

ARCHITECTUUR DOCUMENT

TEAM GAMMA (C)

Rekeningrijden

Guus Hamm, Mathijn van Buul, Frank Hartman, Rick Rongen

v0.1
11-04-2017

Inhoudsopgave

1	Wijzigingen	3
2	Inleiding	4
2.1	Context	4
2.2	Applicatie	4
2.3	Niet-functionele eisen	4
2.4	Doel van dit document	4
3	Domein Model	5
3.1	Klassendiagram	5
3.2	Afbakening	5
4	Persistentie	6
5	Communicatie	7
6	Realisatie niet-functionele eisen	8
6.1	Betrouwbaarheid	8
6.2	Performance	8
6.3	Beveiliging	8
6.4	Schaalbaarheid	8
7	Componenten	9
7.1	Componenten Diagram	9
7.2	Koppeling tussen componenten	9
7.3	Synchronisatie tussen componenten	9
7.4	Services per component	9
7.5	Allocatie van objecten	9
7.6	Remote objecten	9
7.7	Packagestructuur	9
8	Deployment	10
8.1	Deploymentdiagram	10
9	Specificatie van interfaces	11

1 Wijzigingen

Herziening	Datum	Auteur(s)	Beschrijving
0.1	11-04-2017	RR	Initieel document

2 Inleiding

2.1 Context

De overheid van Groot-Brittanni wil een nieuw stelsel invoeren waar de automobilisten betalen voor de hoeveelheid die ze rijden in plaats van het bezitten van een auto.

Om dit te verwezenlijken is het team 'Team Gamma' ingeschakeld om dit te realiseren.

2.2 Applicatie

De applicatie bestaat uit meerdere componenten die onderling communiceren. Deze zijn verantwoordelijk voor het rapporteren van de gereden kilometers, het verwerken hiervan en maandelijks een factuur sturen naar de gebruiker.

2.3 Niet-functionele eisen

Het is belangrijk dat minimaal het registratie gedeelte van de applicatie ten alle tijden beschikbaar is. Als dit niet werkt dan kunnen mensen 'gratis' rijden.

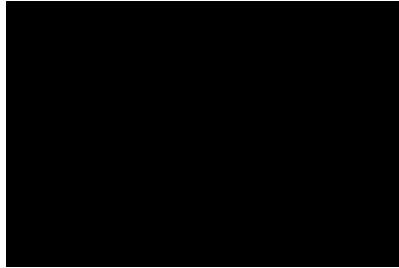
Verder is het belangrijk dat de applicatie accuraat is,

2.4 Doel van dit document

Het doel van dit document is om duidelijk het project weer te geven en alle onderdelen uit te leggen. Dit zodat na het doornemen van dit document het duidelijk is hoe het project in elkaar zit en hoe er aan gewerkt kan worden.

3 Domein Model

3.1 Klassendiagram



Figuur 3.1: Entiteiten Relatie Diagram

3.2 Afbakening

4 Persistentie

De persistentie vindt plaats door middel van een PostgreSQL database. Deze zal in de productie omgeving met meerdere replicaties werken om de prestatie en redundancy te verbeteren, maar in de productie draait het op een enkele database voor de simpliciteit.

De authenticatie maakt gebruik van een OpenLDAP server, deze slaat alleen de authenticatie gegevens van de gebruiker op. De wachtwoorden zijn opgeslagen met het SHA algoritme om te zorgen dat wachtwoorden niet uit te lezen zijn.

5 Communicatie

De communicatie verloopt via een opgesteld HTTP Rest protocol waarmee alle onderdelen beschikbaar zijn. Op deze gegevens rust echter wel een beveiliging zodat alleen de gebruikers die er recht op hebben deze gegevens kunnen inzien. Deze authenticatie verloopt via HTTP Basic authenticatie.

6 Realisatie niet-functionele eisen

6.1 Betrouwbaarheid

6.2 Performance

6.3 Beveiliging

De verbindingen zullen in de productie omgeving beveiligd worden door middel van SSL/TLS. Dit zorgt ervoor dat het praktisch onmogelijk is om als buitenstaander de communicatie af te luisteren.

6.4 Schaalbaarheid

7 Componenten

7.1 Componenten Diagram



Figuur 7.1: Componenten Diagram

7.2 Koppeling tussen componenten

7.3 Synchronisatie tussen componenten

7.4 Services per component

7.5 Allocatie van objecten

7.6 Remote objecten

7.7 Packagestructuur

8 Deployment

8.1 Deploymentdiagram



Figuur 8.1: Deployment Diagram

9 Specificatie van interfaces