaan (R4) southbound	15810.0 m	22:10"	06:58"	24:48"	02:35"	19:29"	00:01"	20:21"	02:11"	24:08"	10:34"	18:03"	08:11"	22:42"	11:19"	23:09"	10:09"	24:40"	10:45"
Kennedylaan (R4) northbound	16938.0 m	20:51"	05:49"	21:44"	04:14"	17:11"	00:00"	24:50"	03:12"	21:26"	08:08"	15:56"	06:10"	20:50"	09:31"	22:03"	07:54"	22:49"	07:24"
Binnenring-Oranien (R4) northbound	10987.0 m	07:44"	00:01"	07:31"	00:00"	06:53"	00:03"	07:15"	00:04"	07:58"	00:01"	07:21"	00:14"	07:30"	00:11"	07:52"	00:00"	09:33"	00:00"

Figuur 19: Handleiding - For Each Provider

11.3.5 By Route

Op het tabblad “By route” wordt dezelfde data weergegeven zoals bij “For each provider”, maar hier wordt de data gesorteerd per route.

Route Overview

Most recent data from 2016-04-13 13:58:06.0

Summary	For each provider	By route
Roosendaal (R40) northbound <ul style="list-style-type: none">Distance: 1355.0 mStandard Travel Time: 182.0Average Delay: 176.0TomTom CTT: 176.0TomTom Delay: 0.0Waze CTT: 153.0Waze Delay: 0.0GoogleMaps CTT: 151.0GoogleMaps Delay: 4.0HereMaps CTT: 178.0HereMaps Delay: 40.0Coyote CTT: 135.0Coyote Delay: 38.0BeMobile CTT: 159.0BeMobile Delay: 55.0BingMaps CTT: 178.0BingMaps Delay: 50.0ViaMichelin CTT: 278.0ViaMichelin Delay: 0.0	Roosendaal (R40) southbound <ul style="list-style-type: none">Distance: 1346.0 mStandard Travel Time: 177.0Average Delay: 172.5TomTom CTT: 201.0TomTom Delay: 0.0Waze CTT: 152.0Waze Delay: 0.0GoogleMaps CTT: 173.0GoogleMaps Delay: 15.0HereMaps CTT: 216.0HereMaps Delay: 83.0Coyote CTT: 44.0Coyote Delay: 0.0BeMobile CTT: 107.0BeMobile Delay: 0.0BingMaps CTT: 216.0BingMaps Delay: 91.0ViaMichelin CTT: 271.0ViaMichelin Delay: 0.0	Gasmeterlaan (R40) eastbound <ul style="list-style-type: none">Distance: 2229.0 mStandard Travel Time: 271.0Average Delay: 299.75TomTom CTT: 314.0TomTom Delay: 0.0Waze CTT: 268.0Waze Delay: 11.0GoogleMaps CTT: 336.0GoogleMaps Delay: 64.0HereMaps CTT: 367.0HereMaps Delay: 99.0Coyote CTT: 84.0Coyote Delay: 0.0BeMobile CTT: 233.0BeMobile Delay: 70.0BingMaps CTT: 377.0BingMaps Delay: 129.0ViaMichelin CTT: 419.0ViaMichelin Delay: 0.0
Dok-Noord (R40) southbound <ul style="list-style-type: none">Distance: 1796.0 mStandard Travel Time: 289.0Average Delay: 286.5TomTom CTT: 267.0TomTom Delay: 0.0Waze CTT: 267.0Waze Delay: 0.0GoogleMaps CTT: 322.0GoogleMaps Delay: 71.0HereMaps CTT: 319.0HereMaps Delay: 111.0Coyote CTT: 177.0Coyote Delay: 52.0BeMobile CTT: 246.0BeMobile Delay: 114.0BingMaps CTT: 351.0BingMaps Delay: 125.0ViaMichelin CTT: 343.0ViaMichelin Delay: 0.0	Dok-Noord (R40) northbound <ul style="list-style-type: none">Distance: 1876.0 mStandard Travel Time: 215.0Average Delay: 263.125TomTom CTT: 245.0TomTom Delay: 2.0Waze CTT: 191.0Waze Delay: 0.0GoogleMaps CTT: 248.0GoogleMaps Delay: 7.0HereMaps CTT: 284.0HereMaps Delay: 66.0Coyote CTT: 201.0Coyote Delay: 84.0BeMobile CTT: 256.0BeMobile Delay: 101.0BingMaps CTT: 336.0BingMaps Delay: 87.0ViaMichelin CTT: 344.0ViaMichelin Delay: 0.0	Blaisantvest (N430) eastbound <ul style="list-style-type: none">Distance: 2672.0 mStandard Travel Time: 363.0Average Delay: 386.125TomTom CTT: 348.0TomTom Delay: 0.0Waze CTT: 291.0Waze Delay: 0.0GoogleMaps CTT: 453.0GoogleMaps Delay: 33.0HereMaps CTT: 334.0HereMaps Delay: 2.0Coyote CTT: 258.0Coyote Delay: 85.0BeMobile CTT: 381.0BeMobile Delay: 174.0BingMaps CTT: 486.0BingMaps Delay: 161.0ViaMichelin CTT: 538.0ViaMichelin Delay: 0.0

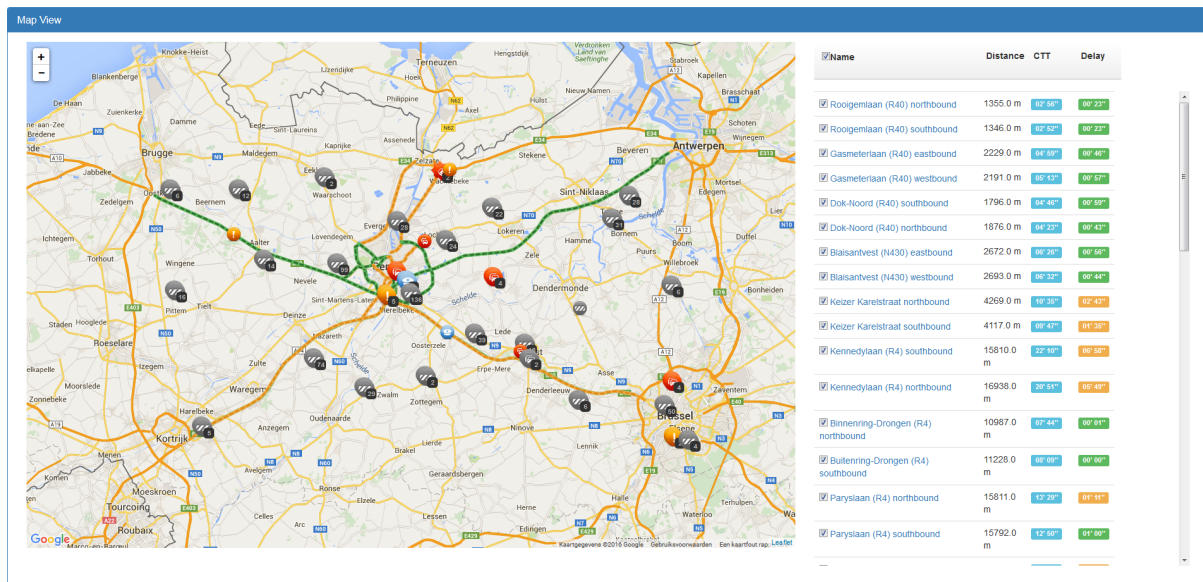
Figuur 20: Handleiding - By Route

11.4 Map View

Op de pagina “Map View” worden alle trajecten afgebeeld op een kaart. Afhankelijk van de opgelopen vertraging wordt een kleur toegekend aan elk traject. Het traject wordt aangeduid met een bewegende route waardoor het duidelijk wordt in welke richting het verkeer zich beweegt. Door een traject af te vinken wordt het niet meer weergegeven op de kaart.

Het is mogelijk om door te klikken op de gegroepeerde POI's om in de regio waar ze van toepassing zijn in te zoomen. Dit geeft een duidelijker beeld waar de POI's zich nu juist bevinden. Door op een niet gegroepeerde POI te klikken kunt u meer informatie opvragen.

Ook kunt u doorklikken op een traject op de kaart om meer informatie te bekomen en rechtstreeks te kunnen doorklikken naar de detail pagina van het traject.



Figuur 21: Handleiding - Map View

11.5 Logs

Op de pagina "Logs" kunnen de logberichten van deze webapplicatie opgevraagd worden. Via het formulier kan een categorie en begin- en eindtijdstip voor de logberichten gekozen worden. Bij het klikken op "Update" worden de logberichten weergegeven. Dit zijn maximaal 500 logberichten per keer.

Filter Logs

Select category

verkeerdomain.POIService

Start Date

03/02/2016 00:00

End Date

12/05/2016 23:59

Update

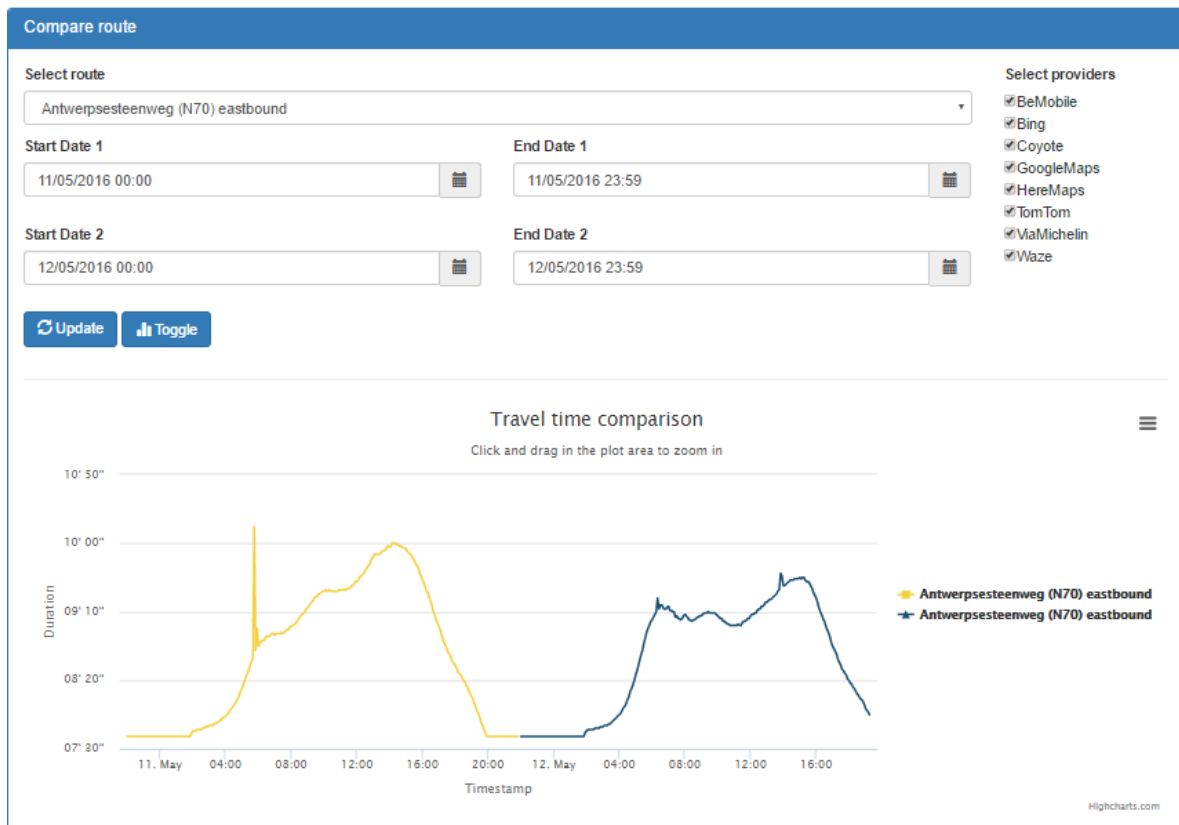
Maximum 500 entries.

Type	Date	Time	Message
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221770-47' for key 'PRIMARY'
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221770-48' for key 'PRIMARY'
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221772-47' for key 'PRIMARY'
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221772-48' for key 'PRIMARY'
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221774-49' for key 'PRIMARY'
Error	2016/04/18	19:21:08	Error in executeUpdate, Duplicate entry '221774-50' for key 'PRIMARY'
Error	2016/04/18	19:26:05	Error in executeUpdate, Duplicate entry '221770-47' for key 'PRIMARY'
Error	2016/04/18	19:26:05	Error in executeUpdate, Duplicate entry '221770-48' for key 'PRIMARY'
Error	2016/04/18	19:26:05	Error in executeUpdate, Duplicate entry '221772-47' for key 'PRIMARY'

Figuur 22: Handleiding - Logs

11.6 Compare Route

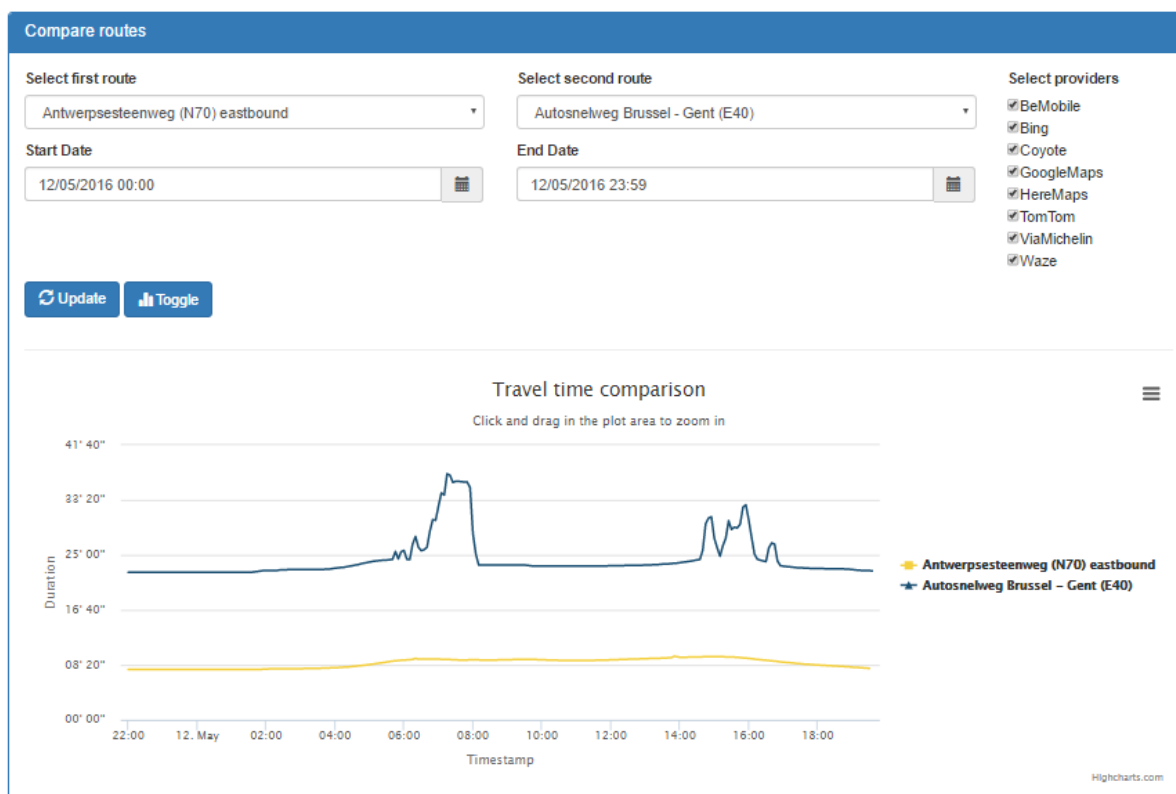
Op de pagina "Compare Route" is het mogelijk om de reistijden op een route op twee verschillende tijdstippen met elkaar te vergelijken. Via het formulier kunnen de route, begin- en eindtijdstip 1 en begin- en eindtijdstip 2 gekozen worden. Daarnaast is het ook mogelijk om bepaalde dataproviders in of uit te schakelen. Via de knop "update" wordt de vergelijkende grafiek gegenereerd. Via de knop "Toggle" kan er gewisseld worden tussen de reistijd en de vertraging. Rechtsboven de grafiek kan via het icoontje met de 3 horizontale streepjes de grafiek geëxporteerd worden naar verschillende bestandsindelingen.



Figuur 23: Handleiding - Compare Route

11.7 Compare Routes

Op de pagina "Compare Routes" is het mogelijk om de reistijden van twee verschillende routes op hetzelfde tijdstip met elkaar te vergelijken. Via het formulier kunnen de routes en het begin- en eindtijdstip gekozen worden. Daarnaast is het ook mogelijk om bepaalde dataproviders in of uit te schakelen. Via de knop "update" wordt de vergelijkende grafiek gegenereerd. Via de knop "Toggle" kan er gewisseld worden tussen de reistijd en de vertraging. Rechtsboven de grafiek kan via het icoontje met de 3 horizontale streepjes de grafiek geëxporteerd worden naar verschillende bestandsindelingen.



Figuur 24: Handleiding - Compare Routes

12 Installatiehandleiding

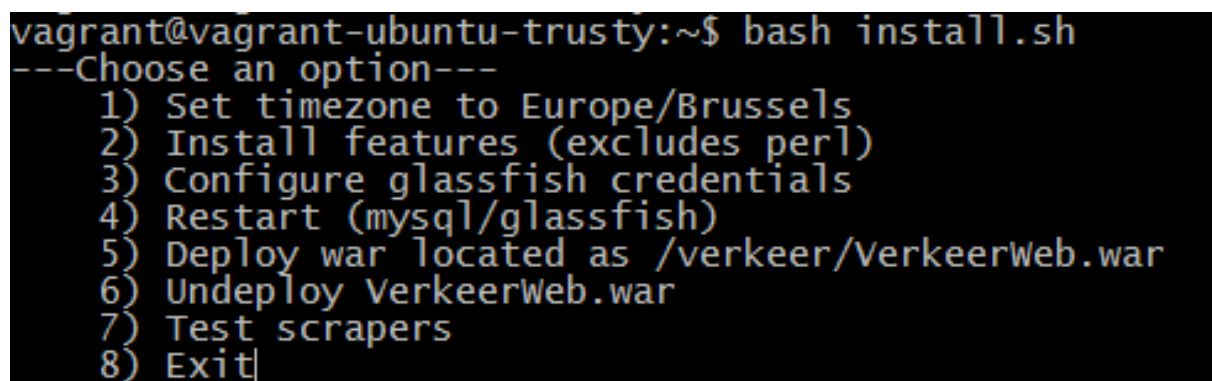
12.1 Software installeren (via install.sh)

In deze sectie wordt de software geïnstalleerd via een installatiescript. De volgende sectie toont aan hoe alles manueel kan gebeuren.

Er is een installatiescript voorzien (te vinden als applicatie/scripts/install.sh) waarmee het meeste automatisch opgezet kan worden.

Het script heeft twee grote functionaliteiten. Enerzijds het installeren van de benodigde software en anderzijds het deployen van de webapplicatie.

Er zijn echter een aantal zaken die nog steeds manueel moeten gebeuren. Volgende reeks stappen geeft een overzicht van de nodige instructies. De manueel uit te voeren commando's staan eronder opgesomd (alsook in volgende sectie).



```
vagrant@vagrant-ubuntu-trusty:~$ bash install.sh
---Choose an option---
 1) Set timezone to Europe/Brussels
 2) Install features (excludes perl)
 3) Configure glassfish credentials
 4) Restart (mysql/glassfish)
 5) Deploy war located as /verkeer/VerkeerWeb.war
 6) Undeploy VerkeerWeb.war
 7) Test scrapers
 8) Exit
```

Figuur 25: Install - Script

- Configureer de juiste tijdzone (optie 1)
- Installeer meeste features (java, glassfish, mysql, ...) (kies voor optie 2 in installatiescript)
- Configureer glassfish (optie 3) (er zal gevraagd worden naar admin credentials).
- (Optioneel:) Restart mysql en glassfish (optie 4)
- Installeer perl manueel
- Configureer en importeer de databank manueel
- Deploy met installatiescript (optie 5)
- (Optioneel:) PollService opzetten
- (Optioneel:) Testen van scrapers (optie 7)
- (Optioneel:) Backup opzetten

12.1.1 Perl installeren

```
1 sudo cpan JSON
2 perl -MCPAN -e shell
3 > install JSON::XS
```

12.1.2 Mysql database verkeer in orde brengen

```
1      mysql -p
2          create database verkeer;
3          exit;
4      mysql verkeer -u root < "verkeer.sql"
```

12.1.3 Poll service in een scriptje

De poll service staat in dit voorbeeld in de root: /root/verkeerPollService.jar. Met volgend script (runpollservice.sh) zorgen we ervoor dat hij automatisch herstart bij crashen:

```
1  #!/bin/bash
2
3
4  cleanup ()
5  {
6      kill -s SIGTERM $pid
7      exit 0
8  }
9
10 trap cleanup SIGINT SIGTERM
11
12 while true
13 do
14     java -Xms256m -jar VerkeerPollService-1.0-SNAPSHOT.jar &
15     pid=$!
16     #restart after 24 hours to prevent memory leak
17     sleep 86400
18     echo "Killing $pid"
19     kill -9 $pid
20 done
```

Het starten van de poll service gebeurt best in een screen zodat je de sessie kan afsluiten terwijl de service blijft lopen. Nadien kan je de screen sessie restoren alsof de sessie nooit was afgesloten. In tegenstelling tot een achtergrond taak is het hierdoor mogelijk om wel nog de output te zien:

```
1      screen -S pollservice
2      ./runpollservice.sh
```

Met CTRL+A,D kan je het detachen, later kan je met 'screen -r' de sessie terug oproepen.

12.1.4 Instellen backup

Volgende commando's voeren een script uit om een backup te maken aan de hand van een crontab.

```
1      crontab -e
2
3      #backup mysql db every midnight
4      0 0 * * * /root/backupverkeer.sh
```

Het backupverkeer.sh script bevat de volgende lijn:

```
1      mysqldump -u root --proot --routines verkeer | gzip > /root/verkeer.gz
```

Elke nacht zal dus de databank gedumped en gezippt worden naar het /root/verkeer.gz bestand.

12.2 Software manueel installeren

Volgende handleiding heeft de software gedeployed op een ubuntu distributie. Elk systeem (of distributie) die Glassfish, Java, Mariadb (of MySQL) en perl ondersteunt, kan gebruikt worden. De uitgevoerde commando's zullen echter verschillen. Werk je bijvoorbeeld op een centos of een fedora dan zal de apt-get packet manager niet werken en zal je yum moeten gebruiken.

Volgende zaken heb je nodig vanop je client systeem bij aanvang van de installatie:

- Terminal/console/putty/git bash die kan ssh'n naar de server.
- Database dump in de vorm van een SQL-bestand (hier: verkeer.sql).
- De pollservice die uitgevoerd wordt op de server (hier: verkeerPollService.jar)
- De scrapers (hier: de map /scrapers/)
- De webapplicatie die gedeployed wordt op glassfish (hier: verkeerweb.war)

Alle commando's die opgesomd zijn, worden verondersteld uitgevoerd te worden op de ubuntu-server! Het SQL-bestand, de pollservice en de map /root/scrapers/ moeten op de server overgezet worden. Dit kan bijvoorbeeld via sftp of scp.

Een voorbeeld is:

```
1 scp verkeer.sql root@x.y.z.q:/verkeer
2 scp pollservice.jar root@x.y.z.q:/verkeer
3 scp -r scrapers root@x.y.z.q:/verkeer/scrapers
```

12.2.1 Datum/uur/tijdzone instellen

```
1 sudo timedatectl set--timezone Europe/Brussels
```

Een andere optie is via:

```
1 apt-get install ntpdate
2 dpkg-reconfigure tzdata
```

12.2.2 Java 8 installeren

```
1 sudo add-apt-repository ppa:openjdk-r/ppa
2 sudo apt-get update
3 sudo apt-get install openjdk-8-jdk -y
```

Controleren kan via:

```
1 java -version
```

12.2.3 Mariadb installeren

```
1 sudo apt-get install mariadb-server -y
```

Er zal gevraagd worden om een wachtwoord te kiezen voor mysql/mariadb.

12.2.4 Extra tools installeren

```
1 sudo apt-get install unzip gcc make libjson-perl -y
```

Unzip gebruiken we om straks te unzippen. De c compiler, make hebben we nodig om de nodige zaken voor de scrapers te installeren.

12.2.5 Glassfish installeren

```
1 wget http://download.oracle.com/glassfish/4.1/release/glassfish-4.1.zip
2 unzip glassfish-4.1.1.zip -d /opt
3 /opt/glassfish4/bin/asadmin start-domain
4 /opt/glassfish4/bin/asadmin change-admin-password
5 /opt/glassfish4/bin/asadmin enable-secure-admin
6 /opt/glassfish4/bin/asadmin restart-domain
```

Met deze commando's doorloop je enkele stappen om een admin gebruiker te maken en het lege wachtwoord te wijzigen. Indien je vaak het asadmin programma gaat gebruiken dan kan het handig zijn om /opt/glassfish4/bin op te nemen in je PATH en dit toe te voegen aan je /.profile. Om een domein te stoppen gebruik je het commando "asadmin stop-domain".

12.2.6 Perl installeren

```
1 sudo cpan JSON
2 perl -MCPAN -e shell
3 > install JSON::XS
```

12.2.7 Mysql database verkeer in orde brengen

```
1 mysql -p
2 create database verkeer;
3 exit;
4 mysql verkeer -u root < "verkeer.sql"
```

12.2.8 Scrapers testen

```
1 root@server:~/scrapers# ./testscrapers.sh
```

Met dit scriptje worden alle scrapers getest. Op deze manier zien we of er data binnengehaald kan worden of niet.

12.2.9 Poll service in een scriptje

De poll service staat in dit voorbeeld in de root: /root/verkeerPollService.jar. Met volgend script (runpollservice.sh) zorgen we ervoor dat hij automatisch herstart bij crashen:

```

1  #!/bin/bash
2
3
4  cleanup ()
5  {
6      kill -s SIGTERM $pid
7      exit 0
8  }
9
10 trap cleanup SIGINT SIGTERM
11
12 while true
13 do
14     java -Xms256m -jar VerkeerPollService-1.0-SNAPSHOT.jar &
15     pid=$!
16     #restart after 24 hours to prevent memory leak
17     sleep 86400
18     echo "Killing_$pid"
19     kill -9 $pid
20 done

```

Het starten van de poll service gebeurt best in een screen zodat je de sessie kan afsluiten terwijl de service blijft lopen. Nadien kan je de screen sessie restoren alsof de sessie nooit was afgesloten. In tegenstelling tot een achtergrond taak is het hierdoor mogelijk om wel nog de output te zien:

```

1  screen -S pollservice
2  ./runpollservice.sh

```

Met CTRL+A,D kan je het detachen, later kan je met 'screen -r' de sessie terug oproepen.

12.2.10 Instellen backup

Volgende commando's voeren een script uit om een backup te maken aan de hand van een crontab.

```

1  crontab -e
2
3  #backup mysql db every midnight
4  0 0 * * * /root/backupverkeer.sh

```

Het backupverkeer.sh script bevat de volgende lijn:

```

1  mysqldump -u root --proot --routines verkeer | gzip > /root/verkeer.gz

```

Elke nacht zal dus de databank gedumped en gezippt worden naar het /root/verkeer.gz bestand.

12.2.11 Uploaden Verkeerweb

Dit gebeurt in de browser door te surfen naar glassfish. Deze is bereikbaar via <http://server:4848>. Inloggen doe je met de credentials die eerder werden ingesteld bij glassfish. Daarna kies je voor deploy application en kies je de war:

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

Location: ☒ **Packaged File to Be Uploaded to the Server**

VerkeerWeb.war

☐ **Local Packaged File or Directory That Is Accessible from GlassFish Server**

Type: * ▼

Context Root:
Path relative to server's base URL.

Application Name: *

Virtual Servers:
Associates an Internet domain name with a physical server.

Status: ☒ **Enabled**
Allows users to access the application.

Implicit CDI ☒ **Enabled**
Implicit discovery of CDI beans

Precompile JSPs: ☒
Precompiles JSP pages during deployment.

Run Verifier: ☐
Verifies the syntax and semantics of the deployment descriptor. Verifier packages must be installed.

Force Redeploy: ☒
Forces redeployment even if this application has already been deployed or already exists.

Keep State: ☐
Retains web sessions, SFSB instances, and persistently created EJB timers between redeployments.

Deployment Order:
A number that determines the loading order of the application at server startup. Lower numbers are loaded first.

Libraries:
A comma-separated list of library JAR files. Specify the library JAR files by their relative or absolute paths. Spec

Description:

Figuur 26: Uploaden Verkeerweb

12.3 Documentatie Scrapers

De applicatie draait in een oneindige lus en zorgt ervoor dat elke 5min de poll methode wordt opgeroepen. In de poll service worden alle routes overlopen en worden gegevens van alle providers tegelijk (in parallel) opgevraagd. Hierdoor is het mogelijk om voor alle routes en providers steeds de reistijden binnen de 5 minuten op te vragen.

Er zijn 2 soorten providers om de route gegevens te bepalen:

- Providers van ISummaryProvider zoals Coyote geven alle routes in één keer terug waardoor de scraper maar éénmaal om de 5 minuten moet opgeroepen worden.
- Andere Providers van IProvider vragen gegevens op per route. Bijna alle providers maken onderliggend gebruik van perl scripts die met curl (commandline) de gegevens opvragen. Enkel bij Google Maps of Bing Maps is er geen scraper voorzien. Bij andere verschillende providers zoals TomTom en Here Maps is er een fallback naar de API met API keys die moeten geconfigureerd worden in de application.conf.

De perl scripts staan onder /root/scrapers in productie² en maken gebruik van het curl programma om de requests door te voeren. Bij sommige scrapers zoals TomTom en Here maps wordt eerst de API Key uit de html pagina gevist om daarna de json request te versturen. Deze api key wordt een uur gecached in een .cache file zodat dit niet steeds bij elke request moet opnieuw verzameld worden. Als de provider zijn api key wijzigt aan de front end wordt dit dus binnen een uur gedetecteerd en opgevangen.

In veel gevallen is de json die geparsed wordt om de route gegevens eruit te halen klein genoeg om met eenvoudige reguliere expressies te parsen. Bij Waze moet er een som gemaakt worden van de segmenten en hebben we de JSON::XS Perl library gebruikt.

Requirements:

De perl scripts moeten met een perl interpreter kunnen uitgevoerd worden. Onder windows kan dit met ActivePerl, in een linux omgeving is perl veelal voorgeïnstalleerd als /usr/bin/perl.

Curl is meestal reeds geïnstalleerd op linux, mocht dit nog niet zo zijn kan je curl via de package manager installeren. Onder windows is er een curl.exe die bij de perl scripts staat die gebruikt wordt.

Opgelet: Voor de json library in perl te installeren zijn volgende commands nodig:

```
> sudo cpan JSON
> perl -MCPAN -e shell
> install JSON::XS
```

²Het pad kan ook worden aangepast in het application.conf configuratie bestand

13 Productbacklog

13.1 Productbacklog - Einde sprint 1

Use case	Must have	Nice to have	Weging (uur)
Verzamel reistijdgegevens	1		55
Bekijk trajectoverzicht	2		25
Bekijk trajectdetail	2		50
Bekijk trajectmap	3		30
Vergelijk providerdata	4		40
Wijzig traject		12	30
Voeg traject toe		14	40
Verwijder traject		14	40
Bekijk statuslogpagina		12	20
Bekijk infopagina's		13	40
Bekijk dashboard		13	15
Aanbieden trajectgegevens met reistijden		11	45
Verzamel POI-gegevens		16	20
Verzamel Weergegevens		16	20
Verzamel Parkeer & bord gegevens		16	20

Tabel 27: Productbacklog - Einde sprint 1

13.2 Productbacklog - Einde sprint 2

Use case	Must have	Nice to have	Weging (uur)
Verzamel reistijdgegevens	1		55
Bekijk routeoverzicht	2		25
Bekijk routedetail	2		50
Bekijk routemap	3		30
Vergelijk providerdata	4		40
Wijzig traject		12	30
Bekijk logpagina		12	20
Bekijk dashboard		13	15
Aanbieden trajectgegevens met reistijden		11	45
Verzamel POI-gegevens		16	20
Verzamel Weergegevens		16	20
Vergelijk routes		16	15

Tabel 28: Productbacklog - Einde sprint 2

13.3 Productbacklog - Einde sprint 3

Use case	Must have	Nice to have	Weging (uur)
Verzamel reistijdgegevens	1		55
Bekijk routeoverzicht	2		25
Bekijk routedetail	2		50
Bekijk routemap	3		30
Vergelijk providerdata	4		40
Wijzig traject		12	30
Bekijk logpagina		12	20
Bekijk dashboard		13	15
Aanbieden trajectgegevens met reistijden		11	45
Verzamel POI-gegevens		16	20
Verzamel Weergegevens		16	20
Vergelijk routes		16	15
Vergelijk route		16	10

Tabel 29: Productbacklog - Einde sprint 3

Lijst van figuren

1	Mockup Dashboard	24
2	Mockup Logging	24
3	Use Case Diagram	25
4	Use Case Diagram - Einde Sprint 1	25
5	Use Case Diagram - Einde Sprint 2	26
6	Diagram - Projectstructuur	27
7	Diagram - ERD	28
8	Diagram - Bepalen File en Oorzaak	29
9	Diagram - Polling verkeersgegevens	30
10	Diagram - Overview Routes	31
11	Apache JMeter demo	42
12	Handleiding - Dashboard	48
13	Handleiding - Overview	48
14	Handleiding - Summary	49
15	Handleiding - Zoom in op Edit	49
16	Handleiding - Detail (1/2)	50
17	Handleiding - Detail (2/2)	50
18	Handleiding - Aanpassen Route	51
19	Handleiding - For Each Provider	52
20	Handleiding - By Route	52
21	Handleiding - Map View	53
22	Handleiding - Logs	54
23	Handleiding - Compare Route	55
24	Handleiding - Compare Routes	56
25	Install - Script	57
26	Uploaden Verkeerweb	62

Lijst van tabellen

1	Taakverdeling sprint 1	7
2	Taakverdeling sprint 2	8
3	Taakverdeling sprint 3	8
4	UC: Verzamel reistijdgegevens	10
5	UC: Bekijk routeoverzicht	11
6	UC: Bekijk routedetail	12
7	UC: Bekijk routemap	13
8	UC: Vergelijk providerdata	14
9	UC: Wijzig route	15
10	UC: Bekijk logpagina	16
11	UC: Bekijk dashboard	17
12	UC: Aanbieden routegegevens met reistijden	18
13	UC: Verzamelen POI-gegevens	19
14	UC: Verzamelen weergegevens	20
15	UC: Vergelijk routes	21
21	Usability Tests (use case testing (1/2))	36
22	Usability Tests (use case testing (2/2))	37
23	Scenario Usability Test Template	38
24	Scenario Usability Test Persoon 1: IT-deskundige	39
25	Scenario Usability Test Persoon 2: niet-IT-deskundige (1)	40
26	Scenario Usability Test Persoon 3: niet-IT-deskundige (2)	41
27	Productbacklog - Einde sprint 1	64
28	Productbacklog - Einde sprint 2	65
29	Productbacklog - Einde sprint 3	66