Scriptie

Het implementeren van een digitaal ondertekenen functionaliteit bij B-ware Business Software

Versie 2.0 | Defintief

Colofon

Versie 2.0
Status Defintief
Auteur(s) Derk van der Heide
Kenmerk -
Laatst gewijzigd 30 oktober 2022
Bestand Scriptie

Versiebeheer

Versie	*Wijzigingen*	*Datum*
0.1	Concept hoofdstuk indeling.	5 Maart 2022
0.2	Inleiding, opdracht en procesbeschrijving toegevoegd.	19 April 2022
0.3	Feedback docent verwerkt.	3 Mei 2022
0.4	Onderzoek toegevoegd en ontwerp onderdelen toegevoegd. Realisatie hoofdstukken verder gedefinieerd	15 Mei 2022
0.5	Onderzoek aangevuld met het implementatie onderdeel van het onderzoek. Feedback Ruud verwerkt op hoofdstuk Opdracht.	19 Mei 2022
0.6	Ontwikkelomgeving toegevoegd en feedback verwerkt begeleider.	20 Mei 2022
0.7	Prototype, diagrammen, interactie systemen, technische onderdelen.	21 Mei 2022
0.8	Conclusie onderzoeken toegevoegd, requirements beter gedefinieerd op conclusies onderzoek.	22 Mei 2022
0.9	Feedback verwerkt van Ruud, verwijzingen aangevuld voor bronvermelding, project methodes herschreven, algemene schoonheidscontrole.	23 Mei 2022
1.0	Usecase, systeemstructuur toegevoegd	25 Mei 2022
1.1	Onderzoek nieuwe constructie, feedback verwerkt.	29 Mei 2022
1.2	Onderzoek uitgebreid en feedback opdrachtgever verwerkt.	30 Augustus 2022
1.3	Ontwerp opnieuw verwerkt gebaseerd op wijzigingen die zijn gemaakt, feedback docent en opdrachtgever verwerkt.	16 Augustus 2022
1.4	Feedback verwerkt, realisatie toegevoegd.	23 Augustus 2022
1.5	ClownPDF ontwerp aangevuld informatie implementatie stappen. Front-end realisatie verder uitgewerkt. Realisatie backend aangevuld. Ontwerp database veranderd, belicht dat tabellen uit analyse komen.	1 Oktober 2022
1.6	Gros taalfouten eruit gehaald, reflectie aangepast, randvoorwaarden en eisen uit elkaar getrokken en aangescherpt, ontwerp en realisatie afgemaakt. Verduidelijking rollen toegevoegd aan ontwerp.	6 Oktober 2022
1.7	Concept verslag afgewerkt, taalfouten en spellingsfouten eruit gehaald, begrippenlijst aangepast, inhoudsopgave bijgewerkt en algemene stijl controle.	7 Oktober 2022
1.8	Feedback op concept verwerkt en laatste hoofdstukken uitgeschreven.	17 Oktober 2022
2.0	Laatste feedback verwerkt van docenten en begeleiders. Test methoden toegevoegd aan realisatie. Voorbeelden toegevoegd aan PDFClown Realisatie. Voorwoord, samenvatting en conclusie toegevoegd. Hoofdstukken realisatie andere volgorde gegeven.	29 Oktober 2022

Voorwoord

Voor u ligt het afstudeerverslag “Het implementeren van een digitaal ondertekenen functionaliteit bij B-ware Business Software”. Het onderzoek voor dit verslag naar digitaal ondertekenen en de functionaliteit daarvan is uitgevoerd bij B-Ware Business Software dat tevens fungeerde als afstudeerbedrijf.

Dit verslag is geschreven tijdens mijn afstudeerperiode bij het genoemde bedrijf om mijn studie “HBO-IT Software Engineer” bij het Saxion af te ronden. Dit heb ik gedaan van 7 januari tot 18 november ‘22. Deze periode was ingedeeld in een onderzoekfase en een realisatiefase. Uiteindelijk hebben beide fases bijgedragen aan het maken van een proof-of-concept om aan te kunnen tonen dat het mogelijk is voor B-ware Business Software om pdf-documenten digitaal te ondertekenen op een manier waarop deze documenten rechtsgeldig zijn in Nederland en binnen de EU.

Ik wil mijn mede afstudeerders Robin Dittrich, Maurijn Besters en Tycho Engberink bedanken voor de feedback, advies en steun die zij hebben geleverd tijdens onze meetings.

Ik wil graag mijn afstudeerdocent Ruud Greven bedanken voor de feedback en support. Daarnaast wil ik ook René van den Nieuwenhoff bedanken voor de feedback op de conceptversie en het beantwoorden van vragen omtrent afstuderen.

Ook binnen B-ware Business Software zijn er een aantal mensen die een prominente rol in mijn afstuderen hebben gespeeld die ik graag zou willen bedanken. Irene Timmerman, Paul Sproates-Gorter en Chris de Jong-Pobedinsky in het speciaal voor de ondersteuning en begeleiding.

Ik wens u veel leesplezier toe en hoop dat het inzicht biedt in mijn afstudeerproces en de mogelijkheden van digitaal ondertekenen.

Derk van der Heide

Enschede, 25 oktober 2022

Samenvatting

Dit document bevat het afstudeerverslag van Derk van der Heide bij B-ware Business Software waar onderzoek is gedaan naar digitaal ondertekenen in PDF Documenten.

B-ware Business Software heeft de student gevraagd een analyse uit te voeren naar digitaal ondertekenen en het genereren van PDF documenten en aan de hand daarvan een ontwerp te maken voor een onderteken functie. Het ontwerp is gerealiseerd in een proof-of-concept waarin kan worden aangetoond dat een PDF-document digitaal kan worden ondertekend.

Het proof-of-concept bestaat uit een front-end, backend en database. Het systeem is gekoppeld met Azure AD voor identificatie en authenticatie van de gebruikers.

De onderteken functies worden verwerkt in de backend zodat het bij doorontwikkeling eventueel kan worden geïntrigeerd met de bestaande systemen van het afstudeerbedrijf.

Digitaal ondertekenen is een juridisch proces en daarom zijn er wetten en richtlijnen omtrent digitaal ondertekenen waaraan de functie moet voldoen. Het is belangrijk dat er een digitaal certificaat in een document wordt geplaatst bij het ondertekenen om de inhoud te versleutelen. Dit wordt versleuteld zodat er geen aanpassingen kunnen worden gemaakt aan de inhoud van het document.

Het is ook belangrijk dat het certificaat dat wordt geplaatst gebonden is aan een natuurlijk persoon. Dit kan worden gedaan met een vertrouwde identiteit provider. Een voorbeeld van zo'n provider is DigiD, echter mag DigiD alleen gebruikt worden voor applicaties voor de publieke sector. Vanuit de Nederlandse staat is er een alternatieve methode bedacht genaamd EHerkenning voor de privé sector. DigiD en EHerkenning gebruiken beide dezelfde inlog protocollen waarvan de documentatie online beschikbaar is.

Om dat na te bootsen in het proof of concept is Azure AD geconfigureerd om dezelfde protocollen te gebruiken als EHerkenning en DigiD zodat bij doorontwikkeling de overstap gemakkelijker gemaakt kan worden.

Inhoudsopgave

Figuurlijst

Begrippenlijst

AGILE	Iteratieve project methode bestaand uit sprints.
DevOps	Developer Operations, wordt in dit document gebruikt voor de Azure omgeving waarin de Userstories en taken staan.
Digitaal Certificaat	Certificaat dat kan worden opgeslagen op een computer, usb of server. Bevat een key die gebruikt wordt in de public-private key infrastructuur voor het verzegelen van documenten bij het ondertekenen.
Fork	Een lokale versie van een online (ontwikkel) project waarin veranderingen kunnen worden gemaakt.
Hashing/Hashen	Versleutelen van een tekst of lijst van bytes met een hashing algoritme.
PDF Dictionary	Een type PDF object waarvan de waarde bestaat uit meerdere objecten.
PDF Object	Een PDF object bestaand uit een naam en een waarde waar PDF documenten uit zijn opgemaakt.
POC	Kort voor proof-of-concept, een software realisatie om aan te kunnen tonen dat een functioneel concept werkt en een oplossing biedt zonder dat compleet moet zijn voor een eindgebruiker.
SAML	Security Assertion Markup Language 2.0, XML protocol voor autorisatie en identificatie doeleinden.
SCRUM	Methode voor indelen rollen teamleden binnen een AGILE project.
TrustChain	Een ketting aan certificaten die onder elkaar hangen, van gebruiker naar het hoogst trouwbare instituut.
TSP	Trusted Service Provider, door de overheid erkende providers die digitale certificaten mogen uitgeven.

1 Inleiding

Op het moment van schrijven, werd er nauwelijks nog aandacht besteed aan de Covid pandemie die de complete wereld in de ban had, en in sommige delen van de wereld nog steeds houdt. Deze gezondheidscrisis heeft een nieuw licht op thuiswerken en werkzaamheden op afstand uitvoeren geworpen. Werken op afstand kreeg een enorme boost in zowel het onderwijs als het bedrijfsleven. Ondanks dat de wetten en regels omtrent digitaal ondertekenen er al een tijd waren voor de pandemie, hebben een groot aantal ondernemers pas de toegevoegde waarde toen het werk op afstand gedaan moest worden ingezien. Hoewel de pandemie nu achter ons ligt, zijn sommige werkwijzen blijven hangen omdat het efficiënt is. Zoals digitaal ondertekenen, wat reistijden bespaart en ook net zo veilig is als tekenen op locatie.

In dit verslag is opgenomen hoe het afstudeerproces bij B-ware Business Software eruitzag voor het implementeren van een digitaal ondertekenen functie. Het afstudeerbedrijf wilde graag inzicht krijgen in het proces, de benodigdheden en technologie door een proof-of-concept te realiseren waarin wordt aangetoond hoe het mogelijk is om documenten te ondertekenen.

B-ware Business Software is een softwarebedrijf gevestigd in Enschede dat zich specialiseert in document generatie en huisstijlautomatisering. Dit doet het bedrijf voornamelijk met het product genaamd DocSys. B-ware gebruikt momenteel een andere partij voor het ondertekenen van documenten, maar zij willen dit liever compleet zelf in beheer nemen. Zo kunnen zij een nog completer pakket aanbieden aan jun klanten vanuit één leverancier.

Het verslag bevat informatie over de technische onderdelen van dit proces, maar zal ook onderdelen bevatten waarbij er wordt gekeken naar de juridische onderdelen. Het uitgangspunt is om documenten te ondertekenen die rechtsgeldig zijn in zowel Nederland en andere EU lidstaten op een veilige manier, dat aansluit op de wens van de klanten van het afstudeerbedrijf.

Voor de onderzoeksfase is er voornamelijk gekeken naar literatuur over digitaal ondertekenen, wet- en regelgeving in zowel Nederland als de Europese Unie, online certificaten voor veilig handelen en digitale identificatie methodieken.

In de ontwerp- en realisatiefase is er rekening gehouden met de kennis over software ontwikkeling dat al aanwezig is bij het bedrijf. Dat is omgezet naar randvoorwaarden voor de keuze in frameworks en programmeertalen van de verschillende systeem onderdelen. Dit is gedaan om ervoor te zorgen dat het proof-of-concept gebruikt kan worden voor doorontwikkeling.

2 Bedrijfsinformatie

2.1 Werkzaamheden

B-ware Business Software is specialist in documentcreatie, zij hebben 20 jaar ervaring met huisstijlautomatisering, het vereenvoudigen van documentcreatie en het optimaliseren van document-workflow. Zij zijn eigenaar, ontwikkelaar en leverancier van DocSys, software waarmee documenten kunnen worden gegenereerd in huisstijl. (over b-ware, sd)

2.2 Missie

B-ware Business Software's missie is om documentcreatie te vereenvoudigen, huisstijl te waarborgen en fouten te voorkomen. Zij zorgen dat hun klanten snel, foutloos en gemakkelijk documenten kunnen maken, zoals brieven, offertes, contracten en presentaties, allemaal in huisstijl en voorzien van actuele gegevens en informatie door integratie van meerdere systemen. (missie-en-visie, sd)

2.3 Bedrijfsprofiel

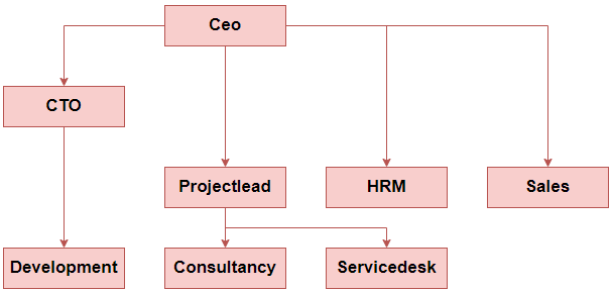
In 1990 is B-ware opgericht met als doel: uitdagende automatiseringsvraagstukken op technisch en administratief gebied oplossen. Na een aantal jaar is B-ware zich gaan specialiseren in het makkelijk maken van kantoorwerk door Microsoft Office-gerelateerde documentprocessen te optimaliseren en zo klanten te helpen snel, makkelijk en foutloos de juiste informatie, in de juiste huisstijl en juiste vorm (o.a. DOCX, PDF, HTML, E-mail) bij de juiste personen te krijgen. Dit heeft geleid tot de ontwikkeling van de huisstijlapplicatie en documentgenerator DocSys. (bedrijfsprofiel, sd)

B-ware Business Software levert software aan de Belastingdienst, Ministerie van Justitie en Veiligheid, Raad van Kinderbescherming, Nederlands Forensisch Instituut, KPMG, Alfa accountants, Concordia de Keizer, Alliander, Heineken en Grolsch. Op deze manier is DocSys marktleider op dit gebied in Nederland. (bedrijfsprofiel, sd)

2.3.1 Medewerkers en bedrijfscultuur

De organisatiestructuur binnen B-ware Business Software is plat en informeel, met korte efficiënte lijnen om de service te garanderen die hun klanten verwachten. Bij B-ware Business Software werken momenteel circa vijfentwintig medewerkers waarvan het merendeel ontwikkelaar is. De meeste medewerkers hebben een HBO- of academische achtergrond. Alle ontwikkelaars in het bedrijf hebben de status van Microsoft Certified Professional of zijn hiervoor in opleiding. (organisatie, sd)

2.4 Organisatie Schema



Figuur 1. Organisatiestructuur afstudeerbedrijf.

3 Opdracht

In dit hoofdstuk wordt beschreven welke opdracht de afstudeerder heeft gekregen en wat de waarde is voor het bedrijf. Er wordt beschreven wat het doel is, wat het gewenste resultaat is en welke activiteiten ondernomen moeten worden om dit te bereiken.

3.1 Opdrachtomschrijving

B-ware Business Software heeft de afstudeerder gevraagd een onderzoek op te zetten naar digitaal ondertekenen om alle specificaties op papier te krijgen waar een ondertekenfunctie aan zou moeten voldoen.

Het doel van het onderzoek is om vanuit die benodigdheden, requirements op te zetten voor de realisatie van een proof-of-concept waarin digitaal ondertekend kan worden.

Het is belangrijk dat zij inzicht krijgen in het ondertekenproces, hoe een ondertekenaar wordt geïdentificeerd en zijn identiteit gekoppeld wordt aan de handtekening, hoe het bestand gecontroleerd kan worden op wijzigingen na het ondertekenen en welke verschillende veiligheidsniveaus er zijn in het ondertekenen en hoe je daaraan kunt voldoen.

Bij het afstudeerbedrijf wordt het .NET framework gebruikt en daar specialiseren zij zich ook in, daarom is gevraagd om de backend in .NET te maken.

3.2 Startsituatie

Bij het bedrijf wordt er al een ondertekenfunctie afgenomen van een softwareleverancier. Er is veel basis kennis over digitaal ondertekenen, maar geen technische kennis over de implementatie van digitaal ondertekenen. Daarnaast is er bij het bedrijf veel kennis over het genereren van documenten.

3.3 Probleemstelling

B-ware Business Software specialiseert zich in document generatie en huisstijl automatisering en zij vinden dat digitaal ondertekenen daar een onderdeel van zou moeten zijn. Het is belangrijk dat het product dat zij aanbieden compleet is en fungeert zonder afhankelijk te zijn van code van andere leveranciers.

Het bedrijf gebruikt op dit moment een ondertekenoptie geleverd door een externe leverancier. Het bedrijf zou dit graag zelf willen ontwikkelen en in beheer willen nemen. Dit stelt het bedrijf in staat om een nog completer pakket aan te bieden zonder dat er gewerkt hoeft te worden met externe partijen.

De optie van de huidige externe leverancier laat sporen achter in het document zoals referenties naar de leverancier. Door zelf een oplossing te ontwikkelen, behoudt het bedrijf controle over de informatie in het document komt te staan. Het bedrijf kan op die manier een completer pakket aanbieden.

3.4 Doelstelling

Het stagebedrijf heeft de student gevraagd om een onderzoek op te zetten en aan de hand daarvan een proof-of-concept te ontwikkelen om aan te tonen dat een document ondertekend kan worden met een certificaat gebonden aan een persoon door middel van digitale identificatie methodieken. Het is belangrijk dat een verstuurder de status van zijn document kan bijhouden en inzien wie er al ondertekend hebben en welke ondertekenaars nog moeten ondertekenen.

Het gros van de functionaliteiten moeten in de backend verwerkt zijn zodat er in de nabije toekomst een koppeling gemaakt kan worden tussen de onderteken functie en bestaande systemen. Dit wordt niet direct gedaan, omdat het om een groot systeem gaat bestaand uit meerdere systemen. Om dit draaiend te krijgen in de ontwikkelomgeving zou veel compilatie tijd kosten en het ontwikkelproces hinderen. Door het in de backend te implementeren kan het worden gekoppeld aan de bestaande systemen.

De hoofdvraag van het onderzoek luidt:

- Hoe kan er een digitaal onderteken functie geïmplementeerd worden die voldoet aan de Nationale en Europese wetgeving bij B-ware Business Software?

Om daar een antwoord op te vinden wordt er een onderzoek opgezet waarin de volgende onderzoeksvragen moeten worden beantwoord:

- Wat houdt digitaal ondertekenen in?
- Hoe wordt digitaal ondertekenen veilig uitgevoerd?
    - Hoe kan er achterhaald worden wie er heeft ondertekend?
    - Hoe kan er achterhaald worden of het document niet gewijzigd is?
- Hoe kan de ondertekenaar worden geïdentificeerd?
- Hoe ondersteunt het PDF-bestandsformaat digitaal onderteken?

De resultaten van dit onderzoek worden gebruikt om een ontwerp te maken dat wordt gerealiseerd in het proof-of-concept om aan te tonen dat de digitale documenten kunnen worden ondertekend.

3.5 Resultaat

Het project is klaar wanneer er een proof-of-concept is gerealiseerd bestaand uit een front-end, backend en database waarin een document kan worden ondertekend.

Om een document te kunnen ondertekenen moet het proof-of-concept aan de volgende punten voldoen:

- Een verstuurder kan inloggen en zich identificeren.
- Een ingelogde gebruiker kan een document uploaden.
- Aan het document kan één of meerdere ontvangers worden toegevoegd die moeten ondertekenen.
- Ontvangers moeten op een veilige manier een code kunnen ontvangen om bij het document te kunnen om dit te kunnen ondertekenen zoals een privé mail.
- De verstuurder kan de status van het document inzien.

De handtekening en het document moet aan de volgende punten voldoen:

- Het document is verzegeld met een digitaal certificaat.
- De verstuurder is gebonden aan de handtekening ter identificatie.
- Het document waarin de handtekening wordt gezet is niet aanpasbaar na het ondertekenen.

In de applicatie moet bijgehouden kunnen worden welke ondertekenaars al hebben ondertekend en welke niet. Indien nodig moet het verzoek om te ondertekenen opnieuw verstuurd worden.

3.6 Activiteiten

Om het gewenste resultaat te bereiken is het project opgezet in twee fases.
De onderzoek- en adviesfase waarin informatie wordt verzameld over het onderwerp en de belangrijkste onderdelen worden omgezet in een advies en een functioneel ontwerp voor de applicatie. In deze fase is het belangrijk om een idee te krijgen wat er gerealiseerd moet worden. Door het advies en ontwerp in deze fase te maken kan er met het bedrijf worden kortgesloten of dit naar wens voor het daadwerkelijk proof-of-concept wordt gerealiseerd.

In de tweede fase wordt het proof of concept gerealiseerd. Voordat een softwareonderdeel kan worden gerealiseerd wordt er eerst een technisch ontwerp gemaakt per onderdeel. Zo kan dit worden meegenomen in het iteratieve proces en aangescherpt worden per softwareonderdeel.

4 Methodieken

Voor het uitvoeren van het onderzoek en de ontwikkeling van het POC zijn er verschillende methodieken gebruikt. In dit hoofdstuk wordt opgenomen welke methodes er zijn gebruikt en waarom.

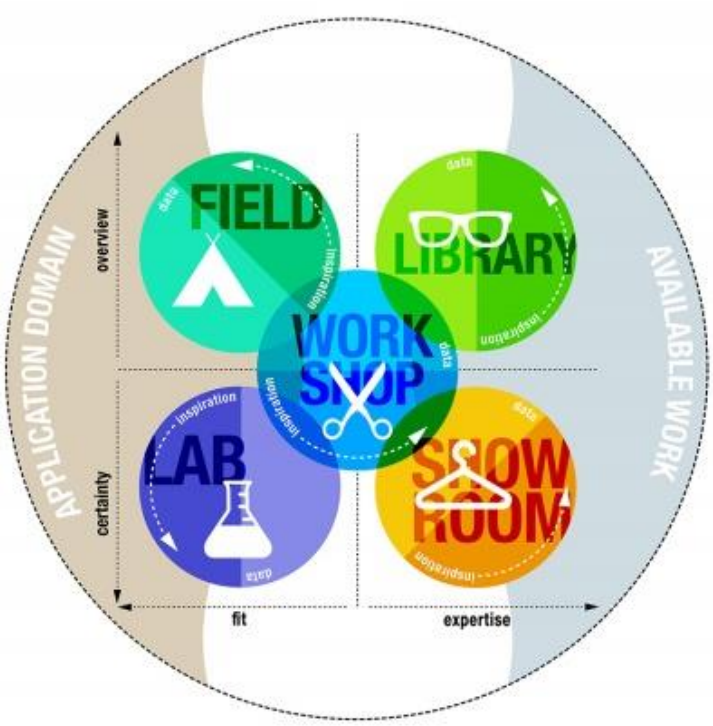
4.1 Onderzoeksmethode

Er is gekozen om gebruik te maken van het DOT-model voor het indelen van de onderzoeksprocessen om zo een duidelijk beeld te scheppen wat bij elk onderdeel verwacht wordt en wat er nodig is voor voltooien van deze processen. Het model is gemaakt om inzicht te creëren voor zowel de opdrachtgever als de opdrachtnemer, door manieren te bieden om onderzoek makkelijker te kunnen communiceren. (The_DOT_Framework, sd)

Het model is onderverdeeld in 3 stukken:

- What (Drie domeinen die richten op wat wordt er onderzocht.)
- Why (Waarom wordt het onderzoek gedaan, de focus ligt op het belang van stakeholders.)
- How (Hoe wordt er onderzoek gedaan.)

Elk onderdeel van het DOT-model is weergegeven in de volgende afbeelding.



Figuur 2 Overzichtsweergave DOT-model. (The_DOT_Framework, sd)

Met dit model zijn bij de onderzoeksvragen passende onderzoeksmethoden gevonden. Het onderzoek vond voornamelijk plaats binnen de “Library” strategie, dit houdt in dat er voornamelijk onderzoek wordt gedaan door informatie te verzamelen dat al beschikbaar is zoals websites.

Voor het realiseren van het prototype werd voornamelijk de “workshop” strategie gebruikt om erachter te komen welke ontwerpen werken en hoe dit zich uit in de realisatie.

4.2 Projectmethode

Zoals eerder vermeld is het complete project opgedeeld in twee fases, een analysefase en een realisatiefase. De activiteiten in beide fases zijn fundamenteel anders en hebben daarom ook andere methodes nodig om succesvol voltooid te kunnen worden.

De onderzoeksfase is opgedeeld in deelvragen en per deelvraag is er een vaste deadline gesteld. De looptijd van elk onderzoek is een of twee weken afhankelijk van een inschatting van de grootte. Voor sommige onderzoeken is dit vrij ruim, wanneer een onderzoek dan eerder klaar is, blijft er meer tijd over voor het ontwerp en de realisatie van het proof-of-concept.

In de realisatiefase is de AGILE met SCRUM projectmethode gebruikt. De sprints zijn opgedeeld per twee weken en het proces wordt bijgehouden in de Azure DevOps omgeving van het afstudeerbedrijf. Een senior ontwikkelaar fungeert als SCRUM-master en de bedrijfsbegeleider fungeert als product-owner.

Een sprint begint met een sprintplanning waarin de student en de SCRUM-master een tijdinschatting maken per taak. De student en SCRUM-master geven beide aan hoeveel tijd ze verwachten dat een taak zal duren. Bij verschillende inschattingen worden de inschattingen verdedigt en gediscussieerd tot er een concessie is over de inschatting.

Na het inschatten van de tijden worden er taken ingepland die haalbaar zijn voor die sprint. Inschattingen staan niet altijd gelijk aan hoeveel tijd er daadwerkelijk nodig is voor een taak, daarom werd er bij uitloop de taken overgezet naar de volgende sprint. In de volgende sprint werden de taken dan opnieuw bekeken en ingeschat met de nieuwe informatie.

Halverwege een sprint werd er een meeting gehouden om de progressie tot dan te volgen en om alvast bij te schakelen indien dat nodig is.

Aan het eind van een sprint werd er een demo gegeven aan de SCRUM-master en de product-owner om te kijken dat het de goede richting in ging en eventueel de richting aan te passen. In dat geval werden er nieuwe taken gemaakt voor deze aanpassingen.

Aan het eind van de demo werd er gereflecteerd op die sprint om te kijken wat er verbeterd kon worden.

5 Onderzoek

In dit hoofdstuk wordt ingegaan op de analysefase van het project. Hierin is het onderzoek beschreven wat digitaal ondertekenen inhoudt, hoe de authenticiteit van de handtekening wordt gegarandeerd en op welke manier dit digitaal wordt ondersteund in een PDF-document.

5.1 Wat houdt digitaal ondertekenen in?

Digitaal ondertekenen is de digitale equivalent van het traditioneel ondertekenen zoals dat met pen en papier wordt gedaan. Een handtekening worden geplaatst op een document om aan te tonen dat een persoon akkoord gaat met de documentinhoud voor juridische doeleinden, bijvoorbeeld bij contracten. Ook brieven worden vaak voorzien van een handtekening om aan te geven dat de ondertekenaar in overeenstemming is met de inhoud van de brief.

Omdat een handtekening wordt gebruikt voor juridische doeleinden is er in de wet vastgelegd wat de juridische kracht is van een handtekening, dit geldt ook voor digitaal ondertekenen. De wetten omtrent digitaal ondertekenen zijn vastgelegd in de oude Telecommunicatiewet dat is aangepast op de verordening van de Europese Unie 910/20214 ook bekend als de eIDAS. (Telecommunicatiewet, 2021)

Daarin staat vastgelegd dat er drie vormen zijn van digitale handtekeningen.

- De normale handtekening
- De geavanceerde handtekening
- De gekwalificeerde handtekening

De normale handtekening is de simpelste vorm van digitaal ondertekenen. Bijvoorbeeld een afbeelding van een digitaal getekende handtekening, een foto van een handtekening of een gescande handtekening. Bij deze vorm is het lastig om vast te stellen wie de handtekening heeft gezet. Deze handtekening kan rechtsgeldig zijn, maar dan moet er aangetoond kunnen worden dat de ondertekenaar daadwerkelijk de handtekening heeft gezet. Dit is lastig te bewijzen en daardoor wordt deze handtekening ook als een onveilige methode van digitaal onderteken gezien. (Hernandez, 2021)

Bij het zetten van een geavanceerde handtekening wordt de ondertekenaar geïdentificeerd en zijn identiteit gekoppeld aan het getekende document. Daarnaast moet het getekende document de mogelijkheid bieden om te kunnen controleren dat het document niet is gewijzigd. Wanneer er wijzigingen hebben plaatsgevonden kan de handtekening ongeldig worden verklaard. (Hernandez, 2021)

Als laatst is er de gekwalificeerde handtekening. Dit is in feite een geavanceerde handtekening alleen moet er een gekwalificeerd digitaal certificaat voor ondertekenen gebruikt zijn bij het ondertekenen. Deze certificaten zijn alleen verkrijgbaar bij partijen die als vertrouwd zijn verklaard door de overheid. (Hernandez, 2021)

Documenten met bepaalde doeleinden hebben meer veiligheid nodig om de rechtsgeldigheid te garanderen. Bijvoorbeeld een contract voor een telefoonabonnement kan op minder veilige wijze getekend worden dan een hypotheekakte. In traditioneel ondertekenen kan het eerste in de winkel worden gedaan tussen verkoper en consument. Een hypotheekakte kan alleen worden ondertekend onder toezicht van een notaris. Bij digitaal ondertekenen neemt het certificaat de rol van de notaris over. Wanneer een document op een veilige manier naar een gebruiker zonder certificaat wordt verstuurd, kan deze ondertekenen met het certificaat van de verstuurder. (Hernandez, 2021)

5.2 Hoe wordt digitaal ondertekenen veilig uitgevoerd?

Digitaal ondertekenen kan veilig worden uitgevoerd door het aan de volgende twee voorwaarden te laten voldoen:

- Er moet achterhaald kunnen worden wie de handtekening heeft gezet.
- Er moet achterhaald kunnen worden dat het document niet is gewijzigd na het ondertekenen.

Op deze manier kan worden aangetoond dat het document authentiek is, dat alle partijen het eens zijn met de inhoud en dat de partijen ook daadwerkelijk het document hebben ondertekend. Dit is belangrijk, omdat het om juridische documenten gaat en als er niet wordt voldaan aan deze voorwaarden het lastig of niet aan te tonen is dat het om legitieme handtekeningen gaat.

5.2.1 Hoe kan er achterhaald worden wie er heeft ondertekend?

De identiteit bij het ondertekenen kan achterhaald worden, door een identiteit van een natuurlijk persoon aan een handtekeningen te verbinden. Dit wordt gedaan door middel van een digitaal certificaat en een erkend inlogmiddel. Het certificaat zorgt ervoor dat de authenticiteit van de handtekening wordt gegarandeerd. Het inlogmiddel en certificaat samen binden het document aan een natuurlijk persoon.

Zoals eerder vermeld zijn er maar een aantal bedrijven die namens de staat deze certificaten mogen uitgeven. Bij het verkrijgen van zo'n certificaat wordt de afnemer ook geïdentificeerd. Deze certificaatuitgevers nemen hun certificaten ook weer af van de staat. Op die manier ontstaat er ketting van vertrouwen vanaf de staat, de uitgever en de afnemer. Deze certificaten staan in een digitaal register en op deze manier kan er gecontroleerd worden of het certificaat legitiem is.

Bij het ondertekenen moet de eerste ondertekenaar zichzelf identificeren. Dit kan alleen worden gedaan met vertrouwde middelen die wederom weer worden erkend door de overheid. Dit hoeven niet dezelfde bedrijven te zijn als die certificaten aanbieden, maar er is wel overlap. De aanbieders van deze online identificatiemiddelen bieden EHerkenning aan. Dit is de naam dat aan online identificatie is gegeven door de overheid. De andere ondertekenaars kunnen zich identificeren door ervoor te zorgen dat zij het document ontvangen op een plek dat alleen voor hen bedoeld is zoals een privémail

EHerkenning is vergelijkbaar met DigiD, echter is DigiD alleen bedoeld voor de publieke sector en Eherkenning speciaal voor de privésector. Beide diensten gebruiken het SAML2 protocol om de datawisseling bij het identificeren in te richten. SAML2 staat voor Security Assertion Markup Language 2.0, het is een login protocol taal gebaseerd op XML gemaakt voor authenticatie en autorisatie. (Koppelvlakspecificatie DigiD SAML Authenticatie, n.d.)

Het proces om aan te sluiten bij een van deze diensten duurt te lang en is vrij prijzig. Het is niet realistisch om dit voor het proof-of-concept te doen en daarom is ervoor gekozen om dit onderdeel van het project na te maken met Azure login. Azure biedt de mogelijkheid om het authenticatie en login proces in te richten met SAML2. Daardoor kan de dataoverdracht en authenticatie stappen van EHerkenning en DigiD worden nagemaakt.

Om inzicht te krijgen in welke protocollen er worden gebruikt voor SAML2 door DigiD. Er is gekeken naar de documentatie van DigiD. De documentatie van het SAML2 protocol is te vinden op afsprakenstelsel.etoegang.nl, hierop staat een verzameling van juridische, organisatorische, technische en functionele afspraken met betrekking tot online identificatie. (etoegang.nl, n.d.)

In de specificaties van het online identificatie van natuurlijke personen staat bijvoorbeeld hoe de attributen moeten heten, het formaat zoals karakterlengte en of dat attribuut verplicht of optioneel is. In hoofdstuk 5.3 wordt dit nog meer uitgelicht. (Attribuutcatalogus natuurlijke personen, n.d.)

5.2.2 Hoe kan er achterhaald worden of het document niet gewijzigd is?

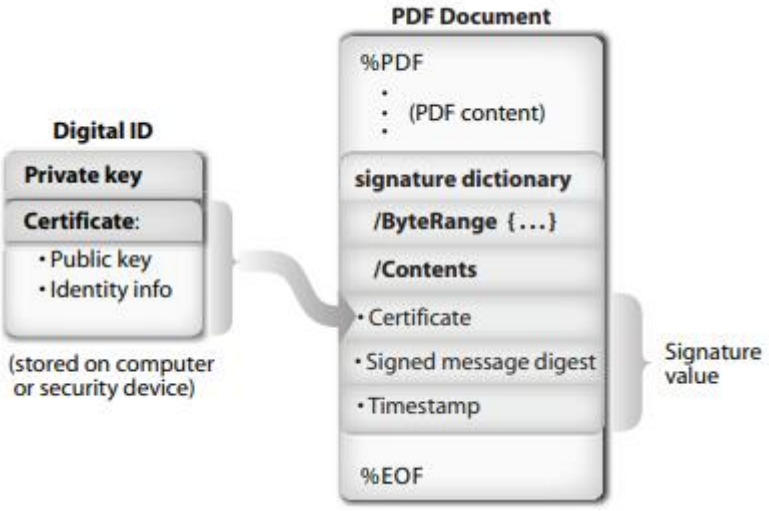
In dit project wordt er alleen gefocust op PDF-documenten en op welke manier daarin wijzigingen kunnen worden bijgehouden. Elk PDF bestaat op de achtergrond uit een soort taal dat bijna gelijk is aan PostScript, een codetaal voor het maken van pagina's ook ontwikkeld door Adobe. (PostScript, 2022)

Hierin zijn objecten gedefinieerd voor alle onderdelen van de pagina, een paar voorbeelden daarvan zijn:

- Basic objects, denk aan Booleans, Integers, Numbers en Key/Value pairs genaamd dictionaries.
- Indirect objects, objecten om naar te verwijzen, denk bijvoorbeeld aan lettertypes.
- Streams, denk aan bytestreams met informatie.

(Endignoux, 2016)

In de volgende afbeelding wordt een voorbeeld structuur getoond van een PDF document met handtekening.

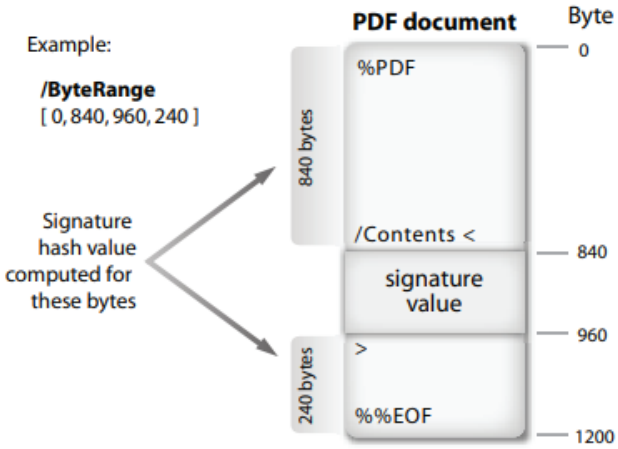


Figuur 3 Overzicht datastructuur PDF bestand met handtekening. (Digital Signatures in a PDF, 2021)

Een handtekeningen in PDF bestaat uit drie onderdelen:

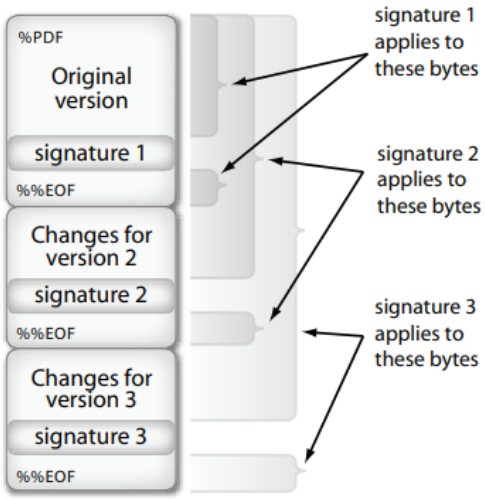
- Het certificaat voor het verzegelen van het document.
- De bytedata van de handtekening.
- Een tijdsstempel.

De bytes die onderdeel uitmaken van de handtekening wordt buiten de content van het document opgeslagen. De content van het document is de inhoud van het document. Dit wordt gedaan zodat de content kan worden versleuteld bij plaatsen van de handtekening. Ook de bytes die zich aan het einde van het document bevinden die na de ruimte die wordt vrijgehouden voor de handtekening wordt in de versleuteling meegenomen. Tijdens de versleuteling wordt er een waarde berekend vanuit de bytes van de content, dit zou een unieke code moeten geven die niet langer gelijk is wanneer de content wordt aangepast. Zie afbeelding 3 hieronder voor een visuele weergave. (Digital Signatures in a PDF, 2021)



Figuur 4 De byterange en de handtekening. (Digital Signatures in a PDF, 2021)

Als er meerdere handtekeningen worden geplaatst op een document dan ziet dit proces er ongeveer hetzelfde uit. Bij elke nieuwe handtekening wordt de vorige versie van het hele document meegenomen in het maken van de berekening voor de versleuteling. Elke keer wordt alleen de nieuwste handtekening niet meegenomen in de berekening. Zoals te zien in de volgende afbeelding. (Digital Signatures in a PDF, 2021)



Figuur 5 Meerdere handtekeningen toevoegen. (Digital Signatures in a PDF, 2021)

Als er wijzigingen worden gemaakt aan het document na de eerste handtekening klopt de hashwaarde niet meer en kan het certificaat niet ontsleuteld worden. Dit betekent dat de handtekening niet langer geldig is. Zolang het certificaat geldig blijft kan er geconcludeerd worden dat er geen wijzigingen zijn gemaakt aan de inhoud van het document.

In hoofdstuk 5.4 wordt dieper ingegaan in hoe dit in een .NET applicatie kan worden gedaan en op welke manier dit gecontroleerd kan worden door de gebruiker.

5.3 Hoe kan de ondertekenaar worden geïdentificeerd bij het ondertekenen?

Zoals genoemd in het vorige hoofdstuk wordt een persoons identiteit gekoppeld bij het plaatsen van een certificaat in de handtekening. Het certificaat is gebonden aan een identiteit bij de verkrijgen van het certificaat.

Deze certificaten worden aangeboden door vertrouwde leveranciers, deze leveranciers zijn:

- Cleverbase
- Digidentity
- KPN
- QuoVadis

Een digitaal certificaat is een bestand op een computer of hardware apparatuur zoals bijvoorbeeld een usb-stick. In dit bestand zit een code vergelijkbaar met een bijvoorbeeld een wachtwoord. Doormiddel van een private-key infrastructuur wordt deze sleutel verbonden met een erkend bedrijf dat kan controleren of de sleutel legitiem is. (Digital Certificates, sd)

Het controleren van de certificaat wordt gedaan met pdf-viewers, zoals de pdf-viewer van Adobe zelf, Acrobat Reader. Hierin is te zien of het document is aangepast en welke certificaten aan de handtekeningen hangen.

Bij het gebruik van het certificaat moet de gebruiker zich identificeren, zoals eerder vermeld is daarvoor EHerkenning nodig of DigiD. Eherkenning heeft een aantal standaarden voor online identificatie bedacht. Deze vier standaarden zijn alle vier bedacht met een incrementele garantie voor veiligheid, oftewel het veiligheidsniveau.

De 4 veiligheid niveaus zijn:

- EH2
- EH2+
- EH3
- EH4

In de volgende 4 tabellen staan de specificaties van elk veiligheidsniveau en waarin elk niveau zich verschilt met de vorige.

EH2 (eIDAS: basis)	
Inlogmethode	Gebruikersnaam en (sterk) wachtwoord.
Aanvraagprocedure	Dit kan online bij een leverancier die erkend wordt door Logius.
Controle identiteit	Wettelijk identiteitsdocument kopie.
Controle van bevoegdheid	Via KVK-registratie.
Uitgifte	Mag online, op basis van een betrouwbaar brondocument.

EH2+ (eIDAS: substantieel)	
Inlogmethode	Door 2-factorauthenticatie via gebruikersnaam en wachtwoord samen met sms-code of tokencode.
Aanvraagprocedure	Volledig online of deels online en deels per post.
Controle identiteit	Kopie van een wettelijk identiteitsdocument.
Controle van bevoegdheid	Via KVK-registratie.
Uitgifte	Activeringscode en wachtwoord. Beide worden apart verstuurd.

Het verschil tussen EH2+ en EH2 is dat EH2+ 2-factorauthenticatie gebruikt bij de inlogmethode.

EH3 (eIDAS: substantieel)	
Inlogmethode	Door 2-factorauthenticatie via gebruikersnaam en wachtwoord, samen met sms-code of tokencode of een app met QR-code.
Aanvraagprocedure	Wette identiteitsdocument en face-to-face controle, dit kan ook online.
Controle identiteit	Identiteitscontrole geldt voor wettelijke vertegenwoordiger, de machtigingenbeheerder en gebruiker.
Controle van bevoegdheid	Via KVK-registratie.
Uitgifte	Online of bijvoorbeeld aangetekende post.

Het verschil tussen EH3 en EH2+ is dat er extra veiligheidsmaatregelen in de controle van de identiteit en bevoegdheden zijn ingebouwd. Daarnaast is de uitgifte online of via aangetekende post.

EH4 (eIDAS: hoog)	
Inlogmethode	Door PKI-certificate of 2-factorauthenticatie.
Aanvraagprocedure	Wettelijk identiteitsdocument, dit mag geen kopie zijn en face-to-face controle.
Controle identiteit	Identiteitscontrole geldt voor de wettelijk vertegenwoordiger, machtigingenbeheerder en gebruiker.
Controle van bevoegdheid	Via KVK registratie
Uitgifte	Aangetekende post of ophalen op locatie.

Het verschil tussen EH4 en EH3 is dat er extra ingebouwde veiligheidsmaatregelen zijn in de inlogmethode door middel van het gebruik van een PKI-certificaat en door de extra controle van de identiteit en bevoegdheden door middel van fysieke identificatie.

In meeste documenten wordt EH2+ of EH3 als voldoende beschouwd. Uit een gesprek met de opdrachtgever is gebleken dat de documenten waar ze voornamelijk mee te maken hebben, de EH2+ voldoende is om een persoon mee te identificeren bij het ondertekenen. (Betrouwbaarheidsniveaus, sd)

5.3.1 Conclusie

In overleg met het bedrijf is ervoor gekozen om EH2+ te realiseren. Dit wordt gedaan door het instellen van de EH2 stappen in Azure AD. Dat wordt gedaan door met SAML2 dezelfde protocollen en data te gebruiken als in de EH2+ specificaties zijn beschreven. Er is voor EH2+ gekozen omdat dat het meest voorkomend en gevraagd is bij de klanten van het afstudeerbedrijf.

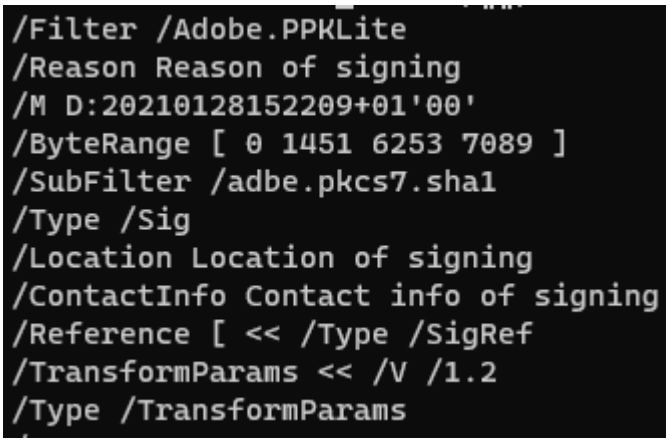
De ondertekenaars die geen certificaat gebruiken worden geïdentificeerd doordat zij op een veilige manier het document ontvangen voor het ondertekenen.

5.4 Hoe ondersteunt het pdf-bestandsformaat digitaal ondertekenen?

In dit hoofdstuk wordt dieper ingegaan in hoe het pdf-documenten er onderwater uitzien en op welke wijze die aangepast kunnen worden.

Zoals eerder vermeld bestaan PDF-documenten uit verschillende objecten en elementen die samen de inhoud en weergave van een document bepalen. Door het gebruik van een Python console programma is het mogelijk om naar de verschillende objecten te kijken in een pdf-bestand, dit programma heet PeePDF. Met PeePDF is er gekeken naar een PDF-document met een het minimaal aantal objecten dat een PDF-document met handtekening nodig heeft. Op die manier kan er gezien worden uit welke objecten een handtekening bestaat.

In de volgende afbeelding wordt het de output van PeePDF getoond, achter elke "/" staat een object en achter het object staat de waarde van dat object. De objecten die getoond worden zijn enkel de objecten die zich bevinden in de handtekening.



Figuur 6 Output PeePDF.

In de volgende tabel wordt elk object in figuur 6 beschreven.

Object	*Beschrijving*
Reason	Reden voor het zetten van de handtekening.
Location	Locatie tijdens het zetten van de handtekening
ContactInfo	De contact informatie van de ondertekenaar.
FilterObject	Dit wordt gebruikt door PDF-viewers om te achterhalen wat voor type handtekening er is gezet. De waarde van het filter is een ander object waarin het handtekening type staat.
Subfilter	Dit object wordt gebruikt door de PDF-viewer om te achterhalen welke versleutelingstype er is gebruikt voor de certificaat.
M	Dit object staat voor de datum en tijd van plaatsen handtekening.
ByteRange	In de byterange wordt bijgehouden welke bytes worden meegenomen in de versleuteling en welke bytes niet, zoals die van de handtekening zelf.
Reference	Dit wordt gebruikt om het gehele handtekening object te hangen aan de PDF-dictionary door middel van een referentie.
TransformParams	Dit object wordt gebruikt om de viewer te laten weten op welke methode de wijzigingen in het bestand moeten worden bijgehouden.

(Adobe, 2008)

Het bedrijf heeft gevraagd om PDF-documenten op een laag niveau aan te passen. Dit houdt in dat er geen libraries gebruikt mogen worden die het gros van het werk al voor de ontwikkelaar doet, zoals verschillende objecten automatisch aanmaken bij het toevoegen van een formulier veld. Het bedrijf heeft dit gevraagd om meer inzicht te krijgen in de generatie van PDF-documenten.

Uit onderzoek naar het aanpassen van PDF-documenten met .NET op een laag niveau is gekomen dat er twee methodes zijn. Deze zijn:

- Een PDF parser realiseren.
- Een library gebruiken genaamd PDFClown.

De eerste optie is tijdrovend en wordt in de praktijk gedaan door grote teams over een lange periode van tijd. Dit is niet realistisch voor de opdracht en daarom is er gekozen om gebruik te maken van PDFClown. PDFClown is de enige open-source library voor .NET die een PDF document kan parsen. PDFClown mist de functionaliteit om handtekeningen toe te voegen, daarom is ervoor gekozen om de PDFClown library uit te breiden met de onderteken functie.

6 Ontwerp

Er is een ontwerp gemaakt voor het uitwerken van de proof-of-concept. In dit hoofdstuk wordt beschreven waar over is nagedacht bij het ontwerpen van de applicatie. Het functionele ontwerp is gemaakt voor de realisatiefase. Het technische ontwerp is gemaakt in de eerste sprint en per sprint uitgebreid afhankelijk van welke functionaliteiten er werden uitgewerkt.

6.1 Requirements en randvoorwaarden

Voor de applicatie zijn een aantal functionele requirements opgezet waar de applicatie aan moet voldoen. Deze zijn geordend op prioriteit zodat er bij calamiteiten op of afgeschaald kan worden.

De requirements zijn samen met de opdrachtgever opgezet gebaseerd op de onderzoeksresultaten.

De prioriteiten van de requirements zijn beoordeeld door het gebruik van de MSCW-methode.

- Must have (Dit moet in het systeem zitten)
- Should have (Zou in het systeem moeten zitten)
- Could have (Het kan in het systeem zitten, maar niet essentieel voor het proof-of-concept.)
- Would like to have (Het zou mooi zijn als het erin zit maar alleen als er tijd over is.)

Nummer	MSCW	Requirements
1	M	De gebruiker moet een PDF/a formaat document kunnen versturen naar ondertekenaars.
2	M	Een ontvanger kan een pdf-bestand ondertekenen zonder in te loggen.
3	S	De gebruiker kan controleren of het bestand is gewijzigd na het is opgestuurd.
4	S	De gebruiker kan inzien met welke certificaat er is ondertekend.
5	M	De gebruiker kan bij houden bij welke gebruikers een document is geweest.
6	M	De gebruiker moet zichzelf identificeren met een EF2 methode bij het plaatsen van een handtekening.
7	W	De gebruiker kan een persoonsgebonden certificaat aanvragen bij dat onder het certificaat van het bedrijf hangt in de trustchain.
8	W	De beheerder kan persoonlijke certificaten uitgeven.
9	W	De beheerder kan de persoonlijke certificaten beheren.

De realisatie moet voldoen aan randvoorwaarden die zijn gezet door het afstudeerbedrijf. Deze randvoorwaarden zijn:

- De front-end moet geschreven zijn in React.
- De backend moet geschreven zijn in C#.
- Documentatie en code is volledig Engelstalig.
- De code slaagt automatisch de stijlvalidatie van het bedrijf.

6.2 Functionaliteiten

Alle functionaliteiten die het proof-of-concept dient te bevatten zijn verwerkt in de volgende tabel. In de tabel zijn de functionaliteiten beschreven als userstories zodat die kunnen worden omgezet naar de Azure DevOps omgeving. Er is genoteerd welke requirements er nodig zijn voor de totstandkoming van elke functionaliteit. Op die manier wordt er afgedekt dat alle requirements zijn behaald als elke user story voltooid is.

Letter	Reqs.	User Story	Omschrijving
A.	1 6	Verstuurder kan PDF uploaden in systeem.	De gebruiker kan een pdf-bestand uploaden.
C.	1 6	Verstuurder kan PDF versturen naar ondertekenaar(s).	De gebruiker kan een link aanmaken die verstuurd kan worden naar een ondertekenaar.
D.	1 6	Verstuurder kan een digitaal certificaat aan het document hangen.	De verstuurder dat zich heeft geïdentificeerd met een EH2+ inlogmiddel, kan zijn persoonlijke certificaat in het document plaatsen.
E.	2	Ondertekenaar kan ondertekenen.	De ondertekenaar kan het document benaderen met een unieke code en zijn handtekening zetten.
F.	4 5	Verstuurder kan document verkeer volgen.	De verstuurder kan kijken bij wie het document is en indien nodig opnieuw versturen.
G.	3 4	Verstuurder kan fouten traceren.	De verstuurder kan inzien wat de status van het document en wanneer er wijzigingen zijn aangebracht na het versturen.
H.	4	Verstuurder kan certificaten inzien.	De verstuurder kan de certificaten ketting van het document inzien.
I.	7	Verstuurder kan certificaat aanvragen	De verstuurder kan een persoonsgebonden certificaat aanvragen bij het systeem. Deze hoeft nog niet legitiem te zijn en wordt gebruikt voor het testen van het systeem.
J.	7 8 9	Verstuurder kan certificaat ontvangen.	De verstuurder krijgt een aangevraagd certificaat van het systeem en deze wordt in het systeem opgeslagen.
K.	7 8 9	Beheerder kan certificaten beheren	De beheerder kan zien welke gebruiker welke certificaten heeft en indien nodig intrekken bij misbruik.

6.3 Systeem gebruikersrollen

Voor het systeem zijn rollen gedefinieerd die uit de requirements komen. Elke rol heeft zijn eigen functies binnen het systeem.

Het systeem krijgt 3 gebruikers rollen.

- Een verstuurder.
- Een ontvanger.
- Een beheerder.

Een verstuurder moet zich identificeren in het systeem en heeft zijn eigen certificaat nodig om een document te kunnen versturen. Deze rol kan een document uploaden, ontvangers toevoegen aan het document en de status bijhouden van het document.

Een ontvanger hoeft zich niet te identificeren in het systeem. De ontvanger kan het document alleen benaderen met een uitnodigingscode die de ontvanger krijgt via mail als een document verstuurd is. De ontvanger kan alleen het PDF inzien en aangeven of die instemt met de inhoud door een handtekening te zetten.

Een beheerder moet zich identificeren in het systeem en kan alleen certificaten beheren. De beheerder kan certificaat aanvragen van gebruikers beantwoorden.

6.4 Project Onderdelen

Het gehele project bestaat uit de ontwikkeling van 5 onderdelen:

- De front-end.
- De backend.
- De uitbreiding van de ClownPDF library.
- Het instellen van de identificatie en authenticatie server.
- Het opzetten van de database.

De front-end wordt geschreven in React, dit op verzoek van het afstudeerbedrijf. Doordat het in een taal is geschreven waar het bedrijf al ervaring mee heeft, kunnen ze makkelijker de code beoordelen op kwaliteit en is het makkelijker om erop door te ontwikkelen. Daarnaast is React een moderne en innovatieve taal waar veel documentatie voor te vinden is.

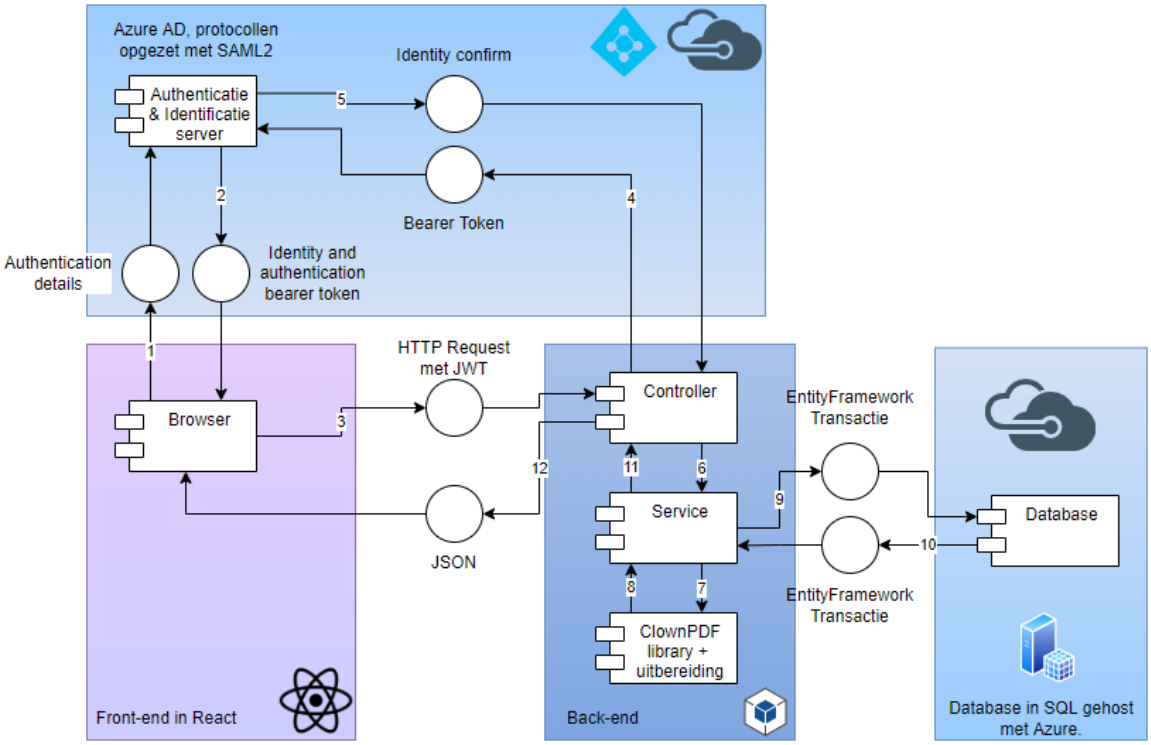
De backend wordt geschreven in .NET, dit omdat het stagebedrijf zich specialiseert in .NET ontwikkeling. Op die manier sluit het goed aan met de rest van de producten van het bedrijf en zorgt het er wederom voor dat door ontwikkeling makkelijker kan worden uitgevoerd. De ClownPDF library is ook geschreven in C# en zal daarmee ook worden uitgebreid.

De identificatie en authenticatie server zal worden ingesteld met het gebruik van het SAML2 protocol in Azure AD. Het stagebedrijf neemt al een licentie af bij Azure voor het gebruik van de services die zij aanbieden. Daarmee kan er een Azure AD ontwikkel omgeving worden ingericht voor dit project. Op die manier kunnen er geen fouten optreden in de bestaande ontwikkelomgevingen in Azure die het bedrijf al in gebruik heeft.

Voor het implementeren van de database is er gekeken naar verschillende SQL en NoSQL oplossingen die worden aangeboden in Azure. Uit de analyse komt voort dat het systeem gevoelige en gestructureerde data gaat verwerken. Er is gekozen voor een SQL database, omdat die als veiliger wordt beschouwd en beter kan werken met gestructureerde data. (Anderson & Nicholson, 2022)

Het .NET framework bevat ook het Entity Framework, dit is een oplossing die ondersteuning biedt voor het koppelen van .NET project en de database. Met die koppeling kan er op een nette en overzichtelijke manier de transacties tussen beide onderdelen worden gerealiseerd. Het biedt ook de mogelijkheid om LINQ te gebruiken om query’s te schrijven om data op te halen. LINQ is een query methode dat op een object georiënteerde manier data kan uitlezen en filteren uit data collecties.

Zie in de volgende afbeelding hoe al deze onderdelen samen komen in het eindproduct. Met nummers is aangegeven in welke volgorde de acties gebeuren in het gehele proces.



Figuur 7. Systeemcomponenten diagram gehele project.

De acties die zijn beschreven in figuur 7, tonen met nummers aan in welke volgorde ze worden uitgevoerd wanneer een gebruiker in het systeem inlogt en een document wil uploaden en versturen.

1. De gebruiker logt in met gebruik van zijn Azure email, wachtwoord en unieke code of telefoon verificatie.
2.Azure AD stuurt de identificatie gegevens van de gebruiker terug met een unieke token voor die gebruiker.
3. De gebruiker wil een document naar een ondertekenaar sturen en upload een document, een verzoek wordt gedaan aan de backend met de token van de gebruiker.
4. De token wordt verstuurd naar Azure AD die controleert of de aanvraag legitiem is.
5. De identiteit wordt bevestigd aan de backend.
6. De controller roept de onderteken service aan om het document klaar te maken voor ondertekenen.
7. De onderteken service gebruikt clown PDF om ruimte in het PDF-bestand te maken voor een handtekening en zet het certificaat van de verstuurder in het document.
8. ClownPDF heeft het document klaargezet en alle objecten toegevoegd voor een valide PDF-document voor ondertekenen.
9. De service maakt een transactie met de het nieuwe document naar de database.
10. De database laat weten dat de transactie is gelukt.
11. De service laat weten dat het document succesvol is geüpload. De controller roept de verstuur service aan om ervoor te zorgen dat de eerste ondertekenaar een unieke code krijgt om het document te kunnen ondertekenen.
12. De gebruiker wordt ingelicht over de status van het document.

In de praktijk zitten er nog wat stappen tussen database, service en controller maar in het voorbeeld is alleen de volgorde aangetoond waarin de verschillende componenten in contact komen.

6.5 Front-end

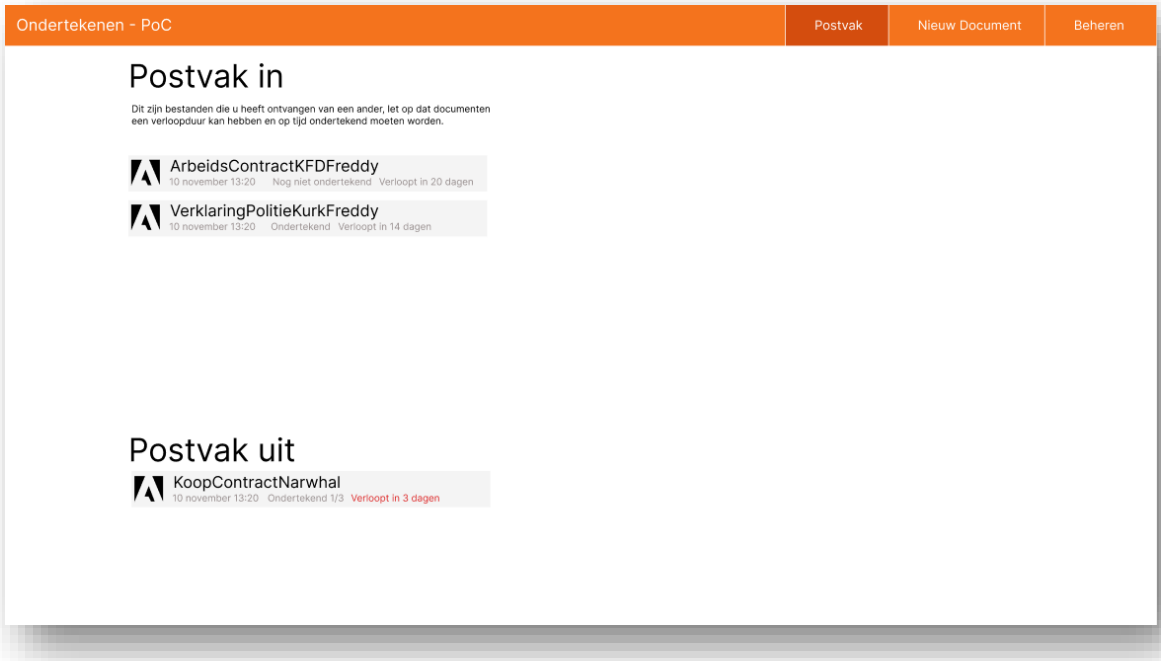
Voor de front-end zijn interactieve ontwerpen gemaakt. Dit houdt in dat het om pagina ontwerpen gaat waar onderdelen zoals het menu en knoppen gebruikt kunnen worden om naar een andere pagina te gaan. Door het gebruik van interactieve ontwerpen, is het makkelijker om de proces-flow en de gebruikers ervaring te testen.

Als testpersonen zijn mensen binnen het bedrijf gebruikt die buiten het project staan. Deze personen kregen de taak om bijvoorbeeld een document te versturen naar een persoon en zij moesten dan zelf proberen van het inlogscherm naar het “versturen voltooid” scherm te komen.

Een medewerker met expertise op het gebied van front-end heeft ook de ontwerpen getest en daarop feedback gegeven. Deze feedback is verwerkt en opnieuw getest, op die manier is iteratief het ontwerp verbeterd tot de huidige versie.

Bij het ontwerpen is er rekening gehouden dat de front-end dient voor het testen en geven van demo’s van de functionaliteiten van de applicatie. Alle requirements zijn hierin verwerkt, maar er is bijvoorbeeld geen footer aanwezig. Wel is er rekening gehouden met kleurkeuze, pagina-indeling en UX-design principes om de pagina zo leesbaar mogelijk te maken voor iedereen.

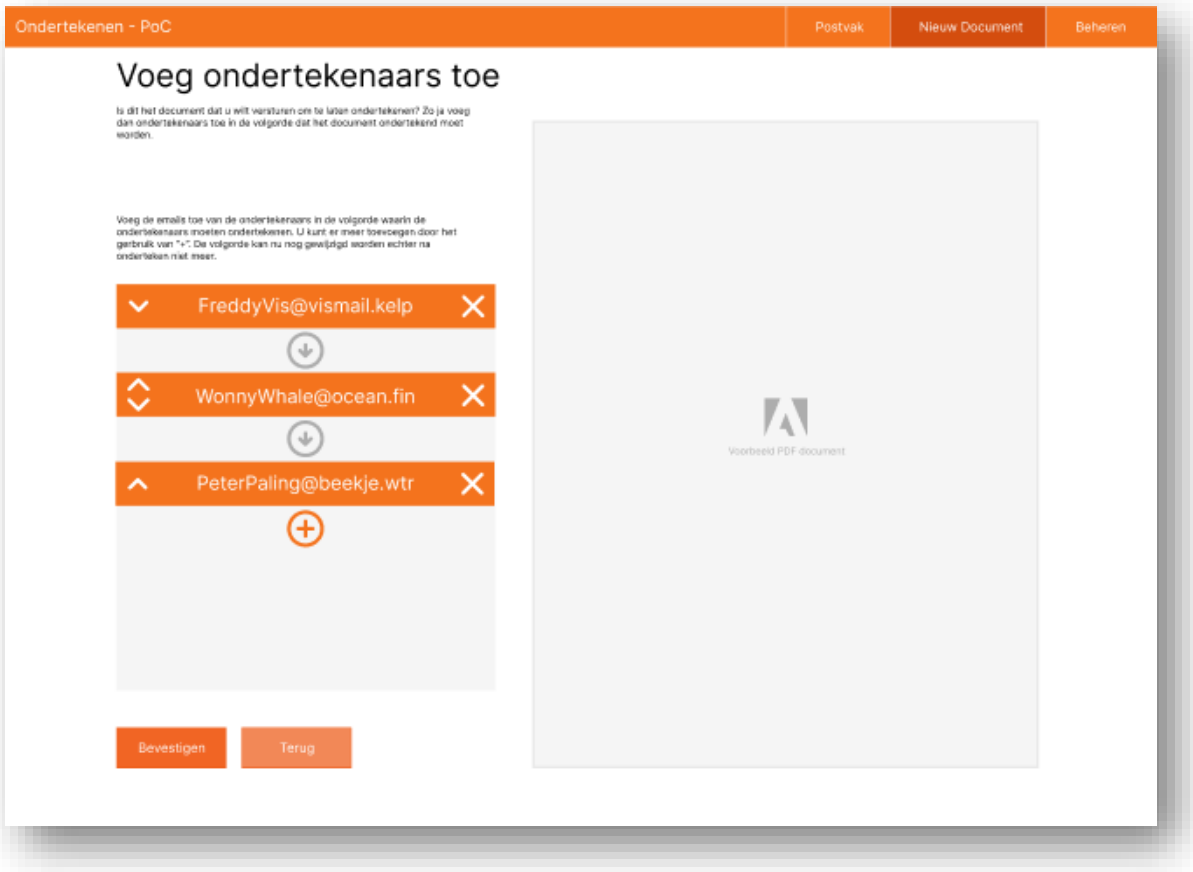
In het volgende figuur wordt het hoofdscherm afgebeeld, dat ook dient als het documentoverzicht.



Figuur 8. Documentoverzicht ontwerp.

Het kleurenschema en de pagina-indeling is gebaseerd op de gebruikersinterface van een ander product van het bedrijf. Documenten waar de deadline van het tekenen nadert, laten met een rode tekst de resterende tijd zien om de gebruiker duidelijk te maken dat er urgentie is voor het afronden van dit document.

Een ander belangrijke pagina is waar de verstuurder ondertekenaars toevoegt aan een document. De gebruiker arriveert op deze pagina na het uploaden van een document. De verstuurder moet hier één of meerdere ondertekenaars toevoegen. Het ontwerp van de pagina is a afgebeeld in het volgende figuur.

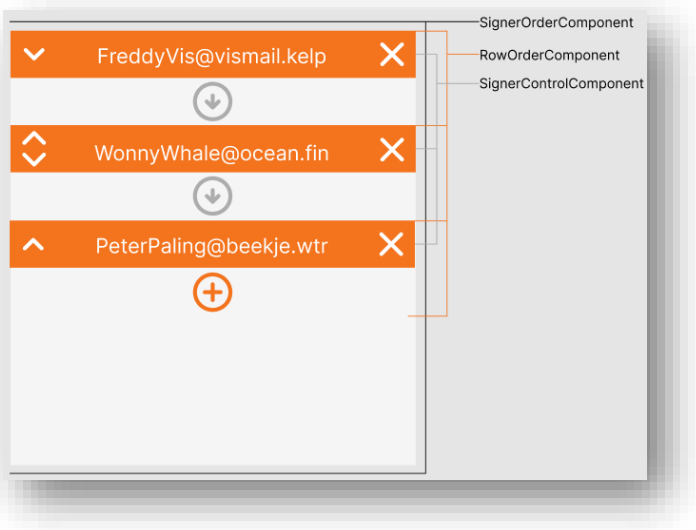


Figuur 9. Ondertekenaars toevoegen ontwerp.

In het rechter vak wordt een preview afgebeeld van het PDF document zodat de verstuurder nogmaals kan controleren of die het juiste document heeft geüpload. Aan de linker kant van de pagina kunnen ondertekenaars worden toegevoegd en de volgorde van ondertekenaars worden aangepast.

Met oranje is aangegeven wat de interactieve onderdelen zijn. De oranje blokken voor de e-mails kunnen worden gesleept om de volgorde aan te passen, maar bevatten ook pijltjes knoppen voor gebruikers die daar makkelijker kunnen werken.

Omdat er met React wordt gewerkt is er ook een componenten ontwerp gemaakt voor interactieve componenten die uit meerdere onderdelen bestaan. Voor het component waarin de ondertekenaars kunnen worden toegevoegd en aangepast, is er een componenten verdeling gemaakt. Deze is te vinden in het volgende figuur.

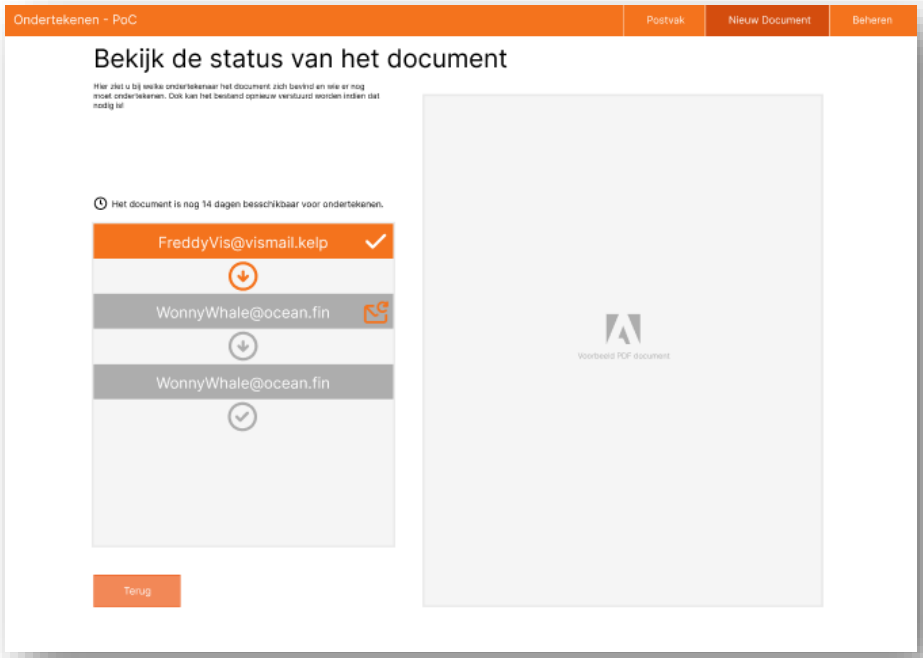


Figuur 10. Componenten verdeling ondertekenaars.

In dit ontwerp is rechts is van boven naar onder afgebeeld wat de componenten zijn van ouder naar kind component. In het kleinste component, hier genoemd “SignerControlComponent”, zit een tekstveld voor de email van de ondertekenaar.
Pijltjes knoppen voor het aanpassen van de volgorde:
Afhankelijk van de volgorde worden alleen de pijltjes knoppen getoond die een functie hebben. Bijvoorbeeld het laatste component zal geen pijltje naar beneden tonen omdat het geen positie lager kan hebben. Als laatst heeft dit component nog een knop met een kruis voor het verwijderen van die ondertekenaar uit de lijst.

Alle rij componenten, hier genoemd “RowOrdercomponent” bestaan uit het eerder genoemde component en een icoon. Het icoon van de laatste rij in de lijst is oranje en heeft een “+” icoon om aan te tonen dat dit aanklikbaar is. Bij klikken zal dit een nieuwe rij toevoegen aan de lijst met een nieuw oranje plusje. Het icoontje van de vorige regel verandert in een grijs pijltje om aan te tonen dat dit niet langer interactief is en nu alleen de volgorde van ondertekenen aangeeft.

Wanneer de gebruiker zijn document heeft verzonden en op een later moment de status van het document wil inzien, kan dit gedaan worden op de document status pagina. Hiervoor is ook een ontwerp gemaakt. Dit ontwerp is te vinden in het figuur op de volgende pagina.



Figuur 11. Ontwerp Status Pagina.

De ontwerpen van de status pagina en het scherm waar de gebruiker ondertekenaars kan toevoegen, lijken op elkaar om consistentie te forceren. Hierdoor kunnen ook React Componenten worden hergebruikt. Het component uit het vorige ontwerp kan wederom worden gebruikt. De logica van de control componenten zullen er anders uit zien, omdat dit andere functionaliteiten heeft. In dit scherm worden de knoppen gebruikt om opnieuw een uitnodiging te versturen naar de volgende ondertekenaar. Dit kan een verstuurder doen als herinnering of wanneer er een fout is opgetreden in het verstuur proces.

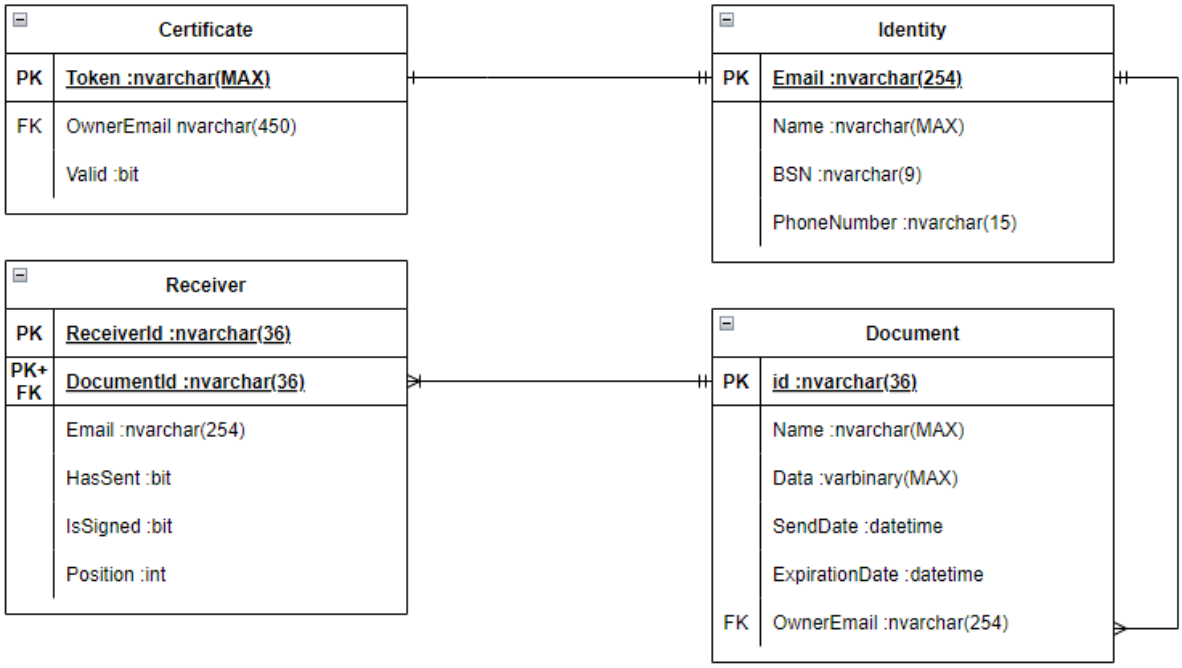
6.6 Database

Voor de databasestructuur is een ontwerp gemaakt. Het ontwerp is ontstaan uit het onderzoek dat is uitgevoerd. Als beginpunt is het document genomen, dit is het belangrijkste object in het systeem en onderdeel van het kernproces. Een document moet ook ondertekend kunnen worden en dit wordt gedaan door ondertekenaars, hier receivers genoemd.
Naast ontvangers is er ook een verstuurder die eigenaar is van het document nodig. De verstuurder moeten zich tevens kunnen identificeren om een document te ondertekenen.

Ondertekenaars krijgen elk een unieke uitnodigingscode om het document te kunnen ophalen in het systeem en te ondertekenen. De verstuurder moet een certificaat hebben voor het ondertekenen.

Al deze objecten komen voort uit de analysefase van het project. Met dit database model wordt elk onderdeel uit het onderzoek meegenomen in het systeem. De contactgegevens die de ondertekenaar invoert bij het ondertekenen worden niet opgeslagen in de database zelf. Deze informatie komt in de objecten die zijn verwerkt in het PDF-document te staan. Ook de controle op wijzigingen in het bestand wordt gedaan in het document zelf en staat buiten het systeem. Dit valt allemaal onder de functies van het PDF-Formaat.

Met deze informatie zijn er vier tabellen gemaakt voor in de database. Deze tabellen en de relaties daartussen worden afgebeeld in het volgende figuur.



Figuur 12. ERD Diagram database model.

Een document heeft alleen een ID, dit bevat:

- De naam van het document zelf;
- De binaire data van de inhoud;
- Een datum van moment van versturen;
- Een vervaldatum om een deadline te zetten voor het ondertekenen;
- De email van de eigenaar van het document.

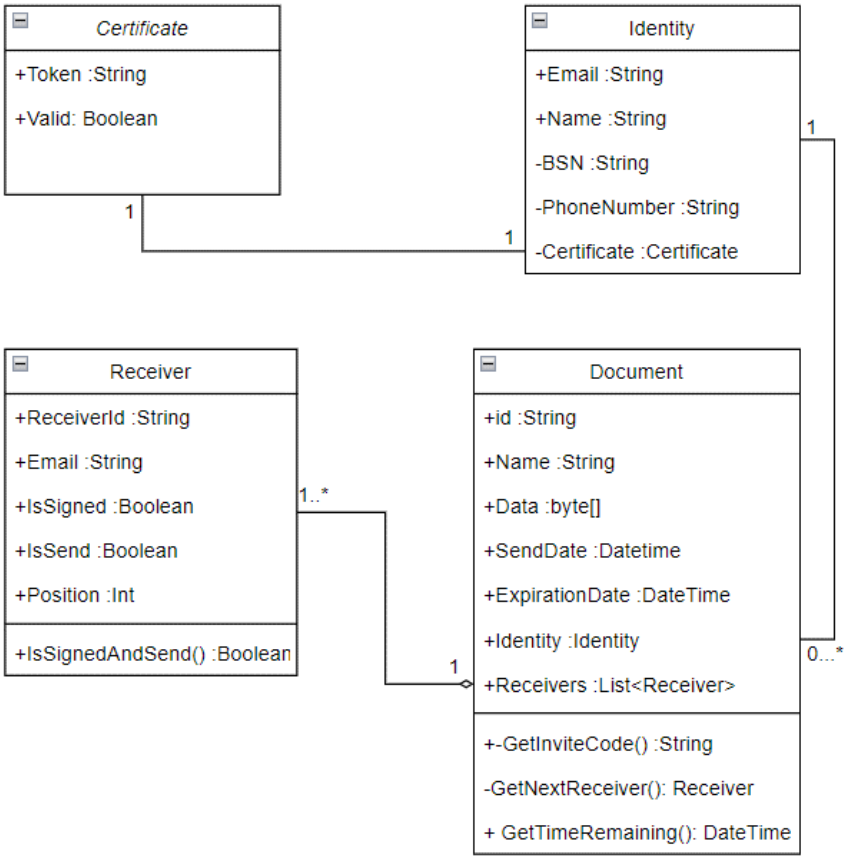
Een ontvanger, in de afbeelding genoemd “Receiver” is een ondertekenaar, die krijgt een unieke code bestaand uit document ID en ontvanger ID. Dit wordt gedaan zodat bestaande ontvangers aan een email kunnen worden gekoppeld en op die manier alle documenten kunnen worden opgevraagd voor die ontvanger. Deze functionaliteit valt niet binnen de scope van het proof-of-concept, maar voor eventuele doorontwikkeling waarbij ondertekenaars zich ook moeten identificeren

In de ontvanger wordt bijgehouden wat de status is van het document. Enkel als de ontvanger het verder afhandelt, kan het document worden doorgestuurd. Op die manier heeft het systeem meer controle over het bijhouden van de status van het document dan als je bijvoorbeeld alleen een huidige positie integer bij houdt. Als er dan een fout op treedt ergens in het proces waardoor de status niet goed wordt aangepast, kan er in de database achterhaald worden wat de meest recente status is.

6.7 Backend

6.7.1 Modellen

De backend wordt gerealiseerd met gebruik van het .NET framework. Voornamelijk houdt dat in dat de code wordt geschreven in C#. Met het gebruik van het Entity Framework in het kort EF, een framework binnen .NET, kan een verbinding worden opgezet met de database. Door de backend modellen te baseren op de database modellen kan het EF de database transacties vertalen naar C# objecten.
Dit werkt ook in omgekeerde richting door objecten om te zetten naar transacties richting de database.
Het model dat is gemaakt voor het EF in de backend ziet eruit zoals in het volgende figuur.



Figuur 13. Class diagram backend.

Zoals te zien is zijn een paar relaties tussen objecten anders dan bij het databasemodel. In dat model hangen documenten aan een identiteit. Dit is gedaan omdat het in de database gaat om het ophalen van lijsten aan data waarin vaak meerdere objecten samen komen. In de backend gaat het voornamelijk het verwerken en versturen van documenten. Om deze reden hangt er een identiteit aan het document.

De backend API zal niet deze complete objecten doorsturen naar de front-end. Alleen de waardes die gebruikt moeten worden in de front-end, daarom zullen de end-points andere uitvoer krijgen dan weergegeven in het model.

Dit wordt gedaan om ervoor te zorgen dat de transacties tussen front-end en backend zo klein mogelijk zijn voor de volgende redenen:

- Kleinere transacties voeren geen onnodige informatie aan de front-end en dit houdt de JSON-objecten overzichtelijk.
- Sensitieve non-essentiële data komt niet in het internet verkeer terecht tussen API en front-end.

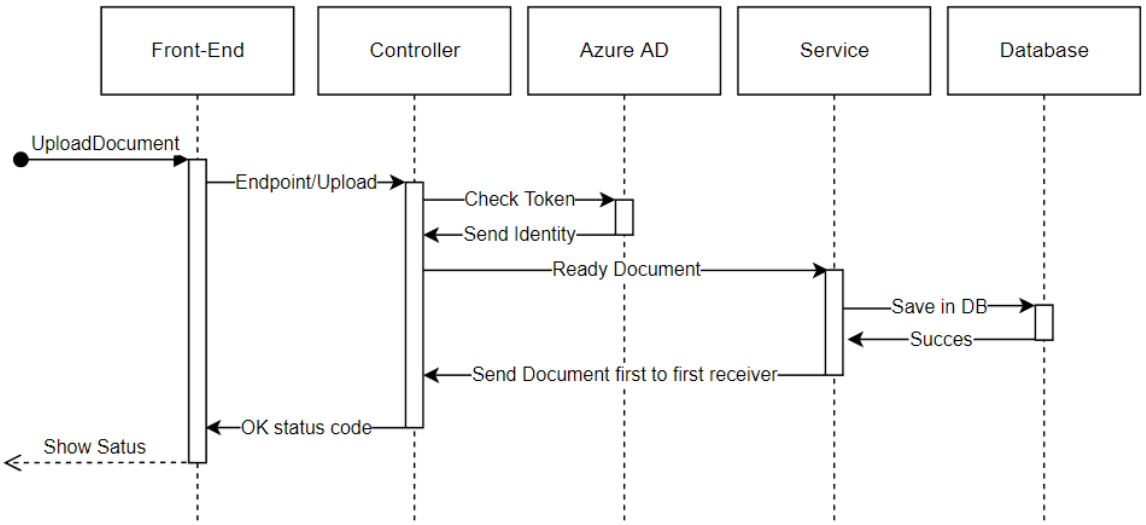
6.7.2 Controllers en services

Het .NET framework maakt gebruikt van controllers en services om structuur te geven aan de data-transacties van de API. De controllers bevatten de endpoints van de API en de logica voor het controleren van de HTTP-requests. Zoals bij het uploaden van een document, wordt er gekeken of het document daadwerkelijk het PDF-Formaat heeft, of er op zijn minst één ontvanger is en dat er een vervaldatum aanwezig is.

In de controllers worden de services aangeroepen om de controle logica en de verwerk logica te scheiden. De services hebben de code die het document verwerkt. Er wordt bijvoorbeeld ruimte gemaakt in het document voor het zetten van de handtekening. De services zorgen er ook voor dat de data uiteindelijk in de database terecht komt.

Bij API calls die vanuit de verstuurder komen, moet er een bearer token worden meegestuurd. Deze token wordt verkregen van Azure AD wanneer de gebruiker inlogt in het front-end systeem. Op die manier wordt er vanuit de backend gecontroleerd dat om een beveiligde aanvraag gaat waarbij de aanvrager een token heeft ontvangen van Azure. De token kan ook gebruikt worden voor het ophalen van de identificatie data van de ondertekenaar, zodat deze direct in de handtekening kunnen worden gezet.

In de volgende afbeelding wordt in een sequentie diagram afgebeeld hoe het upload proces eruit ziet vanaf front-end tot en met de database en terug.



Figuur 14. Sequentie Diagram van upload proces.

6.7.3 PDF Clown

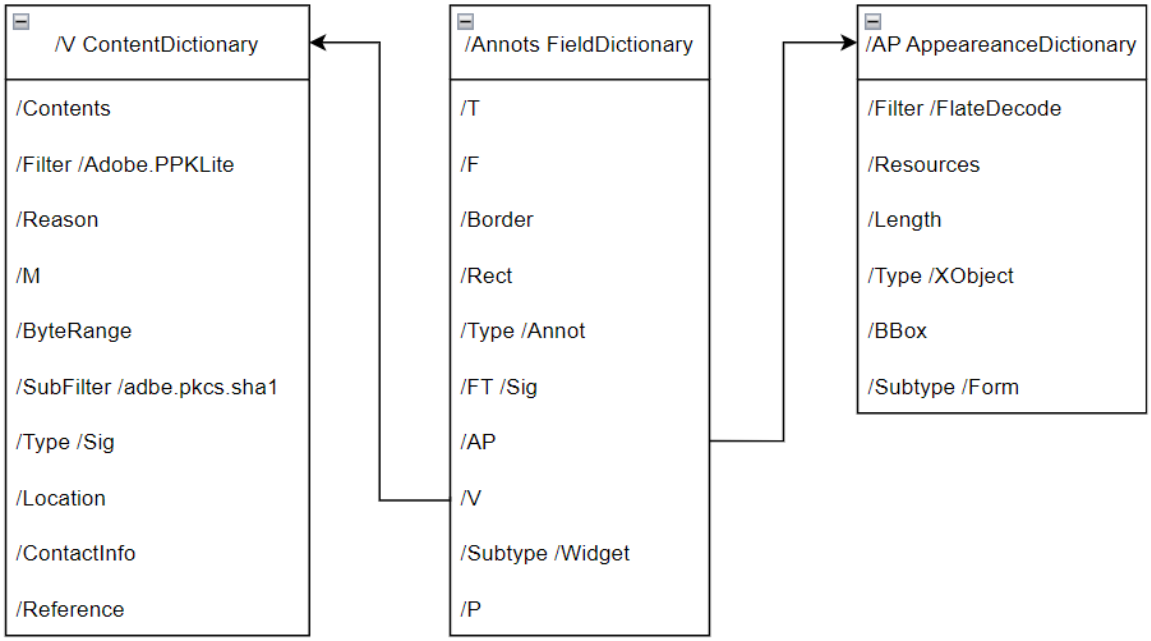
De PDFClown library zit in de backend van de applicatie en wordt gebruikt in de onderteken service. Zoals eerder benoemd bestaat een handtekening in een PDF document uit meerdere objecten. De library heeft hiervoor al een bestaande structuur. Aan deze structuur missen de objecten voor het digitaal ondertekenen, maar het bevat wel de basis objecten zoals collecties, teksten, nummers, booleans en datums. Hiermee kunnen de objecten worden gecreëerd die nodig zijn voor de uitbreiding voor de handtekeningen. Er is een nog het Dictionary object, dit is eigenlijk een object dat bestaat uit meerdere objecten.

Bijvoorbeeld bij het ondertekenen is er een Dictionary nodig voor het formulier. Op de achtergrond zijn handtekening een soort invoerveld binnen een formulier. Dat Dictionary object bevat dan bijvoorbeeld een collectie object met vier integers genaamd “Rect”, dit bepaalt bijvoorbeeld de X en Y coördinaten van het start punt van het formulier en de laatste 2 waardes de hoogte en de breedte van het element.

Vanuit de analyse van de Adobe PDF-specificaties is er een ontwerp gemaakt van een handtekeningenveld. In de PDF specificaties die Adobe biedt staat uitgelegd op welke manier PDF handtekeningen ondersteund. Hierin staat vermeld dat een handtekening een formulier veld is in het document en een aantal PDF objecten nodig heeft om te kunnen functioneren.

In de specificaties staat welke objecten er nodig zijn om een bepaalde handtekeningvorm te realiseren en welke objecten daar nodig bij zijn. Voor het ontwerp van de uitbreiding van PDFClown zijn alleen vereiste objecten meegenomen voor een gekwalificeerde handtekening.

Uit de specificaties blijkt dat het formulierveld uit meerdere objecten heeft waaronder 2 dictionary objecten. Dictionary objecten zijn collecties van andere objecten. Alle benodigde objecten zijn opgenomen in het volgende figuur.



Figuur 15. Ontwerp Objecten PDFClown.

De FieldDictionary is het object van het handtekening formulierveld zelf. Hierin zitten voornamelijk objecten voor de opmaak en structuur, maar ook twee andere dictionary objecten. “/V” dat staat voor Value, oftewel de waarde van de handtekening.

De waarde van de handtekening is het certificaat, locatie van ondertekenen, contact informatie, maar ook objecten die een PDF-viewer helpen met het detecteren van welke versleutelingsmethodes er zijn gebruikt. Dit is belangrijk omdat de PDF-viewer moet weten welke methode die moet gebruiken om het certificaat te ontsleutelen.

De FieldDictionary bevat ook een dictionary object genaamd “/AP” dit staat voor Appearance, oftewel de opmaak van het de handtekening. Denk bijvoorbeeld aan of het een plaatje van een handtekening is of een tekst en welk lettertype die tekst gebruikt.

Elk object begint met een “/”, zoals te zien is zitten er ook objecten in die nog een object hebben zoals bijvoorbeeld “/SubFilter /afbe.pkcs.sha1”. Dat betekent dat het tweede object de waarde is van het eerste, dit komt voornamelijk voor bij objecten die aan de PDF-viewer instructies geven.

Wanneer deze PDF objecten zijn toegevoegd aan de library moet er in de onderteken service in de backend de aangepaste PDFClown worden aangeroepen. PDFClown heeft een eigen class dat gebruikt wordt voor het bewerken van het document. De constructor van deze class heeft een binaire array nodig dat de data bevat van een PDF bestand.

Dan moet de service de eerder genoemde PDF objecten aanmaken en invullen met de data van de ondertekenaar of verstuurder. Dit zijn de contact gegevens, de handtekening als afbeelding of tekst en het certificaat.

De waarde van het certificaat moet worden versleuteld met de hash waarde van de content. De hashwaarde van de content. De PDF-viewer kan de content en certificaat onderscheiden aan de hand van de byterange.

De byterange bestaat uit 4 waardes. De eerste waarde is de start positie van het document, dit is de eerste byte van het gehele document, die is altijd “0”. De tweede waarde is het aantal bytes dat tussen de eerste byte en de laatste byte voor de handtekening bevindt. De derde waarde is de eerstvolgende byte positie na de handtekening en de vierde en laatste waarde is het aantal bytes tussen de derde waarde en het eind van het document.

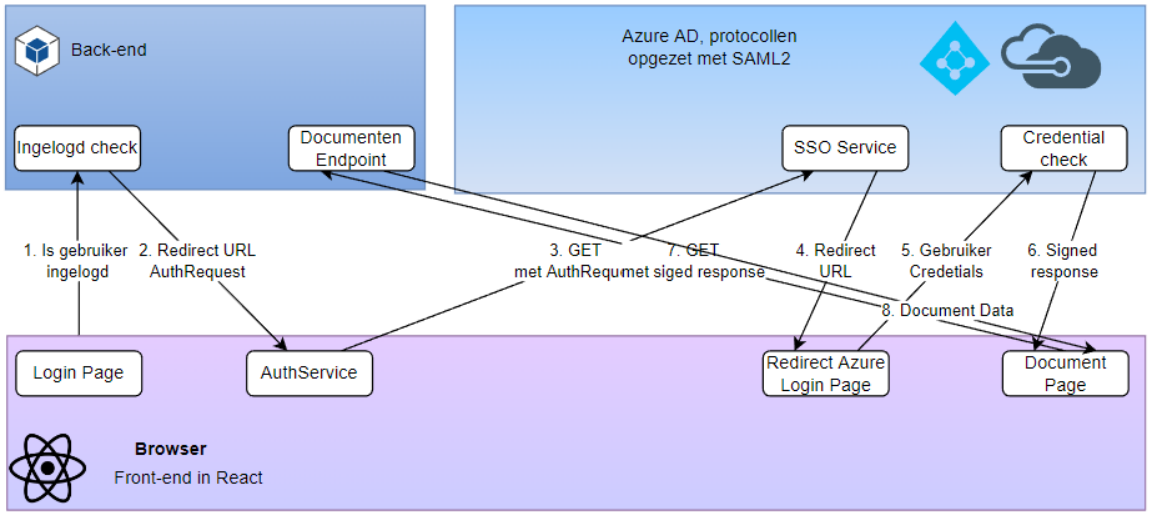
Het SubFilter object van de ContentDictionary wordt ingevuld met “afbe.pkcs.sha1”, zo dat PDF viewer weet welke hash methode er is gebruikt en kan de certificaat worden gelezen en gecontroleerd op geldigheid. Het is niet de taak van het systeem om dit te controleren, daarvoor zijn er erkende PDF-viewers.

Als alle objecten met de correcte waardes zijn toegevoegd aan het bestand moet het weer worden omgezet naar een binaire array zodat het kan worden opgeslagen in het model, dan kan het document verder verwerkt worden.

6.8 Azure AD en SAML2 koppeling

Azure AD wordt gebruikt als identiteitsprovider in het proof-of-concept. In project onderdelen wordt deze communicatie oppervlakkig beschreven. In dit hoofdstuk wordt er dieper ingegaan hoe die communicatie eruit ziet tussen Azure AD, front-end en backend.

Om in kaart te brengen hoe deze drie componenten met elkaar communiceren is er een diagram gemaakt voor de berichten volgorde en inhoud van die berichten bij het ophalen van een documentoverzicht. Dit ontwerp is te zien in het volgende figuur.



Figuur 16. Communicatie in stappen Azure AD.

De gebruiker kan het systeem via een browser benaderen. Als eerst moet er gekeken worden of de gebruiker al is ingelogd door een request te sturen naar de backend. Een authenticatieverzoek voor een doorverwijzings-URL terug gestuurd naar de front-end.

De front-end stuurt het authenticatieverzoek door naar Azure AD en als deze geaccepteerd wordt, dan wordt daarmee een URL gegenereerd. De URL wordt teruggestuurd en daarmee kan de inlog pagina van Azure getoond worden aan de gebruiker. De gebruiker logt in met zijn email, wachtwoord en 2-factor authenticatie middel. Dan krijgt de browser een getekend antwoord met een token. De browser kan met die token data ophalen uit de backend.

Zoals beschreven in het ontwerp van de projectonderdelen moet de backend ook nog afzonderlijk een connectie hebben met Azure AD. Dat is belangrijk voor het ophalen van identiteitsgegevens bij het plaatsen van de handtekening van de verstuurder.

Om SAML2 in te stellen zodat de protocollen gelijk zijn aan EHerkenning en DigiD is er een lijst gemaakt met alle protocollen en attributen die moeten worden ingesteld in de Azure AD omgeving.

Attribuut	Omschrijving	Formaat	Vereist
urn:etoegang:1.9:attribute:Initials	Initialen van de voorna(a)m(en) van de gebruiker	String max 35	Nee
urn:etoegang:1.9:attribute:18OrOlder	Is gebruiker ouder dan 18	true/false	Nee
urn:etoegang:1.9:attribute:16OrOlder	Is gebruiker ouder dan 16	true/false	Nee
urn:etoegang:1.9:attribute:12OrOlder	Is gebruiker ouder dan 12	true/false	Nee
urn:etoegang:1.9:attribute:65OrOlder	Is gebruiker ouder dan 65	true/false	Nee
urn:etoegang:1.9:attribute:PlaceOfBirth	Geboorteplaats van de gebruiker	String max 40	Nee
urn:etoegang:1.9:attribute:Gender	Geslacht van de gebruiker	"M", "F", "U"	Nee
urn:etoegang:1.9:attribute:Email	E-mailadres van de gebruiker	URI, max 320; mailto:<localname>@<domainname> (RFC6068)	Nee
urn:etoegang:1.9:attribute:Mobile	Mobiele nummer van de gebruiker	URI, max 19; tel:+<CountryCode><phonenumber> (RFC3966)	Nee
urn:etoegang:1.9:attribute:FirstName	Beschikbare voorna(a)m(en) van de gebruiker	String max 200	Ja
urn:etoegang:1.9:attribute:FamilyName	Achternaam (geslachtsnaam) van de gebruiker	String max 200	Ja
urn:etoegang:1.9:attribute:DateOfBirth	Geboortedatum van de gebruiker	String 10 jjjj-mm-dd Ook toegestaan: jjjj-mm-dd, jjjj-mm-00, jjjj-00-00, 0000-00-00	Ja
urn:etoegang:1.9:attribute:FamilyNameInfix	Voorvoegsel behorend bij Achternaam, conform Voorvoegseltabel	String max 10	Ja (Als er een tussen-voegsel is.)

Als al deze attributen zijn toegevoegd aan het SAML2 protocol in Azure AD is het gelijk aan DigiD en EHerkenning. Dan kan bij doorontwikkeling Azure AD worden vervangen door DigiD of/en EHerkenning met minimale aanpassing.

7 Realisatie

7.1 Database

Vanuit het database ontwerp is er een database server opgezet met gebruik van het Microsoft Azure Portal. Azure Portal biedt de mogelijkheid om meerdere systemen te beheren die binnen het Azure hangen. In dit project is dat de Azure AD omgeving en de database.

Met gebruik van een wizard worden de gebruikersrollen en andere opties van de database ingesteld. Na dat de server is ingesteld, wordt die direct live gezet in een online omgeving. Met het gebruik van Microsoft SQL Server Management Studio kan er verbinding gemaakt worden met de database, die nog helemaal leeg is. De tabellen uit het ontwerp worden toegevoegd aan de database. Deze tabellen hebben nog alleen een naam maar geen velden.

SQL Server Management Studio biedt een designfunctie, waarin gemakkelijk en snel kolommen kunnen worden toegevoegd aan een tabel. De datatypen en beperkingen worden overgenomen van uit het ontwerp. Wanneer alle tabellen inclusief kolommen zijn toegevoegd, worden de relaties aangebracht in de database door ForeignKeys te zetten tussen tabellen.

7.2 Azure AD

In het Azure portaal is een Azure AD test omgeving opgezet. Dit houdt in dat er test bedrijf is gemaakt met accounts die eigen emailadressen en rollen bevatten. Deze omgeving is ingesteld dat gebruikers verplicht zijn om met 2-factor authenticatie in te loggen. Het protocol voor de authenticatie is ingesteld op SAML2. Voor het testen van de applicatie zijn er 4 test gebruikers opgezet.

De protocollen zijn overgenomen uit het ontwerp en ingesteld in de Azure AD test omgeving. Dit betekent ook dat elke test gebruiker test data moesten krijgen voor bijvoorbeeld naam, achternaam en BSN.

7.3 Front-end

De front-end is geschreven in React en daarvoor is Visual Studio Code gebruikt. Dit is een gratis, moduleerbare code-editor, waarin gemakkelijk modules kunnen worden toegevoegd voor het ondersteunen van het ontwikkelproces van React, zoals bijvoorbeeld een React AutoComplete, React Code-Formatter.

Voor het inloggen is gebruikt gemaakt van de Azure AD NuGet Package om de gebruiker de identificatie pagina te tonen van Azure AD waar die moet inloggen met 2-factor authenticatie. Verdere afhandeling van de identiteit wordt gedaan door de backend.

Om de standaard componenten te maken is Material UI gebruikt. Dit is een component library voor React om standaard componenten toe te voegen zoals het menu, knoppen, lijsten, formulieren en input velden.

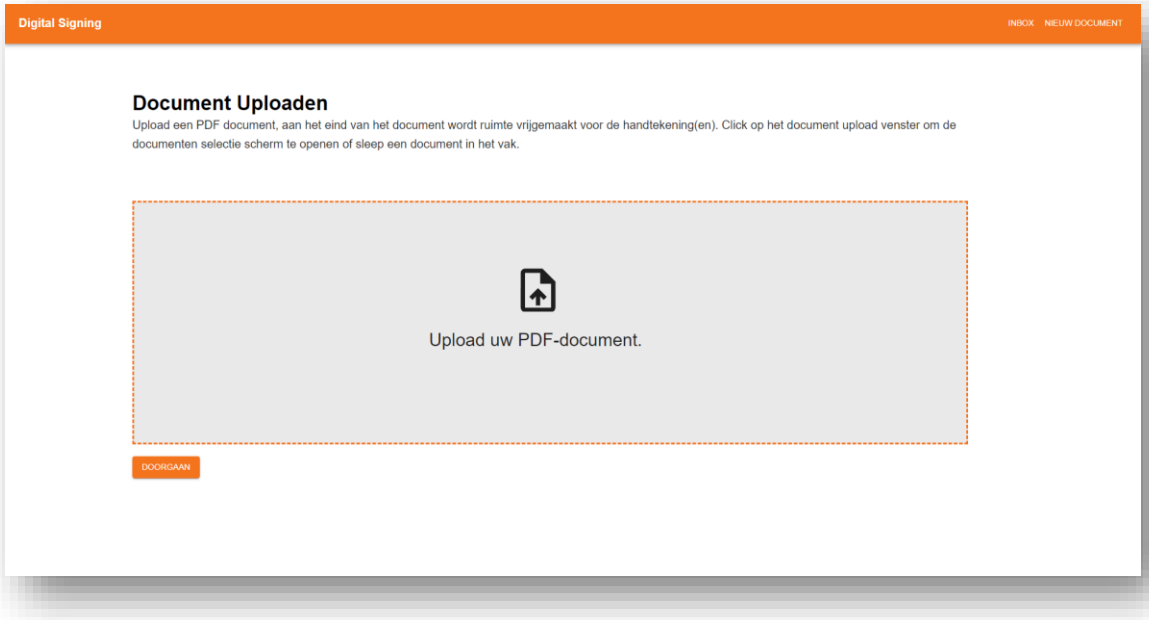
Material UI wordt gebruikt omdat het, het ontwikkelproces versnelt omdat het de ontwikkeltijd van standaard componenten wegneemt. Daarnaast zijn de componenten robuust en volgen ze React conventies om de kwaliteit van de code hoog te houden.

Het kan niet worden gebruikt bij het upload component en de ondertekenaar volgorde component. Deze zijn zelf gemaakt om zo dicht mogelijk bij het ontwerp te blijven.

Er is ervoor gekozen om styling classes te maken, hierdoor staat alle styling buiten de componenten code en kan het worden hergebruikt op verschillende componenten. Dat wordt gedaan om de componenten klein te houden en dus overzichtelijk, maar ook om te voorkomen dat styling code meerdere keren moet worden geschreven voor dezelfde soort componenten.

Elke pagina maakt gebruikt van dezelfde styling class genaamd “Theme” oftewel thema, op die manier wordt ervoor gezorgd dat er consistentie is in de styling van de verschillende pagina’s. Als het kleurenthema veranderd moet worden, kan dit worden gedaan op een centrale plek. Op die manier hoeft niet elke pagina afzonderlijk te worden aangepast bij ene thema wijziging.

De realisatie van de upload pagina is te zien in het volgende figuur.



Figuur 17. Realisatie Upload Pagina.

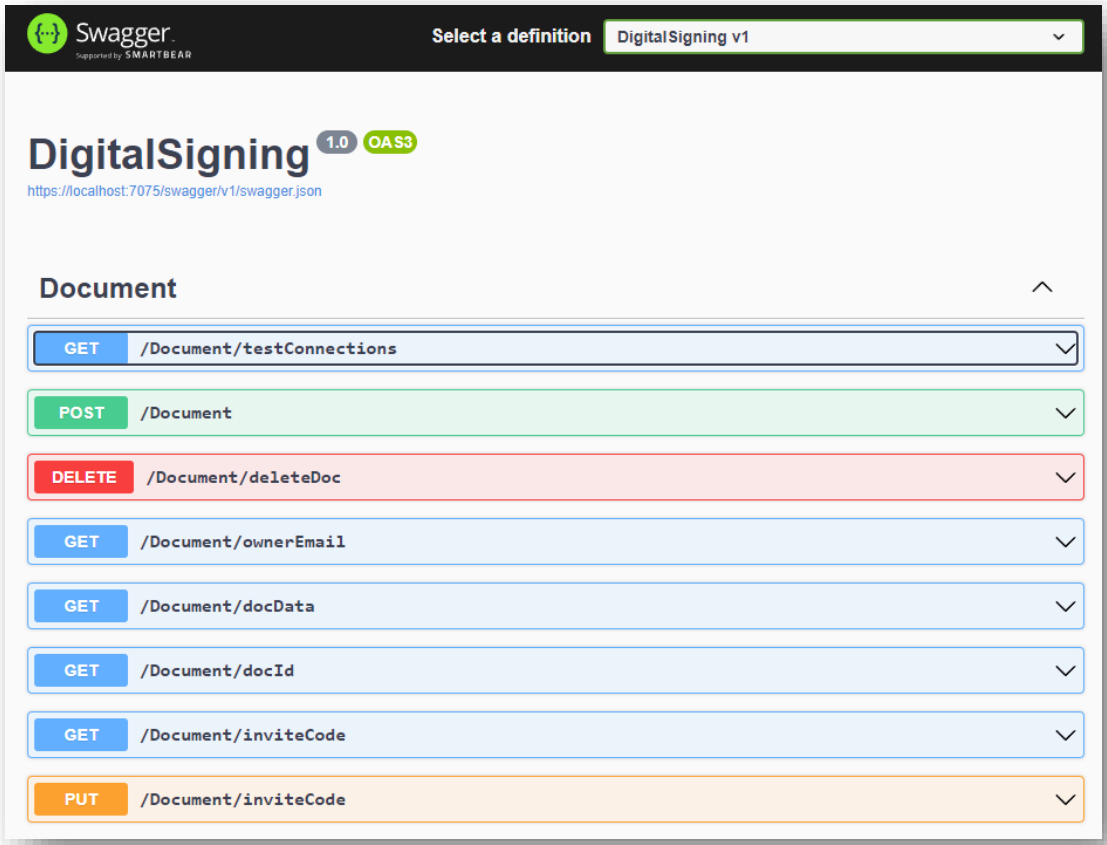
Zoals te zien is in het volgende figuur is er geprobeerd zo dicht mogelijk bij de ontwerpen te blijven. Iconen zijn veranderd, omdat in het ontwerp de iconen zijn gebruikt van de ontwerpsoftware en in de realisatie zijn de iconen gebruikt die worden aangeboden door Material UI.

7.4 Backend

De backend is geschreven in C# en wordt ondersteund door het .NET framework. Voor het ontwikkelen van deze code is JetBrains Rider gebruikt als code-editor. Dit is een moderne editor voor het ontwikkelen van .NET applicaties.

7.4.1 End-points

Voor het project is er een template gebruikt van .NET voor het maken van een API, hiermee worden er automatisch onderdelen gegenereerd zoals een voorbeeld controller en service. Dit template bevat ook een webpagina die wordt gegenereerd bij het runnen van de applicatie, op deze webpagina worden de verschillende end-points afgebeeld. Deze end-points hebben invoervelden waarmee de API-calls kunnen worden getest.



Figuur 18. Swagger End-points voorbeeld.

Swagger laat ook zien welke JSON objecten er worden verwacht en wat er wordt ontvangen, op die manier functioneert het als zowel end-point documentatie en test omgeving voor de API calls.

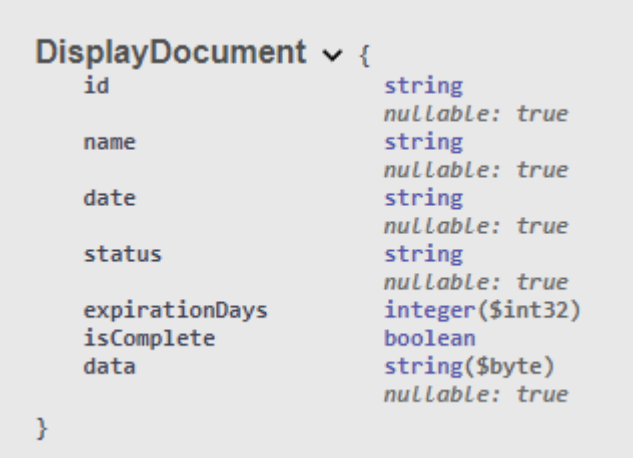
Wanneer een API call binnenkomt voor het uploaden en versturen van een document, wordt het document gecontroleerd of het een valide formaat heeft door PDFClown het bestand te laten uitlezen. Als PDFClown het bestand niet kan lezen wordt er een error gegenereerd dat wordt afgevangen. Het HTTP-verzoek wordt afgewezen met als reden dat het document niet een valide PDF-document is.

Ook de ingevoerde gegevens van de gebruiker worden gevalideerd door te kijken of die gevuld zijn. De contactgegevens die niet veranderen en gekoppeld zijn aan het certificaat van de verstuurder zoals de voor- en achternaam worden opgehaald uit Azure AD een geldige token en hoeven dus niet gecontroleerd te worden. De e-mails van de ontvangers worden gevalideerd met een REGEX expressie zodat het document daadwerkelijk verstuurd kan worden.

Als het verzoek valide is wordt de onderteken service aangeroepen om de certificaat en gegevens te plaatsen in het document. Als hier geen fouten in optreden wordt de verstuur service aangeroepen om het document naar de eerste ondertekenaar te sturen.

Om ervoor te zorgen dat niet alle data in de front-end komt zijn er modellen gemaakt voor de classes die naar de front-end gaan. Dit is gedaan om ervoor te zorgen dat bijvoorbeeld bij het ophalen van de document status, niet alle ID’s van de ontvangers worden meegestuurd voor veiligheidsredenen. Ook wil je niet dat een complete gebruiker wordt gestuurd, omdat die gegevens zoals zijn BSN bevat.

Een voorbeeld hiervan is het document om weer te geven in de documentenlijst van de gebruiker. In plaats van een verloopdatum en een creatie datum, wordt voor het verzenden het aantal dagen tussen de creatie datum en de verloopdatum. En de status wordt berekend aan de hand van wie er al allemaal hebben ondertekend. Zo wordt ervoor gezorgd dat de front-end zo min mogelijk hoeft te verwerken en alleen wordt gebruikt voor het weergeven van de data. Dit voorbeeld is afgebeeld in het volgende figuur dat het JSON-schema bevat van een weergave document.



Figuur 19. JSON schema weergave document.

Ook voor het verzoek om een document te uploaden ziet er anders uit dan het een document model. Dit komt omdat onderdelen van de data worden ingevuld in de backend. Bijvoorbeeld de creatie datum wordt gezet na het document is geüpload in de. Voor de ontvangers worden alleen de mailadressen meegestuurd omdat de ID's en status worden gezet in de backend. Dit model is te zien in het volgende figuur.

Figuur 20. Upload Request Class

7.4.2 Services

In de backend zitten 4 services die elk hun eigen onderdeel van het proces verwerken, deze services zijn:

- De onderteken service.
- De verstuur service.
- De authenticatie service.
- De database service.

Elke service is verantwoordelijk over zijn eigen afgesloten processen, daardoor blijft de code overzichtelijk, zijn fouten makkelijker te traceren en is er ruimte voor doorontwikkeling. Als bijvoorbeeld het bedrijf in de toekomst andere documentformaten dan PDF wilt ondertekenen, kan de huidige onderteken service hernoemd worden naar “PDFSignService” en er een nieuwe “WordSignService” worden toegevoegd. **Onderteken service**
De onderteken service maakt gebruik van de PDFClown library. Deze service plaatst het certificaat en de handtekeningen in het document. In hoofdstuk 7.5 wordt dit proces uitgebreid belicht.

Verstuur service
De verstuurservice zorgt ervoor dat er unieke uitnodigingscodes worden gegenereerd voor elke ondertekenaar van een document. De uitnodigingscode wordt gegenereerd met 2 GUIDS, het document ID en het ontvanger ID. Deze uitnodigingscode wordt verwerkt in een website link dat verwijst naar de front-end van de applicatie. Die link wordt in een mail naar de privé-mail van de ondertekenaar gestuurd.

Er wordt vanuit gegaan dat de ondertekenaar de enige persoon is die toegang heeft op dat emailadres. Dit wordt gedaan in veel moderne applicaties waar bijvoorbeeld een wachtwoord gewijzigd kan worden, door een link met unieke code dat ontvangen is op de mail van de gebruiker. De unieke code die gegenereerd wordt is 72 karakters lang, waarvan de eerste 36 karakters alleen bekend zijn bij de andere ondertekenaars van het document. Dit verlaagt de kans dat een document kan worden ondertekend door een kwaadwillende.

De verstuurservice houdt ook de status van het document bij. Als bijvoorbeeld een gebruiker heeft ondertekend maar door een fout in de mailserver niet het document kan doorsturen naar de volgende ondertekenaar. Vervolgens wordt de status aangepast naar ondertekend maar niet verstuurd. In dat geval kan het document worden opgeslagen in de database zodat de huidige ondertekenaar het niet opnieuw hoeft te ondertekenen en ziet de verstuurder dat het document niet verstuurd is naar de volgende ondertekenaar.

Als elke ontvanger heeft ondertekend wordt het afgeronde document verstuurd naar de mail van elke ondertekenaar. Dat is inclusief de verstuurder zelf.

Authenticatie service
De authenticatie service handelt het verkeer tussen Azure AD en de backend. Deze service haalt de contactgegevens van de verstuurder op uit Azure AD met HTTP verzoeken, wanneer een valide token wordt gegeven vanuit de front-end. Deze service maakt gebruik van de ASP.NET Core Identity en de ASP.NET Authentication WsFederation libraries van Microsoft zelf voor SAML2 SSO communicatie.

Als in de toekomst het bedrijf aansluit bij een EHerkenning inlogmiddel kunnen de end-points URL’s en authenticatie token worden aangepast met de nieuwe waardes van de EHerkenning leverancier. Dit kan naadloos worden geïntegreerd omdat de protocollen zijn overgenomen uit de officiële documentatie van Logius.

Database service
De database service maakt gebruikt van het Entity Framework om transacties te maken tussen backend en database. Het EF zet C# objecten om in SQL opdrachten bij het sturen van data naar de database. Het EF zet C# LINQ query's om in SQL query's die data uit de database halen.

Zo wordt bijvoorbeeld voor de functie om een document uit de database op te halen met het ID van het document opgeschreven als "db.Documents.FirstOrDefault(doc => doc.Id == "1111-1111-1111-1111")" en maakt EF daar een select query van dat in de tabel met documenten het document ophaalt met ID "1111-1111-1111-1111". Dit is gemakkelijker voor de ontwikkelaar om niet tussen SQL en C# te hoeven veranderen tijdens het ontwikkelen.

7.5 PDFClown

7.5.1 Aanpassen PDFClown

Zoals beschreven is in de analyse en het ontwerp mist in de PDFClown library de functionaliteit om handtekeningen toe te voegen. Er is wel een class aanwezig voor het handtekeningenveld met een lege constructor. Aan deze class is een PDF Dictionary veld toegevoegd dat een verplichte parameter van de constructor is. De overige functies die de interface verplicht aan de class zijn overgenomen en aangepast, zodat de PDF Dictionary uitgelezen kan worden bij het generen van het PDF.

PDFClown maakt gebruikt van een class waarin alle PDF-objecten staan in key-value paren. De keys zijn namen die makkelijker zijn voor ontwikkelaars om te begrijpen en de values zijn object namen dat in het bestand worden gezet, die leesbaar zijn in het PDF formaat. Alle objecten die besproken zijn in het ontwerp zijn toegevoegd aan deze class, op die manier begrijpt PDFClown wat die moet genereren bij het toevoegen van een handtekeningen veld.

7.5.2 Inzetten PDFClown

De nieuwe PDFClown library is daarna toegevoegd aan de backend en aangeroepen in de onderteken service. Daar wordt een PDF-document omgezet naar een document object dat PDFClown kan lezen. Een PDF Dictionary is aangemaakt en daaraan zijn alle besproken objecten toegevoegd. De objecten waarin details van de ondertekenaar staan worden ingevuld vanuit de informatie die wordt meegegeven vanuit de controller.

Het PDF bestandsformaat is ontworpen met als uitgangspunt dat het niet aanpasbaar is. Een aanpassing in een PDF document is daarom eigenlijk altijd een compleet nieuw bestand dat het oude bestand overschrijft met de nieuwe aanpassingen.

Er wordt ruimte gereserveerd voor het certificaat door een placeholder certificaat toe te voegen. Het certificaat wordt erin gezet als een lijst van bytes. De lijst van bytes wordt nu nog gevuld met “0” waardes met de maximale lengte wat een certificaat heeft. De echte certificaat kan er nog niet worden ingezet omdat deze moet worden versleuteld met de byterange van het document dat op dit moment in het proces nog niet bekend is, omdat het bestand nog moet worden aangepast.

Om het certificaat te kunnen versleutelen moet de byterange bekend zijn. De byterange is een lijst van 4 byte waardes:

- De byte positie voor het begin het document, dit is altijd 0, omdat daar het document begint.
- Het aantal bytes tussen het begin van het document en de laatste byte voor de handtekening.
- De eerste byte positie na het handtekening object en waard de meta data begint.
- Het aantal bytes tussen de eerste positie na de handtekening en het einde van het document.

Elk document heeft een andere grootte, daarom moet de byterange berekend worden. In die berekening moet de byterange grootte zelf ook worden meegenomen. In een klein document dat bijvoorbeeld maar 1000 bytes heeft in totaal, zou de byterange “[0, 600, 800, 1000]” kunnen zijn. In een groter document met 10.000 bytes kan de byterange “[0, 6000, 8000, 10000]” zijn, omdat elke waarde behalve de eerste een 0 extra heeft betekend dit dat het dit byterange object 3 meer bytes bevat, moeten die bytes ook worden meegenomen in de berekening.

Daarom moet er ook placeholder byterange worden gemaakt met berekende waardes. Het is belangrijk dat deze tijdelijke data exact zoveel bytes heeft als dat het document gaat hebben met de correcte byterange. Dit wordt gedaan door het document eerst op te slaan met de placeholder certificaat en byterange die dezelfde byte grootte hebben als dat definitieve objecten gaan hebben.

De placeholder byterange van het kleine document zou dan “[0, 100, 100, 1000]” worden en “[0, 1000, 1000, 10000]” bij het grotere document. Het document wordt weer opgeslagen zodat de placeholder byterange met de correcte lengte in het document staat.

Met de placeholder byterange in het document wordt de echte byterange berekend en met die waarde wordt het certificaat gehashed en in het document geplaatst samen met de echte byterang. De hash van de certificaat is in de praktijk altijd kleiner dan placeholder, daarom wordt aan het eind van de bytes van het certificaat “0” bytes toegevoegd tot het dezelfde lengte heeft als de placeholder. De “0” bytes worden in de PDF-viewer niet meegenomen in het uitlezen van het certificaat, maar zijn wel belangrijk voor het ontsleutelen van het certificaat. Als er 1 “0” byte te veel of te weinig wordt toegevoegd klopt de byte range niet meer en komt er een andere hash waarde uit de encryptie van de content. Een andere hash betekend dat het certificaat niet langer gelezen kan worden een PDF-viewer.

Om erachter te komen waar de handtekening begint wordt PDFClown ingezet om het bestand uit te lezen op byteniveau. Omdat het veel tijd kost om elke byte uit te lezen tot het einde van content gevonden is wordt, wordt eerst de dictionary van de handtekening gezocht.

PDFClown heeft een lijst met alle dictionaries in het document. In die lijst wordt gezocht naar de dictionary waarin de het handtekening veld zich bevindt. Dit wordt gedaan met de LINQ statement in het volgende figuur.

```
// Get a list of the dictionaries from the temp file and a list of references which contains the offsets of each dictionary.
var refEntries:List<XRefEntry> = pdfFile.Reader.ReadInfo().XrefEntries.Where(x:KeyValuePair<int,XRefEntry> =>
    pdfFile.Reader.Parser.ParsePdfObject(x.Value) is PdfDictionary dict &&
    Equals(dict[PdfName.FT], PdfName.Sig)).Select(x:KeyValuePair<int,XRefEntry> => x.Value).ToList();
```

Figuur 21. LINQ statement ophalen handtekening dictionaries.

Deze lijst bevat de byte locatie van de dictionary van de handtekening. Daarna kan er per PDF object door het document worden gegaan tot het Contents object wordt gevonden. Een PDF document kan meerdere Contents objecten hebben omdat het ook buiten handtekeningen worden gebruikt, daarom kan daar ook niet specifiek naar gezocht worden. Een handtekening dictionary heeft maar 1 content object. Als het Contents object gevonden is kan de byte positie worden opgevraagd voor de tweede waarde in de byterange. In het figuur op de volgende pagina wordt afgebeeld hoe dit wordt uitgevoerd.

```
var pdfParser = pdfFile.Reader.Parser;

var contentByteOffset = 0;
var signatureLength = 0;
if (refEntries.Any())
{
    pdfParser.Seek(refEntries.LastOrDefault().Offset);
    while (contentByteOffset == 0)
    {
        var pdfObj :object? = pdfParser.Token;

        if (pdfObj != null && pdfObj.ToString() == "Contents")
        {
            // Add 1 to go to next start of next token, which is the certificate.
            contentByteOffset = (int)pdfParser.Stream.Position + 1;
            pdfParser.MoveNext();
            signatureLength = pdfParser.Token.ToString().Length;
        }
        else
        {
            pdfParser.MoveNext();
        }
    }
}
```

Figuur 22. Code voor het vinden de 2e byterange waarde

De derde waarde wordt berekend door de lengte van het certificaat te nemen en die bij de tweede byterange waarde op te tellen. De laatste waarde wordt berekend door de derde waarde af te trekken van het aantal bytes dat het complete document heeft.

Als alle waardes correct zijn ingevuld en het document weer wordt opgeslagen. Dan is het document ondertekend met een certificaat en de contact gegevens van de verstuurder. Als een persoon dan alsnog het document aanpast, klopt de hash niet langer met de byterange en weet de PDF-viewer dat het document is aangepast.

Voor elke handtekening die daarna wordt gezet door een ondertekenaar wordt het proces van de byterange opnieuw doorlopen.

7.6 Testen

Voor het testen van het onderteken proces is er een aparte test applicatie gebouwd dat een aantal documenten van verschillende groottes laat ondertekenen. De bytegrootte van het document wordt opgeslagen na het zetten van de placeholder handtekening en de definitieve handtekening. Deze bytegroottes worden met elkaar vergeleken en als daar verschil tussen zit, betekent het dat de byterange niet goed is berekend of dat er door byte conversie er bytes zijn bijgekomen bij de generatie van het document. Wanneer de byte grootte hetzelfde blijft betekent het, dat het document succesvol is geconverteerd. De test applicatie slaat de documenten op in het project zodat er nogmaals met de hand kan worden gecontroleerd of de handtekening leesbaar is in een PDF-viewer.

Ook zijn er op verschillende plekken in het systeem log functies gebouwd dat de grootte van het document weergeven wanneer het project in debug modus staat. Op die manier wordt er getest of de grootte van het document veranderd wanneer het wordt geconverteerd naar een ander formaat. Dit wordt gedaan voor het geüpload wordt in de front-end en wanneer na het weer naar een PDF document wordt geconverteerd in de backend. De bytegrootte wordt ook gecontroleerd na het ondertekenen, voor

het opgeslagen wordt in de database en na het ophalen van het document uit de database. Als die bestandsgroottes hetzelfde blijven betekent het dat de converteer methodes het bestand niet hebben aangetast.

Het testen van de front-end is gedaan door een grote set aan mock documenten te maken met verschillende valide en invalide waardes. Door bijvoorbeeld het aantal dagen tot een document verloopt op een negatieve waarde te zetten. Maar ook documenten met onmogelijke status combinaties zoals een document dat wel is verstuurd naar de volgende verstuurder maar niet is ondertekend door huidige ondertekenaar. Door het project in debug modus te runnen worden deze gebruikt, dan kan in de applicatie zelf gekeken worden of dit voor onverwacht resultaat zorgt.
De API zelf heeft ook een test end-points om te kijken of de die verbinding heeft met Azure AD, de database en de mailserver. Deze end-point geeft dan als antwoord een lijst van deze externe diensten en een “Ok” waarde als die connectie kon maken. Bij een fout geeft deze end-point terug tegen welke error die aanliep bij het testen van de verbinding. De front-end kan deze end-point gebruiken om een gebruiker in te lichten dat documenten tijdelijk niet kunnen worden geüpload of verstuurd.

8 Conclusie

Digitaal ondertekenen is een proces waarin een digitaal document wordt ondertekend met een digitale handtekening. Er zijn verschillende vormen van een digitale handtekening. Een document kan verschillende vormen van digitale handtekeningen bevatten en afhankelijk van welke juridische doeleinden documenten hebben, staat er in de wet vastgelegd welke vormen verplicht zijn.

Met het proof-of-concept is aangetoond dat digitaal ondertekenen in PDF-documenten mogelijk is met een digitaal certificaat. Daarmee kunnen documenten zoals arbeidscontracten worden ondertekend die volgens de Nederlandse en Europese wetten juridisch valide zijn. Het certificaat en de methode waarop het PDF formaat dit gebruikt zorgt ervoor dat er geen aanpassingen kunnen worden gemaakt aan de inhoud van het document na het plaatsen van het certificaat.

Echter wordt in het proof-of-concept nog een zelf gegenereerd certificaat gebruikt dat in de praktijk moet worden vervangen met een certificaat dat is bemachtigd bij een vertrouwde certificaat uitgever. Hierdoor is de vorm van handtekeningen in het proof-of-concept geavanceerd maar nog niet gekwalificeerd. Daarvoor zou een gebruiker zo'n certificaat moeten bemachtigen met gebruik van EHerkenning of DigiD, daarom is Azure AD ingesteld om dezelfde protocollen te gebruiken als die diensten. Zo kan bij doorontwikkeling met minimale aanpassingen dat alsnog worden gerealiseerd.

Door het gebruik van een library dat PDF documenten kan inlezen en aanpassen op laagniveau is er meer inzicht verkregen in het genereren van PDF documenten. Op welke manier de handtekeningen worden gezet en welke informatie in de handtekeningen wordt geplaatst is volledig te beheren vanuit de applicatie. Zo kan het bedrijf bepalen of zij een verwijzing in de handtekening willen achterlaten dat bijvoorbeeld het document ondertekend is met hun product.

Met het proof-of-concept en de bijbehorende analyse kan het afstudeerbedrijf het product verder ontwikkelen en integreren met de bestaande systemen.

9 Reflectie

Tijdens mijn afstudeerperiode ben ik in aanraking gekomen met een onderwerp waar ik eigenlijk helemaal niks van af wist, behalve dat ik op maandelijkse basis een digitale handtekening had gezet of heb laten zetten met een ander systeem bij een vorige werkgever. Technische gezien had ik geen idee hoe dat op de achtergrond eruitziet.

Door mij te verdiepen in de verschillende deelvragen en te kijken naar wat de huidige oplossing was die het afstudeerbedrijf biedt, ben ik erachter gekomen dat er heel veel mogelijk is op het gebied van digitaal ondertekenen. Ik had bijvoorbeeld niet verwacht dat een digitale handtekening helemaal geen krabbel meer hoeft te hebben en het eigenlijk veel meer draait om certificaten en de vertrouwde instanties die onderdeel maken bij de totstandkoming van de handtekening. Ik heb niet alleen veel geleerd over ondertekenen, maar ook over hoe bedrijven samenwerken met overheidsinstanties en wat het belang daarvan is bij internationaal handelen in de Europese Unie.

Hoewel ik wel vaker had gemerkt bij andere applicaties, zoals GIT, waarbij digitale certificaten worden gebruikt, heb ik door dit onderzoek en realisatie meer geleerd over hoe de certificaten worden gecontroleerd en wat het belang van die certificaten is bij veilig handelen tussen servers en gebruikers.

Er is ook meer inzicht verkregen in hoe PDF-documenten zijn opgemaakt vanaf byte-niveau en welke slimme methodes Adobe gebruikt om documenten en de handtekeningen daarin te beveiligen. Hier lag ook een grote valkuil voor mij.

Ik had ervaring met document generatie maar ik wist niet dat het PDF formaat zo complex was. Het begrijpen en schrijven van PDF documenten zonder het gebruik van een library is een vak apart, weggelegd voor gehele ontwikkel teams met jaren aan ontwikkel tijd. Ik ook heb uitgevonden dat ik vaak veel te optimistisch ben in het inschatten van een tijdsbestek voor het ontwikkelen van features waar ik geen of weinig ervaring in heb. Dankzij mijn stagebegeleider die dan op de rem trapte konden we meer realistische tijden inschatten.

Gelukkig realiseerde ik mij op tijd dat het ontzettend lastig zou gaan worden om dit compleet zelf op te pakken en daarom hebben we onderzoek opgezet naar eventuele libraries die deel van het werk konden opvangen.

Over het onderzoek ben ik erg tevreden en dat heeft mij heel veel inzichten gegeven in digitale handtekeningen, digitale certificaten en hoe de overheid daar een rol in speelt. Ik merk wel dat ik te veel alleen heb proberen te doen. Er is binnen B-ware Business Software heel veel kennis over de functionele werking van digitaal ondertekenen en had vaker moeten vragen of wellicht een interview moeten doen om meer inzicht te krijgen in wat het resultaat is van een getekend document en hoe dit proces er bij hen uit ziet. In het project zijn de requirements voornamelijk kortgesloten met de opdrachtgever aan de hand van de onderzoeksresultaten. In de toekomst zal ik ook eerder naar eventuele gebruikers stappen om te vragen wat zij verwachten en wat zij inzien in zo’n proces om nog betere requirements vast te kunnen stellen.

Ook over de applicatie ben ik erg tevreden en denk dat ik een mooie basis heb neergezet voor doorontwikkeling en integratie in de producten van het bedrijf.

10 Literatuurlijst

Aansluiten op DigiD. (sd). Opgehaald van Logius.nl: https://www.logius.nl/diensten/digid/aansluiten-op-digid

Adobe. (2008, 7 1). *Document management Portable Document Format Part 1: PDF 1.7*. Opgehaald van opensource.adobe.com: https://opensource.adobe.com/dc-acrobat-sdk-docs/standards/pdfstandards/pdf/PDF32000_2008.pdf

Anderson, B., & Nicholson, B. (2022, June 12). *SQL vs. NoSQL Databases: What's the Difference?* Opgehaald van IBM.com: https://www.ibm.com/cloud/blog/sql-vs-nosql

Attribuutcatalogus natuurlijke personen. (sd). Opgehaald van afsprakenstelsel.etoegang.nl/: https://afsprakenstelsel.etoegang.nl/display/as/Attribuutcatalogus+natuurlijke+personen

bedrijfsprofiel. (sd). Opgehaald van b-ware.com: https://www.b-ware.com/over-b-ware/bedrijfsprofiel/

Betrouwbaarheidsniveaus. (sd). Opgehaald van Eherkenning.nl: https://www.eherkenning.nl/nl/eherkenning-gebruiken/betrouwbaarheidsniveaus

de Europese Unie. (2014, Augustus 28). *VERORDENING (EU) Nr. 910/2014 VAN HET EUROPEES PARLEMENT EN DE RAAD*. Opgehaald van EUR-Lex.europa.eu: https://eur-lex.europa.eu/legal-content/NL/TXT/PDF/?uri=CELEX:32014R0910&from=EN

Digital Certificates. (sd). Opgehaald van fortinet.com: https://www.fortinet.com/resources/cyberglossary/digital-certificates

Digital Signatures in a PDF. (2021, mei 30). Opgehaald van adobe.com: https://www.adobe.com/devnet-docs/etk_deprecated/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf

digitale handtekening. (sd). Opgehaald van Zynyo.com: https://zynyo.com/digitale-handtekening/

Eherkenning. (sd). Opgehaald van logius.nl: https://www.logius.nl/diensten/eherkenning

Endignoux, G. (2016, Oktober 4). *Introduction to PDF syntax*. Opgehaald van gendignoux.com: https://gendignoux.com/blog/2016/10/04/pdf-basics.html

etoegang.nl. (sd). Opgehaald van afsprakenstelsel.etoegang.nl/: https://afsprakenstelsel.etoegang.nl/

Features. (sd). Opgehaald van pdfsharp.net: http://www.pdfsharp.net/Features.ashx

Free .NET PDF Library. (sd). Opgehaald van marketplace.visualstudio.com: https://marketplace.visualstudio.com/items?itemName=E-iceblueCoLtd.FreeNETPDFLibrary

Hernandez, A. R. (2021, Mei 18). *Soorten digitale handtekeningen: het verschil tussen AES, QES en SES*. Opgehaald van DocuSign.nl: https://www.docusign.nl/blog/soorten-digitale-handtekeningen-het-verschil-tussen-aes-qes-en-ses

Jurado, C. B. (2020, November 18). *IS Git the best CVS ever?* Opgehaald van LeanMind: https://leanmind.es/en/blog/is-git-the-best-vcs-ever/#:~:text=Git%20has%20become%20the%20software,well%20as%20the%20corporate%20world.

Koppelvlakspecificatie DigiD SAML Authenticatie. (sd). Opgehaald van Logius.nl: https://www.logius.nl/diensten/digid/documentatie/koppelvlakspecificatie-digid-saml-authenticatie

missie-en-visie. (sd). Opgehaald van b-ware.com: https://www.b-ware.com/over-b-ware/missie-en-visie/

organisatie. (sd). Opgehaald van b-ware.com: https://www.b-ware.com/over-b-ware/organisatie/

over b-ware. (sd). Opgehaald van b-ware.com: https://www.b-ware.com/over-b-ware/

Over Logius. (sd). Opgehaald van Logius.nl: https://www.logius.nl/onze-organisatie/over-logius

PDF Clown 0.2.0 — Enhanced content handling. (2014, September 12). Opgehaald van pdfclown.com: https://pdfclown.org/2014/09/12/waiting-for-pdf-clown-0-2-0-release/#ContentRenderer

PDF/A. (2022, Februari 28). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/PDF/A

PDF/X. (2022, February 12). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/PDF/X

PDF/X-, PDF/A-, and PDF/E-compliant files (Acrobat Pro). (2021, Augustus 24). Opgehaald van Adobe.com: https://helpx.adobe.com/acrobat/using/pdf-x-pdf-a-pdf.html

PKI Overheid Hoe Werkt Het. (sd). Opgehaald van logius.nl: https://www.logius.nl/diensten/pkioverheid/hoe-werkt-het

PKIoverheid. (sd). Opgehaald van logius.nl: https://logius.nl/diensten/pkioverheid/aanvragen
PKIoverheid aanvragen. (sd). Opgehaald van logius.nl:
https://www.logius.nl/diensten/pkioverheid/aanvragen
PostScript. (2022, Maart 7). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/PostScript
Telecommunicatiewet. (2021, juli 1). Opgehaald van Overheid.nl:
https://wetten.overheid.nl/BWBR0009950/2021-07-01
The_DOT_Framework. (sd). Opgehaald van ictresearchmethods.nl:
https://ictresearchmethods.nl/The_DOT_Framework
Wet op de identifcatieplicht. (2017, Maart 1). Opgehaald van Overheid.nl:
https://wetten.overheid.nl/BWBR0006297/2017-03-01/#HoofdstukI_Artikel1