Testplan

**Project D groep 2**

**INFPRJ01D**

Mohamad Almarawi (0978573)

Parwesh Bhaggan (1005210)

Darshan Alakhramsing (0985709)

Moesab Ajdid (0987916)

Bas Weidmann (1007701)

29-03-2022

Inhoud

[1. Introductie 2](#_Toc1264039656)

[2. Domein 3](#_Toc2022843528)

[2.1 Unit test 3](#_Toc600558204)

[Testobjecten 3](#_Toc1387824324)

[2.2 Preformance test 3](#_Toc1106283304)

[3. Teststrategie 3](#_Toc1201900585)

[3.1 Unit Testing 3](#_Toc93432442)

[4. Hardware en software eisen 3](#_Toc1162312417)

[5. Test schema 4](#_Toc860547995)

[6. Testscenario 4](#_Toc92985187)

[7. Rollen en verantwoording 7](#_Toc839230518)

[8. Afhankelijkheden 7](#_Toc1277293558)

# Introductie

Dit document is een testplan voor Project D realtime OV. Op het moment bezit gemeente Rotterdam een hoop 3D data van de stad. Hier wordt alleen nog niet veel mee gedaan. Wij hebben nu de taak gekregen om deze 3D data te gebruiken voor een uitbreiding op het ov-vlak. In de applicatie moeten de verschillende OV-voertuigen van het RET visueel rond bewegen. Dit moet allemaal op basis gedaan worden van de daadwerkelijke tijden van de OV-voertuigen.

# Domein

## 2.1 Unit test

### Testobjecten

* Het beeld van de gebruiker kan aangepast worden door het kiezen tussen verschillende camera’s.
* Het systeem moet van specifieke RET-trams data kunnen ophalen die in de omgeving van beurs bewegen.
* De voertuigen moeten op basis van realtime data kunnen bewegen naar station beurs.
* Als de applicatie opgestart wordt is de stand van het licht hetzelfde als buiten.
* Tijdens het gebruiken van de applicatie verandert het licht mee met het realtime licht buiten.

## 2.2 Performance test

* Testen of de applicatie instabiel wordt bij het inladen van meerdere ov-figuren
* Timen of de RET-data binnen 30 seconden ingeladen kan worden wanneer de applicatie afspeelt
* Testen hoelang het duurt voordat de applicatie afspeelt

## 2.3 Regressie test

* Scripts aan OV-modellen toevoegen om ze te laten bewegen
* Snelheid formule van een OV-model aanpassen in het script
* OV-modellen verbinden aan een station
* Meerdere trams inladen vanaf dezelfde laad punt
* Kijken of de OV-data goed ingeladen wordt na het toevoegen van de OV-model scripts om te bewegen
* Kijken of de OV-model nog beweegt op basis van de toegevoegde RET-data

# Teststrategie

## 3.1 Unit Testing

Wij hebben gekozen om gebruik te maken van unit testing. Dit hebben wij gedaan omdat ons project erg complex is. Daarom is het belangrijk dat wij elke functie tot in de detail goed testen. Deze methode van testen werkt ook goed met Unity. Unity heeft namelijk een unit testing framework om mee te werken.

3.2 Performance testing

Wij hebben gekozen voor performance testing. Het systeem is afhankelijk van het verkrijgen en verwerken van data en op tijden het laten bewegen van meerdere figuren. Door middel van performance testing kijken wij of het systeem niet onstabiel wordt en dingen gaat doen die niet horen te gebeuren. Als belangrijkst ook om te kijken hoe snel de applicatie alle data verwerkt die wij binnenkrijgen.

3.2 Regressie testing

Er is gekozen voor regressie testing, omdat de applicatie veel gebruikt maak van componenten die van elkaar afhangen. Het gaat dan voornamelijk met de RET-data en OV-modellen die het makkelijkst anders kunnen reageren op aanpassingen aan het systeem.

# Hardware en software eisen

Voor het kunnen gebruiken van de applicatie zal de volgende hardware en software nodig zijn:

Hardware:

* Internetverbinding (LAN of wifi maakt niet uit)
* Een computer of laptop met minimale specificaties.

Software:

* Een Browser (google Chrome, Microsoft edge, etc.)
* Windows 10 of 11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Use case** | **Test situatie** | **Verwachting** | **Verwachte tijdsduur** |
| Unit Testen | Het beeld van de gebruiker kan aangepast worden door het kiezen tussen verschillende camera’s | Gebruiker start de applicatie op | Gebruiker kan tussen verschillende camera’s kiezen tijdens runtime | 5 min |
| Unit Testen | Het systeem moet van specifieke RET-trams data kunnen ophalen die in de omgeving van beurs bewegen | Gebruiker start de applicatie op | Het systeem haalt alleen de gegevens op van trams die van RET zijn en in de omgeving van beurs bewegen | 7 min |
| Unit Test | De voertuigen moeten op basis van realtime data kunnen bewegen naar de stations | Het moment vanaf de voertuigen voorschijnen in de applicatie | De trams komen op de correcte tijden aan bij hun stations | 25 min |
| Unit Test | Als de applicatie opgestart wordt is de stand van het licht hetzelfde als buiten. | Gebruiker start de applicatie op | Als het donker is in real time dan start de applicatie met donker mood en vice versa | 30 min |
| Unit Test | Tijdens het gebruiken van de applicatie verandert het licht mee met het realtime licht buiten. | Gebruiker start de applicatie op | Het licht verandert mee met het licht buiten | 30 min |
| Performance Test | Testen of de applicatie instabiel wordt bij het inladen van meerdere ov-figuren | Gebruiker start de applicatie op | Bij het laden van meerdere ov-figuren wordt de applicatie soepel gelopen. | 15 min |
| Performance Test | Timen of de RET-data binnen 30 seconden ingeladen kan worden wanneer de applicatie afspeelt | Gebruiker start de applicatie op | Als de gebruiker de applicatie afspeelt duurt het inladen van de OV-data 30 seconden. | 1 min |
| Performance Test | Testen hoelang het duurt voordat de applicatie afspeelt | Gebruiker start de applicatie op | Het duurt tussen 15-20 seconden | 5 min |
| Regressie Test | Scripts aan OV-modellen toevoegen om ze te laten bewegen | Ontwikkelaar opent het project | Elke OV-model heeft een eigen script om functional te maken | 10 min |
| Regressie Test | Snelheid formule van een OV-model aanpassen in het script | Ontwikkelaar start de applicatie | De formule klopt en hoort geen errors te geven | 5 min |
| Regressie Test | OV-modellen verbinden aan een station | Gebruiker start de applicatie op | Alle OV-modellen zijn verbonden aan Beurs station. | 15 min |
| Regressie Test | Meerdere trams inladen vanaf dezelfde laad punt | Applicatie blijft in runtime gedurende test | Meerdere tram van dezelfde lijn laden in op het juiste tijdstip | 10 min |
| Regressie Test | Kijken of de OV-data goed ingeladen wordt na het toevoegen van de OV-model scripts om te bewegen | Een tram van elke lijn wordt gevolgd voorbij meerdere haltes | De snelheid past aan bij vertrek van station | 20 min |
| Regressie Test | Kijken of de OV-model nog beweegt op basis van de toegevoegde OV-data | Applicatie blijft in runtime gedurende test | De tram wordt op station Beurs op basis van de gegeven tijd op RET-website. | 15 min |

# Testscenario

Functional Tests

Unit Testing

*Het beeld van de gebruiker kan aangepast worden door het kiezen tussen verschillende camera’s.*

**Pre flow:**

* Gebruiker start de applicatie

**Happy flow:**

* Gebruiker krijgt een overzicht van camera’s
* Gebruiker kiest een camera
* Het zicht van de gebruiker wordt aangepast

**Alternative flow:**

* Het overzicht van camera’s wordt niet getoond
* Wanneer een gebruiker op een andere camera klikt wordt zijn zicht niet verandert
* Admin voegt een naam en email toe
* Admin drukt op de knop toevoegen

*Het systeem moet van specifieke RET-trams data kunnen ophalen die in de omgeving van beurs bewegen.*

**Pre flow:**

* Gebruiker start de applicatie

**Happy flow:**

* Systeem haalt alle ov-gegevens op via de API
* Systeem filtert naar specifiek RET-voertuigen
* Systeem filtert naar omgeving beurs

**Alternative flow:**

* Ophalen van gegevens werkt niet wegens problemen met de API
* Filteren op data werkt niet. Alle data worden telkens opgehaald.

*De voertuigen moeten op basis van realtime data kunnen bewegen naar station beurs.*

**Pre flow:**

* Gebruiker start de applicatie
* Systeem heeft specifieke RET-data opgehaald

**Happy flow:**

* Systeem laadt voertuig in
* Voertuig rijdt richting het station
* Voertuig komt op de behorende tijd aan op het station.

**Alternative flow:**

* Voertuig wordt niet ingeladen door het systeem
* Voertuig rijdt niet
* Voertuig komt te vroeg aan op het station
* Voertuig komt te laat aan op het station

*Als de applicatie opgestart wordt is de stand van het licht hetzelfde als buiten.*

**Pre flow:**

* Gebruiker start de applicatie

**Happy flow:**

* Systeem kijkt naar de huidige tijd van gebruikers apparaat
* Systeem bepaald aan de hand van de tijd de stand.

**Alternative flow:**

* Ophalen van tijd van gebruiker apparaat werkt niet
* Stand van de licht inval is verkeerd

*Tijdens het gebruiken van de applicatie verandert het licht mee met het realtime licht buiten.*

**Pre flow:**

* Gebruiker start de applicatie
* Stand van het licht is bepaald met de tijd van de gebruikers apparaat

**Happy flow:**

* Stand van het licht wordt met een bepaalde grootte aangepast om zo door te blijven draaien

**Alternative flow:**

* Het licht draait te snel
* Het licht draait te sloom
* Het licht staat vast op een bepaalde axis

Non-Functional Tests

Performance Testing

*Testen of de applicatie instabiel wordt bij het inladen van meerdere ov-figuren*

**Pre flow:**

* Gebruiker start de applicatie
* Bij het runnen van de applicatie wordt de RET-dienstregeling automatisch opgehaald

**Happy flow:**

* Als er meerdere OV's bij Beurs aankomen op hetzelfde moment dan crasht de applicatie niet.
* De figuren rijden gewoon verder zoals ze moeten rijden.

**Alternative flow:**

* De applicatie crasht.
* De figuren rijden niet meer zoals ze horen te rijden.

*Timen of de RET-data binnen 30 seconden ingeladen kan worden wanneer de applicatie afspeelt*

**Pre flow:**

* Gebruiker start de applicatie
* Bij het runnen van de applicatie wordt de RET-dienstregeling automatisch opgehaald

**Happy flow:**

* De RET-data wordt binnen 30 seconden ingeladen
* De figuren rijden verder op de RET-data

**Alternative flow:**

* De applicatie crasht tijdens het inladen van de data.
* De figuren rijden niet meer op basis van de ingeladen data.
* De applicatie crasht tijdens het inladen van de vernieuwde data

*Testen hoelang het duurt voordat de applicatie afspeelt*

**Pre flow:**

* Gebruiker start de applicatie
* Bij het runnen van de applicatie wordt de RET-dienstregeling automatisch opgehaald

**Happy flow:**

* De applicatie speelt af
* De figuren rijden op basis van de ingeladen data.
* De figuren worden ingeladen.

**Alternative flow:**

* De applicatie crasht.
* Het inladen van de data mislukt.
* De figuren laden niet in.
* De figuren rijden niet.

Regression Testing

*Scripts aan OV-modellen toevoegen om ze te laten bewegen*

**Pre flow:**

* Ontwikkelaar voegt een script toe aan een OV-model om het te laten bewegen
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.

**Happy flow:**

* De applicatie speelt af
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data.

**Alternative flow:**

* Het OV-model wordt niet ingeladen.
* Het inladen van de data mislukt.
* Het OV-model rijdt, maar niet op de opgehaalde RET-data.
* De RET-data wordt opgehaald, maar het OV-model rijdt niet.

*Snelheid formule van een OV-model aanpassen in het script*

**Pre flow:**

* Ontwikkelaar heeft code geschreven voor de snelheid formule
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.
* De snelheid van het model hangt af hoe ver het station voor Beurs is

**Happy flow:**

* De applicatie speelt af
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data en komt precies op tijd aan bij zijn station.

**Alternative flow:**

* Het OV-model rijdt helemaal niet.
* Het inladen van de data mislukt.
* Het OV-model komt niet op tijd aan bij het station.

*OV-modellen verbinden aan een station*

**Pre flow:**

* Ontwikkelaar heeft de OV-modellen verbonden met hun stations
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.

**Happy flow:**

* RET-data worden opgehaald
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data en komt op tijd bij zijn toegewezen station aan.

**Alternative flow:**

* Het OV-model rijdt helemaal niet.
* Het inladen van de data mislukt.
* Het OV-model komt niet op tijd aan bij het station.
* Het OV-model rijdt de verkeerde richting op.

*Meerdere trams inladen vanaf dezelfde laad punt*

**Pre flow:**

* Ontwikkelaar heeft de OV-modellen verbonden met dezelfde inlaadpunt
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.

**Happy flow:**

* RET-data worden opgehaald.
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data en komt op tijd bij zijn toegewezen station aan.
* Alle OV-modellen die na de eerste inladen, worden precies op dezelfde manier als het eerste OV-model ingeladen.

**Alternative flow:**

* Het OV-model rijdt helemaal niet.
* Het inladen van de data mislukt.
* Het OV-model komt niet op tijd aan bij het station.
* Het OV-model rijdt de verkeerde richting op.

*Kijken of de OV-data goed ingeladen wordt na het toevoegen van de OV-model scripts om te bewegen*

**Pre flow:**

* Ontwikkelaar heeft de OV-data script toegevoegd aan een OV-model
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.

**Happy flow:**

* RET-data worden opgehaald.
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data en komt op tijd bij zijn toegewezen station aan.

**Alternative flow:**

* Het OV-model rijdt helemaal niet.
* Het inladen van de data mislukt.
* Het OV-model komt niet op tijd aan bij het station.
* Het OV-model rijdt de verkeerde richting op.

*Kijken of de OV-model nog beweegt op basis van de toegevoegde RET-data*

**Pre flow:**

* Ontwikkelaar heeft de RET-data script toegevoegd aan een OV-model
* Ontwikkelaar start de applicatie
* Bij het starten van de applicatie wordt de RET-dienstregeling automatisch opgehaald, op basis hiervan zal het OV-model automatisch moeten inladen en rijden.

**Happy flow:**

* RET-data wordt opgehaald.
* Het OV-model wordt succesvol ingeladen
* Het OV-model rijdt op basis van de opgehaalde data en komt op tijd bij zijn toegewezen station aan.

**Alternative flow:**

* Het OV-model rijdt helemaal niet.
* Het inladen van de data mislukt.
* Het OV-model komt niet op tijd aan bij het station.
* Het OV-model rijdt de verkeerde richting op.

# Rollen en verantwoording

* Mohamad Almarawi: Documenteren
* Parwesh Bhaggan: Documenteren
* Moesab Ajdid: Documenteren
* Darshan Alakhramsing: Notulist
* Bas Weidmann: Tester

# Afhankelijkheden

Sinds de applicatie nog niet volledig af is zullen een aantal functies nog niet getest kunnen worden. Deze functies zullen na afronding in zijn volledigheid getest worden.