

# Portfolio Stage Jaar 3

Douwe Zumker 1059179



Hogeschool Rotterdam  
Informatica  
Harborn Digital [1]

# Portfolio derde jaars stage bij Harborn digital

Opleiding: Informatica  
Naam: Douwe Zumker(1059179)  
E-mail: 1059179@hr.nl  
Begeleider: Gijs Straathof  
Docent: Casper Smits

Plaats en datum: Rotterdam 15 januari 2025  
Versie: 0.10

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>3</b>
<b>2</b>	<b>Stagevoorstel</b>	<b>4</b>
2.1	Competenties . . . . .	4
2.1.1	Manage & Control . . . . .	4
2.1.2	Analyseren . . . . .	4
2.1.3	Adviseren . . . . .	4
2.1.4	Ontwerpen . . . . .	4
2.1.5	Realiseren . . . . .	4
2.1.6	Professional Skills . . . . .	5
2.2	Globale planning versie 1 . . . . .	6
2.3	Planning versie 2 . . . . .	7
2.4	Bewijslast per competentie . . . . .	7
<b>3</b>	<b>Manage &amp; Control</b>	<b>8</b>
3.1	Inleiding . . . . .	8
3.2	Bewijs . . . . .	8
3.2.1	Feedbackformulier . . . . .	8
3.3	Reflectie . . . . .	9
<b>4</b>	<b>Analyseren</b>	<b>11</b>
4.1	Inleiding . . . . .	11
4.2	Bewijs . . . . .	11
4.3	Feedback . . . . .	12
4.4	Reflectie . . . . .	12
<b>5</b>	<b>Adviseren</b>	<b>14</b>
5.1	Inleiding . . . . .	14
5.2	Bewijs . . . . .	14
5.3	Feedback . . . . .	14
5.4	Reflectie . . . . .	15
<b>6</b>	<b>Ontwerpen</b>	<b>16</b>
6.1	Inleiding . . . . .	16
6.2	Bewijs . . . . .	16
6.3	Feedback . . . . .	16
6.4	Reflectie . . . . .	17
<b>7</b>	<b>Realiseren</b>	<b>19</b>
7.1	Inleiding . . . . .	19
7.2	Bewijs . . . . .	19
7.3	Feedback . . . . .	20
7.4	Reflectie . . . . .	21
<b>8</b>	<b>Professional Skills</b>	<b>23</b>
8.1	Inleiding . . . . .	23

8.2	Bewijs . . . . .	23
8.3	Feedback . . . . .	23
8.4	Reflectie . . . . .	24
<b>9</b>	<b>Overkoepelende Reflectie</b>	<b>25</b>
<b>10</b>	<b>Advies bedrijfsbegeleider</b>	<b>26</b>
<b>11</b>	<b>Bijlagen</b>	<b>31</b>
I	User Story . . . . .	31
II	Planning . . . . .	32
III	OpenID implementatie . . . . .	33
IV	Unittest . . . . .	35
V	GitHub Actions . . . . .	40
VI	Interface . . . . .	43
VII	BLOB Client . . . . .	44
VIII	Terraform IaC . . . . .	47
IX	Aflevering . . . . .	48
X	Zelfreflecties van de eerste maand . . . . .	49
XI	Analyse . . . . .	50
XII	Advies . . . . .	62
XIII	Sequence Diagram . . . . .	69
XIV	ER Diagram . . . . .	70
XV	Class Diagram . . . . .	71

# Hoofdstuk 1

## Inleiding

Mijn naam is Douwe Zumker en ik ben een derde jaars student Informatica aan de Hogeschool Rotterdam. In de afgelopen 20 weken mocht ik bij Harborn digital te Rotterdam aan de slag als stagiair. Mijn stage had geen opdracht, maar ik mocht meelopen met een team om aan verschillende projecten te zitten om zo meer ervaring op te doen. Daarnaast mag ik met trots zeggen dat ik na mijn stage parttime aan de slag ga als werkstudent.

Aan het begin van de stage had ik zelf voor de zes verschillende competenties Manage & Control, Analyseren, Adviseren, Ontwerpen, Realiseren en Professional skills leerdoelen opgesteld die ik na een periode van 20 weken wil behalen. Dit kan je mooi teruglezen in hoofdstuk 2. In dit hoofdstuk staan ook twee tabellen met de verschillende planningen die ik voor mijzelf had gemaakt en een tabel met de verschillende vormen van bewijslast die ik per competentie heb verzameld.

Dit verslag documenteert mijn leerproces, prestaties, feedback en reflecties binnen de verschillende competenties. Elk hoofdstuk heeft nog een kleine inleiding waarin het opgestelde leerdoel en de vormen van bewijslast nog worden herhaald. Aan het einde van elk hoofdstuk is een reflectie op het leerdoel, de feedback en het leerproces.

Met deze inleiding hoop ik de lezer een helder beeld te geven van de inhoud en opzet van mijn portfolio. Het dient niet alleen als een bewijs van mijn ontwikkeling, maar ook als bron van inspiratie voor tijdens mijn afstudeerstage.

# Hoofdstuk 2

## Stagevoorstel

### 2.1 Competenties

#### 2.1.1 Manage & Control

Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om betere inschattingen te maken over de voortgang van een project en de tijd die ik nodig heb voor verschillende taken. Ik ga dit bereiken door actief deel te nemen aan refinements en inschatting sessies, zodat ik de complexiteit van features beter kan inschatten. Dit houdt in dat ik tijdens inschatting-sessies plus minus 1 punt zit van een inschatting in vergelijking met de senior developers. Ik ben tevreden wanneer ik verschillende feature tijdens refinements beter kan inschatten, waarbij ik dan meer in lijn wil zijn met de developers die bij Harborn digital werken. Daarnaast wil ik tenminste 80% van de taken binnen de aangegeven tijd afronden.

#### 2.1.2 Analyseren

Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn de huidige implementatie van een stabiele applicatie te analyseren om te beoordelen of deze nog voldoet aan de veranderende of vernieuwde standaarden. Ik ga dit bereiken door de applicatie grondig te evalueren aan de hand van de geldende technische en veiligheidsnormen, en de resultaten te verwerken in een analyse. Ik ben tevreden wanneer mijn analyse duidelijke verbeterpunten oplevert en wordt gebruikt in het adviesrapport voor de verbetering van de applicatie.

#### 2.1.3 Adviseren

Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een goed onderbouwd advies te kunnen geven over de selectie van softwareonderdelen van een bestaande applicatie. Ik ga dit bereiken door middel van kostenoverweging en een performance-analyse, en door de resultaten in een adviesrapport te verwerken. Ik ben tevreden wanneer ik een compleet rapport heb geschreven dat gebruikt kan worden om de onderzochte softwareonderdelen te verbeteren.

#### 2.1.4 Ontwerpen

Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een volledig en compleet verbeteringsplan op te stellen voor een nieuw of bestaand softwaresysteem, waarbij ik zowel nieuwe als bestaande componenten meeneem in het ontwerp. Hierbij houd ik rekening met de eisen van de belanghebbenden en met de veiligheids- en schaalbaarheidsnormen die gelden binnen het stagebedrijf. Ik ben tevreden wanneer het verbeteringsplan goedgekeurd wordt en als basis kan dienen voor verdere ontwikkelingen binnen het project.

#### 2.1.5 Realiseren

Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een zelfgebouwde module succesvol te integreren met een bestaand systeem door de bestaande codebase te analyseren en aanpassingen te maken met minimale impact. Ik ben tevreden wanneer mijn code de acceptatie-omgeving doorstaat en in gebruik wordt genomen door de klant.

### **2.1.6 Professional Skills**

Ik wil na afloop van mijn stageperiode van 20 weken instaat zijn om effectief en zelfstandig te werken op het niveau van een junior medewerker binnen het stagebedrijf, door actief hulp en feedback te vragen aan collegas en uit mijn comfortzone te stappen. Ik ben tevreden wanneer ik zonder moeite weet naar wie ik toe kan stappen voor welke soort vragen.

## 2.2 Globale planning versie 1

Week	Taken	Bewijslast	Doorlopend mee bezig	Iedere week
1	Realiseren (syncforce)	Code, Unittests, User stories, Reflectie refinement en eigen initiatief	Professional skills en Manage & Control	Reflectie van refinement sessies met hoe het inschatten gaat en of ik beter word in het inschatten van hoe lang iets gaat duren.
2	Realiseren	Code, Unittests, User stories, Terraform templating code, Reflectie refinement	Professional skills en Manage & Control	Reflectie van hoe goed ik op mijn collega's afstap, waarbij ik rekening moet houden met wie welke kennis heeft
3	Realiseren	Code, Unittests, User stories, Reflectie refinement	Professional skills en Manage & Control	
4	Realiseren	Code, Unittests, User stories, Afleveren van product bij de klant, Reflectie schrijven van de afgelopen weken	Professional skills en Manage & Control	
5	Analyseren van bestaande feature		Professional skills en Manage & Control	
6	Analyseren van bestaande feature	Begin van Analyse rapport	Professional skills en Manage & Control	
7	Analyseren van bestaande feature	Analyse rapport, Reflectie schrijven van hoe het analyseren is gegaan	Professional skills en Manage & Control	
8	Adviseren op basis van de gedane analyses	Begin van Advies rapport, Maken van midterm presentatie voor week 9	Professional skills en Manage & Control	
9	Adviseren op basis van de gedane analyses	Advies rapport, Presentatie, Reflectie schrijven van Adviseren	Professional skills en Manage & Control	
10	Professional skills	Feedback verzamelen van collega's	Professional skills en Manage & Control	
11	Ontwerpen	Transcript van vergadering/en met belanghebbers	Professional skills en Manage & Control	
12	Ontwerpen	Eerste begin van Software Requirements Specification (SRS)	Professional skills en Manage & Control	
13	Ontwerpen	SRS zou nu af moeten zijn, Begin aan software ontwerpen	Professional skills en Manage & Control	
14	Ontwerpen	Software ontwerpen, ERD, UML	Professional skills en Manage & Control	
15	Manage & Control	Terugkijken naar hoe de verschillende refinement sessies zijn gegaan	Professional skills en Manage & Control	
16	Professional skills	Conceptversie van rapport zoveel mogelijk afhebben	Professional skills en Manage & Control	
17	Professional skills	De opmerkingen verwerken van het stagerapport	Professional skills en Manage & Control	
18	Professional skills	Afmaken van het hele stage rapport, Eindpresentatie maken	Professional skills en Manage & Control	



## 2.3 Planning versie 2

Datum	Week	Bewijslast/Waar ik aan ga werken
11-11-2024	11	Presentatie voor de midterm presentatie; <b>Ontwerpen:</b> Verbeterplan voor een bestaand systeem, ER Diagram, Sequence Diagram en UML Diagram; <b>Manage &amp; Control:</b> User Stories opstellen en inschatten voor verbeterplan van een bestaand systeem; Ook het maken/updaten van de planning
18-11-2024	12	<b>Ontwerpen/Professional Skills:</b> Aan de klant het voorstel voorleggen en het ontwerp afmaken, daarnaast dit alles van de afgelopen weken mooi in het stage rapport vastleggen en feedback vragen van de bewijslast
25-11-2024	13	<b>Ontwerpen:</b> Als er nog dingen veranderd moeten worden van het plan naar aanleiding van de feedback van de klant zal ik dit proberen te verbeteren; Realiseren en andere leerdoelen: Technisch gezien is het bewijsmateriaal al verzameld, dus ik zal aan de slag gaan met realiseren en de andere leerdoelen waar mogelijk
2-12-2024	14	<b>Alle leerdoelen:</b> Ik zal verder gaan aan alle leerdoelen en feedback verzamelen als dat van toepassing is, waarschijnlijk vooral bezig aan realiseren; Als er extra bewijslast voor de andere leerdoelen kan komen, neem ik dit mee in het stagerapport
9-12-2024	15	<b>Alle leerdoelen:</b> Ik zal verder gaan aan alle leerdoelen en feedback verzamelen als dat van toepassing is, waarschijnlijk vooral bezig aan realiseren; Als er extra bewijslast voor de andere leerdoelen kan komen, neem ik dit mee in het stagerapport
16-12-2024	16	<b>Alle leerdoelen:</b> Ik zal verder gaan aan alle leerdoelen en feedback verzamelen als dat van toepassing is, waarschijnlijk vooral bezig aan realiseren; Als er extra bewijslast voor de andere leerdoelen kan komen, neem ik dit mee in het stagerapport
23-12-2024	Vakantie	Weekje kerstvakantie
30-12-2024	17	<b>Feedback verzamelen/verwerken:</b> Laatste feedback verzamelen en verwerken in het stagerapport; Reflecteren op de stage, professionele ontwikkeling, en inschattingen van taken en manage & control
6-1-2025	18	<b>Bufferweek/Afmaken</b> van stagerapport en presentatie
13-1-2025	19	<b>Bufferweek:</b> Voor ziekte of afronding van stagerapport en presentatie

## 2.4 Bewijslast per competentie

Bewijslast	Competenties					
	Manage & Control	Analyseren	Adviseren	Ontwerpen	Realiseren	Professional skills
Planning (possibly multiple versions)	X					
Software ontwerpen (ERD, UML Class Diagram, ...)				X		
Analyse Rapport		X				
Adviesrapport			X			
Source Code					X	
Unit Tests					X	
Feedbackformulieren	X					X

# Hoofdstuk 3

## Manage & Control

### 3.1 Inleiding

In dit hoofdstuk ga ik bewijzen dat ik de competentie Manage & Control tijdens mijn stage heb weten te behalen. Dit ga ik doen door middel van de volgende soort bewijsstukken: Feedbackformulieren, planning en een User story.

Ter herinnering, in mijn stagevoorstel had ik dit leerdoel opgesteld: „Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om betere inschattingen te maken over de voortgang van een project en de tijd die ik nodig heb voor verschillende taken. Ik ga dit bereiken door actief deel te nemen aan refinements en inschatting sessies, zodat ik de complexiteit van features beter kan inschatten. Dit houdt in dat ik tijdens inschatting-sessies plus minus 1 punt zit van een inschatting in vergelijking met de senior developers. Ik ben tevreden wanneer ik verschillende feature tijdens refinements beter kan inschatten, waarbij ik dan meer in lijn wil zijn met de developers die bij Harborn digital werken. Daarnaast wil ik tenminste 80% van de taken binnen de aangegeven tijd afronden.”

Ik heb iedere keer als er een inschatting sessie was, binnen het team waar ik bij ben gezet, zo actief mogelijk meegedaan. Om zo uiteindelijk een beter gevoel te krijgen met hoelang je over bepaalde taken hoort te doen. Dit is dan wel bij relevante projecten waar ik zelf meer over weet zodat ik niet blind de refinement sessies inga.

### 3.2 Bewijs

Ik was toen ik bezig was met het realiseren van code om over te stappen van AWS naar Azure, zie hoofdstuk 7 op een raar probleem gestuit. Eerst probeerde ik zelf door veel verschillende manieren het probleem te achterhalen/oplossen, maar niks mocht baten. Dus heb ik het probleem zo goed mogelijk geprobeerd te omschrijven in een User story en het op het scrum bord gezet, zoals je kan zien in bijlage I. Daarnaast heb ik verschillende user stories opgesteld nadat ik het ontwerp in het hoofdstuk 6 een volledig ontwerp heb opgesteld. De user stories die je in bijlage II kan zien hadden we tijdens een refinement opgesteld en worden dan in een sprint getrokken om aan te werken. Een punt staat voor ongeveer 3 tot 4 uur werk.

#### 3.2.1 Feedbackformulier

Ik heb in Google Forms een feedback formulier gemaakt met de volgende vragen:

1. Wat is je rol of relatie tot mijn stageproject?
2. Hoe beoordeel je de nauwkeurigheid van mijn inschattingen in vergelijking met eerdere inschattingen?
3. In welke mate ben ik in staat om de complexiteit van features in te schatten tijdens refinement-sessies?
4. Hoe vaak zijn mijn inschattingen correct gebleken ten opzichte van de werkelijke tijdsduur?

5. Draag ik actief bij aan refinement-sessies door relevante vragen te stellen over requirements en complexiteit?
6. Begrijp ik het doel van elke feature voldoende om een correcte inschatting te maken?
7. Wat zie jij als mijn sterkste punt in het inschatten van taken of features?
8. Wat zou ik kunnen verbeteren om mijn inschattingen nog nauwkeuriger te maken?
9. Eventuele verdere opmerkingen over mijn deelname in het team en bijdrage aan projecten?

Hier heb ik feedback op gekregen en het per persoon samengevat.

Van mijn collega Sander heb ik feedback gekregen over mijn inschattingen en deelnames aan inschatting en refinement sessies. Ik krijg positieve feedback op mijn nauwkeurigheid in het inschatten van taken, met een score van 4/5. Ik ben redelijk goed in staat om de complexiteit van features in te schatten (score 3/5), en mijn inschattingen blijken vaak accuraat te zijn. Ik draag actief bij aan refinement sessies door relevante vragen te stellen, en begrijp meestal het doel van features voldoende om correcte inschattingen te maken. Mijn sterke punten zijn mijn nauwkeurige werkwijze en zorgvuldige documentatie. De suggestie is om door te gaan met mijn huidige aanpak, omdat ik dingen snel oppak en goed werk aflever.

Van mijn begeleider krijg ik over het algemeen positieve feedback op mijn rol als stagiair, vooral op mijn inschattingen en bijdragen aan refinement-sessies. Hij waardeert mijn nauwkeurigheid en vindt dat ik over het algemeen goed begrijp wat de doelstellingen van features zijn, waardoor ik correcte inschattingen kan maken. Mijn sterkste punt volgens hem is mijn vermogen om snel in te zien hoe ik een taak kan aanpakken. Wel raadt hij me aan om vaker alternatieve aanpakken te overwegen en zorgvuldig na te gaan of ik alle randvoorwaarden van een feature heb meegenomen.

### 3.3 Reflectie

De bovenstaande feedback die ik heb gekregen van mijn collega Sander en bedrijfsbegeleider Gijs, heeft mij inzicht gegeven in mijn sterke kanten zoals nauwkeurigheid met inschattingen en actieve deelname aan de inschatting-sessies. Daarnaast kom ik ook voorbereid de refinement sessies binnen en weet ik waar ik over praat. Ook is de documentatie die ik heb geschreven voor het implementeren van nieuwe features zorgvuldig waardoor men ook beter inschattingen kan maken. En ik heb snel inzicht in hoe ik een taak kan aanpakken. Daarnaast als we gaan poken voor de hoeveelheid tijd die we nodig hebben om een bepaalde taak uit te voeren lig ik ook vaak op één lijn met de andere developers. Wat ik kan verbeteren is vaker alternatieve aanpakken van een probleem overwegen en zorgvuldig na te gaan of ik alle randvoorwaarden heb meegenomen om een completer beeld te vormen van een feature tijdens het inschattingsproces.

Op basis van de feedback die ik heb gekregen heb ik mij voorgenomen om tijdens referiments bewuster stil te staan bij mogelijke alternatieven voor het aanpakken van taken en om vragen die ik heb gericht te stellen op randvoorwaarden die ik mogelijk over het hoofd zie. Daarnaast blijf ik mijn huidige aanpak volgen, omdat ik merk dat deze effectief is.

Ik heb qua werkzaamheden binnen dit leerdoel geleerd dat het belangrijk is om context te weten over features waar je inschattingen over maakt. Zodat je dan accuratere inschattingen kan maken, wat niet alleen voor mij maar ook mijn team fijn is. Ook is het niet erg om vragen te stellen als je meer context nodig hebt. Soms kan het zijn dat er met de informatie die we op het moment van inschatten hadden dat we een bepaalde tijd inschatten over de duur van een taak. Dat er toch blijkt dat er meer tijd aan besteed moet worden omdat er dan extra informatie bij is gekomen. Dit betekent dan niet dat een taak slecht is ingeschat of slecht is uitgevoerd aangezien we uitgingen

van de gegeven informatie voor een taak. Uiteindelijk heb ik 17 user stories van het bord opgepakt, waarvan ik 14 binnen de aangegeven tijd en dus op tijd voor de oplevering van de verschillende projecten heb afgerond. Wat betekent dat ik 82% van mijn taken binnen de aangegeven tijd heb afgerond.

Mijn vertrekpunt was een basisvaardigheid in het inschatten van taken, maar beperkte ervaring op de complexiteit binnen features. En was het lastig om actief mee te doen aan inschatting sessies. Inmiddels snap ik veel beter de complexiteit van features en hoelang er gemiddeld over gedaan wordt, ook doe ik nu actief mee aan inschatting sessies en ben ik er veel beter op voorbereid.

Een belangrijk ontwikkelpunt is om dieper in te gaan op de randvoorwaarden van features zodat ik efficiëntere inschattingen kan maken.

In de toekomst zal ik de zorgvuldige aanpak en documentatie vaardigheden die ik nu beheers blijven toepassen en verbeteren, zodat ik accurate inschattingen kan blijven maken. Tijdens mijn afstuderen gaat het dus goed zijn om een uitgebreide analyse fase te hebben, zodat ik voordat ik een planning maak precies begrijpt wat er van mij wordt verwacht. Dit omvat de opdracht en deliverables, tussentijdse deadlines en beoordelingscriteria en afhankelijkheden van anderen zoals begeleiders, tools of externe partijen. Ook ga ik een iteratieve aanpak hebben met een eerste planning en evalueer deze regelmatig om met feedback te verfijnen en aan te passen wanneer er nieuw informatie beschikbaar komt. Een buffer opnemen is ook van belang aangezien er onverwachte uitdagingen kunnen komen (zoals technische problemen, extra onderzoek of feedback) door extra tijd in te bouwen voor belangrijke taken. En als laatste is duidelijke communicatie belangrijk, door te communiceren met mijn begeleiders over mijn aanpak, voortgang en eventuele knelpunten. Dit helpt om misverstanden te voorkomen en vertrouwen te creëren.

Als ik de bovengenoemde punten meeneem, denk ik dat ik mijn zelf opgestelde leerdoel heb weten te behalen. Ik ben nu beter in staat om complexe features te analyseren, inschattingen te maken en actief bij te dragen aan inschatting-sessies. Ook lukt het mij steeds beter om taken binnen de geschatte tijd af te ronden.

# Hoofdstuk 4

## Analyseren

### 4.1 Inleiding

In dit hoofdstuk ga ik door middel van een analyse rapport bewijzen dat ik de volgende doelstelling heb behaald. „Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn de huidige implementatie van een stabiele applicatie te analyseren om te beoordelen of deze nog voldoet aan de veranderende of vernieuwde standaarden. Ik ga dit bereiken door de applicatie grondig te evalueren aan de hand van de geldende technische en veiligheidsnormen, en de resultaten te verwerken in een analyse. Ik ben tevreden wanneer mijn analyse duidelijke verbeterpunten oplevert en wordt gebruikt in het adviesrapport voor de verbetering van de applicatie.”

Ik heb een klein onderzoek mogen doen over Giraf, een tool dat mijn stagebedrijf aanbied aan vastgoedmakelaars en verhuurders om makkelijker en sneller te kunnen reageren op e-mails. Het is een AI-tool dat ze in 2021 hebben ontwikkeld en sindsdien ook aanbieden aan hun klanten en mijn taak was om te analyseren of de prestaties nog wel meegaan met haar tijd.

### 4.2 Bewijs

In bijlage XI staat de hele analyse van de AI-service Giraf. Tijdens het onderzoek heb ik een grondige analyse gedaan van de AI-service en vervolgens een vergelijking gemaakt met een lokale LLM, denk hierbij aan Chat GPT, maar dan op je eigen computer. Om een goeie, maar niet al te grote samplesize te hebben, heb ik 200 e-mails uit een database opgehaald. Vervolgens heb ik zelf eerst beoordeeld of een e-mail gaat over een bezichtiging of iets anders. Toen ik dat had gedaan liet ik de AI-service en een lokale LLM de classificatie erop uitgevoerd. Alle resultaten had ik in een JSON bestand gezet om met code makkelijk te kunnen zien in hoeveel procent van de gevallen de classificatie correct was gegaan. Toen ik dit had gedaan heb ik een Named Entity Recognition (NER)-operatie uitgevoerd op de e-mails. Een NER-operatie kan entiteiten in ongestructureerde tekst identificeren en categoriseren. Bijvoorbeeld: personen, plaatsen, organisaties en hoeveelheden. Dit heb ik door de LLM en door de AI-service laten uitvoeren om vervolgens de resultaten in een JSON bestand op te slaan. Vervolgens ben ik handmatig langs elke mail gegaan en heb ik beoordeeld hoe correct de NER gegevens zijn die uit zo'n mailtje zijn gehaald. Daarnaast heb ik ook een kostenanalyse gedaan om te kijken hoe duur het is om die AI-service draaiende te houden in AWS. En ik heb op de website van AWS gekeken waar de prijzen zijn vermeld voor alle verschillende instances die je kan huren.

Zoals te zien is in hoofdstuk 5 en bijlage XII worden de resultaten uit de analyse gebruikt voor het opbouwen van een advies.

Het analyseren heb ik natuurlijk niet alleen gedaan bij Giraf, maar ook bij andere projecten waar ik aan mee moest gaan werken. Sommige projecten hebben namelijk al een hele codebase die al een tijd lang is opgezet waar ik geen verstand van heb. Dus voordat ik aan de slag kon heb ik eerst de structuur en code geanalyseerd en vragen gesteld als ik dat niet wist. Voor een bedrijf heb ik een API endpoint moeten implementeren waar Single Sign On (SSO) op staat met de OpenID implementatie die zij zelf hebben gebouwd. Normaal gesproken is zo'n implementatie standaard en volgt het de standaard dat ze hebben gekozen om als basis te gebruiken. Maar dat was helaas niet het geval, dus heb ik handmatig een implementatie in python moeten schrijven zie bijlage III. Ik heb namelijk in de browser console geanalyseerd wat er in een inlog request bij een website zit en

op die manier stapgewijs de implementatie nagebootst en geschreven. Normaal gesproken kan je dus gebruik maken van een voorgebouwde library, die dit abstraheerd, alleen kon dit niet doordat de SSO provider het ietsje anders had gemaakt.

### 4.3 Feedback

Tijdens het schrijven van de analyse heb ik advies gevraagd aan mijn stagebegeleider met wat ik eraan kan verbeteren. Dit heb ik in de vorm van reacties in een Google Docs document gekregen en ook meteen verwerkt en aangepast.

- Geef wat meer uitleg over wat Giraf is en doet, inclusief een toelichting van wat de gevolgen zijn van een onderpresterende AI service.
- Ik mis een basis waarop je dit concludeert, je toont geen analyse van kosten van alternatieve oplossingen, dus hoe weet je dat dit niet beter kan?
- Je trekt nog al snel een conclusie terwijl je dit eigenlijk pas later moet doen.
- De lezer kent deze naam van de database niet, ik zou dit beschrijven in plaats van de database naam gebruiken.
- Ik zou minder nadruk leggen op kosten: het gaat in eerste instantie om prestatie optimalisatie. Daarbij moeten kosten in ogenschouw genomen worden.
- Laat meer zien hoe je tot een bepaalde keuze bent gekomen, je identificeert een aantal alternatieven, daarna selecteer je de beste/meest realistische.

En hieronder heb ik de feedback die ik heb gevraagd na het verbeteren van het analyserapport. „Ik vind het een mooie analyse van de AI module van Giraf. Het zou waardevol zijn geweest als er meerdere alternatieven benoemd waren; al is de keuze voor het uiteindelijke alternatief begrijpelijk gezien de recente ontwikkelingen rondom large language models (LLMs). In de kostenanalyse wordt een prijs genoemd voor het hosten van een alternatief model voor LLMs, variërend van \$0,33 tot \$40 per uur. Hier had ik graag wat meer context gezien over de reden waarom deze prijsrange zo groot is. Al met al betreft het een mooie analyse met een conclusie waar we verder mee kunnen. Het is bovendien goed om te zien dat de zelfgetrainde modellen voor classificatie nog steeds lijken te voldoen aan de verwachtingen.”

### 4.4 Reflectie

Ik heb van de feedback die ik had gevraagd tijdens het schrijven van de analyse geleerd dat ik meer uitleg voor de lezer moet geven over wat ik onderzoek en waarom. Ook moest ik wat meer uitleggen hoe ik tot bepaalde conclusies kom. Ik had de neiging een conclusie te trekken zonder de lezer enig bewijs te geven, terwijl ik hier wel over na had gedacht. Sommige dingen of context is helemaal niet relevant voor de lezer en andere achtergelaten context is juist weer wel nodig. Ik moet ook meer gefocust blijven op waar het nou echt om gaat bij een analyse, de kosten was iets wat in ogenschouw genomen moest worden genomen. En laat ook zien hoe je tot bepaalde keuzes bent gekomen.

Ik was het ook eens met de bovenstaande feedback die ik had gekregen en om een nog betere analyse te schrijven heb ik deze feedback dan ook meteen zoveel mogelijk geprobeerd toe te passen. Dus ik heb meer achtergrondinformatie gegeven over Giraf en uitgelegd wat de gevolgen zijn van

een onderpresterende AI. Bij bijvoorbeeld de kosten had ik eerst het volgende staan: „Hier zullen dus geen verbeteringen op gemaakt kunnen worden”, nu heb ik als voorbeeld de kosten van een andere service erbij gepakt en heb ik pas in een later hoofdstuk een conclusie getrokken. Ook heb ik meer uitleg gegeven waarom ik voor een self-hosted LLM heb gekozen om de vergelijking mee te maken met de AI-service, deze keuze had ik eerst helemaal niet onderbouwd en nu wel. En de kosten had ik eerst centraler in het rapport staan, maar dat hoorde eigenlijk niet de focus te zijn, dus heb ik het uit de titel gehaald en het na de prestatieanalyse geplaatst.

Mijn vertrekpunt was een basiskennis in het schrijven van een analyserapport en in het analyseren van code, tools en kosten. Dat heb ik wel een beetje moeten doen voor Project C, maar daarna en daarvoor heb ik er vaak niet bij stil gestaan. Ik ben inmiddels gegroeid in het schrijven van een goed onderbouwd analyserapport en in het toepassen van de feedback om mijn rapport nog beter te schrijven. Ook heb ik nu meer kennis in hoe en wat ik moet analyseren.

De volgende keer als ik meer tijd heb om te onderzoeken wil ik meer kijken naar verschillende alternatieven en dit ook meenemen. Ook wil ik meer context geven bij de kosten voor het hosten van LLM's. Wel ben ik blij met de uiteindelijke conclusie die ik heb getrokken en dat het ook verder gebruikt kan worden door mijn stagebedrijf.

In het begin had ik veel moeite met hoe ik moest beginnen aan een goede analyse van een bepaalde software component, ik was kort door de bocht met veel uitleg die ik gaf en dat was niet altijd even goed. Zoals je in de bijlage XI kan zien heb ik dit ook echt verbeterd en voelt het ook goed gestructureerd.

Wat ik dus ook zeker wil meenemen naar de toekomst voor bijvoorbeeld het onderzoekgedeelte van een afstudeerstage, is dat ik meer mijn keuzes en conclusies moet onderbouwen door meer context te geven en niet te snel denken dat het wel goed is. Bij de eerste opzet van mijn analyse was dit wat er lastig was, aangezien mijn aanpak in het begin niet heel erg gestructureerd was. Waardoor ik snel van de hak op de tak sprong tussen verschillende onderdelen van het onderzoek. De volgende keer wil ik dus ook wat meer rust bouwen in het hele proces en beter na denken over bepaalde conclusies die ik trek.

Concluderend denk ik dat ik dit leerdoel behaald heb. Met een gestructureerde aanpak heb ik de AI-tool Giraf geanalyseerd en belangrijke verbeterpunten geïdentificeerd. Dit heb ik gedaan door een vergelijking te maken met een lokaal LLM-model, een gedetailleerde prestatieanalyse uit te voeren en een kostenvergelijking te doen. Hierdoor heb ik de tool grondig kunnen analyseren. Daarnaast heeft het verwerken van de feedback en het herschrijven van mijn rapport geleid tot een beter gestructureerd en onderbouwd document.

# Hoofdstuk 5

## Adviseren

### 5.1 Inleiding

In dit hoofdstuk ga ik door middel van een adviesrapport bewijzen dat ik mijn zelf opgestelde leerdoel heb behaald. „Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een goed onderbouwd advies te kunnen geven over de selectie van softwareonderdelen van een bestaande applicatie. Ik ga dit bereiken door middel van kostenoverweging en een performance-analyse, en door de resultaten in een adviesrapport te verwerken. Ik ben tevreden wanneer ik een compleet rapport heb geschreven dat gebruikt kan worden om de onderzochte softwareonderdelen te verbeteren.”

In het bewijs gedeelte ga ik een uitleg geven over het advies rapport en hoe het is opgebouwd. Als ik naast dit adviesrapport nog bewijsstukken heb die bijdragen aan dit hoofdstuk zal ik het ook toevoegen.

### 5.2 Bewijs

Geadviseerd over het in gebruik nemen van een goedkopere virtual machine instance in Azure voor het besparen van kosten. Dit is ook geïmplementeerd en in gebruik genomen. Ik had tijdens het realiseren iets ontdekt, het serviceplan dat Harborn gebruikte voor Linux web apps maakte namelijk gebruik van een oude versie van virtual machines. Deze oudere versie was ietsjes duurder dan de nieuwere versie en ik heb geadviseerd om de iets goedkopere, maar betere virtual machine in gebruik te nemen.

Het adviesrapport, zie bijlage XII, dat ik heb geschreven bouwt verder op de analyse die ik heb gedaan, waarbij ik heb gekeken naar de AI-service Giraf. Zie hoofdstuk 4 voor meer informatie. In de inleiding geef ik uitleg over wat Giraf is, waarom er een analyse is gedaan en dat het adviesrapport verder bouwt op wat er in het analyse rapport is geschreven. In het volgende hoofdstuk geef ik een korte samenvatting van het analyserapport en geef ik kort de resultaten van het gedane onderzoek. Verder verwijs ik de lezer naar het analyserapport als ze wat meer informatie willen dan in de samenvatting vermeld staat. In het hoofdstuk daarna geef ik uitleg over de AI Act, een wetgeving van de Europese Unie met betrekking tot veilige AI's. En wat dat voor effect heeft op de implementatie van een AI en/of LLM. Daarnaast benoem ik ook software licenties en wat dat voor effect heeft op het gebruiken van een LLM die we niet zelf bouwen. Als laatste geef ik dan een aanbeveling/advies aan de lezer.

### 5.3 Feedback

Tijdens het schrijven van het adviesrapport heb ik aan mijn stagebegeleider gevraagd wat ik eraan kan verbeteren. Dit heb ik in de vorm van reacties in een Google Docs document gekregen en ook meteen verwerkt en aangepast.

- Ik zou de titel, omdat het een advies is, niet als vraag formuleren.
- Het is aan de lezer/beslisser om dit te concluderen, ik zou dit meer als advies formuleren, of als een mening.



- De samenvatting van de analyse is vrij uitgebreid voor dit rapport: je moet voorkomen dat dit een herhaling is van het andere rapport.
- Maak de ondertitel wat inhoudelijker.

En hieronder heb ik de feedback die ik heb gevraagd na het verbeteren van het adviesrapport. „Het advies bouwt op een logische manier voort op het analyserapport. De beperkingen rondom het gebruik van AI-modellen binnen de EU vormen een belangrijk aspect, en het is positief dat deze in het advies zijn meegenomen. Helaas blijft het advies hierdoor wat algemeen, en zal er eerst een onderzoek moeten plaatsvinden naar beschikbare LLM-modellen die geschikt zijn voor de NER-taak binnen de AI-module.”

## 5.4 Reflectie

Naar aanleiding van de feedback die ik had gevraagd tijdens het schrijven van het advies heb ik geleerd dat je bij het schrijven van een samenvatting alles wat korter mag schrijven en mag verwijzen naar een eerder geschreven rapport. Eerst was mijn "samenvatting" bijna 2 pagina's lang, nu is het enorm erg ingekort en is het minder dan 1 pagina lang en staan er ook alleen de resultaten beschreven. Als de lezer meer context wil of meer details wil weten verwijs ik ze door naar het analyserapport. Daarnaast stelde ik mijn mening op als feit, terwijl de lezer zelf ook een uiteindelijke keuze moet maken op basis van de aanbeveling, dus heb ik dit anders geprobeerd te formuleren. Verder had ik de titel eerst als vraag opgesteld, terwijl de vraag gesteld was in het analyserapport en het adviesrapport geeft juist verschillende mogelijkheden tot verbetering.

Tijdens het werken aan dit leerdoel heb ik geleerd hoe belangrijk het is om een advies duidelijk en goed te onderbouwen met analyses, zoals kostenoverwegingen en performance-analyses. Ook heb ik beter geleerd hoe ik een bondige en volledige samenvatting het beste kan formuleren. Daarnaast kan ik feedback nu ook goed in me opnemen om mijn geschreven tekst te verbeteren.

Mijn vertrekpunt was een basiskennis in het schrijven van rapporten en het geven van adviezen. Inmiddels ben ik gegroeid in het schrijven van gerichte, goed onderbouwde adviezen en in het toepassen van feedback om mijn rapporten verder te verbeteren.

Iets waar ik me nog zeker in wil ontwikkelen, is het schrijven van rapporten die voor mensen die geen technische achtergrond hebben zodat het direct te begrijpen is waar het over gaat. Daarnaast kan ik nog groeien in het balanceren van details en overzicht, zodat mijn rapporten effectief genoeg blijven zondere onnodige details.

Wat ik zeker wil meenemen als ik ga werken aan mijn afstudeerstage is dat een adviesrapport compact, maar volledig moet zijn. Ook wil ik mijn eigen conclusies en adviezen beter scheiden, zodat de lezer duidelijk weet wat zijn of haar beslissing is.

Iets wat ik de volgende keer anders ga doen is al van tevoren voordat ik überhaupt begin met schrijven, eerst de structuur helemaal uitstippel, zodat de verschillende onderdelen van het rapport duidelijk van elkaar onderscheiden zijn. Dit zorgt er dan hopelijk voor dat ik efficiënter kan werken aan het opstellen van het rapport.

Ik heb mijn leerdoel adviseren behaald. Ik kan een adviesrapport schrijven dat goed onderbouwd is met analyses, zoals kostenoverwegingen en prestatie-analyses. Daarnaast heb ik feedback effectief verwerkt om mijn samenvatting te verbeteren en mijn formuleringen aan te passen, zodat de lezer ruimte heeft om zijn of haar eigen keuzes te maken. Ik heb ook geleerd mijn structuur te verbeteren en complexe informatie toegankelijker te maken.

# Hoofdstuk 6

## Ontwerpen

### 6.1 Inleiding

In dit hoofdstuk ga ik bewijzen dat ik de competentie Ontwerpen tijdens mijn stage heb weten te behalen. Dit ga ik bewijzen door middel van Software ontwerpen(ERD, UML, ...).

Ter herinnering, in mijn stagevoorstel had ik het volgende leerdoel opgesteld: „Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een volledig en compleet verbeteringsplan op te stellen voor een nieuw of bestaand softwaresysteem, waarbij ik zowel nieuwe als bestaande componenten meeneem in het ontwerp. Hierbij houd ik rekening met de eisen van de belanghebbers en met de veiligheids- en schaalbaarheidsnormen die gelden binnen het stagebedrijf. Ik ben tevreden wanneer het verbeteringsplan goedgekeurd wordt en als basis kan dienen voor verdere ontwikkelingen binnen het project.”

### 6.2 Bewijs

Een klant van Harborn digital heeft in de Azure Cloud omgeving een Azure REST API dat bepaalde informatie ophaalt. Dit is op zich niet verkeerd, maar de backend operaties zijn stored procedures in de database waar het is opgeslagen. Stored procedures worden vaak in de industrie gebruikt en zijn eigenlijk functies die in een database zijn opgeslagen om bepaalde data op te halen. Alleen waren de functies niet overzichtelijk en staat er ook gevoelige data in, wat niet de bedoeling is. De database was ook niet goed ontworpen en maakt niet gebruik van relaties tussen tabellen zoals one-to-many of many-to-many relaties, maar van RelationID's die als extra veld in een database row zitten om de relaties tussen de verschillende tabellen te hebben. Dit is best wel inefficiënt en onoverzichtelijk, dus heb ik een ERD opgesteld waarin de relaties duidelijker zijn, om dan makkelijker data eruit te halen zie figuur XIV. Bij het opstellen van dit diagram heb ik ook zo veel mogelijk rekening geprobeerd te houden met de huidige structuur van de database, zodat de overstap naar de nieuwe database zo soepel mogelijk verloopt. Ook heb ik een Class Diagram gemaakt om de nieuwe functionaliteiten van het nieuwe ontwerp van de REST API wat duidelijker te maken zie figuur XV. Hierdoor kunnen developers sneller en makkelijker begrijpen wat de uiteindelijke structuur van het programma is. Daarnaast heb ik ook een Sequence Diagram gemaakt om duidelijk uit te leggen wat de flow van een API request is en hoe requests worden geverifieerd zie figuur XIII. Dit heb ik uiteindelijk allemaal samengesteld in een document zodat het voor developers en voor de klant duidelijk is wat er moet gebeuren en hoe. Omdat het document heel erg lang is, heb ik dus ook alleen de drie belangrijkste afbeeldingen/ontwerpen in de bijlagen gestopt.

### 6.3 Feedback

Uiteindelijk heb ik het hele rapport gedeeld met de developers van de klant tijdens een vergadering om te kijken of ik alles goed had begrepen. Dit sluit ook mooi aan bij de competentie adviseren, waarbij ik een verbeteringsplan heb opgesteld dat verschillende punten aanwijst om te verbeteren, zodat wij dat voor de klant kunnen implemeteren. De klant had eerder al aangegeven dat de inzichten van de diagrammen erg fijn en nuttig zijn. Hoewel het stagebedrijf uiteindelijk verantwoordelijk is voor het schrijven en opleveren van de producten, helpen deze diagrammen hen

om beter te begrijpen hoe de systemen zijn opgebouwd. Dit maakt het voor hen eenvoudiger om met onze verbeterplannen en implementaties mee te denken. Ze waarderen vooral de helderheid en professionaliteit van de opgeleverde documentatie en diagrammen.

Ook heb ik wat feedback gevraagd aan mijn stagebegeleider die aanwezig was tijdens de presentatie en ook mijn verbeteringsplan heeft gelezen en goedgekeurd.

#### **De vragen die ik heb gesteld aan mijn stagebegeleider**

1. Zijn de door mij gemaakte ontwerpen (ERD, Class Diagram, Sequence Diagram) technisch correct?
2. Hoe denkt je dat mijn ontwerpwijzigingen, zoals het verbeteren van de database-relaties en de herstructurering van de API, de prestaties, veiligheid en schaalbaarheid van de oplossing voor de klant zullen beïnvloeden?
3. Aangezien ik geprobeerd heb rekening te houden met de huidige structuur van de database, vind je dat mijn aanpak voldoende is om een soepele overgang naar het nieuwe ontwerp te garanderen? Wat kan ik doen om de migratie nog efficiënter te maken?
4. Helpt mijn documentatie, met de opgenomen diagrammen en uitleg, developers en klanten voldoende om het ontwerp en de implementatie te begrijpen?
5. Zijn er nog andere opmerkingen die je kwijt wil?

#### **De antwoorden/feedback van mijn stagebegeleider**

1. De diagrammen zijn duidelijk en correct.
2. Met dit ontwerp wordt deze service een stuk duidelijker en robuster wat de veiligheid en stabiliteit ten goed komt. Daarnaast is het hierdoor beter te onderhouden en uit te breiden door andere ontwikkelaars ten opzichte van de huidige oplossing op basis van stored procedures.
3. Het databaseontwerp wijkt niet heel veel af van de huidige situatie, maar heeft een paar wijzigingen die de schaalbaarheid ten goede komen. Indien er bij de overgang ook een migratie van de huidige database vereist is, zou een migratie script kunnen helpen. Daarbij moet dan rekening gehouden worden met de primary keys in het ontwerp waar die kolommen dat nu nog niet zijn.
4. Naar mijn mening is dit ontwerp heel duidelijk en zal het de ontwikkeling van de service enorm helpen.
5. Met de presentatie aan de klant, die tevreden was met het ontwerp, heb je in mijn ogen voldaan aan de doelstelling van dit leerdoel.

## **6.4 Reflectie**

De feedback heeft mij geleerd dat de technische ontwerpen die ik heb gemaakt correct en waardevol zijn. Dat duidelijke en professionele documentatie waardevol is voor developers en klanten is mij ook duidelijk geworden. Vooral hoe dit dan bijdraagt aan het begrip en de samenwerking binnen projecten. Daarnaast zijn mijn ontwerpen niet alleen op technisch vlak goed, maar ook op schaalbaarheid, veiligheid en onderhoudbaarheid. Ik heb geleerd hoe ik complexe systemen kan analyseren en een samenhangend ontwerp kan maken dat zowel nieuwe als bestaande componenten integreert en verbeterd. Ook heb ik geleerd hoe ik eisen het beste kan balanceren. Dus denk aan het toepassen van functionele en niet functionele eisen, zoals schaalbaarheid en veiligheid in een

ontwerp. En dat effectieve documentatie cruciaal is voor de kennisdeling en samenwerking met klanten en collegas. Mijn vertrekpunt was een basiskennis van softwareontwerp en beperkte ervaring met verbeteringsplan in een professionele context. Ik heb door middel van het maken van de ontwerpen mijn vaardigheden hierin aanzienlijk verbeterd. Ook begrijp ik nu beter hoe mijn ontwerpbeslissingen impact hebben op de veiligheid, prestaties en schaalbaarheid. Een ontwikkelpunt is leren hoe ik een migratiescript moet schrijven zodat de migratie van het ene database design naar de andere zo soepel mogelijk verloopt. Ik neem de volgende dingen mee in de toekomst: Het vroeg en regelmatig betrekken van belanghebbenden tijdens ontwerptrajecten. Iteratief ontwerpen, dus feedback vragen en itereren over meerdere ontwerpen om zo tot een volledig en correct eind resultaat te komen. Ook schaalbaarheid- en migratienormen ga ik zo veel mogelijk in mijn achterhoofd proberen te houden tijdens het ontwerpen van nieuwe systemen. En het schrijven van duidelijke documentatie ga ik ook zeker in de toekomst blijven doen, want dit zorgt gewoon voor een veel makkelijkere en duidelijkere kennisoverdracht met klanten en collegas. Het enige wat ik de in mijn ontwerp proces in de toekomst anders wil doen/verbeteren is eerder rekening houden met migratieprocessen, voor mij betekent dit een migratiescript ook schrijven en toevoegen aan de documentatie.

Ik heb mijn leerdoel ontwerpen behaald. Ik kan nu een volledig en compleet ontwerp maken voor software-integratie waarbij ik zowel nieuwe als bestaande componenten combineer. Ik houd daarbij rekening met eisen van belanghebbenden, schaalbaarheid, veiligheid en onderhoudbaarheid. En mijn ontwerpen gaan ook in het eerste kwartaal van 2025 gebruikt worden om de verbeteringen uit te breiden en te implementeren.

# Hoofdstuk 7

## Realiseren

### 7.1 Inleiding

In dit hoofdstuk ga ik bewijzen dat ik de competentie Realiseren tijdens mijn stage heb weten te behalen. De twee verschillende vormen van bewijsstukken die ik ga gebruiken zijn Code en Unit Tests.

Ter herinnering in mijn stagevoorstel had ik dit leerdoel opgesteld: „Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een zelfgebouwde module succesvol te integreren met een bestaand systeem door de bestaande codebase te analyseren en aanpassingen te maken met minimale impact. Ik ben tevreden wanneer mijn code de acceptatieomgeving doorstaat en in gebruik wordt genomen door de klant.”

Voor het realiseren gedeelte ga ik het hebben over een mooi project waar ik in de eerste paar weken van mijn stage aan heb mogen zitten. Het project houdt in dat een klant van Harborn digital de Cloud omgeving van AWS naar Azure volledig wilt verplaatsen. En voor 1 tool products-sync waarmee de verschillende producten die op hun website worden getoond ook regelmatig gekeken wordt of de informatie geüpdated moet worden. Eerst werd alles met AWS step functions die dan verschillende AWS Lambda functions aanroepen gedaan, maar het werd dus uiteindelijk te duur en moet dus naar Azure verplaatst worden. En met een Cron job zou het dan op een bepaald moment steeds kunnen laten runnen.

Voor de bewijslast ga ik gebruik maken van Unittests en van source code. De Unittests zijn volledig in Python geschreven en de source code is Terraform code, Python code en yaml code.

En ik ga nu met verschillende code fragmenten van de Code Base waar ik trots op ben laten zien hoe dat uiteindelijk is gegaan.

### 7.2 Bewijs

Unittests zorgen ervoor dat je goed kan checken of de code die je hebt geschreven wel betrouwbaar genoeg is. Daarnaast helpen unittests met het checken of de geschreven wijzigingen niet meteen alles kapot maakt en dus waarborg je de minimale impact van je code. Uiteindelijk zijn unittests ook een deel van de acceptatietests die worden uitgevoerd om te kijken of de geschreven code wel naar productie mag. Voor het schrijven van unittests heb ik gekeken naar hoe de unittests zijn gestructureerd en waar ze van gebruik maken. In de bijlage IV kan je zien dat ik tests heb geschreven voor het testen van verschillende API endpoints. Om bijvoorbeeld het aanroepen van bepaalde requests te mocken/na te doen heb ik gebruik gemaakt van de patch module in unittest.mock.

Docker is een technologie die gebruik maakt van containers om applicaties te isoleren in een omgeving die altijd hetzelfde is. Bij het ontwikkelen en integreren van nieuwe functionaliteit, zoals het toevoegen van een nieuwe module, kan Docker helen bij het testen en accepteren van deze wijzigingen met minimale impact op de rest van het systeem. De manier hoe de Docker containers zijn opgezet in dit project waar ik aan mee heb mogen werken zorgt ervoor dat als er een pull request, een aanvraag om code van een andere branch naar de main te mergen, wordt aangemaakt de code ook meteen automatisch wordt getest. Daarnaast is het deployen van de code naar de acceptatie-omgeving ook gestandaardiseerd, omdat de code dus altijd in dezelfde omgeving werkt. Om deze code automatisch te testen bij een pull request kan je gebruik maken van Github Actions

om dan een Docker container op te zetten en dit aan te roepen. In bijlage V kan je zien hoe zo'n yaml bestand eruit ziet.

Een interface is een blauwdruk van hoe een bepaalde class eruit moet zien, dus welke functionaliteiten moeten worden geïmplementeerd zonder details van die implementatie te specificeren. Dit is dus een krachtig hulpmiddel voor het implementeren van nieuwe code met minimale impact, omdat het zorgt voor een loskoppeling tussen verschillende delen van een systeem. De implementatie van een class die een interface implementeert kan veranderen zonder dat andere delen van de applicatie, die afhankelijk zijn van die interface, aangepast hoeven worden. Een van de grotere taken binnen het realiseren gedeelte was het refactoren van de AWS code naar Blob storage code, om dit makkelijker te doen heb ik een abstracte class gemaakt dat als interface dient. Hierbij heb ik vooral gekeken en geanalyseerd met hoe de class werkt, waardoor de interface meteen op de AWS class gezet kan worden zonder dat de functionaliteit ervan veranderd. Zodat classes die hiervan afstammen de juiste en dezelfde method signatures hebben, dit kan je zien in bijlage VI. Deze interface is dan uiteindelijk ook geïmplementeerd in de blob\_client class en dat kan je zien in bijlage VII.

Terraform is een hele handige tool om cloud omgevingen te managen, je kan er namelijk versiebeheer meedoen en dus verschillende versies van je infrastructuur opslaan zodat als er iets fout gaat met een nieuwe versie je toch nog terug kan stappen. Daarnaast is het een tool en taal voor het beschrijven van verschillende cloud omgevingen, waardoor het veranderen van cloud provider makkelijker kan gaan. Ook kan je met Terraform verschillende modules schrijven die je dan kunt herbruiken en opschalen zonder code opnieuw op te schrijven. Het was wel lastig in het begin om al in een bestaande codebase iets nieuws toe te voegen, eerst heb ik veel code geanalyseerd en bekeken met hoe het nou precies werkt om dan vervolgens mijn eigen module te schrijven en zonder veel impact wijzigingen te maken aan de cloud omgeving. In de bijlage VIII kan je een voorbeeld zien van Terraform code die ik heb geschreven voor het makkelijk maken van een container\_app\_job.

En om te bewijzen dat het is geaccepteerd door de klant en het afgeleverd is heb ik in de bijlagen een screenshot gezet van de meeting die in de Google agenda staat.IX

## 7.3 Feedback

Om wat feedback te krijgen over dit leerdoel heb ik een kleine Google Forms opgesteld met verschillende vragen.

Hieruit heb ik van twee individuen een antwoord gekregen, namelijk mijn stagebegeleider Gijs Straathof en een collega die expert is op het gebied van cloud, Guus Ekkelenkamp.

Beiden zijn van mening dat ik mijn opgestelde leerdoel volledig heb behaald.

### **Welke uitdagingen heb je gezien in mijn aanpak of uitvoering?**

- **Gijs:** Jouw uitdagingen bestonden uit onbekendheid met de way-of-working, frameworks, code standards etc. zoals we dat hier verwachten. Daarnaast was er een uitdaging voor het inzetten van Azure cloud diensten waar binnen het bedrijf beperkte ervaring mee is.
- **Guus:** Het gebruik van technologieën die voor Harborn zelf ook nog jong zijn. Hierdoor moet je soms echt zelf dingen uitzoeken. Maar dat heb je volgens mij heel goed opgepakt.

### **Wat vond je goede punten van mijn aanpak bij het behalen van dit leerdoel?**

- **Gijs:** Je bent direct aan de slag gegaan, door de scrum way of working kon dat ook met de vooraf gedefinieerde user stories. Bij uitdagingen wist je de juiste mensen te vinden voor hulp.

- **Guus:** Van start tot finish voor de klant echt een professionele omgeving opgeleverd. Door de implementatie van verschillende development principes is de change van cloud provider minimaal voor de code. Ook is de code klaar om evt nog een keer een grote switch te krijgen en zal die impact ook kleiner zijn. Het neerzetten en deployen van een omgeving voor de klant heb je ook bijna volledig zelfstandig opgepakt. Hoewel ik er niet bij zat, heb je het ook goed live aan de klant gepresenteerd en was die ook tevreden. Denk dat je het niet veel beter kan realiseren dan dat.

#### **Ben je tevreden met de opgeleverde code, waarom wel of niet?**

- **Gijs:** Alle code is gereviewd, ik ben zeer tevreden over de opgeleverde code, dit voldoet aan al onze standaarden.
- **Guus:** Van wat ik heb gezien (vooral terraform) was het prima. Terraform is een beetje een rare taal als je dat nog nooit hebt gedaan of weinig met cloud doet, dus in het begin was dat volgens mij moeilijk. Maar wat ik mooi vond is dat jij feedback echt snapte en niet gewoon doet wat ik zeg. Daardoor komen er ook extra verbeteringen uit naast hetgeen wat je verteld wordt.

#### **Welke aanbevelingen heb je voor mij om mijn aanpak in de toekomst te verbeteren?**

- **Gijs:** Je hebt de neiging snel naar het eindresultaat toe te werken zonder de hele context goed te begrijpen. Je kunt iets meer rust inbouwen, zorgen dat je alle aspecten snapt (goed luistert) voor je aan de slag gaat.
- **Guus:** Geen idee

#### **Zijn er nog andere opmerkingen die je kwijt wil?**

- **Gijs:** Je hebt laten zien creatief te zijn in het vinden van oplossingen.
- **Guus:** Denk dat je het realiseren van je opdracht heel goed opgepakt hebt, complimenten.

## **7.4 Reflectie**

Iets wat ik in de toekomst zeker wil gaan verbeteren en ga meenemen is dat ik het wat rustiger aan kan doen met het beginnen van het werken naar een eindresultaat. Dit kan er dan voor zorgen dat ik alle aspecten van een opdracht beter op een rijtje kan zetten en uiteindelijk ook beter snap wat er moet gebeuren. Dit ging vooral in de eerste fase van het realiseren namelijk wat minder en was een duidelijke opmerking/verbeterpunt die ik heb gekregen van mijn stagebegeleider. Dit heb ik dan ook meegenomen bij de resterende taken die ik heb moeten doen met betrekking tot het voltooiën van dit leerdoel. Het schrijven van de Terraform code heb ik dan ook veel rustiger aangepakt, ik heb wat langer gekeken naar hoe de code is gestructureerd en hoe het werkt. Om daarna de code te schrijven.

Op basis van de feedback van mijn begeleider en collega, ben ik erg tevreden met de snelle start en het efficiënt benutten van de scrum-werkwijze. Ook heeft het zoeken van de juiste mensen bij uitdagingen geholpen om snel stappen te kunnen zetten en de klant een goede oplossing te bieden. Daarnaast heb ik door development principes toe te passen er goed voor weten te zorgen dat ik toekomstbestendige code heb opgeleverd, dat minimale impact heeft op de codebase. Die voldoet aan de standaarden binnen het stagebedrijf, wat ervoor zorgt dat de overstap naar andere cloudproviders veel minder impact op de codebase heeft. In de toekomst wil ik deze houding en werkwijze ook blijven gebruiken, omdat het het hele proces van realiseren veel makkelijker en overzichtelijker heeft gehouden.

Daarnaast heb ik door de interne processen voor het schrijven van veilige code geleerd hoe je unit tests moet schrijven en waarom het nodig is. Bij de projecten in het eerste en tweede jaar van de opleiding heb ik hier eigenlijk helemaal geen rekening gehouden en was het iets wat je op het einde er snel in gooide.

De hierboven genoemde punten samen met de acceptatie van de klant en hun tevredenheid tijdens de live presentatie geven mij vertrouwen dat ik mijn leerdoel bereikt heb.



# Hoofdstuk 8

## Professional Skills

### 8.1 Inleiding

In dit hoofdstuk ga ik door middel van feedbackformulieren (misschien chatgesprekken) bewijzen dat ik mijn leerdoel voor professional skills heb behaald. Ter herhaling is dit mijn opgestelde leerdoel: „Ik wil na afloop van mijn stageperiode van 20 weken instaat zijn om effectief en zelfstandig te werken op het niveau van een junior medewerker binnen het stagebedrijf, door actief hulp en feedback te vragen aan collega's en uit mijn comfortzone te stappen. Ik ben tevreden wanneer ik zonder moeite weet naar wie ik toe kan stappen voor welke soort vragen.”

### 8.2 Bewijs

Het bewijs dat ik ga leveren voor dit is in de vorm van een feedback formulier en 1 op 1 gesprekken met mijn collega's en begeleider.

Ook heb ik wekelijks een korte zelfreflectie geschreven om beter zicht te krijgen op mijn eigen inzicht en om mijn ontwikkeling tijdens de stage te kunnen volgen. Deze zelfreflecties had ik alleen in de eerste maand gedaan. Zie bijlage X Hieruit is gekomen dat ik in de eerste paar weken nog eerst iedereen moest leren kennen en wie welke expertise heeft op bepaalde gebieden. Uiteindelijk toen ik begon aan wat lastigere taken die met de cloud te maken hebben ben ik vaker met gerichte vragen naar dezelfde persoon gestapt die ook veel met dat project heeft gewerkt. Ook heb ik natuurlijk veel vragen aan mijn stagebegeleider gesteld. Ik was vooral de eerste paar keer een beetje nerveus om naar mijn collega's buiten mijn team te stappen, maar doordat de werkplekken allemaal in 1 grote ruimte zijn was dit wel wat makkelijker. En tegenwoordig heb ik helemaal geen moeite met het stellen van vragen aan collega's.

### 8.3 Feedback

Guus, een collega, heeft aangegeven dat ik met de juiste vragen naar hem toe ben gestapt wanneer ik hulp nodig had. Hij vond dat mijn vragen vaak heel gericht waren, wat volgens hem aantoont dat ik eerst zelf voldoende heb geprobeerd. Hierdoor konden we samen over een vraag sparren en ging het vaak om meer dan alleen de makkelijke oplossing. Wel gaf hij aan dat het stellen van vragen soms sneller had gekund, maar dat dit ook aan zijn drukke schema kan liggen.

Gijs, mijn begeleider, gaf aan dat ik vragen op de juiste momenten stel. Ik wacht er niet te lang mee, maar stel ze ook niet te snel. Volgens hem voldoe ik op dit moment zeker aan mijn doelstelling. Mijn begeleider heeft niet het idee dat ik te lang wacht met vragen stellen en merkt dat ik de juiste collega's weet te benaderen voor specifieke hulp. Over het algemeen vindt hij dat ik vragen effectief stel en opensta voor feedback. Wel ziet hij dat ik soms te snel denk iets te begrijpen, waardoor ik minder doorvraag naar alternatieve oplossingen en randvoorwaarden. Toch waardeert hij dat ik weet wie ik moet benaderen voor specifieke feedback en, als ik het niet weet, hem om advies vraag.

## 8.4 Reflectie

Ik heb geleerd dat dieper doorvragen naar oplossingen en context belangrijk is en dat dat dan de kwaliteit van mijn werk verbeterd. Ook is het goed om gerichte vragen te stellen en aan te geven wanneer ik moeite had met het komen tot oplossingen, hierdoor kan ik sneller verder met het schrijven van een oplossing. En het niet al te lang wachten met het stellen van vragen zorgt ook voor een betere samenwerking.

Naar aanleiding van de feedback ben ik dus ook sneller vragen gaan stellen buiten mijn team als ik bijvoorbeeld tegen een probleem aanloop die Gijs, mijn stagebegeleider, niet kan oplossen. Daarnaast ben ik meer gaan reflecteren op of ik een probleem volledig snap, voordat ik er meteen in duik en aan ga werken. Ik probeer nu namelijk een klein tekeningetje te maken van de flow en vraag dan of dit correct is voordat ik het ga implementeren in code.

Mijn vertrekpunt was dat ik in het begin, hierbij moet je denken aan de eerste 3/4 weken, nerveus was om vragen te stellen en ook niet wist naar wie ik toe moest stappen. Nu weet ik dat wel en ben ik er veel zelfverzekerder in. Daarnaast kunnen collega's ook naar mij toe stappen voor vragen omdat ik steeds meer kennis heb over bepaalde projecten waar ik aan heb gezeten. Om dit in de toekomst te verbeteren als ik in een andere of nieuwe werkomgeving terechtkom, ben ik van plan met de teamlead samen te zitten met een koffie. Om dan vragen te stellen wie waar verantwoordelijk voor is binnen het team of bedrijf, zodat ik weet naar wie ik kan toe stappen voor vragen, hulp en feedback.

Ik wil me nog steeds blijven doorontwikkelen op het niet te snel tevreden te zijn met mijn begrip over problemen of projecten. Ik wil eerst kritisch kijken of ik het volledig snap, zo niet dan stap ik naar de relevante collega/medewerker om er vragen over te stellen.

Het belang van openheid, gerichte vragen en reflectie op kennis en begrip zijn voor mij ook belangrijke punten om mee te nemen in de toekomst, zodat als ik aan mijn afstudeerstage begin een goed begrip kan krijgen over de opdracht die ik voorgeschoteld krijg. Ook wil ik door blijven gaan met het actief vragen van feedback, zodat ik mijzelf kan blijven doorontwikkelen.

De volgende keer ga ik sneller feedback vragen bij twijfel of als ik vastloop, hier had ik op sommige momenten namelijk net iets te lang mee gehad.

Op basis van de bovenstaande feedback en punten denk ik dat ik mijn zelfopgestelde leerdoel voor de competentie Professional Skills heb weten te behalen.

## Hoofdstuk 9

# Overkoepelende Reflectie

De afgelopen twintig weken bij Harborn Digital waren een waardevolle leerervaring waarin ik mijn professionele vaardigheden heb verdiept en versterkt. Dit portfolio documenteert mijn groei binnen verschillende competenties en toont hoe deze samen hebben bijgedragen aan mijn ontwikkeling als een toekomstig junior softwareontwikkelaar. In deze reflectie breng ik de rode draad naar voren tussen de inzichten die ik heb opgedaan en wat dit betekent voor mijn afstudeerstage.

Zoals je misschien is opgevallen, heb ik de zes competenties Manage & Control, Analyseren, Adviseren, Ontwerpen, Realiseren, en Professional Skills op afzonderlijke projecten toegepast tijdens mijn stage. Voor mijn afstuderen zal dat niet het geval zijn, daar zal ik een opdracht krijgen waarin deze competenties samenkomen binnen één afstudeer opdracht. Dit betekent dat ik eerst moet analyseren, daarna een ontwerp opstel en een advies formuleer, voordat ik aan de slag kan met het realiseren. Omdat deze onderdelen afhankelijk zijn van elkaar, is het essentieel dat mijn onderzoek goed aansluit op mijn ontwerp en dat beide een sterke basis vormen voor mijn advies. Gebrekkige afstemming kan leiden tot inconsistenties, en om dat te voorkomen, ben ik van plan regelmatig feedback te vragen en terug te koppelen met de opdrachtgever. Zo kan ik elk onderdeel van het project goed laten bijdragen aan het overkoepelende doel.

Een uitdaging die ik tijdens mijn meeloopstage niet volledig heb kunnen oefenen, is het combineren van deze competenties in een samenhangend project. Voor mijn afstuderen zal ik een project aanpakken waarin analyseren, adviseren, ontwerpen, realiseren, professional skills en manage & control op elkaar voortbouwen. Dit vereist een sterke afstemming en consistentie tussen de verschillende fasen van het project.

Een van de belangrijkste lessen die ik heb geleerd, is hoe belangrijk een iteratieve en gestructureerde aanpak is. Door regelmatig feedback te vragen en mijn werkproces continu te evalueren/reflecteren, kon ik beter inspelen op de verschillende uitdagingen. Zo ontdekte ik tijdens de analyse van Giraf het belang van diepgaande context voordat ik conclusies trek.

Daarnaast heeft mijn focus op documentatie mij geholpen om complexe systemen helder over te brengen naar de belanghebbenden. Het structureren van mijn ontwerpen in duidelijke diagrammen heeft bijgedragen aan een betere samenwerking met de klant en collegas. Deze ontwerpen worden tijdens Q1 van 2025 gebruikt voor het verbeteren van de geanalyseerde software componenten.

Om succesvol te zijn in mijn afstudeerstage, zal ik regelmatig feedback vragen aan belanghebbenden en mijn bevindingen met hen afstemmen om inconsistenties te voorkomen. Daarnaast wil ik meer tijd investeren in het analyseren van de probleemstelling, zodat het ontwerp hier goed op aansluit. Ik zal structuur aanbrengen in mijn werkproces door duidelijke mijlpalen te stellen en mijn documentatie en presentaties zo toegankelijk mogelijk te maken, ook voor niet-technische belanghebbenden.

Voor mijn afstuderen ben ik gemotiveerd om de lessen en inzichten uit mijn stage toe te passen. Ik zal mijn methodische aanpak verder verfijnen en bewust tijd besteden aan het balanceren van de verschillende fasen van een project. Door consistent feedback te vragen en een iteratieve werkwijze te hanteren, kan ik elk onderdeel van het afstudeerproject laten bijdragen aan een sterk en samenhangend eindresultaat.

Mijn stage heeft mij niet alleen technische vaardigheden bijgebracht, maar ook inzicht gegeven in hoe ik als professional wil groeien. Door consistent te reflecteren en feedback te gebruiken, heb ik een stevige basis gelegd voor mijn verdere ontwikkeling als developer op technisch en professioneel vlak. Voor mijn afstudeerstage wil ik de lessen uit deze stage toe te passen en de competenties samen te brengen in een succesvol en samenhangend eindproject.

# Hoofdstuk 10

## Advies bedrijfsbegeleider

Advies Bedrijfsbegeleider

Naam student	Douwe Zumker	
Studentnummer	1059179	
<b>Advies bedrijfsbegeleider</b> De student wordt beoordeeld op de zelf opgestelde leerdoelen passend bij de competenties binnen Informatica, die ze in hun portfolio dienen weer te geven. Als u kijkt naar zijn/ haar werkzaamheden en het portfolio, hoe zou u dan de onderstaande leerdoelen beoordelen? U kunt per onderdeel een onvoldoende/ matig/ voldoende/ goed invullen met hierbij uw argumentatie. Uw input wordt gezien als een belangrijk advies.		
Competentie	O/M/V/G	Onderbouwing
<b>Manage &amp; control</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om betere inschattingen te maken over de voortgang van een project en de tijd die ik nodig heb voor verschillende taken. Ik ga dit bereiken door actief deel te nemen aan refinements en inschatting sessies, zodat ik de complexiteit van features beter kan inschatten. Ik ben tevreden wanneer ik verschillende features tijdens refinements beter kan inschatten, waarbij ik dan meer in lijn wil zijn met de developers die bij Harborn digital werken. En ik wil ook de meeste taken binnen de aangegeven tijd afronden.	G	Dit gaat over het inschatten van de technische benodigdheden en de benodigde inzet om het werk tot een goed einde te brengen. Technisch toont Douwe veel begrip van complexiteit, en is hij gegroeid in het bijdragen in een discussie tijdens refinements. Tijdens het beoordelen van de benodigde inzet zit Douwe zijn inschatting doorgaans binnen de bandbreedte van andere collega's. Hij heeft duidelijk de mogelijkheden opgezocht om dit leerdoel te leren en demonstreren.
<b>Analyseren</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn de huidige implementatie van een stabiele applicatie te analyseren om te beoordelen of deze nog voldoet aan de veranderende of vernieuwde standaarden. Ik ga dit bereiken door de applicatie	G	Hiervoor heeft Douwe een analyse uitgevoerd van een bestaand, door Harborn ontwikkeld, SaaS platform waarvan meerdere klanten gebruik maken. Het platform maakt gebruik van kunstmatige intelligentie en deze AI module is waar de analyse zich op gericht heeft. Douwe toonde snel begrip voor de samenhang en technische details van de module. Hij heeft de prestaties van een alternatief AI model vergeleken met de huidige implementatie. Vervolgens heeft hij, rekening houdend met kosten, een duidelijk advies


grondig te evalueren aan de hand van de geldende technische en veiligheidsnormen, en de resultaten te verwerken in een analyse. Ik ben tevreden wanneer mijn analyse duidelijke verbeterpunten oplevert en wordt gebruikt in het adviesrapport voor de verbetering van de applicatie.		gegeven waar we mee aan de slag kunnen tijdens doorontwikkeling. Wat mij betreft voldeed Douwe met deze analyse duidelijk aan de verwachtingen rondom dit leerdoel.
<b>Adviseren</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een goed onderbouwd advies te kunnen geven over de selectie van software onderdelen van een bestaande applicatie. Ik ga dit bereiken door middel van kostenoverweging en een performance-analyse, en door de resultaten in een adviesrapport te verwerken. Ik ben tevreden wanneer ik een compleet rapport heb geschreven dat gebruikt kan worden om de onderzochte software onderdelen te verbeteren.	V	Douwe heeft een adviesrapport geschreven voor verbeteringen aan de AI module van het geanalyseerde SaaS platform. Dit advies was het logisch vervolg op de analyse waarbij er ook gekeken is naar relevante wetgeving rondom het gebruik van AI. Het advies is vrij generiek en luidt voornamelijk dat er meer onderzoek nodig is voordat verbeteringen kunnen worden doorgevoerd.

<b>Ontwerpen</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een volledig en compleet integratieplan op te stellen voor een nieuw of bestaand softwaresysteem, waarbij ik zowel nieuwe als bestaande componenten meeneem in het ontwerp. Hierbij houd ik rekening met de eisen van de belanghebbenden, die ik achterhaal via interviews, en met de veiligheids- en schaalbaarheid normen die gelden binnen het	G	Harborn is gevraagd een applicatie van een klant te vervangen, hiervoor heeft Douwe een ontwerp opgesteld. Hij heeft, samen met de klant, de functionele vereisten besproken en verbeterpunten geïdentificeerd. Vervolgens heeft hij een uitgebreid ontwerp voorstel opgesteld welke door zowel Harborn als de klant als zeer waardevol zijn beoordeeld. Het ontwerp zal binnenkort uitgevoerd worden.

stagebedrijf. Ik ben tevreden wanneer het onderzoeksvoorstel goedgekeurd wordt en als basis kan dienen voor verdere ontwikkelingen binnen het project.		
<b>Realiseren</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om een zelfgebouwde module succesvol te integreren met een bestaand systeem door de bestaande codebase te analyseren en aanpassingen te maken met minimale impact. Ik ben tevreden wanneer mijn code de acceptatieomgeving doorstaat en in gebruik wordt genomen door de klant.	G	Douwe heeft vanaf de start aan meerdere projecten meegewerkt en is in staat om relatief snel een succesvolle bijdrage te leveren. Er is ook een project dat grotendeels door Douwe zelfstandig is opgepakt. Zijn werk is met enthousiasme ontvangen door de opdrachtgever en uitgerold naar productie.
<b>Professional skills</b>		
Ik wil na afloop van mijn stageperiode van 20 weken in staat zijn om effectief en zelfstandig te werken op het niveau van een junior medewerker binnen het stagebedrijf, door actief hulp en feedback te vragen aan collega's en uit mijn comfortzone te stappen. Ik ben tevreden wanneer ik zonder moeite weet naar wie ik toe kan stappen voor welke soort vragen.	G	Binnen Harborn zijn teams end-to-end verantwoordelijk voor de projecten die binnen het team uitgevoerd worden. Sommige specialisaties zijn echter niet in ieder team vertegenwoordigd. Douwe heeft aangetoond zonder schroom zijn hulpvragen bij de juiste personen neer te leggen. Binnen het team functioneert hij als een junior medewerker.

<p>Heeft de student professioneel en integer gedrag vertoond? Waaruit blijkt dit? (denk hierbij aan werkhouding, collegialiteit, eigen initiatief, zelfstandigheid):</p> <p>Douwe functioneert in het team als een volwaardige collega. Hij zet zich proactief in voor de projecten waar hij aan werkt en heeft zijn werk succesvol gepresenteerd aan opdrachtgevers.</p>
<p>Heeft de student zich aan organisatorische regels en plannings gehouden? Waaruit blijkt dit?</p> <p>Zeker: Douwe werkt mee in het team als iedere andere collega en doet daardoor mee aan alle ceremonies zoals planning, retro, refinements en inschattingen.</p>

<p>Wat kunt u zeggen over de samenwerking van de student met de bedrijfsbegeleider en collega's? En wat kunt u zeggen over de communicatieve eigenschappen van de student?</p> <p>Douwe is zeer bekwaam in communicatie met collega's en klanten. Hij weet goed in te schatten wanneer het juiste moment is om een vraag te stellen of aan de bel te trekken als er iets niet goed gaat. Daarbij komt hij ook met mogelijke oplossingen.</p>
<p>Op welke wijze heeft de student blij gegeven van voldoende beheersing van kennis en vaardigheden om de opdracht/werkzaamheden uit te voeren?</p> <p>Zoals bij de leerdoelen aangegeven heeft Douwe vanaf het begin meegewerkt aan projecten. Dat hij op zijn eerste werkdag een constructieve bijdrage heeft geleverd aan het project dat aan hem toegewezen was, overtrof de verwachtingen.</p>
<p>Op welke wijze heeft de student aan persoonlijke kennisontwikkeling gewerkt? En heeft de student nieuwe technieken/nieuwe inzichten aangetoond die voor hem/haar nieuw waren en/of nieuw waren voor het bedrijf?</p> <p>Douwe heeft onze way-of-working en standaarden snel omarmd. Hij heeft Harborn ook geholpen met nieuwe kennis: Het project waar hij het voortouw in heeft genomen diende gehost te worden in een omgeving die voor Harborn relatief nieuw is. Met hulp van de juiste collega's en zijn onderzoeksvaardigheden heeft hij dat succesvol afgerond.</p>
<p>Heeft de student kennis doorgegeven in uw organisatie? En zo ja, hoe?</p> <p>De kennis die hij heeft opgedaan is gedeeld met collega's via code, demo's en discussies.</p>

<p><b>Naam bedrijfsbegeleider:</b> Gijs Straathof</p>
<p><b>Datum:</b> 8 / 1 / 2025</p>
<p><b>Handtekening</b></p> 

# Bibliografie

[1] *Harborn*, <https://www.harborn.com/nl>, Bezocht op: 15 januari 2025.



# Hoofdstuk 11

## Bijlagen

### I User Story

The screenshot shows a Jira user story card for the project 'SyncForce'. The story ID is 'HOL-DH-40'. The story text is: 'As a developer I want to research why Azure Web App crashes after the /jobs endpoint is called'. The card is assigned to 'Back-end' and 'Team-1'. It includes a description, a list of things tried to find the problem, and a theory about the cause of the crashes. The card also has a 'Comments' section with one comment from 'Douwe' dated Friday, October 11th, 16:08, containing a link to 'https://holmatro-products-sync.harborn.dev'. On the right side, there are several floating buttons: 'Due date', 'Size', 'Logged work', and 'Time estimation'.

**SyncForce**

HOL-DH-40

**As a developer I want to research why Azure Web App crashes after the /jobs endpoint is called**

Assign members: Back-end, Team-1, Add tag

Description

The Azure Web App which hosts the api that will be used in synchronizing crashes after the /jobs endpoint has been called. The job will still run in the back, but the fastapi website won't be able to be reached anymore.

Things I have tried to find the problem:

- Sending app service logs to blob storage
- Analyzing the log stream under the monitoring tab
- Analyzing the logs in the logs tab under Deployment Center
- Setting up Sentry logging inside the codebase

One theory that I have is that when calling the /jobs endpoint there maybe could be a large spike in cpu/ram usage which crashes the container. Because I can't see anything inside the logs this is the only theory I can come up with so far.

Add task, Attach, Comments

Write a comment... (type '@' to mention people)

**Douwe** Friday, October 11th, 16:08  
<https://holmatro-products-sync.harborn.dev>

Due date, Size, Logged work, Time estimation

## II Planning

<input type="checkbox"/>	As Holmatro, I want an improved version of the MyHolmatroPortal database so that retrieving data is more efficient and user-friendly. HOL-DH-62 0/3 3 Back-end Team-1	Portal	3
<input type="checkbox"/>	As a registered user with API access, I want to authenticate with a JWT and subscription key to securely access MyHolmatroPortal API servi... HOL-DH-63 0/1 1 Back-end Team-1	Portal	2
<input type="checkbox"/>	Add the /products/salesinfo API endpoint HOL-DH-72 3 Back-end Team-1	Portal	1
<input type="checkbox"/>	Add the /products/{articulo}/availability API endpoint HOL-DH-71 3 Back-end Team-1	Portal	1
<input type="checkbox"/>	Add the /general/user API endpoint HOL-DH-70 3 Back-end Team-1	Portal	1
<input type="checkbox"/>	Add the /products/{articulo}/salesprices API endpoint HOL-DH-69 3 Back-end Team-1	Portal	1
<input type="checkbox"/>	As a developer I want a setup of a repository for the MyHolmatroAPI HOL-DH-66	Portal	2

### III OpenID implementatie

```
1 import urllib.parse
2 from typing import Union
3
4 import httpx
5 from fastapi import Depends, HTTPException, Request
6 from fastapi.security import HTTPAuthorizationCredentials, HTTPBearer
7 from starlette.status import (
8     HTTP_307_TEMPORARY_REDIRECT,
9     HTTP_400_BAD_REQUEST,
10    HTTP_403_FORBIDDEN,
11 )
12
13 from src.settings import get_default_logger
14 from src.shared.env import Env
15 from src.shared.models.user_claims import UserClaims
16
17 security = HTTPBearer()
18 Env = Env()
19 logger = get_default_logger()
20
21
22 def verify_token(credentials: HTTPAuthorizationCredentials = Depends(security)):
23     """
24     Verify the provided Bearer token against the environment configuration.
25     """
26     if credentials.credentials != Env.bearer_token:
27         logger.info("Invalid or expired token provided.")
28         raise HTTPException(
29             status_code=HTTP_403_FORBIDDEN, detail="Invalid or Expired Token"
30         )
31     return credentials.credentials
32
33
34 async def syncforce_sso_auth(request: Request) -> Union[UserClaims, None]:
35     """
36     Unified dependency for authentication and OpenID form-data validation.
37     Ensures valid authentication and handles OpenID validation against the provider.
38     """
39     try:
40         # Parse form data from the request
41         form_data = await request.form()
42         form_dict = dict(form_data)
43         logger.debug(f"Received form data: {form_dict}")
44
45         # Validate required OpenID fields
46         required_fields = [
47             "openid.mode",
48             "openid.assoc_handle",
49             "openid.sig",
50             "openid.signed",
51             "openid.claimed_id",
```

```

52         "openid.identity",
53     ]
54     missing_fields = [field for field in required_fields if field not in form_dict]
55     if missing_fields:
56         logger.info(f"Missing required OpenID fields: {missing_fields}")
57         redirect_to_sso(request)
58
59     # Update mode for validation
60     form_dict["openid.mode"] = "check_authentication"
61     encoded_data = urllib.parse.urlencode(form_dict)
62     oi_url = Env.oid_sync_force_url
63
64     # Make request to OpenID provider
65     async with httpx.AsyncClient() as client:
66         response = await client.post(
67             f"{oi_url}/Provider.ashx",
68             data=encoded_data,
69             headers={"Content-Type": "application/x-www-form-urlencoded"},
70             timeout=10, # Add timeout to prevent long hangs
71         )
72
73     # Check response status
74     if response.status_code != 200:
75         logger.error(
76             f"OpenID provider error: {response.status_code} - {response.text}"
77         )
78         raise HTTPException(
79             status_code=HTTP_400_BAD_REQUEST,
80             detail=f"OpenID provider error: {response.status_code}",
81         )
82
83     # Validate OpenID response
84     if not response.text.startswith("is_valid:true"):
85         logger.error("OpenID validation failed.")
86         raise HTTPException(
87             status_code=HTTP_400_BAD_REQUEST, detail="Validation failed."
88         )
89
90     logger.info("Authentication and validation succeeded.")
91     user_claims = UserClaims.from_openid(form_dict)
92     return user_claims
93
94     except HTTPException:
95         # Let existing HTTPExceptions propagate
96         raise
97     except Exception as e:
98         # Log and redirect for generic errors
99         logger.exception(f"Error during authentication: {e}")
100        redirect_to_sso(request)
101
102
103 def redirect_to_sso(request: Request) -> None:
104     """
105     Generate a redirection to the configured Single Sign-On (SSO) URL.

```

```

106     """
107     REALM = Env.sso_realms
108     return_url = f"{REALM}{request.url.path}"
109     if request.url.query:
110         return_url += f"?{request.url.query}"
111     return_url = urllib.parse.quote(return_url)
112
113     SSO_URL = Env.sso_url
114     redirect_url = SSO_URL.format(returns=return_url, realm=REALM)
115     logger.debug(f"Redirecting unauthenticated user to: {redirect_url}")
116
117     raise HTTPException(
118         status_code=HTTP_307_TEMPORARY_REDIRECT,
119         headers={"Location": redirect_url},
120     )

```

## IV Unittest

```

1  import json
2  from unittest.mock import patch
3
4  import pytest
5  from fastapi.testclient import TestClient
6
7
8  def test_start_step():
9      with patch("src.api.steps.start_step_functions") as mock_start_step_functions:
10         from src.app import app
11
12         client = TestClient(app)
13
14         try:
15             client.post(
16                 "/jobs",
17                 headers={"Authorization": "Bearer your-token"},
18             )
19             mock_start_step_functions.assert_called_once()
20         except Exception: # It throws an exception because of a request to syncforce
21             pass
22
23
24  def test_fetch_access_token():
25      with patch(
26          "src.services.products_import.fetch_sync_force_access_token.handle"
27      ) as mock_fetch_access_token:
28          from src.app import app
29
30          call_data = {}
31          fetch_token = "your_access_token"
32          client = TestClient(app)
33
34          mock_fetch_access_token.return_value = fetch_token

```

```

35
36     response = client.post(
37         "/jobs/fetch_access_token",
38         content=json.dumps(call_data),
39         headers={"Authorization": "Bearer your-token"},
40     )
41     assert response.status_code == 200
42     assert response.json() == fetch_token
43
44
45 def test_fetch_asset_ids():
46     with patch(
47         "src.services.products_import.fetch_sync_force_asset_ids.handle"
48     ) as mock_fetch_asset_ids:
49         from src.api.jobs.models import EventRequestBody
50         from src.app import app
51
52         call_data = {}
53         fetch_token = "your_access_token"
54         client = TestClient(app)
55
56         mock_fetch_asset_ids.return_value = []
57
58         request_body = EventRequestBody(token=fetch_token, event=call_data)
59         response = client.post(
60             "/jobs/fetch_asset_ids",
61             content=request_body.model_dump_json(),
62             headers={"Authorization": "Bearer your-token"},
63         )
64
65         assert response.status_code == 200
66         assert response.json() == []
67
68
69 def test_fetch_all_asset_details():
70     with patch(
71         "src.services.products_import.fetch_sync_force_asset_details.handle"
72     ) as mock_fetch_asset_details:
73         from src.app import app
74
75         call_data = {}
76         fetch_token = "your_access_token"
77         client = TestClient(app)
78
79         mock_fetch_asset_details.return_value = []
80         response = client.post(
81             "/jobs/fetch_all_asset_details",
82             content=json.dumps({"event": call_data, "token": fetch_token}),
83             headers={"Authorization": "Bearer your-token"},
84         )
85         assert response.status_code == 200
86
87
88 def test_fetch_asset_details():

```

```

89     with patch(
90         "src.services.products_import.fetch_sync_force_asset_details.handle"
91     ) as mock_fetch_asset_details:
92         from src.app import app
93
94         call_data = {}
95         fetch_token = "your_access_token"
96         client = TestClient(app)
97
98         mock_fetch_asset_details.return_value = []
99         response = client.post(
100             "/jobs/fetch_asset_details",
101             content=json.dumps({"event": call_data, "token": fetch_token}),
102             headers={"Authorization": "Bearer your-token"},
103         )
104         assert response.status_code == 200
105         assert response.json() == []
106
107
108 def test_fetch_product_ids():
109     with patch(
110         "src.services.products_import.fetch_sync_force_product_ids.handle"
111     ) as mock_fetch_product_ids:
112         from src.app import app
113
114         call_data = {}
115         fetch_token = "your_access_token"
116         client = TestClient(app)
117
118         mock_fetch_product_ids.return_value = []
119
120         response = client.post(
121             "/jobs/fetch_product_ids",
122             content=json.dumps({"event": call_data, "token": fetch_token}),
123             headers={"Authorization": "Bearer your-token"},
124         )
125         assert response.status_code == 200
126         assert response.json() == []
127
128
129 def test_fetch_all_product_details():
130     with patch(
131         "src.services.products_import.fetch_sync_force_product_details.handle"
132     ) as mock_fetch_product_details:
133         from src.app import app
134
135         call_data = {}
136         fetch_token = "your_access_token"
137         client = TestClient(app)
138
139         mock_fetch_product_details.return_value = []
140         response = client.post(
141             "/jobs/fetch_all_product_details",
142             content=json.dumps({"product_data": call_data, "token": fetch_token}),

```

```

143         headers={"Authorization": "Bearer your-token"},
144     )
145     assert response.status_code == 200
146
147
148 def test_fetch_product_details():
149     with patch(
150         "src.services.products_import.fetch_sync_force_product_details.handle"
151     ) as mock_fetch_product_details:
152         from src.app import app
153
154         call_data = {}
155         fetch_token = "your_access_token"
156         client = TestClient(app)
157
158         mock_fetch_product_details.return_value = []
159         response = client.post(
160             "/jobs/fetch_product_details",
161             content=json.dumps({"product_data": call_data, "token": fetch_token}),
162             headers={"Authorization": "Bearer your-token"},
163         )
164         assert response.status_code == 200
165         assert response.json() == []
166
167
168 def test_fetch_proposition():
169     with patch(
170         "src.services.products_import.fetch_sync_force_proposition.handle"
171     ) as mock_fetch_proposition:
172         from src.app import app
173
174         fetch_token = "your_access_token"
175         client = TestClient(app)
176
177         mock_fetch_proposition.return_value = "proposition_data"
178         response = client.post(
179             "/jobs/fetch_proposition",
180             content=json.dumps({"token": fetch_token}),
181             headers={"Authorization": "Bearer your-token"},
182         )
183         assert response.status_code == 200
184         assert response.json() == "proposition_data"
185
186
187 def test_fetch_attributes():
188     with patch(
189         "src.services.products_import.fetch_sync_force_attributes.handle"
190     ) as mock_fetch_attributes:
191         from src.app import app
192
193         fetch_token = "your_access_token"
194         client = TestClient(app)
195
196         mock_fetch_attributes.return_value = "attributes_data"

```



```

197     response = client.post(
198         "/jobs/fetch_attributes",
199         content=json.dumps({"token": fetch_token}),
200         headers={"Authorization": "Bearer your-token"},
201     )
202     assert response.status_code == 200
203     assert response.json() == "attributes_data"
204
205
206 def test_send_notification():
207     with patch(
208         "src.services.products_import.sync_force_send_notification.handle"
209     ) as mock_send_notification:
210         from src.app import app
211
212         call_data = {}
213         notification_data = {"event": call_data}
214         client = TestClient(app)
215
216         mock_send_notification.return_value = "notification_sent"
217         response = client.post(
218             "/jobs/send_notification",
219             content=json.dumps(notification_data),
220             headers={"Authorization": "Bearer your-token"},
221         )
222         assert response.status_code == 200
223         assert response.json() == "notification_sent"

```

## V GitHub Actions

```
1 name: PR Checks
2
3 # Workflow wordt geactiveerd bij een pull request naar de 'main' branch
4 on:
5   pull_request:
6     branches:
7       - main
8
9 jobs:
10  pr-checks:
11    # Gebruik het laatste Ubuntu-besturingssysteem voor de runner
12    runs-on: ubuntu-latest
13
14    steps:
15      # Stap 1: Haal de code op uit de repository
16      - name: Checkout code
17        uses: actions/checkout@v4
18
19      # Stap 2: Stel Python 3.9 in als de Python-versie voor de workflow
20      - name: Set up Python 3.9
21        uses: actions/setup-python@v4
22        with:
23          python-version: 3.9
24
25      # Stap 3: Cache Poetry packages om de afhankelijkheden sneller te installeren bij
26      ↪ toekomstige runs
27      - name: Cache Poetry packages
28        uses: actions/cache@v3
29        with:
30          path: |
31            ~/.cache/pypoetry # Cache voor gedownloade Poetry packages
32            ~/.virtualenvs    # Cache voor virtuele omgevingen
33          key: ${{ runner.os }}-poetry-{{ hashFiles('poetry.lock') }} # Cache-key op
34          ↪ basis van OS en hash van poetry.lock
35          restore-keys: |
36            ${{ runner.os }}-poetry- # Als exact match niet wordt gevonden, probeer
37            ↪ dan op basis van dit sleutelpatroon te herstellen
38
39      # Stap 4: Installeer en configureer Poetry
40      - name: Install and Configure Poetry
41        run: |
42          curl -sSL https://install.python-poetry.org | python3 - # Installeer Poetry
43          ↪ via een script
44          export PATH="$HOME/.local/bin:$PATH" # Voeg Poetry toe aan
45          ↪ het pad
46          poetry config virtualenvs.create true # Configureer Poetry
47          ↪ om een virtuele omgeving te maken
48
49      # Stap 5: Maak een test .env-bestand door een kopie van .env.test aan te maken
50      - name: Create .env file
```

```

45     run: cp .env.test .env # Kopieer .env.test naar .env zodat de testomgeving
      ↪ juiste configuraties heeft
46
47     # Stap 6: Bouw en start de Docker-container in de achtergrond
48     - name: Build Docker Container
49     run: docker compose up -d --build # Docker Compose bouwt en start de container
      ↪ in detach-modus (achtergrond)
50
51     # Stap 7: Voer de tests uit binnen de Docker-container
52     - name: Run Tests in Docker
53     run: |
54         docker compose exec app poetry run pytest tests/ # Voer pytest uit binnen de
      ↪ container met Poetry
55     env:
56     ENV: test # Stel de ENV-variabele in op 'test' om aan te geven dat het om een
      ↪ testomgeving gaat

```

```

1 # Specificeer de Python-versie als een variabele, zodat je eenvoudig kunt wisselen tussen
  ↪ versies
2 ARG PYTHON_VERSION=3.9
3
4 # Gebruik een lichte versie van de officiële Python Docker image gebaseerd op Debian
  ↪ Buster
5 FROM python:${PYTHON_VERSION}-slim-buster
6
7 # Update de apt-get package manager en installeer noodzakelijke build dependencies (gcc
  ↪ voor het bouwen van C-extensies,
8 # python3-dev voor Python headers) en de Azure CLI voor interactie met Azure services.
9 RUN apt-get update && apt-get install -y \
10     gcc \ # Compiler voor het bouwen van Python C-extensies
11     python3-dev \ # Headers en libraries voor Python development
12     curl \ # Utility om data te downloaden (zoals de Azure CLI installer)
13     && curl -sL https://aka.ms/InstallAzureCLIDeb | bash \ # Installeer de Azure CLI
14     && rm -rf /var/lib/apt/lists/* # Verwijder de apt-lijsten om ruimte te besparen
15
16 # Maak een directory voor de applicatie
17 RUN mkdir /app
18
19 # Stel de huidige werkdirectory in op /app
20 WORKDIR /app
21
22 # Kopieer het bestand 'pyproject.toml' naar de /app directory. Dit bevat de
  ↪ projectconfiguratie en dependencies
23 COPY pyproject.toml /app
24
25 # Installeer de 'poetry' dependency manager en configureer deze om geen virtuele
  ↪ omgevingen aan te maken
26 RUN pip install poetry \ # Installeer poetry via pip
27     && poetry config virtualenvs.create false # Configureer poetry om de globale
  ↪ Python omgeving te gebruiken
28
29 # Nogmaals de werkdirectory instellen op /app (hoewel dit al bovenaan werd ingesteld,
  ↪ wellicht overbodig)
30 WORKDIR /app

```

```
31
32 # Kopieer de volledige projectinhoud naar de container
33 COPY . .
34
35 # Stel een standaard AWS-regio in als omgevingsvariabele (voor mogelijk gebruik binnen de
  ↳ applicatie)
36 ENV AWS_DEFAULT_REGION=eu-west-1
37
38 # Installeer de projectafhankelijkheden zoals gedefinieerd in 'pyproject.toml' met poetry
39 RUN poetry install
40
41 # Stel poort 8000 open voor toegang (deze poort kan worden gebruikt voor bijvoorbeeld een
  ↳ webserver)
42 EXPOSE 8000
43
44 # Voeg het pad /app toe aan de Python path, zodat Python-modules gemakkelijk kunnen
  ↳ worden gevonden
45 ENV PYTHONPATH=/app
46
47 # Definieer het standaard commando dat wordt uitgevoerd wanneer de container start.
48 # In dit geval, wordt een shell script 'start.sh' uitgevoerd dat mogelijk de Azure CLI
  ↳ gebruikt om in te loggen.
49 CMD ["/app/start.sh", "az login"]
```

## VI Interface

```
1 from abc import abstractmethod
2 from typing import Optional, Protocol
3
4 from src.shared.result import Result
5
6
7 class StorageInterface(Protocol):
8     @abstractmethod
9     def get_object(self, filename: str) -> Result:
10         """Return the {cloud provider} object for a given filename, or an error if it
11         ↪ doesn't exist."""
12
13     @abstractmethod
14     def get_objects(self, path: Optional[str] = "") -> Result:
15         """Returns a list of {cloud provider} objects for a given path. By default, the
16         ↪ objects in the root folder
17         are returned."""
18
19     @abstractmethod
20     def delete_object(self, filename: str) -> Result:
21         """Deletes a {cloud provider} object with a given filename."""
22
23     @abstractmethod
24     def put_object(self, filename: str, file_contents: bytes) -> Result:
25         """Creates an {cloud provider} object with the specified filename and file
26         ↪ contents."""
27
28     @abstractmethod
29     def object_exists(self, filename: str) -> bool:
30         """Returns a True or False value if a given file exists or not."""
```

## VII BLOB Client

```
1 from typing import Dict, Optional, Union
2
3 from azure.core.exceptions import AzureError
4 from azure.storage.blob import BlobServiceClient
5
6 from src.settings import get_default_logger
7 from src.shared.env import Env
8 from src.shared.result import Err, Ok, Result
9 from src.shared.utils.storage_interface import StorageInterface
10
11 Env = Env()
12
13
14 class BLOBClient(StorageInterface):
15     """
16     This is an early implementation of a BLOB client that uses Azure Blob Storage.
17     It is a work in progress and will be improved in the future.
18     """
19
20     def __init__(self):
21         self.container_name = Env.azure_blob_container_name
22         self.blob_service_client = BlobServiceClient.from_connection_string(
23             Env.azure_storage_connection_string
24         )
25         self.logger = get_default_logger()
26
27         """If you want to use it passwordless, you can use the following code:
28         from azure.identity import DefaultAzureCredential
29
30         def __init__(self, container_name: str, account_url: str):
31             self.container_name = container_name
32             self.default_credential = DefaultAzureCredential()
33             self.blob_service_client = BlobServiceClient(
34                 account_url=account_url, credential=self.default_credential)
35             )
36             self.logger = get_default_logger
37         """
38
39     def get_object(
40         self, blob_name: str, **kwargs
41     ) -> Result[Dict[Union[bytes, str], str], AzureError]:
42         """Returns the blob object for the given path, or an error if it doesn't
43         ↪ exist."""
44         try:
45             # Get the blob client for the specified path
46             blob_client = self.blob_service_client.get_blob_client(
47                 container=self.container_name, blob=blob_name
48             )
49             # Download the blob content
50             blob = blob_client.download_blob(**kwargs)
51             content = {
```

```

51         "content": blob.readall(),
52         "last_modified": str(blob_client.get_blob_properties().last_modified),
53     }
54     # Return the content wrapped in an Ok result
55     return Ok(content)
56 except AzureError as e:
57     self.logger.warning(f"Unable to load object for path '{blob_name}': {e}")
58     return Err(e)
59
60 def get_objects(self, path: Optional[str] = "") -> Result[list, AzureError]:
61     """
62     Returns a list of blob objects. By default, the objects in the container's root
63     ↪ folder
64     """
65     try:
66         # Get a reference to the container client
67         container_client = self.blob_service_client.get_container_client(
68             self.container_name
69         )
70         # Get an iterable of blob objects that match the specified path
71         blob_iterable = container_client.list_blobs(name_starts_with=path)
72         # Collect all the blobs into a list
73         blobs = list(blob_iterable)
74         return Ok(blobs)
75     except AzureError as e:
76         self.logger.warning(f"Failed to fetch objects at path {path}: {e}")
77         return Err(e)
78
79 def delete_object(self, blob_name: str) -> Result[None, AzureError]:
80     """Deletes a blob object with a given path."""
81     try:
82         # Get the blob client for the specified path
83         blob_client = self.blob_service_client.get_blob_client(
84             container=self.container_name, blob=blob_name
85         )
86         # Delete the blob
87         blob_client.delete_blob()
88         return Ok(None)
89     except AzureError as e:
90         self.logger.warning(f"Unable to delete object for path '{blob_name}': {e}")
91         return Err(e)
92
93 def put_object(
94     self, blob_name: str, file_contents: Union[bytes, str]
95 ) -> Result[None, AzureError]:
96     """Creates a blob object with the specified filename and file contents to an
97     ↪ Azure
98     BLOB storage container."""
99     try:
100         blob_client = self.blob_service_client.get_blob_client(
101             container=self.container_name, blob=blob_name
102         )
103         uploaded_blob = blob_client.upload_blob(file_contents, overwrite=True)
104         return Ok(uploaded_blob)

```

```
103     except AzureError as e:
104         self.logger.warning(
105             f"Failed to put object with filename '{blob_name}': {e}"
106         )
107         return Err(e)
108
109     def object_exists(self, blob_name: str) -> bool:
110         """Returns a True or False value if a given file exists or not."""
111         if not blob_name or blob_name == "":
112             self.logger.warning("Blob name cannot be empty or None.")
113             raise ValueError("Blob name cannot be empty or None.")
114         blob_client = self.blob_service_client.get_blob_client(
115             container=self.container_name, blob=blob_name
116         )
117         return blob_client.exists()
```



## VIII Terraform IaC

```
1 # -----
2 # Azure Log Analytics Workspace
3 # -----
4 # Azure Log Analytics Workspace is a service that allows you to collect and analyze data
5 ↪ generated by resources in your cloud and on-premises environments.
6 resource "azurerm_log_analytics_workspace" "this" {
7     name                = "law-${var.application_name}"
8     location             = var.location
9     resource_group_name = var.resource_group
10    sku                  = "PerGB2018"
11    retention_in_days    = 30
12
13    tags = var.tags
14 }
15
16 # -----
17 # Azure Container App Environment
18 # -----
19 # Azure Container App Environment is a service that provides a fully managed environment
20 ↪ for running containerized applications.
21 resource "azurerm_container_app_environment" "this" {
22     name                = "caenv-${var.application_name}"
23     location             = var.location
24     resource_group_name = var.resource_group
25     log_analytics_workspace_id = azurerm_log_analytics_workspace.this.id
26     infrastructure_subnet_id = var.intra_subnet_id
27     infrastructure_resource_group_name = var.resource_group
28
29     workload_profile {
30         name                = var.application_name
31         workload_profile_type = var.workload_profile_type
32     }
33
34     tags = var.tags
35 }
36
37 # -----
38 # Azure Container App Job
39 # -----
40 # Azure Container App Job is a service that allows you to run scheduled or event-driven
41 ↪ jobs in a containerized environment.
42 resource "azurerm_container_app_job" "this" {
43     name                = "cajob-${var.application_name}"
44     location             = var.location
45     resource_group_name = var.resource_group
46     container_app_environment_id = azurerm_container_app_environment.this.id
47
48     replica_timeout_in_seconds = 60
49     replica_retry_limit        = 10
50
51     schedule_trigger_config {
```

```
49     cron_expression = var.cron_expression
50 }
51
52 template {
53     container {
54         image = var.image_name
55         name  = var.application_name
56         cpu   = var.cpu
57         memory = var.memory
58         command = var.command
59     }
60 }
61
62 tags = var.tags
63 }
64
```

## IX Aflevering

### Oplevering Syncforce AWS => Azure

Dinsdag, 22 oktober - 10:30 - 11:30am



Deelnemen via Google Meet



## X Zelfreflecties van de eerste maand

Tijdstempel	Hoe vaak heb ik deze week op collega's afgestapt voor hulp of overleg?	Met wie van mijn collega's heb ik het meest samengewerkt deze week, en waarom?	Hoe goed heb ik rekening gehouden met de kennis en expertise van mijn collega's?	Heb ik tijdens het werk gemerkt dat ik bij de juiste persoon aanklopte voor de juiste vraag?	Wat vond ik moeilijk in het benaderen van collega's	Voelde ik me zelfverzekerd of terughoudend bij het benaderen van collega's?	Zijn er specifieke momenten geweest waarop ik had moeten samenwerken, maar dat niet heb gedaan?	Welke invloed had de samenwerking deze week op mijn werkresultaat?	Heb ik gebruik gemaakt van eerdere ervaringen om deze week beter samen te werken?	Wat zou ik volgende week anders doen in mijn samenwerking met collega's
24-9-2024 8:52:13	Niet	Met Gijs Straathof, hij is mijn stagebegeleider	Gemiddeld	Meestal	Ik ken nog niet iedereen en weet nog niet waar ze goed in zijn.	Terughoudend	Nee	Positief	Nee	Beter kijken naar wie ik kan stappen.
25-9-2024 14:36:09	3-4 keer	Guus en Gijs, dit heeft te maken met dat ik een Terraform configuratie moet gaan bouwen voor holmatro-products-sync.	Zeer goed	Altijd	Mijzelf ervan overtuigen dat vragen stellen niet iets vervelends is.	Neutraal	Nee	Positief	Ja, ik ben erachter gekomen dat ik aan Guus veel vragen kan stellen over Cloud	Niks, ik ga gewoon weer vragen stellen wanneer dat nodig is.
30-9-2024 9:38:08	3-4 keer	Met Guus heb ik veel gewerkt aangezien hij veel weet over de cloud en met mijn stagebegeleider Gijs aangezien hij mij met veel kleine taakjes kan helpen.	Zeer goed	Meestal	De eerste keer naar iemand toe stappen is het lastigst, daarna wordt het steeds makkelijker	Neutraal	Nee, als ik echt hulp nodig had heb ik er ook om gevraagd	Zeer positief	Ja	Niks
07-10-2024 8:35:41	5+ keer	Ik ben weer vaak naar Guus en Gijs gestapt de afgelopen week om hulp te vragen voor Terraform en het opzetten van de productie omgeving in Azure.	Zeer goed	Altijd	Naar mijn begeider toestappen gaat heel erg makkelijk evenals Guus, aangezien ze beiden heel erg open zijn om te helpen. Dus het ging heel erg goed.	Zelfverzekerd	Nee	Zeer positief	Ja	Niks, gewoon door blijven met vragen stellen en sparren over verschillende ideeën om hopelijk tot een zo goed mogelijk eindresultaat te komen.

## XI Analyse

### **Giraf Analyse**

Een analyse van kosten en prestaties  
22 - 10 - 2024

Harborn BV  
Westblaak 39  
3012 KD Rotterdam  
[www.harborn.com](http://www.harborn.com)  
+31(0)104365050  
[info@harborn.com](mailto:info@harborn.com)

# Hoe kunnen we de AI-service van Giraf optimaliseren op prestaties?

Een prestatieanalyse van de AI-service van Giraf rekening houdend met kosten.



# Inhoudsopgave

<b>Inhoudsopgave</b> .....	<b>3</b>
<b>Inleiding</b> .....	<b>3</b>
<b>Kosten</b> .....	<b>4</b>
<b>Prestaties</b> .....	<b>6</b>
Classificatie.....	6
Named Entity Recognition (NER).....	8
Implementatie.....	9
<b>Conclusie</b> .....	<b>11</b>

## Inleiding

Giraf is een AI-gestuurde oplossing voor de vastgoed- en verhuurbranche die helpt met het automatisch beantwoorden van e-mails. Het systeem leest alle binnenkomende e-mails uit in de geselecteerde inbox en genereert daarop passende antwoorden. Deze antwoorden kunnen eerst bekeken worden of meteen direct gestuurd worden. Ook kan Giraf gekoppeld worden aan databases, CRM- of ERP-systemen om e-mailverkeer, zoals rondom bezichtigingen, volledig te automatiseren.

Helaas vertoont Giraf tekenen van veroudering en voldoet niet meer aan de moderne standaarden binnen de AI-wereld die alsmaar verandert en vernieuwt. Daarnaast bevat het systeem regelmatig fouten, wat de efficiëntie en de gebruikerservaring negatief beïnvloedt. Allereerst leidt het tot verminderde klanttevredenheid doordat onnauwkeurige of trage antwoorden frustratie veroorzaken bij klanten en potentiële huurders. Financieel gezien kunnen zulke fouten leiden tot gemiste kansen en verhoogde operationele kosten, omdat medewerkers extra tijd moeten besteden aan het corrigeren van fouten. En dan kan het ook zo zijn dat klanten hun contract niet meer willen verlengen, wat uiteindelijk voor Harborn digital minder opbrengsten oplevert.



## Prestaties

Om goed te kunnen inzien hoe de AI-assistent presteert heb ik eerst gekeken naar verschillende alternatieven. Zelf een volwaardige AI bouwen leek mij me onrealistisch aangezien het buiten mijn expertise valt en ik ook geen tijd heb om dat zelf in elkaar te zetten. Dus heb ik gekeken naar verschillende manieren om dit makkelijker te maken. In de huidige context, waarin geavanceerde AI-technologie op basis van Large Language Models (zoals Chat GPT) wijdverspreid bekend is, is het logisch om zulk soort oplossingen te overwegen voor dit onderzoek. Chat GPT zelf is wel veel te duur hiervoor aangezien je moet betalen voor tokens (1 token is 4 karakters). Daarnaast werken we ook met sensitieve gegevens. Na wat googlen en zoeken op het internet ben ik gestuit op Ollama (Ollama, 2024) een open source tool om zelf je eigen AI modellen te runnen waar je ook maar wil. Dit betekent ook dat alle gegevens lokaal blijven en nergens anders heen gaat, wat zorgt voor betere privacy.

Om rekening te houden met het laag houden van kosten ben ik gaan zoeken naar een AI model dat op goedkopere hardware kan runnen. Je hebt namelijk enorme verschillen tussen verschillende modellen en de hoeveelheid parameters die worden meegegeven aan een prompt. De verschillende zijn 0.5B, 1.0B, 1.5B, 2B, 3B, 7B, 8B, 9B, 14B, 32B en 72B, hierbij staat de B voor Miljard instructies. Dit zijn niet alle verschillende groottes van de hoeveelheid instructies die meegegeven worden. Dus ik heb vooral gekeken naar modellen die een 3B formaat hebben. Uiteindelijk was ik ook bij een artikel gekomen op Medium (Guide, 2024) die 4 verschillende modellen tegen elkaar vergelijkt en uit het artikel was gekomen dat de Llama3.2 3B het beste is. Verder heb ik zelf ook handmatig gekeken welke beter werkt door steekproeven uit te voeren, en Llama3.2 3B (Meta, 2024) was naar mijn gevoel het beste op het moment.

Voor de vergelijking tussen de AI-assistent en Llama3.2 3B heb ik een steekproef van 200 e-mails uit de database waar we elke mail van een klant archiveren.

## Classificatie

Voor de classificatie van e-mails (om te bepalen of het om een bezichtiging gaat of iets anders), heb ik eerst handmatig alle e-mails beoordeeld en ingedeeld in twee categorieën: 'bezichtiging' of 'iets anders'. Daarna heb ik zowel de AWS-classificatie als de classificatie van het model Llama3.2 3B uitgevoerd. De resultaten heb ik in een JSON-bestand gezet, waarmee ik de nauwkeurigheid van beide modellen heb vergeleken.



Hieronder zie je wat je terugkrijgt van AWS, dus niet het een of de ander, maar beide met een bepaald gewicht. Om dit op te schonen heb ik alleen degene met het hoogste gewicht van de twee gepakt.

```
Unset
{
  "OTHER": 0.6408942341804504,
  "VIEWING_REQUEST": 35910582542419434,
}
```

Om goede antwoorden uit de LLM te krijgen heb ik de volgende tekst meegegeven aan elke prompt/mail die naar de LLM wordt gestuurd.

```
Unset
Analyze the email below to determine if it is a "viewing request."

A "viewing request" is an email in which the sender requests to schedule,
confirm, or change a viewing of a home or property.
A "viewing request" email may also be titled "Contact Request."

If the email is part of a thread (indicated by "Re"), ignore all previous
correspondence and focus only on the most recent message. This is the
section before the lines that begin with '>'.

Reply only with 1 of the following responses in all caps:
- "VIEWING_REQUEST"
- "OTHER"
```

Hieronder staat een overzicht van de prestaties van beide modellen:

Naam	Correct	Totaal	Percentage	Tijd
Llama3.2 3B	115	200	57,5%	20 minuten
AWS	146	200	73%	20 minuten

Uit de tabel blijkt dat het **Llama3.2 3B**-model 15,5% minder nauwkeurig is dan het **AWS**-model, met respectievelijk 57,5% en 73% nauwkeurigheid.





## Named Entity Recognition (NER)

Om de kwaliteit van een uitgevoerde Named Entity Recognition (NER) te beoordelen, heb ik mijn evaluatie verdeeld in drie categorieën:

1. **Incorrect:** De extractie bevat geen enkele juiste informatie, of de informatie is volledig fout.
2. **Half correct:** Een deel van de informatie is juist, maar er ontbreken belangrijke elementen of een substantieel deel is onjuist.
3. **Volledig correct:** De belangrijkste velden, zoals de naam van de persoon en het adres, zijn volledig correct en volledig gedekt door de extractie.

Om de verschillende antwoorden uit de LLM te krijgen is de onderstaande tekst meegegeven aan elke prompt die wordt uitgevoerd.

```
Unset
""Please perform a Named Entity Recognition (NER) operation on the email
below and respond only in the following JSON format:
{
  "NAME": "NAME_HERE",
  "AGE": "AGE_HERE",
  "OCCUPATION": "OCCUPATION_HERE",
  "LIVINGSITUATION": "LIVING_SITUATION_HERE",
  "ADDRESS": "PHYSICAL_ADDRESS_HERE",
  "SALARY": "SALARY_HERE",
}

Please adhere strictly to the following guidelines:
1. For "NAME" enter only the person's full name. Leave blank if no name is
found.
2. For "AGE" enter only the age, if mentioned. Leave blank if no age is
found.
3. For "OCCUPATION" enter only the person's occupation, if mentioned. Leave
blank if no occupation is found.
4. For "LIVINGSITUATION" describe whether the person lives alone or with
more than one person. Leave blank if this is unclear.
5. For "ADDRESS" enter only the physical address of the home, if mentioned.
Leave blank if no physical address is available.
6. For "SALARY" enter only the salary amount if mentioned. Leave blank if no
salary is found.
7. Output only the above JSON format without any further explanation or
formatting.""
```



Hieronder een vergelijking van de prestaties van de modellen:

Naam	Correct	Half	Incorrect	Totaal	Tijd
Llama3.2 3B	140 (70%)	42 (21%)	18 (9%)	200	20 minuten
AWS	58 (29%)	25 (12,5%)	117 (58,5%)	200	60 minuten

Uit deze vergelijking blijkt dat **Llama3.2 3B** aanzienlijk beter presteert dan **AWS** in termen van nauwkeurigheid, met 70% volledig correcte resultaten vergeleken met slechts 29% voor AWS. Bovendien is **Llama3.2 3B** ook sneller, met een verwerkingstijd van 20 minuten, terwijl AWS er 60 minuten over doet. Dit kan te maken hebben met wat voor soort hardware er wordt gebruikt bij AWS, misschien duurt het voor **Llama3.2 3B** ook langer vanwege de gelimiteerde hardware.

De prestaties zijn over het algemeen correct, maar het kon ook gebeuren dat de gegeven uitvoer niet in het correcte formaat is, met geluk kon je dan nog een accolade toevoegen om het in een correct JSON formaat te hebben. Maar als er dan ook een aanhalingsteken ontbrak dan is het wat lastiger op te lossen.

## Implementatie

Het was niet lastig of ingewikkeld om lokaal een LLM werkende te krijgen. Op de website van Ollama kan je een installer downloaden, wat dan een CLI tool voor je installeert. Daarna kan je in de terminal heel makkelijk `ollama run llama3.2` typen en wordt het automatisch gedownload. Zodra dat klaar is kan je op 2 manieren met de LLM praten, de eerste manier is heel makkelijk in de terminal prompts uitvoeren en ermee praten. De tweede manier is door middel van http requests, want Ollama runt in de achtergrond een HTTP REST API op port 11434.

Op de onderstaande manier kan je het dus makkelijk aanroepen. En voor meer informatie over alle opties die je kan meesturen in zo'n API request verwijs ik je naar de documentatie op Github (Ollama, 2024).

```
Unset
curl http://localhost:11434/api/generate -d '{
  "model": "llama3.2",
  "prompt": "Why is the sky blue?"
}'
```

Daarnaast heb je voor python een library die zo'n request voor je abstraheert waardoor je het zonder moeite kan aanroepen en het wat makkelijker kan integreren in de huidige codebase voor de AI-assistent. De integratie duurde dan ook helemaal niet zo lang en was over het algemeen niet pijnlijk. Ik moet er wel bij zeggen dat je rekening moet houden met het feit dat een LLM soms niet de juiste antwoorden



teruggeeft en dat je dan nog wat fallbacks moet creëren voor een soepelere ervaring. Hier heb ik wel wat meer tijd aan moeten besteden tijdens dit kleine onderzoek.



## Kosten

In AWS gebruiken we een serverless hosting service die specifiek bedoeld is om AI operaties uit te voeren met de naam SageMaker. Amazon SageMaker is een cloudgebaseerd machine learning platform dat de creatie, training en implementatie door ontwikkelaars van machine learning modellen in de cloud mogelijk maakt. Het kan worden gebruikt om ML/AI-modellen te implementeren op embedded systemen en edge-apparaten. De maandelijkse kosten van de service bedragen gemiddeld \$15, dit komt op jaarbasis uit op ongeveer \$180, ongeveer €164.

Tijdens de eerste 6 maanden van het trainen van het model heeft het zo'n \$1763, ongeveer €1635, gekost in SageMaker. Daarnaast zijn er 145 punten meegegeven aan User Stories voor 'AI & training', met zo'n 4 uur per punt komt dat op 580 uur van actieve inspanning om het uiteindelijk model te trainen.

De onderstaande berekening is gebruikt om in te schatten hoeveel en hoe lang de serverless functie heeft gewerkt.

```
Unset
X = 0,0000520
month = 6,27
seconds = month / X = 120.576,92308
minutes = seconds / 60 = 2.009,61538
hours = minutes / 60 = 33,49359
```

Service total	\$142,90	Uren gewerkt
2024-01-01	\$17,40	92,9
2024-02-01	\$15,60	83,33
2024-03-01	\$16,08	85,9
2024-04-01	\$15,88	84,82
2024-05-01	\$16,79	89,69
2024-06-01	\$15,20	81,19
2024-07-01	\$13,32	71,15
2024-08-01	\$12,51	64,63
2024-09-01	\$13,85	73,98
2024-10-01	\$6,27	33,49
<b>Totaal in dollar</b>	<b>\$142,90</b>	
<b>Totaal uren serverless</b>		<b>761,08 Uur</b>
<b>Totaal in Euro</b>	<b>€ 130,79</b>	<b>€ 0,17 per uur</b>



Uit de tabel kunnen we afleiden dat het runnen van SageMaker in een serverless instance zo'n 17 eurocent per uur kost.

Om alvast te kijken wat potentiële kosten kunnen zijn als we zouden overstappen op een ander AI model heb ik een kijkje genomen op de AWS SageMaker Pricing (Amazon, n.d.) website gekeken. Ik ben er ook achter gekomen dat de serverless instances die we op het moment gebruiken om het AI model op te runnen alleen een cpu (Amazon, 2023) gebruiken. Terwijl je voor het gebruiken van een LLM ook een grafische kaart (Meta, 2024) en genoeg RAM nodig hebt. Dat betekent dan dat je bij het kopje Accelerated Computing (Amazon, n.d.) van Real-Time Inference, wat constant aan blijft, moet kijken waar elke instance tenminste 1 gpu heeft. En dan liggen de prijzen per uur tussen de \$0,33 en \$40,706 per uur.



## Conclusie

Uit de testresultaten blijkt dat de **AI-assistent** 15,5% nauwkeuriger is in het classificeren van e-mails dan het lokale **Llama3.2 3B**-model. Dit was onverwacht, omdat ik verwachtte dat de LLM het beter zou doen. Mogelijk komt dit door een minder optimale prompt voor de LLM, wat zou kunnen verklaren waarom het model niet optimaal presteerde. Aan de andere kant presteerde het LLM-model aanzienlijk beter in **Named Entity Recognition (NER)**. Het model herkende 41% meer belangrijke entiteiten correct in vergelijking met de AI-assistent op AWS. Dit toont aan dat de LLM beter is op het gebied van NER, wat een belangrijke overweging is voor toekomstige implementaties. Daarnaast was het integreren van de LLM helemaal niet lastig en kostte het niet al te veel tijd. Alleen moet je er wel rekening mee houden dat niet altijd het correcte formaat wordt aangehouden bij de teruggegeven antwoorden en dat je het dus moet corrigeren of een fallback moet hebben.

De kosten van de huidige implementatie van de **Giraf AI-assistent** zijn uitzonderlijk laag. De kosten voor het draaien van de AI op **AWS SageMaker instances** bedragen slechts €0,17 per uur. Dit is een zeer efficiënte oplossing, en er is op dit vlak weinig ruimte voor verdere optimalisatie. Sterker nog, de kosten zullen waarschijnlijk toenemen als we een LLM in de cloud willen implementeren. De reden hiervoor is dat de serverless instance van onze AI alleen een cpu gebruikt om alle operaties uit te voeren. Terwijl je voor een LLM tenminste ook een gpu nodig hebt.



## Bronnen

- Amazon. (n.d.). *Machine Learning Service – Amazon SageMaker Pricing – AWS*. AWS.  
Retrieved October 29, 2024, from <https://aws.amazon.com/sagemaker/pricing/>
- Amazon. (2023, July 8). *Deploy LLM serverlessly on Sagemaker*. AWS re:Post.  
Retrieved October 29, 2024, from  
<https://repost.aws/questions/QUte8vPM2PTUKmJqV7YSnBKA/deploy-llm-serverlessly-on-sagemaker>
- Guide, S. (2024, May 13). *Ollama: The landscape for a powerful Opensource LLMs*.  
Medium. Retrieved October 29, 2024, from  
<https://medium.com/@omkamal/ollama-the-landscape-for-a-powerful-llm-from-meta-ai-6792d7dad718>
- Meta. (2024, September). *Llama 3.2 Requirements [What you Need to Use It?]*. Llama 3.1 AI. Retrieved October 29, 2024, from  
<https://llamamodel.com/requirements-3-2/#3B>
- Meta. (2024, September). *Llama 3.2 Requirements [What you Need to Use It?]*. Llama 3.1 AI. Retrieved October 29, 2024, from  
<https://llamamodel.com/requirements-3-2/>
- Ollama. (2024). Ollama. Retrieved October 29, 2024, from <https://ollama.com/>
- Ollama. (2024, September). *ollama/docs/api.md at main · ollama/ollama · GitHub*.  
GitHub. Retrieved October 29, 2024, from  
<https://github.com/ollama/ollama/blob/main/docs/api.md>



## XII Advies

### **Giraf Advies**

Optimalisatie van de AI service  
25 - 10 - 2024

Harborn BV  
Westblaak 39  
3012 KD Rotterdam  
[www.harborn.com](http://www.harborn.com)  
+31(0)104365050  
[info@harborn.com](mailto:info@harborn.com)



# **Mogelijkheden voor de optimalisatie van de AI-service Giraf**

Opdrachtgever: Harborn Digital, auteur: Douwe Zumkers, datum: 28 - 10 - 2024, betreft: De AI service Giraf



2

# Inhoudsopgave

<b>Inhoudsopgave</b> .....	<b>3</b>
<b>Inleiding</b> .....	<b>3</b>
<b>Samenvatting van het onderzoek</b> .....	<b>4</b>
<b>AI Act en licenties</b> .....	<b>5</b>
<b>Aanbeveling</b> .....	<b>6</b>
<b>Bronnen</b> .....	<b>7</b>

## Inleiding

Giraf is een AI-gestuurde oplossing voor de vastgoed- en verhuurbranche die helpt met het automatisch beantwoorden van e-mails. Het systeem leest alle binnenkomende e-mails uit in de geselecteerde inbox en genereert daarop passende antwoorden. Deze antwoorden kunnen eerst bekeken worden of meteen direct gestuurd worden. Ook kan Giraf gekoppeld worden aan databases, CRM- of ERP-systemen om e-mailverkeer, zoals rondom bezichtigingen, volledig te automatiseren.

De huidige AI mail assistent Giraf is al enige tijd in gebruik genomen, maar vertoont tekenen van veroudering. In een steeds sneller evoluerende wereld van AI is het essentieel dat dergelijke systemen voldoen aan de moderne eisen op het gebied van prestaties, kostenoptimalisatie en gebruikerservaring. De eerdere analyse van de prestaties van de Giraf assistent heeft waardevolle inzichten opgeleverd, met name in vergelijking met een lokaal lichtgewicht Language Learning Model (LLM), zoals Llama3.2 3B.

Dit adviesrapport bouwt voort op die analyse en heeft als doel om concrete aanbevelingen te doen voor het optimaliseren van de Giraf AI-service. Hierbij wordt gekeken naar de mogelijkheden om te verbeteren op het vlak van prestaties, waarbij de kosten lichtelijk mee worden genomen in het advies. In het rapport zullen we ingaan op de belangrijkste bevindingen uit de analyse, zoals de prestatieverschillen op het gebied van e-mail classificatie en Named Entity Recognition (NER) tussen Giraf en een lokaal LLM-model.



## Samenvatting van het onderzoek

Tijdens de analyse van Giraf (Harborn digital & Zumker, 2024) heb ik ervoor gekozen om de AI-service te vergelijken met een self-hosted LLM vanwege mijn gebrek aan expertise, tijd om een heel model te bouwen en het leek mij ook leuk om te proeven wat de mogelijkheden zijn. Daarnaast zijn Chat GPT achtige oplossingen zoals LLM's tegenwoordig enorm relevant en bijna iedereen heeft er ook ervaring mee of weet wat het is.

Tijdens het onderzoek is een prestatieanalyse uitgevoerd op de classificatie van e-mails en Named Entity Recognition (NER). Hiervoor zijn 200 e-mails handmatig gecontroleerd en gecategoriseerd als "bezichtiging" of "anders".

Uit de resultaten blijkt dat de AWS AI-service 73% nauwkeurigheid behaalde in e-mail classificatie (Harborn digital & Zumker, 2024, 4-5), terwijl de LLM 57,7% scoorde. Voor NER (Harborn digital & Zumker, 2024, 6-7) was de LLM echter aanzienlijk beter, met 70% volledig correcte extracties, 21% deels correcte, en slechts 9% incorrecte resultaten. De AWS AI-service scoorde daarentegen 29% volledig correct, 12,5% deels correct en 58,5% incorrect. Ook bleek de LLM sneller: 20 minuten voor de analyse versus 60 minuten bij de AWS AI-service.

Daarnaast was het implementeren van de LLM niet ingewikkeld, er is namelijk een open source oplossing met de naam Ollama (Ollama, 2024) waarmee je gemakkelijk lokaal een LLM aan de praat krijgt. Om gebruik te maken van de lokale LLM in python is er ook een kleine abstractie in de vorm van een library (Morgan, 2024), die het makkelijker maakt om ermee te werken.

Als laatste heb ik ook een kleine kostenanalyse (Harborn digital & Zumker, 2024, 9-10) gedaan om te kijken hoeveel de AI-service kost in AWS en wat de trainingskosten zijn. Per uur dat de AI-service haar werk doet kost het ons gemiddeld € 0,17. We hebben ongeveer 580 uur besteed aan het trainen van het model en het heeft ons ook € 1635 gekost om in AWS het model te trainen. Daarentegen gaat het wel een stuk duurder worden om een LLM in AWS te runnen. Want je moet overstappen op een andere service/instance binnen AWS die altijd aanstaat en tussen de 33 cent en 40 dollar per uur ligt.

Als je dieper op de analyse wilt ingaan, verwijs ik je graag naar het analyserapport (Harborn digital & Zumker, 2024).



## AI Act en licenties

Sinds juni 2024 heeft de Europese Unie een AI Act uitgebracht, de Europese Verordening 2024/1689 (Europese Unie, 2024), bekend als de AI Act. In dit artikel van IBM (IBM, 2024) is het wat makkelijker uitgelegd. Waarin heel veel verschillende regels in staan die bepalen of een AI veilig is en in Europa gebruikt mag worden. Dit gaat dan over het ontwikkelen van AI en LLM-modellen. Ontwikkelaars van LLM's moeten hierdoor complexe en tijdrovende processen volgen om te voldoen aan de transparantie- en verantwoordingsvereisten, wat kan leiden tot hogere ontwikkelingskosten en langzamere innovaties.

Dit gaat samen met het feit dat er strengere regels omtrent privacy en data vereisten. Veel LLM's vergen grote hoeveelheden data om getraind te worden en de AI Act stelt hier hogere en strengere eisen aan. Waarbij moet worden vrijgegeven waar de data vandaan is gehaald. Dit maakt het lastig om geschikte data te vinden en te gebruiken zonder juridische risico's of boetes.

De bovenstaande punten zorgen er ook voor dat Meta de Llama AI-modellen niet beschikbaar maakt voor Europeanen, omdat ze niet willen vrijgeven op wat voor data de modellen zijn getraind. Tijdens mijn onderzoek heb ik er echter wel gebruik van kunnen maken doordat het platform Ollama niet strikt bijhoudt wie wel of niet een LLM-model mag gebruiken. Hier was ik dus ook na mijn analyse achter gekomen. Waarschijnlijk hebben ze data van Facebook en Instagram gebruikt zonder daar toestemming voor te vragen en willen ze niet een grote boete daarvoor ontvangen.

Dus als we zelf een LLM of AI model willen implementeren in de Giraf service gaan we naar een ander model moeten zoeken dat goed werkt en aan alle regels van de AI Act voldoet. En ook is het handig om te kijken onder welke soort licentie de verschillende modellen zijn gepubliceerd. Sommige zijn namelijk onder een Apache 2.0-licentie en andere zijn weer onder een strengere licentie.



## Aanbeveling

Op basis van de resultaten van de gedane analyse heb ik een paar aanbevelingen voor het inzetten van AI- en LLM-modellen in de Giraf-service.

Voor het classificeren van e-mails is het handig om voor nu de AI-service in AWS nog steeds te gebruiken. Uit de analyse is gebleken dat de AI-service 15,3% beter is dan de LLM in het identificeren van de juiste e-mails. Bovendien zorgt dit voor lagere kosten, aangezien de serverless inference van AWS lage kosten heeft.

Voor Named Entity Recognition (NER)-operaties is het echter beter over te schakelen op een zelf gehoste LLM. Deze LLM maakt aanzienlijk minder fouten in het extraheren van de correcte informatie uit e-mails en voert deze operaties sneller uit dan de AI-service in AWS. Deze verbetering gaat wel zorgen voor hogere kosten, omdat het draaien van een GPU-gebaseerde instance duurder is. Naar mijn mening is dit echter gerechtvaardigd door de verbetering in prestaties.

Daarnaast is het zinvol om de mogelijkheid te onderzoeken om een LLM te finetunen. Door specifiek een LLM te trainen op data waarop Giraf is getraind, kunnen we ervoor zorgen dat een LLM beter wordt in classificaties en NER-operaties. Dit vraagt in eerste instantie een investering van tijd en geld, maar het kan zorgen voor betere prestaties en dus ook betere klanttevredenheid.

Ten slotte is het van belang om bij de inzet van AI-modellen te voldoen aan de eisen van de EU AI Act. Deze regelgeving stelt strikte eisen op het gebied van transparantie, privacy en verantwoording, wat betekent dat het cruciaal is dat de bron en juridische trainingsdata bekend zijn. Dit beperkt de keuzemogelijkheden, aangezien sommige modellen zoals Llama van Meta momenteel niet beschikbaar zijn in de EU en dus niet ingezet kunnen worden.

Door deze aanbevelingen op te volgen kan de Giraf-service die we nu aanbieden, meer toekomstbestendig worden. En zowel voldoen aan de regelgeving van de EU AI Act, als effectief inspelen op de technische vereisten van de dienst.



## Bronnen

Europese Unie. (2024, June 13). *Verordening - EU - 2024/1689 - EN - EUR-Lex*.

EUR-Lex. Retrieved October 29, 2024, from

[https://eur-lex.europa.eu/legal-content/NL/TXT/?uri=OJ%3AL\\_202401689](https://eur-lex.europa.eu/legal-content/NL/TXT/?uri=OJ%3AL_202401689)

Harborn digital & Zumker, D. (2024, October 24). *Giraf Analyse*.

IBM. (2024, September 20). *What is the EU AI Act?* IBM. Retrieved October 29, 2024,

from <https://www.ibm.com/topics/eu-ai-act>

Morgan, J. (2024, September 9). *ollama · PyPI*. PyPI. Retrieved October 29, 2024, from

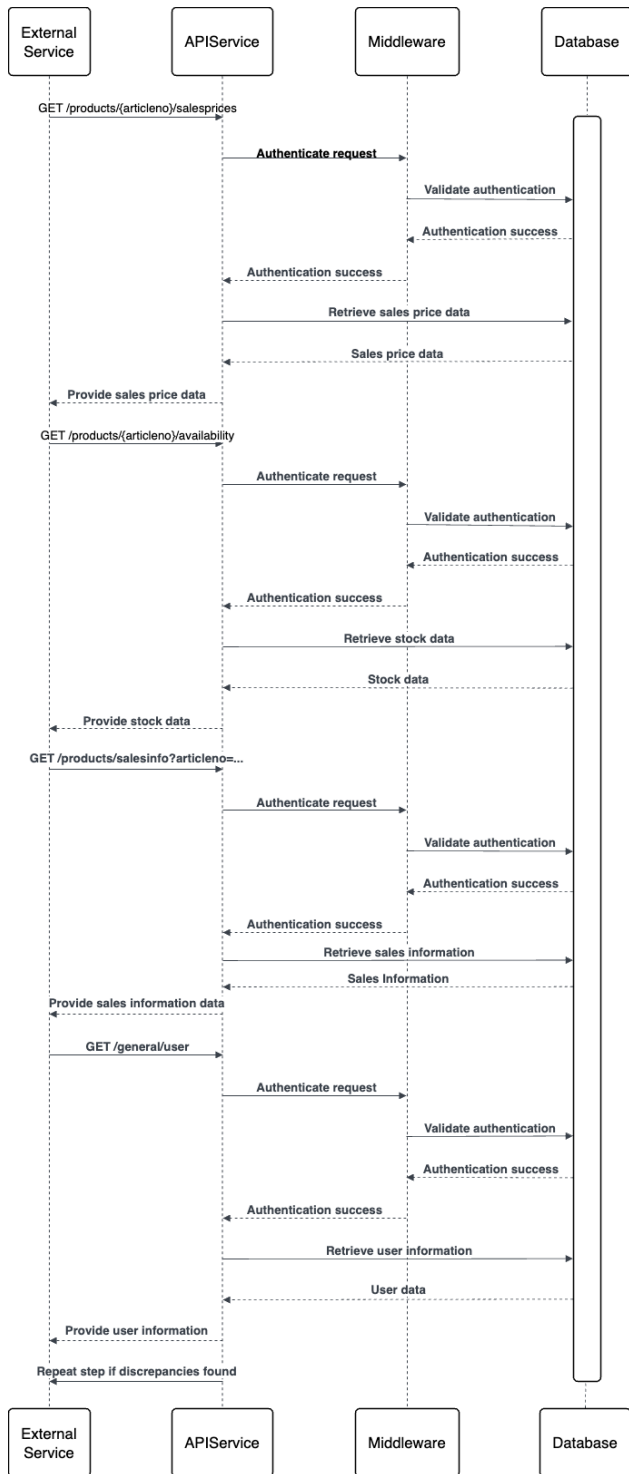
<https://pypi.org/project/ollama/>

Ollama. (2024). *Homepage of the Ollama website*. Ollama. Retrieved October 29, 2024,

from <https://ollama.com>

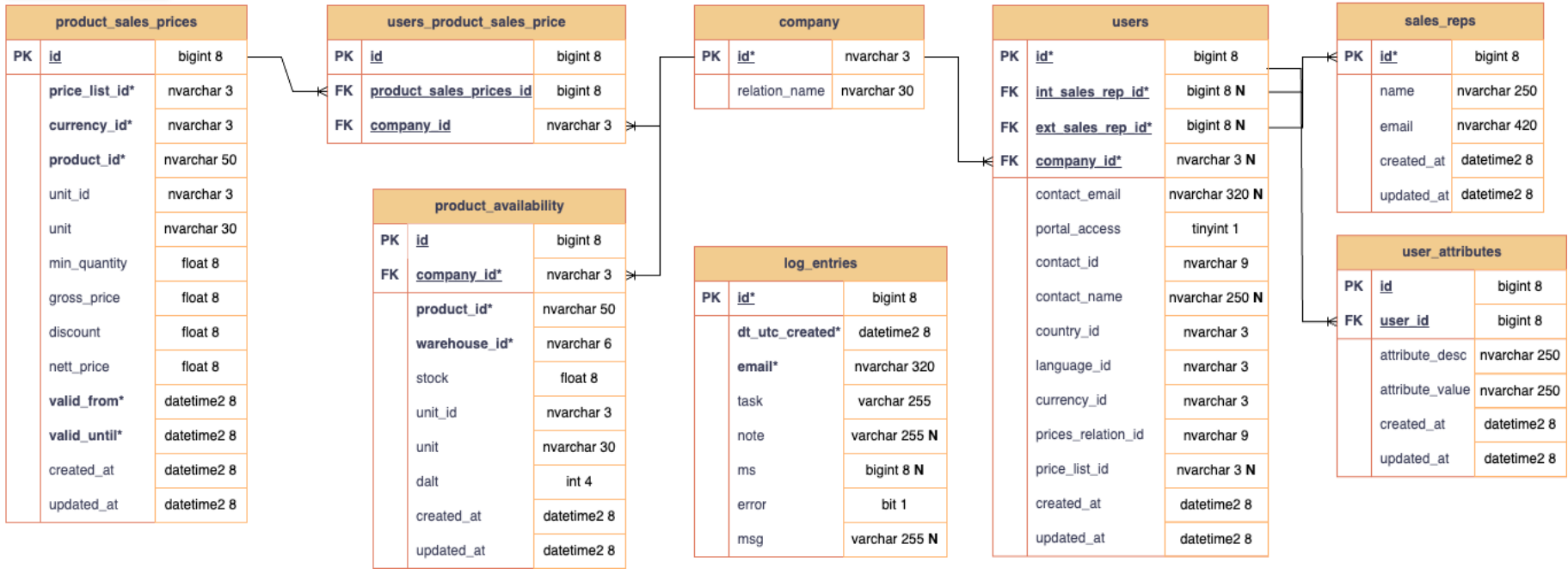


# XIII Sequence Diagram



# XIV ER Diagram

The \* after certain Attributes is there to indicate that the attribute should be indexed on all tables.





# XV Class Diagram

