



3e Bachelor Informatica  
Academiejaar 2015-2016

Faculteit Ingenieurswetenschappen en Architectuur  
Valentin Vaerwijckweg 1 - 9000 Gent

# Verkeerscentrum

Verslag voor bachelorproef (sprint 1)

Groep 2   Mike BRANTS  
              Tobias VAN DER PULST  
              Thomas VANDE WEGHE  
              Simon VERMEERSCH

# Inhoudsopgave

---

<b>Inhoudsopgave</b>	<b>1</b>
<b>1 Behoefteanalyse</b>	<b>2</b>
1.1 Beschrijving project . . . . .	2
1.2 Functionaliteiten . . . . .	2
1.3 Use Cases . . . . .	2
1.4 Functieanalyse van de omgeving . . . . .	2
<b>2 Ontwerp</b>	<b>4</b>
2.1 Functioneel ontwerp . . . . .	4
2.2 Technisch ontwerp . . . . .	4
2.2.1 Hardware . . . . .	4
2.2.2 Paketten . . . . .	4
2.3 Software ontwerp . . . . .	4
2.3.1 Dataproviders . . . . .	4
2.3.2 Structuur van de data (API) . . . . .	4
2.3.3 Databank . . . . .	7
2.3.4 Klassediagram . . . . .	8
2.3.5 Gegevensstroomdiagram . . . . .	8
2.3.6 Bestandstructuur . . . . .	9
2.3.7 Test . . . . .	9

# Hoofdstuk 1: Behoeftanalyse

---

## 1.1 BESCHRIJVING PROJECT

---

Het Mobiliteitsbedrijf van de stad gent is sinds 2014 bezig met het opzetten van een regionaal verkeerscentrum. Het is de bedoeling dat op termijn het verkeer in de regio constant gemonitord wordt, op semi-automatische basis op normale werkdagen en bemand tijdens piekmomenten en evenementen. Tijdens de week is het de bedoeling dat onverwachte incidenten, calamiteiten of significante verhogingen van de reistijden automatisch gesignaleerd worden aan de verantwoordelijke, die dan de nodige acties kan ondernemen. De gegevens zouden ook constant beschikbaar zijn voor het publiek via een website, sociale media en open data. Op die manier kunnen mensen de beste route en het beste moment kiezen om hun verplaatsingen te maken in de regio.

## 1.2 FUNCTIONALITEITEN

---

1. Ophalen van kwalitatieve en vergelijkbare data bij verschillende bronnen
2. Real-time overzicht van de verkeersdruke op vooraf vastgelegde trajecten
3. Analyse op basis van opgehaalde data
4. Aanbieden van gegevens aan externen via REST Api
5. Kwaliteitscontrole van de verkregen data
6. Platform gelinkt met sociale media om snelle communicatie aan te bieden
7. Meldingen genereren wanneer reistijden overschreden worden
8. Bepalen van de oorzaak van een vertraging

## 1.3 USE CASES

---

?

## 1.4 FUNCTIEANALYSE VAN DE OMGEVING

---

1. Gebruikers
  - (a) Ontwikkelaar

- (b) Administrator
  - (c) Operator in het verkeerscentrum
2. Doelstellingen
- De doelstellingen representeren onze product backlog en ze bevatten de taken die het systeem moet kunnen.
- (a) Basisfunctionaliteit
    - i. Data ophalen uit meerdere bronnen
      - A. Google Maps
      - B. Here
      - C. Waze
      - D. TomTom
      - E. Coyote
    - ii. Databank creëren en opvullen met opgehaalde data
  - (b) API met verschillende parameters
    - i. Periode
    - ii. Traject
    - iii. Leverancier
    - iv. Vertraging
  - (c) Dashboard
    - i. Grafische opbouw van de GUI
    - ii. Grafieken/Tabellen genereren
    - iii. Grafische weergave op kaart
    - iv. Selecteren parameters gebruik makend van API
  - (d) Kwaliteitscontrole
    - i. Procedures via MySQL
    - ii. Grafieken/Tabellen genereren

# Hoofdstuk 2: Ontwerp

---

## 2.1 FUNCTIONEEL ONTWERP

---

... voeg figuur Use-Case in ...

## 2.2 TECHNISCH ONTWERP

---

### 2.2.1 Hardware

Tijdens de ontwikkelingsfase zal beroep gedaan worden op de servers van de UGent.

### 2.2.2 Paketten

1. Java EE
2. Php MyAdmin

## 2.3 SOFTWARE ONTWERP

---

### 2.3.1 Dataproviders

#### Google Maps

Aantal aanvragen per dag:  
Link naar API:

#### Here

Aantal aanvragen per dag:  
Link naar API:

### 2.3.2 Structuur van de data (API)

De REST API bestaat uit 3 categorieën: trajecten, routedata en providers. Wanneer je hier aanvragen naar doet krijg je gegevens uit de database terug. In de komende secties wordt een korte toelichting gegeven per categorie, gevolgd door een voorbeeld van aanvraag en antwoord in JSON-formaat.

Er is een online API beschikbaar waarin je meer informatie kan terugvinden, hieronder krijg je alvast een kort overzicht.

### Trajecten

De trajecten zijn de routes waar realtime data van opgeroepen wordt. Er kan g ekozen worden om naam, id en/of alle tussenpunten te tonen voor de trajecten, indien je geen parameters meegeeft zullen standaard naam, id en tussenpunten teruggegeven worden. In de toekomst zal het ook mogelijk zijn om trajecten toe te voegen via de API.

---

#### *Patroon*

**GET** ../routes/id/?fields=route.name,route.id,route.geolocations

#### *Voorbeeld*

**GET** ../routes/1,2,3/?fields=route.name,route.id,route.geolocations

---

#### *Voorbeeld*

```
[
  {
    "name": "R4 Zelzate",
    "id": 2,
    "geolocations": [
      {
        "latitude": 51.192226,
        "name": "Zelzate",
        "longitude": 3.776342
      },
      {
        "latitude": 51.086447,
        "name": "Gent",
        "longitude": 3.672188
      }
    ]
  }
]
```

---

### RouteData

De routedata is de opgehaalde data per traject van de verschillende providers op verschillende tijdstippen. Je kan kiezen voor welke routes je data wil terugkrijgen, tevens zal je begin- en eindpunt kunnen instellen. Dit laatste is nuttig om data in een bepaald interval terug te krijgen. In de parameters heb je ook de mogelijkheid om de provider mee te geven, je kan er op deze manier voor zorgen dat enkel data van bepaalde providers getoond wordt.

---

*Patroon*

**GET** ../routes/id/data/timeStart/timeEnd/?fields=route.name,route.id,  
route.geolocations&provider=provider.name

*Voorbeeld*

**GET** ../routes/1,2,3/data/1456761535931/huidig\_tijdstip/?fields=route.name,  
route.id,route.geolocations&provider=GoogleMaps,Here

---

*Voorbeeld*

```
[
  {
    "data": [
      {
        "duration": 753,
        "distance": 14677,
        "provider": "GoogleMaps",
        "timestamp": "Mon Feb 29 19:49:51 CET 2016"
      },
      {
        "duration": 681,
        "distance": 14685,
        "provider": "Here",
        "timestamp": "Mon Feb 29 19:49:51 CET 2016"
      }
    ],
    "name": "R4 Zelzate",
    "id": 2,
    "geolocations": [
      {
        "latitude": 51.192226,
        "name": "Zelzate",
        "longitude": 3.776342
      },
      {
        "latitude": 51.086447,
        "name": "Gent",
        "longitude": 3.672188
      }
    ]
  }
]
```

---

## Providers

Het is mogelijk om via de REST Api alle providers op te vragen, dit kan handig zijn om in bijvoorbeeld RouteData te gebruiken als parameter.

---

*Patroon*

**GET** ../providers

*Voorbeeld*

**GET** ../providers

---

---

```
[  
  "Here",  
  "GoogleMaps"  
]
```

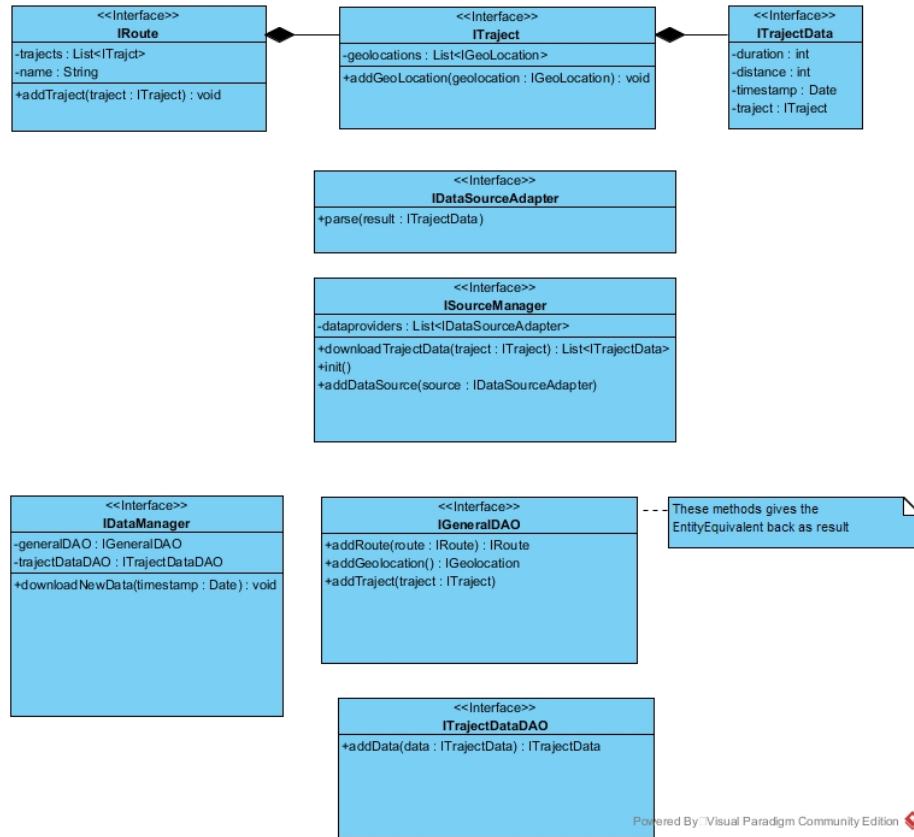
---

### 2.3.3 Databank

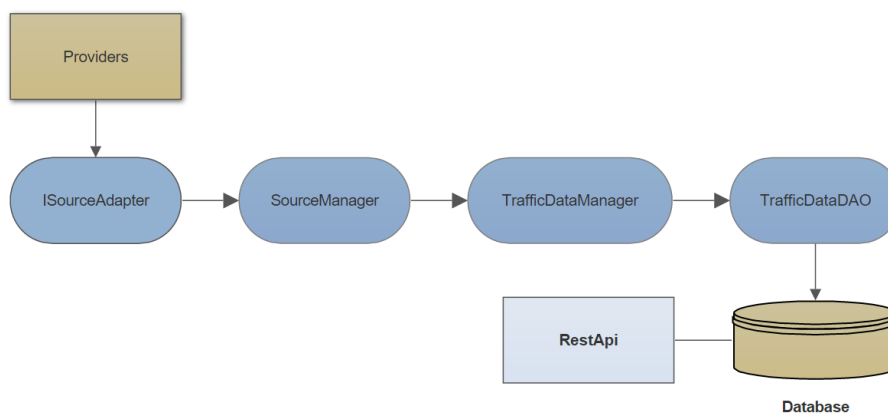
Ik geloof dat er een optie is in SQL Developer om hiervan een mooie representatie te maken zoals: <http://i.stack.imgur.com/Usrhr.png>



### 2.3.4 Klassediagram



### 2.3.5 Gegevensstroomdiagram



De data van de verschillende providers wordt opgehaald met behulp van de SourceAdapters, per provider bestaat er een adapterklasse die de ISourceAdapter

interface implementeert. De verschillende adapters bevinden zich in de SourceManager-klasse. Vanuit de TrafficDataManager wordt data per route aangevraagd, de SourceManager zal deze aanvragen doorsturen aan elke adapter en de ontvangen data teruggeven aan de TrafficDataManager. Om af te sluiten wordt de data doorgegeven aan de TrafficDataDAO, deze klasse zorgt ervoor dat de data in de database terechtkomt. De mogelijkheid bestaat om via Rest-aanvragen data uit de database te halen zoals hierboven reeds beschreven.

### **2.3.6 Bestandstructuur**

Iets over de splittings in beans denk ik dan?

### **2.3.7 Test**

Adapters = Integration Testing  
Enkel db kan goed getest worden?