

Afstudeerverslag



Automatiseren classificatie documenten - Sander Groot Wesseldijk

Versie 2.0

Hogeschool	Saxion Deventer
Opleiding	HBO-ICT
Bedrijf	Topicus Finance
Begeleider	Rogier Hommels
Begeleider – Topicus 1	Yoni Meijberg
Begeleider – Topicus 2	Barthold Derlagen
Kwartiel	4.3 & 4.4
Datum	17-06-2019
Plaats	Deventer

Auteur

Sander Groot Wesseldijk

SanderGrootWesseldijk@gmail.com

Student HBO-ICT Software Engineering

Betrokkenen

Rogier Hommels

r.m.hommels@saxion.nl

Saxion

Yoni Meijberg

Yoni.Mijberg@topicus.nl

Topicus

Barthold Derlagen

Barthold.Derlagen@topicus.nl

Topicus

Versiebeheer

Versie	Auteurs	Datum	Status	Opmerking
0.1	Sander Groot Wesseldijk	27-02-2019	Concept	Opzet afstudeerverslag
0.2	Sander Groot Wesseldijk	24-03-2019	Concept	Toevoegen informatie betreft planning en onderzoek
0.3	Sander Groot Wesseldijk	24-05-2019	Concept	Toevoegen van technische documentatie
0.4	Sander Groot Wesseldijk	31-05-2019	Concept	Beschrijving van het ontwikkelproces. Beschrijving van design keuzes Overzicht applicatie toegevoegd Kwaliteitsbewaking Product conclusie Spellingscontrole
1.0	Sander Groot Wesseldijk	02-06-2019	Eerste versie	Verwerking commentaar Rogier
1.1	Sander Groot Wesseldijk	05-06-2019	Eerste versie	Verwerken commentaar Jan Stroet: • Verbeteren spelfouten • Hoofdstuk indeling veranderd • Herschrijven van retrospective • Scrapen uitgebreide toegelicht • Vergelijking service en scrapen uitgebreid • Benadering van ethiek m.b.t. scrapen
1.2	Sander Groot Wesseldijk	12-06-2019	Eerste versie	Verwerken commentaar Jan Stroet: • Berekening kosten scrapen herschreven • Proof of concept ongesplitst in beschrijving en technische documentatie • Requirements verplaatst naar de beschrijving van het proof of concept • Tooling en framework keuzes herschreven • Dataverkrijgingsmethodiek vergelijking uitgebreider benaderd in "Wat zijn de beste dataelementen om te automatiseren" • Technische documentatie geherstructureerd

2.0	Sander Groot Wesseldijk	17-06-2019	Tweede versie	Verwerken commentaar Topicus begeleiders: • Toevoegen afbeelding “verbinding twee applicaties” • Uitgebreidere onderbouwing voor het uitzoeken van een scrape library
------------	-------------------------------	------------	---------------	---

Terminologie

Index	Begrip	Omschrijving	Pagina eerste voorkomen
1	Financieel advies applicatie	Een applicatie die de consument en de adviseur ondersteund bij het hypotheek advies	7
2	HDN	Hypotheek datanetwerk, een framework dat basisprincipes afdwingt.	7
3	Findesk	Financieel advies applicatie van Topicus	7
4	Breaking changes	Een aanpassing op softwaresysteem dat potentieel andere systemen kan laten falen.	18
5	CORS	Cross-origin resource sharing, een mechanisme dat bepaal welke resources van een webpagina benaderd mogen worden vanaf een ander domein	27
6	E-meeting	Een meeting gevoerd over een digitaal medium	42

INHOUDSOPGAVE

Samenvatting.....	1
Opdracht.....	2
Topicus	3
Mens en cultuur	3
Finance	4
Proces uitvoering onderzoek.....	5
Aanpak per deelvraag.....	5
Scope aanpassingen gedurende het onderzoek	6
Onderzoek - onderzoeksvragen	7
Hoe ziet het huidige dataverwerkingsproces eruit?	7
Welke toetsingen zijn er bij een hypotheekaanvraag en welke data is hier voor nodig?	9
Met welke technieken kan de benodigde data automatisch verkregen worden?	9
Welke data benodigd voor toetsingen is te verkrijgen d.m.v. een externe bron?	10
Wat is OCR en welke OCR-technieken zijn er?	14
Met welke technieken kan de data benodigd voor een toetsing het beste verkregen worden?	18
Wat zijn de beste dataelementen om te automatiseren?	20
Onderzoek - Conclusie.....	22
Proces uitvoering prototype.....	24
Activiteiten en planning	24
Process uitvoering	24
Kwaliteit bewaking	25
Functionele toelichting proof of concept.....	26
Beschrijving proof of concept.....	26
Design keuzes	30
Requirements proof of concept	31
Technische toelichting proof of concept.....	32
Class diagram.....	33
De API	34
Beschrijvingen dataverkrijgingstechnieken.....	35
Testen	40

Initiatief OCR-tool vergelijking	41
Traject van initiatief.....	41
Mijn bijdrage	42
Product Conclusie.....	43
Aanbevelingen.....	44
Productie data PSD2 en OCR	44
Proces dekkende applicatie.....	44
PSD2 data-analyse	46
Reflectie.....	47
Situatie.....	47
Taak	47
Actie.....	47
Resultaat.....	47
Reflectie.....	48
Bronnenlijst	49

Samenvatting

Topicus.Finance maakt onder andere software voor geldverstrekkers om hypotheek te verstrekken aan consumenten. Deze software, genaamd FORCE, wordt ingezet om het gehele aanvraagproces van een hypotheek aan te sturen. Voor de toetsingen in dit proces zijn een aantal documenten benodigd die op het moment met de hand in het systeem gezet worden. De afstudeerder, Sander Groot Wesseldijk, is door Topicus aangenomen om tijdens zijn afstudeerproject te onderzoeken hoe data benodigd voor het hypotheekaanvraagproces het meest economisch verkregen kan worden.

De hoofdvraag van het onderzoek luidt als volgt: “Welk document benodigd voor een toetsing is het meest economisch om automatisch te verwerken?”.

Ter ondersteuning van het onderzoek is er een prototype ontwikkeld. Het prototype maakt gebruik van drie verschillende dataverkrijgingstechnieken:

- OCR
- Scraping
- Services

Opdracht

Aan de afstudeerder, Sander Groot Wesseldijk, was de opdracht om tijdens zijn afstudeerproject te onderzoeken hoe data benodigd voor het hypotheekaanvraagproces het meest economisch verkregen kan worden. Topicus vermoedt dat veel geld en tijd gewonnen kan worden met de automatisering van de dataverkrijging maar heeft op het moment zelf niet de middelen om dit te onderzoeken. Het onderzoek wordt ondersteund met een proof of concept dat gebruik maakt van verschillende dataverkrijgingstechnieken.

Het doel van het onderzoek is om inzichtelijk te maken welke dataelementen het beste geschikt zijn om te automatiseren. Hiervoor zal gekeken moeten worden naar de informatie die benodigd is, wat hier de eigenschappen van zijn en met welke technieken dit het best verkregen kunnen worden.

De hoofdvraag van het onderzoek luidt “Hoe kan data benodigd voor een toetsing het meest economisch verwerkt worden?”.

Om de hoofdvraag te beantwoorden zijn de volgende deelvragen geformuleerd:

- Hoe ziet het huidige dataverwerkingsproces eruit?
- Welke toetsingen zijn er bij een hypotheekaanvraag en welke data is hiervoor nodig?
- Met welke technieken kan de benodigde data automatisch verkregen worden?
- Welke data benodigd voor toetsingen is te verkrijgen door middel van een externe bron?
- Wat is OCR en welke OCR-technieken zijn er?
- Met welke technieken kan de data benodigd voor een toetsing het beste verkregen worden?
- Wat zijn de beste dataelementen om te automatiseren?

Het onderzoek wordt ondersteund door een prototype dat aantoont dat data daadwerkelijk verkregen kan worden door middel van de verschillende dataverkrijgingstechnieken benoemd in het onderzoek. Het prototype maakt gebruik van de volgende dataverkrijgingstechnieken:

- OCR (doordat er geen toegang was tot een tool is er gebruik gemaakt van dummy data)
- Scraping (het scrapen van HTML van overheidswebsites)
- Services (doordat Topicus niet in het bezit is van een PSD2 vergunning is een officiële API nagebootst door de afstudeerder die dummy data beschikbaar stelt)

Topicus

Topicus is een Nederlands softwarebedrijf met ruim 900 medewerkers. Topicus genereert ideeën, maakt systemen gericht op ketenintegratie, investeert in betere oplossingen, ontzorgt organisaties, instellingen en individuen, en ontwikkelt samen met haar partners voortdurend door.

Ketenintegratie leidt tot lagere kosten, flexibele ICT en hogere opbrengsten. Meer bereiken met minder inspanning, dat is de missie van Topicus. (Topicus, sd; Topicus, sd)

MENS EN CULTUUR

De sfeer bij Topicus is ongedwongen en informeel. Toch wordt er serieus en professioneel aan grote projecten gewerkt. Ook is er ruimte voor ontwikkeling van ondernemerschap en creatieve uitpattingen. Die combinatie zorgt ervoor dat Topicus constant technische uitdagingen zoekt. Om tot digitale producten van betekenis te komen.

De kracht van Topicus is de drive anders te denken en te doen. Niet zomaar accepteren dat dingen 'zijn zoals ze zijn', maar jezelf altijd afvragen: hoe kan het anders en beter? Of het nu gaat om het versnellen en versoepelen van kredietaanvragen, het verbeteren van samenwerking tussen gemeenten en regionale zorginstanties of het ondersteunen van passend onderwijs: voor alles werd een innovatieve oplossing bedacht, waarbij de sleutel tot succes ligt in verantwoorde informatiedeling.

Gevolg daarvan is dat Topicus als een magneet werkt voor mensen die iets willen betekenen in de maatschappij. Deze Topicianen worden gestuurd op hun talent en eigen inbreng. Ze krijgen snel verantwoordelijkheden en fouten worden niet bestraft, zolang ze van hun fouten leren. Het bestraffen van fouten werkt namelijk transparantie tegen. En waar Topicus in hun producten openheid nastreeft, doen ze dat ook in eigen gelederen. Het gevolg van die openheid is persoonlijke groei van mensen. De groei van het bedrijf als geheel en het ontstaan van nieuwe initiatieven steeds meer gericht op de consument. (Topicus B.V.)

FINANCE

Topicus werkt in de sectoren Finance, Zorg, Overheid, Onderwijs en Legal. Voor de invulling van de achtergrond van dit verslag zal er ingezoomd worden op Topicus Finance.

Topicus Finance richt zich op de ICT-matige ondersteuning van bedrijfsprocessen in de financiële sector. Dit doet Topicus met behulp van het FORCE framework, een componentgebaseerd raamwerk dat in staat is om praktisch elk financieel proces verregaand geautomatiseerd te ondersteunen.

De visie van Topicus op ondersteuning van bedrijfsprocessen in de financiële sector is gebaseerd op de volgende kerngedachten:

1. Zoveel mogelijk geautomatiseerde afhandelingen van processen
2. Korte time-to-market voor de introductie van nieuwe producten en processen
3. Distributie naar andere partijen is mogelijk via het “common workspace” concept
4. Componenten zijn herbruikbaar, ongeacht het te ondersteunen bedrijfsproces
5. Directe link met de eindconsument
6. De executie van campagnes is onderdeel van de front- en midoffice
7. Mid- en backoffice zijn te ondersteunen middels één applicatie

(Topicus B.V.)

Proces uitvoering onderzoek

AANPAK PER DEELVRAAG

Het onderzoek bestaat uit verschillende deelvragen die ieder op hun eigen manier onderzocht zijn. Hieronder staan de deelvragen van het onderzoek met de bijbehorende aanpak.

Hoe ziet het huidige dataverwerkingsproces eruit?

Om deze vraag te beantwoorden is gebruik gemaakt van de documentatie van Topicus. De gevonden resultaten zijn in twee interviews met medewerkers gevalideerd. De geïnterviewde zijn Maarten Schopman en Nicky Koenders.

Met welke technieken kan de benodigde data verkregen worden?

Om deze deelvraag te beantwoorden is een medewerker geïnterviewd. De geïnterviewde zijn Maarten Schopman en Nicky Koenders.

Welke toetsingen zijn er bij een hypotheekaanvraag en welke data is hiervoor nodig?

Om deze vraag te beantwoorden is gebruik gemaakt van de documentatie van Topicus. De gevonden resultaten zijn aangevuld met een analyse van de test omgeving van de Syntrus versie van FORCE. De gevonden informatie is aangevuld door middel van een interview met de medewerker Lennart Boot.

Welke data benodigd voor toetsingen is te verkrijgen d.m.v. externe bronnen?

Om deze deelvraag te beantwoorden is er onderzoek gedaan naar technieken/methodes voor het verkrijgen van data van externe bronnen.

Om te achterhalen welke externe bronnen gebruikt kunnen worden is een medewerker geïnterviewd, de geïnterviewde was Maarten Schopman. Dit interview leidde tot de website hypotheek-brondata.nl. Op hypotheek-brondata wordt bijgehouden waar relevante informatie voor een hypotheekaanvraag momenteel opgeslagen wordt en er wordt bijgehouden welke initiatieven met betrekking tot het hypotheekaanvraag proces momenteel plaatsvinden. De initiatieven en bronnen zijn geanalyseerd.

Wat is OCR en welke OCR-technieken zijn er?

Om deze deelvraag te beantwoorden is een deskresearch uitgevoerd.

Met welke technieken kan de data benodigd voor een toetsing het beste verkregen worden en wat zijn de kosten van deze technieken?

Om deze deelvraag te beantwoorden zijn de resultaten van de onderzoeksvragen “Welke data benodigd voor toetsingen is te verkrijgen d.m.v. externe bronnen?” en “Wat is OCR en welke OCR-technieken zijn er?” met elkaar vergeleken.

Wat zijn de beste dataelementen om te automatiseren?

Om deze deelvraag te beantwoorden is er een conclusie getrokken uit de resultaten van de bovenstaande deelvragen.

SCOPE AANPASSINGEN GEDURENDE HET ONDERZOEK

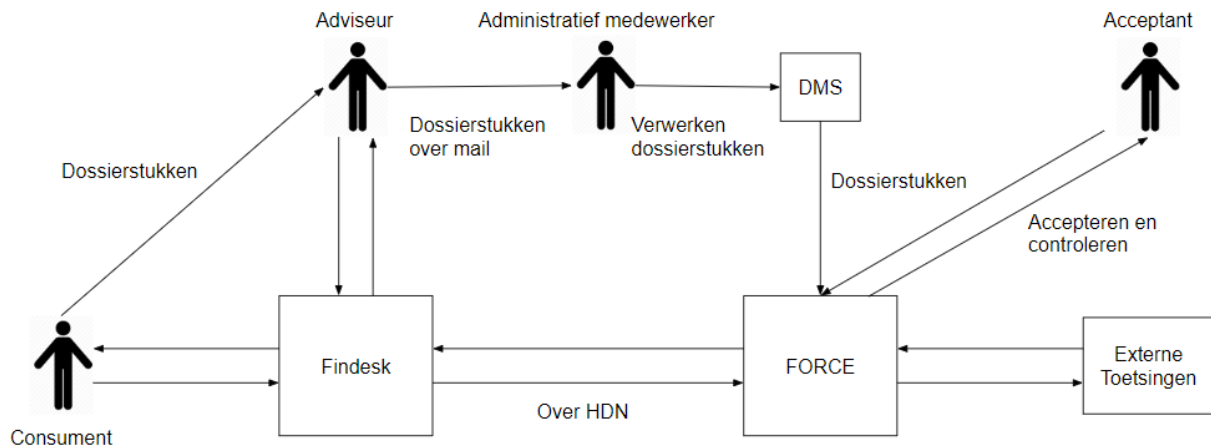
Tijdens het onderzoek waren een aantal momenten waarop de afbakening van het onderzoek en van het project vernauwd zijn. Hieronder worden de gemaakte keuzes verantwoord.

Oorzaak/reden	Gevolg/afbakening aanpassing
Documentatie over toetsing is verspreid en van slechte kwaliteit. Dit probleem wordt erger wanneer er gekeken werd naar klant specifieke toetsingen.	Er zal alleen gekeken worden naar toetsingen benodigd voor de basis versie van FORCE.
De OCR-techniek zal niet voor ieder dataelement gebruikt kunnen worden en er zijn veel verschillende documenten die geleverd kunnen worden bij een hypotheekaanvraag. Het onderzoeken van de eigenschappen van alle documenten is dus onhandig.	De eigenschappen van een document zullen niet vastgelegd worden in het onderzoek.
In de laatste week van het onderzoek moest er nog onderzocht worden hoe vaak dataelementen benodigd zijn voor een toetsing. Het onderzoeken hiervan is echter veel werk.	Er zal niet onderzocht worden hoe vaak een dataelement benodigd is maar er zal wel rekening gehouden worden met hoe vaak een dataelement voor kan komen bij een aanvraag.

Onderzoek - onderzoeksvragen

HOE ZIET HET HUIDIGE DATAVERWERKINGSPROCES ERUIT?

FORCE krijgt data van een financieel advies applicatie¹ (FAA), adviseurs en van externe toetsingen. Informatie van een aanvraag wordt verstuurd vanuit een FAA over HDN². De adviseur levert via post of email de benodigde dossierstukken van de consument. De externe toetsingen bepalen of een consument wel of geen hypotheek mag of kan aanvragen. In de rest van dit hoofdstuk worden deze verschillende stappen verder toegelicht.



Figuur 1 Overzicht dataverkeer van en naar FORCE met Findesk³ als FAA

Financieel advies applicatie

Door middel van een FAA ontvangt FORCE-gegevens van de consument, deze gegevens zijn door de consument zelf ingevuld en zijn niet gevalideerd. Deze gegevens worden geleverd over het HDN. HDN staat voor Hypotheken Data Network, door dit netwerk verloopt het berichtenverkeer rond het hypotheekaanvraag traject automatisch. Dossierstukken worden normaliter niet geleverd door een FAA maar direct door de adviseur.

Dossierstukken

De dossierstukken benodigd voor een hypotheekaanvraag worden bijgehouden in het DMS. DMS staat voor document managementsysteem, dit is de omgeving waarin document behorende bij een hypotheekaanvraag opgevraagd kunnen worden. De documenten worden geleverd door de adviseur.

Verkrijgen van dossierstukken

Bij het starten van een hypotheekaanvraag geeft de consument zijn situatie aan, zijn situatie bepaalt welke informatie en dus ook welke dossierstukken, van hem/haar benodigd zijn.

De benodigde dossierstukken worden door de consument geleverd aan de adviseur, de adviseur stuurt het document op naar de hypotheekverstrekker door middel van post of email, bij levering door middel van post worden de documenten gedigitaliseerd. Een medewerker koppelt de documenten aan het dossier van een consument waarnaar de documenten geclassificeerd worden. Het classificeren is het aangeven van het type document.



Figuur 2 Overzicht verwerking van een dossierstuk

Externe toetsingen

Bij het hypotheekaanvraag proces worden veel toetsingen uitgevoerd om te bepalen of de klant een hypotheek aan mag en kan vragen. Een aantal van deze toetsingen worden gedaan door externe partijen. Een voorbeeld van één van deze externe partijen is het BKR, BKR staat voor Bureau Krediet Registratie. Het BKR houdt van elke persoon met een in Nederland afgesloten krediet een elektronisch dossier bij waar betalingsachterstanden in worden bijgehouden. Personen krijgen bij een achterstand melding een BKR-code die aangeeft wat de situatie omtrent de betalingsachterstand is. Afhankelijk van de BKR-code die een persoon toegewezen is kan zijn/haar verzoek tot een hypotheek afgekeurd worden.

BKR-code	Omschrijving
A	Achterstandsmelding: op een lening is een achterstandsmelding gegeven door de kredietgever.
H	Herstel: een lening met achterstandsmelding (code A) krijgt code H zodra de betalingsachterstand is ingelopen.
1	er is sprake van een geregistreerde betalingsachterstand waarbij inmiddels een regeling getroffen is voor de nog openstaande lening.
2	het nog openstaande gedeelte van de lening is opeisbaar gesteld. Dit betekent dat de lening uit handen is gegeven aan een incassobureau.
3	tweehonderd vijftig euro of meer is op de schuld afgeboekt.
4	degene die de lening is aangegaan is voor de kredietgever onbereikbaar.

Figuur 3 Mogelijke BKR-codes (Wat betekenen de coderingen op mijn overzicht?, sd)

Valideren van informatie

Doordat de door de consument ingevulde informatie geleverd door een FAA niet gevalideerd is moet deze nog gecontroleerd worden door een medewerker van de hypotheekverstrekker. Deze controle wordt gedaan door de geleverde informatie te vergelijken met de informatie in de dossierstukken. Dit betekent dat wanneer een medewerker een dataelement wil controleren hij het juiste dossierstuk moet openen uit het DMS en deze naast de geleverde informatie moet houden. Het uitvoeren van de totale validatie duurt per aanvraag dertig tot zestig minuten.

WELKE TOETSINGEN ZIJN ER BIJ EEN HYPOTHEEKAANVRAAG EN WELKE DATA IS HIER VOOR NODIG?

Topicus heeft meerdere klanten die het product FORCE afnemen, iedere klant heeft een eigen versie van FORCE waarin functionaliteiten zitten die specifiek zijn voor de klant. Er is besloten om alleen toetsingen te onderzoeken van de basis versie van FORCE. Deze keuze is gemaakt omdat de toetsingen in de standaardversie betrekking hebben op alle klanten.

De toetsingen zijn te vinden in de bijlage "Toetsingen.xlsx". Hierin staan de toetsingen met de dataelementen die benodigd zijn voor de toetsing. De toetsingen zijn afgeleid van de documentatie van Topicus.

MET WELKE TECHNIKEN KAN DE BENODIGDE DATA AUTOMATISCH VERKREGEN WORDEN?

De data benodigd voor een hypotheekaanvraag komt momenteel uit dossierstukken. Dit zijn documenten die door een medewerker handmatig verwerkt worden.

De benodigde informatie kan automatisch verkregen worden door het gebruik van een externe digitale bron. Informatie van een extern systeem kan opgenomen worden in de systemen van Topicus. Wanneer de bron beschikt over een API dan kan hierop aangesloten worden, wanneer de bron niet beschikt over een API maar wel informatie weergeeft door dit te serveren door middel van HTML dan kan de informatie verzameld worden door middel van scrapen (ook wel bekend als web scraping). De mogelijkheden van scrapen kunnen echter gelimiteerd worden wanneer de te scrapen HTML niet publiek toegankelijk is.

De benodigde informatie is aanwezig in de geleverde documenten. Door middel van de OCR-techniek kan informatie door computers gehaald worden uit gedigitaliseerde documenten. (Optical Character Recognition (OCR) - How it works, 2012)

WELKE DATA BENODIGD VOOR TOETSINGEN IS TE VERKRIJGEN D.M.V. EEN EXTERNE BRON?

Externe data kan op twee verschillende manieren verkregen worden, door middel van het gebruik van een service of door middel van scrapen.

Methodes

Service

Veel data wordt beschikbaar gesteld door middel van services, dit is vaak in de vorm van een API. Bij gebruik van een service kan het gaan over open data of betaalde data. Data is open data wanneer:

- De data openbaar is
- De data geen auteursrecht of andere rechten heeft
- De data bekostigd is uit publieke middelen
- De Data voldoet aan open standaarden

(Wat is open data, sd)

Scrapen

Wanneer data door een partij niet beschikbaar gesteld is door middel van een service dan kan scrapen toegepast worden om data uit een website te halen, indien de te extraheren data gepresenteerd wordt door middel van HTML. Scrapen is een computertechniek waarbij software wordt gebruikt om informatie van webpagina's te extraheren en al dan niet te analyseren.

Het is echter niet toegestaan om scrapen toe te passen zonder toestemming wanneer de data beschermd wordt door het databankrecht. Zoals beschreven door het RVO:

De structuur van de databank is beschermd door het auteursrecht als het gaat om:

- Een verzameling van werken, gegevens of andere zelfstandige elementen
- Die systematisch of methodisch geordend
- En afzonderlijk, met elektronische middelen of anderszins toegankelijk zijn
- En waarvan de verkrijging, de controle of de presentatie van de inhoud, in kwalitatief of kwantitatief opzicht getuigt van een substantiële investering

(Databankrecht, sd)

Een groot nadeel van scrapen is dat de partij die gescraped wordt hier geen rekening mee houdt. Hierdoor zijn er geen garanties met betrekking tot de stabiliteit van de scraper. Wanneer de HTML van de bron pagina verandert zal de scraper breken, om de aangepast pagina te scrapen moet de scraper geherprogrammeerd worden.

Is scrapen ethisch en legaal?

Scrapen is een relatief nieuwe technologie op de markt. Door de natuur van scrapen zijn veel mensen bezorgd over de ethiek achter het scrapen. In mijn mening moet om hier iets over te zeggen onderscheid gemaakt worden tussen twee verschillende types van bronnen. Bronnen die publiek data beschikbaar stellen en bronnen die data beschikbaar stellen achter een vorm van authenticaties (zoals bijvoorbeeld een inlog door middel van DigiD voor MijnOverheid).

Bronnen met publieke data

Bij publieke data is er geen sprake van gevoelige of persoonlijke gegevens, het zal bijna altijd gaan om een lijst van feiten. Indien de bron zelf veel belang heeft bij het beschikbaar stellen van de gegevens dan is deze zeer waarschijnlijk beschermd door het databankrecht. Zolang de scraper de databankrecht wet niet schendt zijn er geen problemen.

Bronnen met authenticatie

Bij het scrapen van bronnen waar de informatie zich bevindt achter een vorm van authenticatie is het ingewikkelder om te stellen of het scrapen van de bron legaal en ethisch is. Uit een bespreking tussen DigiD, Logius en de Nationale Hypotheekbond blijkt dat voldaan moet worden aan de volgende voorwaarden:

- De scraper mag geen databankrecht schenden.
- In geval van persoonsgegevens mag de scraper de AVG-wetgeving niet schenden.
- De scraper moet toestemming hebben van de persoon met wiens gegevens ingelogd worden.
 - Wanneer er sprake is van een inlog door middel van DigiD moet de gebruiker zelf inloggen op de website van DigiD.

(Limburg, 2017)

Service of scrapen?

Het gebruik van een Service heeft de voorkeur, want:

- Het ontwikkelen van een scraper is meer werk dan het benutten van een aangeboden service.
 - Het direct scrapen van HTML vereist ontwikkeling van foutgevoelige code, data kan opgeslagen zijn in verschillende HTML-structuren die afhankelijk zijn van het request.
 - Het scrapen vereist meer stappen dan het benutten van een service.
 - Meerdere requests om HTML verkrijgen.
 - Het parsen van de HTML.
- Het gebruik van een scraper neemt risico's met zich mee.
 - De bronpartij houdt geen rekening met de verbinding die door de scraper is opgezet. Wanneer de structuur van de bron HTML verandert zal de scraper niet meer na behoren functioneren. De scraper zal geherprogrammeerd moeten worden voor iedere aanpassing op de structuur van de bron HTML.

Externe bronnen met een service

Hieronder worden de bronnen met een service benaderd, er wordt per bron aangegeven welke informatie hiervan verkregen kan worden. FORCE heeft al verbindingen met een aantal externe bronnen, op deze bronnen zal niet verder worden ingegaan.

Basisregistratie adressen en gebouwen (BAG)

Basisregistratie adressen en gebouwen, afgekort BAG, is een API van de overheid die gegevens van het kadaster beschikbaar stelt. Om toegang te krijgen tot de gegevens achter de API moet een mail gestuurd worden naar dataplatform@kadaster.nl. (Basisregistratie Adressen en Gebouwen (BAG) (1.0.0), sd)

De BAG API levert op het moment maar één relevant dataelement voor het aanvragen van een hypotheek. Op basis van het adres en postcode van de woning kan het bouwjaar opgevraagd worden.

PSD2

Op 19 februari 2019 is in Nederland PSD2 in werking getreden. PSD2 is de nieuwe Europese wet voor het betalingsverkeer van consumenten en bedrijven. PSD2 staat voor Payment Services Directive. Met PSD2 kunnen derde partijen toegang krijgen tot data van een consument bij een bank indien de consument de derde partij hier toegang toe geeft. Deze informatie kan worden verkregen van een API die door de bank beschikbaar wordt gesteld. Relevante informatie die beschikbaar gesteld wordt zijn:

- Huidig saldo
- Transactie geschiedenis (tot hoe ver in het verleden is verschillend per bank)

Op het moment heeft geen Nederlandse bank een volledig ondersteunde API staan. De meeste banken bevinden zich in een prototype fase en bieden alleen toegang tot een sandbox omgeving. (PSD2, sd; DeVolksbank PSD2 sandbox, 2019; Betaalrichtlijn PSD2. Wat kun je ermee?, sd)

PSD2 toestemming van consumenten

PSD2 is een relatief nieuw concept en ligt voor veel mensen op de grens van wat wel en niet zou kunnen omtrent privacy. Om het gebruik van een PSD2 een succes te maken moeten consumenten bereid zijn om informatie te delen door middel van de PSD2 API, als er geen toestemming van een consument is dan is het niet mogelijk om de data te verkrijgen. Hieronder staan een aantal redenen waarom consumenten wel toestemming zouden geven voor het verkrijgen van de informatie.

Nauwelijks sprake van nieuwe informatie

Indien de partij die informatie vraagt van de consument met het gebruik van de PSD2 service alleen salarissen opvraagt is er nauwelijks sprake van nieuwe informatie. Op het moment wordt de informatie die in de salarisoverschrijving staat verkregen uit door de consument opgestuurde documenten. Informatie die wel nieuw is bij het gebruik van PSD2 is een betalingskenmerk.

Makkelijker voor alle partijen

Het gebruik van PSD2 maakt het overhandigen van de informatie van consument naar hypotheekverstrekker eenvoudiger. De consument hoeft geen documenten aan te vragen of op te zoeken maar geeft op een applicatie/website toegang voor het verkrijgen van de informatie door middel van PSD2.

Voor hypotheekverstrekkers is PSD2 een aantrekkelijkere optie dan het verkrijgen van data door middel van het opsturen van documenten, dit komt doordat:

- De informatie digitaal geleverd wordt, dit maakt het automatiseren van het proces eenvoudiger
- De informatie van een vertrouwde bron komt, documenten kunnen vervalst worden en moet daardoor gecontroleerd worden door medewerkers van een hypotheekverstrekker. De data van een PSD2 service kan echter als waarheid beschouwd worden.

Hierdoor zou het voor hypotheekverstrekkers zo aantrekkelijk worden dat ze het gaan eisen van consumenten in plaats van een loonstrook of een soortgelijk document.

Betrouwbare informatievrager

De partij die informatie opvraagt van een consument door middel van PSD2 is geen willekeurige partij. De partij is een hypotheekverstrekker (of een derde partij in dienst van de hypotheekverstrekker) die moet voldoen aan strikte regels en controles om een PSD2 vergunning te krijgen en te behouden.

Externe bronnen zonder een service

Met behulp van Hypotheekbrondata zijn een aantal externe bronnen onderzocht naar dataelementen om te extraheren door middel van scrapen. In de bijlage “Externe_bronnen.xlsx” is een uitgebreid overzicht te zien van externe bronnen met de door bijpassende dataelementen.

WAT IS OCR EN WELKE OCR-TECHNIEKEN ZIJN ER?

OCR staat voor Optical Character Recognition, letterlijk vertaald naar het Nederlands is het optische karakterherkenning. OCR is de mechanische of elektronische conversie van een afbeelding naar machine-gecodeerde tekst, vaak in de vorm van TXT, DOC of pdf-bestanden. (Optical Character Recognition (OCR) - How it works, 2012)

Geschiedenis

De vroegste concepten van OCR zijn te herleiden tot 1870 waar de Fournier d'Albe's Optophone en de Tauschek's Reading Machine ontwikkeld werden om blinden te helpen te lezen. (Schantz, 1982)

De eerste OCR-hulpmiddelen verschenen tussen 1931 en 1954, deze hulpmiddelen waren instaat om Morse Code te interpreteren en tekst als geluid af te spelen. (The First OCR System: "GISMO", sd)

De algemene werking

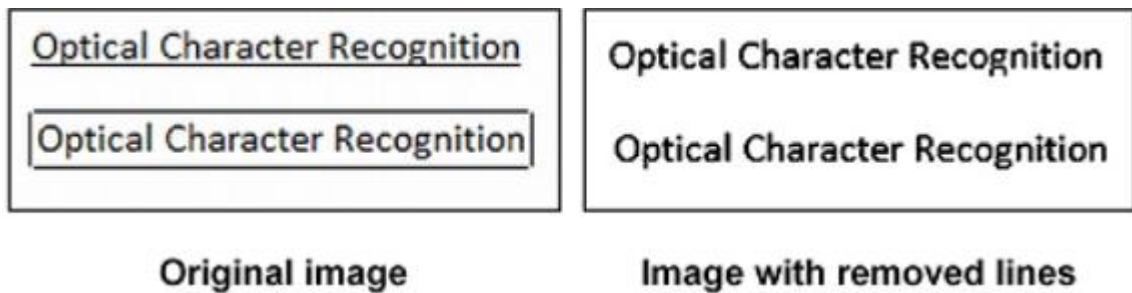
Het proces dat een afbeelding converteert is gescheiden in meerdere stappen. Iedere stap bestaat uit een aantal algoritmes die een gedeelte van het OCR-proces uitvoeren. De standaard stappen binnen een OCR-proces zijn:

- Het inladen van een afbeelding van een bron. Een bron kan een bestand zijn, de typen bestanden die ondersteund worden is afhankelijk van het OCR-systeem.
- Het detecteren van de eigenschappen van de afbeelding, zoals resolutie.
- Het verbeteren van de kwaliteit van de afbeelding om ruis eruit te halen.
- Het binarization proces wordt uitgevoerd, dit is het converteren van afbeeldingen in kleur of grijs tinten naar een zwart-wit afbeelding.



Figuur 4 Voorbeeld van binarization

- Het detecteren en verwijderen van lijnen, het verwijderen van lijnen verbeterd de pagina layout analyse bij onderliggende tekst.



Figuur 5 Voorbeeld van lijn verwijdering

- De pagina lay-out analyse, dit is het detecteren van posities en types van alle belangrijke gebieden op de pagina.
- Het detecteren van tekstlijnen en woorden.
- Het analyseren van Combined-broken karakters, dit is de analyse op karakters die bestaan uit meerdere delen en de analyse op karakters die in aanraking komen met elkaar.



Figuur 6 Voorbeeld van Combined-broken karakters

- Het herkennen van karakters. Dit is het hoofd algoritme van OCR. Een afbeelding van iedere karakter moet geconverteerd worden naar de bijpassende karakter code. Het kan voorkomen dat het algoritme meerdere karakter codes genereert voor onduidelijke afbeeldingen. Bijvoorbeeld, het herkennen van de "l" letter kan de codes voor "l", "|", "1" en "I" genereren, de definitieve code wordt in een later stadium bepaald.
- Het vergelijken van de resultaten met een woordenboek, door de resultaten te vergelijken met een woordenboek kunnen sommige fouten in de verwerking verholpen worden.
- Het opslaan van de resultaten.

Dit is niet een complete lijst. Veel kleinschalige algoritmes moeten ook nog geïmplementeerd worden om tot een goede herkenning te komen met verschillende omstandigheden en verschillen tussen verschillende OCR-systemen.

Iedere stap in het OCR-proces is belangrijk. Wanneer één stap in het OCR-proces faalt dan faalt het gehele OCR-proces. Iedere algoritme moet zo goed mogelijk werken op zoveel mogelijk verschillende afbeeldingen. Het is echter mogelijk om een OCR-systeem te hebben voor specifieke omstandigheden wat het OCR-proces versimpelt. (Optical Character Recognition (OCR) - How it works, 2012)

Technieken

Er zijn verschillen vormen van OCR, iedere vorm heeft zijn eigenschappen. Hieronder zal ik de verschillende vormen van OCR benaderen om zo te achterhalen wat de verschillende eigenschappen zijn.

Standaard OCR

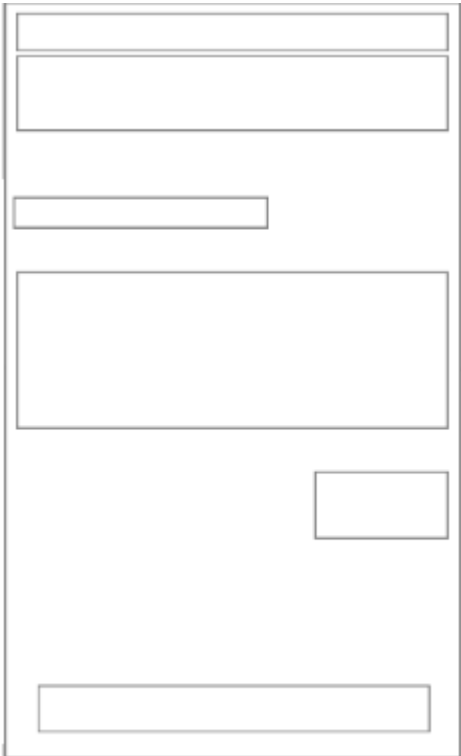
De term standaard OCR, soms ook wel genaamd 'out of the box Optical Character Recognition', beschrijft OCR-systemen die een afbeelding in zijn geheel converteren naar doorzoekbare en aanpasbare documenten. Deze documenten zijn vaak, maar niet uitsluitend, van de bestandstypes PDF, WORD en txt. (Using Zonal OCR to Extract Data Fields From Scanned Documents, sd)

Zonal OCR

Zonal OCR gaat een stap verder dan standaard OCR. In plaats van alleen instaat zijn om afbeeldingen te converteren naar tekst kan een zonal OCR-systeem getraind worden om de structuur en hiërarchie van een document te begrijpen.

Door een zonal OCR-systeem te leren waar welke informatie staat is het mogelijk om gericht data te extraheren van een document. Een ander voordeel van zonal OCR is dat de nauwkeurigheid van het OCR-proces vergroot wordt door:

- De karakter invoer te beperken.
- Het mogelijk wordt om een lexicon te gebruiken op een veld per veld niveau.

	 BRASS FLAMINGO BAR AND GRILL 2550 WEST OAK AVE SEATTLE, WA 01-17-2017 10:31PM SERVER JOHN TABLE 1 <table border="0"><tr><td>1</td><td>MIX DRINK - PREMIUM</td><td>\$6.25</td></tr><tr><td>1</td><td>DRAFT BEER 16OZ</td><td>\$4.25</td></tr><tr><td>2</td><td>HOUSE WINE</td><td>\$6.25</td></tr></table> CASH SUB-TOTAL: \$18.75 TAX \$7.00 TOTAL \$25.75 THANK YOU	1	MIX DRINK - PREMIUM	\$6.25	1	DRAFT BEER 16OZ	\$4.25	2	HOUSE WINE	\$6.25
1	MIX DRINK - PREMIUM	\$6.25								
1	DRAFT BEER 16OZ	\$4.25								
2	HOUSE WINE	\$6.25								
Template	Example document									

Figuur 7 Voorbeeld van het aangeven van datavelden (Infrd, 2018)

Het trainen van een zonal OCR-systeem is een eenmalig procedure voor ieder document lay-out. Het trainen van een zonal OCR bestaat uit het aangeven welke datavelden op welke locaties voor kunnen komen. Zonal OCR werkt alleen door middel van aangegeven locaties, dit betekent dat data extraheren van semigestructureerde documenten lastig of onmogelijk is met zonal OCR. (Using Zonal OCR to Extract Data Fields From Scanned Documents, sd; United States Patentnr. 537,729, 2012)

Machine learning OCR

Machine learning OCR maakt gebruik van machine learning om data te vinden binnen een document. Hiervoor zal het OCR-systeem getraind moeten worden. Bij de eerste 'batch' aan documenten geeft een gebruiker aan waar het systeem de datavelden kan vinden, het systeem leert hiervan om zo toekomstige documenten beter te verwerken. Wanneer het systeem niet zeker genoeg is over een document zal er aan een gebruiker gevraagd worden om aan te geven waar het systeem de data kan vinden waar deze onzeker over is, wederom zal het systeem hiervan leren.

Machine learning OCR werkt bijzonder nuttig wanneer de documenten die verwerkt worden verschillende lay-outs hebben. (Ephesoft, 2016)

MET WELKE TECHNIKEN KAN DE DATA BENODIGD VOOR EEN TOETSING HET BESTE VERKREGEN WORDEN?

In de bovenstaande onderzoeksvragen zijn drie technieken onderzocht voor het automatisch verkrijgen van de benodigde data. Deze technieken zijn: Het gebruik van een service, het gebruik van scrapen en het gebruik van OCR.

Hieronder worden de drie technieken met elkaar vergeleken op kosten en betrouwbaarheid. Verstrekking van hypotheek is een proces waar het om veel geld gaat. Hierdoor is het minder tot niet interessant om te kijken naar de eenmalige kosten. Relevant zijn de langdurige kosten zoals het onderhoud en updaten van software of mogelijke extra kosten per aanvraag door het betrekken van een externe partij.

Kosten per request

Een service heeft alleen kosten per request wanneer er sprake is van een betaalde API.

Een scrape-tool heeft geen kosten per request.

Een OCR-tool kan per request kosten hebben uit twee bronnen:

- Het gebruik van een OCR-tool kan geld kosten per te verwerken document, dit is afhankelijk van de OCR-tool. Open source OCR-tools bestaan voornamelijk uit standaard OCR.
 - Deze kosten kunnen oplopen tot \$0.065 per pagina.
- Het corrigeren en valideren van een OCR-tool moet gedaan worden door een medewerker.
 - Deze kosten variëren afhankelijk van hoe goed het document verwerkt kan worden.
 - Bij een gemiddelde nauwkeurigheid van 95% tot 100% zal de medewerker niet tot weinig te hoeven corrigeren.
 - Bij een nauwkeurigheid onder 95% zal de medewerker regelmatig moeten corrigeren.

Onderhoudskosten

De onderhoudskosten van de verschillende dataverkrijgingstechnieken heb ik geschat op basis van mijn ervaring. Ik heb voor deze schatting ongeveer twee dagen per techniek besteed om de techniek uit te proberen.

De onderhoudskosten van een verbinding met een service zijn laag. Dit komt doordat de service bewust is van de verbinding en deze ook ondersteunt. Wanneer de service een update ondergaat zal dit meestal niet van invloed zijn op de al beschikbare informatie. Wanneer er sprake is van breaking changes⁴ dan zal de gebruiker van de service geïnformeerd worden en een periode hebben om de wijzigingen door te voeren.

De onderhoudskosten van een scrape tool zijn afhankelijk van hoe vaak de structuur van de bron HTML verandert. Wanneer de bronpartij zijn structuur aanpast zal de scrape tool aangepast moeten worden, de verandering op de bron kunnen echter gebeuren zonder waarschuwing.

De onderhoudskosten van een OCR-tool zijn afhankelijk van de documenten die verwerkt worden. Wanneer de structuur van een document verandert dan moet de OCR-tool ook bijgewerkt worden. Hoe lastiger het document is om te verwerken des te meer interactie benodigd zal zijn van een medewerker.

Het is voor mensen lastig om in te schatten hoeveel werk/tijd het uitvoeren van een taak kost, hierom heb ik een relatieve schatting gemaakt gebruik makend van de rij van Fibonacci. Een service kost de minste onderhoudskosten, deze heb ik vastgezet op 2 punten. Relatief aan een service schat ik de onderhoudskosten van scraping ongeveer als de helft zoveel, de onderhoudskosten van scraper heb ik dus op 3 punten geschat. De onderhoudskosten van OCR zijn aanzienlijk hoger dan de onderhoudskosten van een service of een scraper, de onderhoudskosten van OCR heb ik hierom geschat op 8 punten.

Betrouwbaarheid

Het gebruik van een service en het gebruik van een scrape-tool zijn beiden zeer betrouwbaar doordat de informatie gehaald wordt van een vertrouwde bron.

Het gebruik van een OCR-tool is zeer betrouwbaar, echter zal een OCR-tool documenten niet verwerken met 100% zekerheid.

Vergelijking van de technieken

	Service	Scrapen	OCR
<i>Kosten per request per dataelement</i>	≈0	≈0	\$0.065+. Afhankelijk van aantal documenten en nauwkeurigheid
<i>Onderhoudskosten in relatieve punten</i>	2	3	8
<i>Betrouwbaarheid (percentage correcte verwerkingen)</i>	≈100%	≈100%	< 100%. Afhankelijk van document

Uit de vergelijking is af te leiden dat het gebruik van een service de beste techniek is voor het automatisch verkrijgen van de benodigde informatie.

Doordat de kosten per request van een scrape-tool aanzienlijk lager zijn dan het gebruik van een OCR-tool is het gebruik van een scrape tool de een na beste techniek.

WAT ZIJN DE BESTE DATAELEMENTEN OM TE AUTOMATISEREN?

Het verwerken van een enkel dataelement kost geen tot weinig tijd. Momenteel zit de meeste tijd voor het verwerken van de benodigde data in het zoeken, valideren, openen en scannen van het document waarin de data genoteerd staat. Hierdoor wordt de meeste tijd gewonnen wanneer een volledig document geautomatiseerd wordt.

Het document dat het vaakst gevraagd wordt bij een aanvraag is het beste om te automatiseren. In een voorafgaand onderzoek van Topicus is onderzocht wat de verhoudingen zijn van de verschillende benodigde documenten. Hieronder zijn de tien meest voorkomende documenten te zien, voor een volledig overzicht zie de bijlage “Verhouding_van_benodigde_documenten.xlsx”.

Documenttype	Aantal	Percentage
Identificatie	91847	20%
Salarisspecificatie	67259	15%
Werkgeversverklaring	64100	14%
Koopcontract	48496	10%
Aangifte inkomstenbelasting	38360	8%
Jaarrapport	35592	8%
Schuldrestbewijs hypotheek	14895	3%
Persoonlijk hypotheekadvies	14658	3%
Overig	87550	19%

Op de volgende pagina worden de vier meest voorkomende documenten toegelicht. Er wordt aangegeven welke dataelementen met welke methodes geautomatiseerd kunnen worden. In de bijlage “Externe toetsingen.xlsx” is een overzicht te zien van verschillende externe bronnen met welke dataelementen hiervan verkregen zouden kunnen worden.

Identificatie

Dataelement	Service	Scrapen	OCR
Voornaam	Nee	Ja	Ja
Achternaam	Nee	Ja	Ja
Documentcode	Nee	Ja	Ja
Legitimatienummer	Nee	Nee	Ja
Nationaliteit	Nee	Ja	Ja

Salarisspecificatie

Dataelement	Service	Scrapen	OCR
Bruto jaarsalaris	Ja	Ja	Ja
Ingangsdatum	Ja *indien genoeg geschiedenis.	Nee	Ja
Onregelmatigheidstoeslag	Nee	Nee	Ja
Vakantietoeslag	Ja	Nee	Ja

Werkgeversverklaring

Dataelement	Service	Scrapen	OCR
Einddatum inkomsten	Nee	Nee	Ja
Functie toeslag	Nee	Nee	Ja
Onregelmatigheidstoeslag	Nee	Nee	Ja
Overwerk	Nee	Nee	Ja
Provisie	Nee	Nee	Ja
Type loondienst	Ja	Nee	Ja
Variabele inkomsten	Nee	Nee	Ja
Vaste 13 ^e maand	Ja, *indien genoeg geschiedenis. (Niet te onderscheiden van "Vaste eindejaarsuitkering")	Nee	Ja
Vaste eindejaarsuitkering	Ja, *indien genoeg geschiedenis. (Niet te onderscheiden van "Vaste 13 ^e maand")	Nee	Ja
VEB-toelage	Nee	Nee	Ja

Koopcontract/koopovereenkomst

Dataelement	Service	Scrapen	OCR
Bouwjaar	Ja	Nee	Ja
Energie neutraal(ja/nee)	Nee	Ja	Ja
Energielabel of EPC	Nee	Ja	Ja
Huisnummer	Nee	Nee	Ja
Huisnummer toevoeging	Nee	Nee	Ja
Plaatsnaam	Nee	Nee	Ja
Postcode	Nee	Ja	Ja

Onderzoek - Conclusie

Uit het onderzoeksverslag is gekomen dat de beste te automatiseren dataelementen elementen zijn die samen de meest voorkomende documenten weg automatiseren. Dit komt doordat de meeste kosten bij het handmatig verwerken komen van het inladen, scannen en opzoeken van een document. Het daadwerkelijk één op één overnemen van de informatie kost relatief weinig tijd. Voor het volledige onderzoeksrapport zie de bijlage "onderzoeksrapport.pdf".

Op de volgende pagina staan de vier meest voorkomende documenten. Er wordt aangegeven welke dataelementen benodigd zijn van het documenten, met welke methode deze het beste automatisch verkregen kunnen worden en wat de bron voor de methode is. Voor een lijst van alle dataelementen met best mogelijke automatiseringsmethode zie de bijlage "Dataelementen.xlsx".

Identificatie

Dataelement	Methode	Bron
Voornaam	Scrapen	Mijn.Overheid
Achternaam	Scrapen	Mijn.Overheid
Documentcode	Scrapen	Mijn.Overheid
Legitimatienummer	OCR	Identiteitskaart
Nationaliteit	Scrapen	Mijn.Overheid

Salarisspecificatie

Dataelement	Methode	Bron
Bruto jaarsalaris	Service	PSD2
Ingangsdatum	Service *indien genoeg geschiedenis	PSD2
Onregelmatigheidstoeslag	OCR	Salarisspecificatie
Vakantietoeslag	Service	PSD2

Werkgeversverklaring

Dataelement	Methode	Bron
Einddatum inkomsten	OCR	Werkgeversverklaring
Functie toeslag	OCR	Werkgeversverklaring
Onregelmatigheidstoeslag	OCR	Werkgeversverklaring
Overwerk	OCR	Werkgeversverklaring
Provisie	OCR	Werkgeversverklaring
Type loondienst	Service	PSD2
Variabele inkomsten	OCR	Werkgeversverklaring
Vaste 13 ^e maand	Service *indien genoeg geschiedenis.	PSD2
Vaste eindejaarsuitkering	Service *indien genoeg geschiedenis	PSD2
VEB-toelage	OCR	Werkgeversverklaring

Koopcontract/koopovereenkomst

Dataelement	Methode	Bron
Bouwjaar	Service	BAG
Energie neutraal(ja/nee)	Scrapen	Energielabel.nl
Energielabel of EPC	Scrapen	Energielabel.nl
Huisnummer	OCR	Koopovereenkomst
Huisnummer toevoeging	OCR	Koopovereenkomst
Plaatsnaam	OCR	Koopovereenkomst
Postcode	Scrapen	WOZ-waardeloket

Proces uitvoering prototype

De ontwikkelmethodiek die ik gebruik heb is Scrum. Ik heb gewerkt in sprints van een week, met de stop/start dag op vrijdag.

ACTIVITEITEN EN PLANNING

Het project is op te delen in drie fases; de voorbereiding, het onderzoek en het ontwikkelen. Er zal uitgegaan worden van twintig werkweken doordat de afstudeerder beschikt over tien vakantie dagen.

Fase	Resultaat	Periode
Vorbereiding	Plan van Aanpak	Week 7 t/m week 11
Onderzoek	Onderzoeksrapport	Week 10 t/m week 16
Ontwikkelen	Prototype	Week 17 t/m week 24

PROCESS UITVOERING

Definition of Ready

Voor een scrum project is het van belang dat het duidelijk is wanneer een story klaar is om ontwikkelt te worden, dit is de "definition of ready". Mijn story's waren klaar om ontwikkelt te worden als:

1. De story als een user story opgeschreven was (Bijvoorbeeld: "Als een [Type gebruiker] wil ik [functie] zo dat [voordeel].")
2. De acceptatiecriteria gedefinieerd waren en begrepen werden.
 - a. De acceptatiecriteria van een story beschrijven de voorwaarden waaraan een story moet voldoen.
3. De story gepokerd was.

Definition of Done

Voor een scrum project is het van belang dat het duidelijk is wanneer een story klaar is, dit is de "definition of done". Mijn story's waren klaar als:

1. De acceptatiecriteria getoetst waren.
2. De story getest was.
3. De story gedocumenteerd was.

Story Workflow

De workflow van een story zag er als volgt uit:

1. De story wordt onderzocht door de ontwikkelaar.
2. De story wordt gepokerd.
3. De story wordt ontwikkeld.
4. De story wordt getest op de Definition of Done, indien deze niet voldoet gaat de workflow verder vanaf stap 4.
5. De story wordt afgerond.

Retrospective

In plaats van een retrospectieve zoals die beschreven wordt door scrum heb ik mijn sprints afgesloten met een meeting waarbij beide begeleiders vanuit Topicus aanwezig waren. In deze meeting werd terug gekeken op de voorgaande sprint en werd vooruit gekeken naar de komende sprint.

KWALITEIT BEWAKING

Het proof of concept moet in de toekomst bruikbaar zijn voor Topicus, hier is het belangrijk dat de kwaliteit van het proof of concept gevalideerd is. Hiervoor zijn de volgende afspraken gemaakt met betrekking tot het proces:

- De stagebegeleiders worden wekelijks op de hoogte gehouden van de voortgang van de ontwikkeling.
- De code wordt gereviewed door een softwareontwikkelaar van Topicus.

Functionele toelichting proof of concept

BESCHRIJVING PROOF OF CONCEPT

De twee ontwikkelde applicaties vormen samen een proof of concept die het onderzoeksverslag ondersteunen. Het proof of concept toont aan, zo ver mogelijk, dat het mogelijk is om met de onderzochte dataverkrijgingstechnieken data te verkrijgen. De applicaties verzamelen gegevens die benodigd zijn bij de aanvraag van een hypotheek en stellen deze beschikbaar door middel van een API.

Dataverkrijgingstechnieken

Het proof of concept verkrijgt informatie uit verschillende dataverkrijgingstechnieken, deze technieken zijn uitgebreid behandeld in het onderzoeksrapport. Het proof of concept maakt gebruik van de volgende dataverkrijgingstechnieken:

- PSD2, het verbinden met API's van banken om transactiegegevens op te vragen.
- Scraping, het verkrijgen van data door html van een website te analyseren.
- OCR, een techniek die karakters en woorden kan herkennen op afbeeldingen.

Limitaties met betrekking tot het implementeren van de dataverkrijgingstechnieken

Tijdens het ontwikkelen van het proof of concept waren we niet in bezit van een OCR-tool en niet in het bezit van een PSD2 vergunning. Op het moment is Topicus in overleg met meerdere OCR partijen en is een aanvraag voor een PSD2 vergunning in behandeling.

Doordat tijdens het ontwikkelen geen toegang was tot een OCR-tool of een PSD2 vergunning is voor het opvragen van gegevens met betrekking tot deze technieken gebruik gemaakt van dummy data:

- De OCR dummy data is in zijn geheel door mij gegenereerd en zal waarschijnlijk grote verschillen bevatten ten opzichte van de werkelijkheid.
- Omdat er geen verbinding gemaakt kan worden met een PSD2 API is er een dummy API ontwikkeld die de SNS PSD2 API imiteert, de dummy API is ontwikkeld door gebruik te maken van de door SNS openbaar gestelde documentatie. De dummy API implementeert geen vorm van authenticatie.

DigiD inlog

Overheidswebsites waar consumenten persoonsgegevens kunnen inzien zijn afgeschermd door middel van DigiD. Het is wettelijk vastgelegd dat een consument alleen zelf in mag loggen op zijn DigiD. De sessie die verkregen wordt van de DigiD inlog is echter wel benodigd voor het verkrijgen van de te scrapen HTML.

Om de sessie van de DigiD inlog te bemachtigen moet binnen de applicatie het volgende mogelijk zijn:

- De applicatie moet een venster met de DigiD inlog kunnen openen.
- De applicatie moet kunnen detecteren wanneer de inlog afgerond wordt.
- De applicatie moet de sessie van inlog uit de cookies van het venster kunnen halen.

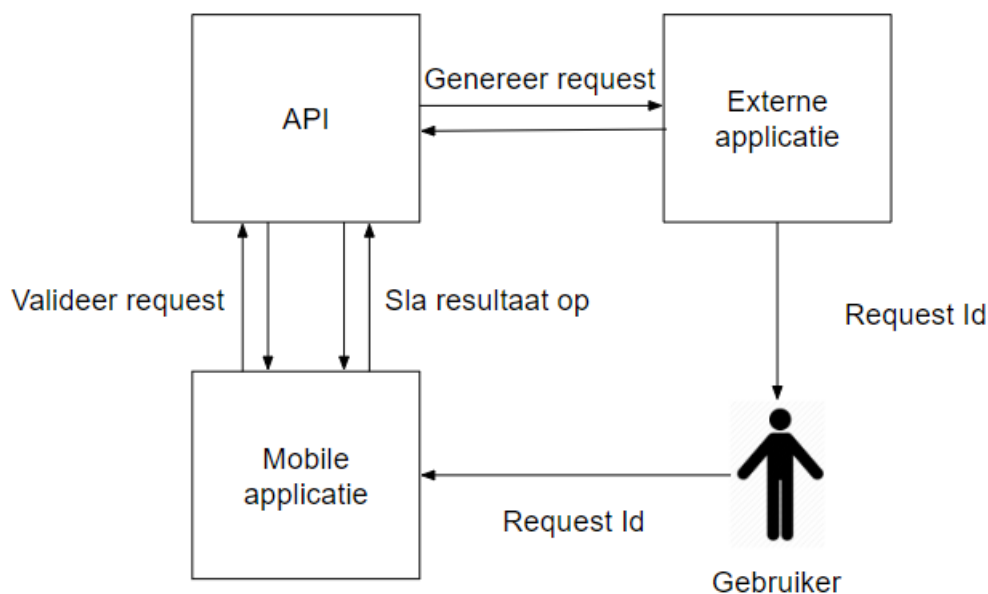
Het is niet mogelijk om dit vanuit een webapplicatie te realiseren doordat CORS⁵ op overheidswebsites niet toegestaan is. De door mij bedachte oplossing voor dit probleem is het gebruik van een mobile applicatie. Een mobile applicatie kan een browsers starten, hierdoor vermijdt je problemen met betrekking tot CORS. Door javascript te injecteren in de browser is het mogelijk om te detecteren wanneer de inlog afgerond is en is het mogelijk om de sessie af te vangen.

De hierboven beschreven methode wordt op dit moment al toegepast in de praktijk door de applicatie Ockto. Doordat deze techniek al succes vol wordt toegepast in de applicatie Ockto wordt de haalbaarheid en doeltreffendheid hiervan verondersteld voor de context van deze opdracht. Om deze reden maakt het ontwikkelde proof of concept geen verbinding met een externe website als bron voor de HTML maar maakt het proof of concept gebruik van statisch opgeslagen HTML.

De twee applicaties

Het proof of concept bestaat uit twee applicaties, een webapplicatie en een mobiele applicatie. De mobile applicatie is zo opgezet dat deze niet zelfstandig het gehele scrapen kan uitvoeren. De mobile applicatie vraagt bij het opstarten om een identifier, het is alleen mogelijk om te scrapen wanneer een valide identifier is opgegeven. De identifier is afkomstig van de webapplicatie, hieraan kan gevraagd worden om een scrape-request te starten, bij het succesvol aanmaken van een scrape-request wordt een identifier van het request teruggestuurd.

Wanneer de mobile applicatie klaar is met het scrapen wordt de eerder opgegeven identifier gebruikt om de verzamelde data aan het matchende scrape request te hangen met een request naar de webapplicatie. De webapplicatie slaat de gegevens op in een dummy database. Het resultaat van een scrape-request kan opgevraagd worden door middel van de API.



Figuur 8: Interactie applicaties

De gekozen aanpak heeft de volgende voordelen:

- De mobile applicatie zou in de toekomst data kun vergaren voor meerdere andere applicaties, bij een scrape request zou bijgehouden kunnen worden voor welke partij de data in te zien moet zijn.
- Alle data benodigd voor een hypotheekaanvraag is op één centrale plek te verkrijgen, namelijk de webapplicatie.
- Aldus AVG worden persoonsgegevens zo min mogelijk opgenomen in het systeem.
 - De persoonsgegevens kunnen nu alleen verkregen worden wanneer deze daadwerkelijk benodigd zijn voor een hypotheekaanvraag (door het genereren van een scrape request), dit komt overeen met de eisen van AVG.
 - Alle (persoons) gegevens kunnen uit de database verwijderd worden wanneer de hypotheekaanvraag afgerond is.

Tooling en frameworks

De webapplicatie is geschreven in C# asp.Net, hiervoor is gekozen omdat Topicus zelf veel projecten ontwikkelt met .Net en omdat ik hier zelf ook ervaring in heb.

De mobileapplicatie is ontwikkeld in Xamarin.Forms, hiervoor is gekozen omdat Xamarin.Forms een hybrid framework is (waardoor met één code base een applicatie geschreven kan worden voor Android en voor IOS) en omdat Xamarin.Forms uitgebreide opties biedt met betrekking tot het manipuleren van een webbrowser, dit is nodig indien in de toekomst de DigiD inlog ontwikkeld wordt zoals beschreven in Het hoofdstuk “DigiD inlog”.

Scrape library

De mobile applicatie zal voor het scrapen vaak niet meer dan dertig HTML-pagina's scrapen, dit is voor een HTML library relatief weinig. Hierdoor was de performance van een scrape library niet relevant voor mij. De relevante eigenschap van een scrape library was voor mij hoe goed gedocumenteerd de library was en hoe makkelijk het was om vragen te beantwoorden tijdens het ontwikkelen. Ik heb twee scrape libraries voor C# gevonden die voldeden aan deze eis, HtmlAgilityPack en Scrapy. Scrapy is een framework dat het gehele proces rondom het scrapen beheert, ook het verkrijgen van de HTML. Scrapy kan echter niet goed overweg met het verkrijgen van HTML dat zich achter een inlog bevindt. Hierom heb ik gekozen om zelf de HTML te verkrijgen en voor het scrapen HtmlAgilityPack te gebruiken. HtmlAgilityPack is een veelgebruikte en goed gedocumenteerde scrape library.

DESIGN KEUZES

De SOLID-principles

Tijdens het ontwikkelen heb ik de SOLID-principles gevolgd. De SOLID-principles beschrijven hoe goed te onderhouden object georiënteerde code geschreven kan worden.

De S staat voor Single Responsibility Principle, dit principe beschrijft dat een klasse altijd één doel moet hebben. Dit heeft als gevolg dat de klasse leesbaarder is en dat wanneer de klasse in een later stadium verandert moet worden er een minder grote kans is dat bugs optreden.

De O staat voor Open/Closed Principle, dit principe beschrijft dat een klasse open moet zijn voor uitbreiding maar gesloten voor aanpassingen. Hierdoor is er een kleinere kans dat code stuk gaat in een later stadium.

De L staat voor Liskov Substitution Principle, dit principe beschrijft dat je iedere klasse zou moeten kunnen vervangen door een klasse die van de klasse afgeleid is. Code zou nooit moeten controleren met wat voor subtype het te maken heeft. Dit voorkomt rare bijeffecten van type casting.

De I staat voor Interface Segregation Principle, dit principe beschrijft dat interfaces klein gehouden moeten worden. Door interfaces te scheiden per deelverzameling krijg je geen “general purpose” interface.

De D staat voor Dependency Inversion Principle, dit principe beschrijft dat hoge modules niet afhankelijk zouden moeten zijn van lage modules. Beiden zouden afhankelijk moeten zijn van abstracties. Hierdoor is het makkelijker om hoge modules te hergebruiken.

Scheiding van projecten

Verbindingen tussen projecten verloopt door middel van interfaces, dit is gedaan zodat Topicus in de toekomst gedeeltes van de applicatie los kan trekken voor hergebruik. De applicatie is opgebouwd uit zeven projecten (exclusief interface projecten), deze projecten zijn:

Project	Beschrijving
SGA.Webapi	Web API die verkeer naar en van de applicatie beheert.
SGA.Logic	Project dat in en output centraliseert.
SGA.PSD2	Project dat verbindingen met PSD2 services beheert.
SGA.Scraper	Project dat de verbinding met de (dummy)database beheert.
SGA.Ocr	Project dat de verbindingen met OCR frameworks/applicaties beheert.
SGA.DummyDatabase	Project dat een database imiteert.
SGA.DummyPSD2Api	Web API die een PSD2 API imiteert.

REQUIREMENTS PROOF OF CONCEPT

Het proof of concept bestaat uit twee applicaties, de webapplicatie en de mobileapplicatie. Hieronder zijn de requirements te zien per applicatie. Wanneer een requirement over data redeneert dan mag voor het vervullen van de requirement gebruik gemaakt worden van dummy data.

Technische requirements webapplicatie

Requirement
Het moet mogelijk zijn om data van een PSD2 service op te vragen door middel van de webapplicatie.
Het moet mogelijk zijn om data van een scrape request op te vragen door middel van de webapplicatie.
Het moet mogelijk zijn om data van een OCR-service op te vragen door middel van de webapplicatie.
Het moet mogelijk zijn om een scrape request te genereren door middel van de webapplicatie.
Het moet mogelijk zijn om een scrape request te valideren door middel van de webapplicatie.
Het moet mogelijk zijn om de resultaten van een scrape request op te laten slaan door de webapplicatie.

System requirements mobileapplicatie

Requirement
Het moet alleen mogelijk zijn om de mobiele applicatie te gebruiken wanneer een valide scrape request ingevoerd wordt.
Het moet mogelijk zijn om data te verzamelen uit de HTML van MijnOverheid.
Het moet mogelijk zijn om data te verzamelen uit de HTML van MijnPensioenOverzicht..
Het moet mogelijk zijn om data te verzamelen uit de HTML van MijnUwv.
Het moet mogelijk zijn om te simuleren dat de verzamelde data naar de webapplicatie gestuurd wordt.

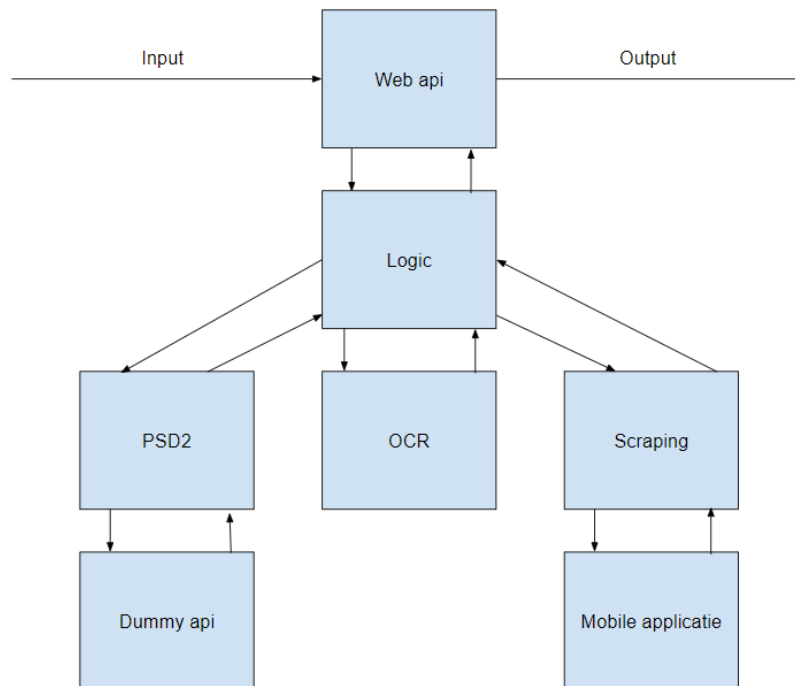
Requirements code

Voor Topicus is het belangrijk dat zij in de toekomst verder kunnen werken met de ontwikkelde applicaties. Op het moment is het nog niet duidelijk of Topicus verder gaat met alle aspecten van het proof of concept. Het proof of concept zou doorontwikkeld kunnen worden tot een volledige applicatie maar er zou ook enkel een deel van het proof of concept gebruikt kunnen worden. Hieruit zijn de volgende requirements gekomen:

- De webapplicatie voldoet aan de C# en .net conventions
- De mobileapplicatie voldoet aan de Xamarin.Forms en C# conventions
- Classes binnen de applicaties zijn opgesplitst op basis van verantwoordelijkheden
- Projecten binnen de applicaties zijn opgesplitst op basis van verantwoordelijkheden
- Projecten binnen de applicaties zijn opgesplitst door middel van interfaces
- Code wordt leesbaar geschreven, dit houdt in het goed kiezen van variables, methode en class namen, goed gebruik maken van formatting en witregels en het limiteren van nesting

Technische toelichting proof of concept

De twee door mij ontwikkelde applicaties vormen samen een proof of concept die het onderzoeksverslag ondersteund. De applicaties verzamelen gegevens die benodigd zijn bij de aanvraag van een hypotheek en stellen deze beschikbaar door middel van een API. In dit hoofdstuk zullen de technische aspecten van het proof of concept toegelicht worden.



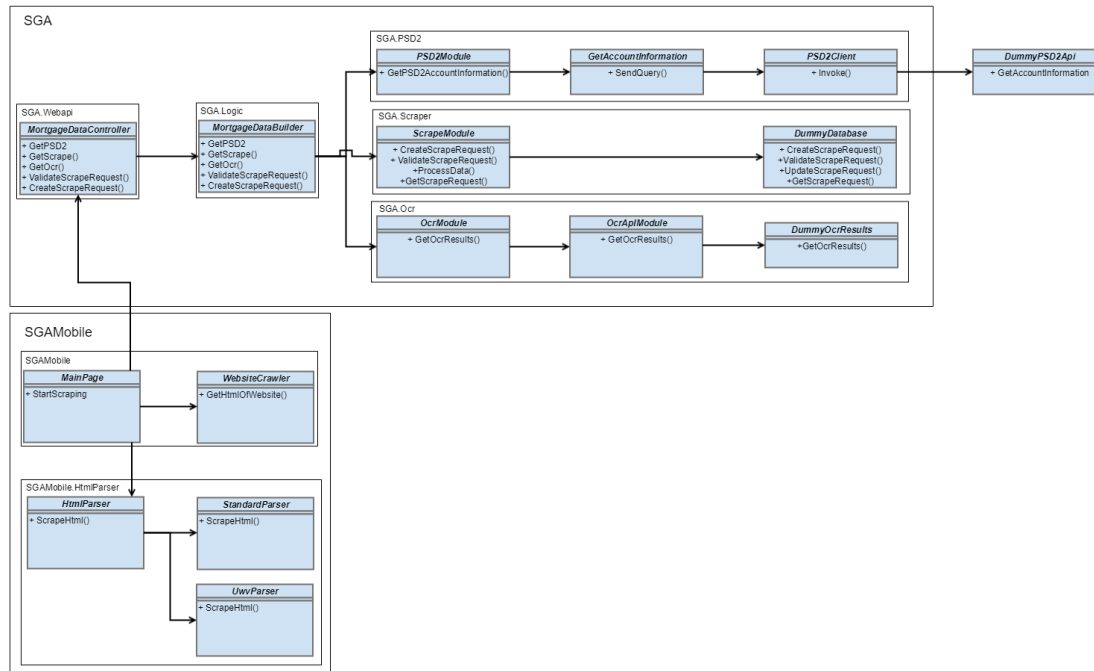
Figuur 9: Overzicht van het proof of concept

Een request komt binnen door middel van de Web API. Afhankelijk van het binnengekomen request laat Web API het Logic project informatie verzamelen. Het Logic project heeft kennis van de verschillende dataverkrijgingstechnieken en biedt opties voor het verzamelen van data door middel van deze dataverkrijgingstechnieken. Iedere dataverkrijgingstechniek heeft een corresponderend project met de verantwoordelijkheid om de dataverkrijgingstechniek uit te voeren. Het PSD2 project staat in verbinding met een dummy API en de het scraping project staat in verbinding met de mobile applicatie.

CLASS DIAGRAM

Hieronder is een class diagram te zien van het proof of concept. De bijlage ClassDiagram.png bevat een beter leesbare versie. Om het overzichtelijk te behouden bevat het onderstaande class diagram niet alles maar highlight het alleen de modules en projecten.

SGA Overview Diagram



Figuur 10: Class diagram van het proof of concept

DE API

De webapplicatie stelt de verkregen informatie beschikbaar door middel van een API. Er is gekozen voor een API doordat Topicus voor haar eigen infrastructuur gebruik maakt van de microservices structuur, het is makkelijk om een API hierin te verwerken. Hieronder staat een lijst van de mogelijke calls op de API.

Route	Beschrijving	Methode	Input	Output
mortgageData/psd2	Opvragen data van PSD2 services	GET	• nameOfUser • nameOfBank	Data van PSD2 services
mortgageData/scrape	Opvragen data van een scrape request	GET	• scrapeRequestid	Data van scraping
mortgageData/scrape/validate	Het valideren van een scrape request	GET	• requestId	Boolean
mortgageData/scrape/	Het creëren van een scrape request	POST	-	Requestid
mortgageData/scrape/	Het toevoegen van data aan een scrape resultaten	PUT	• requestId • scrapeResults	-
mortgageData/ocr	Opvragen data uit OCR tools	GET	• userId	Data van OCR

Omdat de API-deel is van een proof of concept maakt deze geen gebruik van authenticatie. Dit heeft ontwikkeltijd bespaard en maakte het testen van de applicatie eenvoudiger.

BESCHRIJVINGEN DATAVERKRIJGINGSTECHNIKEN

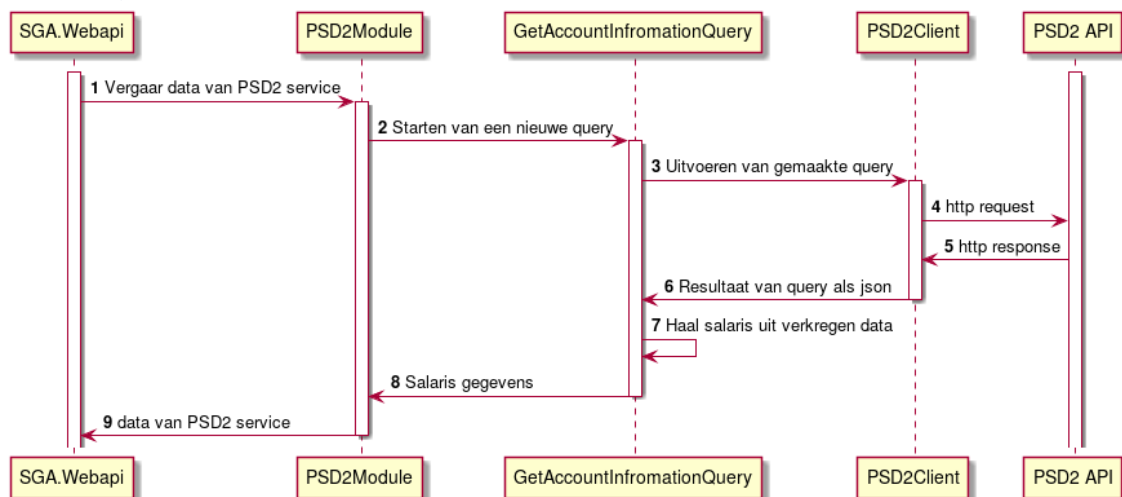
In dit hoofdstuk zal per dataverkrijgingstechniek beschreven worden hoe de techniek geïmplementeerd is.

PSD2

Het PSD2 project heeft als doel het opvragen van gegevens van PSD2 API's. Op het moment beschikt Topicus nog niet over een PSD2 vergunning, om deze reden is het onmogelijk om een verbinding op te stellen met een PSD2 API. Omdat er geen verbinding gemaakt kan worden met een PSD2 API is er een dummy API ontwikkeld die de SNS PSD2 API imiteert, de dummy API is ontwikkeld door gebruik te maken van de door SNS openbaar gestelde documentatie. De dummy API implementeert geen authenticatie.

Door de data te analyseren kan veel interessante informatie voor de verstrekking van een hypotheek verkregen worden uit de verkregen gegevens van de PSD2 API. Op het moment is de applicatie alleen in staat om salarissen uit de transactie geschiedenis van een gebruiker te filteren. Deze filtering wordt gedaan door te kijken of het type van de transactie overeenkomt met "Salaris" of "Salary".

Sequence diagram



Figuur 11: Sequence diagram van het PSD2 project.

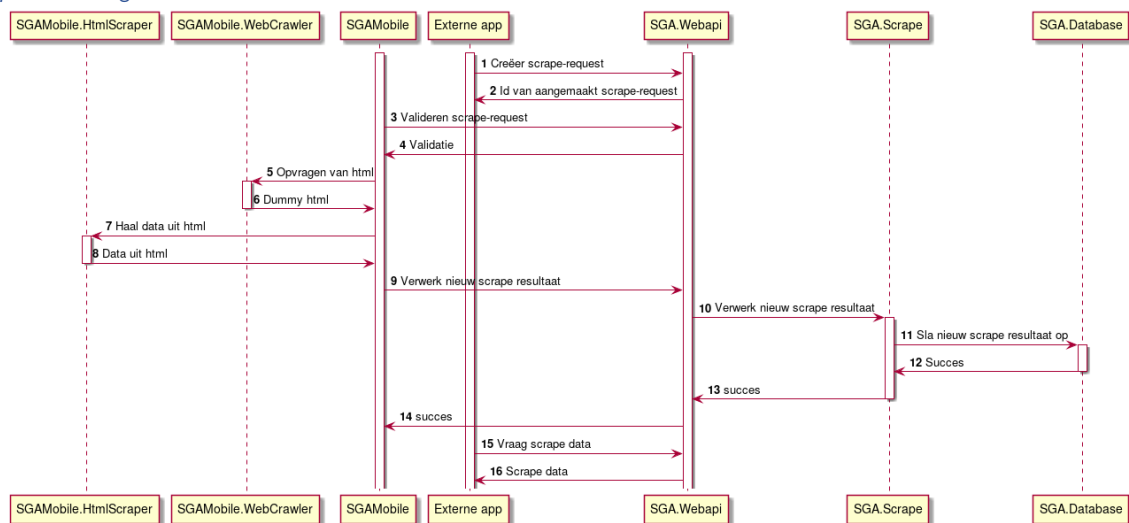
De verbinding met externe PSD2 API's verloopt door middel van de PSD2client.cs class en door middel van query classes. Een query class beschrijft een HTTP GET request naar een PSD2 service en gebruikt de PSD2Client om de gemaakte query uit te voeren over HTTP. De implementatie van een query is vastgelegd in IQuery.cs.

Scraping

Het scrapen heeft als doel om data te lezen uit de HTML van overheidswebsites zoals mijn.overheid.nl/ en www.mijnpensioenoverzicht.nl/. Zoals beschreven in het hoofdstuk “Functionele toelichting proof of concept - Dataverkrijgingstechnieken” maakt de scraper gebruik van in de applicatie opgeslagen HTML. Deze HTML wordt verwerkt met de library HtmlAgilityPack.

Voordat de mobile applicatie begint met scrapen moet een identifier van een scrape-request meegegeven worden. Scrape-requests worden beheerd door de webapplicatie. Na het valideren van de identifier wordt de data geëxtraheerd uit de HTML. De verkregen data wordt opgestuurd naar de webapplicatie, deze slaat de gegevens op in een dummy database.

Sequence diagram



Figuur 12: Sequence diagram van het scrape proces. De externe app zal in de praktijk waarschijnlijk het product FORCE van Topicus zijn

Het scrapen van de html

Data wordt vaak op een vergelijkbare wijze gepresenteerd op websites. Deze wijze van presentatie kan zowel overeenkomen tussen verschillende websites, als tussen verschillende pagina's binnen websites. Hiermee heb ik rekening gehouden tijdens het ontwikkelen van de scraper. In de helper class `DataElementLocationHelper.cs` wordt bijgehouden welke data-elementen verkregen moeten worden uit welke urls. Per data-element wordt bijgehouden hoe dit element gevonden kan worden in de html, hiervoor wordt onderscheid gemaakt van data-elementen die wel en die niet binnen een lijst staan. De class `StandardParser.cs` verwerkt de html op basis van de specificaties verkregen van de `DataElementLocationHelper.cs` class.

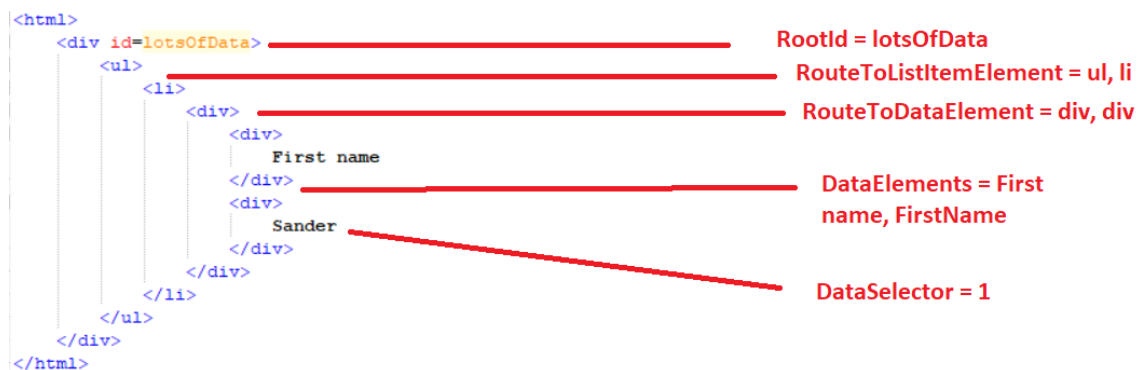
Verwerken van meerdere data-elementen binnen een lijst

Wanneer één of meerdere te verwerken data-elementen zich bevinden binnen een lijst dan kunnen deze beschreven worden met de `LocationOfElementsInsideList.cs` class.

`LocationOfElementsInsideList.cs` beschrijft een element dat zich binnen een lijst bevindt, hiervan wordt het volgende bijgehouden:

- `StartId`, het id van het html element waar de scraper op moet beginnen.
- `DataSelector`, geeft aan op welke positie het dataelement staat binnen een list item.
- `RouteToListItemElement`, de route door de html naar het list element vanaf het startId.
- `RouteToDataElement`, de route door de html naar de combinatie van datalabel en data vanaf het list element.
- Data-element en label, key value pair van het label en de naam van het data-element.

Hieronder staat een simpel voorbeeld



Figuur 13: Voorbeeld HTML-data-elementen binnen een lijst

Verwerken van een enkel data-element

Wanneer een te verwerken data-element zich niet binnen een lijst bevindt dan kan deze beschreven worden met de `LocationOfElementOutsideList` class. `LocationOfElementOutsideList` beschrijft de locatie van een enkel element, hiervan wordt het volgende bijgehouden:

- `StartId`, het id van het html element waar de scraper op moet beginnen.
- `RouteToDataElement`, de route door de html naar de data vanaf het root element. Opgebouwd uit key value pairs van element type en index.
- `DataElement`, naam van het data-element.

Hieronder staat een simpel voorbeeld



Figuur 14: Voorbeeld HTML enkel data-element

Verwerken van UWV data-elementen

De data die verkregen wordt van UWV is uniek opgebouwd en hier hierdoor zijn eigen custom scrape methodiek nodig. De data van UWV die verkregen wordt is een geschiedenis van salarissen. Deze salarissen worden weergegeven in blokken per werknemer waaronder een consument gewerkt heeft. Voordat de applicatie begint met scrapen is het onmogelijk om te weten hoeveel blokken van werknemers verwerkt moeten worden, hierdoor kan dit niet gescraped worden met de standaard scrape methodiek.

Doordat de UWV de enige website is met deze unieke opbouw is de methodiek niet generiek opgenomen in de applicatie.

```
private static IEnumerable<Salary> ParseSalariesHistoryList(HtmlDocument doc)
{
    var salariesFromHtmlList = new List<Salary>();
    var rootIdElement = doc.GetElementById(UwvDataLocationHelper.SalaryHistoryRootId);
    var salarySections = rootIdElement.Elements("section").ToArray();

    //parse every company listed
    foreach (var salarySection in salarySections)
    {
        var panelBody = salarySection.Element("div");
        var headerElement = panelBody.Element("dl");

        //get name of creditor
        var creditor = headerElement.Elements("dd").ToArray()[0].InnerHtml;
        //get payrollTaxNumber (dutch = loonheffingsnummer)
        var payrollTaxNumber = headerElement.Elements("dd").ToArray()[1].InnerHtml;

        var tableBodySection = panelBody.Element("table").Element("tbody");
        var salariesElement = tableBodySection.Elements("tr");

        //parse every salary the company has listed
        var salariesDictionary = new Dictionary<string, string>();
        foreach (var salary in salariesElement)
        {
            //get date of salary
            var date = salary.Elements("td").ToArray()[0].InnerHtml;
            //get amount of salary
            var amount = salary.Elements("td").ToArray()[2].InnerHtml;

            salariesDictionary.Add(date, amount);
        }
        salariesFromHtmlList.Add(new Salary(creditor, payrollTaxNumber, salariesDictionary));
    }

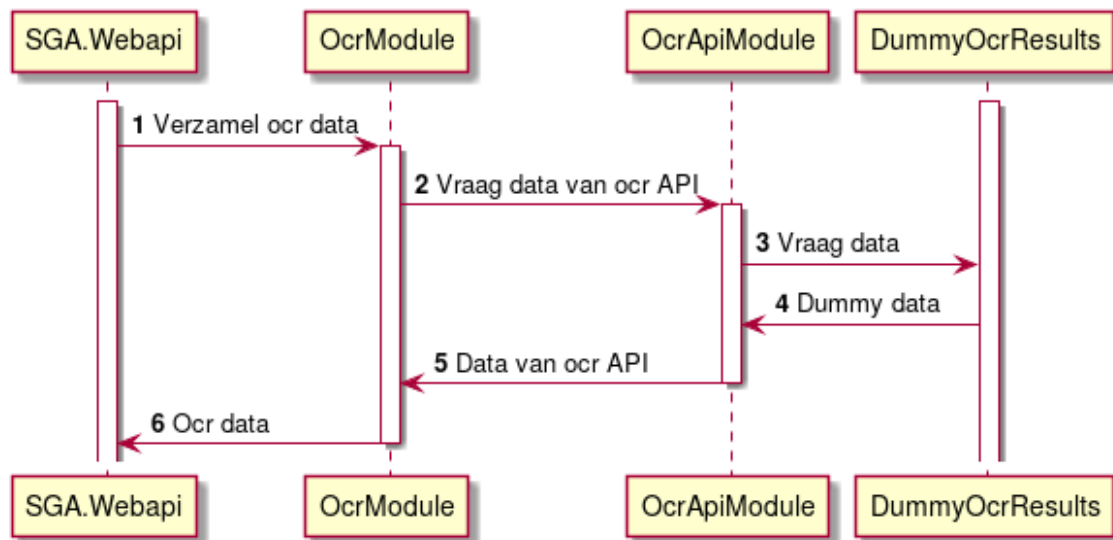
    return salariesFromHtmlList;
}
```

Figuur 15: Scrapen van UWV-code voorbeeld

OCR

Het gebruik van OCR heeft als doel om automatisch data te verkrijgen van documenten die opgestuurd worden bij een hypotheek aanvraag. Tijdens het ontwikkelen van de proof of concept applicatie was er geen toegang tot een OCR-tool, hierdoor wordt er gebruik gemaakt van dummy data. De applicatie imiteert een connectie met een OCR-tool en 'verkrijgt' op die manier de informatie.

Sequence diagram



Figuur 16: Sequence diagram OCR-proces

TESTEN

De applicaties die ontwikkeld zijn tijdens de afstudeerstage zijn onderdeel van een Proof of Concept. Bij een Proof of Concept is het belangrijk om de haalbaarheid aan te tonen, de stabiliteit en foutgevoeligheid zijn minder belangrijk. Hierdoor heb ik geen grote focus gelegd op testing. Ik heb functionaliteiten voornamelijk getest door de applicaties te starten met verschillende argumenten.

Voorbeelden hiervan zijn:

- Scrapen voor verschillende gebruikers van verschillende sites
- PSD2 informatie ophalen van verschillende gebruikers
- OCR info ophalen van verschillende gebruikers

Indien ik meer ontwikkeltijd gehad zou hebben voor het Proof of Concept dan had ik het parsen van de HTML willen testen met gebruik van Unit Tests. Dit is omdat het aantonen dat data in verschillende situaties correct verkregen kan worden uit de HTML meewerkt aan het bewijzen van de haalbaarheid van de applicatie.

Initiatief OCR-tool vergelijking

Topicus heeft op het moment geen toegang tot een OCR-tool, op het moment is er een initiatief binnen Topicus om de OCR-tools op de markt met elkaar te vergelijken. Ik ben betrokken geraakt bij dit initiatief door mijn kennis met betrekking tot OCR en doordat ik een idee heb waar rekening mee gehouden moet worden om een OCR-tool aan te sluiten op het domein van Topicus.

Het initiatief heeft de volgende doelen:

- Eén of meerdere OCR-tools vinden die voor Topicus inzetbaar zijn
- Oriënteren op de OCR-technologie
- Awareness, met betrekking tot de mogelijkheden van OCR, creëren binnen Topicus en klanten van Topicus

TRAJACT VAN INITIATIEF

Het initiatief is opgedeeld in vier fases, momenteel bevindt het initiatief zich aan het einde van fase 1. Het initiatief bestaat uit de volgende fases:

Fase	Beschrijving	Doel
Fase 1	Kennis maken met verschillende OCR partijen	Connectie opstellen met OCR partijen op de markt
Fase 2	Verwerken van een tiental geanonimiseerde documenten op een trial omgeving	Aantoonbaar bewijs verzamelen van de potentie van de OCR partijen
Fase 3	Verwerken van duizenden documenten van een productie omgeving en op basis hiervan de verschillende partijen te vergelijken	Een uitgebreid onderbouwde vergelijking van de verschillende OCR partijen
Fase 4	Resultaten delen met Topicus en klanten van Topicus	Awareness, met betrekking tot de mogelijkheden van OCR, creëren binnen Topicus en klanten van Topicus

MIJN BIJDRAGE

Zoals hierboven vermeld zit het initiatief nog in de oriënterende fase 1, deze fase bestaat uit het kennis maken met verschillende OCR partijen van over de hele wereld door middel van e-meetings⁶.

Mijn bijdrage bij de eerste fase bestaat uit:

- Overleggen bijwonen
- Zoeken naar OCR partijen
- E-meetings organiseren en plannen
- Contact richting OCR partijen behouden
- E-meetings bijwonen

Mijn rol in de e-meetings was vanuit een technische hoek vragen beantwoorden en zorgen dat de volgende vragen beantwoord werden door de OCR partij:

- Wat is er nodig van Topicus om een trial omgeving op te zetten?
- Hoe kan Topicus de tool aansluiten op haar domein?
 - Beschikt de tool over een Api?
- Kan de tool documenten identificeren of moet die gedaan worden voordat de documenten verwerkt kunnen worden?
- Hoe werkt het leren van de tool?
 - Leert de tool constant of in intervallen?
 - Moet de partij waar de tool staat acties ondernemen voor het leren? (Zoals medewerkers inzetten die velden corrigeren)
- Hoe veel menselijke interacties verwachten zij dat benodigd is per document type?
- Hoe zeker is de tool over zijn extractie en is hiervoor een minimum die ingesteld kan worden?
- Met welke type documenten verwachten ze dat de tool overweg kan?
 - Op korte en lange termijn

Product Conclusie

Uit mijn afstudeerstage zijn voor Topicus drie belangrijke resultaten gekomen, het onderzoeksverslag, het proof of concept en een bijdrage aan het vergelijken van verschillende OCR-tools. In dit hoofdstuk wordt beschreven wat de eigenschappen van deze resultaten zijn.

Het onderzoeksverslag

Het onderzoeksverslag heeft de volgende aspecten inzichtelijk gemaakt:

- De werking van OCR en de verschillen tussen verschillende OCR-technieken
- Een lijst van potentiële bronnen voor het verkrijgen van data benodigd voor een hypotheek
- Een lijst van benodigde dataelementen voor het verkrijgen van een hypotheek
- Een vergelijking van drie verschillende dataverkrijgingstechnieken

Het proof of concept

Het proof of concept voldoet aan de gestelde requirements. Het proof of concept heeft de volgende functionaliteiten:

- Data van een realistische dummy PSD2 service kan opgevraagd worden
- Data die weergegeven wordt op overheidswebsites kan opgevraagd worden
- Data van een dummy OCR-resultaat kan opgevraagd worden

Het proof of concept heeft de volgende eigenschappen:

- Projecten zijn generiek opgesteld waardoor deze herbruikbaar zijn en dus gebruikt kunnen worden in andere applicaties
- Het proof of concept volgt de SOLID-principles waardoor het proof of concept onderhoudbaar is

Vergelijking van verschillende OCR-tools

Het initiatief dat verschillende OCR-tools vergelijkt voor Topicus heeft op het moment de volgende 7 potentiële OCR partijen voor Topicus gevonden:

- DocuCharm
- Ephisoft Transact
- Hyarchis
- Hypatos
- Rossum ai
- Xtracta

Aanbevelingen

Het door mij gebouwde proof of concept is een basis die veel mogelijkheden biedt, in dit hoofdstuk benader ik een aantal ideeën die ik Topicus aanbeveel uit te zoeken.

PRODUCTIE DATA PSD2 EN OCR

Het proof of concept maakt gebruik van dummy data. Het verkrijgen van de dummy data en de structuur van de dummy data is gemaakt om zo veel mogelijk overeen te komen met hoe dit er later uit zal komen te zien. Echter zal het gebruik van productiedata waarschijnlijk verschillen van de dummy data opbouw en zal dit hierdoor nieuwe inzichten en/of problemen kunnen geven. Hierom adviseer ik om zo snel mogelijk te testen met productiedata.

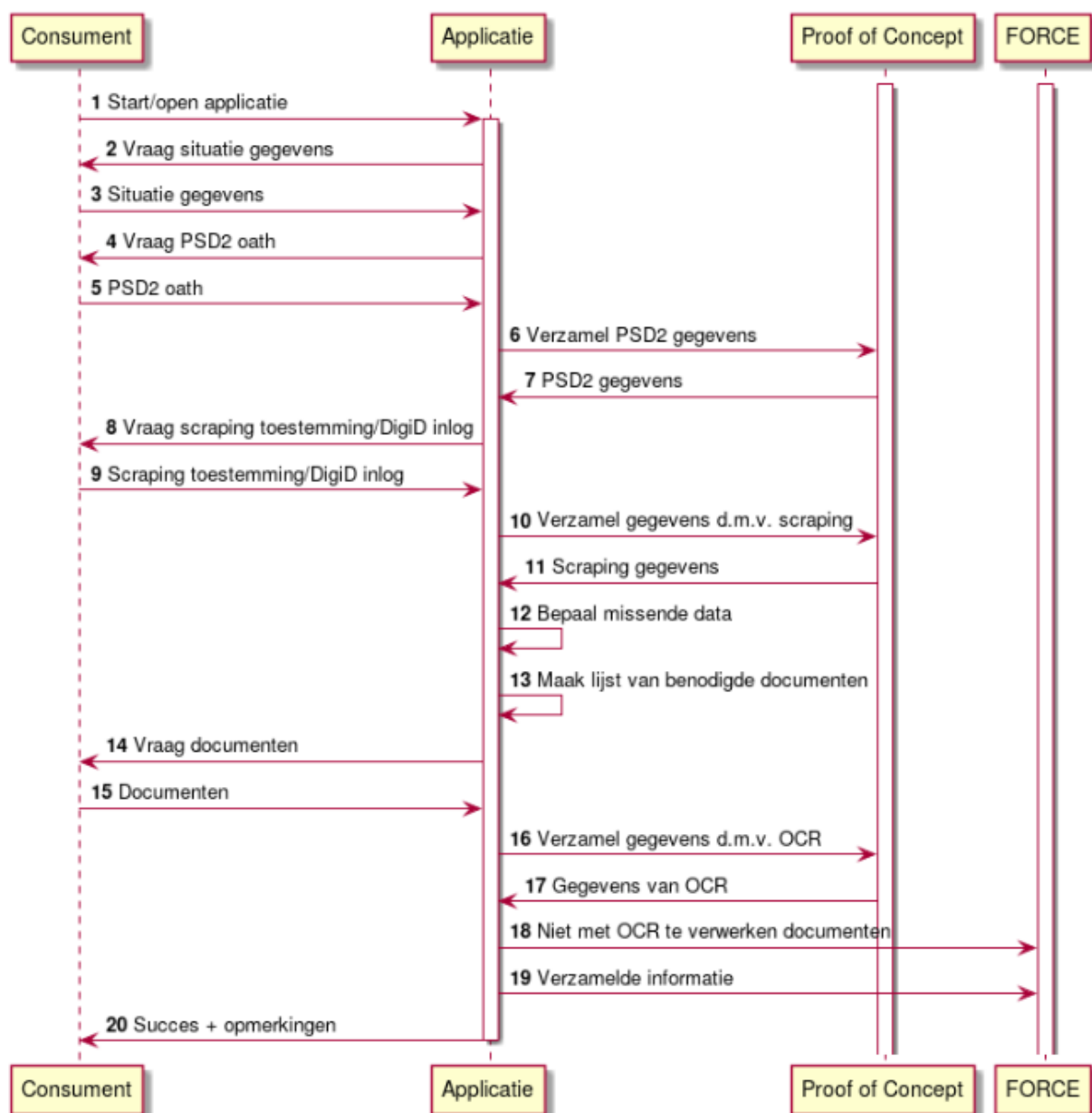
PROCES DEKKENDE APPLICATIE

Op het moment zijn er veel lange stappen benodigd om de data uit een document te halen. Een consument moet weten welke documenten benodigd zijn, dan moeten deze aangeleverd worden aan de adviseur, de adviseur stuurt de documenten naar de bank(en) en als laatste haalt een medewerker bij een bank de data handmatig uit het document. Ik adviseer om het aantal stappen benodigd voor het verwerken van een document te minimaliseren in zowel hoeveelheid als in de tijd benodigd. Ik denk dat dit het beste gerealiseerd kan worden met een applicatie die zowel de consument ondersteund met het aanleveren van de documenten als data vergaart door middel van de door mij gebouwde applicatie.

De applicatie

In de applicatie vinden de volgende stappen plaats:

1. De consument voert zijn situatie in
2. De applicatie bepaald welke data benodigd is, op basis van de opgegeven situatie
3. De applicatie laat de gebruiker data verzamelen door middel van PSD2
4. De applicatie laat de gebruiker data verzamelen door middel van scraping
5. De applicatie bepaald een lijst van documenten waar de informatie in staat die benodigd is maar niet verkregen kon worden met PSD2 of scraping
6. De consument uploadt de gevraagde documenten op de applicatie
7. Indien mogelijk verwerkt de applicatie de documenten met gebruik van OCR en wanneer dit niet mogelijk is worden de documenten verwerkt aldus de oude situatie.



Figuur 17 Diagram voor aanbevolen applicatie

De voordelen van deze aanpak zijn:

- Het proces verloopt sneller
- Een consument is in één keer klaar met het proces
- Niet/minder wachten op handmatige arbeid
- Er is een kleinere kans dat de consument de verkeerde documenten aanlevert
- Foutmeldingen of opmerkingen over de geleverde informatie kan direct aan de consument getoond worden

PSD2 DATA-ANALYSE

Door verdere analyse te doen op de transactiegegevens verkregen door middel van PSD2 kan er meer gezegd worden over de financiële situatie van de consument. Op basis van de financiële situatie van de consument kunnen de voorwaarden van de hypotheek gewijzigd worden. Aspecten die geanalyseerd zouden kunnen worden zijn:

- Staat de consument regelmatig rood?
- Spaart de consument?
- Spendeert de consument net zoveel als dat hij/zij ontvangt?
- Heeft de consument op het moment van de aanvraag een hoog saldo?
- Ontvangt de consument voor een langere periode inkomsten uit dezelfde bron?
- Is er een consistente stijging of daling in het te besteden geld van de consument?

Dit idee is mogelijk te “eng” voor consumenten doordat er nog meer persoonlijke informatie verkregen wordt, dit zal hiervoor uitgebreider onderzocht moeten worden.

Reflectie

Aan het einde van mijn stage heb ik kritisch gereflecteerd over de gehele stage, dit heb ik gedaan met de STARR-methode. Doordat de situatie, mijn taken, mijn acties en het resultaat uitgebreid beschreven zijn in dit verslag heb ik deze enkel kort benaderd.

SITUATIE

Topicus heeft een systeem dat het proces rondom het verstrekken van een hypotheek grotendeels automatiseert. In de huidige situatie is de langste stap in het proces het handmatig verwerken van de documenten die geleverd worden door consument. Het doel van de stage was het onderzoeken van technieken om deze stap geheel of gedeeltelijk te automatiseren en dit te ondersteunen met een Proof of Concept

TAAK

Mijn taken tijdens de stage bestonden uit het uitvoeren van het onderzoek en het bouwen van het proof of concept.

ACTIE

Mijn activiteiten kort opgesomd waren:

- Onderzoeken huidige situatie
- Onderzoeken dataverkrijgingstechnieken
- Bouwen Proof of Concept
- Topicus adviseren
- Topicus ondersteunen met het uitzoeken van een OCR-tool

RESULTAAT

Het resultaat van mijn stage is een uitgebreid onderzoeksverslag en minimal viable product Proof of Concept. Het Proof of Concept benaderd de core functionaliteiten en bewijst, zo ver mogelijk zonder toegang tot een OCR-tool en een PSD2 API, dat het verzamelen van data mogelijk is zoals beschreven in het onderzoeksverslag.

REFLECTIE

Terugkijkend op mijn stage ben ik zeer tevreden met hoe het verlopen is, alhoewel ik het jammer vind dat het Proof of Concept niet heel uitgebreid is.

Tijdens het project heb ik geleden onder een aantal obstakels, onder andere het overlijden van mijn opa gevolgd door een goede twee weken ziek zijn. Samen met Topicus ben ik hier goed mee omgegaan.

Gedurende mijn afstudeerstage heb ik relatief weinig ontwikkeltijd gehad. Ik ben veel bezig geweest met onderzoeken, overleggen en afstemmen. Voor mijn gevoel heb ik dan ook een grote persoonlijk ontwikkeling gehad in deze aspecten door de ervaring die ik hier bij heb opgedaan. Hier ben ik zelf erg tevreden mee doordat ik voor mijn gevoel in deze aspecten achterliep relatief tot mijn mede studenten.

Het eindresultaat was waarschijnlijk beter geweest als ik een week of twee minder besteed had aan het onderzoek en deze tijd besteed zou hebben aan het ontwikkelen. Het huidige Proof of Concept, samen met de documentatie, is in mijn mening echter wel voldoende om mijn technische kennis te bewijzen voor het afstuderen.

Ik ben erg tevreden met de persoonlijke ontwikkeling die ik behaald heb tijdens het afstuderen.

Bronnenlijst

- Basisregistratie Adressen en Gebouwen (BAG) (1.0.0)*. (sd). Opgehaald van Overheid.nl:
<https://bag.basisregistraties.overheid.nl/restful-api?articleid=1927964#>
- Betaalrichtlijn PSD2. Wat kun je ermee?* (sd). Opgehaald van KVK: <https://www.kvk.nl/advies-en-informatie/financiering/betaalrichtlijn-psd2-wat-kun-je-ermee/>
- Databankrecht*. (sd). Opgehaald van Rijksdienst voor Ondernemend Nederland:
<https://www.rvo.nl/onderwerpen/innovatief-ondernemen/octrooien-ofwel-patenten/octrooi-anders-beschermen/databankenrecht>
- De Volksbank PSD2 sandbox*. (2019, Maart). Opgehaald van De Volksbank N.V.:
<https://openbanking.devолksbank.nl/build/image/pdf/devолksbank-psd2-sandbox.pdf>
- Ephesoft, I. (2016, 11 5). *Ephesoft Transact Machine Learning Data Extraction - Part 2*. Opgehaald van Youtube: <https://www.youtube.com/watch?v=cXlfcpryRbc>
- Infrd. (2018, Januari 11). *Templates vs Machine Learning OCR*. Opgehaald van Becominghuman:
<https://becominghuman.ai/templates-vs-machine-learning-ocr-b8ffe92d1ce3>
- Limburg, C. (2017, Oktober 5). 'Waarom mag Ockto wel data via DigiD opvragen en Nationale Hypotheekbond niet?'. Opgehaald van Amweb: <https://www.amweb.nl/financiele-planning/nieuws/2017/10/waarom-mag-ockto-wel-data-via-digid-opvragen-en-nationale-hypotheekbond-niet-101103510>
- Optical Character Recognition (OCR) - How it works*. (2012, februari 05). Opgehaald van nicomsoft:
<https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>
- PSD2*. (sd). Opgehaald van De nederlandse bank:
<https://www.dnb.nl/betalingsverkeer/psd2/index.jsp>
- Saund, E. (2012). *United States Patentnr. 537,729*.
- Schantz, H. F. (1982). *The history of OCR, optical character recognition*. Recognition Technologies Users Association; 1st edition (1982).
- The First OCR System: "GISMO"*. (sd). Opgehaald van HistoryofInformation:
<http://www.historyofinformation.com/detail.php?entryid=885>
- Topicus*. (sd). Opgehaald van Dutch Software: <https://www.dutchsoftware.nl/isvs/topicus/>
- Topicus*. (sd). Opgehaald van Banken.nl: <https://www.banken.nl/partners/topicus>
- Topicus B.V. (sd). Opgeroepen op 09 14, 2017, van Topicus: <https://www.topicusfinance.nl/wp-content/uploads/2015/06/Topicus-FORCE-Framework-whitepaper.pdf>
- Topicus B.V. (sd). *Topicus: Tehnologie met impact op ons leven*. Opgeroepen op 09 14, 2017, van Informatiestad: <http://www.informatiestad.nl/verhalen/case/15973/topicus-technologie-met-impact-op-ons-leven/>
- Using Zonal OCR to Extract Data Fields From Scanned Documents*. (sd). Opgehaald van Docparser:
<https://docparser.com/blog/zonal-ocr/>
- Wat betekenen de coderingen op mijn overzicht?* (sd). Opgehaald van BKR:
<https://www.bkr.nl/veelgestelde-vragen/wat-betekenen-de-coderingen-op-mijn-overzicht/>
- Wat is open data*. (sd). Opgehaald van Overheid:
<https://web.archive.org/web/20120501095346/http://data.overheid.nl/>