

CONVERSATIE CHATBOT VOOR BEZOEKERS VAN CAPGEMINI

Software engineering – De Haagse Hogeschool Zoetermeer



Timo Frionnet - 15072924

31 mei 2019



Het ontwerpen van een conversatie chatbot prototype binnen Capgemini

Organisatie:	Capgemini
Opdrachtgever:	dhr. P. Tonen
Bedrijfsmentor:	dhr. P. Tonen
Begeleidend Examiner:	dhr. H. Al-Ers
Expert Examiner	dhr. T. Biegstraaten
Opleiding:	HBO ICT - Software Engineering Zoetermeer
Opleidingsinstituut:	De Haagse Hogeschool
Auteur:	Timo Frionnet
Plaats:	Utrecht Leidsche Rijn

Voorwoord

In de periode van 4 februari tot 31 mei 2019 heb ik mijn afstudeeropdracht uitgevoerd bij Capgemini. Capgemini heeft mij vooraf een aantal projecten aangeboden die uit verschillende innovatieve trends voortkomen. Zo was het bijvoorbeeld mogelijk om een project uit te voeren over de volgende onderwerpen: AI, AR/VR, IOT, blockchain of chatbots. Voorafgaande aan deze stage heb ik een minor Applied Data Science gevolgd aan de Haagse Hogeschool. Tijdens de minor heb ik vragen geanalyseerd die verwerkt zouden worden door de opdrachtgever in een chatbot. Tijdens dit project was er helaas geen mogelijkheid om zelf de chatbot te ontwikkelen. Dankzij dit project is mijn interesse gewekt ten aanzien van chatbots en natural language understanding (NLU). Toen ik hoorde dat er een opdracht beschikbaar was, waarbij er een chatbot ontwikkeld zou worden, heb ik hiervoor gekozen. Ik was zelf ook op zoek naar een andere uitwerking van een chatbot. Deze wens sloot dus goed aan op de focus naar een conversatie chatbot binnen mijn afstudeeropdracht. De uitwerking hiervan zou anders worden dan een conventionele chatbot die alleen een reeks vragen kan beantwoorden.

Vanuit de studie en mijn persoonlijke interesse blijft het belangrijk om de opdracht praktisch te houden. Het theoretisch onderzoek doen is onderdeel van het afstuderen, maar om juist de gevonden theorie om te zetten tot een prototype zorgde ervoor dat ik voor deze opdracht heb gekozen.

Inhoud

Voorwoord	2
Inleiding.....	6
Organisatiebeschrijving	7
De huidige situatie.....	7
De opdracht.....	7
Probleemstelling.....	8
1. Sprint 1 14 februari tot en met 28 februari.....	9
1.1 Frontend	9
1.2 Natural Language Understanding Module	9
1.2.1 Dialogflow.....	10
1.2.2 Rasa NLU.....	10
1.2.3 Conclusie	10
1.3 Fulfillment Backend.....	11
1.4 Externe databronnen	11
1.4.1 Interview Peter-Paul.....	11
1.5 Werkwijze planning	11
1.6 Retrospective.....	12
2. Sprint 2 28 februari tot en met 14 maart.....	13
2.1 Software architectuur overzicht.....	13
2.2 NS.....	14
2.3 Dialogflow.....	14
2.4 Externe databronnen:	15
2.5 Facilitator	15
2.6 Interview met S H	15
2.7 Retrospective.....	16
3. Sprint 3 14 maart tot en met 28 maart	17
3.1 Chatbot input en visualisatie.....	17
3.2 Nieuwe functionaliteit.....	17
3.3 Vernieuwde software architectuur	18
3.4 Retrospective.....	20
4. Sprint 4 28 maart tot en met 11 april.....	21
4.1 Implementeren NS-vertrektijden	21
4.2 Opzetten van test casussen.....	22
4.2.1 NS	22

4.2.2 Koffie/thee bestelling afhandeling.....	23
4.3 Retrospective.....	24
5. Sprint 5 11 april tot en met 25 april	25
5.1 Opzetten Dialoog testen.....	25
5.2 Bestelling overzicht voor de kiosk	25
5.3 Frontend optimalisatie	25
5.4 Dialogflow uitbreidingen	26
5.5 Retrospective.....	26
6. Sprint 6 25 april tot en met 9 mei	27
6.1 Audio input chatbot.....	27
6.1.1 Dialogflow audiobestand.....	27
6.1.2 Web Speech API	27
6.1.3 Conclusie	27
6.2 Audio output chatbot	27
6.3 Retrospective.....	27
7. Sprint 7 9 mei tot en met 23 mei	28
7.1 Deployment.....	28
7.1.1 Meteor deployment	28
7.1.2 Python deployment.....	28
7.1.3 Fysieke deployment prototype	28
7.2 Retrospective.....	28
Evaluatie Aanpak	29
Keuze van functionaliteiten.....	29
Keuze tools	29
Ontwerpen software architectuur	29
Dialoogtesten	30
Evaluatie Opgeleverde producten.....	31
Chatbot interface	31
Chatbot Dialogflow.....	31
Toekomstvisie.....	31
Evaluatie Competenties	32
A1. Analyseren probleemdomein & opstellen probleemstelling.....	32
Gc. Kritisch, onderzoekend en methodisch werken.....	32
Gf. Leren leren.....	32
C6. Ontwerpen software	33

C11. Ontwerpen van human computer interaction	33
D15. Testen.....	33
Ge. Innovatief en creatief werken.....	33
Bibliografie	34
Lijst met afkortingen	35
Bijlage 1 – Afstudeerplan	36
Bijlage 2 – Plan van aanpak	40
Bijlage 3 – Testplan.....	46
Bijlage 4 – Test bevindingen.....	48

Inleiding

De afdeling binnen Capgemini waar ik mijn opdracht heb kunnen uitvoeren is de Applied Innovation Exchange[1] (AIE). Deze afdeling is de thuisbasis voor bijna alle stagiaires die stage lopen bij Capgemini en onderzoek doen op het gebied van nieuwe technologieën. Maar ook voor werknemers van Capgemini die tussen twee projecten in zitten en zelf graag onderzoek willen doen of zich willen specialiseren in een nieuwe technologie. De AIE zal ook de afdeling zijn waarbinnen ik om hulp of feedback kan vragen over mijn chatbot. Binnen de AIE is Peter-Paul Tonen mijn aanspreekpunt en opdrachtgever.

De oorsprong van de stageopdracht komt voort uit de wens van de AIE om de ruimte waar de receptie van Capgemini zich bevindt te vernieuwen. De huidige ruimte bevat weinig technologische innovaties en hier zien zij graag verbetering in. Een van de ideeën die de AIE graag gerealiseerd ziet worden is de ontwikkeling van een conversatie chatbot die bezoekers kan verwelkomen bij Capgemini. Deze chatbot zal een aantal voorgeprogrammeerde onderwerpen hebben die zij kunnen bespreken met de bezoeker, zoals de vertrektijden van de NS of informatie over Capgemini.

Het chatbot prototype zou als demo ingezet worden om klanten geïnteresseerd te krijgen in deze nieuwe technologie. Het project zal dus niet direct in opdracht van klanten van Capgemini zijn, maar zou wel kunnen functioneren als voorbeeld. Met dit prototype kunnen de mogelijkheden van een conversatie chatbot worden gedemonstreerd. Binnen de AIE is veel kennis en ervaring aanwezig. Er kan makkelijk worden samengewerkt of feedback worden gevraagd aan andere collega's.

Momenteel kan er onderscheid gemaakt worden tussen twee type chatbots, namelijk de flowchart chatbots en een conversatie chatbot.

Hierbij wordt de flowchart chatbot vooral ingezet bij verkoop van producten of voor het geven van meer informatie. De conversatie chatbot wordt vooral ingezet voor bijvoorbeeld trivia spellen, apps om meningen te verzamelen. Het grootste verschil tussen de twee opties is het doel van de chatbot. Bij een flowchart probeert de chatbot zo direct mogelijk tot het goede eindpunt uit te komen, waarbij een conversatie chatbot ervoor zorgt dat de gebruiker zoveel mogelijk tijd spendeert met de chatbot en graag terug wil komen voor een ander gesprek.

Capgemini wil graag een combinatie van de twee types, waarbij de gebruiker het idee heeft alsof hij een conversatie heeft. Maar in realiteit is het een combinatie van een aantal kleinere flowchart chatbots. Door deze flowchart chatbots bij elkaar te nemen, vormt er al zelf een conversatie.

De hoofdfunctionaliteit van de chatbot is het stellen van vragen aan bezoekers en op basis van het antwoord verder informatie te verstrekken. Als een gesprek is gestart moet de chatbot degene zijn die initiatief neemt en de bezoeker vragen stelt. Op basis van het antwoord van de bezoeker moet hij eventueel door kunnen vragen of een informatief antwoord teruggeven. Vooraf worden een aantal gespreksonderwerpen verwerkt in de chatbot die in een willekeurige volgorde besproken worden.

Organisatiebeschrijving

Capgemini is een consultancybedrijf op het gebied van ICT. Capgemini is internationaal werkzaam en heeft alleen in Nederland al meer dan 5000 werknemers. In totaal biedt Capgemini werk aan meer dan 210.000 werknemers. De internationale omzet bedraagt 10,1 miljard euro en ze bezitten 9 kantoren in Nederland. Capgemini is een Europees bedrijf. Het hoofdkantoor is in Parijs gevestigd; maar elk land stuurt zichzelf aan.

De afdeling waar ik van februari tot juni kom te werken, is gericht op innovatie en heet: Applied Innovation Exchange. Er wordt hier niet direct voor de klant gewerkt, maar er wordt vooral onderzoek gedaan naar opkomende technologieën en mogelijke toepassingen hiervan. De AIE is wereldwijd beschikbaar als innovatiecentra voor toegepaste innovatie. Met deze bevindingen gaat Capgemini kijken hoe zij deze opkomende technologieën weer kunnen aanbieden aan hun huidige klanten. Binnen deze afdeling werken naast de reguliere werknemers, ongeveer 30 andere stagiaires die onderzoek doen of ook bezig zijn met het ontwikkelen van prototypes in dezelfde stageperiode.

De huidige situatie

Moment wordt er binnen Capgemini nog geen gebruik gemaakt van chatbots. Bezoekers die bij Capgemini binnenkomen, melden zich aan via de baliemedewerker van de receptie. Hierna kunnen zij plaats nemen om te wachten op hun afspraak. Dit proces maakt nog geen gebruik van nieuwe technologie, en hier wil Capgemini een verschil in gaan maken. Naast het idee om een chatbot te ontwikkelen die een gesprek kan aangaan met de bezoeker, is Capgemini ook op zoek naar andere vernieuwende technologieën waar de bezoeker gebruik van kan maken tijdens zijn wachtperiode bij Capgemini. Andere stagiaires werken hier tijdens hun stageopdrachten aan.

De opdracht

Mijn stageopdracht omvat het onderzoeken en ontwikkelen van een Nederlandse prototype conversatie chatbot. Vooraf is nog niet vastgesteld welke functionaliteiten de chatbot zou moeten hebben. Ook is nog niet besloten welke onderwerpen er besproken kunnen worden met de bezoeker. Als het prototype aan het einde van de stageopdracht inzetbaar is, zal de chatbot beschikbaar worden gesteld voor bezoekers van Capgemini. De bezoekers van Capgemini zijn dan ook de primaire doelgroep van de chatbot die ontwikkeld gaat worden. De functionaliteit van de chatbot is dan ook volledig op de bezoeker gericht. Een medewerker van Capgemini zal de chatbot alleen kunnen gebruiken om een idee te krijgen wat mogelijk is met deze technologie. Er is voor een Nederlandse chatbot gekozen, omdat het grootste deel van de bezoekers bij Capgemini Nederland ook uit Nederland komen.

In de eerste fase van het ontwerpen van de chatbot zijn verschillende functionaliteiten bedacht die voor de chatbot ontwikkeld zouden kunnen worden. Deze functionaliteiten zijn vooral gericht op het geven van informatie over verschillende onderwerpen binnen Capgemini of onderwerpen die van nut zouden kunnen zijn voor de bezoeker voor, tijdens of na een bezoek; bijvoorbeeld informatie over de te bezoeken medewerker, het koppelen van de weersverwachting of NS-vertrektijden waren de eerste ideeën voor de chatbot. Met behulp van een aantal vragen zou de chatbot informatie kunnen geven die op dat moment voor de bezoeker nuttig is. Verder is het belangrijk dat het niet een simpel informatiepunt zal worden. Het is bijvoorbeeld de bedoeling dat de bezoeker opgeeft dat hij na zijn afspraak in de richting van Den Haag gaat met de trein. Er zullen dan geen vertrektijden worden gegeven in de andere richting.

Probleemstelling

Capgemini is een bedrijf die zich vooral bezighoudt met het ontwikkelen van softwaresystemen. Zij streven ernaar om indien mogelijk de nieuwste innovaties toe te passen in hun projecten en de klant hiervan op de hoogte te houden. Ondanks alle innovatieve projecten bij klanten loopt Capgemini op het hoofdkantoor niet voorop als het om nieuwe technologieën gaat. De AIE heeft hierom de opdracht gekregen van Capgemini Nederland om te werken aan producten die het Capgemini kantoor innoveren. Dit kan in de vorm van een applicatie zijn of een andere technologische innovatie. Een onderdeel van deze missie was het innoveren van een bezoek van een klant aan het Capgemini kantoor in Utrecht Leidsche Rijn. Door het beschikbaar stellen van een chatbot aan de klant bij het binnenkomen krijgt hij direct een indruk van wat er mogelijk is bij Capgemini.

Er zit geen urgentie aan dit project, omdat het niet in opdracht is van een klant. Wel wil Capgemini natuurlijk zo snel mogelijk weten of deze vernieuwing in de technologische wereld haalbaar is en wat voor meerwaarde zij de klanten hiermee kunnen bieden. Vandaar dat Capgemini de urgentie ziet om een demo chatbot te ontwikkelen. Hiermee kan worden aangetoond of het een interessante toepassing is van deze vorm van technologie. De AIE heeft verder geen vereisten gesteld aan de te gebruiken ontwikkelomgeving. Wel zijn zij benieuwd naar de functionaliteit van het prototype en een korte toelichting over de gebruikte tools en een eventuele aanbeveling hierop.

1. Sprint 1 14 februari tot en met 28 februari

De eerste sprint bestond voornamelijk uit onderzoekstaken voor het project. De opdrachtgever had geen specifieke eisen gesteld aan het gebruik van bepaalde tools. Ook waren er geen randvoorwaarden over de te gebruiken programmeertalen. Dit was onder andere onderdeel van de opdracht om dit vooraf te onderzoeken. De chatbot gaat zelf uit drie onderdelen bestaan. Een frontend gedeelte, waar de gebruiker direct mee in contact komt te staan, een NLU-module die de intentie van de gebruiker gaat herkennen en een data verrijkend-backend dat alle externe koppelingen moet regelen die nodig zijn voor een goede chatbot. Het backend is onder te verdelen in een natural language understanding tool en een data verrijkend-backend. De natural language understanding tool analyseert de input van de gebruiker. Hierna wordt er bepaald wat de intentie is van de gebruiker en wat de volgende vraag kan zijn die de gebruiker vervolgens gesteld krijgt. Het data verrijkend-backend zal gebruikt worden om externe databronnen te benaderen. Voor de chatbot zal bijvoorbeeld de treintijden opgevraagd moeten kunnen worden. Het data verrijkend-backend zal daarna de opgevraagde gegevens naar de gebruiker sturen.

1.1 Frontend

Het frontend moet als visuele ondersteuning zijn voor de chatbot. Vanuit Capgemini zijn er geen vereisten aan het visuele gedeelte gesteld. Er is meer belangstelling voor de technologie achter de chatbot die het gesprek mogelijk maakt. Ik heb onderzoek gedaan naar een aantal verschillende frontend frameworks om een goede keuze te maken voor het prototype. De volgende frameworks zijn onderzocht: Vue[2], React[3], Angular[4] en Meteor[5].

Er is gekeken naar gemak van gebruik en functionaliteiten die elke tool te bieden heeft bij dit project. Hieruit is gebleken dat React en Angular een hoge learning curve hebben. Vanwege de lage prioriteit van de uitwerking van het frontend is ervoor gekozen om niet te kiezen voor een complex framework.

Het grootste verschil tussen Meteor en Vue is de integratie van een MongoDB en een publish-subscribe pattern om dataflows te reguleren. Vue kan alleen gebruikt worden om de chatbot te visualiseren en er zal nog een extra backend opgezet moeten worden voor het opslaan van gegevens.

Door te kiezen voor Meteor zal er meer tijd over zijn voor het uitwerken van de functionaliteiten van de chatbot. Omdat dit ook de prioriteit is van de afstudeerstage is hier dan ook voor gekozen.

1.2 Natural Language Understanding Module

De taak van de Natural Language Understanding (NLU) module is het herkennen van een intentie en het filteren van entiteiten. De intentie van een zin wordt geclassificeerd door het model op de vooraf vastgestelde verwachte intenties. Deze vooraf vastgestelde intenties worden vooraf opgesteld door een persoon en worden getraind met voorbeeldzinnen. Omdat het nooit mogelijk is om oneindig veel voorbeeldzinnen te geven, wordt er met behulp van Machine Learning een model getraind. Dit model berekent voor elke intentie de kans dat een zin overeenkomt met de trainingszinnen. De intentie die de hoogste score heeft, wordt vervolgens aan de zin gekoppeld. Na het bepalen van de intentie is het belangrijk om de relevante informatie uit de zin te filteren. Het herkennen van deze informatie wordt met behulp van entiteit herkenning gedaan. Een entiteit is een algemene variabele die aan woorden wordt gekoppeld. Bekende voorbeelden van entiteiten zijn: achternaam, tijdstip, locatie, of bedrag. Binnen de chatbot die tijdens deze stage ontwikkeld zal worden, kan de manier van transportatie bijvoorbeeld een entiteit zijn. Deze entiteiten worden dan exact of met behulp van machine learning gezocht in de opgegeven zin.

In het volgende onderdeel wordt voor beide onderzochte NLU-modules gekeken naar de accuraatheid en het gemak waarmee de intenties en entiteiten werken. Daarnaast wordt er gekeken welke functies de tools nog aanbieden naast deze basisfuncties. Er is in het vooronderzoek gekeken naar Dialogflow van Google[6] en naar Rasa NLU[7]. Beide modules bieden gratis opties die gebruikt kunnen worden voor dit project. Er is bij beide tools gekeken hoe snel, gemakkelijk en accuraat mogelijk het was om een eerste implementatie te ontwikkelen waarin de bezoeker gevraagd wordt met wie en wanneer ze een afspraak hebben. Na een korte toelichting per tool heb ik op deze punten mijn ervaring beschreven om uiteindelijk een keuze te maken met welke tool er gewerkt zal worden bij het project.

1.2.1 Dialogflow

Dialogflow is een online tool van Google. De module is te benaderen met behulp van een API. De tool kan zinnen omzetten naar een intent analyse en het herkennen van entiteiten. De tool gebruikt voorbeeldzinnen om een Machine learning model te trainen. Dialogflow is closed-source en geeft de gebruiker ook niet de mogelijkheid om verschillende machine learning algoritmes te testen. Na het invoeren van de voorbeeldzinnen wordt het model direct getraind en opgeslagen. Deze tool wordt gehost door Google en is alleen te benaderen via de API en de web-interface. Dialogflow biedt verder een aantal standaard integraties met externe tools zoals de Google Assistent en Skype of Facebook messenger. Dit zal voor mijn project niet van pas komen, maar het is zeker voor bedrijven die hun chatbot voor zoveel mogelijk klanten beschikbaar wilt stellen, een goede functionaliteit.

1.2.2 Rasa NLU

Rasa NLU is een python package dat een model traint met behulp van een vooraf gedefinieerd bestandsformaat (JSON). Dit model kan na het trainen worden geladen en kan de intenties classificeren. Ook bieden zij de mogelijkheid met externe packages samen te werken om entiteiten te herkennen. Rasa NLU heeft een eigen formaat ontwikkeld waarin het model te trainen is met voorbeeldzinnen. Rasa NLU heeft verder geen gebruikersvriendelijke interface en zal ook op een eigen server gehost moeten worden. Wel is Rasa NLU open-source en moet de gebruiker zelf ook kiezen welke machine learning features en model hij wil gebruiken voor de chatbot. Dit heeft als voordeel dat er meer vrijheid is bij de keuze. Wel zal het meer tijd kosten, omdat er getest moet worden welk model het beste presteert bij de opgegeven voorbeeldzinnen.

1.2.3 Conclusie

Na beide tools getest te hebben op het herkennen van intenties en entiteiten is gebleken dat Dialogflow beter presteert. Bij Dialogflow werd de goede intentie al bij minder dan 15 zinnen met hoge zekerheid (95%) herkend. Bij Rasa NLU is de accuraatheid nog niet hoog bij 20 zinnen (60%). Het zou kunnen dat er bij Rasa NLU gebruik is gemaakt van een verkeerde soort features en machine learning algoritme. Als dit het probleem zou zijn, geeft dit ook aan dat het niet makkelijk is om de goede algoritmes te vinden voor het trainen van de chatbot.

Een voordeel van Rasa NLU is de offline functionaliteit. Deze is voor de Capgemini chatbot niet nodig, omdat deze toch in verbinding met het internet moet staan om API-connecties te maken.

Waar Dialogflow vooral in uitblinkt is de voorgeprogrammeerde mogelijkheid om entiteiten te herkennen. In het herkennen van een voornaam moest Rasa NLU een voor gedefinieerde lijst met alle Nederlandse namen hebben om tot een redelijke accuraatheid te komen tot het labelen van voornamen. Bij Google Dialogflow was deze entiteit al beschikbaar. Hierdoor is het enkel nodig om in een voorbeeldzin een voornaam te definiëren waardoor in alle nieuwe zinnen de voornamen correct worden gelabeld. In hetzelfde geval was er bij Dialogflow al een entiteit beschikbaar die de gesproken tijd omzet naar een gegeneraliseerde tijd formaat. Deze functionaliteiten moeten extern gezocht

worden om te gebruiken bij Rasa NLU. Hierbij moet er ook altijd opgelet worden of de functionaliteit beschikbaar is in de gewenste taal.

Dialogflow biedt ook de mogelijkheid om een gesprek te ontwerpen met behulp van follow-up intenties. Er kan worden verwacht wat er hierna gevraagd of beantwoord zal gaan worden. Dit zorgt ervoor dat conversaties makkelijker ontworpen kunnen worden.

Op basis van deze bevindingen is er uiteindelijk gekozen voor Dialogflow. Rasa NLU biedt daarnaast de mogelijkheid om een Dialogflow agent om te zetten naar Rasa NLU. Dit zou als back-up gebruikt kunnen worden als Dialogflow uiteindelijk toch niet geschikt is.

1.3 Fulfillment Backend

Het herkennen van de intentie van een gebruiker is de eerste stap van een chatbot, maar het waardevol kunnen antwoorden met gedetailleerde informatie moet door een ander systeem worden uitgevoerd. De hierboven gekozen Google Dialogflow maakt dit mogelijk via het fulfillment systeem. Per intentie kan gekozen worden om vooraf eerst een post-request te maken naar een backend. Dit backend systeem kan dan een custom response maken. Via dit systeem kunnen externe data sources gebruikt worden om de antwoorden van de chatbot te verrijken. Uit eigen ervaring is er gekeken naar de mogelijkheden van een python backend. Na enig experimenteren en in combinatie met Ngrok is het gelukt om een custom response terug te sturen naar Dialogflow. Ngrok is een web tunneling tool die de mogelijkheid biedt om webhooks te testen vanuit een lokale server. Door mijn ervaring met Python is ervoor gekozen om met deze ontwikkeltaal verder te werken.

1.4 Externe databronnen

In de eerste sprint is er gekeken naar externe databronnen die aan de chatbot gekoppeld zouden kunnen worden. Er is meteen ook gekeken of het mogelijk is een kleine demo te maken. Een veelvoorkomend gespreksonderwerp in Nederland is: het weer. Met behulp van een weers-API zou dit gebruikt kunnen worden. Met behulp van openweathermap [8] is het gelukt om de weersomstandigheden te verwerken in een zin.

Daarnaast is er gezocht naar waardevollere koppelingen die voor bezoekers nuttig zouden kunnen zijn. Zo zou een koppeling met een medewerker systeem en de NS/OV een meerwaarde kunnen bieden. Met de NS-koppeling zou het bijvoorbeeld mogelijk zijn om bezoekers te kunnen vertellen of er vertraging wordt verwacht.

1.4.1 Interview Peter-Paul

Op donderdag 28 februari is er tijdens een gesprek met de stagebegeleider over mogelijke koppelingen gesproken. Tijdens dit gesprek hebben we het ook gehad over een mogelijke koppeling met het interne systeem van Capgemini voor het aanmelden van gasten. Er zal contact opnemen met de beheerder van dit systeem om een mogelijkheid te bespreken over deze koppeling tussen systeem en chatbot. Daarnaast is de koppeling met de NS besproken en is er een mailadres gedeeld met een vast contactpersoon van de NS voor Capgemini.

1.5 Werkwijze planning

Er is gekozen om in sprints van twee weken te werken. Dit komt overeen met de algemene werkwijze binnen Capgemini. Daarnaast staat er alvast voor de gehele stage gepland dat om de donderdag er een pitch gegeven moet worden. Tijdens deze sprint wordt de sprint in het kort besproken. Het gaat hier om wat er goed en minder goed ging en wat er voor de volgende twee weken op de planning staat. Dit is een mooi moment om de sprint te starten en af te sluiten. De sprint planning wordt wel

individueel uitgevoerd. Er is vooraf een zo compleet mogelijke backlog gemaakt met prioriteiten. Vanuit deze backlog wordt er een schatting gemaakt hoeveel taken er in de volgende sprint voltooid kunnen worden. Er is afgesproken om met een kleine groep stagiaires een dagelijkse scrum uit te voeren. Er zal gekeken worden na twee weken hoeveel toegevoegde waarde dit heeft tot het algemene proces. Er moet hier rekening worden gehouden met het feit dat elke stagiair een compleet andere stageopdracht heeft en er verder geen echte rollen zijn binnen de groep.

1.6 Retrospective

Sprint 1 is volgens planning verlopen. Alle benodigde onderwerpen zijn onderzocht en er is een keuze gemaakt over de tools die gebruikt zullen worden in het project. Dankzij het goede verloop van de eerste sprint is het mogelijk om tijdens de volgende sprint te beginnen met het ontwikkelen van een eerste versie van de chatbot. In sprint 2 zal er onderzoek moeten worden gedaan naar de verdere mogelijkheden van de gekozen tools. Ook zal er een korte lijst met vereisten moeten worden opgezet van de eerste versie van de chatbot. Met behulp van deze lijst kan de tool worden ontwikkeld en kan achteraf makkelijker worden geëvalueerd of het product voldoet aan de verwachtingen. Hierna kan er een start worden gemaakt aan het ontwikkelen van de chatbot zelf.

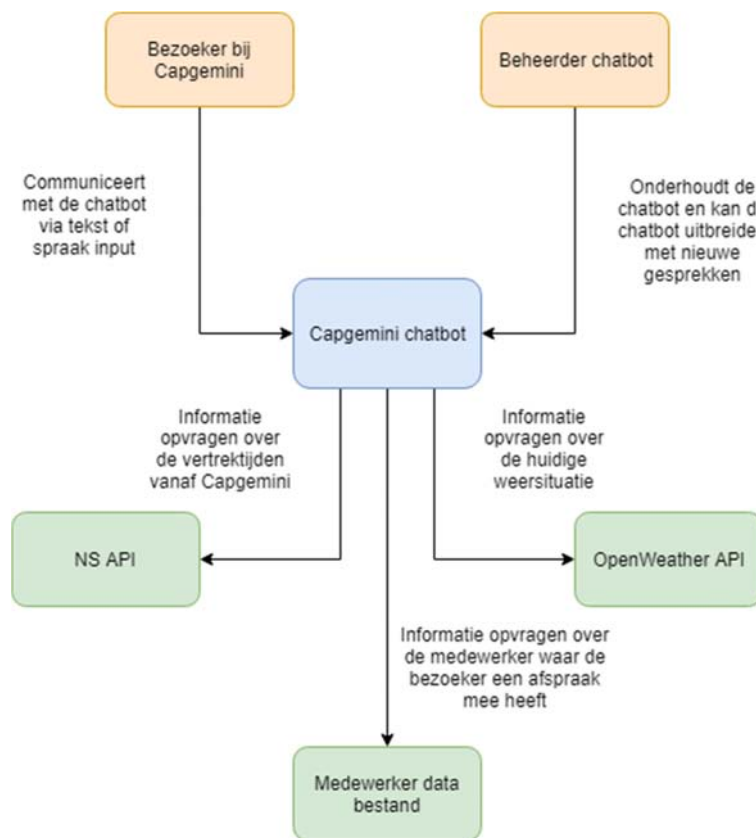
2. Sprint 2 28 februari tot en met 14 maart

De tweede sprint begint met het maken van een goede planning. Dit in tegendeel tot de algemene planning die gemaakt was voor sprint 1. De geplande taken zijn toegevoegd aan sprint 2, waarna deze sprint kan beginnen. Deze sprint bestaat grotendeels uit het onderzoeken hoe een gesprek geprogrammeerd kan worden met Dialogflow. Daarnaast moet er contact worden gezocht met de services die nodig zijn om waarde aan het gesprek toe te kunnen voegen.

2.1 Software architectuur overzicht

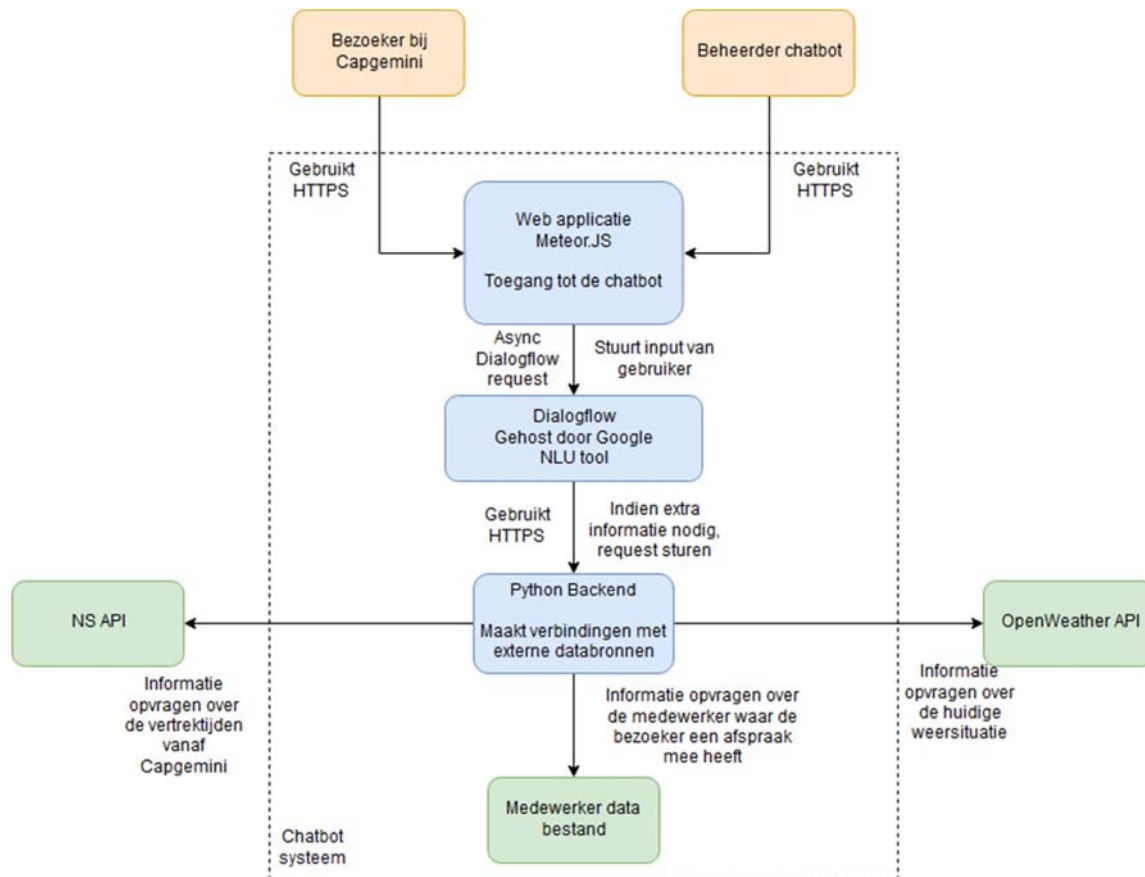
Na het bepalen van de te gebruiken tools in sprint 1, zijn er twee type overzichten gemaakt die de architectuur beschrijven van het systeem. Het system-context overzicht beschrijft het te maken systeem als een geheel. De factoren die gebruik zullen gaan maken van het systeem worden los beschreven in oranje. In dit geval zal het gaan om de bezoekers van Capgemini. De chatbot zal hen in de toekomst gaan helpen en gaat er ook vanuit dat de gebruiker een bezoeker is. Daarnaast zal er toegang gemaakt moeten worden voor de beheerder van de chatbot. De beheerder zal steeds toezien of de chatbot nog naar verwachting werkt. Ook zal de beheerder steeds kijken naar nieuwe mogelijkheden om bijvoorbeeld nieuwe onderwerpen toe te voegen.

De services die los staan van het te maken systeem staan in groen aangegeven. Deze systemen zijn in dit geval twee API's en een interne databron. De chatbot zal primair door bezoekers van Capgemini worden bestuurd. Deze bronnen worden door de chatbot benaderd als er data nodig is om de gebruiker te helpen.



Figuur 1 System Context

In figuur 2 is te zien uit welke onderdelen de chatbot zal bestaan. Binnen het aangegeven gestippelde vierkant staan alle onderdelen van de chatbot beschreven. Hierin staan de losse onderdelen die in sprint 1 zijn gekozen in het blauw aangegeven. Het medewerker databestand staat binnen het systeem, omdat dit geen extern koppeling zal worden. Daarnaast is hierin te zien hoe het Python backend alle externe data flows regelt en dit niet direct naar het frontend stuurt.



Figuur 2 Containers

2.2 NS

Via een intern contact bij Capgemini is er contact gelegd met iemand van de NS. Na enige toelichting over het gebruik van de API, is er een API-sleutel toegewezen. Naast de koppeling moet er ook nog bepaald worden op welke manier deze data gebruikt zal gaan worden.

2.3 Dialogflow

Dialogflow is tijdens de eerste sprint alleen gebruikt voor het direct beantwoorden van een vraag. In de tweede sprint staat gepland om een lopend gesprek te ontwerpen met behulp van Dialogflow. Er is voor gekozen om dit te proberen met de follow-up intent functie. Deze functie is gemaakt als er een vervolgvraag wordt gesteld na het herkennen van een intent. Er zijn al een aantal vooraf gedefinieerde follow-up intents, zoals: ja, nee, later, volgende, vorige. Deze follow-up intents bevatten een lijst met voorbeeldzinnen, zodat de gebruiker dit niet zelf hoeft in te vullen.

Een voorbeeld van het gebruik van deze functie is een bevestiging. Iemand vraagt of de chatbot een pizza wil bestellen, dan kan er als antwoord worden gevraagd of de bestelling klopt en kan er met twee

follow-up intents de functionaliteit worden toegevoegd waarbij er een 'ja' of een 'nee' wordt verwacht.

Door deze functionaliteit op dezelfde manier in te zetten na de vraag te stellen of iemand met het openbaar vervoer is gekomen of met de auto, kan er op basis van het antwoord een keuze worden gemaakt waar het gesprek vervolgens heen gaat. Bij elke follow-up intent wordt aangegeven wat verwachte antwoorden kunnen zijn en op basis van deze input traint het model zich om zo te kunnen herkennen wat een gebruiker bedoelt

2.4 Externe databronnen:

Tijdens de tweede sprint is ook gekeken naar de mogelijkheid om een geautomatiseerd systeem te maken die periodieke gegevens van de Capgemini expert pagina [9] kan ophalen.

Op deze webpagina is per expert een naam, foto, functie, startdatum en een link naar een foto opgeslagen. Als deze informatie van de site te verzamelen is, zou het gebruikt kunnen worden om de bezoeker meer informatie te geven over de persoon waar zij zo een afspraak mee hebben.

De webscraper is in het Python backend project ontwikkeld. Hierdoor kan de verzamelde data makkelijker toegevoegd worden aan de chatbot.

Bij vooronderzoek naar populaire tools voor het webscrapen via Python, zijn de volgende twee als optimaal gebleken: Requests en BeautifulSoup [10]. Requests wordt gebruikt om html op te halen via een http request. Daarnaast wordt BeautifulSoup ook gebruikt om data uit de ruwe html te filteren. Met behulp van BeautifulSoup kan er bijvoorbeeld worden gezocht naar diverse elementen en id tags die op de website gebruikt worden om een bepaalde waarde op te maken in CSS.

Na het ontwikkelen van de tool is er in overleg met de begeleider bepaald dat er ook naar het privacy aspect van het gebruik van deze data moet worden gekeken. De begeleider kon hier verder zelf weinig met zekerheid over zeggen, dus is er met iemand van de Legal afdeling binnen Capgemini contact gezocht.

2.5 Facilitator

Facilitator is de interne tool die gebruikt wordt om gasten aan te melden bij Capgemini. Omdat Facilitator een interne tool is, is er ook direct onderzoek gedaan naar het privacy aspect. Uit een telefonisch gesprek met iemand die deze tool beheert binnen Capgemini, is er besloten dat een koppeling met Facilitator niet mogelijk is. Omdat de tool sterk verouderd is, zou dit te hoge kosten met zich mee brengen. Daarnaast wordt aanbevolen om eerst in overleg te gaan met de stagebegeleider of het toegestaan is om deze informatie sowieso beschikbaar te maken via de chatbot. Dit zou ook gelden voor de informatie die beschikbaar is op de website van Capgemini. Ook al is deze informatie publiekelijk beschikbaar, zal er mogelijk opnieuw toestemming gevraagd moeten worden om deze data ook in de chatbot te mogen verwerken.

2.6 Interview met S H

S H is een werknemer binnen Capgemini die afgelopen jaar onderzoek heeft gedaan naar het ontwikkelen van conversatie chatbots met LUIS van Microsoft. Dit is de Language Understanding tool van Microsoft die op eenzelfde manier ingezet kan worden als Dialogflow. Hij heeft vooral onderzoek gedaan in samenwerking met experts op het gebied van conversatie en heeft dit uiteindelijk in een demo uitgewerkt. Deze demo was verder niet omdat ik de keuze had gemaakt om met Dialogflow te werken was hij erg benieuwd hoe dat zou verlopen. We hebben het over de verschillende functies gehad die mijn chatbot zou hebben. Aan het einde van het gesprek heeft hij aangeboden om

beschikbaar te zijn indien ik vragen zou hebben bij het ontwikkelen van mijn chatbot of om feedback te vragen als een onderdeel getest moet worden.

2.7 Retrospective

De sprint 2 is vrijwel geheel volgens plan gelopen. Een verandering ten opzichte van het plan van aanpak is het gebruik van persoonlijke gegevens in de chatbot. Vanwege de onzekerheid of deze privacygevoelige data gebruikt mag worden is er besloten om hiervan af te zien. In de volgende sprint moet er worden gezocht naar nieuwe functionaliteiten voor de chatbot, die hier geen gebruik van maken.

Tijdens de sprint is goede voortgang gemaakt met Dialogflow. Er is nu inzicht verkregen hoe een gesprek ontworpen kan worden en hoe de vervolgvragen uitgewerkt zullen gaan worden.

De backlog is geüpdatet door het verwijderen van de taken die met de personeelsgegevens te maken hebben. Deze taken zijn vervangen met onderzoekende taken naar nieuwe functionaliteiten.

3. Sprint 3 14 maart tot en met 28 maart

3.1 Chatbot input en visualisatie

Na afloop van een gesprek met Sebastiaan over de visualisatie en input van chatbots zijn er een aantal nieuwe mogelijkheden aan het licht gekomen die verder bestudeerd kunnen worden. In de huidige situatie is de visuele omgeving die voor de chatbot is ontwikkeld nog niet heel visueel toereikend. Omdat de visuele omgeving niet bij de primaire taken hoorde, is hier ook nog niet direct tijd in gestoken. Vanwege het gebruik van Google Dialogflow zou het ook goed mogelijk kunnen zijn om de Google Assistent interface te gebruiken voor de chatbot. Deze interface is beschikbaar op Google apparaten en alle apparaten met een recente versie van Android.

Een voordeel van het gebruik van de Google interface is de visuele omgeving die niet meer geprogrammeerd hoeft te worden. Omdat het prototype mogelijk ingezet kan gaan worden voor klanten van Capgemini, zou het prettig zijn wanneer de interface er visueel goed uitziet.

Een nadeel van het gebruik van deze interface zijn de beperkte mogelijkheden tot uitbreidingen aan de functionaliteit van het systeem. In een zelfontworpen systeem zou het mogelijk zijn om een functionaliteit te ontwerpen die de pagina kan herstellen voor een nieuwe gebruiker. Daarnaast wordt er vanuit de Google Assistent interface vanuit gegaan dat de gebruiker het gesprek start met een vraag aan de assistent. Dit is in het geval van het afstudeerproject niet zo en zal de chatbot starten met een vraag.

Een tweede punt waarover gediscussieerd is, is de manier van interactie met de chatbot. Het eerste prototype van de chatbot was enkel te benaderen via tekst input. Er is gekeken of het mogelijk is om dit uit te breiden met steminput. Deze functionaliteit zal de chatbot meer authenticiteit geven, omdat de gebruiker niet zijn antwoord handmatig hoeft in te voeren. Er is uiteindelijk besloten om beide mogelijkheden te implementeren. De voorkeur ligt zoals hierboven besproken bij steminput, maar vanwege de onzekerheid van geluidsoverlast in een publieke ruimte zal er ook de mogelijkheid worden gegeven om met een toetsenbord input te geven.

3.2 Nieuwe functionaliteit

Vanwege de privacyaspecten en de informerende kern van de voorbedachte functionaliteiten die de chatbot zou hebben, is er ook over een nieuwe functionaliteit gedacht. In tegenstelling tot de informerende taken, is er voor deze nieuwe functionaliteit meer in de richting van een ondersteunende taak gedacht. Na een brainstormsessie is er besloten om de chatbot als hoofdfunctie de mogelijkheid om een bestelling op te nemen.

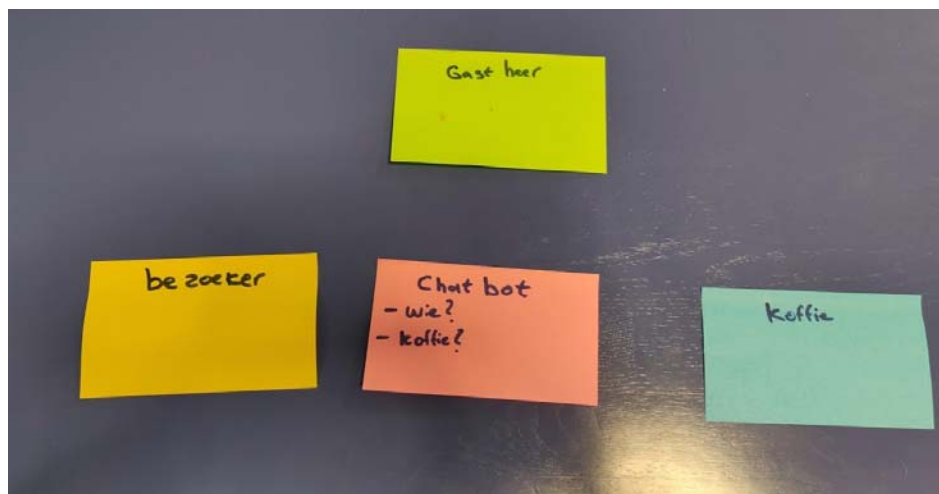
In de huidige situatie bij Capgemini is het mogelijk voor bezoekers om iets te drinken te bestellen bij de kiosk van Capgemini. Een nadeel van deze service is dat de kiosk niet direct toegankelijk is voor de bezoeker. De bezoeker zal eerst moeten worden opgehaald door een werknemer van Capgemini, waarna hij toegang krijgt tot het bedrijf en iets te drinken kan bestellen.

De uitwerking van de bestel-functie zal bestaan uit een gespreksvorm tussen de chatbot en de gebruiker, die een drankje kan bestellen. Het gesprek zal eenzelfde vorm krijgen als een gesprek bij een koffiebar waar iets te drinken kan worden besteld om mee te nemen. De informatie die nodig is voor het doorgeven van een drankje is: de naam van de bezoeker en de bestelling. Na het herkennen van deze entiteiten wordt de bestelling en naam in een database opgeslagen. Deze database wordt daarna gevisualiseerd bij de kiosk, zodat een medewerker van de kiosk het drankje klaar kan maken. Nadat iemand het gesprek met de chatbot heeft afgerond, wordt de bezoeker zoals gewoonlijk

opgehaald door een werknemer van Capgemini. Als ze samen bij de kiosk aankomen hoeft er alleen maar gezegd: “Ik wil graag de koffie voor [naam] meenemen,” en kan deze worden klaargemaakt.

Een nice-to-have requirement voor deze functionaliteit is een extra tussenstap voor de werknemer van Capgemini die het tijdstip doorgeeft wanneer hij de bezoeker gaat ophalen. Op het moment dat iemand namelijk doorgeeft dat hij de persoon nu gaat ophalen, kan ervan uit worden gegaan dat ze binnen een aantal minuten bij de kiosk zullen zijn. Met behulp van deze implementatie zou er niet meer gewacht te hoeven worden op de klaargemaakte drank.

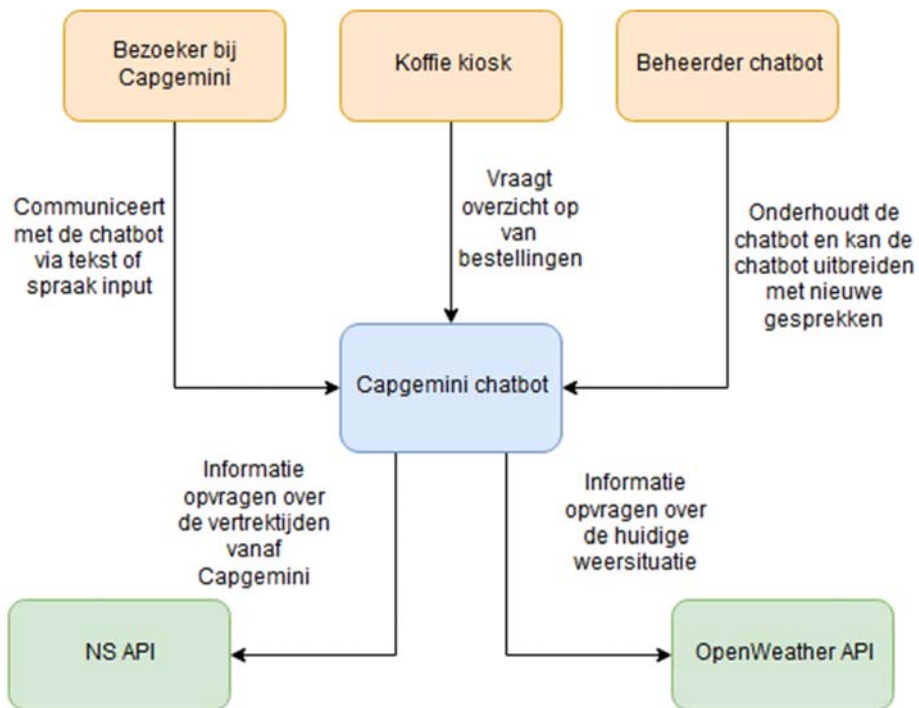
Deze functionaliteit moet met een extra bevestiging worden uitgevoerd, omdat het mogelijk is dat de bezoeker een kwartier te vroeg is en alvast het gesprek aangaat met de chatbot. Als de koffie dan al direct wordt gemaakt, zou deze koud kunnen zijn wanneer de werknemer en bezoeker daadwerkelijk bij de kiosk aan komen.



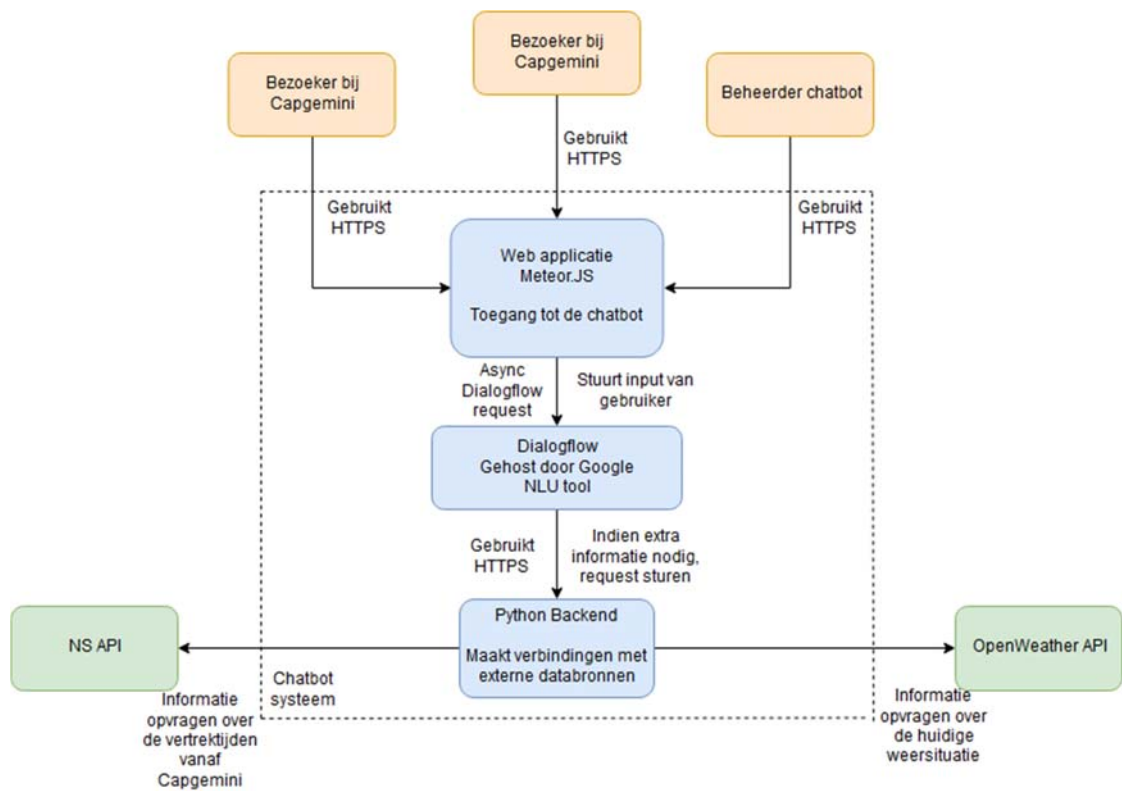
Figuur 3 eerste uitwerking koffie/thee functionaliteit

3.3 Vernieuwde software architectuur

Door het veranderen van de functionaliteit van de chatbot zal ook de architectuur veranderen. Hieronder zijn net als in sprint 2 de vernieuwde overzichten toegelicht met focus op de veranderingen. De enige verandering in het context overzicht is de koffie kiosk als externe partij. De chatbot stuurt de gemaakte bestellingen door naar een interface bij de kiosk, zodat voor hen zichtbaar wordt wat er besteld is. Vanwege de beperkte tijd is ervoor gekozen om niet nog een los onderdeel te ontwikkelen voor het overzicht dat de kiosk gaat gebruiken. Deze interface zal in hetzelfde project worden ontwikkeld als de chatbot. Vanwege de integratie van Mongo DB in Meteor is het ook mogelijk om vanuit de chatbot direct een bestelling toe te voegen in de database. Hierdoor hoeven er geen extra koppelingen gemaakt te worden tussen de systemen. In figuur 5 is te zien dat het medewerker databestand geen onderdeel meer uitmaakt van het systeem. Daarnaast is er wel een extra gebruiker bij gekomen die een eigen interface in de webapplicatie zal krijgen.



Figuur 4 System Context



Figuur 5 Containers

3.4 Retrospective

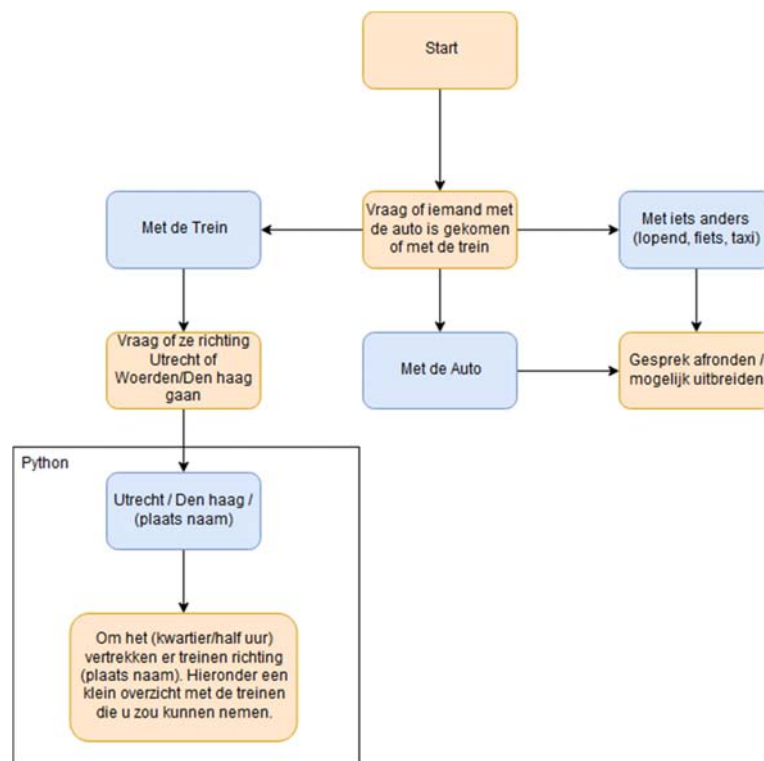
Ondanks de verandering in de functionaliteit van de chatbot in de vorige sprint is er deze week direct gezocht naar nieuwe functionaliteiten. Na een brainstormsessie met de technische begeleider is er een goed idee naar boven gekomen. Dit is later ook besproken met de algemeen begeleider en die vond het ook een goed alternatief. Verder is er deze sprint extra tijd gestoken in het updaten van de architectuur van het systeem. De lay-outs zijn geüpdatet en gedocumenteerd. Nu deze functionaliteit vast staat, kan dit verder in detail worden uitgewerkt in de volgende sprint. Ook zal er tijd zijn voor het uitwerken van de NS-functionaliteit. Verder zal er een start worden gemaakt met het schrijven van een testplan en het opzetten van een testproject. Er zal verder worden gekeken naar verschillende manieren van testen en hoe die binnen dit systeem verwerkt kunnen worden.

4. Sprint 4 28 maart tot en met 11 april

4.1 Implementeren NS-vertrektijden

In de vorige sprint is er toegang gekregen tot de NS API. Het doel in deze sprint is het implementeren van een basisfunctie uit deze API. Het idee was om de gebruiker te vragen of zij na hun gesprek met de auto of trein gaan. Als ze met de trein reizen, krijgen zij de vervolgvraag of ze richting Den Haag of Utrecht gaan. Hierna moet er bepaald worden wat de gemiddelde duur is van een gesprek, zodat er alvast kan worden gekeken of de treinen voor de terugreis op dat moment ook nog rijden. In eerste instantie is ervan uitgegaan dat een gesprek gemiddeld anderhalf uur duurt. Wanneer iemand dus een gesprek heeft met de chatbot om 5 voor elf, krijgt hij treintijden teruggegeven van de treinen van 5 voor half 1. Op deze manier kan de gebruiker zien hoe vaak zijn trein ongeveer gaat en of er al verwachte vertragingen zijn waar de bezoeker rekening mee kan houden. Deze functionaliteit zal deels in Python en deels in Dialogflow worden ontwikkeld. In eerste instantie wordt het gesprek geleid door Dialogflow die de gebruiker de vragen zal stellen en uiteindelijk aankomt bij de vraag in welke richting de bezoeker gaat reizen na het bezoek aan Capgemini. Als deze intentie wordt gevonden in het antwoord van de gebruiker zal er een webrequest naar Python worden gestuurd met de gevonden richting. Het Python backend zal dan het benodigde API webrequest uitvoeren en met behulp van het antwoord een goed geformuleerd antwoord teruggeven. Dit wordt dan weer teruggestuurd naar Dialogflow en zal dan worden weergegeven aan de gebruiker via de Meteor webapplicatie.

Dit proces is verder uitgewerkt in figuur 6. Al deze stappen worden verwerkt in Dialogflow. Deze eerste implementatie gaat ervan uit dat de gebruiker een helder antwoord geeft en heeft daardoor nog niet veel extra antwoorden. Dit is iets wat later makkelijk uitgebreid kan worden nadat de eerste versie goed is geïmplementeerd



Figuur 4 Vervoer conversatie flowchart

4.2 Opzetten van test casussen

Een tweede focus in de vierde sprint is het opzetten van een algemene test omgeving. Deze omgeving moet zekerheid bieden over de werking van de chatbot. Vooraf is een testplan (bijlage 3) opgezet met een beschrijving van alle functionaliteiten en hoe deze getest zouden moeten worden. Hieronder zijn de twee grootste onderdelen beschreven die getest zullen moeten worden.

Het systeem is onder te verdelen in twee systemen: Dialogflow en de eigen logica in het systeem. Beide systemen zullen getest moeten worden om tot een goed eindproduct te komen. Dialogflow biedt zelf geen oplossingen voor het testen van hun software en dialogen. De uitwerking die is gekozen voor het testen van Dialogflow is een userinterface met vooraf opgestelde gesprekken en de verwachte intenties die herkend zouden moeten worden per input. In de applicatie die online komt te staan, kan de beheerder van de chatbot de tests met een knop draaien en zo inzien of de test geslaagd is of dat een bepaald gesprek verkeerd wordt uitgevoerd.

Deze testomgeving kan niet gezien worden als unittests, omdat ze alleen handmatig getest kunnen worden en niet bij het bouwen van het project worden gedraaid. Wel zal dit onderdeel van het project de eigenaar een gerustheid kunnen geven dat de chatbot werkt zoals hij behoort te werken. De eigenaar krijgt liever via de web interface te horen dat een bepaald gesprek stuk loopt en verkeerd herkend wordt dan dat een bezoeker een opmerking over het niet fungeren van de chatbot maakt.

Het tweede gedeelte van het project bestaat voor de helft uit Python logica code en voor de andere helft uit Javascript dat het frontend functionaliteit geeft. Beide delen kunnen met behulp van een aantal unittests worden getest. De functionaliteiten die getest moeten worden zijn met behulp van use-cases beschreven. Per use-case worden de verschillende onderdelen in code beschreven en toegelicht op welke manier deze code is getest.

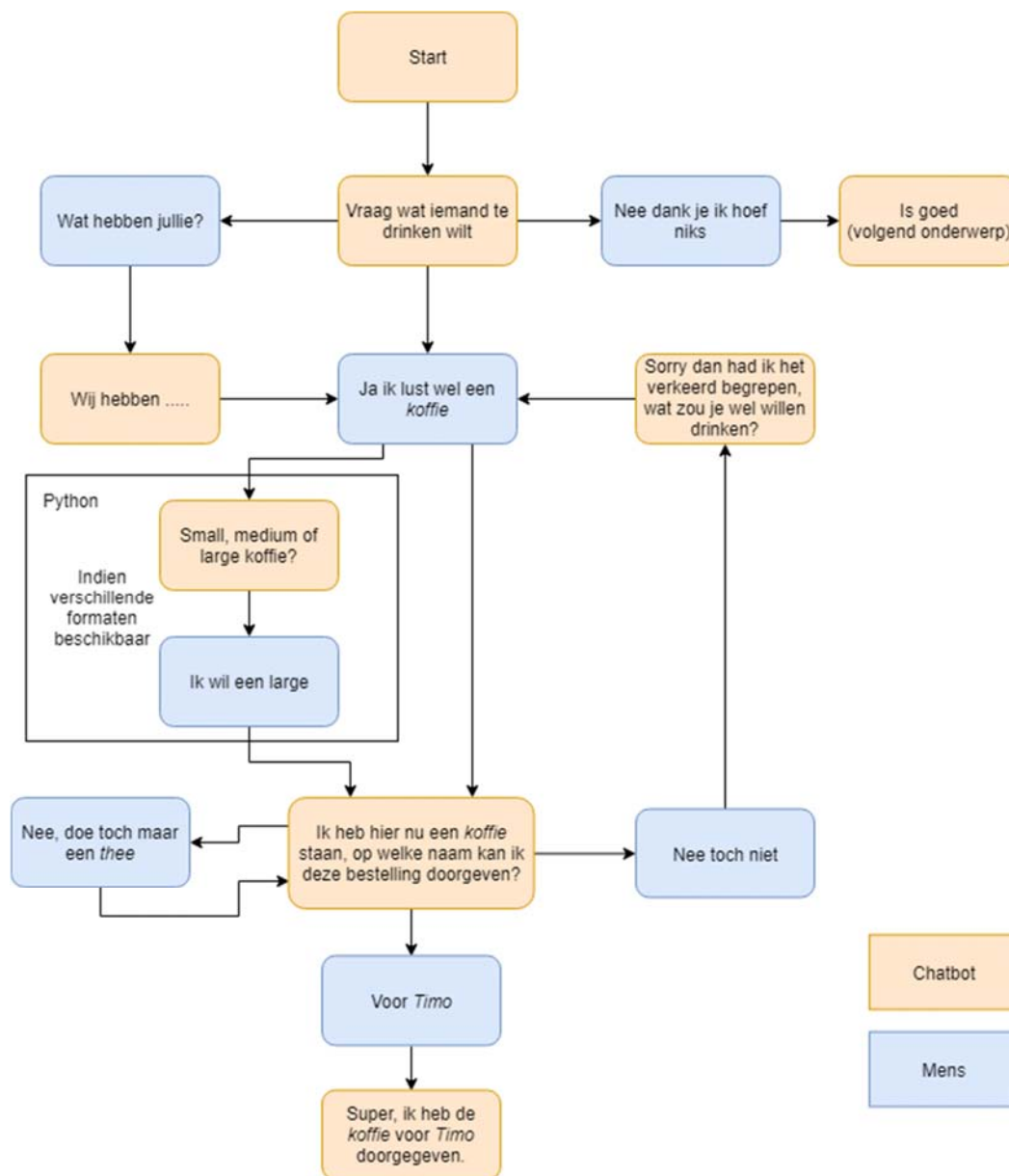
4.2.1 NS

De NS-koppeling is in figuur 6 terug te vinden. In tegenstelling tot de koffie/thee bestelling hoeft hier alleen bij het Python gedeelte een aantal unittests opgezet te worden. Omdat hier gewerkt wordt met een externe API, moet de resultaten van de API worden opgeslagen om de logica van de code te testen. Vanaf Utrecht Leidsche Rijn gaan alleen treinen in de richting van Utrecht Centraal of Woerden, deze gaan om het kwartier. Met deze kennis kan dus een fallback worden gemaakt als er geen locatie wordt herkend, die in ieder geval deze informatie met de gebruiker deelt. Hiervoor zal dus een unittest gemaakt moeten worden, waarin een locatie wordt meegegeven wanneer deze niet wordt herkend door de NS API. Het kan ook zo zijn dat Dialogflow geen locatie entiteit herkent. Wanneer dit gebeurt zal ook gebruik worden gemaakt van de fallback functionaliteit.

Als er wel een locatie wordt herkend zal dit naar de NS API worden gestuurd met de vraag wanneer de eerstvolgende treinen vertrekken vanaf Utrecht Leidsche Rijn naar de locatie. Als er resultaten zijn voor deze query wordt er een resultaat geformuleerd voor de gebruiker. Dit zal voor de verschillende locaties worden getest.

4.2.2 Koffie/thee bestelling afhandeling

In figuur is het hele bestelproces van de koffie/thee omschreven. De tekst in de oranje rechthoeken komt niet overeen met de werkelijke tekst die de chatbot zal gebruiken. In de blauwe rechthoeken valt te zien welke 'happy paths' er zijn geprogrammeerd. Het gaat hier om de gesprekken die precies zoals gepland gaan tussen de gebruiker en de chatbot. Daarnaast is hier goed te zien welke intenties worden herkend. Elke blauwe rechthoek is een intenties die herkend moet worden. Per intentie zijn er minstens 15 voorbeeldzinnen toegevoegd, zodat het model een goede accuraatheid kan krijgen. Deze functionaliteiten worden allemaal getest door het dialoog test-tool in het Meteor frontend. Het Python gedeelte waarin wordt gecontroleerd of er verschillende formaten van de bestelde drank bestaan kan getest worden met unittests. Daarnaast wordt in de laatste stap een bestelling toegevoegd in het frontend gedeelte van de applicatie.



Figuur 5 Koffie bestel proces

4.3 Retrospective

Het opzetten van de unit tests bij de code heeft in deze sprint erg veel tijd gekost. Dit kwam deels doordat er meer unit tests nodig waren dan verwacht. De inschatting was gebaseerd op een ouder design waar minder integraties aanwezig waren. Ook is er extra tijd besteed aan het ontwerpen van de dialoog testomgeving. Dit ontwerp kan in de volgende sprint verder worden ontwikkeld. Beide begeleiders van Capgemini zijn positief over de voortgang van het project. Ze zijn beiden erg geïnteresseerd in de implementatie van de conversatie tester en zijn benieuwd naar mijn bevindingen ten aanzien van het ontwikkelen van deze functionaliteit. Dialogflow biedt zelf niet zo een functionaliteit aan en kan dus niet makkelijk worden getest, behalve door handmatig de chatbot te testen. Een losse implementatie van de dialoog tester zou dan ook mogelijk als losstaand product aangeboden kunnen worden.

5. Sprint 5 11 april tot en met 25 april

5.1 Opzetten Dialoog testen

Het hoofddoel van deze sprint is het opzetten van de dialoog testing-interface. Deze functionaliteit bestaat uit een frontend gedeelte dat inzichtelijk moet maken welke tests slagen en waar het eventueel fout gaat. Daarnaast moeten de testen gestart kunnen worden vanuit het frontend en is er onderzoek gedaan naar de mogelijkheid om dezelfde tests periodiek te laten lopen. Naast het frontend gedeelte moet het daadwerkelijke testen geïmplementeerd worden. Als het om een conversatie gaat, is het noodzaak dat dit synchroon gebeurt, en moet er bij elke zin worden getest of de correcte intentie is herkend.

Voor elke test worden de resultaten opgeslagen in de Mongo database. Hierdoor is makkelijk terug te zien welke testen al langer niet getest zijn en welke niet meer slagen. Door dit overzichtelijk te maken hoeft er niet direct een ontwikkelaar beschikbaar te zijn om de werking van de chatbot te testen. Ook kunnen er vanuit de interface nieuwe test cases opgesteld worden. Wel is er hiervoor gekozen om de intenties vast te zetten, omdat de koppeling met Dialogflow te veel tijd zou kosten voor deze extra functionaliteit.

Uiteindelijk worden alle testen weergegeven in de een tabel die te zien is in figuur 8. Hierin is terug te zien welke testen geslaagd zijn en wanneer ze voor het laatst geslaagd zijn.

Aantal zinnen	Beschrijving	Zinnen	Laatst succesvolle test	Geslaagd	Test nu
4	Medium koffie voor Timo	Ik lust wat te drinken, ik wil graag koffie, Een medium, Voor Timo	24-4-2019 12:40:37	Geslaagd	Klik hier
4	Small thee voor Timo	Ik lust wat te drinken, ik wil graag thee, Een small, Voor Timo	24-4-2019 12:40:37	Gefaald	Klik hier
5	Thee, oh nee toch koffie voor Sander	Ik lust wat te drinken, ik wil graag koffie, Een medium, Oh nee toch liever thee, Voor Timo	24-4-2019 12:40:37	Geslaagd	Klik hier

Figuur 6 Overzicht van voorbeeldtesten

5.2 Bestelling overzicht voor de kiosk

Voor het opslaan van koffie/thee bestellingen is er een database schema gemaakt. Per bestelling wordt er de naam van de bezoeker, bestelde drank en het formaat opgeslagen.

In het frontend is deze data verwerkt in een tabel. De bestellingen kunnen handmatig verborgen worden, maar er is verder gekozen om alleen alle bestellingen van een dag weer te geven. In figuur 9 is te zien hoe dit overzicht eruitziet.

Type drank	Persoon naam	Formaat drank	Bestelling tijd	Verbergen
koffie	Timo	medium	18:48 Vrijdag	Verbergen
thee	monique	medium	18:46 Vrijdag	Verbergen

Figuur 7 Overzicht bestellingen

5.3 Frontend optimalisatie

Er is een overzicht gemaakt met alle frontend verbeteringen die nog moeten worden uitgevoerd. Deze bestaan deels uit het visuele onderdeel dat weinig prioriteit heeft voor de chatbot zelf. Daarnaast moet er nog stem invoer en spraak worden toegevoegd. Google Dialogflow heeft hier ondersteuning voor, dus dit zal waarschijnlijk nog te realiseren zijn binnen de stagetijd. Daarnaast is uit een gesprek met de manager van de kiosk en het restaurant binnen Capgemini nog een andere wens gekomen. Hij zou graag een overzicht willen hebben, waarin alle bestellingen overzichtelijk worden. Dit wordt ook een nice-to-have requirement voor het project. Aangezien de data al aanwezig en opgeslagen blijft, zou het mogelijk zijn om hier op een nieuwe pagina een gedetailleerd overzicht van te geven.

5.4 Dialogflow uitbreidingen

Vanuit het testen van Dialogflow bleken er een aantal problemen te zijn met gesprekken die niet volgens de 'happy path' verliepen. Een voorbeeld hiervan was bij het aanpassen van de bestelde drank. Dit werd in eerste instantie niet goed herkend door Dialogflow. Door de entiteit van een drank toe te voegen bij de cancel intentie werd dit al beter herkend. Daarnaast is er ook na het gesprek met de manager van de kiosk een nieuwe functie toegevoegd waarbij herkend kon worden wat voor formaat drank de persoon wilt. Hierbij is opnieuw gebruik gemaakt van het Python backend die de correcte vraag kon stellen na het herkennen van de drank.

5.5 Retrospective

Alle werkzaamheden zijn goed verlopen tijdens deze sprint. Ook was er duidelijk wat er nog nodig was om het project af te ronden. De meeste tijd kan nu besteed worden aan stem invoer en spraak van de chatbot. Daarnaast is het nog van belang om het start en eind proces te optimaliseren. Aan het einde van deze sprint moet de gebruiker de pagina nog verversen om een nieuw gesprek met Dialogflow te starten. Er is voor gekozen om dit met een timer en nog te specificeren triggers aan te pakken. Als een gesprek duidelijk is afgelopen, kan de tool het gesprek resetten en een nieuwe sessie ID voor Dialogflow aanmaken. Daarnaast kan het ook gebeuren dat een gebruiker wegloopt bij het apparaat en het gesprek dus nooit zal afmaken. Deze situatie zou verholpen kunnen worden, wanneer er een minuut geen input volgt. Deze taken zullen verdeeld worden over de komende sprints.

6. Sprint 6 25 april tot en met 9 mei

6.1 Audio input chatbot

Een van de kenmerken van een chatbot is vaak de stembesturing. Dit is bijvoorbeeld terug te zien bij de Google Assistent of de Amazon Alexa. Om deze functionaliteit mogelijk te maken voor de chatbot is er onderzoek gedaan naar de mogelijkheden. Er is eerst gekeken naar de mogelijkheden om deze functionaliteit toe te voegen via Dialogflow. Vanuit de Dialogflow library is er namelijk een functie waarbij er een intentie herkend kan worden uit een audiobestand. Een tweede mogelijkheid was een uitwerking waarbij client side audio wordt omgezet naar tekst. Nadat de audio omgezet is wordt dit als tekst naar Dialogflow gestuurd.

6.1.1 Dialogflow audiobestand

Vanuit de Dialogflow library is het mogelijk om een request te versturen naar Dialogflow zelf met een audiobestand. Dit wordt vervolgens bij Google geanalyseerd en verwerkt. Een van de nadelen richting de gebruiker van de chatbot is het missen van de functionaliteit om visuele feedback te geven aan wat hij zegt. Pas nadat het geluid is opgestuurd naar Dialogflow en alles verwerkt is kan er gevisualiseerd worden wat de gebruiker heeft gezegd. Deze methode zal uit twee onderdelen bestaan. Naast de connectie die opgezet moet worden met Dialogflow moet het geluid ook nog goed opgenomen worden. Een van de uitdagingen zal zijn om de opname op het goede moment te stoppen. Als dit niet goed werkt zal er of te weinig opgenomen worden of zal de gebruiker langer moeten wachten op antwoord nadat hij klaar is met spreken.

6.1.2 Web Speech API

Deze functionaliteit werd voor het eerst in begin 2013 geïntroduceerd in Google Chrome. De webkit kon gebruikt worden om in realtime audio te transcriberen in verschillende talen. De Web Speech API kan zowel door client als server side worden uitgevoerd en biedt directe feedback op wat er gezegd wordt. Zo kan de gebruiker direct zien wat de API heeft verstaan. Daarnaast heeft de Web Speech API ingebouwde functies om te herkennen wanneer een gebruiker stopt met praten. Een nadeel van het gebruik van de webkit Speech Recognition is de compatibiliteit met verschillende browsers. Op het moment wordt deze functie namelijk alleen maar ondersteund door Google Chrome.

6.1.3 Conclusie

Ondanks de compatibiliteit beperking van de Web Speech API is er de keuze gemaakt om hiervoor te gaan. Omdat ik de uiteindelijke ontwerper ben van het prototype kan ik ook de keuze maken welke browser er gebruikt gaat worden in het eindproduct. Daarnaast heeft de Web Speech API een groot aantal nuttige functionaliteiten al ingebouwd. Zo kan er live getranscribeerd worden in het front-end gedeelte en wordt er automatisch herkend wanneer een gebruiker klaar is met praten.

6.2 Audio output chatbot

Er is in overleg met de opdrachtgever besloten om de chatbot zelf geen stem te geven. Dit zou namelijk voor overlast kunnen zorgen bij de receptiemedewerkers van Capgemini. Zij zijn de hele dag in de buurt van de chatbot aan het werk en dit zou afleidend kunnen werken. De tekst op het scherm zal nu dus de enige manier van interactie zijn voor het prototype.

6.3 Retrospective

Tijdens deze sprint is de chatbot nu in staat om steminvoer te verwerken. Hierdoor is de laatste functionaliteit toegevoegd aan de chatbot en zijn er verder geen taken meer voor nieuwe functionaliteit. In de laatste sprint van de stage zal de focus alleen liggen op het deployen van het project. Er is al een aanvraag gedaan voor server capaciteit en een domein bij de AIE van Capgemini.

7. Sprint 7 9 mei tot en met 23 mei

7.1 Deployment

De laatste stap binnen het project was het daadwerkelijk beschikbaar maken van het project. De chatbot wordt beschikbaar gemaakt via een webserver. Via Capgemini heb ik toegang gekregen tot een AWS Lightsail virtual server. Daarnaast heb ik een subdomein gekregen voor de chatbot. Na het deployen van het project zal er ook gebruik moeten worden gemaakt van een SSL-verbinding. Naast het veiligheidsaspect is hiervoor ook gekozen omdat dit een vereisten is vanuit Dialogflow. Om fulfillment toe te passen vanuit Dialogflow is er namelijk een https endpoint nodig.

7.1.1 Meteor deployment

Voor het deployen van een Meteor project heb ik gebruik gemaakt van een deployment tool genaamd Meteor Up. Deze tool geeft de mogelijkheid door eenmalig een configuratiebestand op te zetten en daarna met een commando het hele project te deployen. Het maakt gebruik van Docker wat veel voordelen met zich mee brengt. De NodeJS app en MongoDB draaien beide in een losse container. Meteor up biedt daarnaast ook eenvoudige SSL-support aan. Het maakt gebruik van de gratis tool lets encrypt die voor de ondertekening van een SSL-certificaat zorgt. Het is uiteindelijk niet gelukt om een ssl certificaat te kunnen koppelen aan het subdomein. Ik heb er toen voor gekozen om het project eerst te testen op een eigen server met een normaal domein. Nadat dit allemaal gelukt was kon ik de conclusie trekken dat er iets fout ging met het deployen van het project naar een subdomein voor de SSL-certificaat. Vanwege het naderende eind van de stageperiode heb ik ervoor gekozen om niet meer tijd te steken in het uitzoeken hoe de applicatie op een subdomein zou moeten werken. Na mijn stageperiode zal ik nog een gesprek hebben met iemand die het project na mijn stage gaat overnemen om uit te leggen wat ik heb geprobeerd en hoe hij het eventueel zou kunnen oplossen.

7.1.2 Python deployment

Voor het deployen van een python project gebruik ik Docker-compose. Hierdoor draait ook het python project in een container. De Flask applicatie is beschikbaar op de gespecificeerde poort van de server. Flask heeft de mogelijkheid om een SSL-certificaat te specificeren bij het opstarten. Uiteindelijk was dit alleen niet mogelijk, omdat Meteor Up de webserver (nginx) en SSL-certificaten ook in een Docker container heeft aangemaakt. Vanwege deze complicaties is er uiteindelijk voor gekozen om de Python functies in de Meteor applicatie in te bouwen. Vanwege het principe dat het alleen om de functionaliteit gaat van de chatbot was dit een mogelijke optie. Doordat deze functies en de HTTPS endpoint nu allemaal in hetzelfde project bestonden is het ook makkelijker om aan het project te werken.

7.1.3 Fysieke deployment prototype

De chatbot zelf zal uiteindelijk op een desktop gaan draaien die bij de receptie staat. Een van de vereisten van de desktop is de aanwezigheid van een goede microfoon, zodat de stem van de gebruikers zo goed mogelijk wordt opgevangen. Daarnaast zal op een kleine tablet de koffie bestellijst beschikbaar worden voor de kiosk bij Capgemini. Het overdragen van het development project zal op een later tijdstip plaatsvinden met een ontwikkelaar binnen Capgemini.

7.2 Retrospective

Ondanks de mislukte poging om het project zoals gepland te deployen is de sprint goed verlopen. Uiteindelijk is het wel gelukt om het prototype werkend te krijgen op het internet. Na deze sprint zal ik alleen nog mijn werk moeten overdragen aan de volgende ontwikkelaar die de leiding zal nemen over het project.

Evaluatie Aanpak

In dit onderdeel worden de verschillende stappen beschreven die tijdens dit project zijn uitgevoerd. Deze zijn onder te verdelen in: keuze van de functionaliteiten, keuze tools, ontwerpen van de software architectuur en het dialoog-testsysteem.

Keuze van functionaliteiten

Vanwege de absentie van vooropgestelde vereisten en functionaliteiten had dit prioriteit aan het begin van het project. Door eerst in gesprek te gaan met de verschillende actoren die op een bepaalde manier invloed hebben op de chatbot, is er een lijst gemaakt met mogelijke functionaliteiten. Deze lijst is toen besproken met de begeleider, waar de gekozen functionaliteiten uit voort kwamen. Achteraf heeft het mij heel veel geholpen om eerst met de verschillende actoren te praten. Hierdoor kreeg ik namelijk direct een beter beeld van de organisatie en de werkzaamheden. Ook werd voor mij het proces die bezoekers doorlopen duidelijker en kon ik goed kijken waar de chatbot waarde kan toevoegen. In sprint 2 is duidelijk geworden dat het gebruik van persoonlijke gegevens een mogelijk knelpunt zou worden. Er is toen direct besloten om naar een nieuwe hoofdfunctionaliteit te kijken. Dankzij deze keuze had ik nog genoeg tijd om op zoek te gaan naar een nieuwe mogelijkheid. Wel heb ik nog mail contact gehad met verschillende partijen binnen Capgemini om te kijken of de functionaliteit achteraf toch nog toegevoegd kan worden. Achteraf gezien was het een verstandige keuze om direct naar een nieuwe functionaliteit te zoeken. Hierdoor heb ik uiteindelijk meer tijd gehad om deze uit te werken en is er verder geen tijdnoed ontstaan. Daarnaast heeft dit mij ook geleerd dat het vragen om toestemming in grote bedrijven op een andere manier gaat dan bij kleinere bedrijven. Dit heeft uiteindelijk niet voor problemen gezorgd binnen het prototype project, maar het is wel iets waar ik meer rekening mee had moeten houden. Na de keuze om de chatbot als ondersteuning voor de kiosk te bouwen is er nauwelijks wat veranderd. Zelf ben ik tevreden met de functionaliteiten die de chatbot nu heeft. Ook Capgemini is tevreden met de kwaliteit van de functionaliteiten van het prototype.

Keuze tools

Door voor het vaststellen van de functionaliteiten al gekozen te hebben welke tools ik wilde gebruiken bracht dit een aantal risico's met zich mee. Bij mijn volgende projecten zou ik eerst duidelijke functionaliteiten vastgesteld hebben. Een van de risico's is de vertraging die opdoet als de architectuur aangepast moet worden dankzij een verandering van functionaliteiten. Achteraf gezien zijn bijna alle keuzes goed geweest. Alleen de python https endpoint was niet goed te implementeren. Ondanks dat kon de Python functionaliteiten snel worden overgezet naar de Meteor applicatie om een werkend prototype te leveren. Deze inschatting is verkeerd uitgelopen, door ervan uit te gaan dat het regelen van een SSL-verbinding voor de Flask endpoint op een correcte manier mogelijk was. Als ik hier verder onderzoek naar had gedaan, had ik al eerder voor een andere tool kunnen kiezen. Maar om low coupling te stimuleren binnen de architectuur is ervoor gekozen om dit als losse applicatie te realiseren.

Ontwerpen software architectuur

Na het kiezen van de meest geschikte tools was het een goed moment om te kijken hoe alles samen zou gaan werken. Ondanks de verandering in functionaliteiten in sprint 3 is de eerste versie van de architectuur nauwelijks veranderd. De diagrammen die zijn gemaakt zijn gebruikt voor het inrichten en toelichten van het systeem. Er is een goede afweging gemaakt tussen de beschikbare stagetijd en 'best practices' bij het ontwerpen van de software architectuur. Ook hebben de diagrammen geholpen voor het verduidelijken van de werking van de chatbot en de onderliggende technieken. Ondanks de

uiteindelijke uitwerking van de chatbot, kunnen de eerder gemaakte diagrammen gebruikt worden voor een toekomstige implementatie van een nieuwe chatbot. Er moet dan alleen voor een andere uitwerking gekozen worden gebaseerd op de endpoint. Ik ben zelf tevreden met de ontwerpen van de architectuur en de uitwerking hiervan. Als ik meer tijd had gehad, zou het mogelijk zijn geweest om alles zoals gepland uit te werken. Voor Capgemini heeft het gelukkig geen verschil gemaakt en ging het uiteindelijk om de werking van het prototype en die is gelijk gebleven ondanks de verschillen.

Dialogtesten

Ik ben erg tevreden met het systeem dat ik heb ontworpen voor het testen van Dialogflow. Omdat deze functionaliteit nog niet beschikbaar is in Dialogflow, zou dit ook als losse tool gebruikt kunnen worden. Dankzij deze functionaliteit hebben zowel de ontwikkelaar van de chatbot als de opdrachtgever een zekerheid dat de uitgestippelde gesprekken goed zullen verlopen. Ook dankzij de onzekerheid van Dialogflow die constant door Google geüpdatet wordt, kan het zomaar zijn dat een chatbot ineens andere intenties herkent. De visuele uitwerking zorgt er ook voor dat niet alleen ontwikkelaars inzag hebben over de effectiviteit van de chatbot.

Evaluatie Opgeleverde producten

Aan het einde van het project is er een prototype van de chatbot opgeleverd. Dit is in de vorm van een webapplicatie met verschillende interfaces. De uiteindelijke uitwerking van de plaatsing van de chatbot zal later na de stageperiode nog besproken worden. Hieronder zal ik per verschillende interfaces beschrijven of deze overeenkomen met de verwachtingen en planning.

Chatbot interface

Deze interface is voor de gebruiker nu bruikbaar. Alle geplande functies zijn aanwezig en de gebruiker kan met zijn stem met de chatbot praten. Omdat er minder aandacht is besteed aan de UX van de applicatie is dit iets wat nog mogelijk kan worden verbeterd. Ook is er niet direct rekening gehouden met Capgemini kleurenschema en logo's. Omdat het om een prototype ging is dit verder geen probleem en zoals afgesproken. Daarnaast is het niet gelukt om de applicatie in de meeste browsers werkend te krijgen. De huidige versie werkt alleen in Google Chrome op een desktop en Android. Om de applicatie beschikbaar te maken in Firefox zal er naar een andere manier van voice input gekeken moeten worden.

Chatbot Dialogflow

De chatbot mogelijkheden om te antwoorden op de gebruiker is goed afgerond. De 'happy path' van de chatbot werkt altijd. Daarnaast is er nog rekening gehouden met twijfel of verandering van de bestelling. Ook zijn er conditionele vragen en zo min mogelijk bevestigende vragen gesteld. Dit zorgt voor een zo natuurlijk mogelijk gesprek tussen de bezoeker en de chatbot. Verder is de verbinding tussen Dialogflow en het front-end verlopen zoals verwacht. Ik ben erg tevreden over de uitwerking van de chatbot, en als prototype is deze goed inzetbaar.

Toekomstvisie

De interface van de chatbot kan vooral verbeterd worden door er een gebruiksvriendelijkere UI/UX te ontwerpen. Dit staat los van de functionaliteiten en kan dus later worden aangepast wanneer hier tijd voor is.

De chatbot kan in de toekomst verder uitgebreid worden met nieuwe gespreksonderwerpen. Zo kan bijvoorbeeld de NS-functionaliteit uitgebreid worden waarbij de gebruiker alleen zijn bestemming hoeft te noemen en dan mogelijke reisopties te zien krijgt. Ook kan er bijvoorbeeld een spel worden ontwikkeld waarbij de bezoeker een kleine quiz kan spelen met trivia vragen over Capgemini tijdens het wachten. Dit is een functionaliteit die zeer goed uitgewerkt kan worden in een chatbot met Dialogflow.

Evaluatie Competenties

Vooraf vastgelegde leerdoelen worden hieronder nader toegelicht; hoe deze zijn toegepast en bewezen tijdens de afstudeerstage. Dit wordt deels met referenties gedaan naar kopjes en met directe toelichting per competentie. Ook wordt vergeleken op welke wijze de competentie is gedemonstreerd ten opzichte van het stageplan.

A1. Analyseren probleemdomein & opstellen probleemstelling

Door aan het begin van het project in gesprek te gaan met experts en de stakeholders van het project is het probleemdomein geanalyseerd. Deze analyse is verder uitgewerkt in het plan van aanpak voor de afstudeerstageopdracht. In de eerste en tweede sprint is er onderzoek gedaan naar de huidige uitwerkingen van chatbots en naar mogelijkheden om een conversatie chatbot te ontwikkelen. Er is ook onderzoek gedaan naar de verschillende tools die beschikbaar zijn voor het ontwikkelen van het prototype en er is een overwogen keuze gemaakt tussen de verschillende opties. Dankzij deze analyse kon goed in kaart worden gebracht welke risico's er mogelijk zijn en alvast meerdere mogelijkheden uitgewerkt worden indien bepaalde problemen zich zouden voordoen.

Gc. Kritisch, onderzoekend en methodisch werken

Elke twee weken werden er pitches gegeven door een groep stagiaires. Bij deze pitches kreeg iedereen een aantal minuten de tijd om zijn afgelopen twee weken toe te lichten. Je moest ook aangeven wat goed en wat minder goed ging in die periode. Hierdoor was het belangrijk dat je kritisch naar je eigen werkwijze keek. Daarnaast kon je dan met de groep bespreken hoe je het een volgende keer beter had kunnen aanpakken en is er ruimte om zelf die reflectie uit te voeren.

Naast het ontwikkelen van het prototype als eindproduct is er direct een onderzoeksverslag bijgehouden. Hierin is genoteerd welke keuzes er gemaakt zijn en hoe deze tot stand zijn gekomen. Indien er belangrijke keuzes zijn gemaakt is dit gedocumenteerd in het onderzoeksverslag. Door op deze manier te werken is er altijd een mogelijkheid om terug te blikken op de gemaakte keuzes. Door deze mogelijkheden direct te verwerken in een document was dat ook een geschikt moment om te reflecteren. Hiermee kunnen fouten worden voorkomen en kan er direct worden ingegrepen wanneer er toch iets niet klopt of onduidelijk is.

Gf. Leren leren

Deze competentie is gedemonstreerd door voorafgaand aan het ontwikkelen goed literatuuronderzoek te hebben uitgevoerd. Ik heb een onderscheid moeten maken tussen relevante en niet relevante literatuur. Hiermee heb ik een goede basis kunnen vormen voor het project.

Capgemini biedt daarnaast een veelvoud trainingen aan voor zijn medewerkers. De meeste van deze trainingen worden door andere medewerkers gegeven die graag kennis deelt met anderen. Door de trainingen blijf ik bij in het vakgebied en deze kunnen daarnaast ook verbredend zijn. De cursussen "The selling consultant" en "Human Profiling" waren voor mij erg waardevol. Naast het ontwikkelen van mooie software zijn dit twee onderwerpen die naar mijn mening minder vaak terugkomen in het onderwijs. Wel zijn deze zeer belangrijk, zeker bij het werken als een bedrijf als Capgemini waar je direct bij de klant ingezet wordt.

Met recente internet artikelen heb ik mij verder kunnen verdiepen in de chatbot ontwikkelingen. In verschillende blogs en artikelen werd veel gesproken over verschillende manieren van inzetten van chatbots en de toegepaste technologieën. Omdat de chatbot technologie nog in de kinderschoenen staat, wordt er nog veel nieuws ontwikkeld en het is daardoor van belang om up-to-date te blijven.

C6. Ontwerpen software

Deze competentie is gedemonstreerd door een software architectuur voor de chatbot te ontwerpen. Er zijn tijdens de eerste sprints verschillende diagrammen gemaakt waarin de samenwerking van de verschillende systemen wordt omschreven. Ook is er gekeken naar de verschillende actoren die gebruik zullen maken van het systeem. Er is rekening gehouden met de beveiliging en best practices op de verschillende niveaus. Zo is er bij het ontwikkelen van de applicatie rekening gehouden met veelvoorkomende beveiligingsrisico's. Daarnaast is er op server niveau gebruik gemaakt van Docker en wordt er gebruik gemaakt van een beveiligde verbinding via SSL. Ook was het tijdens het project van groot belang dat er een goede afweging gemaakt was tussen beschikbare tijd en de prioriteit van de functionaliteiten van de chatbot. De keuzes op meso en macroniveaus zijn beide verder grotendeels toegelicht in sprint 2 en 3.

C11. Ontwerpen van human computer interaction

Deze competentie is gedemonstreerd door een prototype te ontwikkelen van de chatbot die kan communiceren met de bezoeker. Capgemini heeft hierdoor direct een fysiek voorbeeld van de mogelijkheden van een conversatie chatbot. Ook de bezoekers krijgen hierbij direct een impressie van Capgemini, dat als bedrijf constant aan het experimenteren is met nieuwe technologieën. Daarnaast laat het prototype zien dat het mogelijk is om in een stageperiode van een half jaar een chatbot te ontwikkelen die gebruik maakt van mens-machine interactie. Door het inzetten van een prototype is te zien wat er goed werkt en wat verbeterd of anders zou kunnen. De uitwisseling van informatie tussen de machine en de bezoeker wordt bij het prototype gedaan met behulp van stem en tekst. Door gebruik te maken van stemherkenning kan de bezoeker makkelijk praten met de chatbot. Hierna kan hij op het scherm lezen wat voor antwoord de chatbot geeft en wat de volgende vraag is.

D15. Testen

Door het opstellen van een testplan en deze te volgen tijdens het project is er meer zekerheid ontstaan over de werking van het prototype. Met behulp van flowcharts voor de chatbot is er een dialoog testsysteem ontwikkeld. Hiermee werden de verwachte uitkomsten beschreven voor de chatbot en kan gevalideerd worden of Dialogflow consistent werkt. Daarnaast is er met behulp van unittests de basis functionaliteit van de webapp en endpoint gewaarborgd. Door op deze methodische manier aan het werk te zijn is dit zowel voor mij als voor de opdrachtgever zekerheid dat de chatbot werkt zoals verwacht en alle basisfunctionaliteiten een verwachte uitkomst hebben. De uitwerking van het testplan en dialoogtestsysteem is terug te vinden in sprint 4 en 5.

Ge. Innovatief en creatief werken

Deze competentie is aangetoond door het introduceren van nieuwe technologieën binnen Capgemini. Een voorbeeld hiervan is de stemherkenning en de toepassing van NLU bij deze chatbot. De chatbot zal uiteindelijk als innovatie voor de bezoekers van Capgemini worden ingezet. In tegenstelling tot hoe het nu is, waar de bezoeker gevraagd wordt plaats te nemen tot de werknemer van Capgemini hen komt halen. Door de introductie van een nieuwe technologie kunnen de bezoekers van Capgemini geïnspireerd en 'vermaakt' worden en wordt het proces om koffie/thee te bestellen ondersteund.

Daarnaast heb ik zelf creatief gewerkt door de functionaliteiten uit te werken van de chatbot. Omdat er bij de start van het project nog geen duidelijke rol was voor de chatbot moest dit eerst bepaald worden. Zelf moest ik bedenken hoe ik op de verschillende functionaliteiten kon komen. De bedachte functionaliteiten heb ik later moeten toetsen of deze nuttig zouden zijn voor de gebruiker. Uiteindelijk is er zo een lijst met functionaliteiten uitgekomen voor het project.

Bibliografie

- [1] Capgemini, “Applied Innovation Exchange.” [Online]. Available: <https://www.capgemini.com/nl-nl/diensten/transformation-innovation/applied-innovation-exchange/>. [Accessed: 09-Apr-2019].
- [2] Vue.js, “Vue.js.” [Online]. Available: <https://vuejs.org/>. [Accessed: 20-Mar-2019].
- [3] Facebook Inc., “React – A JavaScript library for building user interfaces.” [Online]. Available: <https://reactjs.org/>. [Accessed: 20-Mar-2019].
- [4] Google, “Angular.” [Online]. Available: <https://angular.io/>. [Accessed: 07-May-2019].
- [5] Meteor Development Group Inc., “Build Apps with JavaScript | Meteor.” [Online]. Available: <https://www.meteor.com/>. [Accessed: 20-Mar-2019].
- [6] Google, “Dialogflow.” [Online]. Available: <https://dialogflow.com/>. [Accessed: 09-Apr-2019].
- [7] Rasa Technologies GmbH, “Rasa: Open source conversational AI.” [Online]. Available: <https://rasa.com/>. [Accessed: 09-Apr-2019].
- [8] OpenWeatherMap, “Weather API - OpenWeatherMap.” [Online]. Available: <https://openweathermap.org/api>. [Accessed: 20-Mar-2019].
- [9] Capgemini, “Our Experts.” [Online]. Available: <https://www.capgemini.com/nl-nl/experts/>. [Accessed: 20-Mar-2019].
- [10] C. O’Keefe, “Practical Introduction to Web Scraping in Python – Real Python.” [Online]. Available: <https://realpython.com/python-web-scraping-practical-introduction/#getting-the-popularity-score>. [Accessed: 20-Mar-2019].

Lijst met afkortingen

AIE – Applied Innovation Exchange

API – Application programming Interface

Async – Asynchroon

AWS – Amazon Web Services

DB – Database

HTTPS – Hyper Text Transfer Protocol Secure

JS – Javascript

LUIS – Language Understanding Intelligent Service

NLP – Natural Language Processing

NLU – Natural Language Understanding

SSL – Secure Sockets Layer

UI – User Interface

UX – User Experience

Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2019-1.1 (start uiterlijk 4 februari 2019)

Startdatum uitvoering afstudeeropdracht:

Inleverdatum afstudeerdossier volgens jaarrooster: 31 mei 2019

Studentnummer: 15072924

Achternaam: dhr Frionnet

Voorletters: T

Roepnaam: Timo

Opleiding: HBO ICT Software Engineering

Locatie: Zoetermeer

Variant: voltijd

Naam studieloopbaanbegeleider: Vincent Broeren

Naam begeleidend examiner: Hani Al-Ers

Naam expert examiner: Ton Biegstraaten

Naam bedrijf: Capgemini

Afdeling bedrijf: Applied Innovation Exchange

Bezoekadres bedrijf: Reykjavikplein 1

Postcode bezoekadres: 3500 GN

Postbusnummer: 2575

Postcode postbusnummer:

Plaats: Utrecht

Telefoon bedrijf:

Telefax bedrijf:

Internetsite bedrijf: <https://www.capgemini.com/nl-nl/>

Achternaam opdrachtgever: dhr Tonen

Voorletters opdrachtgever: P

Titel opdrachtgever: Director Applied Innovation Exchange

Achternaam bedrijfsmentor: dhr Tonen

Voorletters bedrijfsmentor: P

Titel bedrijfsmentor: Director Applied Innovation Exchange

Doorkiesnummer afstudeerder:

Functie afstudeerder (deeltijd/duaal):

Titel afstudeeropdracht:

Ontwikkeling van een prototype conversatie chatbot bij Capgemini.

Opdrachtomschrijving

1. Bedrijf

Capgemini is een consultancybedrijf op het gebied van ICT. Capgemini is internationaal werkzaam en heeft alleen in Nederland al meer dan 5000 werknemers. In totaal biedt Capgemini werk aan meer dan 210.000 werknemers. De internationale omzet bedraagt 10,1 miljard euro en ze bezitten 9 kantoren in Nederland. Het hoofdkantoor is in Parijs gevestigd en Capgemini is ook een Europees bedrijf. Wel heeft elk land zijn eigen verantwoordelijkheid.

De afdeling waar ik kom te werken is gericht op innovatie en heet Applied Innovation Exchange. Er wordt niet direct voor de klant gewerkt, maar er wordt vooral onderzoek gedaan naar opkomende technologieën en mogelijke toepassing hiervan. Met deze bevindingen gaat Capgemini kijken hoe ze dit kunnen aanbieden bij hun huidige klanten. Binnen deze afdeling werken meestal 12 andere stagiaires die onderzoek doen of ook bezig zijn met het ontwikkelen van prototypes.

2. Probleemstelling

Chatbots worden al snel gezien als simpele tools die voorgeprogrammeerde vragen kunnen beantwoorden en zo de menselijke support kunnen vervangen. Capgemini wilt verder dan dit idee en wilt een chatbot ontwikkelen die echt conversatie kan maken met mensen. Die door initiatief te nemen en door verder te vragen naar details uit het antwoord een gesprek kan aangaan en kan aanhouden. Er zit geen urgentie aan dit project, omdat het niet in opdracht is van een klant. Wel wil Capgemini natuurlijk zo snel mogelijk weten of deze vernieuwing in de technologische wereld iets is wat ze aan klanten kunnen aanbieden of niet. Vandaar dat Capgemini ook als prioriteit ziet om een demo te ontwikkelen. Hiermee kan worden getoond of het een interessante toepassing is van de technologie.

Op het moment wordt er aan een eerste demo gewerkt die de basis functionaliteiten bevat van een chatbot. Dit prototype is nog niet ver uitgewerkt op gebied van initiatief nemen en doorvragen dus dat zal voor mij de focus zijn van het project.

3. Doelstelling van de afstudeeropdracht

De doelstelling is het opleveren van een werkende demo/prototype die kan laten zien hoe een chatbot in de praktijk kan worden ingezet als bijvoorbeeld portier of op een website om mensen welkom te heten. De chatbot moet initiatief nemen en moet met behulp van de gegeven antwoorden het gesprek op gang kunnen houden.

4. Resultaat

Het gevonden resultaat zal indien naar wens verder ontwikkeld kunnen worden of als prototype kunnen worden gepresenteerd aan klanten die wellicht geïnteresseerd zijn in eenzelfde soort oplossing voor hun bedrijf. Het zal als een proof of concept worden gebruikt met het doel klanten geïnteresseerd te krijgen in deze technologische mogelijkheid.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Deze stage zal beginnen met het opzetten van een globaal plan van aanpak. Er is vooraf nog niet duidelijk hoe we tot het eindproduct zullen komen en welke tools daarvoor bijvoorbeeld nodig zijn. In het plan van aanpak zal in globale lijnen beschreven worden welke onderdelen belangrijk zijn tijdens de stage met een algemene tijdsplanning.

Er zal onderzoek moeten worden gedaan naar chatbots in het algemeen en naar conversatie chatbots.

Voordat er geprogrammeerd kan worden moet er worden gekeken welke tools er nu worden gebruikt en welke tools beschikbaar zijn voor het ontwikkelen van de chatbot. Na het bepalen van de basis tools kan er een ontwerp worden gemaakt van het programma. Er wordt vastgesteld in welke programmeertaal gewerkt gaat worden en waarom hiervoor is gekozen. Indien ervoor wordt gekozen om dit via bijvoorbeeld een API te maken is het minder van belang om te weten op welke manier de tool later gebruikt gaat worden. Indien dit niet het geval is, is het wel belangrijk om vooraf

te weten op welke manier de tool ingezet gaat worden zodat hier rekening mee kan worden gehouden.

Ik ben van plan om met behulp van sprints te werken waardoor er naar elke sprint wordt gekeken hoe de chatbot functioneert en of het nog naar wens is. Ook zal er meteen gekeken worden hoe hij verbeterd kan worden en op welke fronten het nog fout gaat.

Globale planning:

Plan van aanpak: sprint 1

Aanvullende oriëntatie: sprint 1

Onderzoeken: sprint 2 / 3

Ontwikkelen / programmeren: sprint 3 / 4 / 5

Ontwerp: sprint 2 / 3

Testen: sprint 4 / 5 / 6

Adviseren: sprint 6 / 7

Implementeren: sprint 6 / 7 / 8

6. Op te leveren (tussen)producten

Onderzoek:

Het onderzoek dat ik heb uitgevoerd om te bepalen hoe ik de tool ga ontwikkelen. Mijn beslissingen zullen hier van waarde zijn voor Capgemini. Dit kan worden gebruikt voor toekomstig onderzoek en als ondersteuning van mijn product.

Demo/Prototype:

Het gemaakte product zal opgeleverd moeten worden in hoeverre het afgerond is aan het einde van mijn stage. Natuurlijk is het doel om aan het einde van elke sprint een potentiële shippable product op te leveren. Dit is dan ook wat gezien kan worden als tussenproduct.

7. Te demonstreren competenties en wijze waarop

A1 – Analyseren probleemdomein & opstellen probleemstelling: Wordt gedemonstreerd door het opstellen van het plan van aanpak en door onderzoek te doen naar het probleemdomein.

Gc – Kritisch, onderzoekend en methodisch werken: Wordt gedemonstreerd door het onderzoeksplan dat bijgehouden gaat worden en door de beslissingen die gemaakt zijn tussen verschillende methodes. Hierin moet kritisch worden gekeken naar de voor en nadelen voordat er zomaar een bepaalde programmeertaal wordt gekozen.

Gf – Leren leren: voorbereiden op volgende studiefase en beroep: Wordt gedemonstreerd door alle werkzaamheden en onderzoek dat gedaan zal worden. Omdat het om nieuwe vaardigheden gaat leer ik hoe ik dit op een andere manier kan oppakken. Het grootste verschil met mijn eerste stage is dat ik hier bij Capgemini veel meer vrijheid heb gekregen qua keuze van tools en methodes, wat ervoor zorgt dat ik echt zelf eerst onderzoek moet doen naar de meeste optimale keuzes.

C6 – Ontwerpen Software: Wordt gedemonstreerd door het zelfontworpen project. Door zelf de vrijheid te krijgen voor het ontwerpen van de tool die ontwikkeld moet worden is dit zeker een niveau 3 taak. Vanuit Capgemini zijn er geen restricties voor het kiezen van welke tools er gebruikt gaan worden en laat het zelf de student deze keuzes uitzoeken en maken.

D15 - Testen: Wordt gedemonstreerd door zelf een testprocedure op te stellen waar wordt gekeken naar de algemene kwaliteit van het gesprek met de chatbot. Naast het zelf kunnen testen van de

chatbot moet er ook een objectieve manier worden ontwikkeld om te kijken of de chatbot beter wordt in het communiceren met een niet computer.

Ge – Innovatief en creatief werken: Wordt gedemonstreerd doordat de software die ontwikkeld moet worden nog niet beschikbaar is als een standaardproduct. Er moet dus onderzoek worden gedaan naar innovatieve technieken om tot een combinatie te komen die voor een goed resultaat zorgt.

C11 – Ontwerpen van Human Computer Interaction: Wordt gedemonstreerd door de tool die ontwikkeld zal worden. Met behulp van een conversatie chatbot wordt er een nieuwe manier van Human Computer Interaction ontwikkeld. Het idee van de tool is dat hij de conversatie start en in standhoudt met de gebruiker in tegenstelling tot reguleren Human Computer Interaction.

Inleiding

Dit document beschrijft het plan van aanpak voor de afstudeeropdracht waarbij een chatbot prototype wordt ontwikkeld voor Capgemini. In een periode van 17 weken, wordt er verwacht dat er een prototype wordt opgezet van een conversatie chatbot. Dit houdt in een chatbot die de conversatie aan kan gaan met de gebruiker. Op basis van de response van de gebruiker moet de chatbot het gesprek op gang houden.

In dit verslag wordt eerst een korte toelichting gegeven over het project zelf en het bedrijf waar het project wordt uitgevoerd. Er wordt beschreven wat de doelstellingen zijn van het project en wat er wordt verwacht vanuit het bedrijf en vanuit de Haagse Hogeschool. Uiteindelijk wordt er toegelicht welke ontwikkelmethoden er wordt toegepast tijdens het verslag en wordt er een globale planning gemaakt samen met een risicoanalyse.

Achtergrond van het project.

Capgemini is een consultancybedrijf dat IT gerelateerde diensten aanbiedt. De kerndienst van Capgemini is het bedenken, realiseren en beheren van IT-applicaties. Om op de hoogte te blijven van de nieuwste technologieën hebben zij de Applied Innovation Exchange (AIE) opgezet. Dit is een interne afdeling waarin werknemers onderzoek kunnen uitvoeren naar nieuwe technologieën. Een onderdeel van de AIE zijn de stagiaires die onderzoek doen naar mogelijkheden van nieuwe technologieën. De stakeholder in mijn project is de AIE. Zij willen graag met mijn demo zien op wat voor manier een chatbot kan worden ingezet om bijvoorbeeld mensen bij de receptie te verwelkomen. De AIE bevindt zich buiten de algemene bedrijfsstructuur. Dit omdat de AIE niks ontwikkeld wat direct wordt ingezet bij klanten. De projecten die worden uitgevoerd binnen de AIE komen voort uit vragen vanuit klanten of naar marktonderzoek. De projecten blijven binnen Capgemini en zullen enkel als bron worden gezien voor toekomstige uitwerkingen van eenzelfde soort technologie bij een klant. Chatbots worden op het moment steeds vaker ingezet om producten te verkopen. Het nut van een Alexa is om meer te verkopen via Amazon. Hetzelfde geldt voor chatbots op webpagina's die de gebruiker helpen om het beste product te vinden dat bij hen past. Waar Capgemini meer onderzoek naar wil doen is een chatbot die het gesprek aangaat. Hier is een stuk minder onderzoek naar gedaan en is ook nog niet duidelijk hoe het mogelijk ingezet kan worden bij een klant. Capgemini heeft op het moment deze vraag nog steeds liggen en wilt hier graag meer over weten om eenzelfde soort product in de toekomst aan te kunnen bieden bij klanten.

Doelstelling

Capgemini is constant bezig met onderzoek naar nieuwe technologieën. Zo ook naar conversatie chatbots. Zij zijn vooral benieuwd naar de use cases van deze technologie en hoe dit kan worden ingezet bij klanten. Het doel van deze opdracht is dan ook het ontwikkelen van een demo dat als inspiratie kan worden gebruikt om eenzelfde soort tool te ontwikkelen bij klanten.

Opdracht

Als opdracht is er een project gestart om een conversatie chatbot te ontwikkelen die als portier bij Capgemini mensen begroet. Na de begroeting gaat de chatbot een conversatie

aan met de bezoeker en probeert met behulp van directe vragen een beeld te krijgen van de bezoeker. De chatbot moet op basis van vooraf bepaalde vragen een gesprek kunnen leiden. Daarnaast moet het antwoord van de gebruiker geïnterpreteerd kunnen worden en moet hier indien mogelijk op door kunnen worden gevraagd. Met behulp van deze demo kan Capgemini zien of het inzetten van een chatbot op deze manier nuttig is en of het ook mogelijk aan klanten aangeboden kan worden. Ook kan met behulp van mijn onderzoek worden gekeken of chatbots ook nog andere toepassingen kennen die op het moment nog niet worden toegepast.

Op te leveren resultaten

Het op te leveren resultaat is de demo applicatie die met behulp van een microfoon een gesprek kan aangaan bij de ingang van Capgemini. Na het stellen van vragen worden de gegeven antwoorden geïnterpreteerd en waar mogelijk is wordt hierop doorgevraagd. Uiteindelijk sluit de chatbot het gesprek goed af en kan het een gesprek aangaan met de volgende bezoeker.

Projectgrenzen

De projectdeadline is gezet op 31 mei. Dit is dezelfde dag dat ook het verslag ingeleverd moet worden. Het project bevat naast de basis functionaliteit van de chatbot ook nog verdere implementaties. Deze implementaties vallen niet onder de afstudeeropdracht, maar worden gezien als vervolgoopdracht voor een volgende afstudeerder of als extra bezigheid indien het project voorspoedig wordt afgerond.

Enkele voorbeeld van deze externe implementaties:

- Het koppelen van externe API's zoals een weersvoorspelling of met sociale media zoals LinkedIn of Twitter om meer context te geven aan een vraag.
- Het herkennen van gezichten waardoor een gesprek op een later moment vervolgd kan worden.
- Het herkennen van emotie met behulp van sentiment- of gezichtsanalyse. De resultaten van deze module kunnen worden gebruikt als input voor het kiezen van een pad.

Randvoorwaarden

De deadline die is gesteld door de Haagse Hogeschool voor het afstudeerrapport is op 31 mei 2019. Op 60% van de afstudeerstage vindt een bespreking plaats van het concept afstudeerdossier. Deze bespreking heeft geen deadline, maar moet wel uitgevoerd worden.

Op 80% vindt er een tussentijds assessment plaats. Dit is dan ongeveer 4 weken voor het inlevermoment en wordt gebruikt om te bepalen of de student op schema loopt om op 31 mei het rapport in te leveren of dat er misschien verlenging van de stage moet worden aangevraagd.

De begeleider vanuit Capgemini is vaak aanwezig op kantoor. Indien er korte vragen zijn, kan er altijd direct of via de mail contact worden gezocht. Er is afgesproken dat om de week er tijd is om samen met de begeleider en 5 andere studenten samen te komen. Met behulp van pitches wordt de voortgang bewaakt en wordt er overlegd wat er niet goed gaat en waar nog vragen over zijn. Na de pitches is er een uur de tijd om nog benodigde zaken persoonlijk te bespreken.

Om het project te starten kan er op de laptop van de student gewerkt worden. Indien de basis functionaliteiten afgerond zijn, moet er een webserver beschikbaar zijn voor de student. De chatbot zal dan vanaf die server beschikbaar worden gemaakt. Daarnaast zal er met behulp van een camera, microfoon en scherm een setup moeten worden ontworpen om tot de laatste versie van de chatbot te komen. Dit zijn verder allemaal geen vereisten om het project mee te starten. Er moet ook eerst meer onderzoek worden gedaan voordat er zomaar wat gekocht wordt, om te voorkomen dat er wat onnodigs wordt besteld.

Oplevering en kwaliteitseisen

Capgemini

Vanuit Capgemini wordt er een demo applicatie verwacht. Deze applicatie moet aangeven wat een mogelijke toepassing zou kunnen zijn voor de chatbot technologie. Er zijn geen verdere kwaliteitseisen vastgesteld waar de chatbot aan moet voldoen. Er wordt juist vanuit de student verwacht dat er wordt aangegeven wat mogelijk is en wat niet mogelijk is en waarom.

De Haagse Hogeschool

Vanuit de Haagse Hogeschool wordt er de volgende zaken verwacht aan het einde van de stage.

- Goedgekeurde afstudeerplan
- Plan van aanpak
- Overige opgeleverde (tussen)producten, die benoemd zijn in het afstudeerplan
- Afstudeerverslag
- Wijze van aantonen van de beroepstaken
- Tussentijdse besluiten en beoordelingen
- Schriftelijke rapportage van de opdrachtgever c.q. bedrijfsmentor.

Ontwikkelmethoden

Tijdens het project zal er gebruik worden gemaakt van een iteratieve ontwikkelmethoden. Het project zal individueel uitgevoerd worden. Er is hierdoor niet voor een standaard scrum methode gekozen. De onderdelen van scrum die blijven staan is de sprint planning en sprint review en sprint retrospective. De sprint planning zal plaats vinden voor het pitch moment, zodat de planning ook direct gepresenteerd kan worden. De sprint review en retrospective worden tijdens de pitch door de student behandeld en met behulp van input van de begeleider ondersteund. Na deze pitch start ook direct de volgende sprint. De gemaakte planning wordt waar nodig nog aangepast met behulp van de gegeven feedback. De daily standup meeting zal samen met een groep studenten worden uitgevoerd. Niet alle studenten voeren een programmeerstage uit, maar er is besproken dat iedereen wel behoefte heeft aan een kort moment om te bespreken waar ze mee bezig zijn en of ze ergens tegenaan lopen. Op deze manier kunnen we elkaar helpen door mee te denken en feedback te geven op de bezigheden.

Projectorganisatie en communicatie

Vanuit Capgemini wordt het project in de eerste fase ondersteund door Peter-Paul Tonen. Dit zal na een aantal weken kunnen veranderen indien er specifiekere ondersteuning nodig is en dit beter door iemand anders kan worden gegeven. Vanuit de Haagse Hogeschool wordt de student ondersteund door Hani Al-Ers als stagebegeleider en door Ton

Biegstraaten als expert begeleider. Om de week is er een vast moment voor een pitch en gesprek met de begeleider vanuit Cag Gemini. Daarnaast is hij altijd bereikbaar voor vragen via de mail. Het gesprek dat om de week is gepland is in groepsverband. De groep bestaat uit 11 andere stagiaires die ieder hun eigen opdracht hebben. Iedereen heeft de tijd om een pitch te geven over de voortgang van het project en de problemen waar ze tegenop lopen. Hierna wordt er feedback gegeven op de pitch en krijgt de student advies over vervolgstappen om tot nieuwe oplossingen te komen. Na het pitchen is er een vast uur gepland waarin er tijd is om 1-op-1 gesprek aan te gaan met de begeleider om de opdracht en voortgang verder te bespreken indien nodig.

Planning

De volgende globale planning is gemaakt om een overzicht te geven van de bezigheden van de student. Er is een planning gemaakt voor acht sprints van twee weken plaats vinden. Dit komt overeen met de looptijd van de stage, waarbij de eerste week wordt gezien als een introductie week. Er is gekozen voor een iteratieve ontwikkelmethoden zullen ontwerpen, ontwikkelen en testen veelal samen worden uitgevoerd. Aan het einde van een sprint zal er met behulp van de gegeven feedback worden gekeken of er toch nog wat moet veranderen in het ontwerp.

Taak	Toelichting	Verwacht product
Plan van aanpak	Een inleidend verslag voor het project dat de student gaat uitvoeren. Hierin wordt een globale planning gemaakt.	Plan van aanpak
Aanvullende oriëntatie	Vervolgonderzoek waar nieuwe bevindingen in worden genoteerd, dat als input kan worden gebruikt voor het verdere proces.	Input voor het onderzoek doen en voor de verdere stage
Onderzoeken	Onderzoek dat bij elke sprint opnieuw zal worden uitgevoerd. In deze taak zal een document voortkomen waarin keuzes worden vastgelegd en beargumenteerd.	Documentatie en onderbouwing voor de gemaakte keuzes
Ontwerpen	Het ontwerpen van het project zal elke sprint weer plaats vinden. Indien er aanpassingen nodig zijn of er nieuwe bevindingen zijn gedaan kan het ontwerp worden aangepast.	Ontwerp voor het systeem en de samenwerking van de tools
Ontwikkelen	Deze taak houdt het daadwerkelijk programmeren in. Ook het opzetten van een omgeving en het uitwerken van de requirements zal in deze taak worden verwerkt.	De ontwikkelde tools die besproken zijn

		in het plan van aanpak
Testen	Tijdens deze taak zal ik vooral unittests opzetten die kunnen garanderen dat de chatbot goed genoeg werkt. Hiermee kan de chatbot gebruikt worden om te kijken of het antwoord dat gegeven wordt ook overeenkomt met wat ervan tevoren is bedacht.	De benodigde testen die de kwaliteit van de chatbot kunnen waarborgen.
Adviseren	Bij deze taak wordt er een kort advies document verwacht, waarin keuzes kort worden toegelicht en waarin wordt beschreven wat werkt en wat niet werkt. Dit kan dan bij vervolgonderzoek worden ingezien om niet dezelfde fouten te maken.	Advies document voor Capgemini waarin de bevindingen beschreven staan
Implementeren	Deze taak bevat de implementatie van de chatbot. Dit zal in de vorm van een beeldscherm worden gedaan met als input een camera en microfoon.	Het implementeren van de tool met hardware zodat het kan worden gebruikt in de welkomst

Tabel 1. Toelichting taken

Sprint	Plan van aanpak	Aanvullende oriëntatie	Onderzoeken	Ontwerpen	Ontwikkelen	Testen	Adviseren	Implementeren
1	X	X						
2	X	X	X	X	X			
3			X	X	X	X		
4			X	X	X	X		
5			X	X	X	X		
6					X	X	X	X
7						X	X	X
8							X	X

Tabel 2. Taken planning over de beschikbare sprints

Risico's

Uit het vooronderzoek is gebleken dat er geen risico's zijn gevonden die de doelstelling van het project kunnen voorkomen. In het geval dat het de student niet lukt om een werkende demo te ontwikkelen binnen 16 weken is dat ook een resultaat en kan dit worden gebruikt als input dat het ontwikkelen van een conversatie chatbot moeilijker is dan verwacht en er verder onderzoek gedaan moet worden.

Risico's	Wat is de impact	Hoe groot is de impact	Kans	Mogelijke maatregel
De benodigde gegevens die gebruikt worden als input voor de chatbot kunnen niet beschikbaar worden gemaakt	Sommige vragen kunnen niet worden geïmplementeerd en de chatbot zal minder uitgebreid worden	Middelmatig	Gemiddeld	In overleg met de opdrachtgever kan naar mogelijke andere manieren worden gekeken om toch nog aan de informatie te komen.
Vanwege privé redenen absentie student	De gemaakte planning zal moeten worden herzien en mogelijk aangepast. Dit kan voor een minder compleet eindproduct zorgen.	Groot	Klein	n.v.t.

Bijlage 3 – Testplan

Testplan prototype chatbot

In dit testplan wordt beschreven op welke manieren de werking van de chatbot wordt gewaarborgd. Er is beschreven welke kwaliteitsattributen van belang zijn en hoe deze worden gewaarborgd.

Test Objecten

De volgende onderdelen van de chatbot zullen getest worden

- Meteor Frontend
- Python Backend
- Dialogflow

Meteor Frontend

Er zal hier vooral de werking van het visuele gedeelte van de chatbot getest worden. Ook moet er gekeken worden of de functionaliteiten van het programma werken zoals deze ontwikkeld zijn. Dit zal vooral met behulp van usability tests worden uitgevoerd. Vanuit het belang van de opdrachtgever is hiervoor gekozen. Een usability test levert uiteindelijk feedback over de applicatie zelf. Hierdoor krijgt ook de opdrachtgever de zekerheid dat het is getest, voordat deze beschikbaar wordt gemaakt aan bezoekers. Het is niet mogelijk om met behulp van code erachter te komen of een applicatie gebruiksvriendelijk is of niet. Het plan is om minstens vijf medewerkers de applicatie te laten testen door koffie te bestellen. Achteraf heb ik hen om hun feedback gevraagd over de webapplicatie. Op deze manier wordt duidelijk wat veelvoorkomende problemen of verbeteringen zijn voor de applicatie. Deze kunnen dan geïmplementeerd worden in de chatbot.

Python Backend

Met behulp van unittests zal hier getest worden of het verwachte antwoord naar Dialogflow wordt gestuurd. De onderdelen die hier getest zullen worden zijn de NS-koppeling en de functie die de verschillende koffiesoorten terugstuurt naar Dialogflow. De code coverage wordt bepaald aan de hand van path coverage. Vanwege het aantal if statements en verdere simpliciteit van de code is dit de meest voor de hand liggende manier om code coverage te bepalen.

Dialogflow

Dialogflow bevat zelf geen mogelijkheden om de integriteit te testen van het bepalen van de intenties. Door zelf deze functionaliteit te ontwerpen kan er toch garantie worden geboden over de werking van Dialogflow. Er is ervoor gekozen om deze testen op eenzelfde manier als een unittest te ontwerpen. Elke test bestaat uit een verwachte intentie en een zin die naar Dialogflow wordt gestuurd. Nadat Dialogflow heeft gereageerd met de herkende intentie wordt deze vergeleken met wat er werd verwacht. Deze testen zullen overzichtelijk worden gemaakt in een visueel overzicht. Hierin kan de beheerder van de chatbot altijd nagegaan of de chatbot nog naar behoren werkt.

Integriteit chatbot

De integriteit van de chatbot kan op twee plekken worden getest. Deels hiervan heeft met herkennen van de goede intentie door Dialogflow. Omdat Dialogflow als service gehost wordt door Google, wordt deze constant geüpdatet in de achtergrond. Dit heeft als nadeel dat de verwachte uitkomst niet meer gelijk is aan de werkelijke uitkomst. Om dit te testen is er een omgeving ontwikkeld

in het Meteor frontend waarin te zien is of alle vooraf gemaakte testen slagen en de correcte intentie herkend wordt. Daarnaast wordt het Python backend gebruikt voor het verrijken van de antwoorden van de chatbot. Hier zal getest moeten worden of er geen onverwachte fouten in de code zitten en er altijd een antwoord wordt verstuurd zoals gepland.

[Dialoog testen methode](#)

De reden waarom ervoor is gekozen om de dialoog tester in Meteor te bouwen is de integratie van MongoDB. De testen kunnen structureel opgeslagen worden in de database en hier kan ook de geschiedenis van de resultaten in worden opgeslagen. Daarnaast maakt het gebruik van Meteor het ook mogelijk om een visueel overzicht te geven via een webapplicatie om zo overzichtelijk te maken hoe Dialogflow presteert.

Het is ook mogelijk om zonder de applicatie opnieuw te deployen een test aan te passen of toe te voegen. Deze mogelijkheid zorgt voor extra flexibiliteit.

Er zal extra aandacht moeten worden besteed aan de opbouw van een gesprek. Indien een test case bestaat uit de bestelling van een kleine koffie, voor Timo moet de juiste volgorde van de chatbot worden aangehouden. Indien eerst “voor Timo” wordt gestuurd naar Dialogflow zal hij niet de goede intent herkennen, omdat dit normaal pas na een bestelling wordt gezegd. Dialogflow werkt dus met een context waarin hij bepaalde antwoorden verwacht en hierdoor is het dus van belang dat de antwoorden sequentieel en in de goede volgorde worden verzonden.

Bijlage 4 – Test bevindingen

Test bevindingen prototype chatbot

Aan de hand van het opgestelde test plan, zijn bepaalde resultaten behaald. De bevindingen van de testen zijn in deze bijlage beschreven per testonderdeel.

Python Backend

Type behaalde coverage: Path coverage, ieder pad en iedere combinatie van paden is getest. Block coverage behaald: 83%

Uit de testen is gebleken dat niet alle mogelijkheden goed opgevangen werden. Zo liep het programma stuk als de bestelde drank niet in de opgeslagen lijst met dranken staat. Dankzij de test is de code aangepast en stuurt hij nu wel een correct antwoord terug naar Dialogflow.

Dankzij deze testen is de integriteit van het Python backend gegarandeerd. Met behulp van de testcases kan ervan uit worden gegaan dat er altijd een correct antwoord wordt verstuurd door het backend.

Usability testen

De bevindingen van de usability test worden hieronder genoemd. Per gebruiker is puntsgewijs benoemd hoe zij het gebruik hebben ervaren van de webapplicatie.

De cijfers hieronder gegeven, zijn met een schaal van 1 op 5. Waarbij 1 het laagste cijfer is en 5 het hoogste. Per gebruiker is een score gevraagd op basis van

- Interface, tot in hoeverre het userinterface intuïtief is.
- Gebruik, tot in hoeverre het gebruik van de applicatie soepel verloopt.
- Aanbeveling, tot in hoeverre de gebruiker de applicatie zou aanbevelen aan anderen.
- Performance, beoordeling over de snelheid van het gebruik van de applicatie.

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
Interface	4	2	4	5	5	4	4	4	3	4
Gebruik	4	4	4	5	3	5	4	4	3	3
Aanbevelen	5	3	5	4	4	4	4	5	4	3
Performance	5	5	4	5	4	5	5	5	5	4

Tabel 1 resultaten usability test

Aanvullende opmerkingen buiten meetwaarden voor usability test.

Gebruiker 1: Beperkte mogelijkheden

Gebruiker 2: Interface kan wat meer kleur krijgen

Gebruiker 3: Bij typefouten altijd opnieuw invoeren nadeel

Gebruiker 5: Vaak dezelfde melding als de chatbot het niet begrijpt

Gebruiker 7: Mooi prototype om de functionaliteiten van een chatbot te visualiseren

Gebruiker 8: Door Capgemini kleuren toe te voegen, zal het beter passen binnen het bedrijf.

Gebruiker 10: Niet helemaal duidelijk waar de koffie gehaald kan worden na het gesprek

Deze feedback is verwerkt tot de volgende taken:

- Verbeteren van de interface door meer kleur en Capgemini huisstijl toe te voegen.
- Chatbot responses uitbreiden bij Dialogflow met meer afwisseling en betere toelichting.

Ook is eruit op te maken dat er niks verandert hoeft te worden in de performance van de applicatie. Iedereen heeft dit aspect van de applicatie als positief beoordeeld.

Dialogflow dialoog testen

Het is gelukt om een interface te ontwikkelen dat gebruikt kon worden voor het testen van verschillende interfaces. In figuur 1 hieronder is te zien hoe de interface de dialoogtesten overzichtelijk maakt. Deze functionaliteit kan nog verder uitgebreid worden met functies zoals periodieke testen en een notificatie systeem.

De huidige uitwerking zorgt voor een zekerheid bij de gesprekken die zijn uitgewerkt als test case.

Aantal zinnen	Beschrijving	Zinnen	Laatst succesvolle test	Geslaagd	Test nu
4	Medium koffie voor Timo	Ik lust wat te drinken, Ik wil graag koffie, Een medium, Voor Timo	31-5-2019 19:26:30	Geslaagd	Klik hier

Figuur 1 overzicht dialoog test

Conclusie

Door de behaalde code coverage van het Python backend kan ervan uit worden gegaan dat er altijd een correcte response wordt gegeven. De dialoog testen zorgen voor integriteit bij het herkennen van de intenties door Dialogflow. Er kunnen verschillende scenario's getest worden, die overeenkomen met werkelijke gesprekken die bezoekers met de chatbot kunnen houden.