Portfolio

Fontys ICT – semester 6



|  |  |
| --- | --- |
| **Datum** | 09-06-2022 |
| **Versie** | 0.5 |
| **Student naam** | Lars van den Brandt |
| **Studentnummer** | 434565 |
| **Docent Fontys** | Tom Meulensteen & Marcel Boelaars |

#### Versie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Aanpassingen** | **Status document** |
| 0.1 | 16-03-2022 | Lars van den Brandt | Eerst opzet | Niet af |
| 0.2 | 18-03-2022 | Lars van den Brandt | Controle spelling door familie | Wachten op feedback |
| 0.3 | 08-04-2022 | Lars van den Brandt | Toevoegen sprint 2 | Wachten op feedback |
| 0.4 | 26-05-2022 | Lars van den Brandt | Toevoegen sprint 4 | Wachten op feedback |
| 0.5 | 09-06-2022 | Lars van den Brandt | Toevoegen sprint 5 | Af. |

#### Verspreiding

|  |  |  |
| --- | --- | --- |
| **Versie** | **Datum** | **Naam** |
| 0.2 | 18-03-2022 | Canvas |
| 0.3 | 08-04-2022 | Canvas |
| 0.5 |  | Canvas |

Inhoud

[Introductie 5](#_Toc105679375)

[De opdracht 6](#_Toc105679376)

[Persoonlijk 6](#_Toc105679377)

[Groep 6](#_Toc105679378)

[Leeruitkomsten 7](#_Toc105679379)

[1. Enterprise-software ontwikkelen door middel van teaminspanning 7](#_Toc105679380)

[Ontwikkelschaal 7](#_Toc105679381)

[Reflectie op vooruitgang 7](#_Toc105679382)

[2. Op context gebaseerd onderzoek verrichten 9](#_Toc105679383)

[Ontwikkelschaal 9](#_Toc105679384)

[Reflectie op vooruitgang 9](#_Toc105679385)

[3. Voorbereiding op een leven lang leren 10](#_Toc105679386)

[Ontwikkelschaal 10](#_Toc105679387)

[Reflectie op vooruitgang 10](#_Toc105679388)

[4. Schaalbare architecturen 11](#_Toc105679389)

[Ontwikkelschaal 11](#_Toc105679390)

[Reflectie op vooruitgang 11](#_Toc105679391)

[5. Development & operations (DevOps) 13](#_Toc105679392)

[Ontwikkelschaal 13](#_Toc105679393)

[Reflectie op vooruitgang 13](#_Toc105679394)

[6. Cloud services 16](#_Toc105679395)

[Ontwikkelschaal 16](#_Toc105679396)

[Reflectie op vooruitgang 16](#_Toc105679397)

[7. Beveiliging door middel van ontwerp 18](#_Toc105679398)

[Ontwikkelschaal 18](#_Toc105679399)

[Reflectie op vooruitgang 18](#_Toc105679400)

[8. Gedistribueerde gegevens 20](#_Toc105679401)

[Ontwikkelschaal 20](#_Toc105679402)

[Reflectie op vooruitgang 20](#_Toc105679403)

[Reflectie 21](#_Toc105679404)

[Sprint 1 21](#_Toc105679405)

[sprint 2 21](#_Toc105679406)

[sprint 4 (oplevering 3) 21](#_Toc105679407)

[Sprint 5 Conclusie 22](#_Toc105679408)

[Bronnen 23](#_Toc105679409)

[0. Sprint deliveries 23](#_Toc105679410)

[1. Pitch 23](#_Toc105679411)

[2. Analyse 24](#_Toc105679412)

[3. Onderzoek 24](#_Toc105679413)

[4. Polargram 25](#_Toc105679414)

# Introductie

Om te beginnen zal ik kort wat over mijzelf vertellen. Ik ben Lars van den Brandt en ben 19 jaar oud. Na het halen van mijn havo diploma ben ik gelijk begonnen met HBO ICT. Buiten het vak informatica op de havo, had ik weinig tot geen kennis over de ICT wereld. Ik heb als eerste gekozen voor software als profiel, vervolgens Cyber Security als specialisatie.   
In semester 5 heb ik stage gelopen bij de Koninklijke Luchtmacht, in het Cyber Warfare Team. Dit is, zoals de naam van het team al zegt, een Cyber Security stage.

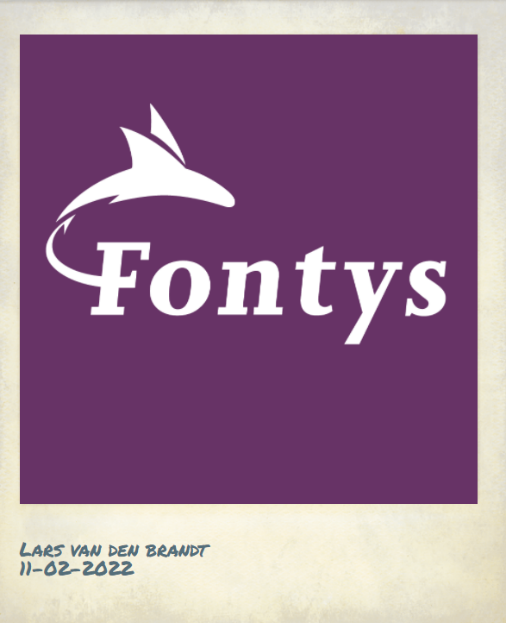
Ik heb nog weinig tot geen kennis over het ontwikkelen van Enterprise Software. Het is daarom ook mijn doel om hierover te leren tijdens dit semester.

Dit document dient als een leeswijzer door het portfolio van software semester 6 op de Hogeschool Fontys ICT. Het doel van dit document is om een duidelijk beeld te geven aan welke opdrachten en competenties is gewerkt gedurende dit semester. Na het lezen van dit document zou de lezer een duidelijk beeld moeten krijgen op welk niveau ik dit semester heb gepresteerd.

In de leeswijzer staat per competentie beschreven hoe ik deze heb aangetoond door middel van een beschrijving. Mocht het nodig zijn, staat er een link naar een bijbehorend document dat dient als support voor het onderbouwen van mijn aangetoonde competenties.

# De opdracht

## Persoonlijk

Als persoonlijk project heb ik gekozen voor PolarGram. PolarGram lijkt op een social media platform zoals Instagram, maar post de foto’s in de vorm van een Polaroid. Toen in het begin van het semester een project gekozen moest worden, wist ik niet zeker wat ik moest doen. Er werd een voorbeeldproject: ‘Kwetter’ gegeven. Kijkend naar dit project, dat gebaseerd is op Twitter, leek het mij een goed idee om in het social media thema te blijven. Zo denk ik met dit project alle leeruitkomsten aan te tonen.

Waarom PolarGram? Ik merk dat bijna iedereen tegenwoordig Instagram gebruikt. Maar ik merk in mijn eigen vriendengroep dat Polaroid foto’s weer helemaal aan het terugkomen zijn. Het leek mij daarom een leuk idee om de moderne digitale wereld met de oude foto generatie te combineren.

## Groep

Voor het groep project zit ik in team EightWest. De naam is ontstaan omdat, ook al zijn we met zes mensen, we resultaten hebben beloofd voor acht mensen. West, in onze naam, is gebaseerd op de regio dat wij hebben gekregen voor ons groepsproject.

Als groepsproject heeft onze groep de EnergyGrid opdracht gekregen. Het is een opdracht met een brede scope, waar we zelf min of meer mogen bepalen wat we willen maken.   
We hebben besloten om een applicatie te maken voor drie verschillende gebruikers: een huidhouden, een stad en de energy grid. Op de applicatie kan een dashboard worden getoond, waar verschillende overzichten staan over de desbetreffende gebruiker. Zo kan bijvoorbeeld worden gekeken hoeveel energie er wordt gebruikt en wordt opgehaald.  
Buiten overzichten op het dashboard, komt er een functionaliteit die energie opbrengsten, van bijvoorbeeld zonnepanelen, kan voorspellen. Zo kan worden gekeken wanneer de energie het beste kan worden gebruikt, verkocht of worden gekocht.

# Leeruitkomsten

## Enterprise-software ontwikkelen door middel van teaminspanning

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Beginning/ proficient |

### Reflectie op vooruitgang

#### sprint 1

Na de eerste sprint is het idee voor de Enterprise-software bedacht. Zo zijn we aan de slag gegaan met het coderen en zijn de onderzoeken gestart. Dit is zowel in het groepsproject, als het persoonlijke project het geval.

#### Sprint 2

Voor zowel het groepsproject als het persoonlijke project zijn er veel stappen gemaakt binnen deze learning outcome.   
Zo zijn we binnen de groep gestart met het onwikkelen van de enterprise software op een agile (Scrum) manier van werken.

Zowel binnen het groeps- als het persoonlijke project wordt er gewerkt met microservices. Deze worden geüpload naar een Git repository. Zo wordt de code automatisch opgeslagen en kunnen we met meerdere mensen tegelijk werken, zonder dat er teveel merge conflicts ontstaan.

#### Sprint 4 (oplevering 3)

Voor zowel het groepsproject als het persoonlijke project zijn er veel stappen gemaakt binnen deze learning outcome.

Binnen het groepsproject loopt alles vlekkeloos, er moest veel onderzoek worden gedaan, maar er moest ook nog veel gebeuren binnen de ontwikkeling van de applicatie. We hebben de taken verdeeld door gebruik te maken van het Jira board. Ik ben zeer tevreden over de vooruitgang die we hebben geboekt in de afgelopen sprint. De feedback van de docenten is erg goed, waar wij als groep ook erg blij mee zijn.

In het persoonlijke project ben ik tegen een aantal dingen aangelopen, zo had ik moeite met het opzetten van RabbitMQ, het aanpassen van de testen, en het implementeren van Kubernetes. Na veel onderzoek, veel proberen, en wat hulp van mijn groepsgenoten en docenten is het gelukt om mijn doelen te behalen. Kubernetes is nog niet volledig geimplementeerd, dit ga ik komende sprint doen.

Al met al, ben ik erg tevreden over de afgelopen sprint, hopelijk kan ik dit zo doorzetten in de laatste sprint.

Link naar Jira: [Jira Board](https://jira.fhict.nl/secure/RapidBoard.jspa?rapidView=127&projectKey=SRBGROU2&view=planning&selectedIssue=SRBGROU2-119&issueLimit=100)

Link naar persoonlijke repository (GitHub): [PolarGram](https://github.com/LarsvdBrandt/PolarGram)  
Link naar groeps repository back-end (GitLab): [Back-end](https://git.fhict.nl/I438583/energygrid-backend)  
Link naar groeps repository front-end (GitLab): [Front-end](https://git.fhict.nl/I438583/energygrid-frontend)

## Op context gebaseerd onderzoek verrichten

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Proficient |

### Reflectie op vooruitgang

#### sprint 1

In het groepsproject hebben we de taken verdeeld en is er al veel onderzoek gedaan. Zo is er onderzoek gedaan naar verschillende authenticatie mogelijkheden, naar de frontend, naar de backend en naar de message broker.   
Voor het persoonlijke project heb ik een onderzoeksplan gemaakt voor de emerging trends onderzoek. Hier heb ik het onderwerp: Domain-Driven Design (DDD) gekozen.

#### Sprint 2

Mijn onderzoeksplan naar Domain-Driven-Design doorgenomen door Marcel, hierop heeft hij positieve feedback gegeven. Volgend semester ga ik hier verder mee aan de slag.   
Voor het groepsproject hebben we met het volledige team zitten brainstormen over het Design Oriented Research Plan. Hierop heeft Gertjan feedback gegeven, deze feedback hebben we meteen verwerkt.

#### Sprint 4 (oplevering 3)

Tijdens deze sprint heb ik veel gewerkt aan zowel mijn persoonlijke onderzoek als het groepsonderzoek.

Voor mijn persoonlijke onderzoek heb ik de emerging trends research over Domain-Driven Design afgerond. Dit heb ik besproken met de docenten. Afgesproken om het onderzoek op Canvas in te leveren, zodat het onderzoek bij het eindbeoordelingsgesprek kan worden besproken.

Voor het groepsproject hebben we het Design Oriënted Research afgerond. Dit was een groot onderzoek met veel vragen, we hebben het onderzoek verdeeld tussen alle teamleden, zodat iedereen een stukje onderzoek deed. Na het afronden van het onderzoek hebben we dit opgestuurd naar Gertjan, hij heeft hier uitgebreid feedback op gegeven, dit was erg fijn. Na het ontvangen van de feedback hebben we de punten opnieuw verdeeld tussen de teamleden, en vervolgens direct afgerond.

Ik ben zelf geen grote fan van het onderzoeken, maar ik snap dat het een belangrijk onderdeel is van de studie. Ik heb zowel in het persoonlijke als het groepsonderzoek veel gebruik gemaakt van het DOT Framework, was voor mij persoonlijk veel structuur gaf aan het onderzoek. Hierdoor was het onderzoeken makkelijker en heb ik veel geleerd.

Link naar persoonlijk onderzoek: [Onderzoek](https://github.com/LarsvdBrandt/PolarGram/tree/master/Personal%20Documentation/3.%20Onderzoek)

Link naar Design oriënted research: [Onderzoek](https://docs.google.com/document/d/1hPSdHg_-H24-BgR-4iYD5Sjl2tzd_HYZM-FE7-I2UbI/edit)

## Voorbereiding op een leven lang leren

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Proficient |

### Reflectie op vooruitgang

#### sprint 1

Voor deze leeruitkomst heb ik nog niet veel gedaan dit semester. Natuurlijk weet ik zelf waar ik goed in ben en wat ik leuk vind, dit is nog niet op papier gezet en hier moet ik mij dit semester nog meer mee bezig houden.

Het signaleren van trends heb ik onbewust gedaan naar mijn mening. Zo heb ik bij mijn persoonlijke project de terugkomende trend van de polaroid gecombineerd met de jarenlange Instagram trend.

#### sprint 2

Kijkend naar de uitleg van deze leeruitkomst denk ik dat ik hier onbewust al veel aan heb gewerkt. Na mijn HBO ICT wil ik graag naar de Koninklijke Militaire Academie (KMA) binnen de luchtmacht. Hier wil ik mijn militaire droom combineren met mijn, dan behaalde, ICT diploma.   
Vorig semester (semester 5), heb ik stage gelopen bij de Koninklijke Luchtmacht in het Cyber Warfare Team (CCWT). De stage die ik heb gelopen is mij persoonlijk heel goed bevallen, waardoor mijn keuze om naar de KMA te gaan nog meer is bevestigd.

Naast de stage heb ik mij ook aangemeld voor een minor abroad. Ik ben geselecteerd voor mijn eerste keuze: Dongguk University, in Zuid-Korea.   
Mensen kijken mij altijd gek aan en vragen: ‘Waarom Korea?’. Ik heb gekozen voor Zuid-Korea omdat het een land is met een compleet verschillende cultuur en levensstijl dan wij hier kennen in Nederland. Het is iets wat nog niemand die ik ken heeft gedaan en het lijkt me een ervaring voor het leven. De minor die ik op deze school ga volgen is nog niet duidelijk, deze opties komen ook pas eind mei beschikbaar. Wel weet ik dat ik graag een economische minor wil doen. Dit omdat ik na mijn militaire carrière graag nog in het bedrijfsleven wil werken, en deze kennis dan van pas zal komen.

#### sprint 4 (oplevering 3)

Ik ben begonnen met het zoeken naar een stage. Ik wil graag mijn afstudeerstage lopen bij ASML, dit lijkt mij een interessant bedrijf omdat het een van de grootste ‘tech’ bedrijven van de wereld is, en omdat het een Nederlands bedrijf is. Door hulp te vragen aan mensen die werken bij ASML heb ik een aantal contacten weten te krijgen waarmee ik verder kan werken. Ik ben van plan om komende sprint mijn motivatie en CV op te sturen.

Naast het uitwerken van de stage heb ik voor Zuid-Korea nog een aantal formulieren moeten invullen ter bevestiging van mijn aanmelding, ik wacht hierop nog op een antwoord van Dongguk University.

## Schaalbare architecturen

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Proficient |
| 0.5 | Sprint 5 | Proficient/ advanced |

### Reflectie op vooruitgang

#### sprint 1

Aan deze leeruitkomst heb ik al veel gedacht bij zowel het groeps- als het persoonlijke project. Zo heb ik in mijn persoonlijke project in de backend een API gateway staan die gelinkt is aan een microservice. Ik heb dit direct zo opgezet bij het begin van het coderen van dit semester, omdat ik in semester drie ook deze techniek had gebruikt. Zo kon ik deze kennis opnieuw gebruiken.   
In het groepsproject heeft Daan zich vooral bezig gehouden met de architectuur.

#### Sprint 2

Om mijn architectuur in kaart te brengen heb ik C-modellen gemaakt. Daan heeft hieraan gewerkt voor ons groepsproject. De architectuur van het groepsproject is vervolgens met de hele groep doorgenomen, hierdoor hebben we nog een aantal dingen kunnen aanpassen om het naar iedereens wens te maken.

Naast het in kaart brengen van de architectuur heb ik een ‘walking skeleton’ gemaakt, zoals aangeraden door Tom en Marcel. Zo het ik een post microservice, die verantwoordelijk is over de data van alle posts. De afbeeldingen die opgeslagen moeten worden, worden nu in een aparte service opgeslagen.

#### Sprint 4 (oplevering 3)

In mijn vorige sprint meende ik de ‘walking skeleton’ afgerond te hebben. Na dit besproken te hebben met zowel Tom als Marcel, was dit toch niet de realiteit.   
Ik ben deze sprint aan de slag gegaan met RabbitMQ om de communicatie tussen microservices te maken.

Ik kwam er op het begin niet goed uit, omdat ik hier nog nooit mee had gewerkt, of over had gehoord (buiten dit semester). Na het verrichten van wat onderzoek heb ik een proefapplicatie opgesteld om dit te implementeren. Hierbij was het gelukt om tussen twee applicaties te communiceren door middel van RabbitMQ.

Vervolgens heb ik dit in mijn persoonlijke project verwerkt en getest. Ik heb de scalability gestest door meerdere subscribers op te starten en te kijken of er maar één van de subscribers de request uit de queue zou halen, en niet beide subscribers. Deze test was geslaagd.

Naast RabbitMQ ben ik aan de slag gegaan met Kubernetes, om de services scalable te maken. Hier liep ik tegen een aantal problemen aan. Met de hulp van zowel mijn groepsgenoten als de docenten heb ik deze problemen verholpen.  
Ik heb kubernetes nog niet volledig werkend, maar dit verwacht ik komende sprint voor elkaar te krijgen.

#### Sprint 5

Op deze leeruitkomst lag voor mij geen focus in deze sprint. Ik had hier de vorige sprint al proficiënt voor, en aangezien ik eerst de andere leeruitkomsten af wilde hebben, ben ik daar aan gaan werken.

Ik ben deze sprint aan de slag gegaan met Kubernetes, omdat dit nodig was voor DevOps als deployment. Na dit werkend te krijgen heb ik loadbalancing toegevoegd op de services. Hierdoor kon ik kijken of de services ook daadwerkelijk schaalbaar waren. Dit heb ik ook getest door het uitvoeren van load testen. Na het verhogen van de CPU van de service, wordt er een extra pod aangemaakt om de load te balanceren.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Klik [hier](#_Sprint_5) om de documentatie van de loadbalancers en de bijbehorende testen te bekijken.

Link naar persoonlijke C-modellen (GitHub): [C-Modellen-Persoonlijk](https://github.com/LarsvdBrandt/PolarGram/tree/master/Personal%20Documentation/2.%20Analyse/1.%20C-Models)  
Link naar groep C-modellen (Google Docs): [C-Modellen-Groep](https://drive.google.com/drive/folders/1zDDex6b9A-zx6MTbdnYFk-mfhQgY00uq?ths=true)

## Development & operations (DevOps)

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Beginning |
| 0.5 | Sprint 5 | Proficient/ advanced |

### Reflectie op vooruitgang

#### sprint 1

Zoals eerder benoemd wordt er in zowel het persoonlijke als het groepsproject gewerkt met microservices. Zo heeft elke service de verantwoordelijkheid over een deel van de applicatie. Mocht een service uitvallen, blijft de rest nog wel werkend. In het groepsproject wordt er gewerkt met Docker containers. Hier heb ik voor mijn persoonlijke project nog niks gedaan.

#### Sprint 2

Voor mijn persoonlijke project heb ik de CI pipeline opgezet. Op het moment dat ik mijn project naar mijn GitHub push, wordt de front-end en back-end gebouwd. De testen die ik heb geschreven worden ook meteen gerund.   
Naast de CI pipeline heb elke service in Docker images gezet, deze zijn geüpload naar DockerHub. Het volledige project kan worden gedraaid door de Docker-Compose te starten. Dit is ook beschreven in de README.md file.

Afbeelding hieronder bevat het resultaat van de CI workflow, na het bewerken van zowel de front-end als back-end. Beide slagen ze.

#### Sprint 4 (oplevering 3)

Na het implementeren van RabbitMQ werkte mijn testen in de backend niet meer. Dit probleem heb ik al snel verholpen.   
Verder heb ik zoals bij ‘Schaalbare Architecturen’ uitgelegd dat ik aan de slag ben gegaan met Kubernetes. Dit is nog niet volledig afgerond, maar ik dit verwacht ik komende sprint voor elkaar te krijgen.

Mijn vorige feedback voor deze leeruitkomst was om naar SDLC en SLA te kijken, dit ga ik ook komende sprint doen.

#### Sprint 5

Na de vorige sprintoplevering had ik nog een aantal verbeterpunten gekregen van Tom en Marcel. Deze punten heb ik in deze sprint verwerkt.

**Auto deploy:**

Afbeelding met tekst

Automatisch gegenereerde beschrijvingVoor auto deploy heb ik mijn applicatie in Kubernetes werkend gekregen. Ik was hier de vorige sprint al mee bezig maar ik kreeg de connectie tussen de services niet werkend.   
Het uiteindelijke probleem was: ik had een typfout gemaakt bij de import van een file.

Het is een domme fout die vaak voorkomt bij software engineering. Ik ben zeer tevreden over hoe dit nou staat kubernetes.  
Naast het deployen heb ik loadbalancing toegevoegd (HPA). Op het moment dat de load op een service te hoog wordt, wordt er een extra pod aangemaakt.

Voor het voorbeeld heb ik de max cpu usage van een pod op 10% gezet. Als er nou meer cpu wordt gebruikt, wordt er automatisch een extra pod aangemaakt voor load balancing. Zie afbeelding hieronder.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**End 2 end tests:**

Afbeelding met tekst

Automatisch gegenereerde beschrijvingAfbeelding met tekst

Automatisch gegenereerde beschrijvingNa het deployen in kubernetes ben ik aan de slag gegaan met E2E testing. Voor E2E testing heb ik jest en react-testing-framework gebruikt in de frontend. Ik test de volledige CRUD van een microservice (de commentservice).

Het testen ging mij vrij makkelijk af, aangezien ik dit in een ander semester ook al op deze manier had toegepast. Wel liep ik tegen het probleem aan dat ik nou authentication gebruik. Ik kon dus op de testen geen gebruiker laden.

Na wat geprobeerd te hebben, heb ik auth0 kunnen mocken. Hierbij mock ik een gebruiker, waarmee ik de testen kon uitvoeren.

**Performance test:**

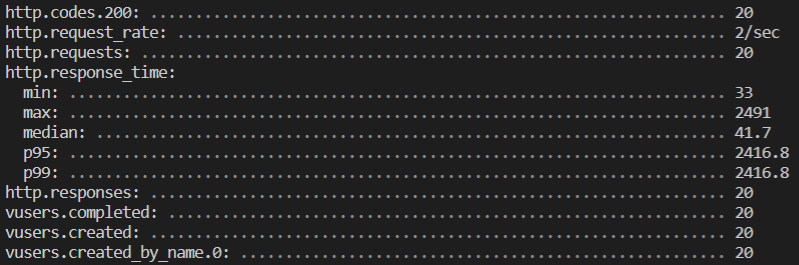
Naast de E2E testen en de deployment had ik nog als feedback om performance testen uit te voeren. Voor de performance testen gebruik ik Artillery.

In Artillery doe ik load testing, waarbij ik verschillende calls stuur naar de backend. Hierbij wordt gekeken naar de response die de service teruggeeft.

Ik heb gekozen voor load testing als performance testen, omdat ik de load balancer heb toegevoegd bij het deployen van mijn applicatie in kubernetes. Door de load testen kon ik dus ook zien of deze goed werkte.

De data die ik terugkreeg van de testen was erg positief. Door de load testen kon ik ook zien dat de services scalable waren. Zo werd er bij hoge CPU usage een extra pod aangemaakt.

Resultaat Artillery test:



Gevolg Artillery test (replica wordt 2, omdat CPU te hoog wordt):

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Link naar persoonlijke Docker images: [DockerHub](https://hub.docker.com/u/434565)   
Link naar CI workflow (GitHub): [Workflow](https://github.com/LarsvdBrandt/PolarGram/tree/master/.github/workflows)

## Cloud services

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.3 | Sprint 2 | Undefined |
| 0.4 | Sprint 4 | Proficient |
| 0.5 | Sprint 5 | Proficient |

### Reflectie op vooruitgang

#### sprint 1

Hier heb ik nog niet mee gewerkt in deze sprint, dit verwacht ik ook pas later dit semester toe te passen.

#### Sprint 2

Aan deze leeruitkomst zijn geen extra ontwikkelingen gemaakt. Zoals benoemd in de vorige sprint, verwacht ik hier richting het einde van het semester aan te werken.

#### Sprint 4 (oplevering 3)

Nu het einde van het semester nadert, heb ik ook voor deze leeruitkomst grote stappen gemaakt.   
Als eerste heb ik Auth0 voor authenticatie. Dit is een cloud service, die gebruik maakt van Amazon Web Services.

Verder heb ik Azure cloud service gebruikt voor het opslaan van mijn afbeeldingen, ze worden hier in een blob image opgeslagen, deze images kunnen vervolgens weer op de frontend ingeladen worden.   
Na het fijne werken met de blob image in Azure, heb ik mijn databases van mijn services ook op Azure laten draaien. Dit werkte goed, maar na een aantal dagen bleek het dat ik ongeveer [€](https://nl.wikipedia.org/wiki/Euroteken)8,- per dag, per database moest betalen. Dit kon, i.v.m. mijn [€](https://nl.wikipedia.org/wiki/Euroteken)100,- studentencredit niet lang doorgaan. Ik heb mijn applicatie vervolgens weer omgebouwd naar MongoDB Atlas, dit is een cloud service die indirect ook gebruik maakt van Azure, maar zonder te betalen.

Terugkijkend ben ik zeer tevreden over de cloud services die ik op dit moment gebruik binnen mijn applicatie. Het betalen van de database voor mijn services was een tegenslag, maar dit is gelukkig opgelost door de applicatie om te zetten naar MongoDB.

#### Sprint 5

Ik heb deze sprint niks gedaan aan cloud services, dit is omdat ik de focus wilde leggen op de learning outcomes waar ik feedback op had gekregen, zodat alles op proficient komt te staan.

In de vorige sprint heb ik MongoDB als cloud service toegevoegd. Dit heb ik gedaan omdat ik wilde aantonen dat ik binnen mijn project cloud services kon gebruiken. Ik heb gekozen voor MongoDB omdat het gratis is om te gebruiken.  
Het nadeel van MongoDB is dat het een NoSQL server is, in mijn project zou MySQL beter van toepassing zijn. Het nadeel hiervan is dat het minder nauwkeurig is in vergelijking met MySQL. NoSQL is bedoeld om grote hoeveelheden data in op te slaan, die niet specifiek rekening hoeven te houden met de colommen die zijn opgegeven. Dit is wel het geval bij MySQL.   
Ik had eerst Azure gebruikt als database, deze gebruikt wel MySQL, het bleek dat ik voor deze database moest betalen, en aangezien ik maar een studentenlimiet heb van €100,- heb, is dit niet te doen. Mijn blob storage vind wel plaats op Azure, dit is wel voor gebruik.

Het is in dit project dus beter om MySQL toe te passen, maar aangezien MongoDB gratis was voor gebruik, ben ik hiervoor gegaan.

## Beveiliging door middel van ontwerp

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.2 | Sprint 2 | Beginning |
| 0.4 | Sprint 4 | Proficient |
| 0.5 | Sprint 5 | Proficient |

### Reflectie op vooruitgang

#### sprint 1

Hier heb ik nog niet mee gewerkt in deze sprint, dit verwacht ik ook pas later dit semester toe te passen.

#### sprint 2

Zoals eerder benoemd heb ik het design van mijn architectuur in kaart gebracht via de [C-modellen.](https://github.com/LarsvdBrandt/PolarGram/tree/master/Personal%20Documentation/2.%20Analyse/1.%20C-Models)   
Ook ben ik deze sprint begonnen met authenticatie voor zowel mijn persoonlijke als groepsproject. Voor beide projecten gebruiken we auth0. Het begin hiervoor heb ik opgezet voor mijn persoonlijke project. Er zijn nog geen rollen of restricties, maar de data die wordt doorgegeven bij de login, wordt wel gebruikt bij het maken van een post.

Daarnaast ben ik voor het groepsproject, samen Daan en Simone, aan de slag gegaan met het testplan. Hierin worden de test methodes en test cases benoemd. Dit kan later worden toegepast binnen het project zelf.   
Voor mijn persoonlijke project heb ik in de back-end al een aantal testen geschreven, hierbij wordt gebruik gemaakt van mock-data. Deze testen worden door de CI ook gerund bij een update binnen mijn GitHub Repository.   
Voor het testen in de front-end heb ik nog niks gedaan.

#### sprint 4 (oplevering 3)

Zoals eerder benoemd werkte mijn testen niet meer door de implementatie van RabbitMQ. Dit probleem heb ik deze sprint verholpen.   
Daarnaast hebben we in het groepsproject het testplan uitgevoerd, hier zijn een aantal verbeterpunten uitgekomen die we meteen in het project hebben aangepast.

Ik heb de afgelopen dagen veel gekeken naar de OWASP top 10, hier wist ik al veel vanaf i.v.m. mijn cyber security profiel achtergrond. Ik ben van plan om komende sprint een document te schrijven over de, naar mijn mening, belangrijkste in deze top 10 voor mijn applicatie.

#### Sprint 5

In de vorige sprint had ik als plan om in deze sprint een document te schrijven over de OWASP top 10. In dit document heb ik alle regels van de top 10 benoemd.   
Daarbij heb ik per regel een voorbeeld gegeven, hoe dit kan worden toegepast in realiteit. Daarbij staat een afbeelding met de workflow, hoe een aanvaller (hacker) een applicatie of netwerk kan aanvallen met de bijbehorende regel.

In het document staat ook wat ik voor elke regel uit de top 10 heb gedaan om mijn eigen applicatie zo veilig mogelijk te maken.

Naast het document van de OWASP Top 10, heb ik zoals eerder benoemd E2E testen en performance testen uitgevoerd. Klik [hier](#_Sprint_5) om de documentatie hiervan te lezen.

Terugkijkend op mijn project, ben ik overtuigd dat ik deze leeruitkomst zeker heb aangetoond.

Link naar groepsproject testplan (Google Docs): [Testplan](https://docs.google.com/document/d/1K3BXR_DPjSxjk8pTj2xkheY4eK2WyJORM6TtvyxxA3A/edit)  
Link naar back-end testen: (GitHub): [Back-end testen](https://github.com/LarsvdBrandt/PolarGram/tree/master/PolarGram/TestServicePost)

## Gedistribueerde gegevens

### Ontwikkelschaal

|  |  |  |
| --- | --- | --- |
| **Versie** | **Beschrijving** | **Niveau** |
| 0.1 | Sprint 1 | Nog niet bekend |
| 0.2 | Sprint 2 | Orienting/Beginning |
| 0.4 | Sprint 4 | Proficient |
| 0.5 | Sprint 5 | Proficient/ advanced |

### Reflectie op vooruitgang

#### sprint 1

Hier heb ik nu nog niet veel mee gedaan. De gegevens die ik heb, worden per microservice in een andere database gezet. Nu heeft elke microservice een aparte inmemory database, omdat de applicatie nog in ontwikkeling is.

#### Sprint 2

Zoals benoemd in sprint 1, heeft in mijn persoonlijke project elke service een aparte inmemory database. Deze wordt in de toekomst nog omgezet, maar i.v.m. development hou ik het nu nog zo.   
In mijn persoonlijke project heb ik een aparte service in de back-end gemaakt om afbeeldingen op te slaan. De titel wordt door middel van een string opgeslagen in de database van de ServicePost.

#### Sprint 4 (oplevering 3)

Voor deze leeruitkomst heb ik veel vooruitgang gemaakt. Ik heb elke service een aparte database gegeven. Deze databases staan in de cloud. Eerst had ik dit opgezet in Azure, maar al snel bleek het dat ik hiervoor moest betalen. Ik ben opzoek gegaan naar een andere oplossing, en heb vervolgens MongoDB gebruikt. Dit is een cloud service die gebruik maakt van azure cloud services.

Voor mijn afbeeldingen heb ik azure blob storage gebruikt. Hier sla ik de afbeeldingen in de cloud op, in plaats van in de mappenstructuur van de service zelf.

#### Sprint 5

Na de vorige sprintoplevering had ik nog een aantal verbeterpunten gekregen van Tom en Marcel. Deze punten heb ik in deze sprint verwerkt. Ik moest nog rekening houden met GDPR binnen mijn project.

Ik was al van plan om GDPR in deze sprint te verwerken binnen mijn project. Ik heb als eerste een onderzoek gedaan naar GDPR, waarin staat wat het precies is, wat persoonsgegevens zijn, en wat de daadwerkelijke GDPR regels zijn. Vervolgens heb ik benoemd wat ik bij al deze regels heb gedaan om deze leeruitkomst aan te tonen.

Ook heb ik de functie toegevoegd, waarmee een gebruiker binnen een knop alle gebruikersgegevens kan ophalen. Dit houd in dat alle posts en comments die door de gebruiker zijn gemaakt, worden opgehaald. Ook kan de gebruiker al zijn/haar gegevens binnen een knop (en een bevestigingsknop) verwijderen. Daarmee heb ik GDPR naar mijn mening zeker aangetoond.

# Reflectie

## Sprint 1

Zelf ben ik erg tevreden over de verloop van sprint 1. Na bedenken van het idee en het opzetten van de requirements ben ik snel aan de slag gegaan met het coderen. Omdat ik dit al een tijdje niet meer had gedaan, was het weer wennen en moest ik een aantal dingen opnieuw onderzoeken om het werkend te krijgen.

Ik ben lang bezig geweest met de post microservice, omdat deze aan een inmemory database moest worden gekoppeld na de feedback van Marcel en Tom. Dit werkt nu goed dus kan ik weer verder werken.

Voor het groepsproject hebben we ook grote stappen gemaakt. De communicatie binnen de groep is goed en de taken worden eerlijk verdeeld.

Al met al, denk ik dat het een goede sprint is geweest.

## sprint 2

Ook over de verloop van deze sprint ben ik erg tevreden. Zo heb ik veel vooruitgang gemaakt binnen zowel mijn persoonlijke project, als mijn groepsproject.

In mijn persoonlijke project heb ik me veel bezig gehouden met de authenticatie, en het verder ontwikkelen van de services. Ook heb ik een CI pipeline opgezet, zodat mijn project meteen wordt nagekeken op het moment dat ik het upload naar GitHub. Voor CD heb ik docker images gemaakt voor zowel de back-end als front-end. Deze images kunnen met docker compose worden gerund.   
Naast het ontwikkelen van de code ben ik bezig geweest met onderzoeken voor verdere ontwikkeling van mijn applicatie.

In het groepsproject heb ik samen met de rest van de groep een goed begin gemaakt aan de applicatie. Zo hebben we, na de feedback van Marcel en Tom, een ‘Walking skeleton’ gemaakt. Ik heb veel gewerkt met de front-end. In de backend heb ik een service gemaakt, om de integratie te kunnen testen.   
Ook in het groepsproject heb ik veel onderzoek gedaan.

Ik ben erg tevreden over deze sprint, ik denk dat ik grote stappen heb gemaakt, en ik ben gemotiveerd om nog door te werken aan dit project.

## sprint 4 (oplevering 3)

Ik ben zeer tevreden over de afgelopen sprints. Ik heb grote stappen gemaakt binnen het onderzoek, voor zowel het persoonlijke als het groepsproject. En heb veel vooruitgang gemaakt binnen de applicatie zelf.

In mijn persoonlijke project heb ik mij veel bezig gehouden met messaging tussen de services om de ‘Walking Skeleton’ af te ronden. Ook ben ik veel bezig geweest met de cloudservices en distributed data. De emerging trends research is afgerond, waar ik ook zeer tevreden mee ben.   
Verder heb ik nog gewerkt met het afronden van de testen in de backend. Ook ben ik begonnen met het opzetten van Kubernetes om de applicatie scalable te maken. Dit is nog niet volledig gelukt, maar hier ga ik komend semester aan werken.

In het groepsproject heb ik veel gewerkt aan het onderzoek en de integratie tussen front- en backend.   
Voor de applicatie zelf heb ik gewerkt aan meerdere services die belangrijk waren voor de functionaliteit van de applicatie.   
Het onderzoek hebben we opgedeeld tussen alle teamleden. Deze is afgerond en opgestuurd voor feedback. Na het ontvangen van de feedback hebben we deze direct verwerkt in de applicatie.

Al met al, ben ik erg tevreden over deze sprint, ik heb veel vooruitgang gemaakt in zowel het persoonlijke als het groepsproject. Kijkend naar de learning outcomes, heb ik naar mijn mening nog niet alles op proficient staan, maar dit ga ik zeker afronden in de volgende, en laatste sprint.

## Sprint 5 Conclusie

In het afgelopen semester heb ik enorm veel geleerd. Ik had al eerder gewerkt met de microservices structuur, maar nog nooit op zo’n grote en efficiënte manier. Ik ben van mening dat ik zowel in het groep als het persoonlijke project grote stappen heb gemaakt om de leeruitkomsten aan te tonen.

In het groepsproject hebben we gestructureerd gewerkt volgens de scrum methode. We maakte gebruik van een Jira board als scrum board, waarbij Daan de scrummaster was. Door het uitvoeren van een standup op de maandagen en donderdagen (dagen waarop we op school aanwezig waren) was iedereen constant op de hoogte van elkaars taken. De taken werden eerlijk verdeeld, waarbij iedereen zelf kon bepalen waar hij/zij op dat moment zin in had. Binnen het groepsproject was er constant een goede sfeer, wat het ontwikkelen van software met een team een stuk leuker maakt. Als ik een vraag had, kon ik dit in de groep gooien, hierop kreeg ik vrijwel altijd meteen een reactie met iemand die mij te hulp schoot. Dit werkte ook andersom, ik heb ook groepsgenoten kunnen helpen, vooral richting het einde van het project, waarbij ik mensen kon helpen met deployment en testen.

In het persoonlijke project heb ik wat minder gestructureerd gewerkt. Ik keek hierbij naar de leeruitkomsten, en bepaalde per dag waar ik graag aan wilde werken. Dit heb ik gedaan omdat ik wilde kijken of deze methode ook werkt, aangezien ik alleen werk. Mij is opgevallen dat deze methode voor mij persoonlijk goed werkt, maar dat er ik wel rekening mee moet houden dat ik feedback blijf vragen, zodat ik weet wat de volgende stappen zijn.   
Ik ben erg tevreden over het eindproduct dat ik heb neergezet, het project bevat minder services dan ik had verwacht in het begin van het semester. Dit is omdat ik bij het maken van het plan geen rekening had gehouden met de tijd die de nieuwe functies (waar ik nog nooit mee heb gewerkt, bijv. RabbitMQ) zou kosten.   
Het werken met RabbitMQ en het werken met de deployment hebben mij bijzonder veel tijd gekost, omdat ik hier nog nooit mee had gewerkt en dus nog geen kennis over had. Hierdoor kwamen er veel errors binnen mijn project. Door veel onderzoek te doen en wat hulp van mijn groepsgenoten heb ik alles werkend gekregen, waar ik erg tvreden mee ben.

# Bronnen

Als eerste bron zet ik een link naar mijn GitHub voor mijn persoonlijke project: https://github.com/LarsvdBrandt/PolarGram.git

De bijlage bijgevoegd in de portfolio, gesorteerd op de mappenstructuur.

**1. Group:**

In de map “Group” zitten tot de laatste sprint nog geen documenten bijgevoegd. Dit omdat de groep inleverdatums op een andere datum zijn dan de persoonlijke inleverdatums. Alle groepsdocumenten worden in google docs gemaakt: <https://drive.google.com/drive/folders/1a0P3XgN7i27Lr5HU7Lx8YZL06HcfSJxI>

**2. Personal:**

## 0. Sprint deliveries

|  |  |
| --- | --- |
| **Naam document** | Sprint delivery presentaties |
| **Type document** | Pptx |
| **Beschrijving** | Sprint delivery 1 release note. |

|  |  |
| --- | --- |
| **Naam document** | Sprint delivery notes |
| **Type document** | Docx |
| **Beschrijving** | Sprint delivery 1 release note. |

## 1. Pitch

|  |  |
| --- | --- |
| **Naam document** | Pitch |
| **Type document** | Pptx |
| **Beschrijving** | Pitch voor PolarGram. |

|  |  |
| --- | --- |
| **Naam document** | Design |
| **Type document** | Png |
| **Beschrijving** | Eerste design PolarGram, meegeleverd met de pitch. |

|  |  |
| --- | --- |
| **Naam document** | Favicon.ico / PolarGramLogo |
| **Type document** | Ico / png |
| **Beschrijving** | Logo polarGram. |

## 2. Analyse

|  |  |
| --- | --- |
| **Map** | 1. C-models |
| **Document** | Alle gemaakte C-models |
| **Type document** | Png / Draw.io |
| **Beschrijving** | C-models in png + C-models in draw.io format. Overzicht architectuur PolarGram |

|  |  |
| --- | --- |
| **Document** | Analyse |
| **Type document** | Docx |
| **Beschrijving** | Bevat functional en non-functional requirements. |

## 3. Onderzoek

|  |  |
| --- | --- |
| **Document** | Emerging Trends Research Plan |
| **Type document** | Docx |
| **Beschrijving** | Research plan voor het onderzoek naar emerging trends. Mijn gekozen onderwerp: Domain-Driven Design. |

|  |  |
| --- | --- |
| **Document** | RabbitMQ |
| **Type document** | Docx |
| **Beschrijving** | Onderzoek naar RabbitMQ en andere message busses. |

|  |  |
| --- | --- |
| **Document** | Emerging Trends Research |
| **Type document** | Docx |
| **Beschrijving** | Emerging trends research naar Domain-Driven Design. |

|  |  |
| --- | --- |
| **Document** | OWASP Top 10 |
| **Type document** | Docx |
| **Beschrijving** | Onderzoek naar OWASP Top 10, met bijgevoegde eigen behandelde onderwerpen. |

|  |  |
| --- | --- |
| **Document** | GDPR |
| **Type document** | Docx |
| **Beschrijving** | Onderzoek naar GDPR, met bijgevoegde eigen behandelde onderwerpen. |

## 4. Polargram

|  |  |
| --- | --- |
| **Document** | Front-end |
| **Type document** | Dir |
| **Beschrijving** | Front-end PolarGram. |