



Stage dotNET lab: DNL InfoSec Portal

Realisatie

Bachelor in de Toegepaste Informatica
Keuzerichting Application Development

Michiel Van Loy

Academiejaar 2024-2025

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

INHOUDSTAFEL	3
1 DANKWOORD	4
2 INLEIDING	5
2.1 Stagebedrijf	5
2.2 Opdracht	5
3 AANPAK EN VERLOOP	7
3.1 Planning	7
3.1.1 Verwachte planning	7
3.1.2 Werkelijke planning	7
3.2 Communicatie	7
3.2.1 Meetings	8
3.2.2 Vragen.....	8
3.3 Plan van aanpak	8
4 CURSUSSEN	9
4.1 Test driven development	9
4.2 SOLID fundamentals	9
4.3 TypeScript fundamentals.....	10
4.4 CI pipeline.....	10
4.5 Clean architecture	10
5 ANALYSE	11
5.1 Waarom.....	11
5.2 Hoe.....	11
5.3 Prototypes.....	12
5.4 Datamodel.....	13
6 CODEREN.....	14
6.1 Back-end	14
6.1.1 API.....	14
6.1.2 Application	14
6.1.3 Infrastructure	14
6.1.4 Domain	14
6.2 Front-end	15
6.2.1 Styling	15
6.2.2 Componenten	15
6.2.3 Libraries.....	15
6.3 Authenticatie en autorisatie	16
6.4 CI/CD Pipeline	16
6.5 Hosten.....	16
6.6 Resultaat.....	17
6.6.1 Gebruiker	17
6.6.2 Administrator.....	19
7 CONCLUSIE.....	29
8 LITERATUURLIJST	30

1 DANKWOORD

Voordat ik jullie mijn stageproject voorstel wil ik eerst wat mensen bedanken.

Om te beginnen wil ik mijn stageplaats bedanken, maar ook een aantal specifieke werknemers. Het eindresultaat is mede door hun tot stand gekomen. Allereerst wil ik Ive Verstappen en Jef Meijvis bedanken voor het verzorgen van een interessant, leerrijk project. Het was geen makkelijke opgave, maar dankzij goede uitleg en ondersteuning van hun is het tot een goed einde gekomen.

Ook wil ik de experts Tim Deschryver en Jamie Vereecken bedanken voor de goede input die ze gaven over het kiezen van bepaalde programmeertechnieken en libraries. Ook de uitleg die ze gaven was zeer goed en duidelijk wat alles toch net iets makkelijker maakte.

Ook wil ik vooral de junior developers Derre Evers, Rikko Keuppens, Simon Horemans en Dave Saenen bedanken voor het beantwoorden van vragen en helpen met het project. Zonder de ondersteuning die zij gaven was ik denk ik niet zo ver geraakt als nu. Ook hebben vooral zij ervoor gezorgd dat ik me thuis voelde bij dotNET lab.

Als laatste wil ik mijn stagebegeleider Michaël Cloots nog bedanken voor de ondersteuning doorheen de gehele stageperiode.

2 INLEIDING

Vanaf 16 september tot en met 13 december heb ik 13 weken stage gelopen bij dotNET lab. dotNET lab is een consultancy bedrijf gelegen in Heultje en is voor velen onder ons al gekend door de jaarlijkse samenwerking met Thomas More voor de Angular challenge. In dit document zul je alles te weten komen over mijn stage.

2.1 Stagebedrijf



Figuur 1: logo dotNET lab

Oorspronkelijk bestond dotNET lab uit twee bedrijven, namelijk IT4IT en Bitconsult, deze twee bedrijven hebben eind 2020 hun krachten gebundeld en dotNET lab, maar ook dotNET academy, opgericht.

Zoals ik eerder al heb vermeld is dotNET lab een consultancy bedrijf. Dit wil zeggen dat de programmeurs van dotNET lab niet alleen interne projecten maken, maar vooral externe projecten. Door de jaren heen zijn ze enorm gegroeid en hebben ze enorm veel ervaring opgedaan. Dit maakt dat dotNET lab ondertussen meer dan 60 gecertificeerde Microsoft professionals heeft en meer dan 20 jaar ervaring.

Sinds kort hebben ze ook een klein cybersecurity onderdeel binnen dotNET lab waarbij ze klanten gaan helpen en beoordelen op basis van hun security. Dit wil niet zeggen dat de klanten dotNET lab kunnen contacteren bij eventuele problemen, hiervoor zouden ze iemand anders moeten inhuren.

Ook dotNET academy is verder gegroeid vanuit IT4IT en Bitconsult en is eigenlijk een afdeling binnen dotNET lab zelf waarbij ze net afgestudeerde studenten de kans geven een extra opleiding te volgen om op deze manier makkelijker in het werkveld te rollen. Na dit traject starten ze meestal als junior developers bij dotNET lab om op deze manier hun eerste ervaring op te doen.

2.2 Opdracht

Mijn opdracht was het maken van een Engelse webapplicatie waarbij de klanten hun security scores konden zien. Dit natuurlijk in een mooi overzicht met de geschiedenis van de scores erbij om duidelijk te maken hoe de klant was geëvolueerd doorheen de verschillende kwartalen. Ook het invoeren van verbetering was onderdeel van deze applicatie. Zoals je al verwacht was dit een opdracht vanuit het cybersecurity deel van dotNET lab.

De werknemers, en eventueel externe security experts, moesten de mogelijkheid hebben een security score en verbeterpunten toe te voegen aan een specifieke klant. Deze klant moest op zijn beurt dan weer de mogelijkheid krijgen om zijn scores en verbeterpunten te bekijken.

Deze score werd berekend op basis van 18 verschillende eigenschappen van Center for Internet Security, afgekort CIS (CIS Center for Internet Security, z.d.). Deze eigenschappen worden controls genoemd en zijn 18 verschillende categorieën waarop de klant een score kreeg. Deze scores werden dan samengevoegd tot één score die de klant te zien kreeg op het scherm. Een voorbeeld van één van deze controls is "data protection" wat op zijn beurt is opgesplitst in nog een aantal onderdelen waaronder "Establish and maintain a data inventory". Zo is het voor elke control opgebouwd.

Zoals ik eerder al vermeldde was het de bedoeling een Engelse applicatie te ontwikkelen aangezien sommige werknemers binnen dotNET lab enkel Engels spraken, maar ook omdat ze een aantal buitenlandse klanten hadden die deze applicatie ook moesten kunnen gebruiken.

De focus van dit project lag vooral op de veiligheid van de applicatie. Doordat de security score en verbeterpunten van verschillende klanten op onze applicatie beschikbaar was, moesten we ervoor zorgen dat enkel de juiste personen die data konden zien. Wanneer een buitenstaander de verbeterpunten van een klant te zien krijgt is dit zeer gevaarlijk aangezien dit de zwakke plekken van de klant zijn.

3 AANPAK EN VERLOOP

In dit onderdeel bespreek ik kort hoe de stage is verlopen, van de voorbereiding voor het projectwerk, tot de communicatie tijdens het project. Verdere details over de ontwikkeling van de specifieke onderdelen van het project komen verder in dit document aan bod.

3.1 Planning

Van school uit waren we verplicht een planning te maken om op deze manier al eens goed na te denken over het verloop van de stage. Dit zorgde ervoor dat we ook al rekening moesten houden met eventuele veranderingen of problemen die ontstonden.










Zoals je kan zien op onderstaande afbeeldingen was de verwachte planning niet echt haalbaar. Dit kwam namelijk doordat ik had verwacht een analyse te moeten maken zoals we tijdens de lessen moesten doen, maar dit was niet het geval. De analyse was veel meer in detail dan ik had verwacht, dit maakte het enorm leerrijk, maar dus ook een langer proces. In het onderdeel over de analyse ga ik hier in meer detail op in.

3.1.1 Verwachte planning

Opdracht	Week 1 - 3	Week 4 - 6	Week 7 - 10	Week 11 - 13
Cursus TDD + SOLID				
SOLID + Typescript oef				
Plan van aanpak + analyse				
Einde analyse + setup project				
Log in system + coderen met default data				
Coderen met echte data				
Testing				
Final checks				

Figuur 2: Verwachte planning

3.1.2 Werkelijke planning

Opdracht	Week 1 - 3	Week 4 - 6	Week 7 - 10	Week 11 - 13
Cursus TDD + SOLID				
SOLID + Typescript oef				
Plan van aanpak + analyse				
Einde analyse + setup project				
Coderen met default data				
Log in systeem				
Coderen met data op basis van gebruiker				
Testing				
Final checks				

Figuur 3: Werkelijke planning

3.2 Communicatie

Voor het maken van een goed project was duidelijke en consistente communicatie enorm belangrijk. Dit zowel onderling met collega's, maar ook met de leerkrachten van school. Dit zorgde ervoor dat er zo min mogelijk misverstanden ontstonden en alles zo vlot mogelijk verliep. Ook zorgde dit ervoor dat ze op elk moment wisten waar ik mee bezig was en wat er op de planning stond.

3.2.1 Meetings

Ook vonden er wekelijkse stand-up meetings plaats met Jef Meijvis en Ive Verstappen. Zij waren de projectleiders van het DNL InfoSec Portal. Tijdens deze meetings moest ik even kort aanhalen wat ik de voorgaande week had gedaan, en wat er op de planning stond voor de daaropvolgende week. Jef en Ive beoordeelden op hun beurt dan weer mijn geleverd werk. Deze aspecten noteerde ik dan weer in een document om ze vervolgens aan te passen.

Anderzijds moest ik ook een wekelijkse update sturen naar mijn stagebegeleider Michaël Cloots en mijn contactpersoon binnen dotNET lab, Bjorn Ruysen. Zo kon Michaël meevolgen met de stand van zaken van het project en bekijken of alles nog vlot verliep. Ook Bjorn vond het fijn om deze informatie te krijgen zodat ook hij een beeld had bij de vooruitgang en stand van zaken van het project.

3.2.2 Vragen

Indien er bij mij problemen ontstonden tijdens het programmeren kon ik altijd hulp vragen bij collega's. Wanneer ik aanwezig was op kantoor kon ik dit altijd rechtstreeks aan hun vragen. De dagen dat er van thuis werd gewerkt gebeurde dit via Teams berichten voor kleine vragen en via een videogesprek voor de grotere en complexere vragen. Dit omdat dit een stuk makkelijker was dan schermafbeeldingen doorsturen etc.

Voor vragen over het echte project kon ik altijd terecht bij Jef of Ive. De meeste vragen werden gesteld tijdens de wekelijkse stand-up meeting, maar wanneer het toch een dringende vraag was kon ik hun altijd een Teams bericht sturen.

3.3 Plan van aanpak

Bij de voorbereiding voor een project hoort natuurlijk ook een plan van aanpak. Dit moest ik in de derde week van de stage voorstellen aan Bjorn, Ive en Michaël die het kantoor van dotNET lab kwam bezoeken. Door deze presentatie kreeg niet alleen ik, maar ook Michaël een beter beeld over mijn stage.

In deze presentatie legde ik kort uit wie dotNET lab was en waar ze zich mee bezig houden. Vervolgens ging het over wat het project juist inhield en waarom dit project nodig was, welke waarde het kon toevoegen aan, in dit geval, de klant. Ook werd hier de eerder aangehaalde planning in voorgesteld om een beeld te geven over hoe ik het stagetraject voor me zag. Als laatste werd er kort iets vermeld over de communicatie tussen mij en dotNET lab en tussen mij en Thomas More.



Figuur 4: Intro Plan van aanpak

4 CURSUSSEN

Voor de start van het projectwerk moest ik nog wat online cursussen volgen. Dit was simpelweg om alle benodigde informatie nog eens op te frissen zodat het projectwerk zo vlot mogelijk kon verlopen. Deze cursussen waren zowel "interne", zelfgemaakte cursussen van onder andere Tom Peeters en Vincent Boeijen, die beschikbaar waren op het dotNET Academy leerplatform. Ook waren er een aantal andere online cursussen die ik in dit geval moest bekijken op Pluralsight.

4.1 Test driven development

De eerste cursus die ik had gevolgd was, zoals de titel al zegt, over test driven development (Huber, 2024). Dit was een korte cursus die beschikbaar was op Pluralsight. Het begon met wat uitleg over unit testen in het algemeen waarbij ik tijdens de video's oefeningen mee moest maken. Vervolgens gingen de video's steeds meer in detail over het schrijven van testbare code, het toepassen van test driven development en ook even kort over het implementeren van de testen in de CI pipeline.

4.2 SOLID fundamentals

Hierna kwam de cursus over de SOLID principes (Peeters, z.d.) die gemaakt was door Tom Peeters en beschikbaar was op het dotNET Academy leerplatform. Deze cursus was een stuk uitgebreider aangezien dit een enorm belangrijk onderdeel was van mijn stage. Door deze principes toe te passen kon ervoor zorgen dat de code op de juiste manier werd opgebouwd om aan zoveel mogelijk SOLID principes te voldoen.

Deze SOLID principes zullen niet door iedereen gekend zijn, vandaar deze hele korte uitleg. SOLID is een acroniem voor verschillende principes, namelijk SRP (Single responsibility principle), dit wil zeggen dat elke klasse één specifieke verantwoordelijkheid moet hebben. OCP (Open closed principle) zegt dat een klasse moet open staan voor uitbreiding, maar gesloten voor wijzigingen. LSP (Liskov substitution principle) wil dan weer zeggen dat een subtype vervangbaar moet zijn door zijn supertype. ISP (Interface segregation principle) vertelt dat klanten niet afhankelijk mogen zijn van een interface en je dus beter meerdere kleine interfaces maakt dan één grote interface. Als laatste is er DIP (Dependency inversion principle) wat wil zeggen dat een high level module niet afhankelijk mag zijn van een low level module.

Na het volgen van de SOLID cursus was er een grote Game of Goose oefening waar ik aan zoveel mogelijk SOLID principes moest voldoen. Hierbij moest ik een console ganzenbord maken vanaf nul door gebruik te maken van onder andere interfaces en abstracte klassen. Deze oefening is ook een aantal keer gecontroleerd door Tom om na te kijken dat ik de SOLID principes wel echt door had. Wanneer ze zagen dat ik dit onder de knie had kon ik doorgaan naar de volgende cursus.


```

C:\Users\michi\Downloads\Stage exercise project\masterclass-base-main\solid-fundamentals\GameOfGoose\ConsoleApp\bin\Debug...
Welcome to my Game of Goose!
With how many do you want to play? (2 - 4)
3
    Player1      Player2      Player3
Turn 1
  5+6: S11      2+4: S6->S12    2+1: S3
[Press ENTER to play next turn 2]

Turn 2
  3+5: S19      4+2: S18->S24    4+4: S11
[Press ENTER to play next turn 3]

Turn 3
  / : S19      2+4: S30      3+2: S16
[Press ENTER to play next turn 4]

Turn 4
  1+3: S23      5+2: S37      3+1: S20
[Press ENTER to play next turn 5]

Turn 5
  2+3: S28      6+3: S46      5+4: S29
[Press ENTER to play next turn 6]

Turn 6
  1+4: S33      1+2: S49      1+5: S35
[Press ENTER to play next turn 7]

```

Figuur 5: Game of goose

4.3 TypeScript fundamentals

Om mijn TypeScript kennis nog wat op te frissen waren er een aantal oefeningen voorzien. Ditmaal zonder een online cursus, moest ik een tiental kleine oefeningen op een pure manier maken. Er werd een start bestand voorzien waar ik op mijn beurt interfaces moest aanmaken om de voorziene data te kunnen gebruiken. Waarna ik met zo min mogelijk if-else en for loops de oefeningen moest maken.

4.4 CI pipeline

Bij dit onderdeel van de cursussen moest ik, zoals je al verwacht, een CI pipeline maken. Dit heb ik gedaan via GitHub actions om voor zowel de Game of Goose oefening, als de typescript oefeningen de gemaakte testen te laten lopen in de pipeline. Dit natuurlijk met de bedoeling om in het echte project een pipeline te maken om op deze manier altijd automatisch testen te laten lopen.

4.5 Clean architecture

De laatste cursus waar ik aan was gestart was die over clean architecture (Peeters & Boeijen, z.d.), deze cursus was wel wat ouder, maar het principe over clean architecture bleef hetzelfde. Dit was nodig omdat ik bij het opzetten van de back-end gebruik moest maken van deze architectuur. Ik had uiteindelijk niet heel de cursus afgemaakt omdat ik moest starten aan de analyse van het project. Uiteraard wel enkel wanneer ik een goed gevoel had over mijn kennis in verband met de clean architecture.

5 ANALYSE

Voor ik echt kon starten met het programmeren van het project heb ik eerst een zeer gedetailleerde analyse gemaakt. In deze analyse ben ik over elk aspect van het project gegaan om op deze manier goed voorbereid te zijn. Deze analyse heeft een zestal weken geduurd en is gedurende deze tijd van een simpele oppervlakkige analyse geëvolueerd naar een gedetailleerde en uitgebreide analyse. Uiteraard met hulp en feedback van Jef en Ive. Zij hebben tijdens de wekelijkse meetings verschillende aanpassingen en uitbreidingen doorgegeven aan mij, die ik op mijn beurt op de juiste manier implementeerde.

5.1 Waarom

Een analyse is altijd belangrijk voor het starten aan een project, maar voor dit project was het nog belangrijker. Dit komt omdat er na mijn stage nog aan dit project gaat verder gewerkt worden. Voor deze reden moest de analyse enorm gedetailleerd zijn zodat de werknemer die in de toekomst aan dit project gaat werken ook duidelijk weet wat het doel en de visie van het project was.

Ook vond Ive deze analyse super belangrijk omdat dit ervoor zorgt dat het programmeren van de echte applicatie een heel stuk vlotter gaat. Dit komt omdat er over elk aspect van de applicatie al is nagedacht. Zo moest ik bijvoorbeeld tijdens het programmeren niet meer zoeken naar welke libraries ik moest gebruiken. Dit was allemaal al onderzocht in de analyse waardoor ik het een stuk eenvoudiger kon implementeren.

5.2 Hoe

Om de analyse te maken ben ik gestart met een hoofding te maken voor elk component van de applicatie. Dit gaat niet alleen over functionele componenten zoals een beheer pagina, maar ook over bijvoorbeeld de hosting en beveiliging van de applicatie.

Vervolgens heb ik voor elke component de analyse nog in drie andere onderverdelingen verdeeld, namelijk *Waarom*, *Wat* en *Hoe*. Het *waarom* en *wat* gedeelte was vrij vanzelfsprekend. Het *waarom* gedeelte bestond enkel uit een uitleg waarom deze component nodig en belangrijk was voor het uitwerken van een goed project. Het echte pad dat de gebruiker zou nemen of kunnen nemen werd dan weer uitgelegd in het *wat* onderdeel. Dit werd gedaan aan de hand van prototypes die ik doorheen de weg had gemaakt.

Het *hoe* gedeelte bleek na verschillende versies een stuk moeilijker en gedetailleerder te moeten zijn. In dit *hoe* gedeelte werd namelijk de echte implementatie van deze component uitgelegd. Dit wil zeggen dat ik voor bijvoorbeeld het login systeem stap voor stap heb uitgelegd hoe dit geïmplementeerd moest worden. Ook werd in dit onderdeel besproken welke bronnen en welke database tabellen we nodig hadden. Hierna volgde nog een onderdeel over beveiligingsmaatregelen en de audit trail. Deze audit trail zou elke actie die een gebruiker deed opslaan in de database. Vooral de gebruiker, de actie en de datum en tijd waren het belangrijkste voor de audit trail, maar dit kon zo uitgebreid zijn als we wilden. Dit om ervoor te zorgen dat de webapplicatie zo veilig mogelijk zou zijn en we op elk moment weten wat er in de applicatie gebeurde. Als laatste was er nog een onderdeel over waarom en hoe we deze component gingen testen.

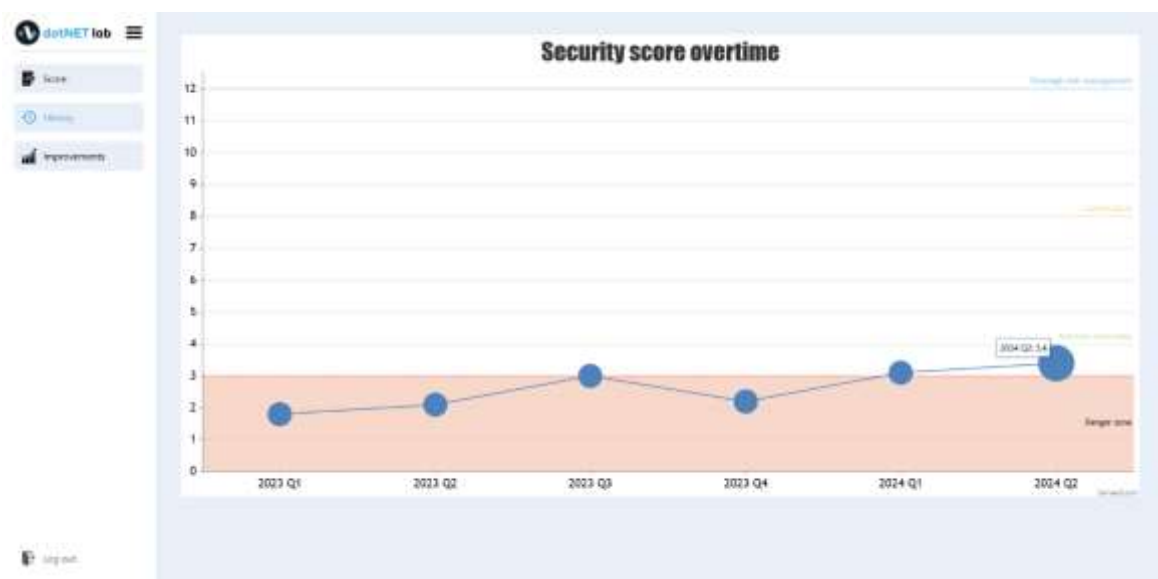
5.3 Prototypes

Bij elke analyse horen natuurlijk ook prototypes om een beeld te vormen van het project. Dit heb ik dus ook gedaan door te starten met wat oppervlakkige en simpele (low fidelity) prototypes om te kijken op welke manier ze de applicatie wilden stijlen. Dit werd tijdens de wekelijkse stand-up meetings besproken om op deze manier elke week iets gedetailleerder op de prototypes in te gaan.

Door de weken heen, na meerdere aanpassingen zijn er dan toch prototypes uitgekomen voor elke pagina. Zo kan je hier het prototype zien van de *History* pagina, daaronder kan je dan de echte uitgewerkte pagina zien. Dat maakt extra duidelijk hoe belangrijk en handig prototypes zijn, ze zijn als het ware de plattegrond van het uiteindelijke resultaat. Natuurlijk mag dit resultaat afwijken van het prototype, het is gewoon een goede basis.



Figuur 6: Prototype History pagina



Figuur 7: Uitgewerkte History pagina

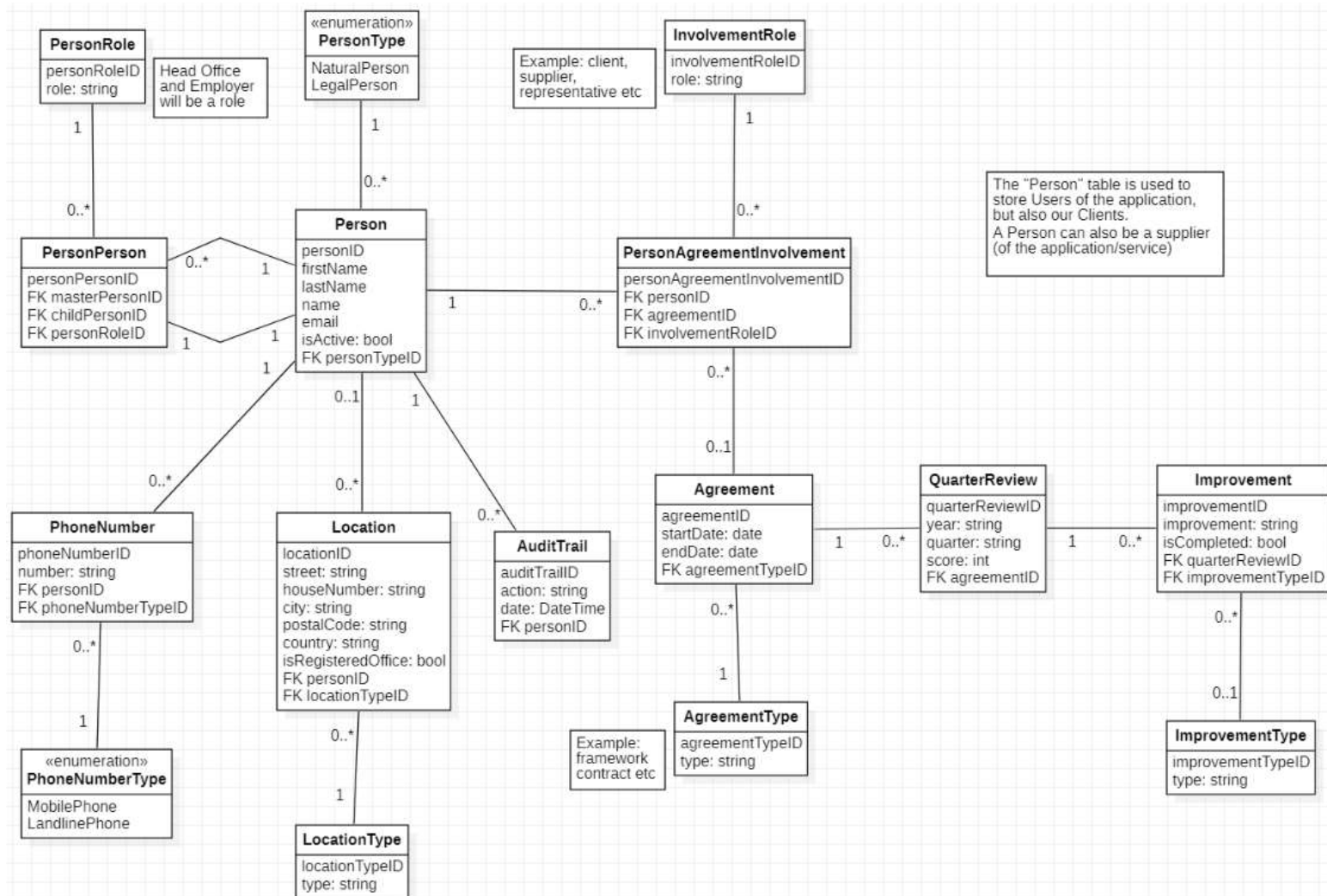
5.4 Datamodel

Ook het datamodel was een enorm belangrijk onderdeel van de analyse, het maken van een correct datamodel was echter wel een enorme opgave. Het datamodel was gestart als een simpel model met een vijftal tabellen wat na heel wat aanpassingen en overleggen met een collega, Derre Evers, is getransformeerd naar een volwaardig datamodel voor een "typisch administratief systeem".

Dit komt vooral door de uitleg die Ive gaf tijdens de infosessie over datamodelleren. Dit soort infosessie/opleiding gebeurt vaker bij dotNET lab om op deze manier toch op de hoogte te blijven van bijvoorbeeld nieuwe technologieën.

Zoals je op onderstaande afbeelding kan zien is het een enorm groot model geworden wat niet meer lijkt op iets wat we tijdens de opleiding zagen. Dit komt omdat dit een model is zonder context, dat is een zeer belangrijk aspect van een datamodel voor een typisch administratief systeem. Dit wil zeggen dat het voor verschillende doeleinden gebruikt kan worden, zo zie je dat we geen *User* en *Client* tabel hebben, maar een *Person* tabel waar ze beide inpassen.

Ook zie je een *Agreement* tabel die het mogelijk maakte om contracten op te slagen in dit systeem. Gelinkt aan deze tabel zie je dan weer de *PersonAgreementInvolvement* tabel die het mogelijk maakt om bij te houden welke rol elke persoon heeft binnen een specifiek contract.



Figuur 8: Datamodel

6 CODEREN

In dit onderdeel leg ik uit welke technologieën ik heb gebruikt voor het project en wat het uiteindelijke eindresultaat is geworden. Oorspronkelijk was ik gestart met eerst de volledige back-end te maken, maar Ive wees me erop dat hij het beter vindt om eerst de front-end te maken. Toen ik het eenmaal op deze manier deed ontdekte ik dat het inderdaad een stuk makkelijker was, aangezien ik veel beter wist wat ik nodig had van de back-end.

De applicatie is niet helemaal afgeraakt, maar dit was ook geen vereiste. Het was belangrijker dat de applicatie goed werd opgezet zodat het altijd uitgebreid kon worden. Het doel wat ik voor mezelf had gezet was het kunnen ingeven van een score en verbeterpunten, die op hun beurt dan weer beschikbaar waren voor de klant. Dit was gelukt waardoor niet alleen ik, maar ook Jef en Ive tevreden waren over het resultaat.

6.1 Back-end

Voor het maken van de back-end hebben we gekozen voor C# en gebruik te maken van het .NET 8.0 framework met een ASP .NET Core Web API om op deze manier de data naar de front-end te sturen.

Een enorm belangrijke vereiste van de back-end was de clean architecture. Dit zorgt ervoor dat elk onderdeel van de back-end in zijn eigen project zit om zo een goed overzicht te behouden over alle bestanden. Zo was het back-end project opgesplitst in nog vier andere projecten, namelijk API, Application, Infrastructure en Domain.

6.1.1 API

In het API project bevonden zich alle controllers die nodig waren voor de data op te halen vanuit de front-end. Deze controllers wisten eigenlijk niets over wat of hoe dit gebeurde, die logica zat op een andere locatie. Het API project was enkel het aanspreekpunt van de front-end om de data op te halen.

6.1.2 Application

Deze controller gaat vervolgens naar het Application gedeelte van de back-end. Dit onderdeel was op zijn beurt dan weer verantwoordelijk voor het terugsturen van de data naar de controller. Ter verduidelijking, dit onderdeel was niet verantwoordelijk voor het ophalen van de data, enkel voor het in de juiste structuur terug te sturen naar de controller.

6.1.3 Infrastructure

Het onderdeel dat de data echt ging ophalen was het Infrastructure onderdeel. Hier bevonden de repositories zich om de data door middel van Entity Framework Core op te halen uit de database. Ook bevonden zich hier de configuratie bestanden om de linken te leggen tussen de verschillende tabellen/modellen.

6.1.4 Domain

Het laatste onderdeel was het Domain gedeelte waar al de modellen en repository interfaces zich in bevonden. Dit was als het ware de basis van het project aangezien hier werd besloten welke datavelden er in elk model moesten komen. Ook de interfaces bepaalden wat er allemaal moest aangemaakt worden in de repository.

6.2 Front-end

De front-end werd dan weer gemaakt met het Angular 18 framework. Dit was dan ook gekozen door Ive, wat voor mij enorm positief is aangezien ik dit tijdens de opleiding enorm leuk vond en er ook al het meeste ervaring mee had. Hiervoor waren er TypeScript oefeningen voorzien voor mij zodat ik terug wat gewend raakte aan deze taal.

6.2.1 Styling

Voor de styling heb ik samen met wat sturing van een collega, Derre Evers, gekozen om Tailwind te gebruiken. Hier was ik persoonlijk ook het meest ervaren in aangezien dit bij voorgaande projecten ook gebruikt werd. Verder werd er gevraagd dat de applicatie ongeveer dezelfde stijl kreeg als de interne applicatie van dotNET lab. Dit wou vooral zeggen dat de navigatiebalk aan de zijkant van het scherm moest komen en dat deze inklapbaar zou zijn. Verder waren er ook wat algemene kleuren en locaties van bepaalde componenten gekozen.

6.2.2 Componenten

Voor het maken van een makkelijk beheerbare applicatie was het belangrijk om te werken met componenten. Vandaar dat ik voor zoveel mogelijk onderdelen van de applicatie een component had gemaakt. Zo had ik bijvoorbeeld een aparte component aangemaakt voor de navigatiebalk, die in elke pagina werd geïmplementeerd.

Ook van het modal had ik een component gemaakt, aangezien dit op meerdere pagina's gebruikt werd. Dit maakte het enorm makkelijk om de algemene stijl van het modal aan te passen. Het verwijdermodal was ook een component op zich, die op zijn beurt ook gebruik maakte van het normale modal. Hierdoor moesten we geen code kopiëren en moesten we maar één bestand wijzigen om de aanpassing door te voeren op elke pagina.

6.2.3 Libraries

In plaats van bijvoorbeeld een grafiek volledig vanaf nul te maken, wat enorm veel tijd kost, zijn er betere opties online te vinden in de vorm van libraries. Het kiezen van een library was niet zomaar de eerste kiezen die ik tegenkwam. Ik heb een weighted decision matrix gemaakt om op deze manier de juiste keuze te maken. Hierbij moest ik onder andere rekening houden met de prijs, maar ook de mogelijkheden van beide libraries.

Om de tabel te maken hadden we ervoor gekozen om dit te doen met PrimeFaces. Dit omdat het een gratis library was, die er al zeer goed uitzag en aan bijna alle eisen voldeed. Voor de tabel waren namelijk de prijs en de filter- en zoekmogelijkheden het belangrijkste. Vandaar dat we voor PrimeFaces hadden gekozen. Dit is een online open source componenten library waar je allerlei verschillende componenten kunt vinden. Bij dit project had ik enkel de tabel nodig, maar er zijn nog veel meer mogelijkheden met deze library.

Dat was niet het enige bestaande component dat ik heb gebruikt, ik heb namelijk ook gebruik gemaakt van CanvasJS. Dit is ook een online library, maar dan enkel voor grafieken te maken. Deze keuze was een stuk makkelijker aangezien er voor mijn gevoel minder mogelijkheden waren en ik wist dat de grafiek er kon uitzien zoals het hoort. Dit komt doordat ik al eens kort hiermee had gewerkt en wist wat de mogelijkheden hiervan waren.

6.3 Authenticatie en autorisatie

Misschien wel het belangrijkste onderdeel van de applicatie was het authenticeren en het autoriseren van de gebruiker. Zeker in deze security-gerichte applicatie. Dit hebben we gedaan door middel van Microsoft Entra ID (Microsoft, z.d.). Dit is een "cloud-based identity and access management service", wat wil zeggen dat je met deze service kunt configureren wie toegang heeft tot de applicatie en tot welk onderdeel.

Voorlopig konden enkel gebruikers van dotNET lab inloggen op de applicatie. Dit komt omdat ik dit samen met een collega, Rikko Keuppens, ging implementeren. Hij gaf aan dat het te geavanceerd was om dit op zo een korte tijd volledig uit te werken. Hij had het inloggen voor meerdere interne applicaties al opgezet en had dus een goed beeld op wat mogelijk was en wat niet.

Als oplossing hadden we het daarom wat vereenvoudigd, natuurlijk met de toestemming van Jef en Ive. De basis voor het inloggen was vooral gemaakt door Rikko. Dit heeft hij gedaan door een nieuwe applicatie toe te voegen aan de al bestaande tenant van dotNET lab. In de code werd dan weer de Angular Auth OIDC Client library geïnstalleerd. Deze library zorgde ervoor dat de applicatie kon communiceren met Microsoft Entra ID zodat we op deze de ingelogde gebruiker konden ophalen.

Met deze informatie van de gebruiker konden we dan bekijken welke rol deze persoon had en dus ook op welke pagina's hij of zij mocht komen. Voor klanten werd er daarnaast dan ook gekeken naar welk bedrijf er aan deze klant was gekoppeld zodat hij of zij enkel de data van dat specifieke bedrijf kon zien.

6.4 CI/CD Pipeline

Een volgend belangrijk onderdeel van een applicatie is een Continuous Integration en Continuous Deployment (CI/CD) pipeline. Deze pipeline automatiseert het testen en deployen van code. Dit zorgt ervoor dat elke keer wanneer er code naar de Azure DevOps omgeving wordt gepusht, gespecificeerde testen zullen worden uitgevoerd. Na het lopen van de pipeline krijg je te zien of deze testen zijn geslaagd of niet. Op deze manier zorg je ervoor dat de code altijd op de juiste manier blijft werken.

Helaas had ik voor mijn project nog geen CI/CD pipeline gemaakt. Doordat ik vrij laat ben gestart met het programmeren wou ik er eerst voor zorgen dat er een werkende en bruikbare applicatie was. De volgende stap was dan het maken van deze pipeline, waar ik dus niet toe ben gekomen.

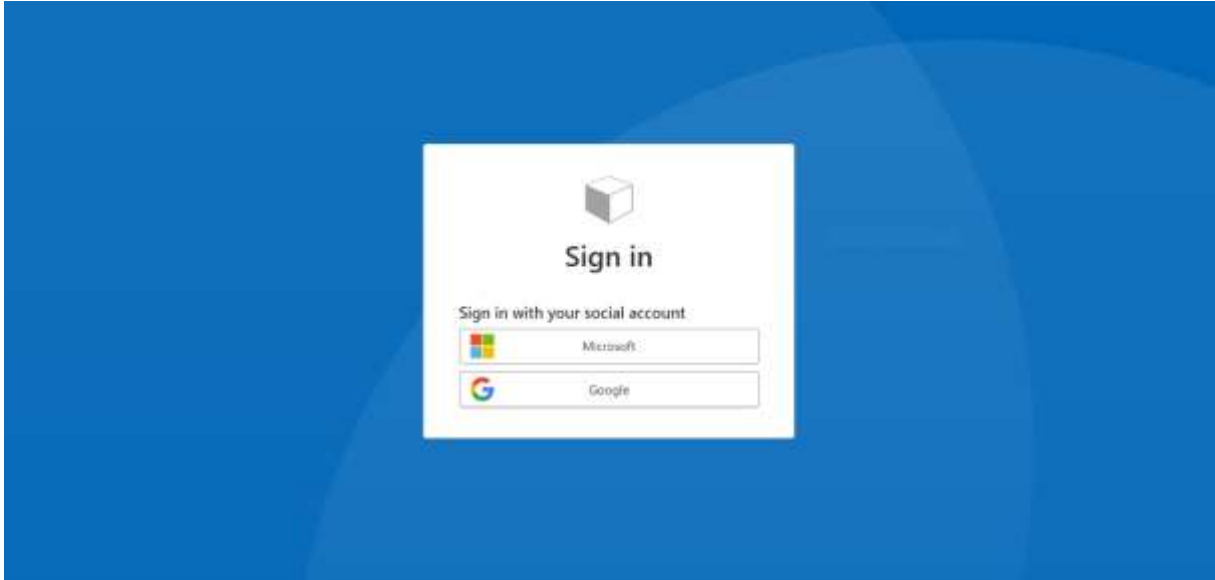
6.5 Hosten

Jammer genoeg is ook het hosten van dit project niet gelukt. Dit kwam simpelweg omdat er niet genoeg tijd was en ik niet de juiste rechten had om dit te kunnen doen. Het was ook geen vereiste vanuit dotNET lab dat dit al gehost moest worden. Dit maakte dat de focus vooral lag op de applicatie zelf. Het had het natuurlijk extra mooi gemaakt, maar zowel zij als ik waren tevreden met het resultaat dat is behaald.

Het was de bedoeling zowel de front- en back-end, maar ook de database te hosten op Azure. Dit was een vereiste vanuit dotNET lab aangezien hier alle applicaties van hun op worden gehost. Ze zijn ook Microsoft partner, wat het logisch maakt dat ze services van Microsoft gebruiken, Azure in dit geval.

6.6 Resultaat

Na 13 weken goed mijn best te doen heb ik, naar mijn mening, toch een mooie applicatie weten te maken. Dit zal ik in dit onderdeel duidelijk maken door wat pagina's voor zowel gebruikers als administrators te laten zien. Het eerste wat natuurlijk te zien is op de applicatie is het scherm om in te loggen, wanneer een persoon niet is ingelogd krijgt hij of zij niets te zien.



Figuur 9: Login pagina

6.6.1 Gebruiker

Wanneer een gewone gebruiker is ingelogd komt hij eerst op de home pagina terecht. Het was oorspronkelijk de bedoeling dat de gebruiker rechtstreeks op de score pagina terecht kwam, maar dan was dit ook het geval voor een administrator, wat niet de bedoeling was. Ook met hulp van een collega kreeg ik dit niet opgelost waardoor ik ervoor heb gekozen een simpele homepagina te maken.



Figuur 10: Gebruiker home pagina

Onderstaande pagina is de score pagina. Op deze pagina krijgt de gebruiker de vijf meest recente en de huidige security score te zien. Bij het hoveren van de muis over de databollen kan je duidelijk zien welke score je op dat moment hebt. Zo zie je ook dat er onderverdelingen zijn gemaakt in de grafiek. Dit laat zien wat de security score eigenlijk betekent, want vanaf dat je een drie hebt zitten klanten eigenlijk al in de veilige zone. De scores die de klant hier te zien krijgt, worden ingegeven door een werknemer van dotNET lab.



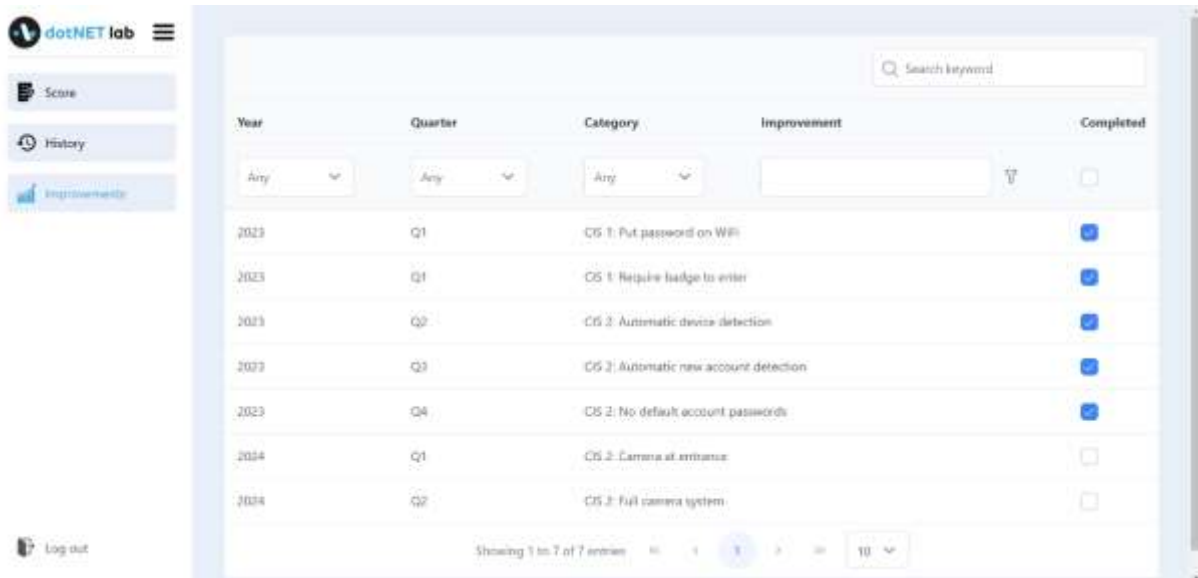
Figuur 11: Score pagina

Ook op de history pagina kan je de details van een specifieke score bekijken. Dit zijn al de scores van de klant. In dit geval zijn het er maar zes, maar in de toekomst kunnen dit er een stuk meer worden. Deze pagina geeft een mooi overzicht over de evolutie van de score doorheen de tijd.



Figuur 12: History pagina

De laatste, maar misschien wel de belangrijkste pagina voor de gebruiker is de improvements pagina. Op deze pagina zullen al de verbeterpunten komen die de klant moet implementeren. Deze worden ook door een dotNET lab werknemer ingegeven. De klant heeft op deze pagina ook de mogelijkheid om te filteren op verschillende zaken. Ook kan hij of zij bepaalde verbeterpunten aanduiden als "completed". Dit wil zeggen dat ze zijn geïmplementeerd volgens de klant. Een dotNET lab werknemer zal het volgende kwartaal dan nakijken of dit verbeterpunt wel degelijk is geïmplementeerd en of het op de juiste manier is gedaan.

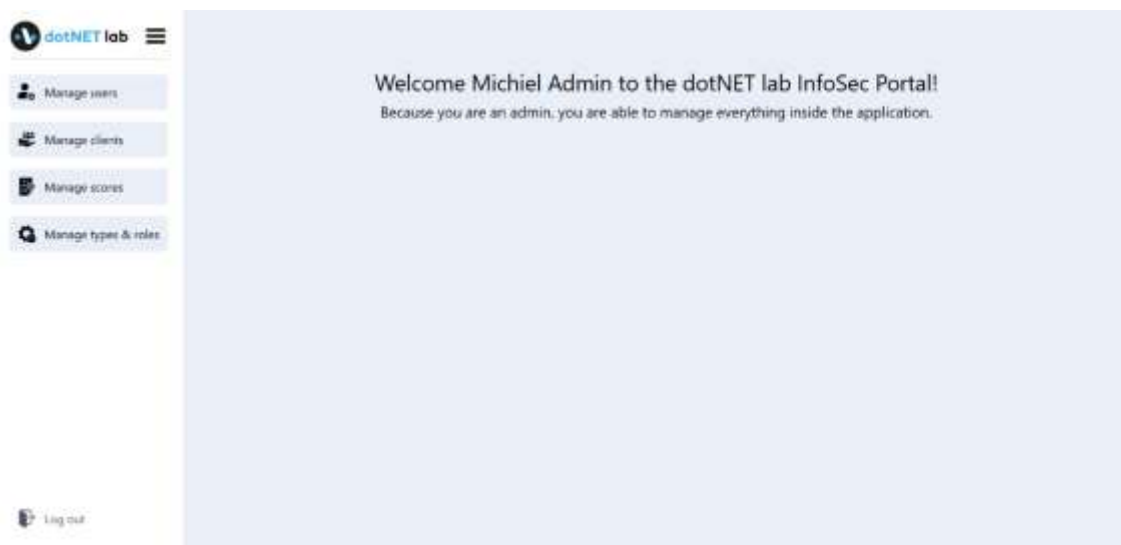


Year	Quarter	Category	Improvement	Completed
Any	Any	Any		<input type="checkbox"/>
2023	Q1	CS 1: Put password on WiFi		<input checked="" type="checkbox"/>
2023	Q1	CS 1: Require badge to enter		<input checked="" type="checkbox"/>
2023	Q2	CS 2: Automatic device detection		<input checked="" type="checkbox"/>
2023	Q3	CS 2: Automatic new account detection		<input checked="" type="checkbox"/>
2023	Q4	CS 2: No default account passwords		<input checked="" type="checkbox"/>
2024	Q1	CS 2: Camera at entrance		<input type="checkbox"/>
2024	Q2	CS 2: Full camera system		<input type="checkbox"/>

Figuur 13: Improvements pagina

6.6.2 Administrator

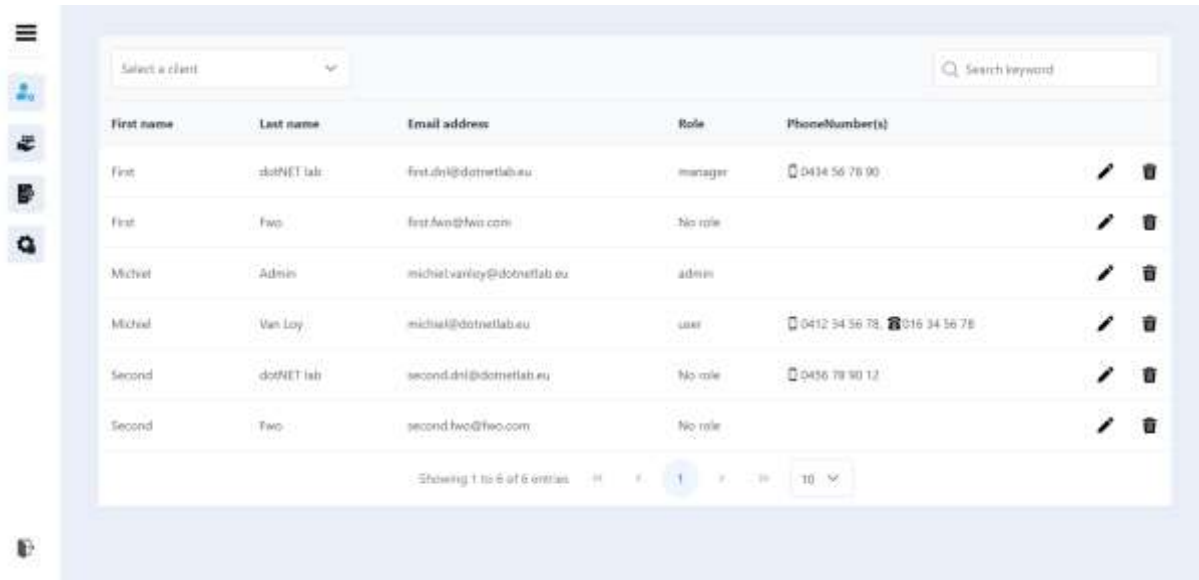
Ook de administrator komt eerst op een home pagina terecht. Dit komt doordat na het inloggen het inladen van de pagina gebeurt voor dat het systeem weet wat voor rol deze gebruiker heeft. Om deze reden was het de veiligste optie om naar de home pagina te navigeren. Vanuit deze pagina kon de administrator dan ook kiezen welke pagina hij wou bezoeken.



Figuur 14: Administrator home pagina

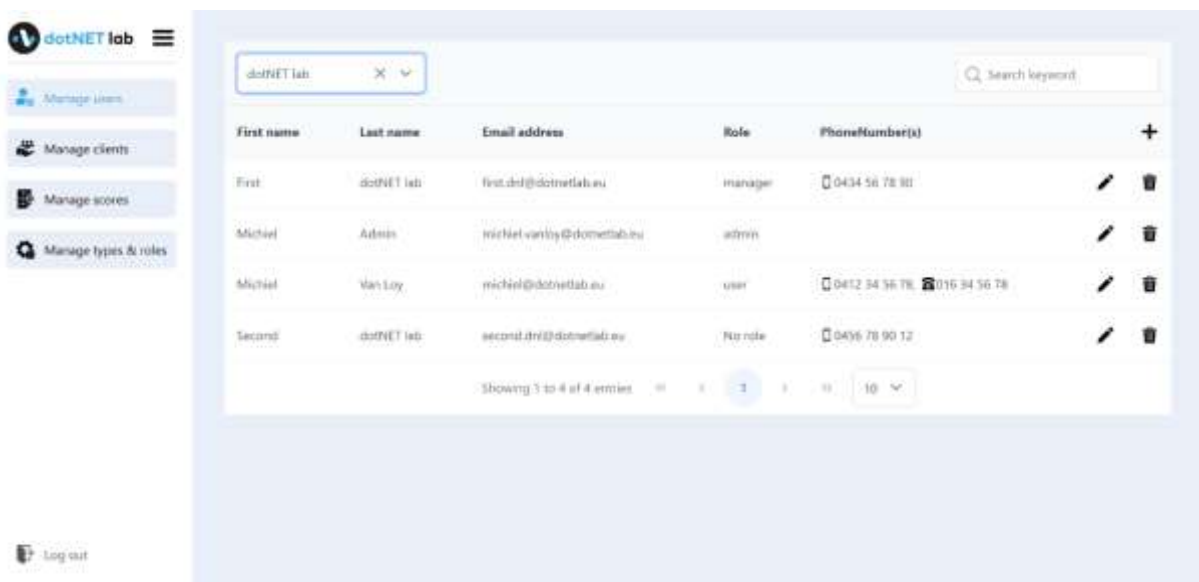
6.6.2.1 Manage users

Als eerst was er de manage users pagina waar de administrator de mogelijkheid kreeg om de gebruikers van de applicatie te beheren. Zo kan je zien dat de pagina is geopend met al de gebruikers, waarbij er de mogelijkheid was om te filteren op de gebruikers van een specifieke klant. Op onderstaande afbeelding zie je dat de navigatiebalk ook klein gemaakt kon worden. Zo konden mensen met een kleiner scherm ook op een gebruiksvriendelijke manier de pagina gebruiken.

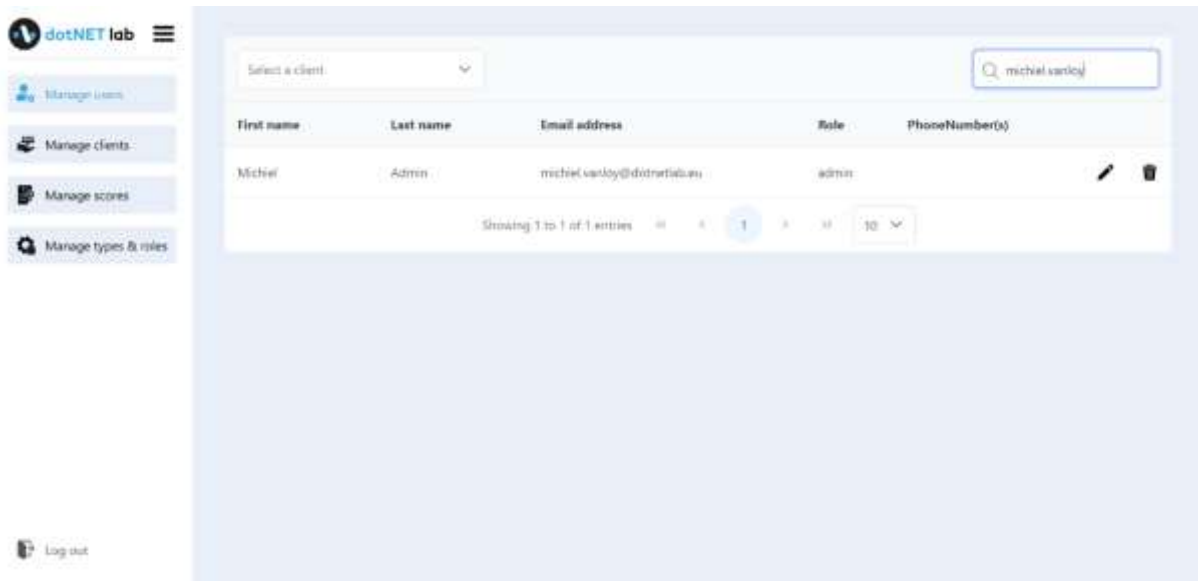


Figuur 15: Manage users pagina

Op onderstaande afbeeldingen zie je dat het filteren en zoeken de data op de juiste manier heeft gefilterd. Dit zorgde ervoor dat de administrator specifieke gebruikers op een makkelijke manier kon vinden om hun gegevens aan te passen.



Figuur 16: Manage users pagina - filter



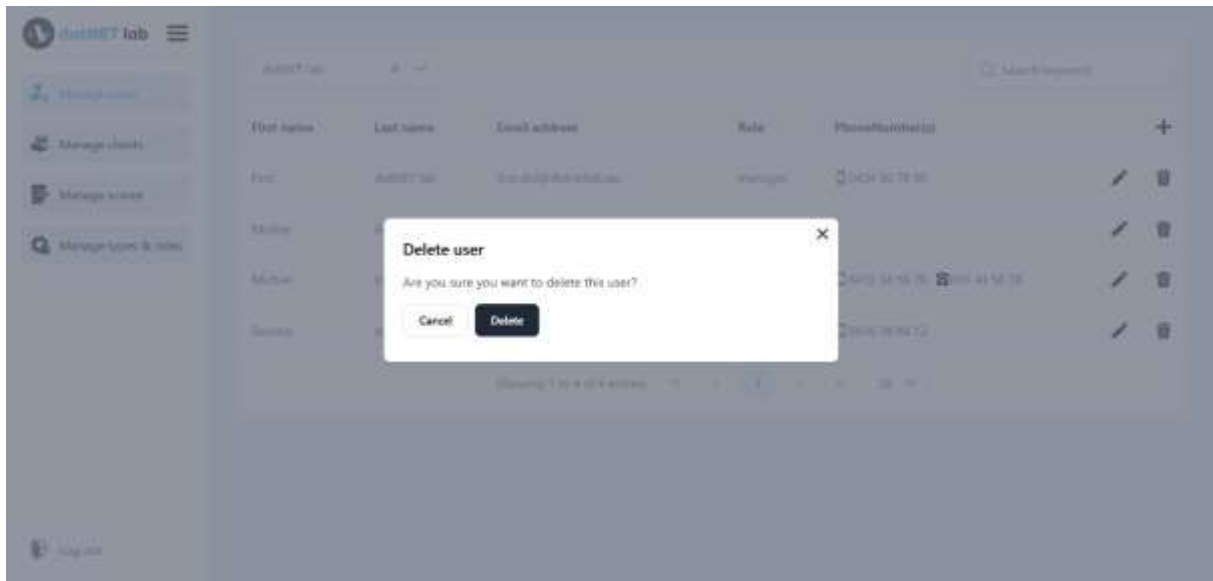
Figuur 17: Manage users pagina - zoek

Natuurlijk was ook het creëren, bewerken en verwijderen van de gebruikers onderdeel van deze pagina. Dit gebeurde allemaal in een algemeen modal dat opende na het klikken op de bijbehorende knop. Zo zie je op de afbeelding hieronder het modal dat opende wanneer er op de bewerk knop werd gedrukt. Dit was een algemene component die elke keer werd hergebruikt voor de verschillende pagina's.



Figuur 18: Manage users pagina - edit

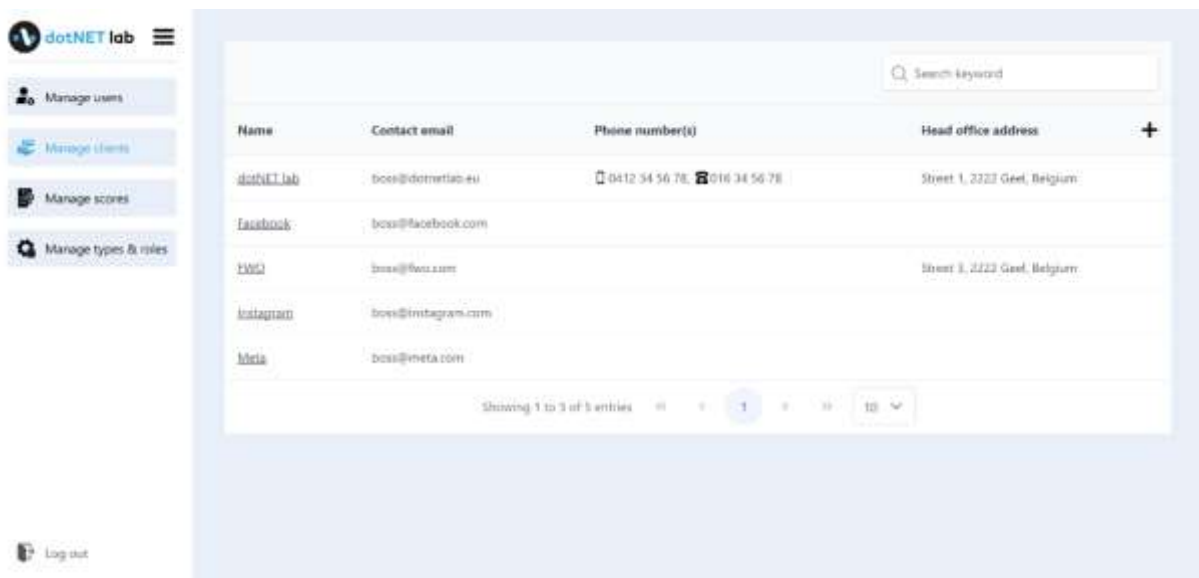
Ook de verwijderknop opende een modal, maar dan om bevestiging te vragen of je de gebruiker wel echt wou verwijderen. Dit verwijdermodal werd gebruikt op elke pagina en kan in de toekomst ook gebruikt worden om de administrator te laten kiezen of hij de gebruiker echt wil verwijderen of op niet-actief zetten.



Figuur 19: Manage users pagina - verwijder

6.6.2.2 Manage clients

De volgende pagina in de navigatiebalk was de manage clients pagina waar de administrator de mogelijkheid kreeg om de klanten te bewerken. Ook was er op deze pagina de mogelijkheid om specifieke klanten te zoeken, om op deze manier zo snel mogelijk de juiste klant te kunnen vinden.



Figuur 20: Manage clients pagina

Doordat een klant uit enorm veel data kon bestaan, namelijk meerdere telefoonnummers, adressen en dochterbedrijven, hadden we ervoor gekozen om de data te tonen op een detail scherm. Zo konden we ervoor zorgen dat al de data duidelijk zichtbaar bleef. Een andere reden dat we hadden gekozen voor een detail pagina was omdat scrollen in een modal niet erg gebruiksvriendelijk is. Doordat bijna alle data op de detailpagina te zien was hadden we er ook voor gekozen de bewerk- en verwijderknop onderaan de pagina te plaatsen.

Figuur 21: Manage clients pagina - details 1

Figuur 22: Manage clients pagina - details 2

Natuurlijk kozen we er dus voor om het detailscherm ook te gebruiken als creëer- en bewerksscherm. Hier kreeg de administrator de mogelijkheid om telefoonnummers, adressen en dochterbedrijven toe te voegen, te bewerken of te verwijderen. Het adres dat werd aangeduid als "head office", was het adres dat op de hoofdpagina te zien was.

dotNET lab [Menu Icon]

- Manage users
- Manage clients
- Manage scores
- Manage types & roles

Log out

Edit a client

Back

Name: dotNET lab Contact person email address: boss@dotnetlab.eu

Phone number(s):

0412 34 56 78 Mobile Phone

016 34 56 78 Landline Phone

Address(es):

Address 1:

Street: Street House number: 2

Postal code: 2222 City: Geel

Country: Belgium Location type: Office Head office: ☐

Address 2:

Street: Street House number: 1

Figuur 23: Manage clients pagina - edit 1

dotNET lab [Menu Icon]

- Manage users
- Manage clients
- Manage scores
- Manage types & roles

Log out

Address(es):

Address 1:

Street: Street House number: 1

Postal code: 2222 City: Geel

Country: Belgium Location type: Office Head office: ☒

Address 2:

Street: Street House number: 2

Postal code: 2222 City: Geel

Country: Belgium Location type: Office Head office: ☐

Daughter company(s):

Cancel Save

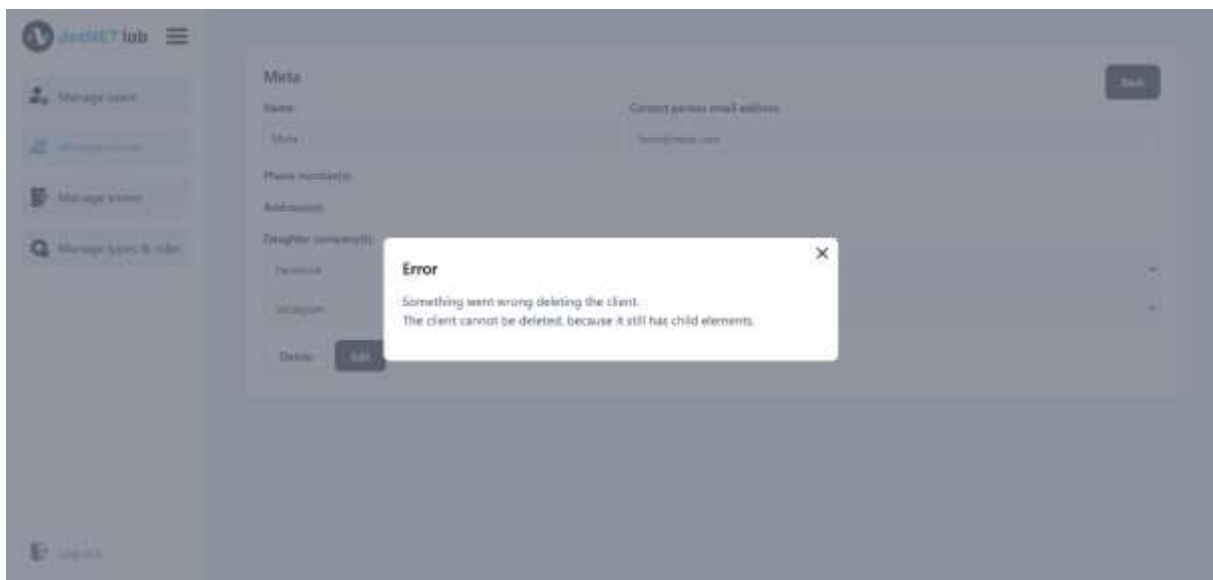
Figuur 24: Manage clients pagina - edit 2

Hieronder zie je dan weer een voorbeeld waarbij er wel dochterbedrijven zijn aangewezen, maar geen telefoonnummers of adressen. Zo zie je dat ook dit onderdeel van de pagina dezelfde stijl had.

The screenshot shows the 'dotNET lab' interface with a sidebar containing 'Manage users', 'Manage clients', 'Manage scores', and 'Manage types & roles'. The main content area displays the 'Meta' client edit form. The form includes fields for 'Name' (Meta), 'Contact person email address' (jane@meta.com), 'Phone number(s)', 'Address(es)', and 'Daughter company(s)' (Facebook and Instagram). There are 'Delete' and 'Edit' buttons at the bottom of the form.

Figuur 25: Manage clients pagina - edit 3

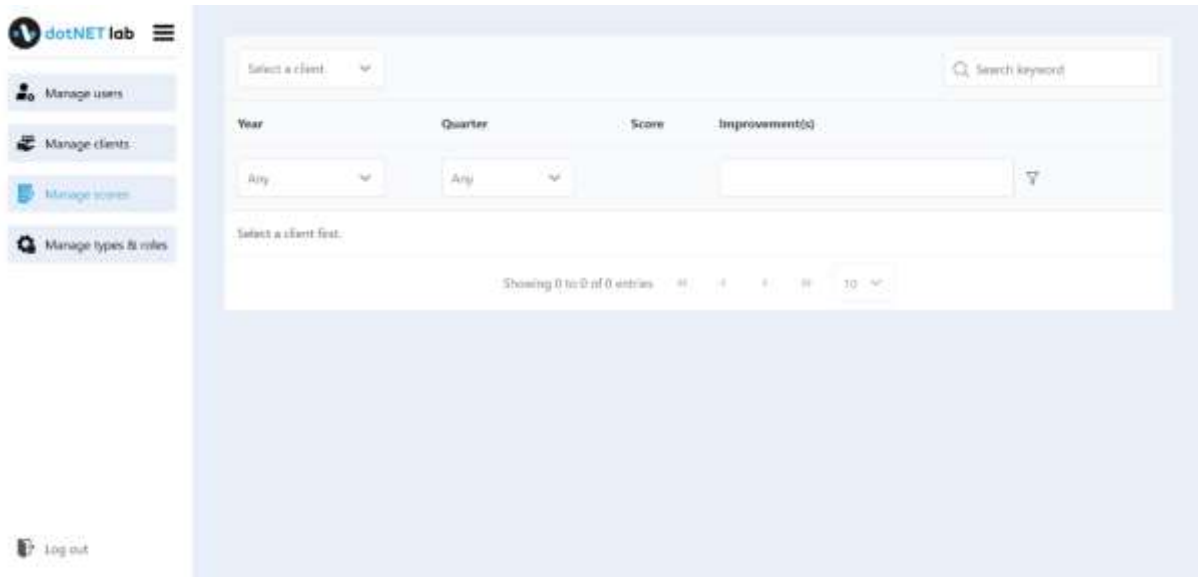
Ook op deze pagina is er gebruik gemaakt van het algemene verwijdermodal. Het enige verschil was dat het op deze pagina niet mogelijk was een klant te verwijderen zolang deze klant nog dochterbedrijven of gebruikers aan hem had aangewezen. Dit kan je zien op onderstaande afbeelding.



Figuur 26: Manage clients pagina - verwijderen niet toegestaan

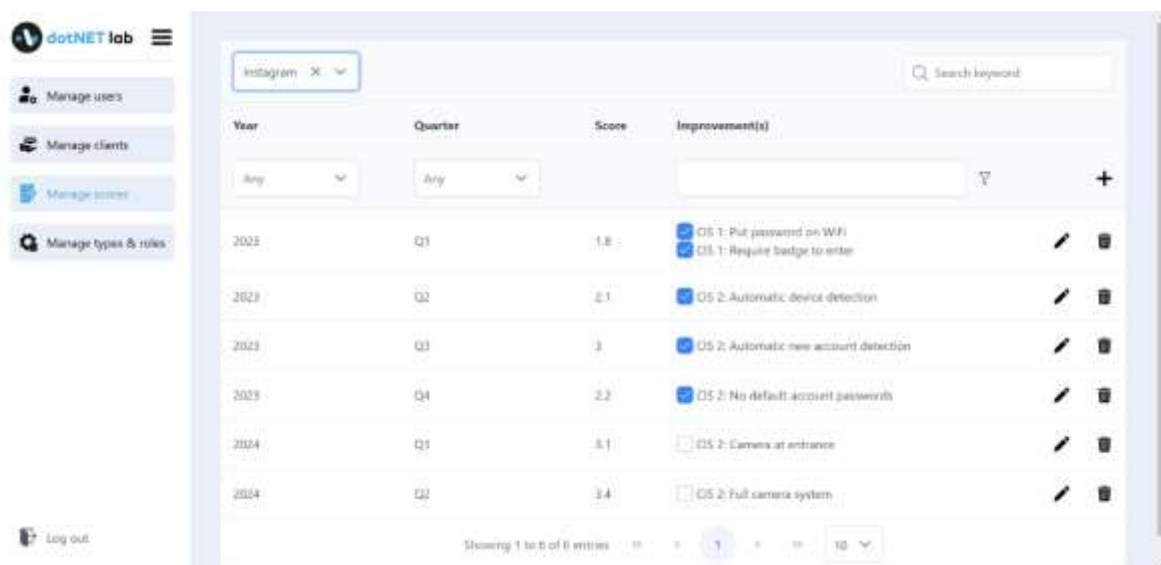
6.6.2.3 Manage scores

De manage scores pagina was misschien wel de belangrijkste administrator pagina van allemaal, doordat hier al de scores en verbeterpunten worden ingegeven. Ik had ervoor gekozen om de administrator eerst een klant te laten kiezen en hem dan pas de mogelijkheid te geven om een nieuwe score toe te voegen. Dit zorgde ervoor dat er in de toekomst altijd een externe persoon aangewezen kon worden om een kleine groep specifieke klanten te beheren. Deze externe persoon zou dus enkel kunnen kiezen uit deze groep en op deze manier onmogelijk de data van de andere klanten kunnen zien.



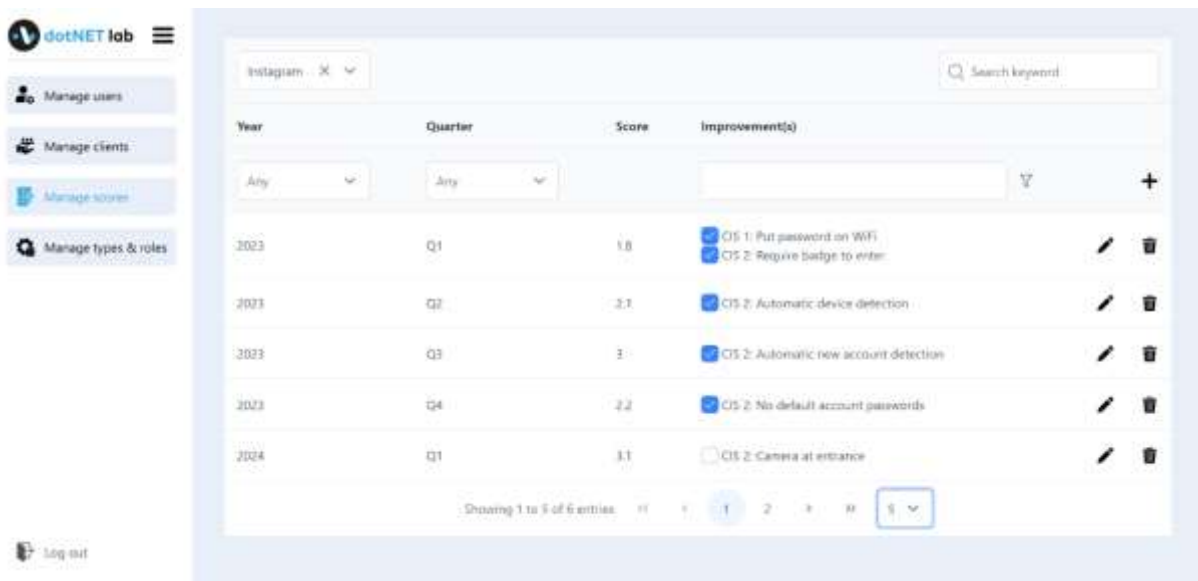
Figuur 27: Manage scores pagina

Wanneer er dan een specifieke klant werd gekozen kwam alle data tevoorschijn. Zoals je ziet was er ook hier de mogelijkheid om te filteren en te zoeken op verschillende zaken. Ik had er ook voor gekozen om het op deze pagina al mogelijk te maken een verbeterpunt aan te duiden als "geïmplementeerd". Dit zorgt ervoor dat de administrator snel en eenvoudig dit kan veranderen, zonder het bewerk scherm voor elke score te openen.



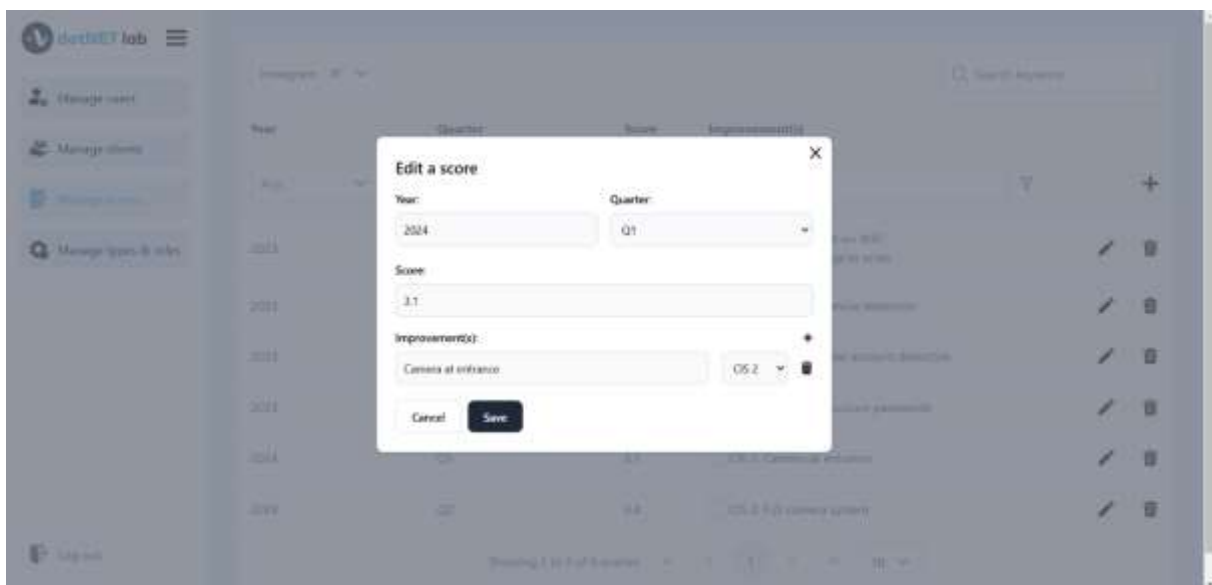
Figuur 28: Manage scores pagina - filter

Op onderstaande afbeelding kan je zien dat deze pagina, net zoals alle andere pagina's, de mogelijkheid gaf het aantal scores per pagina aan te passen. Ook dit zorgt ervoor dat de administrator zelf kon kiezen op welke manier hij de data wou bekijken. Dit gaf de mogelijkheid om ook administrators met een klein scherm een mooie interface te geven.



Figuur 29: Manage scores pagina - paginatie

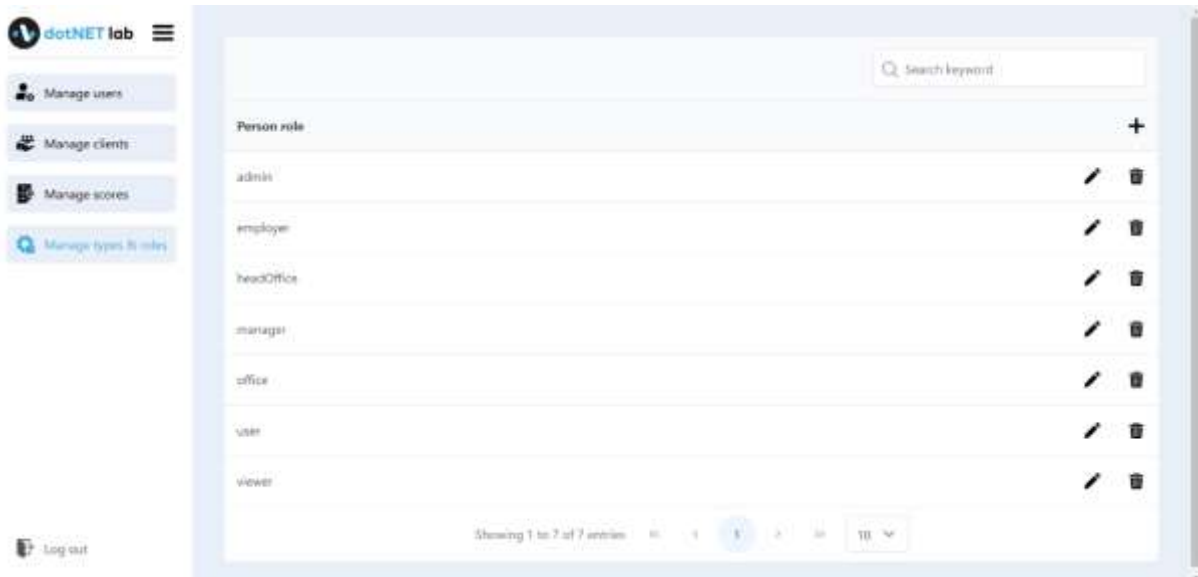
Uiteraard was ook deze pagina uitgerust met een modal voor het creëren, bewerken en verwijderen van de scores. Zo zie je op onderstaande afbeelding het bewerkscherm waar ook de verbeterpunten in konden worden bewerkt. Dit gebeurde op dezelfde manier als op de andere pagina's om de stijl overall gelijk te houden.



Figuur 30: Manage scores pagina - edit

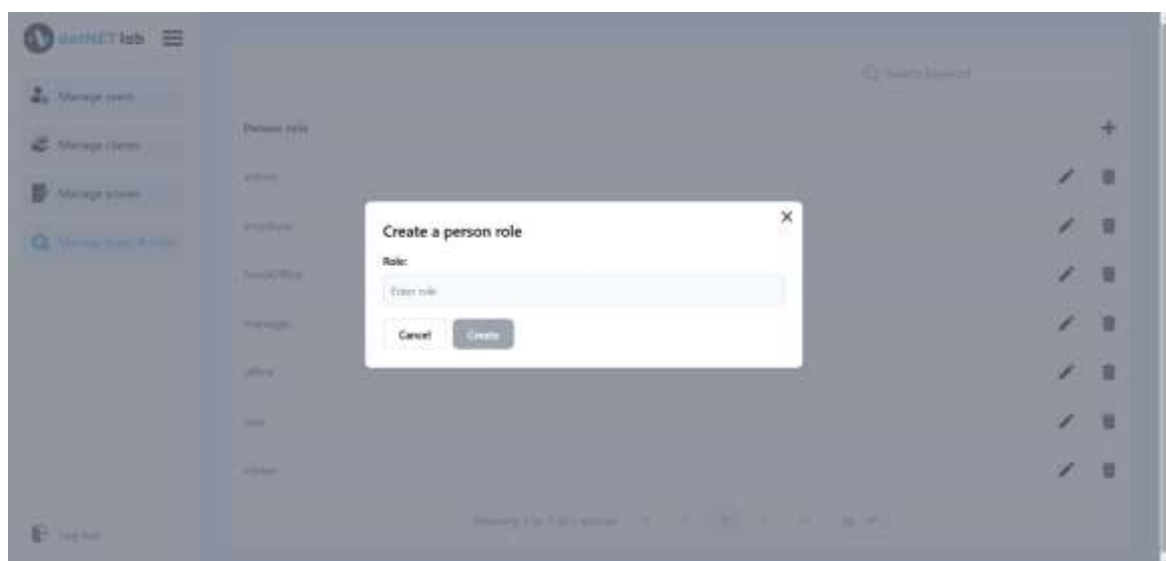
6.6.2.4 Manage types and roles

De laatste pagina die afgemaakt is, was de manage types and roles pagina. Tot op heden bevatte deze pagina enkel een scherm voor het bewerken van de rollen voor personen, maar in de toekomst ging deze gebruikt worden voor het beheren van alle verschillende rollen en types. Dit was nog niet in orde gebracht aangezien ik de focus had gelegd op het maken van een minimaal bruikbare applicatie. Het maken van vijf dezelfde pagina's maakte de applicatie niet bruikbaar en waren ook niet zo indrukwekkend als de andere pagina's.



Figuur 31: Manage types and roles pagina

Zoals je al verwacht had ook deze pagina een modal voor het creëren, bewerken en verwijderen van de rollen. Zo zie je op onderstaande afbeelding het creëer scherm. Je kan ook zien dat de "Create" knop grijs is en niet zwart zoals de voorgaande knoppen. Dit komt doordat de applicatie nakeek of alle data aanwezig was. Zolang er niets was ingevuld in het role-veld kon er niets worden aangemaakt.



Figuur 32: Manage types and roles pagina - edit

7 CONCLUSIE

Ik heb de stageperiode kunnen afsluiten met een werkende en bruikbare applicatie. Hierdoor ben ik enorm tevreden over het eindresultaat van mijn stage. Ik heb enorm veel bijgeleerd over verschillende zaken van zowel de front- als de back-end, maar ook over hoe het er in een echt bedrijf aan toegaat. Daarnaast heb ik mijn vaardigheden op het gebied van projectvoorbereiding en -analyse sterk verbeterd.

Door de applicatie die ik heb gemaakt kan dotNET lab hun cybersecurity aanbod uitbreiden en hun klanten bruikbare informatie geven voor het evolueren tot een veiliger bedrijf. De applicatie is ook voorzien op uitbreidingen waardoor ze er op een makkelijke manier aan kunnen verder werken. Dit mede door de gedetailleerde analyse die ik heb gemaakt.

De ervaringen die ik heb opgedaan gaan mij zeker en vast helpen in mijn professionele carrière. Van het maken van een zeer gedetailleerde analyse, tot het zelfstandig ontwikkelen van een applicatie zonder specifiek team. Dit heeft me niet alleen technische vaardigheden bijgebracht, maar ook heeft het me meer zelfzeker gemaakt. Het werken zonder teamgenoten heeft ervoor gezorgd dat ik zelfstandiger ben geworden en meer zelf kan beslissen.

Het was een waardevolle en prettige ervaring waar ik met een zeer positief beeld op zal terugkijken.

8 LITERATUURLIJST

CIS Center for Internet Security. (z.d.). *CIS*. Beschikbaar op <https://www.cisecurity.org/>

Huber, T. C. (2024). *C# Unit Testing*. Pluralsight. Beschikbaar op <https://app.pluralsight.com/library/courses/c-sharp-10-unit-testing/table-of-contents>

Peeters, T. (z.d.). *SOLID dotNETacademy* [Onuitgegeven intern document]. dotNET lab.

Peeters, T., & Boeijen, V. (z.d.). *ASP.NET CORE MASTER* [Onuitgegeven intern document]. dotNET lab.

Microsoft. (z.d.). *What is Microsoft Entra ID?*. Beschikbaar op <https://learn.microsoft.com/en-us/entra/fundamentals/whatis>