Opdrachtomschrijving

Rick van Schijndel  
Open Innovation minor

**Geschiedenis**

|  |  |  |
| --- | --- | --- |
| Datum | Auteur | Wijzigingen |
| 11 september 2019 | Rick van Schijndel | Eerste opzet, context |
| 12 september 2019 | Rick van Schijndel | Specifieke beschrijving |
| 16 september 2019 | Rick van Schijndel | Projectfocus, tooling |
| 26 + 27 september 2019 | Rick van Schijndel | Samenvoegen opdrachtomschrijving + projectplan |
| 30 september 2019 | Rick van Schijndel | Uitbreiding deelvragen |
| 1+2 oktober 2019 | Rick van Schijndel | Uitbreiding deelvragen |
| 21 oktober 2019 | Rick van Schijndel | Uitwerking programmeertaal onderzoek |
| 29 oktober 2019 | Rick van Schijndel | Gebruikersanalyse |

Inhoudsopgave

[Context 3](#_Toc23928249)

[Project 4](#_Toc23928250)

[Over Interactive Ellie 4](#_Toc23928251)

[Waarom? 4](#_Toc23928252)

[Waarom een chatbot? 4](#_Toc23928253)

[Waarom een persoonlijkheid? 4](#_Toc23928254)

[Specifieke beschrijving 4](#_Toc23928255)

[Complexiteit 5](#_Toc23928256)

[Functionaliteiten 5](#_Toc23928257)

[Projectfocus 6](#_Toc23928258)

[Tooling 7](#_Toc23928259)

[Globale planning 8](#_Toc23928260)

[Lijst van producten 8](#_Toc23928261)

[Onderzoeksvragen 9](#_Toc23928262)

[**Hoofdvraag**: “Op welke manier draagt een chatbot/AI-systeem met persoonlijkheid en emoties mee aan de interactie/ervaring bij het reserveren van een vergaderruimte?” 9](#_Toc23928263)

[**Deelvraag**: “Wat voegt een primitieve chatbot toe aan een bestaande (web)applicatie?” 10](#_Toc23928264)

[**Deelvraag**: “Waarom worden chatbots vaak een positieve (statische) persoonlijkheid gegeven?” 12](#_Toc23928265)

[**Deelvraag**: “Welke functionaliteiten verwachten gebruikers van het te realiseren AI-systeem?” 13](#_Toc23928266)

[**Deelvraag**: “Wat voor bijdrage levert een complex AI-systeem/chatbot aan een interactief gebouw?” 14](#_Toc23928267)

[**Deelvraag**: “Wat is de meest geschikte back-end programmeertaal voor het ontwikkelen van een chatbot/complex AI-systeem?” 15](#_Toc23928268)

[Vergelijkingstabel 15](#_Toc23928269)

[**Deelvraag**: “Welke mogelijkheden zijn beschikbaar voor het ontwikkelen van een front-end bij mijn chatbot/complex AI-systeem en zijn hiervoor het meest geschikt?” 17](#_Toc23928270)

[**Deelvraag**: “Welke karaktereigenschappen en emoties kan ik toepassen in mijn chatbot applicatie?” 18](#_Toc23928271)

# Context

Voor de Open Innovation minor (semester 7, richting Software Engineering) werken studenten zelfstandig of in een groep aan een project. De op te leveren producten, betrokken tooling/technieken en planning is geheel te bepalen door de studenten – mits er hier een externe product owner of stakeholder bij betrokken wordt met overzicht op het gehele traject.

In dit document beschrijf ik gedetailleerd de aard van mijn project, werkmethodiek, tooling en het op te leveren eindproduct.

De Open Innovation minor loopt van 2 september 2019 tot begin februari (ong. 20 weken).

# Project

Mijn project valt binnen het lectoraat Interaction Design, gebaseerd op ideeën en theorie van “Interactive Ellie”.

## Over Interactive Ellie

Interactive Ellie is een collectieve term/thema dat gebruikt wordt voor het ontwikkelen en brainstormen van technologie dat reageert op gebruikers met emoties en persoonlijkheid; het visualiseren van deze emoties kan bijvoorbeeld m.b.v. gekleurd licht en geluidspitch. In mijn project focus ik voornamelijk op het AI-aspect van dit thema; een chatbot.

## Waarom?

De aard van dit project brengt twee belangrijke vragen naar boven:

* Wat is het nut van een chatbot (in tegenstelling tot b.v. een standaard webapplicatie met een FAQ-sectie)?
* Wat is het nut van emoties en persoonlijkheid toevoegen aan een chatbot?

### Waarom een chatbot?

Chatbots worden langzaam steeds meer populair voor complexe websites. Het geeft ontwikkelaars en personeel de mogelijkheid om antwoord te geven op specifieke vragen zonder dat de gebruiker hier zelf een antwoord voor hoeft te zoeken. Daarnaast voelt deze oplossing veel natuurlijker voor vele mensen (wanneer je een product zoekt in de winkel dan “vraag je het wel even”). Sommige websites hebben zulke uitgebreide chatbots dat ze zelfs niet eens meer expliciet vermelden dat je met een bot aan het praten bent.

### Waarom een persoonlijkheid?

De kernfocus van dit project is de toevoeging van een (dynamische) persoonlijkheid en emoties aan het systeem.

#### Mirrored personality

Een standaardpraktijk dat gebruikt wordt bij het ontwikkelen van chatbots (met een **statische** persoonlijkheid) is om deze te baseren op de algemene doelgroep van de website. Een pizza-bestel website gericht op kinderen/jongvolwassenen zal bijvoorbeeld niet veel formeel taalgebruik toepassen. Op een feestje zul je in het algemeen meer interesses delen met leeftijdsgenoten dan oudere mensen. Mensen voelen zich comfortabeler wanneer interacteren met een vergelijkbare partij.  
In het geval van mijn systeem zal ik de persoonlijkheid dynamisch opstellen zodat deze zich baseert op de input van de gebruiker.

#### Experimenteel

Het Interactive Ellie researchthema richt zich ook op het simpelweg experimenteren/exploreren van mogelijkheden. Hoe reageren gebruikers wanneer de hardware/software die ze al die jaren als emotieloos hebben beschouwen plotseling een persoonlijkheid tonen? Wat kunnen ontwikkelaars hiervan leren? Is dit een functionaliteit die we bij meer applicaties moeten implementeren?

## Specifieke beschrijving

In o.a. de campussen en het InnovatieLab (Strijp-T, Eindhoven) van Fontys bevinden zich reserveerbare lokalen/vergaderkamers met tablets op deze ruimtes te reserveren, voor zowel studenten als docenten. Interactie met deze tablets is op het huidige moment minimaal met voor studenten geen mogelijkheid om een reservatie in te plannen voor later op de dag of een duratie langer dan 60 minuten.

Mijn projectvoorstel is om deze reserveerbare kamers te gebruiken als basis voor een chatbot/AI. Dit systeem zou kunnen worden toegevoegd aan de bestaande tablets of deze geheel vervangen.

### Complexiteit

De minimale verwachtingen van een standaard, primitieve chatbot is dat deze commando’s herkent en hier een vooraf ingesteld antwoord op kan geven – verdere complexiteit is gewenst maar niet vereist.  
Voor mijn systeem wil ik echter wel de volgende complexiteit implementeren:

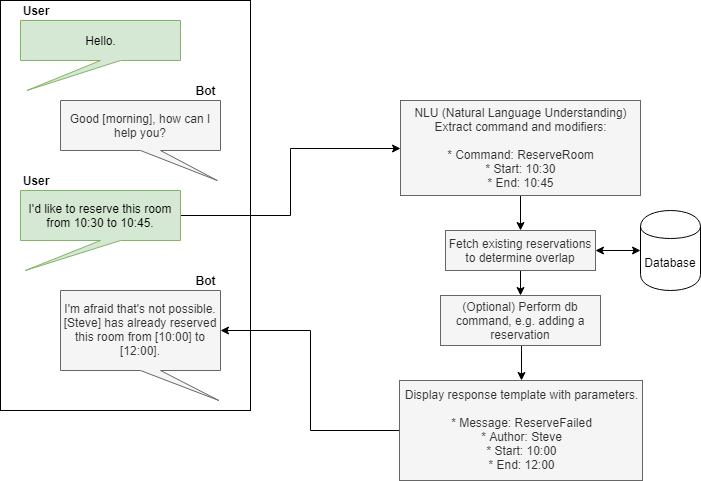
* Het systeem bevat een persoonlijkheid gebaseerd op een aantal emoties (woede, tevredenheid, verlegenheid) met ieder verschillende waarden. Deze worden vervolgens gebruikt als basis voor de reacties die het systeem kan geven op commando’s (zoals met veel tegenzin een taak uitvoeren wanneer woede een hoge waarde bevat).
* De bovenstaande persoonlijkheid is dynamisch en wordt steeds een klein beetje bijgewerkt met iedere gebruikersinteractie. Wanneer gedetecteerd wordt dat een gebruiker gefrustreerd is, is het gewenst (en zelfs menselijk herkenbaar) dat het systeem ook gefrustreerd raakt.
* Een “debug”-modus waarmee ontwikkelaars en eventueel zelfs eindgebruikers eenvoudig kunnen inzien waarom het systeem een bepaalde reactie geeft of emotie uit. In deze modus zal het AI-systeem geen persoonlijkheid of emoties uiten – in plaats daarvan geeft het systeem directe antwoorden en kunnen emoties direct bijgesteld worden. Wegens de experimentele staat van het project maakt een debug-modus het makkelijk voor mij om het systeem en de invloed van gebruikersinteractie te meten en beheren.
  + Om de rol van deze modus beter te kunnen toelichten volgt er hieronder een voorbeeld:
  + 1. Gebruiker vraagt op een rustige wijze of er een kamer beschikbaar is.
  + 2. Systeem antwoord op een gefrustreerde wijze.
  + 3. Gebruiker spreekt een specifiek commando om het systeem onmiddellijk om te schakelen naar debug-modus.
  + 4. Gebruiker vraagt om het meest recente antwoord te verklaren.
  + 5. Systeem weergeeft het herkende commando (kamer-beschikbaarheid controleren) en de huidige dominante emotie (woede/frustratie).

### Functionaliteiten

Het te realiseren AI-systeem zou de volgende kernfunctionaliteiten bevatten:

* Checken of een kamer beschikbaar is
* Deze kamer reserveren (met starttijd en eindtijd)
* Een reservering annuleren

Hieronder volgt een diagram dat de standaard werkwijze van mijn AI-systeem visualiseert.



## Projectfocus

Het project valt op te splitsen in de verschillende categorieën;

* **Software**: Ontwikkelen van een primitieve chatbot.
* **Software**: Implementeren van emoties en persoonlijkheid.
* **Hardware**: Koppeling aan de bestaande (of geheel nieuwe) tablets/dergelijke hardware die geplaatst kan worden bij vergaderkamers.
* **Interactie**: Verzamelen en beheren van data resulterend uit interacties tussen gebruikers en mijn systeem. Deze data vervolgens gebruiken als basis van verdere updates.

Mijn persoonlijke prioriteit binnen dit project zijn de twee software-gerelateerde categorieën. Ik wil voornamelijk in dit project uitzoeken wat de standaard “practices” zijn voor het ontwikkelen van een chatbot, welke programmeertalen en tooling hiervoor het meest geschikt is en uiteindelijk een chatbot AI te realiseren met persoonlijkheid.  
De andere categorieën wil ik ook graag aan bod laten komen in mijn project – maar met een lagere prioriteit, voornamelijk omdat het **hardware**-aspect een losstaand project zou kunnen zijn (de reserveertablets zijn een voorbeeld van geschikte hardware) en het **interactie**-aspect merendeels geautomatiseerd kan worden met een goed-geïmplementeerde dynamische persoonlijkheid in de software.

De daadwerkelijke implementatie van mijn AI-systeem op Strijp TQ (of hoofdcampus Rachelsmolen/Stappegoor) valt niet binnen mijn project.

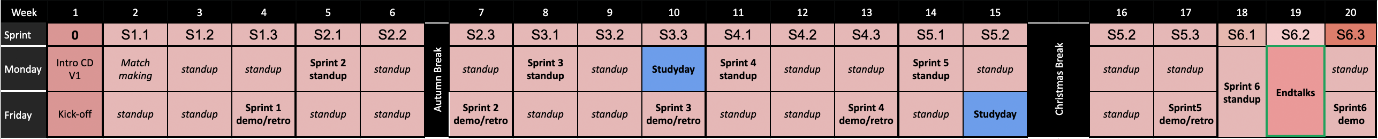
## Tooling

De meest geschikte programmeertalen voor het coderen van een chatbot/complex AI-systeem zijn C++, Java en Python. ([bron](https://www.oodlestechnologies.com/blogs/4-Programming-Languages-You-Can-Use-to-Create-Chatbots/))  
Deze programmeertalen hebben ieder hun eigen selectie aan geschikte IDE’s (Integrated Development Environments). Ik zal later nog een klein onderzoek doen naar de beste taal voor dit project.  
Alle software kan ik zelf ontwikkelen op mijn werklaptop (Windows).

Op een later stadium van het project zal ik (in overleg met een Technology-deskundige/student) kijken naar geschikte hardware om het reserveringsysteem op te zetten.

# Globale planning

In Open Innovation (S7) wordt ieder project opgedeeld in sprints – met voor iedere sprint een gemiddelde duratie van drie weken en een totale hoeveelheid van zes sprints.



De indeling van de werkzaamheden tijdens deze sprints is geheel aan mij, zolang ik er actief voor probeer te zorgen dat ik aan het einde van iedere sprint een presentabel “eindproduct” oplever (prototype, onderzoeksdocument, etc).

Zelf prefereer ik een Scrum-werkwijze tijdens dit project en wil ik dus niet van tevoren alle taken al per sprint indelen. Wel kijk ik aan het begin van iedere sprint (gebaseerd op contact met product owners en stakeholders) naar de prioriteit van iedere openliggende taak en baseer ik hierop een lijst van taken voor de huidige sprint.

Op dit moment zie ik in ieder geval de volgende producten aan bod komen in dit project:

## Lijst van producten

* Hoofdproduct; de applicatie/chatbot dat ik realiseer in dit project.
* Prototypes, zowel functioneel als niet-functioneel.
* Testsessies, waarmee ik gebruikersinteractie meet en meeneem als feedback naar de volgende productiteratie – met een mogelijke gebruikersanalyse.
* Requirementsanalyse, voor zowel software requirements als hardware criteria.
* Acceptatietestplan en testrapportage voor **iedere** sprint met een bruikbaar prototype voor het AI-systeem.
* Advies/integratieplan, waarin ik duidelijk de software/hardware criteria van de requirementsanalyse meeneem en een aanbeveling doe betreft integratie. Inclusief gebruikte frameworks/functionaliteit en mogelijke alternatieven.
* Kort ontwerpdocument dat de functionaliteit van mijn applicatie beschrijft en deze opsplitst in verschillende, afgebakende lagen.

# Onderzoeksvragen

De aard van het project is om te ontdekken welke mogelijkheden er zijn binnen de ICT voor het toevoegen van persoonlijkheid en emoties aan een moderne applicatie en wat hiervan de meerwaarde is. Echter is de implementatie van een chatbot een iets bekender concept; daarom kan ik het onderzoek in twee delen opsplitsen:

* Wat is de meerwaarde van een chatbot/AI?
  + Dit is een deductief onderzoek. Vele websites en webapplicaties hebben in recente jaren gekozen voor de toevoeging van een chatbot, voornamelijk om website bezoekers antwoord te geven op veel gestelde vragen. Het heeft duidelijk een positieve invloed, maar waarom?
* Wat is de meerwaarde van **dynamische** persoonlijkheid en emoties in een chatbot/AI?
  + Dit is een inductief onderzoek en een lastige vraag om te beantwoorden; het praktisch nut van een statische persoonlijkheid in chatbots is prima te onderzoeken – helaas is een dynamische persoonlijkheid lastig te implementeren en (of juist daarom) een zelden voorkomende eigenschap in chatbots. Het is aan mij om goed door te nemen waarom het researchthema Interactive Ellie van start is gegaan en welke vindingen hieruit zijn gekomen.
  + Dit deel van mijn onderzoek zal in veel deelvragen uitbranchen:

## **Hoofdvraag**: “Op welke manier draagt een chatbot/AI-systeem met persoonlijkheid en emoties mee aan de interactie/ervaring bij het reserveren van een vergaderruimte?”

De hoofdvraag van dit onderzoek luidt als volgt:

“Op welke manier draagt een chatbot/AI-systeem met persoonlijkheid en emoties mee aan de interactie/ervaring bij het reserveren van een vergaderruimte?”

Met behulp van deze algemene hoofdvraag kan ik kijken naar de meerwaarde van een chatbot, de meerwaarde en voordelen/nadelen van een dynamische persoonlijkheid tegenover een statische persoonlijkheid en een eventuele aanbeveling doen voor sommige bedrijven (gebaseerd op complexiteit en budget).

## **Deelvraag**: “Wat voegt een primitieve chatbot toe aan een bestaande (web)applicatie?”

Steeds meer moderne bedrijven implementeren chatbots in hun professionele website; deze chatbots helpen websitebezoekers met het antwoorden van veelgestelde vragen en kunnen simpele taken uitvoeren zoals het helpen bestellen van een product, traceren van een pakket of een bestelling annuleren.

Het implementeren van een primitieve chatbot heeft een aantal voordelen, zoals:

* **Chatbots kunnen simpele, repeterende taken opvangen zoals standaardvragen beantwoorden en helpen met simpele processen zoals het annuleren van een bestelling.**
  + Chatbots herkennen sleutelwoorden in vragen die een gebruiker stelt om op basis hiervan het juiste commando te herkennen.
  + Het antwoorden van veelgestelde vragen zou voor een daadwerkelijke medewerker gezien worden als saai en repetitief. Bij een chatbot wordt dit juist een voordeel; des te vaker een vraag gesteld wordt, des te hoger de prioriteit om deze te laten herkennen door een chatbot.
* **Chatbots zijn goedkoper dan medewerkers.**
  + Uiteraard is de daadwerkelijke kostenberekening iets ingewikkelder; het kost tijd en geld voor ontwikkelaars om de chatbot naar wens te realiseren. De onderhoudskosten zijn echter veel lager dat het betalen van een standaard salaris aan een klant supportmedewerker.
  + Ze maken niet per sé klant supportmedewerkers obsoleet. Er zou bij de chatbot altijd een commando of knop beschikbaar kunnen zijn dat het gesprek overschakelt naar een medewerker – voor als de websitebezoeker graag contact wil met een daadwerkelijk persoon voor bijvoorbeeld het afvangen van een lastige/minder vaak voorkomende vraag.
  + Waarschijnlijk is de meest kost besparende factor het feit dat medewerkers gelimiteerd zijn aan een gesprek tegelijk (hooguit een paar tegelijk). Voor de chatbot is dit limiet niet aanwezig.
* **Chatbots zijn theoretisch altijd beschikbaar.**
  + Zolang de website 24/7 beschikbaar is bestaan er weinig redenen voor de chatbot om niet ook 24/7 beschikbaar te zijn. De chatbot hoeft niet in de avond naar huis om NOS Sport te kijken.
  + Echter zijn er wel degelijk een aantal websites die ervoor kiezen om chatbots alleen beschikbaar te stellen tijdens standaard werkuren. Dit kan bijvoorbeeld zijn omdat:
    - Gesprekken tussen websitebezoekers en chatbots worden bijgehouden door werknemers die eventueel kunnen “ingrijpen” wanneer de chatbot iets niet begrijpt of de bezoeker vraagt om een persoon.  
      *“Zodra we iets geks zien kan ik meteen het gesprek overnemen.”*, aldus **Lars Janssen**, student en chatbot ontwikkelaar.
    - Sommige websites gaan zelfs zo ver dat ze de chatbot maskeren als een beschikbare klantsupportmedewerker. Nergens staat vermeld dat de communicatie plaatsvindt tussen de bezoeker en een AI.

**Gebruikte onderzoeksmethodes**:

* Community Research (LIBRARY)
* Expert Interview (LIBRARY)
* Problem Analysis (FIELD)

**Bronnen**:

* *“7 Benefits Of Using Chatbots For Your Business”*
  + <https://yellow.systems/blog/7-benefits-of-using-chatbots-for-your-business>
* *“What Is a Chatbot and How Is It Changing Customer Experience?”*
  + <https://www.salesforce.com/blog/2019/04/what-is-a-chatbot.html>
* *“Would You Interact with a Chatbot that's Unfriendly?”*
  + <https://www.entrepreneur.com/article/339248>

## **Deelvraag**: “Waarom worden chatbots vaak een positieve (statische) persoonlijkheid gegeven?”

Het is mogelijk om chatbots zo statisch en neutraal mogelijk te ontwikkelen, maar veel bedrijven geven juist de voorkeur aan een chatbot met een zeer positieve, vriendelijke persoonlijkheid. Maar waarom eigenlijk? Als gebruikers weten dat ze met een AI in gesprek gaan, willen ze dan niet liever een zo direct mogelijk antwoord op hun vragen?

* Net als vriendelijke werknemers vinden gebruikers het prettig om met een vriendelijk AI te communiceren. In dit aspect heeft de chatbot dan ook het voordeel dat deze nooit een slecht humeur zal uiten (ervan uitgaande dat de ontwikkelaars voor een zeer positieve persoonlijkheid hebben gekozen).
* Er is geen concrete, “juiste” aanpak op dit gebied van chatbot ontwikkeling. De discussie betreft “Friendliness VS Functionality” op het gebied van AI is al jaren bezig; er zijn redenen om AI vriendelijk te maken net zoals er redenen zijn om AI juist zo direct mogelijk te laten antwoorden. [Dit artikel](https://chatbotsmagazine.com/how-do-you-like-your-chatbots-functional-or-friendly-8066daae43cf) beschrijft een aantal tegensprekende argumenten over exact deze functionaliteit.

**Gebruikte onderzoeksmethodes**:

* Best, good and bad practices (LIBRARY)
* Community research (LIBRARY)
* Literature study (LIBRARY)

**Gebruikte bronnen**:

* *“How do you like your Chatbots. Functional or friendly?”*
  + <https://chatbotsmagazine.com/how-do-you-like-your-chatbots-functional-or-friendly-8066daae43cf>
* *“9 of the Most User-Friendly Chatbots Online”*
  + <https://chatbotslife.com/9-of-the-most-user-friendly-chatbots-online-5afe24195361>

## **Deelvraag**: “Welke functionaliteiten verwachten gebruikers van het te realiseren AI-systeem?”

Om te achterhalen welke functionele en niet-functionele eisen de uiteindelijke doelgroep van mijn reserveringssysteem verwachten heb ik een soort gebruikersanalyse+requirementsanalyse uitgevoerd met een testversie/prototype van mijn chatbot, gerealiseerd in DialogFlow. Uit deze testsessies heb ik geconcludeerd dat ik op het goede pad zit richting de huidige geplande commando’s en uitspraken.

Voor meer uitgebreide informatie over deze onderzoeksvraag, zie het document “[*Gebruikersanalyse*](Gebruikersanalyse.docx)*“*.

Voor latere iteraties van mijn chatbot applicatie zou ik DialogFlow kunnen gebruiken als framework.

## **Deelvraag**: “Wat voor bijdrage levert een complex AI-systeem/chatbot aan een interactief gebouw?”

Een zeer lastige deelvraag om zelfstandig te beantwoorden – daarom heb ik dit deel van het onderzoek als laatste uitgevoerd met behulp van veel (en uitgebreide) user tests.

Tijdens user tests heb ik diverse personen gevraagd om mijn chatbot te gebruiken voor het reserveren van een kamer en uitvinden van andere mogelijke functionaliteit. Zelf probeer ik ze hierbij zo min mogelijk te sturen.

De opzet, uitvoering en conclusie van deze tests zijn genoteerd in het document [*“Testplan”*](Testplan.docx).

## **Deelvraag**: “Wat is de meest geschikte back-end programmeertaal voor het ontwikkelen van een chatbot/complex AI-systeem?”

Er is geen absoluut specifieke programmeertaal dat vooruitloopt in de ontwikkeling van chatbot/AI-systemen. Echter zijn er wel – voornamelijk wegens de stijgende populariteit in chatbots – vele online discussies over precies dit onderwerp.

De meest populaire back-end programmeertalen voor de ontwikkeling van chatbots zijn:

* Python
* Java
* C#
* PHP
* Ruby

Uit deze talen heb ik al eerdere ervaring opgedaan in Java, C# en PHP. Wegens de verdeling tussen back-end en front-end bij deze applicatie bevat PHP geen voordeel over de andere talen betreft back-end.

Hieronder vergelijk ik de verschillende mogelijke programmeertalen in een tabel met per categorie een score (1-10). De taal met de hoogste totale score is dus volgens mijn eigen criteria de meest geschikte taal voor deze back-end.

### Vergelijkingstabel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Python | Java | C# | PHP | Ruby |
| Persoonlijke ervaring | 2 | 9 | 8 | 7 | 5 |
| Support voor chatbot frameworks | 10 (REST) | 10 (REST) | 10 (REST) | 10 (REST) | 9 (REST via third party client) |
| Database compatibiliteit | 9 | 9 | 9 | 10 | 8 |
| OOP | 8 | 10 | 10 | 7 | 9 |
| IDE support | Pycharm, Spyder | Intellij, NetBbeans, Eclipse | Visual Studio, Rider | PHPStorm, NetBeans | VS Code, RubyMine |
| Messaging service/REST | 8 (REST) | 9 (REST + JMS) | 8 (REST) | 9 | 9 |

Met een totale score van 47 punten lijkt Java voor nu de meest passende programmeertaal. Uiteraard kan ik altijd nog kiezen voor een alternatief wanneer blijkt dat ik tegen verschillende problemen aanloop in Java (zoals databasekoppeling of compatibiliteit met een specifiek NLA framework).

**Gebruikte onderzoeksmethodes:**

* Expert Interview (LIBRARY)
* Available Product Analysis (LIBRARY)
* Multi-criteria Decision Making (WORKSHOP)

**Gebruikte bronnen:**

* *“Chatbot Development Frameworks”*
  + <https://geekflare.com/chatbot-development-frameworks/>
* *“6 Programming Languages to choose from to build an AI chatbot”*
  + <https://analyticsindiamag.com/6-programming-languages-to-choose-from-to-build-ai-chatbot/>
* *“Which are the best programming languages to build an AI chatbot”*
  + <https://chatbotnewsdaily.com/which-are-the-best-programming-languages-to-build-an-ai-chatbot-39eccf703aec>
* *“What programming languages are used to build chatbots?”*
  + <https://www.quora.com/What-programming-languages-are-used-to-create-chatbots>

## **Deelvraag**: “Welke mogelijkheden zijn beschikbaar voor het ontwikkelen van een front-end bij mijn chatbot/complex AI-systeem en zijn hiervoor het meest geschikt?”

Wegens de scheiding tussen front-end en back-end systeem in mijn eindapplicatie staat de selectie in front-end niet meteen vast na het kiezen van een geschikt back-end systeem – deze zal wel invloed hebben maar omdat de back-end alleen maar losstaande REST calls verwacht kan het front-end in praktisch iedere programmeertaal gerealiseerd worden.

Om deze reden wil ik gebruik maken van een simpel Javascript framework voor het opstellen van een of twee webpagina’s. Meer verwacht ik niet nodig te hebben in de scope van dit project.

Populaire Javascript frameworks voor het opzetten van een kleine webapplicatie zijn:

* React
* Angular
* Ember
* Vue
* Backbone

Uit deze lijst staan React en Angular zeer bekend om hun flexibiliteit en schaalbaarheid bij grote (enterprise-)applicaties. Vue’s reputatie daarentegen is gebaseerd op simpliciteit en het gebruik van standaard Javascript (in tegenstelling tot bijvoorbeeld Typescript of JSX). In tegenstelling tot mijn back-end analyse kan ik hierbij geen eenvoudige vergelijkingsmatrix opzetten tussen deze Javascript frameworks; ze zijn allemaal geschikt en de keuze hieruit hangt bijna geheel af van de developer zijn/haar persoonlijke voorkeur.

Omdat ik al eerder ervaring op heb gedaan in Vue tijdens proftaak projecten en deze het meest geschikt lijkt voor de simpele aard van mijn front-end kies ik bij deze voor Vue. Deze keuze kan ik echter altijd nog wijzigen bij latere toegevoegde complexiteit aan de eisen van mijn front-end – maar voor nu lijkt het me slim om (voornamelijk omdat front-end maar een klein aspect is in dit project) geen gebruik te maken voor geheel nieuwe tools/frameworks waar ik nog nooit eerder mee heb gewerkt. Wel is het goed om bij deze achterhaald te hebben welke alternatieven frameworks beschikbaar zijn.

**Gebruikte onderzoeksmethodes:**

* Available product analysis (LIBRARY)
* Community research (LIBRARY)

**Gebruikte bronnen:**

* *“The Best JS Frameworks for Front-End”*
  + <https://rubygarage.org/blog/best-javascript-frameworks-for-front-end>
* *“Comparing Angular vs Vue”*
  + <https://buttercms.com/blog/comparing-angular-vs-vue>
* *“React vs Vue: Which is better for 2020?”*
  + <https://towardsdatascience.com/react-vs-vue-which-is-better-for-2020-c484f22c67a8>

## **Deelvraag**: “Welke karaktereigenschappen en emoties kan ik toepassen in mijn chatbot applicatie?”

Een zeer groot aspect in mijn project is de implementatie van emoties en/of karaktereigenschappen bij mijn chatbot/AI-systeem. Het doel van dit project is namelijk om te achterhalen welke invloed zo’n implementatie heeft op gebruikers, hoe zij hier op reageren en of deze technologie meerwaarde bevat voor toekomstige, vergelijkbare projecten.

Om dit onderzoek te voltooien zal ik dus moeten bepalen **welke** eigenschappen de chatbot bezit, **hoe** de chatbot deze eigenschappen toont in zijn/haar gedrag en reacties en **hoe** deze eigenschappen dynamisch worden beïnvloed door de gebruiker zelf. Bijvoorbeeld:

* **Welke** eigenschap? De chatbot is verlegen.
* **Hoe** wordt dit getoond? De chatbot geeft korte antwoorden en start nooit gesprekken uit zichzelf.
* **Hoe** wordt deze eigenschap beïnvloedt? De gebruiker is zeer vriendelijk en geeft lange antwoorden. De chatbot vangt dit op en begint zichzelf ook meer open te stellen met langere antwoorden en vriendelijke opmerkingen.

De uitwerking van deze onderzoeksvraag is te vinden in het document “[*Karaktereigenschappen*](Karaktereigenschappen.docx)”.