



Stage LHS Realisatie Document

Stage 2023-2024

Academiejaar 2023-2024

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Tymo Verhaegen 3APP02

Inhoudstafel

1 Dankwoord	5
2 Terminologie	6
3 Opdracht	7
3.1 Bedrijf	7
3.2 Team	7
3.3 Projectbeschrijving	7
4 Gebruikte technologieën en methodologieën	9
4.1 Technologieën	9
4.1.1 Django / Python	9
4.1.2 HTML & Tailwind CSS	9
4.1.3 PostgreSQL	9
4.1.4 Javascript	9
4.1.5 Docker	9
4.2 Programma's / Platformen	10
4.2.1 Figma	10
4.2.2 (Star)UML	10
4.2.3 Github	10
4.3 Methodologieën	10
4.3.1 SCRUM	10
4.4 Pipeline	11
4.4.1 CI/CD	11
5 Realisaties	12
5.1 Analyse	12
5.1.1 Use cases	12
5.1.1.1 Use Case Descriptions (klant)	13
5.1.1.2 Use Case Descriptions (personeel)	16
5.1.1.3 Use Case Descriptions (bedrijf)	18
5.1.1.4 Use Case Descriptions (tool)	20
5.1.2 Database design	21
5.1.3 Sitemap	22
5.1.4 Prototypes	23
5.1.4.1 Agenda (Weekweergave)	23
5.1.4.2 Agenda (Maandweergave)	23
5.1.4.3 Afspraak (details)	24
5.1.4.4 Statistieken & logs	24
5.1.4.5 Statistieken & logs (details)	25
5.1.4.6 Contacten	25
5.1.4.7 Contacten (aanmaken)	26
5.1.4.8 Contacten (details)	26
5.1.4.9 Middelen	27
5.1.4.10 Middelen (details)	27
5.1.4.11 Middelen (aanmaken filiaal)	28
5.1.4.12 Bedrijfsinstellingen	28

5.1.4.13 Bedrijfsinstellingen (details)	29
5.1.4.14 Promotiecodes	29
5.1.4.15 Promotiecodes (partners)	30
5.1.4.16 Promotiecodes (details)	30
5.1.4.17 Booking	31
5.1.4.18 Booking (cont.)	31
5.2 Front-end	32
5.2.1 Navigatie	32
5.2.2 Login	33
5.2.3 Agenda	33
5.2.4 Statistieken & logs	34
5.2.5 Klanten	35
5.2.6 Werknemers	36
5.2.7 Diensten & pakketten	38
5.2.8 Middelen	40
5.2.9 Promoties & aanbiedingen	41
5.2.10 Reviews	42
5.2.11 Bedrijfsinstellingen	42
5.2.12 Booking (end-user)	43
5.3 Backend	44
5.3.1 Navigatie	44
5.3.2 Login	45
5.3.3 Agenda	45
5.3.4 Statistieken & logs	47
5.3.5 Klanten, Werknemers en Partners	47
5.3.6 Reviews	50
5.3.7 Diensten & pakketten	51
5.3.8 Middelen	54
5.3.9 Promoties & aanbiedingen	57
5.3.10 Bedrijfsinstellingen	58
5.3.11 Booking (end-user)	59
6 Conclusie	60

1 DANKWOORD

Bij deze neem ik graag even de kans om LHS by Spot Groep een bedankje te doen voor mij de kans te geven om stage te doen bij het bedrijf. Dit had niet mogelijk geweest zonder Steven. Ik heb veel bijgeleerd tijdens deze stage en ik heb ervan genoten.

Ik wil ook mijn collega's en medestudenten bedanken, voornamelijk Thomas Verbruggen en Joppe Kerkhofs, zij waren mijn groepsgenoten voor het project en ik heb ervan genoten om met hen samen dit project uit te werken.

Uiteraard mogen we mijn stagementor Patrick Dielens niet vergeten. Hij was een zeer aangename en goede stagementor die altijd klaar stond als we hem nodig hadden.

2 TERMINOLOGIE

TERM	BETEKENIS
Framework	Platform of structuur voor het bouwen van software, dat tools en conventies biedt om ontwikkeling te vereenvoudigen.
Opmaaktaal	Een taal die de structuur van documenten definieert.
Responsief	Een ontwerpaanpak die zorgt voor een goede werking en weergave van software op verschillende apparaten.
Containerisatie technologie	Een methode om applicaties te verpakken en uit te voeren met behulp van containers, voor een consistente uitvoering op verschillende omgevingen.
UML	Unified Modelling Language : Een gestandaardiseerde modelleringstaal voor visualisatie, ontwerp en documentatie van softwareprojecten.
CI/CD	Continuous Integration / Continuous Deployment : Een praktijk voor het automatiseren van processen zoals testen en implementeren van code.
SEA/SEO	<ul style="list-style-type: none"> - Search Engine Advertisement Dit zijn betaalde advertenties in zoekmachines zoals Google Ads. - Search Engine Optimization Optimaliseren van webcontent om hoger te ranken in de organische (niet-betaalde) zoekresultaten
Use cases	Beschrijvingen van hoe gebruikers zullen interageren met een systeem/product. Ze schetsen de stappen die een gebruiker kan nemen en de mogelijke reacties hierop.
Sitemap	Visuele weergave van de structuur en hiërarchie van pagina's binnen een applicatie of website. Het toont de relatie tussen de pagina's.
Wireframes	Wireframes zijn schematische overzichten van een website/applicatie, voornamelijk gericht op layout en functionaliteit.
Prototypes	Prototypes zijn interactieve representaties van hoe het finale product er zal uitzien en hoe het zal functioneren
MVC	<p>Model View Controller is een ontwerppatroon dat het ontwerp van complexe toepassingen opdeelt in drie eenheden met verschillende verantwoordelijkheden</p> <ul style="list-style-type: none"> - Model: datamodel - View: datapresentatie - Controller: applicatielogica

3 OPDRACHT

3.1 Bedrijf

LHS is een klein bedrijf dat zich specialiseert in het bouwen van websites en webapplicaties, sinds 2023 is LHS deel van Spot Group. Spot Group is een groter bedrijf met meer capaciteit. Hierdoor is LHS nu onderdeel van een bedrijf dat meer diensten kan bieden op een grotere schaal, dit zijn diensten zoals SEA/SEO, hosting, marketing en nog veel meer.

3.2 Team

Het team waarmee ik dit project heb ontwikkeld bestond uit Thomas Verbruggen, Joppe Kerkhofs en mezelf. We zijn alle drie applicatieontwikkeling studenten van Thomas More in Geel.

Dit is niet het eerste project dat we samen hebben ontwikkeld, we hebben namelijk al schoolprojecten samen gemaakt wat voordelen had voor het stageproject. We weten namelijk de sterke en zwakke punten van elkaar en kunnen elkaar helpen waar nodig.

3.3 Projectbeschrijving

Een complexe boekingstool en modulair management systeem met een uitgebreid scala aan functionaliteiten, functionaliteiten omvatten onder anderen:

- **Boekingssysteem**

Het centrale aspect van de webapplicatie, het geeft klanten de mogelijkheid om eenvoudig afspraken, reserveringen of boekingen te maken op basis van beschikbaarheid.

- **Resource management**

Het systeem zal functies bevatten voor het beheer van bedrijfsmiddelen, zoals ruimtes, apparatuur, personeel, Dit omvat dus ook het bijhouden van beschikbaarheid van de middelen, toewijzingen van personeel, ...

- **Kalender integratie**

Integratie van een kalender zal worden aangeboden, waardoor zowel klanten als medewerkers eenvoudig afspraken kunnen bekijken, wijzigen en bijhouden.

- **Diensten en pakketten**

Het systeem zal functies bevatten voor het beheren van diensten en pakketten, zo kunnen klanten makkelijk de juiste optie voor hun behoeften kiezen.

- **Betalingssysteem**

Een veilig betalingssysteem zal worden geïmplementeerd om online betalingen voor boekingen mogelijk te maken. Dit wordt geïmplementeerd om tijd te besparen en om voorafbetalingen te doen.

- **Klantenbeheer**

Een soort van klantenbestand zal worden bijgehouden, met informatie over de klant, persoonlijke informatie, boekingsgeschiedenis en voorkeuren.

- **Beheer van aanbiedingen en promoties**

Het systeem zal de functionaliteit bieden om aanbiedingen, promoties en kortingen te kunnen beheren en toe te passen op de boekingen van de klanten.

- **Communicatie en notificaties**

Klanten krijgen automatisch meldingen via e-mail wanneer ze een boeking doen, of wanneer er veranderingen gebeuren aan hun boeking.

- **Statistieken en logs**

Medewerkers zullen statistieken kunnen bekijken zoals aantal boekingen, aantal gebruikers en meer, ze zullen ook een gedetailleerde geschiedenis kunnen bekijken van alle klanten en hun boekingsgeschiedenis

- **Cronjobs**

Het systeem zal geautomatiseerde taken en processen uitvoeren, zoals het optimaliseren van de kalender en boekingen.

- ...

4 GEBRUIKTE TECHNOLOGIEËN EN METHODOLOGIEËN

4.1 Technologieën

4.1.1 Django / Python

Django is een high-level open-source Python based web framework dat snelle ontwikkeling en een strak, pragmatisch ontwerp aanmoedigt. Django maakt gebruik van het MVC principe.

Reden: Django was de voorkeur van LHS voor dit project.

4.1.2 HTML & Tailwind CSS

HTML is een opmaaktaal die wordt gebruikt om de structuur van webpagina's te definiëren, Tailwind is een framework dat gebruikt wordt om webpagina's te stylen, om snel en efficiënt responsieve, op maat gemaakte gebruikersinterfaces op te bouwen.

Reden: HTML is verplicht om een webapplicatie te maken, Tailwind was de persoonlijke voorkeur van de groep. Dankzij het uitgebreide gebruik van deze technologie voor schoolprojecten was dit een eenvoudige keuze.

4.1.3 PostgreSQL

PostgreSQL is een open-source relational databasebeheersysteem (RDBMS) dat gebruikt wordt om gegevens op te slaan en te beheren.

Reden: PostgreSQL was de voorkeur van LHS voor dit project.

4.1.4 Javascript

Javascript is een programmeertaal die wordt gebruikt voor het ontwikkelen van interactieve webpagina's. Het geeft ontwikkelaars de mogelijkheid om de inhoud van webpagina's te manipuleren.

Reden: Javascript is zowat verplicht om een goed functionerende webapplicatie te maken.

4.1.5 Docker

Docker is een platform dat gebruikt wordt voor het ontwikkelen, verzenden en uitvoeren van applicaties. Het maakt gebruik van containerisatie technologie om applicaties te verpakken met alle benodigde code.

Reden: Docker was een persoonlijke voorkeur van de groep, zo zou iedereen altijd dezelfde test data bezitten en zou alles een beetje makkelijker lopen.

4.2 Programma's / Platformen

4.2.1 Figma

Figma is een ontwerptool dat wordt gebruikt voor het maken van gebruikersinterfaces, wireframes, prototypes en andere ontwerpelementen.

Reden: Figma was een persoonlijke voorkeur van de groep voor het maken van sitemaps, wireframes en prototypes. Dit is tevens ook een industriestandaard.

4.2.2 (Star)UML

UML is een gestandaardiseerde modelleringstaal die wordt gebruikt voor het visualiseren, ontwerpen en documenteren van softwareprojecten.

Reden: StarUML was een persoonlijke voorkeur van de groep om onze database en use cases in te ontwerpen, wegens het uitgebreide gebruik ervan tijdens schoolprojecten.

4.2.3 Github

Github is een platform dat gebruikt wordt voor versiebeheer en samenwerking tijdens het ontwikkelen van software. Op Github kunnen ontwikkelaars hun code opslaan, bewerken en delen met anderen.

Reden: Github was vanzelfsprekend, het was de voorkeur van LHS en van de groep wegens het uitgebreide gebruik tijdens schoolprojecten en dit is ook de industriestandaard.

4.3 Methodologieën

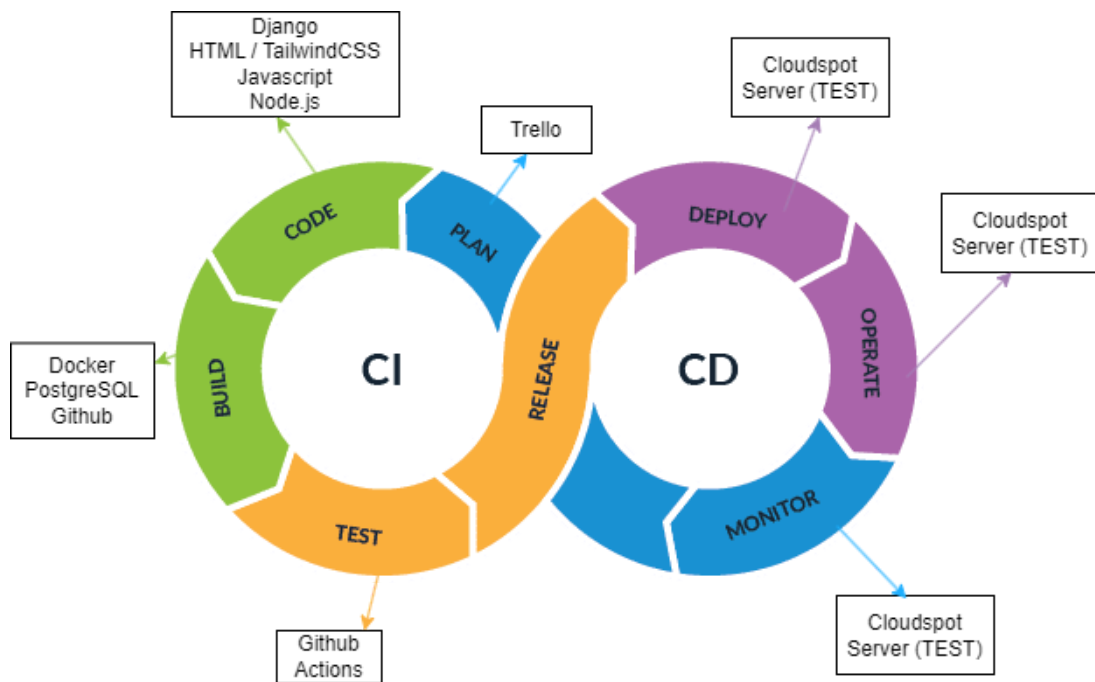
4.3.1 SCRUM

Een agile methodologie die werkt op basis van sprints, typisch van twee tot vier weken. Het omvat ook nog retrospectives waarin het team terugblijkt op de vorige sprint en elkaar feedback bezorgt over de positieve en negatieve punten van die sprint.

Reden: SCRUM-methodologie was voor onze groep de voorkeur van werken, door het extensief gebruik van deze methodologie voor schoolprojecten was dit een makkelijke keuze.

4.4 Pipeline

4.4.1 CI/CD



5 REALISATIES

5.1 Analyse

De analyse van het project heeft plaatsgevonden in week 4, in deze fase heeft het team een grondige analyse gemaakt van de gebruikersbehoefte.

5.1.1 Use cases

Voor het maken van de use cases is er gebruik gemaakt van StarUML



5.1.1.1 Use Case Descriptions (klant)

- Use Case: Login
 - **Beschrijving:** De klant kan inloggen op het systeem door zijn/haar email en wachtwoord in te geven
 - **Na-omstandigheden:** De klant ziet de homepage van de klant-view

- Use Case: Logout
 - **Beschrijving:** De klant kan zich afmelden uit het systeem door op de logout knop de klikken
 - **Voorwaarden:** De klant is ingelogd
 - **Na-omstandigheden:** De klant is afgemeld en wordt naar het login scherm gestuurd

- Use Case: Afspraak maken
 - **Beschrijving:** De klant kan een afspraak maken door een dienst of een pakket te selecteren. Hierna een tijdstip te selecteren en vervolgens op bevestigen te klikken. De klant zal 50% van het bedrag op voorhand moeten betalen (*zie use case betalen*)
 - **Voorwaarden:** De klant is ingelogd
 - **Na-omstandigheden:** De klant wordt naar de homepage van de klant-view gestuurd. Een bevestigingsmail wordt naar de klant gestuurd
 - **Alternatieve flows:**
 - Personeel is ingelogd
 - Het personeelslid kan op alle uren een afspraak maken. Hij/zij kan het voorstel van het systeem overschrijven

- Use Case: Afspraak bewerken
 - **Beschrijving:** De klant kan de diensten en het uur aanpassen van zijn/haar afspraak
 - **Voorwaarden:** De klant is ingelogd
 - **Na-omstandigheden:** De aangepaste afspraak wordt opgeslagen in het systeem en een bevestigingsmail met de veranderingen wordt naar de klant gestuurd

- Use Case: Betalen
 - **Beschrijving:** De klant wordt naar het Stripe betaalsysteem gestuurd waar de betaling afgerond zal worden
 - **Voorwaarden:** De klant is ingelogd en heeft een geldige afspraak in het systeem

- Use Case: Waardebon gebruiken
 - **Beschrijving:** De klant kan tijdens de betaling een code ingeven van een waardebon of kortingscode
 - **Voorwaarden:** De klant is ingelogd, heeft een geldige afspraak in het systeem en is bezig met de betaling
 - **Na-omstandigheden:** Het eindbedrag heeft een vermindering in prijs

- Use Case: Beoordeling achterlaten
 - **Beschrijving:** De klant kan na zijn/haar afspraak een beoordeling achterlaten
 - **Voorwaarden:** De klant is ingelogd en heeft een voltooide afspraak
 - **Na-omstandigheden:** De beoordeling wordt in het systeem opgeslagen en kan gebruikt worden als "testimonial" op de website van het bedrijf

- Use Case: Medische informatie delen
 - **Beschrijving:** De klant kan zijn/haar medische informatie invullen
 - **Voorwaarden:** De klant is ingelogd
 - **Na-omstandigheden:** De medische informatie van de klant wordt opgeslagen in het systeem

- Use Case: Prijslijst raadplegen
 - **Beschrijving:** De klant kan de prijslijst van het bedrijf bekijken

- Use Case: Openingsuren raadplegen
 - **Beschrijving:** De klant kan de openingsuren van het bedrijf bekijken

- Use Case: Beoordeling achterlaten
 - **Beschrijving:** De klant kan na zijn/haar afspraak een beoordeling achterlaten
 - **Voorwaarden:** De klant is ingelogd en heeft een voltooide afspraak
 - **Na-omstandigheden:** De beoordeling wordt in het systeem opgeslagen en kan gebruikt worden als "testimonial" op de website van het bedrijf

5.1.1.2 Use Case Descriptions (personeel)

Personeel erft alle use cases van klant

- Use Case: Werkuren beheren
 - **Beschrijving:** Het personeel kan zijn/haar werkuren ingeven per dag
 - **Voorwaarden:** Het personeelslid is ingelogd
 - **Na-omstandigheden:** De werkuren worden opgeslagen in het systeem

- Use Case: Agenda raadplegen
 - **Beschrijving:** Het personeel kan zijn/haar agenda bekijken
 - **Voorwaarden:** Het personeelslid is ingelogd

- Use Case: Betalingen beheren
 - **Beschrijving:** Het personeel kan de betalingen loggen
 - **Voorwaarden:** Het personeelslid is ingelogd

- Use Case: Facturen beheren
 - **Beschrijving:** Het personeel kan na het afronden van een behandeling een factuur aanmaken dat automatisch verstuurd wordt naar de mailbox van de klant
 - **Voorwaarden:** Het personeelslid is ingelogd en de behandeling is voltooid
 - **Na-omstandigheden:** De klant ontvangt een factuur in zijn/haar mailbox

- Use Case: Beoordelingen raadplegen
 - **Beschrijving:** Het personeel kan beoordelingen bekijken
 - **Voorwaarden:** Het personeelslid is ingelogd

- Use Case: Medische informatie raadplegen
 - **Beschrijving:** Het personeel kan kijken naar de medische informatie van een klant
 - **Voorwaarden:** Het personeelslid is ingelogd en de klant heeft een valide medische fiche geüpload
 - **Na-omstandigheden:** De werkuren worden opgeslagen in het systeem

- Use Case: Waardebonnen beheren
 - **Beschrijving:** Het personeel kan waardebonnen aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het personeelslid is ingelogd
 - **Na-omstandigheden:** De klant ontvangt promoties in zijn/haar mailbox

- Use Case: Log raadplegen
 - **Beschrijving:** Het personeel kan logs bekijken van alle boekingen
 - **Voorwaarden:** Het personeelslid is ingelogd

5.1.1.3 Use Case Descriptions (bedrijf)

- Use Case: Personeel beheren
 - **Beschrijving:** Het bedrijf kan personeel aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** Personeel accounts worden opgeslagen in het systeem. Nadien kan personeel op het platform

- Use Case: Resources beheren
 - **Beschrijving:** Het bedrijf kan resources aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** Resources worden opgeslagen in het systeem

- Use Case: Promotiecodes beheren
 - **Beschrijving:** Het bedrijf kan promotiecodes aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** Promotiecodes worden opgeslagen in het systeem en kunnen gebruikt worden door de end-user

- Use Case: Commissies beheren
 - **Beschrijving:** Het bedrijf kan commissie codes aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het bedrijf is ingelogd, en er moet een valide partner bestaan in het systeem waaraan deze code gelinkt kan worden
 - **Na-omstandigheden:** Commissie codes worden opgeslagen in het systeem. Als een klant deze codes gebruikt zal de gelinkte partner een deel van het bedrag ontvangen

- Use Case: Diensten beheren
 - **Beschrijving:** Het bedrijf kan diensten aanmaken, bewerken en verwijderen
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** De diensten worden opgeslagen in het systeem. Prijslijsten van het bedrijf worden automatisch geüpdatet

- Use Case: Pakketten beheren
 - **Beschrijving:** Het bedrijf kan pakketten aanmaken, bewerken en verwijderen.
 - **Voorwaarden:** Het bedrijf is ingelogd, en er moeten valide diensten bestaan in het systeem
 - **Na-omstandigheden:** Pakketten worden opgeslagen in het systeem

- Use Case: Openingsuren beheren
 - **Beschrijving:** Het bedrijf kan openingsuren aanmaken of bewerken
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** Openingsuren worden opgeslagen in het systeem

- Use Case: Bedrijf beheren
 - **Beschrijving:** Het bedrijf kan bedrijfsgegevens aanpassen
 - **Voorwaarden:** Het bedrijf is ingelogd
 - **Na-omstandigheden:** Tool wordt aangepast naar voorkeur van bedrijf

5.1.1.4 Use Case Descriptions (tool)

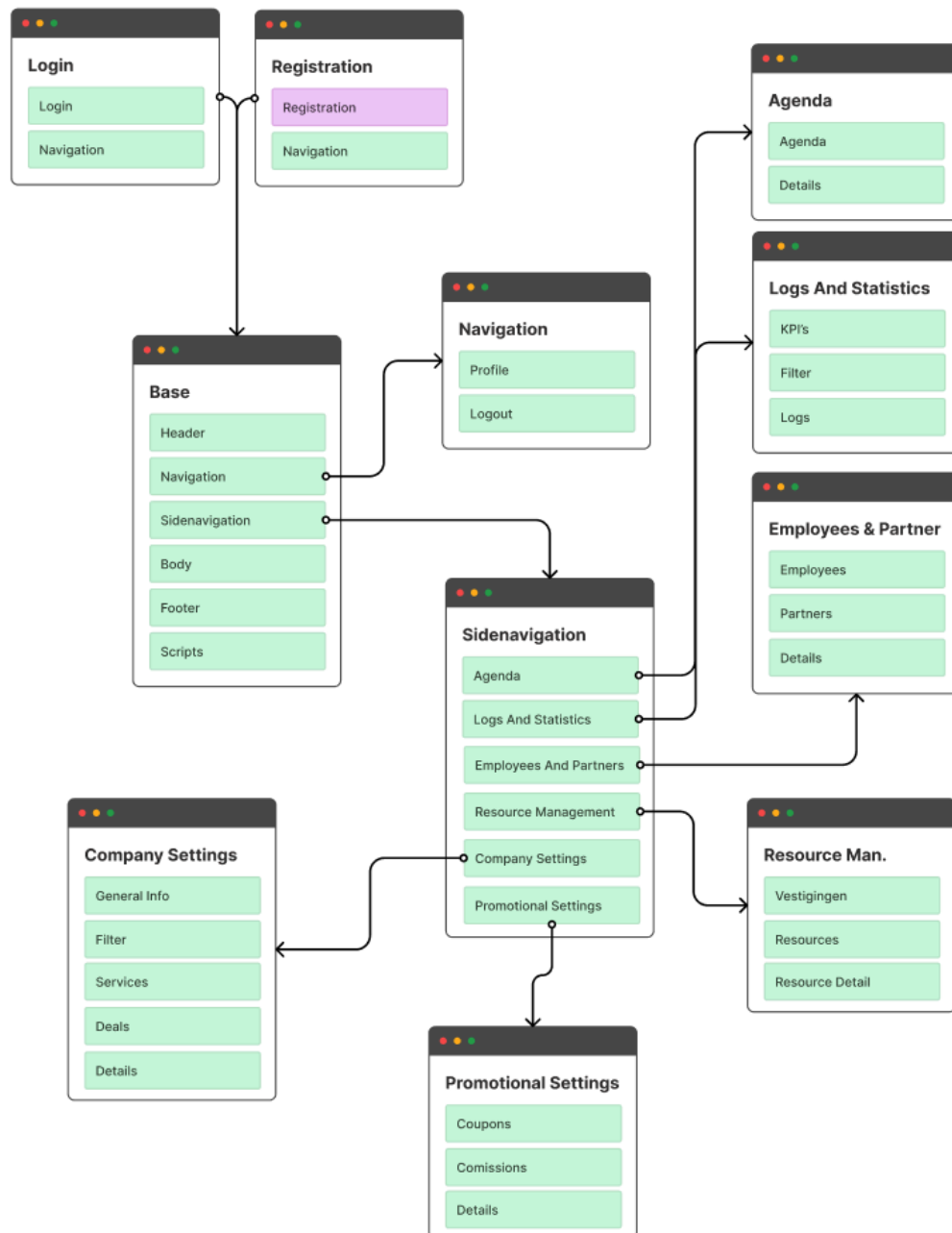
- Use Case: Agendaoptimalisatie
 - **Beschrijving:** Het systeem gaat automatisch zoeken naar mogelijke optimalisaties in de agenda, door boekingen te verplaatsen. Als er een optimalisatie gevonden is dan zal het systeem automatisch een mail sturen naar de respectievelijke klant om te zien of de boeking verplaatst kan worden. Als de klant hiermee akkoord is zal het systeem optimalisaties uitvoeren.
 - **Voorwaarden:** Er zijn boekingen in het systeem
 - **Na-omstandigheden:** De agenda is geoptimaliseerd
 - **Alternatieve flows:**
 - Klant gaat niet akkoord met verplaatsing van boeking
 - Systeem zal mail sturen naar de volgende klant, als deze akkoord is met de verplaatsing wordt de afspraak van deze klant verzet. Het systeem blijft zo oneindig doorgaan.
 - Klant gaat akkoord maar daagt niet op
 - Personeelslid zal dit ingeven in het systeem als "no-show". Vervolgens wordt er een mail gestuurd naar de klant die niet is komen opdagen. Ook wordt er een mail gestuurd naar de volgende klant om te zien of hij/zij hun afspraak wil verplaatsen.
 - Personeelslid is afwezig
 - Dit wordt aangegeven in het systeem en de klanten voor het afwezige personeelslid worden allemaal geïnformeerd via mail. Er wordt nog steeds een optimalisatie uitgevoerd om te zien of een ander personeelslid de werkdruk kan opvangen

5.1.2 Database design



5.1.3 Sitemap

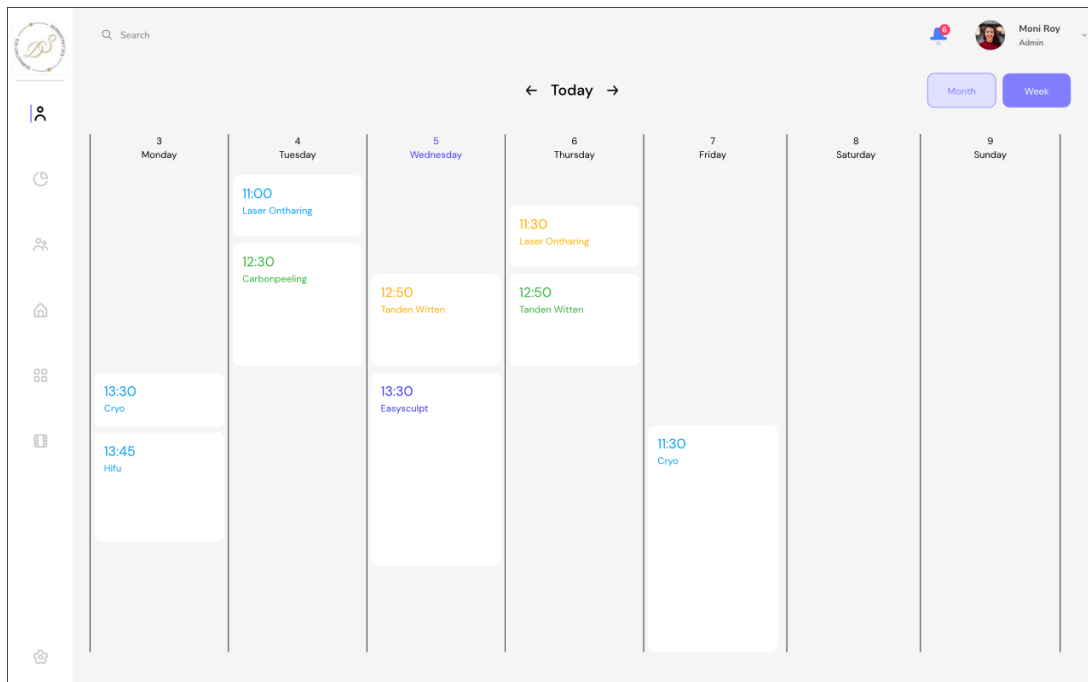
Voor het maken van de sitemap is Figma gebruikt.



5.1.4 Prototypes

Wegens het feit dat we de projectopgave pas op het einde van week 3 hebben gekregen, heb ik besloten om direct prototypes te maken, om zeker te zijn dat er geen tijdgebrek zou zijn. Voor het maken van de prototypes is Figma gebruikt

5.1.4.1 Agenda (Weekweergave)



5.1.4.2 Agenda (Maandweergave)

Today < Maart 2024 >

MON	TUE	WED	THE	FRI	SAT	SUN
25	26	27	28	29	30	1
2	3 Design Conference	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
Weekend Festival	23	24	25	26	27 Glastonbury Festival	28
23	24	25	26	27	28	29
30	31	1 Glastonbury Festival	2	3	4	5

5.1.4.3 Afspraak (details)

Afspraak: Hifu

Tijdstip: 13:45 - 15:00

Datum: 26-03-2024

Klant / Patient: Darrell Caldwell

Email: darrel.cwl@outlook.com

Sluit

Q Search

Mani Roy
Admin

Afspraak

Tijdstip

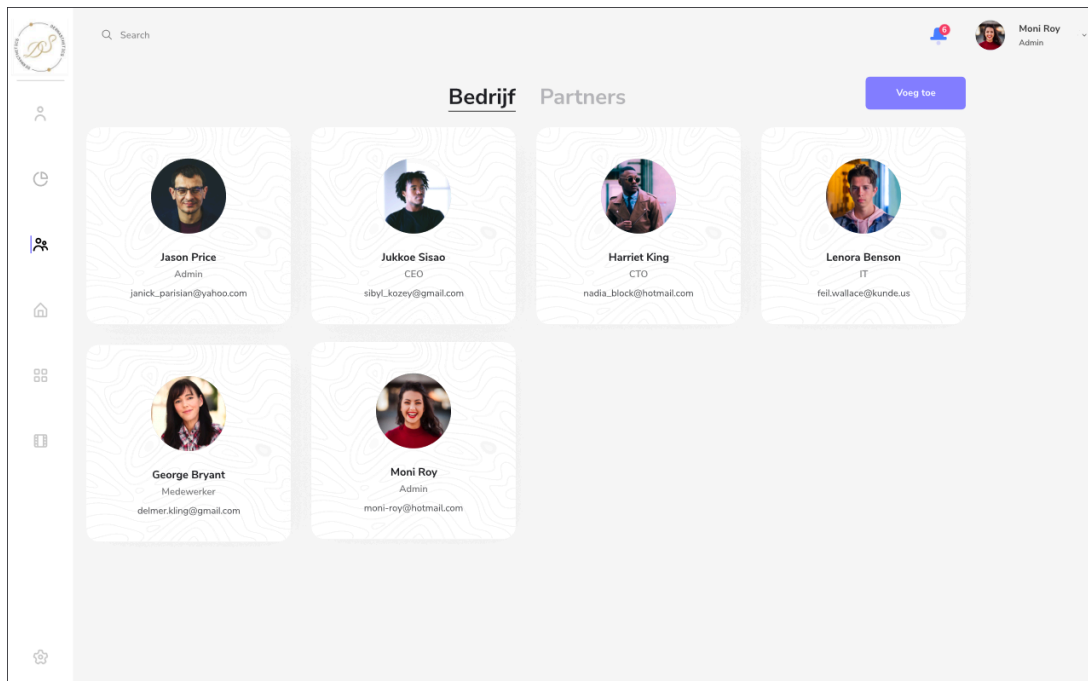
Datum

Klant / Patient

Email

Sluit

5.1.4.6 Contacten



5.1.4.7 Contacten (aanmaken)

Search

Upload Photo

Voornaam
Voer voornaam in

Achternaam
Voer achternaam in

Email
Voer email in

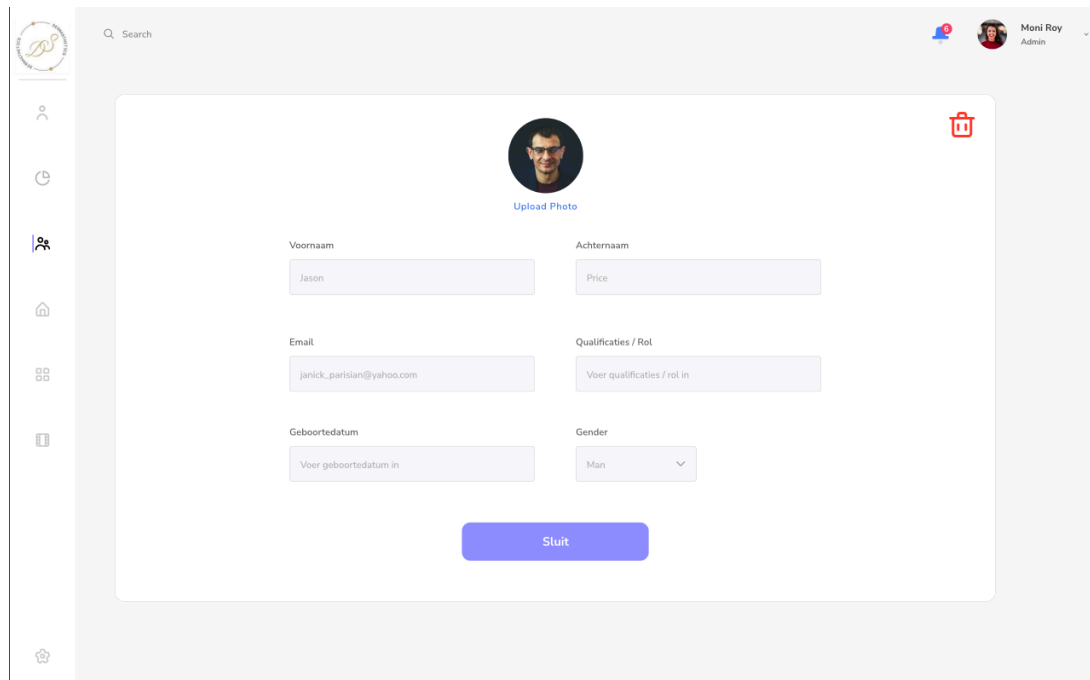
Qualificaties / Rol
Admin

Geboortedatum
20-03-1980

Gender
Man

Voeg Toe

5.1.4.8 Contacten (details)



Search

Moni Roy Admin

Upload Photo

Voornaam
Jason

Achternaam
Price

Email
janick_parsian@yahoo.com

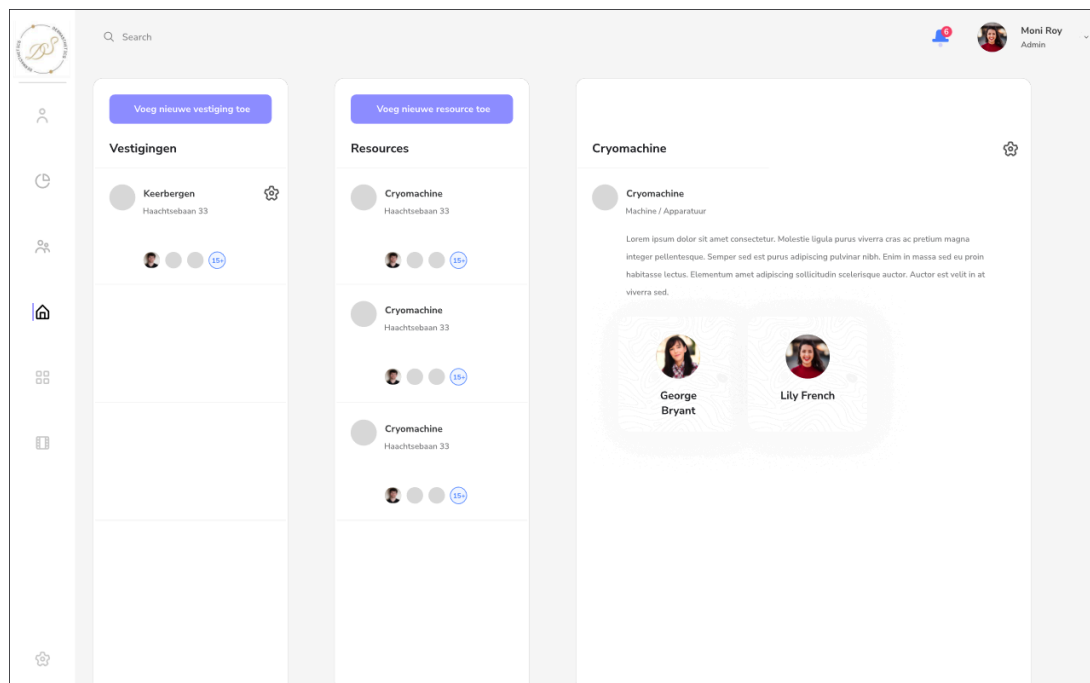
Qualificaties / Rol
Voer qualificaties / rol in

Geboortedatum
Voer geboortedatum in

Gender
Man

Sluit

5.1.4.9 Middelen



Search

Moni Roy Admin

Vestigingen

Voeg nieuwe vestiging toe

Keerbergen
Haachtsebaan 33

Resources

Voeg nieuwe resource toe

Cryomachine
Haachtsebaan 33

Cryomachine
Haachtsebaan 33

Cryomachine
Haachtsebaan 33

Cryomachine

Cryomachine
Machine / Apparatuur

Lorem ipsum dolor sit amet consectetur. Molestie ligula purus viverra cras ac pretium magna integer pellentesque. Semper sed est purus adipiscing pulvinar nibh. Enim in massa sed eu proin habitasse lectus. Elementum amet adipiscing sollicitudin scelerisque auctor. Auctor est velit in at viverra sed.

George Bryant

Lily French

5.1.4.10 Middelen (details)

Naam: Cryomachine

Type: Machine / Apparatuur

Beschrijving: Lorem Ipsum ...

Specificaties: Medewerker

Sluit

5.1.4.11 Middelen (aanmaken filiaal)

Naam: Keerbergen

Adres: Haachtsebaan 33, Keerbergen

Openingsuren: Maandag, Dinsdag, Woensdag, Donderdag, Vrijdag, Zaterdag, Zondag

Sluit

5.1.4.12 Bedrijfsinstellingen


The screenshot shows a business profile for 'Dermaesthetics'. The profile includes a logo, contact information (Haachtsebaan 33, Keerbergen; Tel: 015 69 03 60; IG: dermaesthetics4you; FB: DermaestheticsByTaoStudio), and a list of services. The 'Diensten' (Services) tab is active, showing three 'Cryo' services, each priced at €75.00. The interface includes a search bar, a user profile (Moni Roy Admin), and a sidebar with navigation icons.

5.1.4.13 Bedrijfsinstellingen (details)

The screenshot shows the 'Bedrijfsinstellingen (details)' modal form. It contains fields for 'Naam' (Name), 'Instagram', 'Telefoon' (Phone), and 'Facebook'. To the right of these fields are three radio buttons for selection. Below the fields are sections for 'Categorieën' (Categories) and 'Specialisaties' (Specialties), each with three buttons: 'Machine', 'Hilt', and 'Voeg toe' (Add). A 'Sluit' (Close) button is at the bottom of the modal.

ID	PARTNER	CODE	KORTING	DATUM	COMMISSIE	STATUS
00004	Spot Group	2003944920	20%	14 Apr 2024	20%	Actief
00005	LHS	220209120	15%	22 May 2024	10%	Actief
00006	Darrell Caldwell	3930202230	40%	14 Feb 2019	20%	Inactief

5.1.4.16 Promotiecodes (details)



Search

Moni Roy
Admin

Code

931001929

Hoeveelheid

10

Bedrag

10€

Vervaldatum


25 Mei 2024

Type

Coupon

Sluit

5.1.4.17 Booking



Afspraak


Hifu

← Today →

Month Week

3 Monday	4 Tuesday	5 Wednesday	6 Thursday	7 Friday	8 Saturday	9 Sunday
11:00 Boek	11:00 Bezet		11:30 Bezet			
11:30 Boek	12:30 Bezet		12:50 Bezet			
12:00 Boek		12:50 Bezet	12:50 Bezet			
		13:30 Bezet				
13:45 Bezet				11:30 Bezet		

5.1.4.18 Booking (cont.)



Voornaam

Darrell

Achternaam

Caldwell

Email

darrelcwl@outlook.com

Invoice email

darrelcwl@outlook.com

Telefoon

+32 479 80 86 39

Adres

Gelderweg 1

Post code

3304

Stad

Aarschot

Opmerking

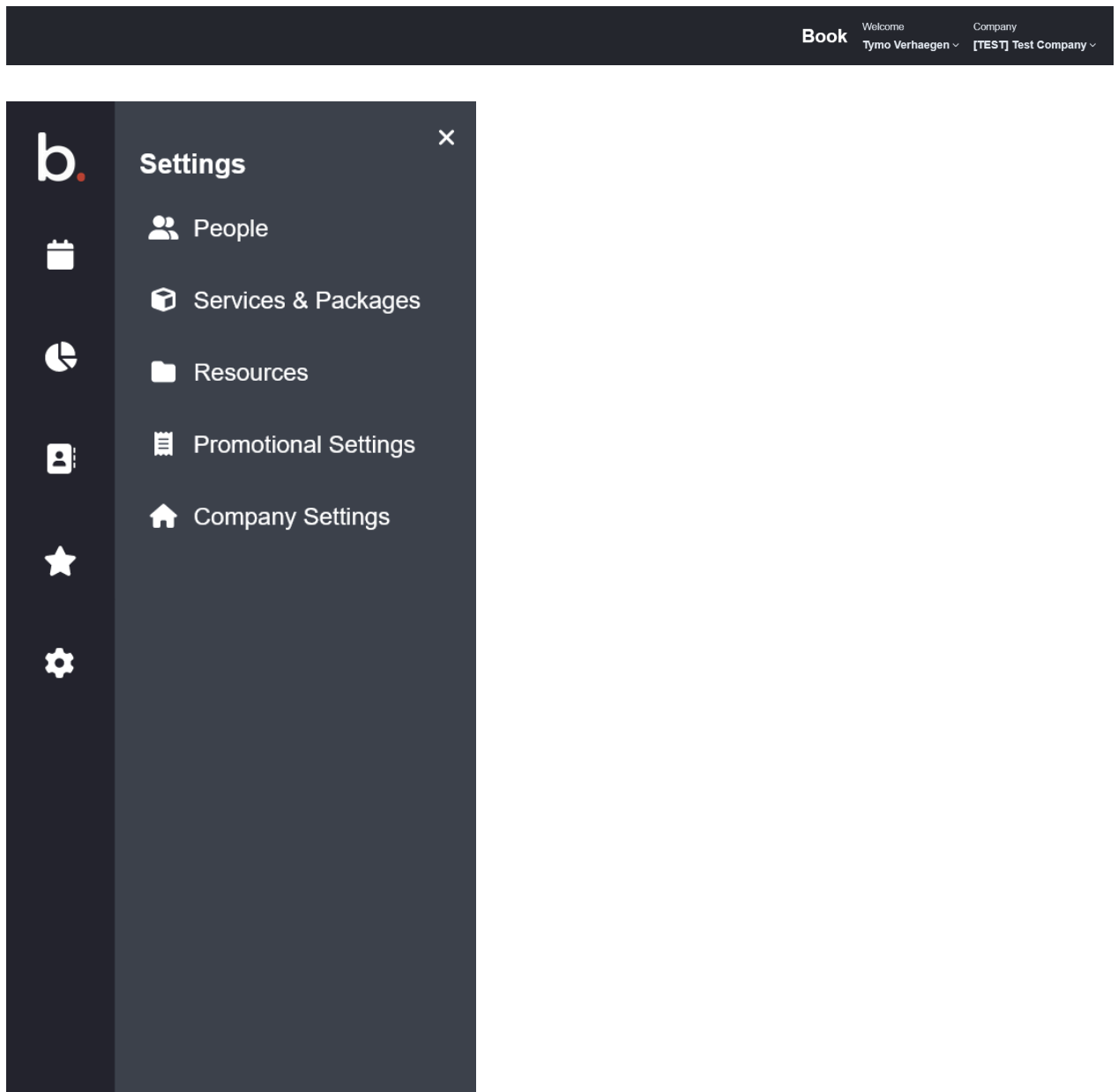
Bevestig booking

5.2 Front-end

Het front-end gedeelte van onze applicatie is gemaakt in Django & HTML, Tailwind CSS is gebruikt voor styling en Javascript is gebruikt voor animaties, pop-up windows en meer. Er zijn beide dark- en light mode voor de webapp.

5.2.1 Navigatie

Ontwikkeld door mezelf en Joppe Kerkhofs



5.2.2 Login

Ontwikkeld door Joppe Kerkhofs

The login form is centered on a dark background. It features the title 'Inloggen' in white. Below the title are two input fields: 'Email:' and 'Wachtwoord:'. A blue 'Login' button is positioned below the password field.

Field	Value
Email:	
Wachtwoord:	

Login

5.2.3 Agenda

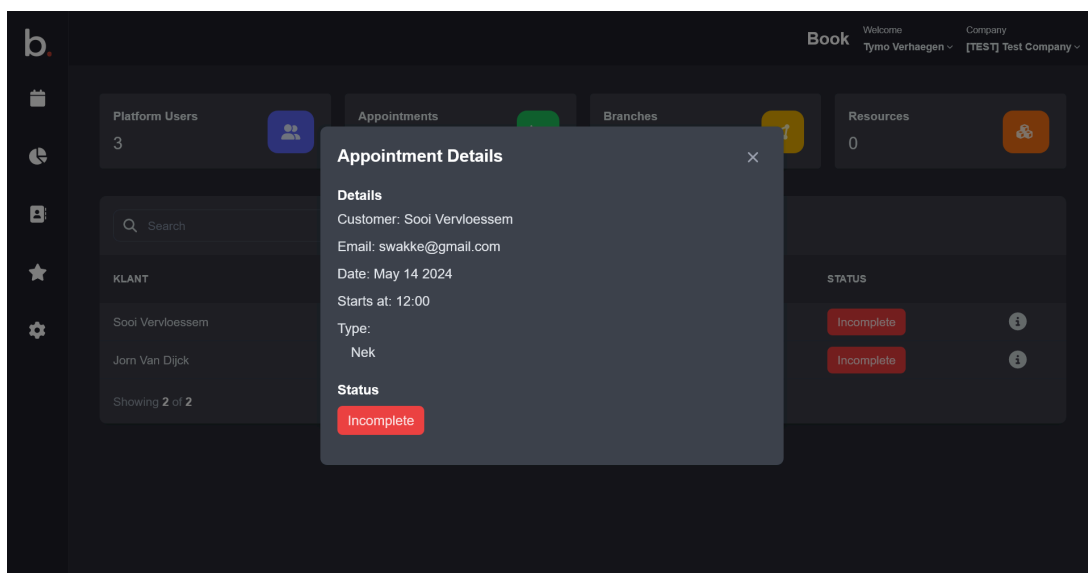
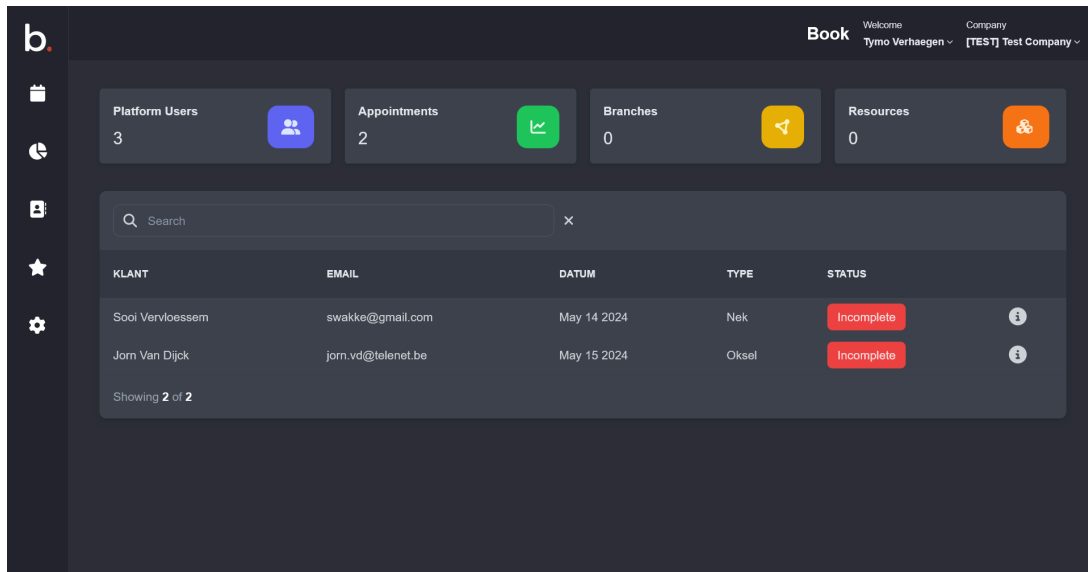
Ontwikkeld door Thomas Verbruggen

The agenda view displays a calendar for the week of May 12 to 18, 2024. The interface includes a sidebar with navigation icons, a top bar with user information, and a main calendar grid. The grid shows events for each day, with specific events highlighted for Tuesday, May 14th and Wednesday, May 15th.

	Sun 5/12	Mon 5/13	Tue 5/14	Wed 5/15	Thu 5/16	Fri 5/17	Sat 5/18
all-day							
8am							
9am							
10am							
11am							
12pm			12:00 - 1:30 Soci Vervloessem - Nek				
1pm							
2pm				2:00 - 2:30 - Jeroen Van Oort			
3pm							

5.2.4 Statistieken & logs

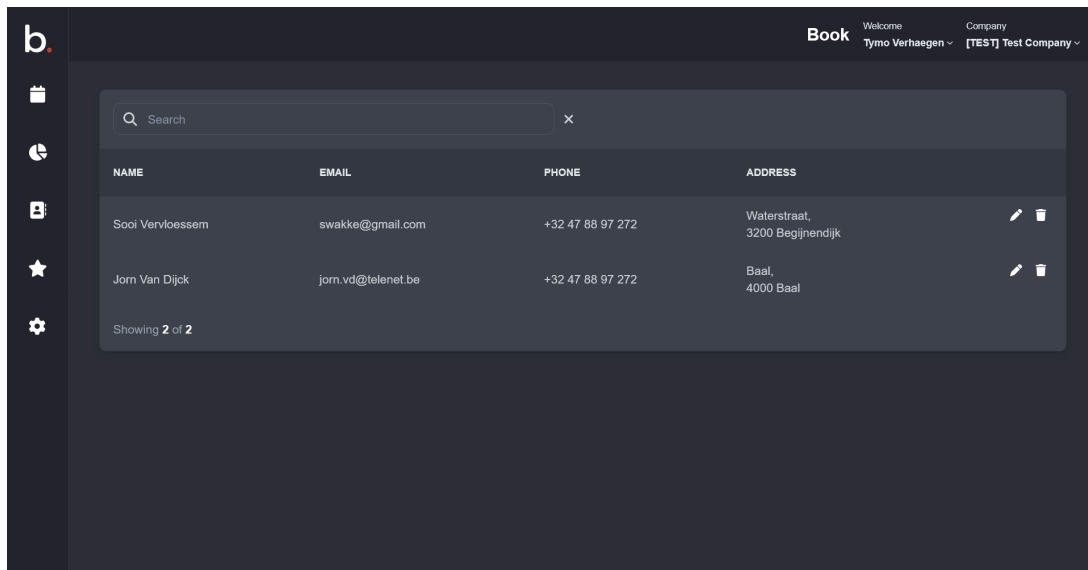
Ik heb deze volledige pagina ontwikkeld, zowel front-end als backend.



5.2.5 Klanten

Ontwikkeld door Joppe Kerkhofs en mezelf

Customer view



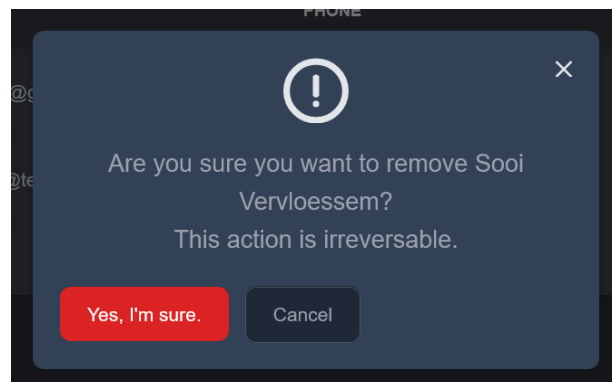
Update customer

Update - Sooi Vervloessem

FIRST NAME	LAST NAME
Sooi	Vervloessem
EMAIL	PHONE
swakke@gmail.com	+32 47 88 97 272
ADDRESS	ZIPCODE
Waterstraat	3200
CITY	
Begijnendijk	

Update Customer

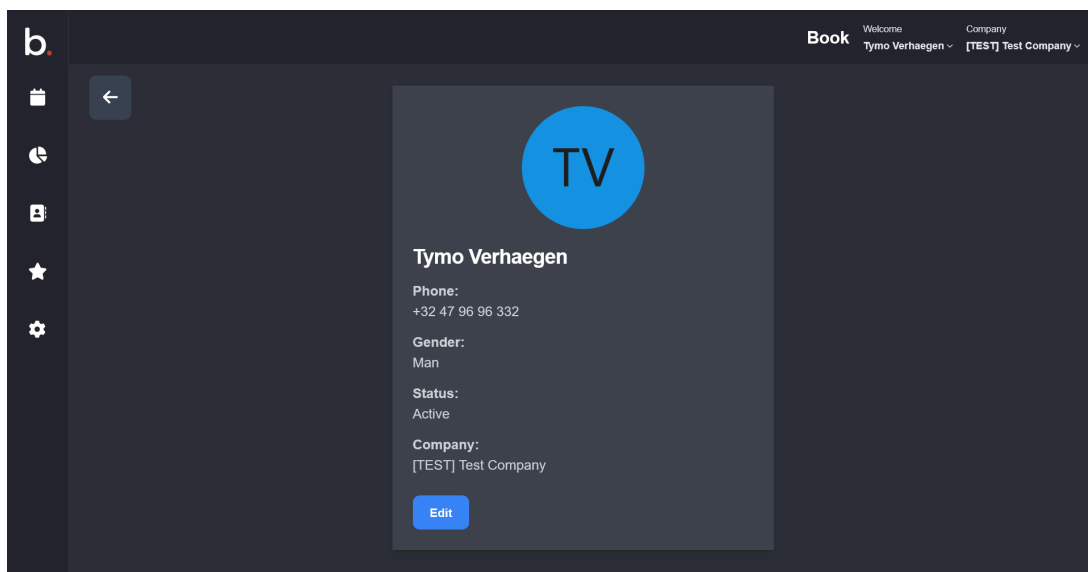
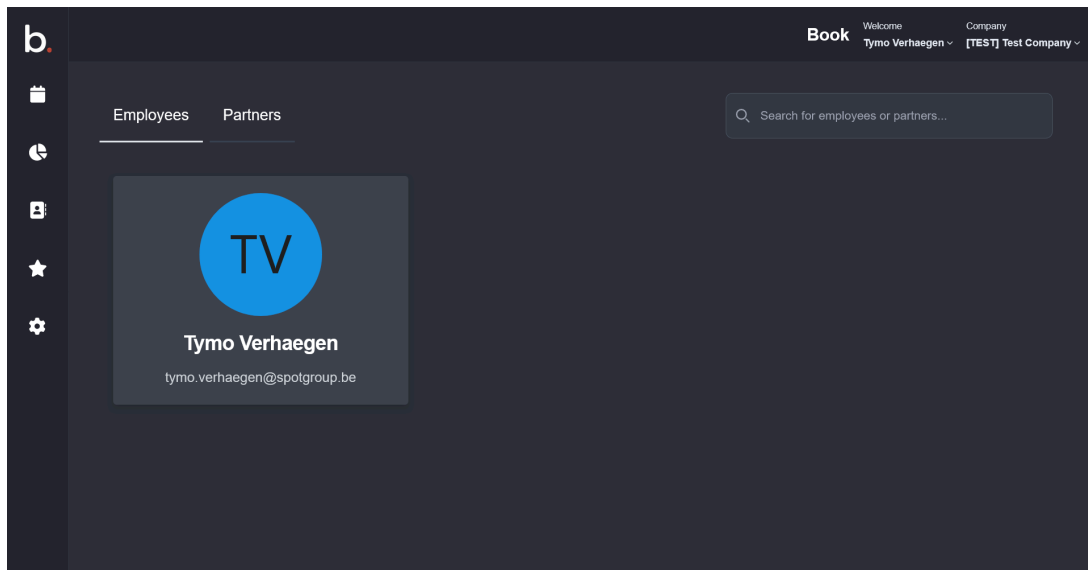
Delete customer



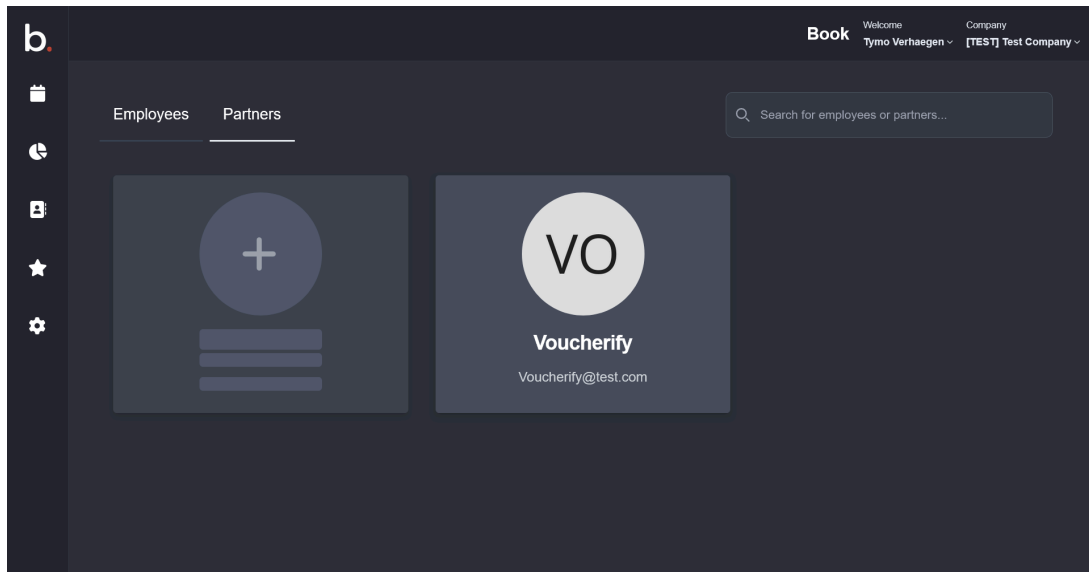
5.2.6 Werknemers

Ontwikkeld door Joppe Kerkhofs en mezelf

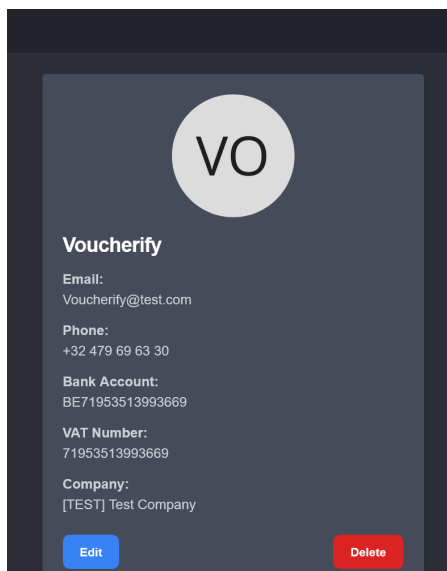
Werknemer tab



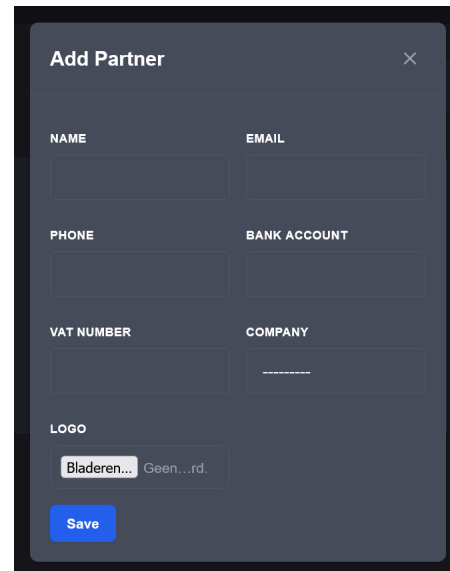
Partner tab



Partner info



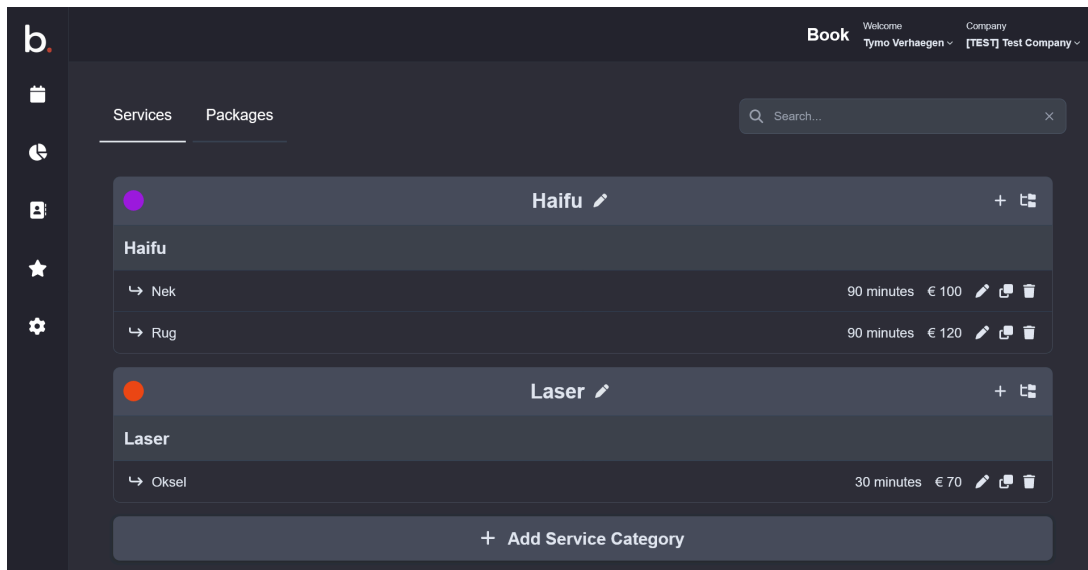
Partner add



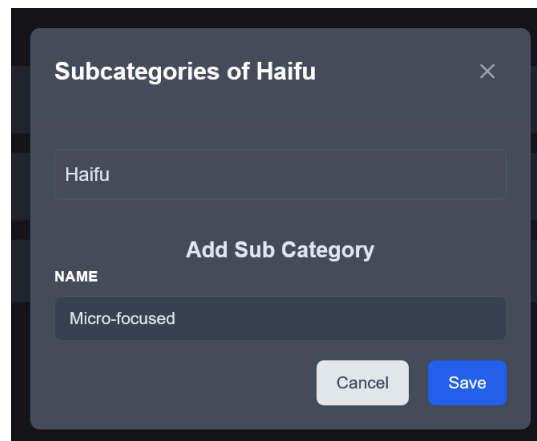
5.2.7 Diensten & pakketten

Ontwikkeld door Joppe Kerkhofs en mezelf

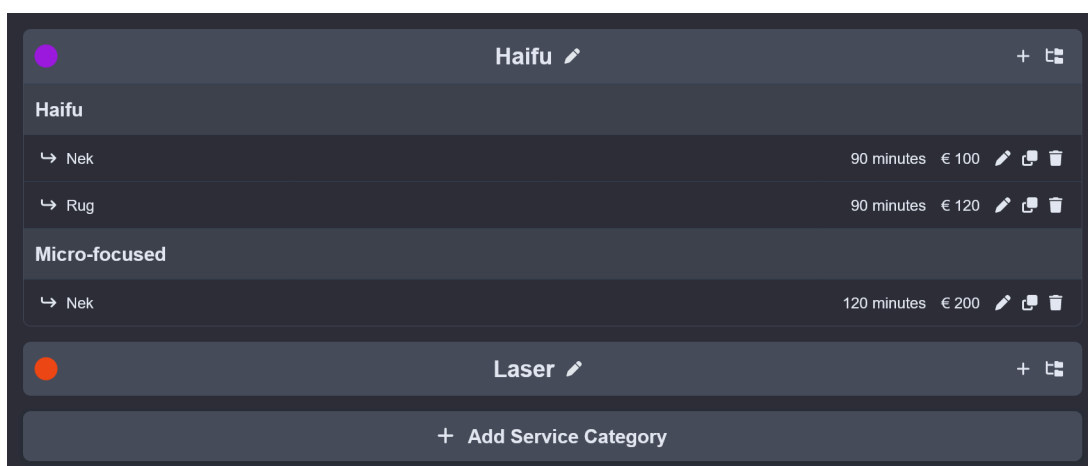
Main view



Create subcategory



Meerdere subcategorieën



Edit van dienst

×

Edit Nek

NAME

Nek

DESCRIPTION

PRICE

100.00

⌵

DURATION

90

⌵

WORKLOAD

100

⌵

SERVICECATEGORY

Haifu

SERVICSUBCATEGORY

Haifu

NEEDED EMPLOYEES

☒ Tymo Verhaegen

Save

Verwijderen van dienst

×

!

Are you sure you want to remove Nek?

This action is irreversable.

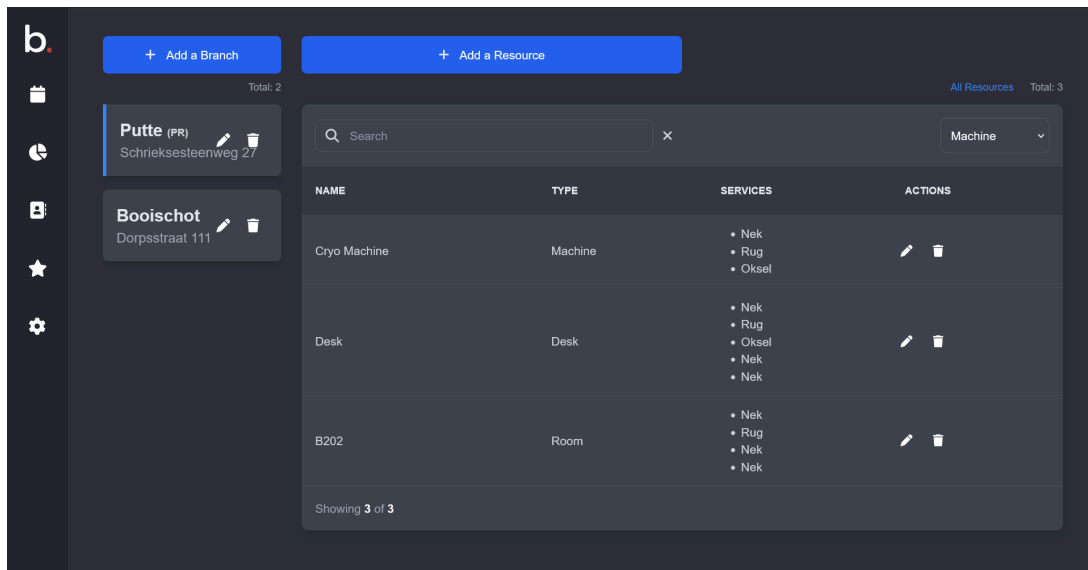
Yes, I'm sure.

Cancel

5.2.8 Middelen

Ontwikkeld door de hele groep, Thomas Verbruggen, Joppe Kerkhofs en mezelf

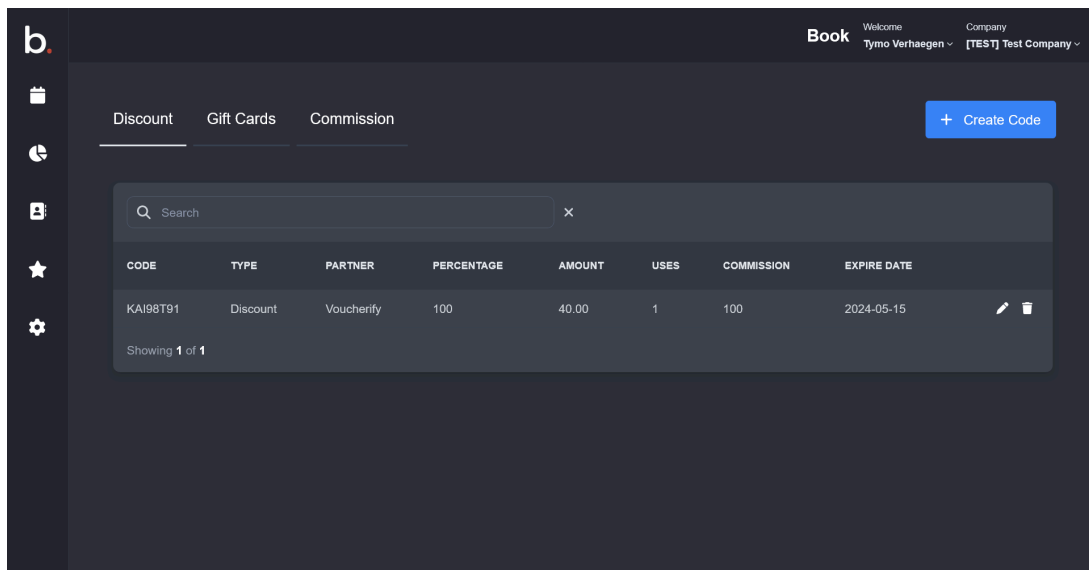
Main view



Add resource

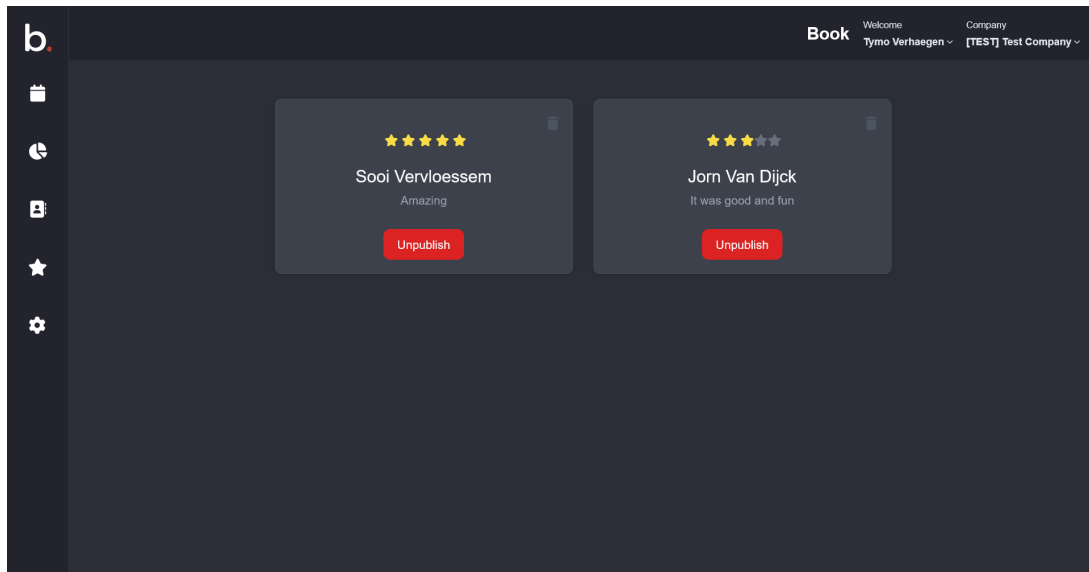
The screenshot shows the 'Add Resource' modal form. It has a title bar with the text 'Add Resource' and a close button. The form is divided into several sections: 'NAME' and 'DESCRIPTION' (with a plus icon between them), 'RESOURCE TYPE' and 'BRANCH', and 'SERVICE'. Each section has a text input field. The 'SERVICE' section has a list of checkboxes with labels: 'Nek', 'Rug', 'Oksel', 'Nek', and 'Nek'. At the bottom of the form is a blue button labeled 'Add Resource'.

Ontwikkeld door Thomas Verbruggen en mezelf



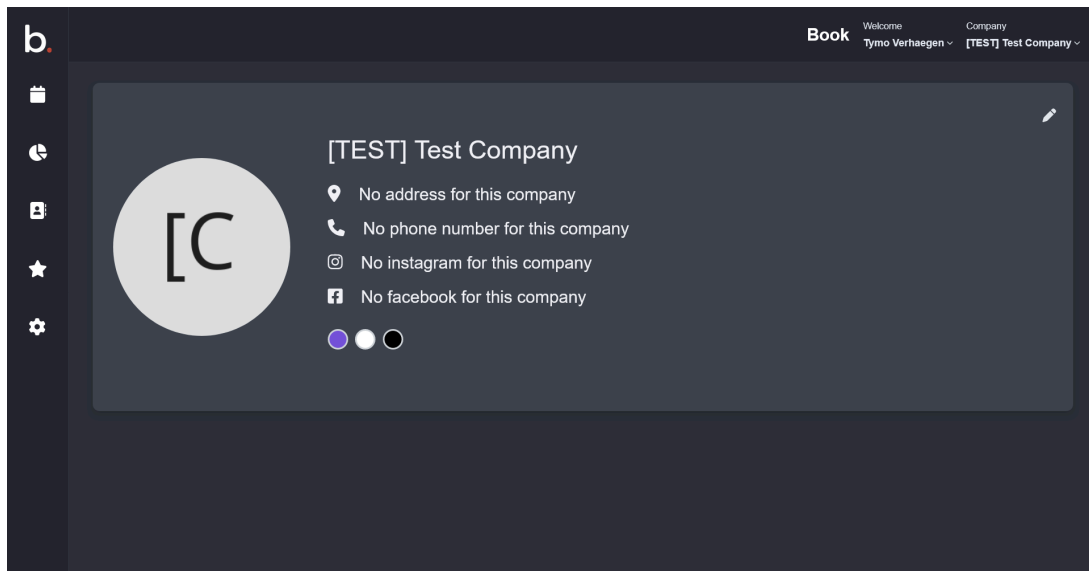
5.2.10 Reviews

Ontwikkeld door Thomas Verbruggen



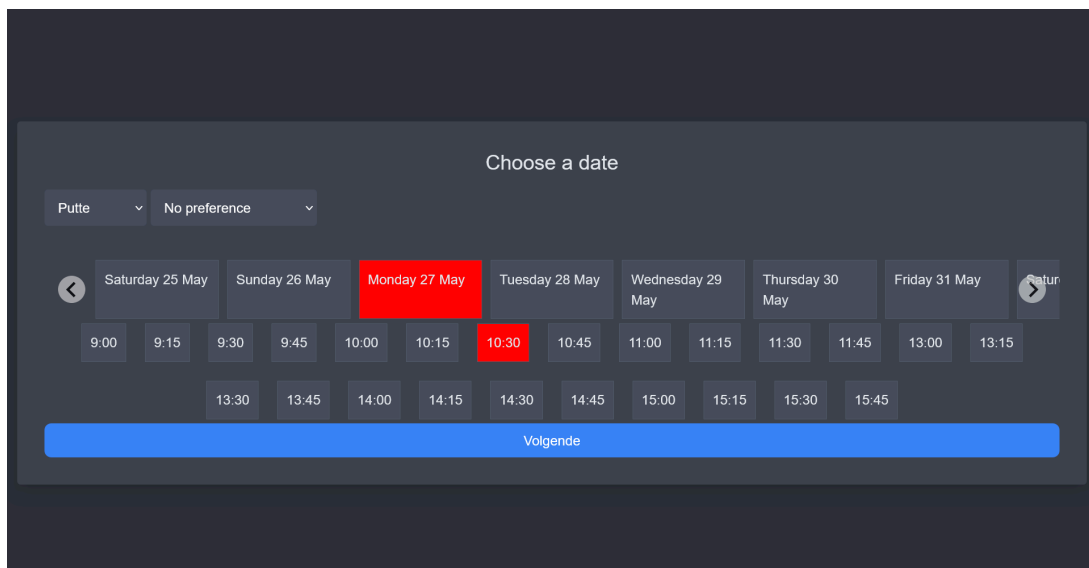
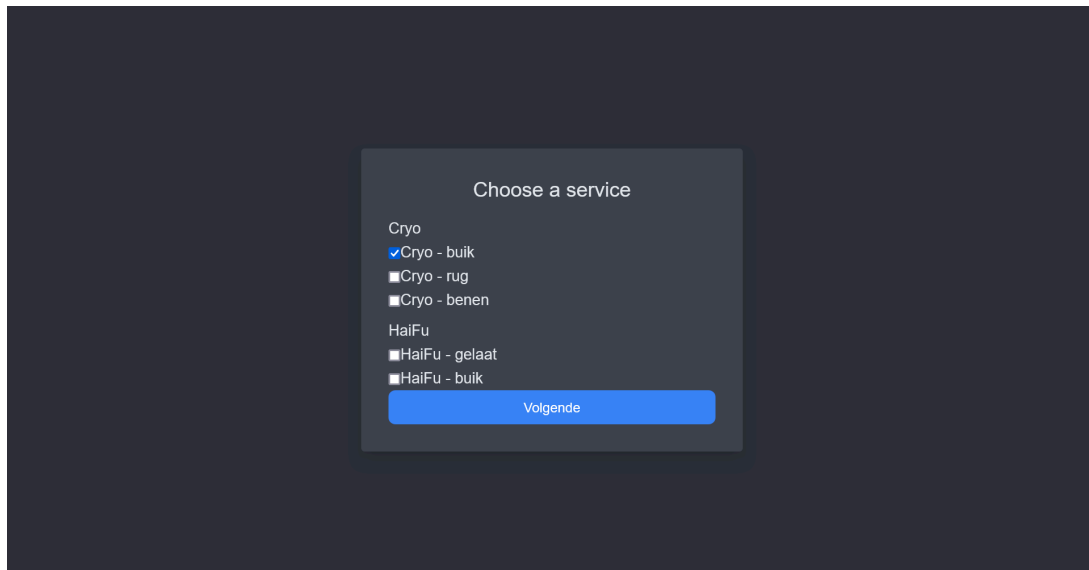
5.2.11 Bedrijfsinstellingen

Ontwikkeld door Thomas Verbruggen en mezelf



5.2.12 Booking (end-user)

Ontwikkeld door de hele groep, Thomas Verbruggen, Joppe Kerkhofs en mezelf



Persoonlijke informatie

Tes

test

test@test.com

test

Factuur E-mail

test

3000

test

Opmerking

Betalen

5.3 Backend

Het backend gedeelte van onze webapplicatie is ontworpen met Django views en alle data wordt weggeschreven en opgeslagen in een PostgreSQL database.

5.3.1 Navigatie

De navigatie wordt gecontroleerd door een urls.py file, hierin staan alle mogelijk links dat een end-user kan bezoeken

Ontwikkeld door Joppe Kerkhofs en mezelf

```
urlpatterns = [
    # Redirects to agenda page
    path('', RedirectView.as_view(url='/agenda/', permanent=False), name='home'),

    # First login page
    path('first_login/', views.FirstLoginView.as_view(), name='firstLogin'),

    # AGENDA
    path('agenda/', views.AgendaView.as_view(), name='agenda'),
    path('event/<uuid:event_id>', views.EventDetailView.as_view(), name='event_details'),
    path('update_appointment_field/<uuid:event_id>', views.update_appointment_field.as_view(), name='updateAppointment'),

    # COMPANY
    path('company/', views.CompanyView.as_view(), name='company'),
    path('company/edit/', views.EditCompanyView.as_view(), name='updatecompany'),

    # PEOPLE
    path('people/', views.PeopleView.as_view(), name='people'),
    # Employee CRUD operations
    path('people/user/info/<uuid:user_id>', views.UserInfoView.as_view(), name='updateEmployee'),
    # Partner CRUD operations
    path('people/partner/add/', views.PartnerAddView.as_view(), name='addPartner'),
    path('people/partner/info/<uuid:id>', views.PartnerView.as_view(), name='updatePartner'),
    path('people/partner/remove/<uuid:id>', views.PartnerRemoveView.as_view(), name='removePartner'),
    # Customer CRUD operations
    path('customers/', views.CustomersView.as_view(), name='customers'),
    path('customer/update/<uuid:pk>', views.UpdateCustomerView.as_view(), name='updateCustomer'),
    path('customer/remove/<uuid:pk>', views.DeleteCustomerView.as_view(), name='removeCustomer'),

    # RESOURCES
    path('resources/', views.ResourcesView.as_view(), name='resources'),
    # Branch CRUD operations
    path('resources/<uuid:branch_id>', views.EditBranchView.as_view(), name='updateBranch'),
    path('resources/', views.ResourcesView.as_view(), name='addBranch'),
```

```
path('resources/<uuid:branch_id>', views.EditBranchView.as_view(), name='updateBranch'),
path('resources/', views.ResourcesView.as_view(), name='addBranch'),
path('resources/branch/remove/<uuid:id>', views.RemoveBranchView.as_view(), name='removeBranch'),
# Resource CRUD operations
path('resources/info/<uuid:id>', views.EditResourceView.as_view(), name='updateResource'),
path('resources/', views.ResourcesView.as_view(), name='addResource'),
path('resources/remove/<uuid:id>', views.RemoveResourceView.as_view(), name='removeResource'),
path('resources/name/add', views.ResourceNameView.as_view(), name='resourceName'),
path('resources/name/update/<uuid:id>', views.UpdateResourceNameView.as_view(), name='updateResourceName'),
path('resources/name/remove/<uuid:id>', views.RemoveResourceNameView.as_view(), name='removeResourceName'),

# SERVICES
# Service CRUD operations
path('services/', views.ServicesView.as_view(), name='services'),
path('services/update/<uuid:id>', views.EditServiceView.as_view(), name='updateService'),
path('services/remove/<uuid:id>', views.DeleteServiceView.as_view(), name='removeService'),
path('services/duplicate/<uuid:serviceId>', views.ServiceDuplicateView.as_view(), name='duplicateServices'),
# Service Category
path('services/category/', views.ServiceCategoryView.as_view(), name='serviceCategory'),
path('services/subcategories/', views.ServiceSubCategoryView.as_view(), name='serviceSubCategory'),
path('services/subcategories/update/<uuid:id>', views.UpdateServiceSubCategoryView.as_view(), name='updateServiceSubCategory'),
path('services/subcategories/remove/<uuid:id>', views.RemoveServiceSubCategoryView.as_view(), name='removeServiceSubCategory'),
# Package CRUD operations
path('packages/add/', views.AddPackageView.as_view(), name='addPackage'),
path('packages/update/<uuid:id>', views.EditPackageView.as_view(), name='updatePackage'),
path('packages/remove/<uuid:id>', views.DeletePackage.as_view(), name='removePackage'),

# PROMOTIONS
path('promotions/', views.PromotionsView.as_view(), name='promotions'),
path('promotions/create/', views.CreatePromotionView.as_view(), name='createCode'),
path('promotions/update/<uuid:id>', views.EditPromotionView.as_view(), name='updateCode'),
path('promotions/remove/<uuid:id>', views.DeletePromotionView.as_view(), name='removeCode'),

# STATISTICS
path('statistics/', views.StatisticView.as_view(), name='statistics'),
```

5.3.2 Login

Ontwikkeld door Joppe Kerkhofs

FirstLoginView:

- Functionaliteit:
 - Rendert view wanneer gebruiker voor het eerst inlogt
 - Een POST-verzoek verwerkt de ingevulde gegevens van de gebruiker
 - Als het formulier geldig is
 - Controleert het of UserInfo al bestaat
 - Bij bestaande UserInfo wordt de al bestaande informatie geüpdatet
 - Bij nieuwe UserInfo wordt er een nieuwe gebruiker aangemaakt
 - Bedrijven worden geladen via de methode *load_companies()*
 - De gebruiker wordt doorgestuurd naar de agenda pagina
 - De methode *load_companies()* laadt bedrijven in de database door te controleren op CloudSpot bedrijven en deze toe te voegen aan het model *CompanyInfo*
 - Bij het ophalen van contextgegevens wordt een formulier toegevoegd aan de context

5.3.3 Agenda

Ontwikkeld door Thomas Verbruggen

AgendaView:

- Functionaliteit:
 - Rendert view voor agenda-pagina
 - Bij een GET-verzoek controleert het of de gebruiker voor het eerst inlogt met behulp van de *check_first_login()* functie
 - Het haalt werknemers voor het bedrijf op
 - Het bouwt een selectie-element voor werknemers
 - Het verzamelt afspraken en bijbehorende services uit de database en past berekeningen toe om de duur en eindtijd van afspraken te bepalen

- Het filtert afspraken op geselecteerde werknemers en bereidt contextgegevens voor om naar het template te sturen

EventDetailsView:

- Functionaliteit:
 - Deze view rendert de detailpagina van een specifiek evenement (afspraak) binnen de agenda
 - Bij een GET-verzoek haalt het de informatie op over het geselecteerde evenement en de bijbehorende klant
 - Het verzamelt informatie over de services die aan de afspraak zijn gekoppeld en berekent de totale prijs
 - Het stuurt de verzamelde gegevens door naar het template om weer te geven

update_appointment_field:

- Functionaliteit:
 - Deze view verwerkt POST-verzoeken om de status van een afspraak bij te werken, zoals 'te laat geïnformeerd' en 'niet verschenen'
 - Het controleert of de juiste velden in het verzoek zijn opgenomen en werkt de status van de afspraak bij
 - Het slaat de wijzigingen op in de database en geeft de gebruiker door naar de detailpagina

5.3.4 Statistieken & logs

Ontwikkeld door mezelf

StatisticView:

- Functionaliteit:
 - Deze view toont statistieken met betrekking tot afspraken, gebruikers, vestigingen en middelen.
 - Bij een GET-verzoek verzamelt het de benodigde gegevens voor de statistieken.
 - Het haalt het aantal gebruikers, afspraken, vestigingen en middelen op.
 - Het voert een zoekfunctionaliteit uit op afspraken op basis van de zoekquery in de URL.
 - Het werkt de status van afspraken bij op basis van de huidige tijd, waarbij afspraken als "beëindigd" worden gemarkeerd als de verwachte eindtijd is verstreken.
 - Er is pagination functionaliteit inbegrepen voor de bulk aan logs overzichtelijker te maken

5.3.5 Klanten, Werknemers en Partners

Ontwikkeld door Joppe Kerkhofs en mezelf

PeopleView:

- Functionaliteit:
 - Deze view beheert informatie met betrekking tot "mensen" of instanties zoals partners
 - Bij het ophalen van contextgegevens, haalt het gegevens op uit de database, waaronder gebruikers, klanten, bedrijven en partners
 - Het bouwt een formulier voor het toevoegen van nieuwe partners
 - Bij een POST-verzoek verwerkt het het ingediende formulier voor het maken van een nieuwe partner
 - Als het formulier geldig is, wordt de nieuwe partner opgeslagen en wordt de gebruiker doorgestuurd naar de pagina voor het beheren van mensen met een succesmelding

UserInfoView:

- Functionaliteit:
 - Deze view toont en werkt gebruikersinformatie bij
 - Bij het ophalen van contextgegevens, haalt het de gebruikersinformatie op en voegt het een formulier toe voor het bijwerken van de informatie
 - Bij een POST-verzoek verwerkt het het ingediende formulier voor het bijwerken van de gebruikersinformatie
 - Als het formulier geldig is, wordt de gebruikersinformatie bijgewerkt met de nieuwe gegevens

PartnerView:

- Functionaliteit:
 - Deze view toont en werkt partner informatie bij
 - Bij het ophalen van contextgegevens, haalt het de partner informatie op en voegt het een formulier toe voor het bijwerken van de informatie
 - Bij een POST-verzoek verwerkt het het ingediende formulier voor het bijwerken van de partner informatie
 - Als het formulier geldig is, wordt de partner informatie bijgewerkt met de nieuwe gegevens

PartnerRemoveView:

- Functionaliteit:
 - Deze view verwijdert een partner
 - Bij een POST-verzoek wordt de partner geïdentificeerd en verwijderd uit de database

PartnerAddView:

- Functionaliteit:
 - Deze view voegt een nieuwe partner toe

- Bij een geldig formulier valideert het de gegevens en maakt het een nieuwe partner aan
- Als de partner al bestaat, wordt er een melding weergegeven

CustomerView:

- Functionaliteit:
 - Deze view toont een lijst met klanten
 - Bij een GET-verzoek haalt het de klanten op die aan het bedrijf van de huidige gebruiker zijn gekoppeld
 - Het biedt functionaliteit voor het zoeken naar klanten op basis van de ingevoerde zoekquery
 - Er is pagination toegevoegd voor een grote bulk aan klanten overzichtelijker te maken

UpdateCustomerView:

- Functionaliteit:
 - Deze view werkt klantgegevens bij
 - Het gebruikt een formulier om klantgegevens te bij te werken
 - Bij een geldig formulier valideert het de gegevens en werkt het de klantgegevens bij

DeleteCustomerView:

- Functionaliteit:
 - Deze view verwijdert een klant
 - Het geeft een bevestigingspagina weer voor het verwijderen van de klant
 - Bij het bevestigen van de verwijdering wordt de klant verwijderd en wordt de gebruiker doorverwezen naar de lijst met klanten

5.3.6 Reviews

Ontwikkeld door Thomas Verbruggen

ReviewsView:

- Functionaliteit:
 - Deze view toont reviews
 - Het haalt een queryset van reviews op, gefilterd op afspraken van het bedrijf van de gebruiker

DeleteReview:

- Functionaliteit:
 - Deze view verwijdert een review.
 - Bij een POST-verzoek wordt de review verwijderd.
 - Er wordt een succesmelding weergegeven na het verwijderen van de review.

ToggleReviewPublicView:

- Functionaliteit:
 - Deze view schakelt de openbare status van een review om.
 - Bij een POST-verzoek wordt de openbare status van de review

5.3.7 Diensten & pakketten

Ontwikkeld door de hele groep, Thomas Verbruggen, Joppe Kerkhofs en mezelf

ServicesView:

- Functionaliteit:
 - Haalt gegevens uit de database en bereidt de context voor het renderen voor
 - Verwerkt POST-verzoeken om een nieuwe dienst toe te voegen
 - Voorziet formulieren voor nieuwe diensten en/of pakketten toe te voegen

ServiceCategoryView:

- Functionaliteit:
 - Beheert het maken of bijwerken van een servicecategorie
 - Verwerkt POST-verzoeken om een nieuwe servicecategorie toe te voegen of een bestaande bij te werken
 - Toont een succesbericht na succesvolle toevoeging of bijwerking van de servicecategorie

ServiceSubCategoryView:

- Functionaliteit:
 - Beheert het maken van een subcategorie voor services
 - Verwerkt POST-verzoeken om een nieuwe subcategorie toe te voegen aan een bepaalde servicecategorie
 - Controleert of de vereiste gegevens in de POST-verzoeken aanwezig zijn en voegt vervolgens de nieuwe subcategorie toe

UpdateServiceSubCategoryView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een bestaande service-subcategorie bij te werken
 - Controleert of de vereiste gegevens, zoals de subcategorie-ID en naam, aanwezig zijn in het POST-verzoek
 - Update de naam van de service-subcategorie en slaat de wijzigingen op

RemoveServiceSubCategoryView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een service-subcategorie te verwijderen
 - Controleert of de subcategorie services heeft voordat deze wordt verwijderd
 - Verwijder de service-subcategorie als deze geen services heeft

ServiceDuplicateView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een service te dupliceren
 - Maakt een nieuwe service met dezelfde gegevens als de originele service
 - Dupliceert ook de *UserAvailableService*-instanties voor de nieuwe service
-

EditServiceView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een service bij te werken
 - Update de eigenschappen van een bestaande service met de gegevens uit het formulier
 - Update ook de *UserAvailableService*-instanties voor de bijgewerkte service

DeleteServiceView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een service te verwijderen
 - Verwijdert de geselecteerde service samen met alle gerelateerde *UserAvailableService*-instanties

AddPackageView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een nieuw pakket toe te voegen
 - Maakt een nieuw pakket aan met de ingediende gegevens en koppelt geselecteerde services aan het pakket

EditPackageView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een pakket bij te werken
 - Update de eigenschappen van een bestaand pakket met de gegevens uit het formulier en werkt de bijbehorende services bij

DeletePackageView:

- Functionaliteit:
 - Verwerkt POST-verzoeken om een pakket te verwijderen
 - Verwijdert het geselecteerde pakket uit de database

5.3.8 Middelen

Ontwikkeld door de hele groep, Thomas Verbruggen, Joppe Kerkhofs en mezelf

ResourcesView:

- Functionaliteit:
 - Verwerkt zowel GET- als POST-verzoeken voor het beheren van resources
 - Haalt resources op en past filters toe op basis van zoekopdrachten, branch en resource type
 - Toont resources in paginering en biedt formulieren voor het toevoegen van nieuwe branches en resources
 - Verwerkt het toevoegen van nieuwe branches en resources, en koppelt services aan resources
 - Voorziet formulieren voor zowel resources- als branches aan te maken

EditBranchView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaand filiaal te bewerken
 - Toont een formulier voor het wijzigen van de gegevens van een filiaal, zoals naam, adres, contactgegevens, etc
 - Bij het indienen van het formulier wordt het filiaal bijgewerkt met de nieuwe gegevens en wordt de gebruiker doorgestuurd naar de middelen-pagina met een succesbericht

RemoveBranchView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaand filiaal te verwijderen
 - Verwerkt POST-verzoeken voor het verwijderen van een filiaal door het filiaal uit de database te verwijderen
 - Na succesvol verwijderen wordt de gebruiker doorgestuurd naar de middelen-pagina met een melding dat het filiaal succesvol is verwijderd

EditResourceView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaande bron te bewerken
 - Toont een formulier voor het bewerken van de gegevens van een bron, zoals naam, type, beschikbaarheid, enz
 - Verwerkt het formulier en werkt de bron bij met de nieuwe gegevens
 - Maakt het mogelijk om geselecteerde diensten voor de bron te kiezen en deze toe te voegen of te verwijderen
 - Na succesvolle bewerking wordt de gebruiker teruggestuurd naar de middelen-pagina met een succesbericht

RemoveResourceView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaande bron te verwijderen
 - Verwerkt POST-verzoeken voor het verwijderen van een bron door de bron uit de database te verwijderen
 - Na succesvol verwijderen wordt de gebruiker teruggestuurd naar de middelen-pagina met een melding dat de bron succesvol is verwijderd

ResourceNameView:

- Functionaliteit:
 - Maakt het mogelijk om een nieuwe bronnaam toe te voegen.
 - Verwerkt POST-verzoeken voor het toevoegen van een nieuwe bronnaam aan de database.
 - Controleert of de opgegeven naam al bestaat en voegt deze toe als deze uniek is.
 - Geeft een succesmelding weer als de naam succesvol is toegevoegd, en een foutmelding als de naam al bestaat of niet is opgegeven.
 - Na het toevoegen wordt de gebruiker teruggestuurd naar de pagina met middelen.

UpdateResourceNameView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaande bronnaam bij te werken.
 - Verwerkt POST-verzoeken voor het bijwerken van een bestaande bronnaam in de database.
 - Haalt de nieuwe naam op uit het verzoek en werkt de naam van de bronnaam bij.
 - Geeft een succesmelding weer na succesvol bijwerken van de naam.
 - Na het bijwerken wordt de gebruiker teruggestuurd naar de pagina met middelen.

RemoveResourceNameView:

- Functionaliteit:
 - Maakt het mogelijk om een bestaande bronnaam te verwijderen.
 - Verwerkt POST-verzoeken voor het verwijderen van een bestaande bronnaam uit de database.
 - Verwijdert de opgegeven bronnaam uit de database.
 - Geeft een succesmelding weer na succesvol verwijderen van de naam.
 - Na het verwijderen wordt de gebruiker teruggestuurd naar de pagina met middelen

5.3.9 Promoties & aanbiedingen

Ontwikkeld door Thomas Verbruggen en mezelf

PromotionsView:

- Functionaliteit:
 - Laat promoties zien op een pagina
 - Haalt alle promotiecodes op uit de database
 - Stelt de mogelijkheid in om te zoeken naar promotiecodes op basis van de zoekquery
 - Toont de lijst van promotiecodes met behulp van paginering
 - Biedt een formulier aan voor het toevoegen van nieuwe promotiecodes

CreatePromotionView:

- Functionaliteit:
 - Stelt gebruikers in staat om een nieuwe promotiecode toe te voegen
 - Verwerkt het POST-verzoek voor het maken van een nieuwe promotiecode
 - Geeft een succesmelding weer na succesvol toevoegen van de promotiecode

EditPromotionView:

- Functionaliteit:
 - Biedt gebruikers de mogelijkheid om een bestaande promotiecode te bewerken
 - Haalt de bestaande promotiecode op om deze weer te geven in het bewerkingssformulier
 - Verwerkt het POST-verzoek voor het bijwerken van een bestaande promotiecode
 - Geeft een succesmelding weer na succesvol bijwerken van de promotiecode

DeletePromotionView:

- Functionaliteit:
 - Maakt het mogelijk om een promotiecode te verwijderen
 - Verwerkt het POST-verzoek voor het verwijderen van een promotiecode uit de database
 - Geeft een succesmelding weer na succesvol verwijderen van de promotiecode

5.3.10 Bedrijfsinstellingen

Ontwikkeld door Thomas Verbruggen en mezelf

CompanyView:

- Functionaliteit:
 - Toont de bedrijfsinformatiepagina aan de gebruiker
 - Haalt de bedrijfsinformatie op van het ingelogde gebruikersaccount
 - Geeft de bedrijfsinformatie weer in de context van de pagina

EditCompanyView:

- Functionaliteit:
 - Biedt gebruikers de mogelijkheid om bedrijfsinformatie te bewerken
 - Laadt het formulier voor het bewerken van bedrijfsinformatie
 - Haalt de huidige bedrijfsinformatie op om deze weer te geven in het formulier
 - Verwerkt het POST-verzoek voor het bijwerken van de bedrijfsinformatie
 - Controleert en formatteert het telefoonnummer voordat het wordt opgeslagen
 - Geeft een succesmelding weer na succesvol bijwerken van de bedrijfsinformatie
 - Leidt gebruikers om naar de bedrijfsinformatiepagina na succesvol bijwerken

5.3.11 Booking (end-user)

Ontwikkeld door de hele groep, Thomas Verbruggen, Joppe Kerkhofs en mezelf

addServiceToAppointment:

- Functionaliteit:
 - Controleert of de opgegeven afspraak en dienst bestaan
 - Controleert of er al een AppointmentService met dezelfde dienst is
 - Controleert of de nieuwe dienst past bij de reeds gekoppelde diensten van de afspraak
 - Creëert een nieuwe AppointmentService indien bovenstaande controles slagen
 - Retournt de gecreëerde AppointmentService

endAppointment:

- Functionaliteit:
 - Markeert een afspraak als beëindigd
 - Retournt de bijgewerkte afspraak

checkWorkloadInTimeFrame:

- Functionaliteit:
 - Berekent de werklast van een medewerker in een opgegeven tijdsbestek
 - Verzamelt alle afspraken en de bijbehorende diensten van de medewerker
 - Berekening van de werklast gebaseerd op de duur en de werklast van de diensten
 - Retournt de totale werklast

checkIfResourceAvailable:

- Functionaliteit:
 - Controleert of een resource beschikbaar is binnen een opgegeven tijdsbestek
 - Verzamelt afspraken binnen dat tijdsbestek
 - Controleert of de resource wordt gebruikt door een AppointmentService binnen dat tijdsbestek
 - Retournt True als de resource beschikbaar is, anders False

checkIfResourceAvailable:

- Functionaliteit:
 - Controleert of de nieuwe dienst past bij de reeds gekoppelde diensten van een afspraak
 - Vergelijkt de categorieën en subcategorieën van de diensten
 - Retournt True als de nieuwe dienst past, anders False

6 CONCLUSIE

Tijdens mijn stage bij LHS heb ik waardevolle inzichten opgedaan over het creëren van websites en webapplicaties. Ik heb ook een dieper begrip ontwikkeld van het hele proces, beginnende bij analyse en eindigende bij een volledig geïmplementeerd project.

Ik heb ook geleerd hoe het is om te werken in een bedrijf en hoe hier talloze dingen kunnen mislopen, denk maar aan miscommunicatie of misinterpretatie van onderwerpen en zaken. Dit is iets dat heel verschillend was tegenover schoolprojecten en iets waar ik echt mee moest leren omgaan.

Een van de belangrijkste zaken voor het project was ook zelfstandig werken, aangezien LHS een relatief klein bedrijf is, was dit iets dat ons van in het begin al duidelijk was gemaakt. Ik denk uiteindelijk dat dit goed is gegaan en dat we fier mogen zijn op het project dat we hebben neer gezet.

Al bij al heeft mijn stage bij LHS me niet alleen technische vaardigheden en kennis opgeleverd maar ook mijn voorgaande vaardigheden verfijnd. Denk maar aan professionele communicatie en deadline-management, dit zijn zaken dat op school aangeleerd worden maar die voor mij echt verfijnd zijn tijdens de stageperiode. Deze ervaringen zullen van waarde zijn voor mijn toekomstige carrière.

7 GENERATIVE AI POLICY

Tijdens de stage is er gebruikgemaakt van generatieve AI tools zoals ChatGPT, Gemini en Phind.

Deze hebben geholpen tijdens struikelblokken die tijdens de projectontwikkeling kwamen opduiken, denk maar aan errors oplossen, debuggen, Ook was het handig om deze te gebruiken voor de analytische fase, deze tools hielpen bijvoorbeeld bij dilemma's die we tegenkwamen bij het designen van de database of bij het brainstormen over wireframes.

Door deze tools te gebruiken konden we efficiënter werken en grotere sprongen maken in de ontwikkeling van het project. Het heeft ons ook geholpen om nieuwe vaardigheden te ontwikkelen en onze kennis te vergroten door directe ondersteuning en begeleiding te bieden bij het oplossen van problemen.

Voor het maken van mijn realisatie document heb ik ook gebruikgemaakt van generatieve AI, bijvoorbeeld voor de terminologie of voor het verbeteren van spellingsfouten en grammaticale fouten.

Kortom, generatieve AI was een geweldig hulpmiddel dat me veel heeft geholpen maar ook veel heeft bijgeleerd tijdens deze stageperiode.