

Booking Tool Meeting Rooms

Realisatiedocument

Tom Bulen
Student Bachelor in de Toegepaste Informatica – Applicatieontwikkeling

Inhoudsopgave

1. INLEIDING	3
1.1. Stagebedrijf: we+ Benelux	3
1.2. Probleemstelling & Opdracht	4
2. ANALYSE	5
2.1. Planning	5
2.2. Vergelijking van bestaande oplossingen	7
2.3. Gebruikte technologieën	9
2.4. Microsoft Multi-Tenant Research	11
2.5. Prototypes	12
3. APPLICATIE	14
3.1. Ruimtes	14
3.2. Kantoren	16
3.3. Kenmerken	19
3.4. Auditlogboeken	21
3.5. Outlook	22
3.6. Database	24
3.7. Werking van de applicatie	26
3.8. Hosting	27
4. PROJECTMANAGEMENT	28
5. BESLUIT	29

1. Inleiding

In dit document beschrijf ik de uitwerking van mijn stageopdracht bij we+ Benelux. Mijn project is de "Booking tool voor meeting rooms". Er wordt verdiept in de aanpak die ik heb gevolgd om dit probleem op te lossen. Ook wordt er in dit document grondig omschreven welke resultaten er zijn behaald van het project en de uitdagingen die ik ben tegengekomen gaandeweg. Daarnaast wordt er ook gereflecteerd op mijn stageperiode en de vaardigheden die ik heb opgedaan tijdens deze periode.

Wat ik verwacht van deze stage periode is dat ik meer ervaring ga op doen met de programmeer taal Java en Angular. Ook zou ik graag willen dat ik meer ervaring ga op doen in verband met Outlook agenda en Microsoft. Ik zie deze stageperiode als een mogelijkheid om mijn kennis uit te breiden over deze topics en mijn professioneel netwerk uit te breiden.

1.1. Stagebedrijf: we+ Benelux

Het bedrijf waar ik mijn stage heb gedaan is we+ Benelux, dat een IT-consultancybedrijf is, wat gespecialiseerd is in technologische oplossingen en digitale transformatie. Het is ontstaan in september 2024 en bestaat uit drie ondernemingen namelijk Jidoka, Digitalum en Continuum die een sterke reputatie hebben in IT-projectontwikkeling, e-commerce en softwareontwikkeling, wat leidde tot een zeer krachtige samenwerking van diensten en expertise.

we+ Benelux is deel van het wereldwijde Alan Allman Associates-ecosysteem (AAA) en biedt een breed assortiment aan diensten, waaronder training, consulting, managed services, end-to-end projectlevering en strategie en advies. De expertisegebieden van we+ Benelux omvatten klantervaring, infrastructuur en cloudoplossingen, projectbeheer, applicatieontwikkeling, maar ook data en innovatie.

Het leveren van op maat gemaakte oplossingen die bedrijven helpen bij het optimaliseren en transformeren van hun processen is de richtpijl van we+ Benelux. Zowel in Europa als in Azië wil het bedrijf de aanbidding van technische, strategische en operationele verbeteringen aan de klanten toegankelijk maken. Dit doen ze door middel van een team dat bestaat uit meer dan tweehonderd experts en de aanwezigheid in meerdere locaties over heel de wereld.

we+ Benelux investeert veel in het creëren van een gebalanceerde werkomgeving, coaching en trainingen voor werknemers om er voor te zorgen dat hun persoonlijke en professionele groei bevorderd wordt. Daarom hecht het bedrijf ook veel waarde aan een cultuur waarin betrokkenheid en vertrouwen de twee belangrijkste elementen zijn. Zo willen ze de werknemers aanmoedigen om hun motivatie en passie in te zetten voor het leveren van kwaliteitsvolle diensten aan de klanten van het bedrijf.

1.2. Probleemstelling & Opdracht

Als een werknemer van het eerste bedrijf een reservering maakt via de Outlook agenda om een ruimte te boeken met het huidige systeem dit alleen zichtbaar is voor de andere werknemers van het eerste bedrijf. Hier door krijgen de werknemers van het tweede bedrijf de boeking niet te zien. Dit leidt tot het risico van dubbele reservaties van dezelfde ruimte. Door dit gebrek aan transparantie tussen de twee bedrijven ontstaat er een inefficiënt gebruik van de beschikbare ruimtes en verwarring.

Ook hebben de bedrijven geen centrale koppeling tussen de Microsoft 365-omgevingen. Ieder bedrijf moet zelf handmatig dezelfde ruimtes aan hun eigen Microsoft 365-omgeving toevoegen. Dat is dus heel foutgevoelig en tijdrovend. Er is geen transparante en uniforme manier om de beschikbaarheid van de verschillende ruimtes over al de bedrijven te beheren door dit probleem.

Er wordt voorgesteld om een centrale boeking tool te koppelen aan Outlook agenda om er zo voor te zorgen dat er geen dubbele boekingen en geen gebrek aan overzicht is tussen verschillende bedrijven. De actuele beschikbaarheid van vergaderruimtes kunnen door alle gebruikers van alle bedrijven bekeken worden door deze koppeling. Hierdoor voorkomen we misverstanden en wordt het reserveren veel gemakkelijker. Ook gaat het samenwerken tussen bedrijven een stuk soepeler en duidelijker.

Via hun Outlook agenda kunnen gebruikers vergaderruimtes reserveren. Relevante metadata zoals datum, tijd en gebruikersnaam worden automatisch doorgestuurd naar de boeking tool tijdens het maken van een reservering. Het is heel belangrijk dat gebruikers voor dat ze reserveren kunnen zien als er een beschikbare ruimte is. Het maakt niet uit door welk bedrijf de reservering gemaakt is, iedere werknemer van ieder bedrijf ziet deze reservering. Hierdoor gaat er een gedeelde zichtbaarheid over de beschikbaarheid van de verschillende ruimtes ontstaan.

Door de beschikking van beheerders over een aparte managementinterface waarmee ze vergaderruimtes kunnen beheren kunnen ze ruimtes aanpassen, verwijderen of aanmaken. Deze beheerders beheren ook belangrijke metadata zoals de naam, het aantal zitplaatsen, locatie en de aanwezigheid van voorzieningen zoals grote schermen of videoconferentie apparatuur.

De beschikbaarheid van ruimtes worden automatisch en bedrijfs overstijgend gesynchroniseerd door de integratie met Outlook agenda. Iedereen beschikt over dezelfde informatie en dubbele boekingen worden voorkomen doordat de reserveringen realtime worden verwerkt.

Alle wijzigingen en reserveringen worden overzichtelijk bijgehouden door dat de boeking tool een audit logging-functionaliteit bevat. Het terugkijken van de reserveringsgeschiedenis wordt hierdoor mogelijk gemaakt als een analyse nodig is. Ook verhoogt dit de transparantie.

2. Analyse

Hier wordt er een overzicht gegeven van al mijn voorbereidingen die gedaan zijn voordat het project is uitgevoerd. Er zijn verschillende technologieën en tools onderzocht. Ook heb ik veel afwegingen gemaakt en heb ik deze keuzes met behulp van een Weighted Decision Matrix (WDM) onderbouwd. Ik heb ondersteunend beeldmateriaal per onderdeel samen met een korte toelichting.

2.1. Planning

Aan het begin van mijn stage in de onderzoeksfase heb ik een globale planning opgesteld om een duidelijk overzicht te geven van mijn taken op mijn stage. Deze planning diende als houvast voor zowel mezelf als mijn stagebegeleiders binnen het bedrijf. Omdat het bedrijf volgens de Agile methode werkt, kon ik mijn planning makkelijk aanpassen waar nodig.

Hieronder vind je een overzicht van alle planningen die ik tijdens het project heb opgesteld.

Eerste planning:

Planning 2025															
Naam 	Week van 	24 feb	3 mrt	10 mrt	17 mrt	24 mrt	31 mrt	7 apr	14 apr	21 apr	28 apr	5 mei	12 mei	19 mei	26 mei
Onderzoeksfase															
UC Diagram & ERD/User stories															
Prototype															
Beheerder ruimtes beheren															
Unit & integratie tests															
Outlook agenda ruimtes syncen															
Outlook agenda data uithalen															
Outlook agenda ruimtes reserveren															
Beheerder inloggen (Microsoft)															
Applicatie testing															
Oplevering															
Documentatie															

Tweede planning:

Deze planning heb ik opgesteld kort na het eerste terugkommoment van school. Gedurende deze periode is er veel veranderd. Er ontstonden problemen bij het verkrijgen van een Microsoft 365-licentie en bijbehorende omgeving, omdat dit geregeld moest worden via het moederbedrijf, AAA. Zij gaven aan dat dit nog even kon duren, waardoor ik besloot om het Microsoft-gedeelte voorlopig uit te stellen. In plaats daarvan ben ik begonnen met het lokaal opzetten van het project op mijn eigen computer. In die fase ontving ik weinig feedback vanuit het bedrijf, omdat mijn mentor niet veel tijd had.

Toen ik een deel van het project had afgerond, heb ik gevraagd voor uitgebreidere feedback op mijn code. Omdat ik nog steeds moest wachten op de Microsoftomgeving, kreeg ik in de tussentijd per onderdeel van mijn code gerichte feedback over wat verbeterd kon worden en wat al goed was opgezet. In deze tweede fase kreeg ik ondersteuning van twee extra mentors, waardoor het proces sneller en gestructureerd verliep. Dit zorgde er wel voor dat met name de testfase en de fase rond het beheren van beheerdersruimtes meer tijd in beslag namen dan oorspronkelijk gepland.

Planning 2025															
Naam 	Week van 	24 feb	3 mrt	10 mrt	17 mrt	24 mrt	31 mrt	7 apr	14 apr	21 apr	28 apr	5 mei	12 mei	19 mei	26 mei
Onderzoeksfase															
UC Diagram & ERD/User stories															
Prototype															
Beheerder ruimtes beheren															
Unit & integratie tests															
Outlook agenda ruimtes syncen															
Outlook agenda data uithalen															
Outlook agenda ruimtes reserveren															
Beheerder inloggen (Microsoft)															
Applicatie testing															
Oplevering															
Documentatie															

Derde planning:

Na het tweede terugkommoment op school heb ik de planning opnieuw aangepast. Dit was nodig omdat we in de week van 28 april tegen problemen aanliepen bij het synchroniseren van ruimtes tussen de verschillende bedrijven. Dit bleek meer tijd te kosten dan verwacht. Daarom hebben we de hele planning met een week verschoven. Daarnaast hebben we besloten om de functionaliteit voor het inloggen en uitloggen uit het project te halen voor de rest van de stageperiode.

Planning 2025															
Naam 	Week van 	24 feb	3 mrt	10 mrt	17 mrt	24 mrt	31 mrt	7 apr	14 apr	21 apr	28 apr	5 mei	12 mei	19 mei	26 mei
Onderzoeksfase															
UC Diagram & ERD/User stories															
Prototype															
Beheerder ruimtes beheren															
Unit & integratie tests															
Outlook agenda ruimtes syncen															
Outlook agenda data uithalen															
Outlook agenda ruimtes reserveren															
Applicatie testing															
Oplevering															
Documentatie															

2.2. Vergelijking van bestaande oplossingen

Als eerste heb ik onderzocht welke tools er al bestonden. Ik heb 11 tools gevonden en opgezocht wat de voor- en nadelen, prijzen en de functies zijn van deze tools. Dat heb ik gedaan om ideeën op te doen voor functies die handig zouden kunnen zijn als ik later zelf een tool zou maken. Ook wilde ik weten of er al bestaande tools waren waarop ik eventueel verder kon bouwen.

Nadat ik alle informatie verzameld had, heb ik een Weighted Decision Matrix (WDM) opgesteld waarin ik deze 11 tools heb vergeleken. Ik heb ook een eigen ontwikkelde tool als optie toegevoegd om te kunnen vergelijken hoe die zou scoren in vergelijking met de andere bestaande oplossingen.

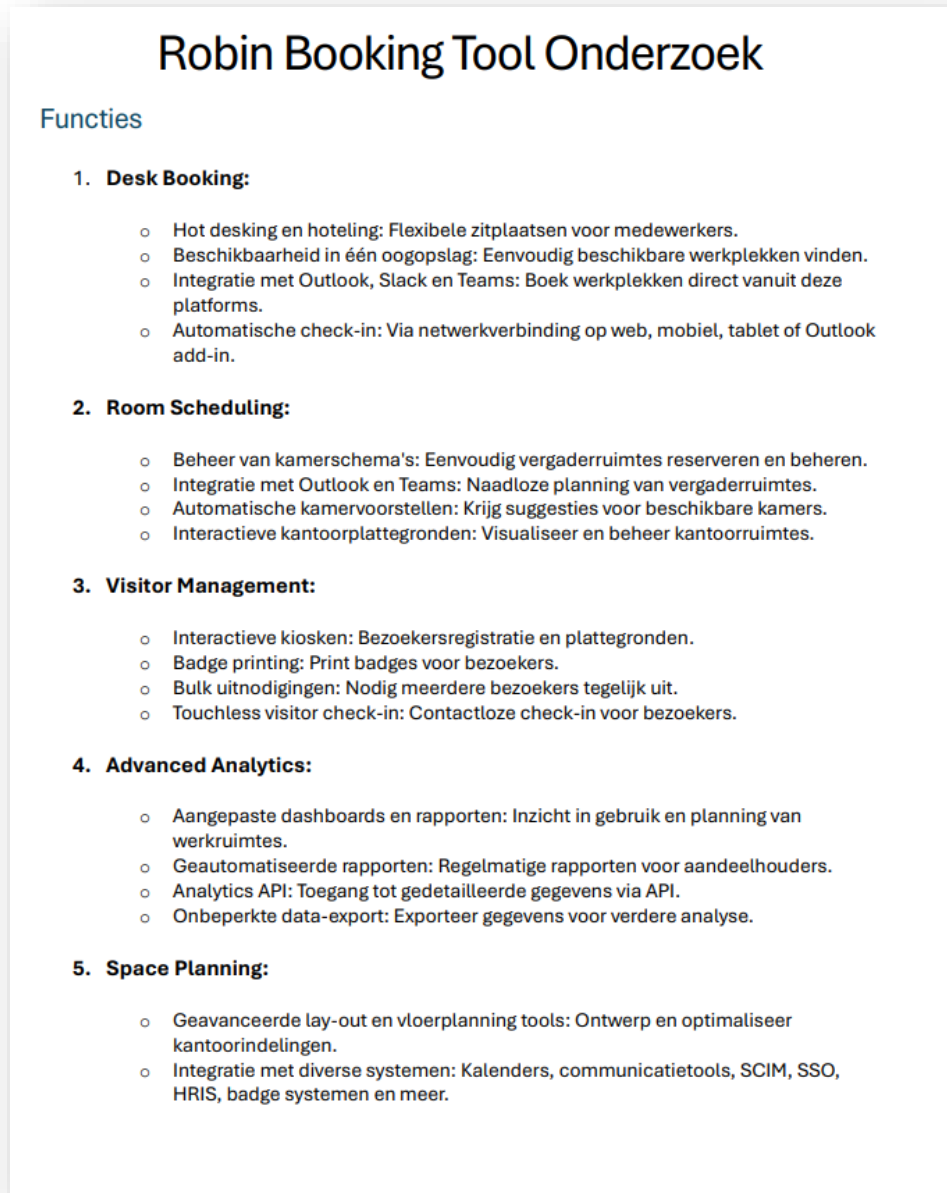
Een WDM is een hulpmiddel waarmee je verschillende opties objectief kunt vergelijken. Je kiest eerst de belangrijkste factoren en geeft die een weging (bijvoorbeeld: hoe belangrijk is de Outlook-integratie?). Vervolgens geef je elke tool een score per factor. Die score vermenigvuldigt je met het gewicht en je telt alles op om tot een totaalscore per tool te komen.

Hieronder zie je de WDM die ik heb opgesteld met de scores per tool:

Weighted Decision Matrix						
	Outlook-integratie	Gebruiksvriendelijkheid	Functionaliteiten	Audit Logging	Aanpasbaarheid	
Factor	5	4	5	3	4	
Optie						Totaalscore
Microsoft Bookings	3	3	3	2	3	60
Robin	4	4	4	3	4	81
Condeco	4	2	5	3	3	74
Reflex RoomManager	4	3	4	3	3	73
GoBright	4	2	4	3	4	73
Avesqo	4	3	3	2	3	65
SpacelQ	4	3	4	3	3	73
Roomzilla	3	3	3	3	3	63
Teem	4	3	4	3	4	77
Skedda	4	3	4	3	3	73
Envoy	4	4	3	3	4	76
Maatwerkoplossing	5	4	5	3	4	91

Uit de matrix bleek dat een maatwerkoplossing het beste scoorde. Op de tweede plaats kwam Robin, een bestaande tool die veel functionaliteit biedt. Daarom heb ik een document opgesteld over Robin, waarin ik alle functies, voordelen en nadelen heb uitgelegd, en ook heb toegelicht waarom we uiteindelijk toch kiezen voor het bouwen van een eigen applicatie.

Hieronder staat een fragment uit mijn onderzoek document over Robin:



Dit document heb ik gedeeld met mijn stagebedrijf zodat zij ook een duidelijk overzicht hadden van de mogelijkheden. Ze waren erg enthousiast over mijn analyse en vonden het nuttige input voor de volgende stappen in het project.

2.3. Gebruikte technologieën

Tijdens mijn stageperiode heb ik veel verschillende technologieën gebruikt. Hier vertel ik welke ik heb gebruikt voor het maken van mijn applicatie, zowel voor de frontend (wat de gebruiker ziet) als de backend (achter de schermen), en waarom ik voor deze technologieën heb gekozen.

ANGULAR

Het bedrijf gebruikt zowel Angular als React. Omdat ik vóór mijn stage al wat ervaring had met React, stelde het bedrijf voor dat ik met Angular zou werken. Zo kon ik iets nieuws leren en meer ervaring opdoen met een ander framework. Dus dit was een keuze van het bedrijf. De keuze was ook gemaakt zodat ik mijn kennis kon uitbreiden tijdens mijn stageperiode.

TYPESCRIPT

Ik heb samen met het bedrijf gekozen voor TypeScript voor de ontwikkeling van de frontend van mijn applicatie. Ook hangt het samen met Angular. Het is een uitbreiding van JavaScript die statische types toevoegt. Deze types helpen om eerder in het ontwikkelingsproces fouten te vinden. Ze helpen ook met het onderhoud en de leesbaarheid van de code.

Door de duidelijke typeaanduidingen en betere foutdetectie verbetert de kwaliteit van de code. Hierdoor kozen we voor TypeScript. Dit was vooral belangrijk voor het project, omdat het de kans op bugs verkleinde. Ook maakt TypeScript het gemakkelijker om grotere en complexere applicaties te beheren, wat goed aansloot bij de vereisten van de applicatie.

TAILWIND CSS

Ik heb gekozen om te werken met Tailwind CSS in plaats van standaard CSS of SASS, omdat ik er al veel ervaring mee had. Tailwind is een utility-first CSS-framework, wat betekent dat je met kant-en-klare klassen snel de opmaak van je pagina kunt regelen zonder veel zelfgeschreven CSS.

Ik vind Tailwind makkelijk om mee te werken, vooral omdat ik de meeste klassen al goed ken. Daardoor kon ik sneller werken en mijn opmaak precies zo maken als ik wilde, zonder tijd te verliezen aan extra opmaak of stylingbestanden.

JAVA SPRING BOOT MET GRADLE

Voor de backend van mijn applicatie heb ik tijdens mijn stage gewerkt met Java en Spring Boot. Spring Boot maakt het makkelijk om snel een backend op te zetten met veel handige functies al ingebouwd.

In het begin gebruikte ik Maven om mijn project te bouwen en afhankelijkheden te beheren, maar het bedrijf werkte met Gradle. Ze hadden ook eigen instellingen en configuraties gemaakt voor Gradle, dus ben ik overgestapt om beter aan te sluiten bij hun werkwijze. Hierdoor werkte alles zoals het bedrijf gewend was.

FLYWAY

Ik heb Flyway gebruikt voor database versiebeheer. Met Flyway kon ik alle veranderingen aan de database bijhouden in de vorm van versiebestanden (SQL-scripts).

Het bedrijf koos hiervoor zodat elke ontwikkelaar of server altijd met dezelfde versie van de database werkt. Flyway voerde automatisch de juiste scripts uit bij het opstarten van de applicatie, waardoor updates aan de database gestructureerd en foutloos verliepen. Hierdoor kon ik veilig en gecontroleerd wijzigingen aanbrengen in de database zonder dat er handmatig iets aangepast hoefde te worden.

MICROSOFT GRAPH API

Ik heb de Microsoft Graph API gebruikt om te communiceren met Outlook. Daar haalde ik alleen kalenderdata uit om zo te kunnen zien wanneer er een vergadering werd georganiseerd en in welke ruimte. Ik had deze data nodig omdat ik alle vergaderingen moest synchroniseren met andere bedrijven. Dit was de enige manier dat ik data uit outlook kon halen.

POWERSHELL

Ik heb PowerShell gebruikt om scripts te maken waarmee ruimtes automatisch konden worden aangemaakt in Outlook via Microsoft Exchange. Dit was nodig omdat de Microsoft Graph API dit niet ondersteunde.

De scripts bevatten specifieke commando's om ruimtes toe te voegen. Ik kon deze scripts gebruiken vanuit mijn Java-backend wanneer dat nodig was. PowerShell bracht geen problemen bij het in productie brengen van de applicatie, omdat het platformafhankelijk is en ook werkt op Linux en macOS. Daardoor kon ik de oplossing makkelijk inzetten in verschillende omgevingen.

MICROSOFT EXCHANGE (VIA POWERSHELL)

Ik heb Microsoft Exchange gebruikt om ruimtes toe te voegen aan Outlook. Dit was nodig omdat de Microsoft Graph API geen ondersteuning biedt om ruimtes via code aan te maken.

Het bedrijf wilde dat dit automatisch kon gebeuren. Daarom heb ik een oplossing gebouwd in PowerShell die gebruikmaakt van Exchange om dit proces te automatiseren.

POSTGRESQL

PostgreSQL is een database die ik gebruik heb voor mijn applicatie. Dit was een keuze van het bedrijf, omdat deze database betrouwbaar is en goed omgaat met veel data. Het is een krachtige relationele database die snel gegevens kan opslaan en ophalen, dit is goed voor als we later met veel data gaan werken.

DOCKER

Ik heb Docker gebruik om een deel van mijn ontwikkelomgeving te vereenvoudigen. Het bedrijf heeft hier voor gekozen zodat iedereen met dezelfde instellingen te werk kan gaan. Hierdoor werden installatieproblemen voorkomen. Doordat ik Docker Compose gebruikte kon ik in een keer zowel een PostgreSQL-database als een Adminer, wat een eenvoudige databasebeheerder is, opstarten. Daardoor hoefde ik deze onderdelen niet handmatig te installeren en kon ik snel aan de slag. Adminer gaf me een handige webinterface om de database-inhoud te bekijken en te bewerken tijdens het testen. Docker zorgde ervoor dat alles overal op dezelfde manier werkte, ongeacht op welke computer ik ontwikkelde.

NGROK

Ik heb Ngrok gebruikt om een tijdelijke, openbare link te maken naar mijn applicatie op mijn eigen computer. Dit had ik nodig om een webhook van Microsoft Rooms te laten werken, die gebruikt werd voor het ophalen van kalendergegevens. Omdat die webhook alleen werkt met een publiek bereikbare URL, bood Ngrok een snelle en makkelijke oplossing zonder dat ik iets op een echte server hoefde te zetten. Zo kon ik mijn applicatie lokaal testen terwijl de webhook gewoon verbinding kon maken.

2.4. Microsoft Multi-Tenant Research

Voor ik aan de backend begon, heb ik eerst een document gemaakt om duidelijkheid te krijgen hoe multi-tenancy in Microsoft werkt, vooral rond Room Finder en Outlook. Dit hielp me beter te snappen hoe ik tenants moet beheren en hoe ik met de Microsoft Graph API kan werken. Ik heb dit document gemaakt om te beginnen met Java met Spring Boot en Gradle, zodat ik een duidelijk technisch vertrekpunt had.

Hieronder staan een aantal fragmenten van dat document.

Multi-Tenant Room Finder Setup Using Java (Spring Boot + Gradle)

Overview

This guide outlines how to configure Microsoft Room Finder for multiple tenants in Outlook programmatically. It utilizes **Java (Spring Boot + Gradle)** and the **Microsoft Graph API**, specifically leveraging the `/places` endpoint, to manage room mailboxes and room lists at scale.

1. Key Concepts

- **Room Mailboxes:**
 - Special mailboxes representing physical meeting spaces (e.g., conference rooms).
 - Used for scheduling and resource management in Outlook.
- **Room Lists:**
 - Distribution groups that organize room mailboxes (e.g., by location, floor, or type).
 - Essential for efficient filtering in Room Finder.
- **Multi-Tenant Applications:**
 - A single app interacts with multiple Azure AD tenants.
 - Each tenant consents to the app accessing its resources.

2. Prerequisites

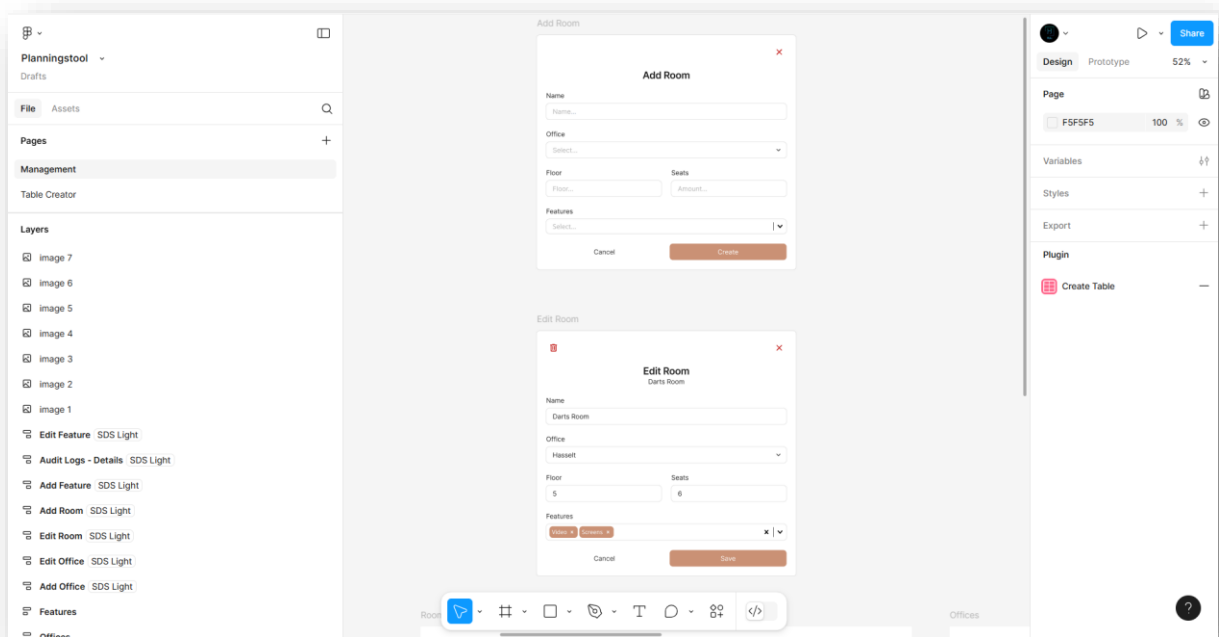
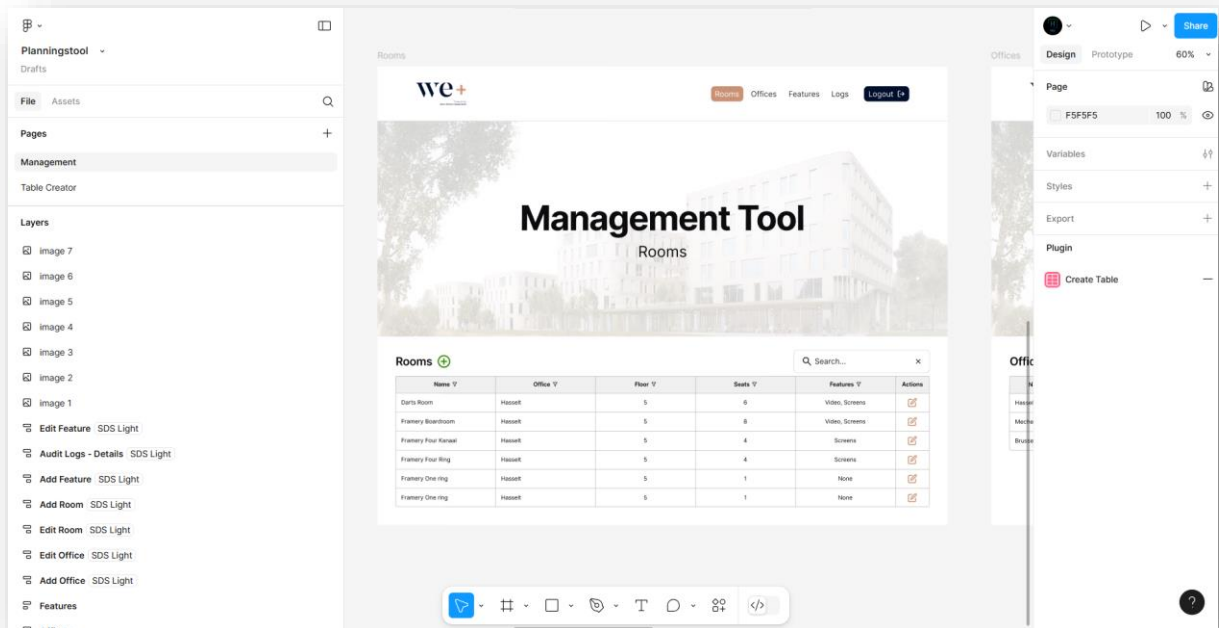
1. **Azure AD Multi-Tenant Application:**
 - Register your app in Microsoft Entra ID (formerly Azure AD).
 - Set it to support multi-tenancy: "Accounts in any organizational directory (multi-tenant)".
 - Enable user consent for tenants.
2. **API Permissions:**
 - Grant permissions to your app for:
 - `Places.ReadWrite.All`
 - `Directory.ReadWrite.All`
 - `Calendars.ReadWrite`
3. Create a **Client Secret** under **Certificates & Secrets** to authenticate your application securely.
4. **Java Development Environment:**
 - Use **Spring Boot** for application structure.
 - Use **Gradle** for dependency management.

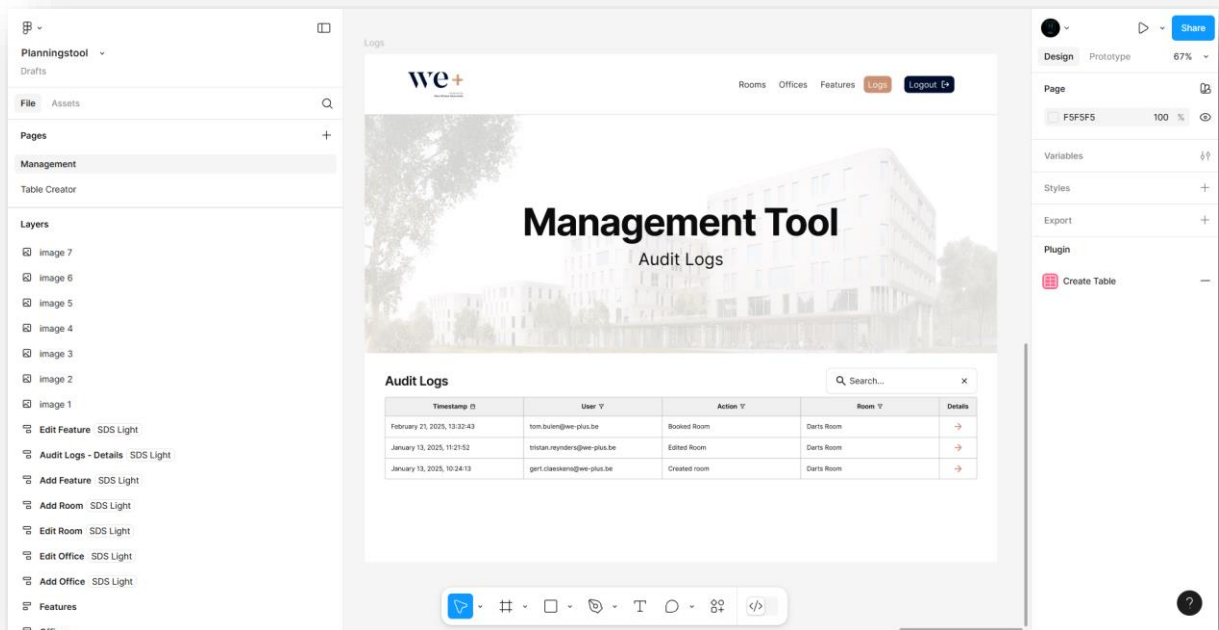
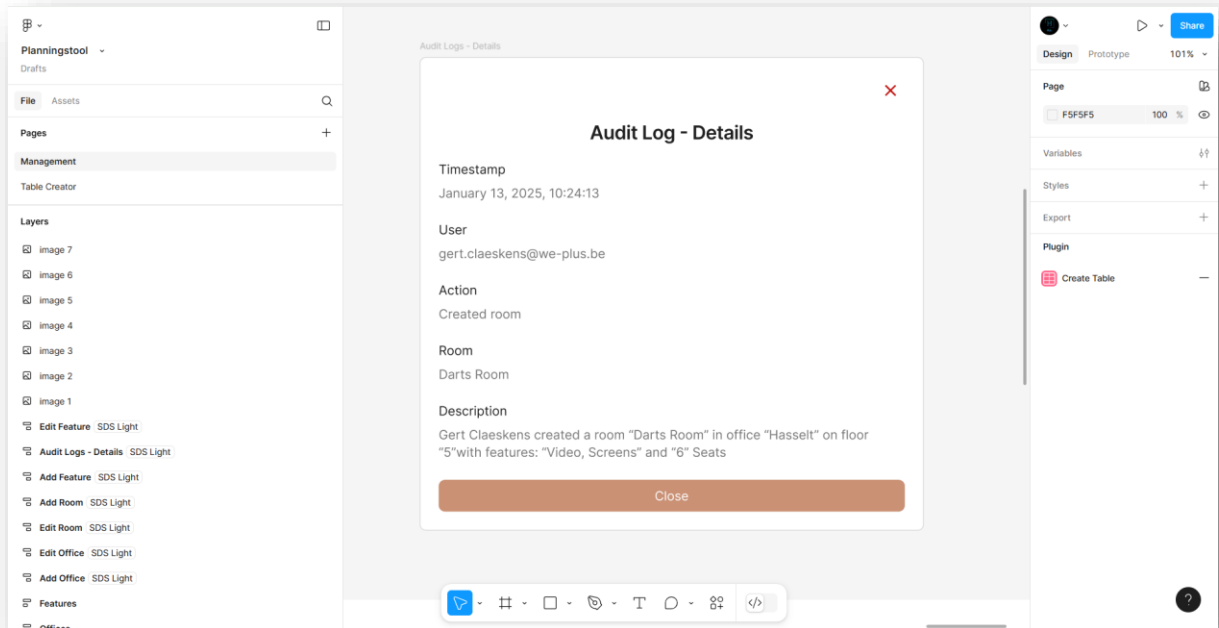
2.5. Prototypes

Voor het uitwerken van mijn prototype heb ik Figma gebruikt. Dat is een handige online tool waarmee je makkelijk een ontwerp kunt maken zonder al meteen te hoeven coderen. Omdat ik er al ervaring mee had en het ook binnen het bedrijf gebruikt wordt, was dat een logische keuze voor mij.

Toen ik klaar was met het ontwerp, heb ik het laten zien aan mijn stagementor. Hij vond het er goed uitzien en was positief verrast. Daarna kon ik aan de slag met de bouw van het project.

Hieronder zie je een paar screenshots van mijn prototypes:



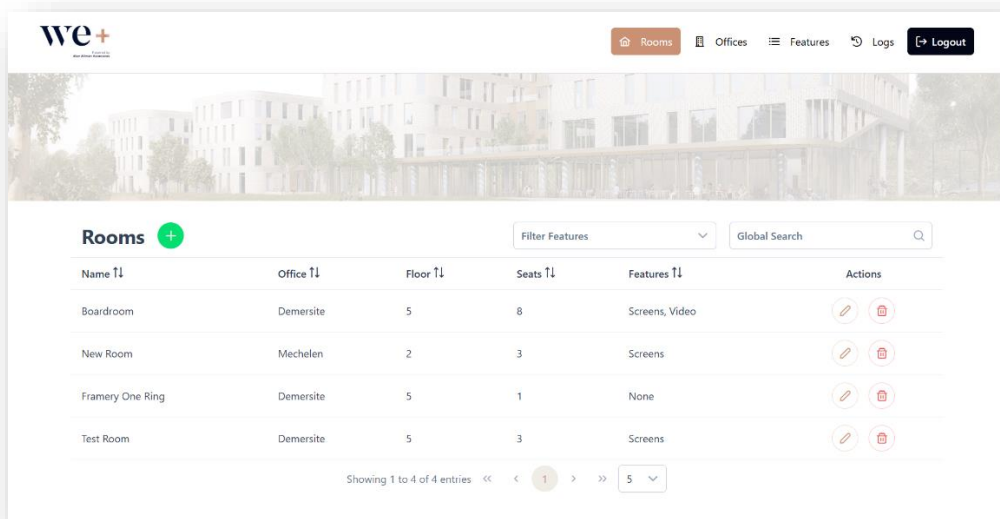
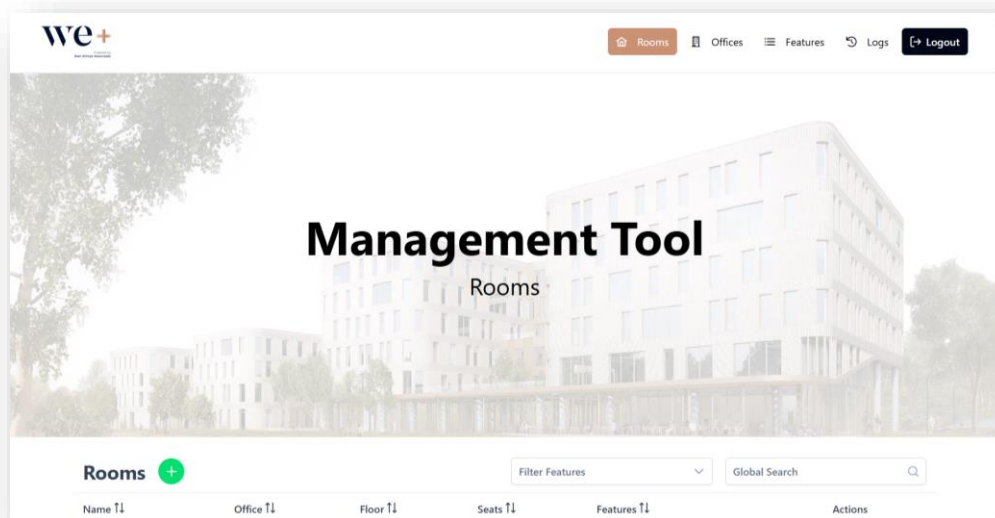


3. Applicatie

In dit onderdeel leg ik uit hoe de applicatie werkt en welke functionaliteiten er zijn. Voor elke functionaliteit wordt uitgelegd wat deze doet, ondersteund met één of meerdere afbeeldingen. De volledige applicatie is ook mobiel vriendelijk en dus ook goed bruikbaar op smartphones en tablets.

3.1. Ruimtes

Op deze pagina kan de administrator ruimtes binnen het systeem bekijken, filteren op kenmerken zoals naam, verdieping, aantal zitplaatsen of aanwezige apparatuur (zoals schermen of videoapparatuur), en sorteren op alfabet of getallen. Zo kunnen ruimtes snel gevonden worden op basis van specifieke wensen of criteria.



RUIMTES AANMAKEN

Bij het aanmaken van een nieuwe ruimte kan de administrator de naam, de aanwezige apparatuur en het aantal zitplaatsen invullen. Ze kunnen ook kiezen op welk verdiep en in welk kantoor de ruimte is. Let op: de naam van de ruimte is permanent, omdat deze wordt gebruikt om automatisch e-mailadressen te genereren. Microsoft biedt geen manier om deze later aan te passen, dus de naam moet zorgvuldig gekozen worden. Na het aanmaken verschijnt de ruimte in het overzicht op de Ruimtes-pagina. Ook worden deze ruimtes, zodra ze zijn aangemaakt, toegevoegd aan elk bedrijf dat gekoppeld is met de applicatie en automatisch gesynchroniseerd met hun Outlook-agenda.

The screenshot shows the 'Add Room' modal form. It has a title bar with a close button (X). The form contains the following fields: 'Name' (required, with a warning that it is permanent and cannot be changed later), 'Office' (dropdown), 'Floor' (number input), 'Seats' (number input), and 'Features' (optional dropdown). At the bottom are 'Cancel' and '+ Create' buttons. The background shows a blurred view of the 'Rooms' list page.

RUIMTES AANPASSEN

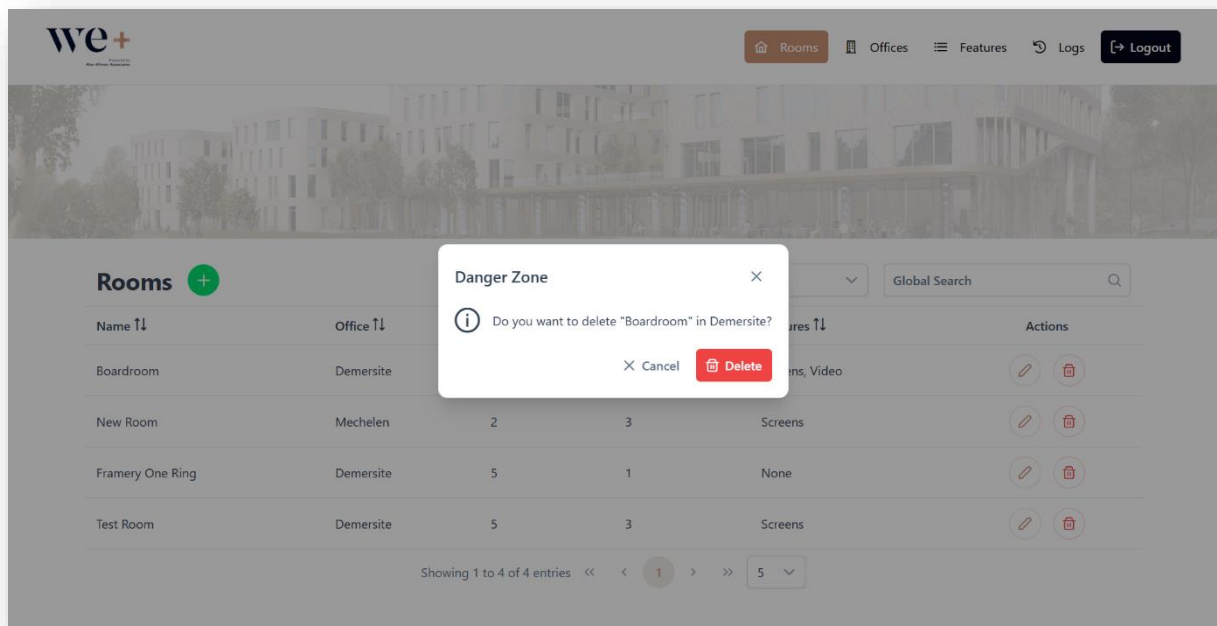
Bij het aanpassen van een ruimte kunnen gebruikers de naam van de ruimte niet wijzigen, deze is permanent bij het aanmaken van een ruimte. Ze kunnen echter wel alle andere gegevens aanpassen, zoals het aantal zitplaatsen, de aanwezige apparatuur en de locatie (kantoor en verdieping). Alle wijzigingen worden meteen doorgevoerd in de applicatie, en ook gesynchroniseerd met elk gekoppeld bedrijf en hun Outlook-agenda.

The screenshot shows the 'Edit Room: Boardroom in Demersite' modal form. The 'Name' field is disabled and labeled '(Name is read-only)'. Other fields include 'Office' (dropdown), 'Floor' (number input), 'Seats' (number input), and 'Features' (optional dropdown). At the bottom are 'Cancel' and 'Save' buttons. The background shows a blurred view of the 'Rooms' list page.

The screenshot shows the 'Edit Room: Boardroom in Demersite' modal form on a mobile phone. The layout is adapted for a smaller screen, with the 'Name' field at the top, followed by 'Office', 'Floor', 'Seats', and 'Features'. The 'Save' button is at the bottom right. The background shows a blurred view of the 'Rooms' list page.

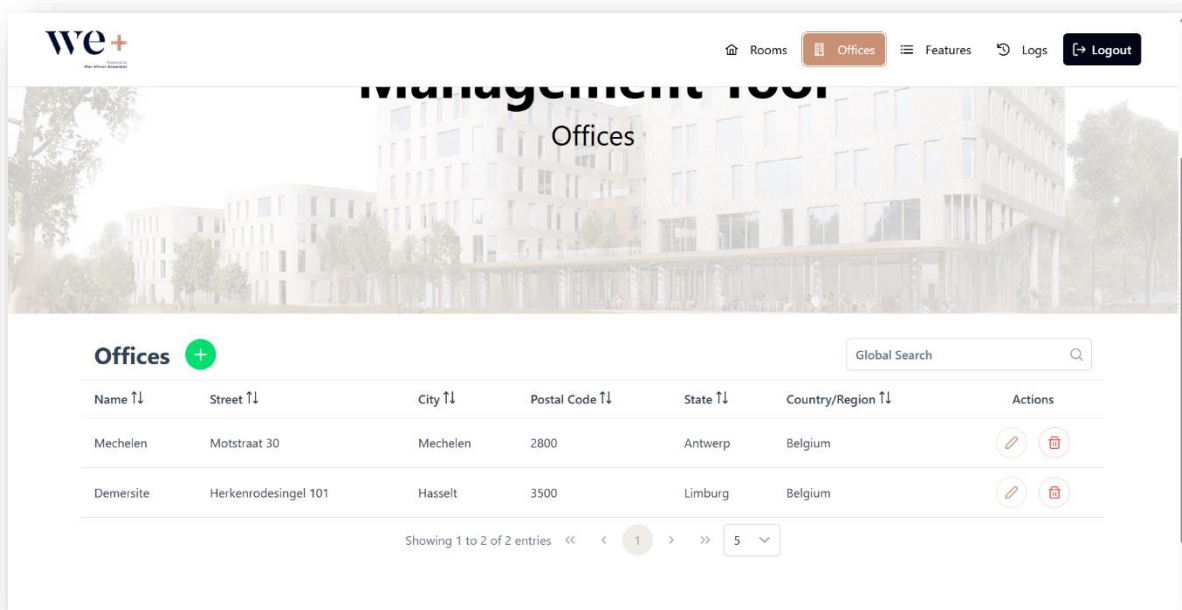
RUIMTES VERWIJDEREN

Ruimtes kunnen eenvoudig verwijderd worden via een knop in het overzicht. Bij het verwijderen verschijnt een pop-up met de vraag: "Do you want to delete '<NAAM>' in <KANTOOR>?" Na bevestiging wordt de ruimte verwijderd uit de applicatie, bij alle gekoppelde bedrijven, en uit hun Outlook-agenda.



3.2. Kantoren

Op deze pagina kan de administrator kantoren binnen het systeem bekijken, filteren op kenmerken zoals naam, stad, postcode, of land/regio, en sorteren op alfabet of andere velden. Zo kunnen kantoren snel teruggevonden worden op basis van specifieke gegevens.



KANTOREN AANMAKEN

Bij het aanmaken van een nieuw kantoor kan de administrator gegevens invullen zoals de naam, straat, stad, postcode, staat en land/regio. Na het aanmaken verschijnt het kantoor in het overzicht op de Kantoren-pagina. Kantoren worden gebruikt om ruimtes aan een locatie te koppelen, maar worden zelf niet gesynchroniseerd met Outlook, omdat Outlook alles bijhoudt via de ruimtes.

The screenshot shows the 'Add Office' form in the we+ management tool. The form is a modal window with the following fields:

- Name ***: Text input field.
- Street ***: Text input field.
- City ***: Text input field.
- Postal Code ***: Text input field.
- State ***: Text input field.
- Country/Region ***: Text input field.

At the bottom of the form are two buttons: 'Cancel' and '+ Create'. The background shows a blurred view of the 'Offices' table with columns 'Name' and 'Street'.

KANTOREN AANPASSEN

Bij het aanpassen van een kantoor kunnen gebruikers alle gegevens wijzigen, zoals de naam, straat, stad, postcode, staat en land/regio. De aanpassingen worden direct doorgevoerd in de applicatie. Omdat kantoren niet rechtstreeks gekoppeld zijn aan Outlook, vindt er geen synchronisatie van het kantoor zelf plaats met de agenda's van gekoppelde bedrijven. Wel worden de adresgegevens en het kantoor van de ruimtes die aan dit kantoor gekoppeld zijn automatisch bijgewerkt, ook in Outlook.

The image shows two screenshots of the 'Edit Office' form in the we+ management tool. The desktop screenshot on the left shows the form with the following fields:

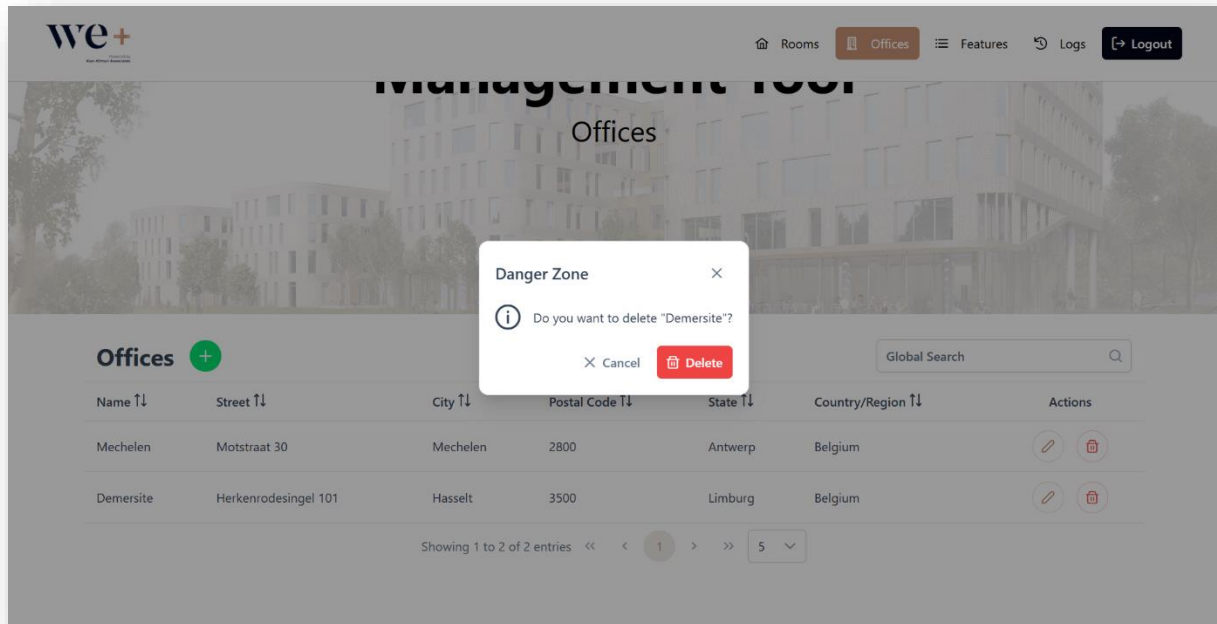
- Name ***: Text input field with value 'Demersite'.
- Street ***: Text input field with value 'Herkenrodesingel 101'.
- City ***: Text input field with value 'Hasselt'.
- Postal Code ***: Text input field with value '3500'.
- State ***: Text input field with value 'Limburg'.
- Country/Region ***: Text input field with value 'Belgium'.

At the bottom of the form are two buttons: 'Cancel' and 'Save'. The background shows a blurred view of the 'Offices' table with columns 'Name' and 'Street'.

The mobile screenshot on the right shows the same form on a smartphone screen. The fields and values are identical to the desktop version.

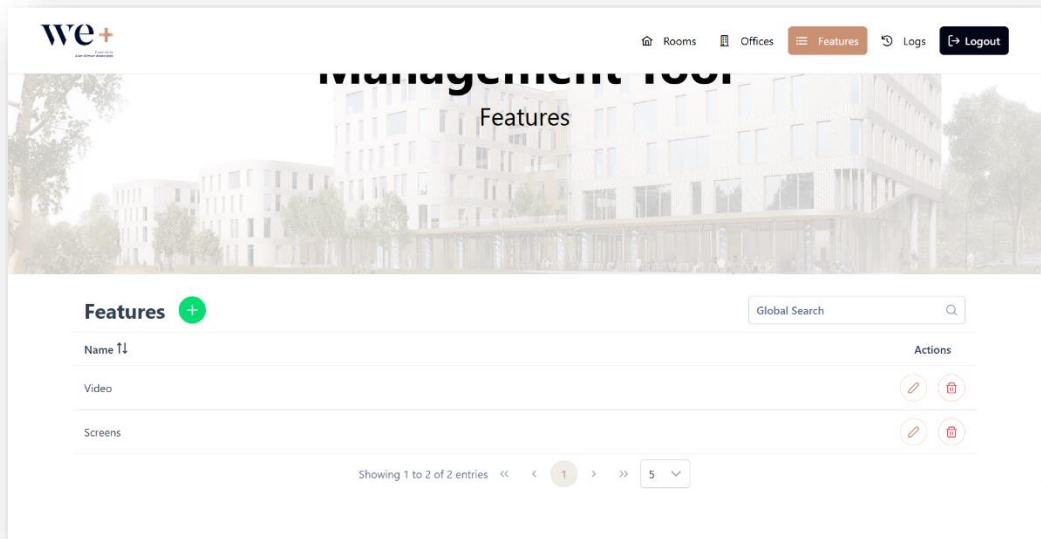
KANTOREN VERWIJDEREN

Kantoren kunnen eenvoudig verwijderd worden via een knop in het overzicht. Bij het verwijderen verschijnt een pop-up met de vraag: "Do you want to delete '<NAAM>'?" Na bevestiging wordt het kantoor verwijderd uit de applicatie. Ruimtes die aan het kantoor gekoppeld zijn, worden automatisch mee verwijderd en verdwijnen ook uit Outlook.



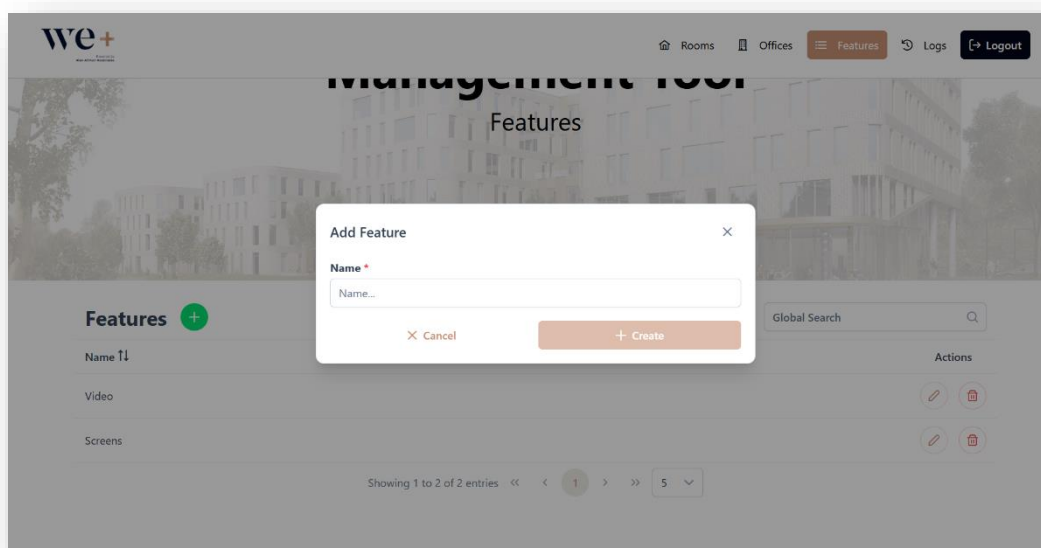
3.3. Kenmerken

Op deze pagina kan de administrator kenmerken (features) beheren die aan ruimtes gekoppeld kunnen worden, zoals "schermen" of "videoapparatuur". De kenmerken worden weergegeven in een overzicht waar eenvoudig gezocht en gesorteerd kan worden op naam. Zo kunnen bestaande kenmerken snel gevonden en beheerd worden. Nieuwe kenmerken kunnen worden toegevoegd en bestaande kunnen aangepast of verwijderd worden.



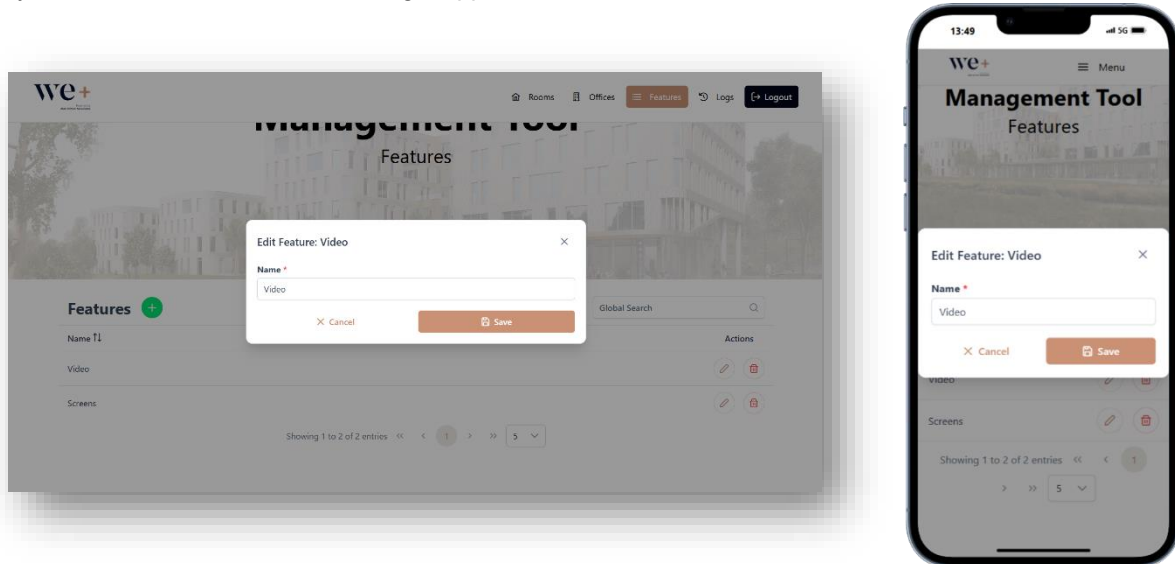
KENMERKEN AANMAKEN

Bij het aanmaken van een nieuw kenmerk (feature) kan de administrator een naam opgeven, zoals "scherm", "whiteboard" of "videocall-apparatuur". Na het aanmaken verschijnt het kenmerk in het overzicht op de Kenmerken-pagina. Kenmerken kunnen vervolgens gekoppeld worden aan ruimtes, zodat gebruikers gericht kunnen filteren op aanwezige voorzieningen. Deze kenmerken worden samen met de ruimte gesynchroniseerd met Outlook, omdat ze deel uitmaken van de informatie die aan een ruimte is verbonden.



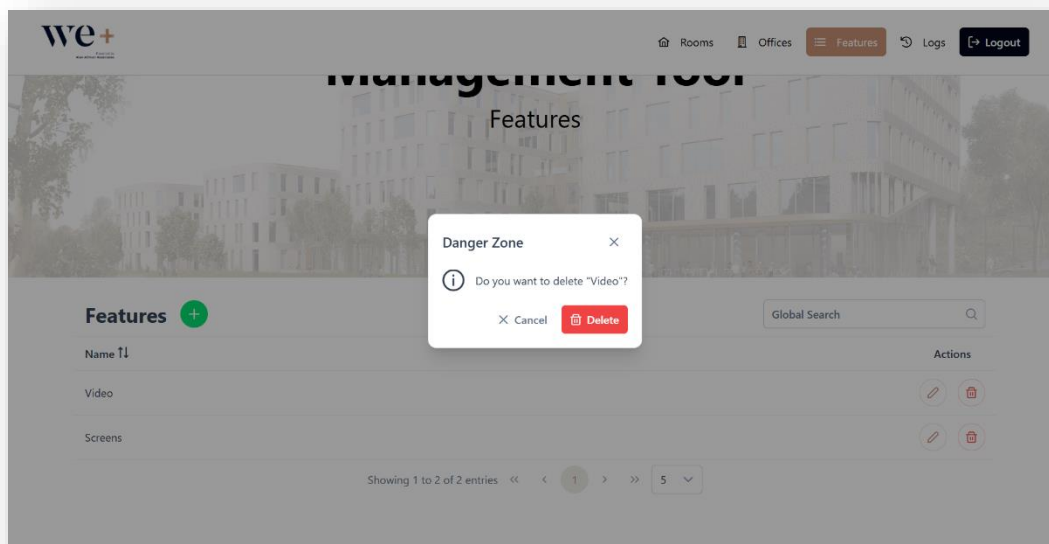
KENMERKEN AANPASSEN

Bij het aanpassen van een kenmerk kan de administrator de naam van het kenmerk wijzigen. De wijziging wordt direct doorgevoerd in de applicatie en automatisch bijgewerkt bij alle ruimtes die dit kenmerk gebruiken. Omdat kenmerken onderdeel zijn van de ruimtegegevens, worden deze updates ook gesynchroniseerd met Outlook via de gekoppelde ruimtes.



KENMERKEN VERWIJDEREN

Kenmerken kunnen eenvoudig verwijderd worden via een knop in het overzicht. Als je op de vuilbak drukt verschijnt een pop-up met de vraag: "Do you want to delete '<NAAM>'?" Na bevestiging wordt het kenmerk verwijderd uit de applicatie. Alle ruimtes die dit kenmerk hadden, verliezen automatisch de koppeling met het verwijderde kenmerk, maar blijven verder gewoon bestaan. De wijziging wordt ook doorgevoerd in Outlook via de gekoppelde ruimtes.



3.4. Auditlogboeken

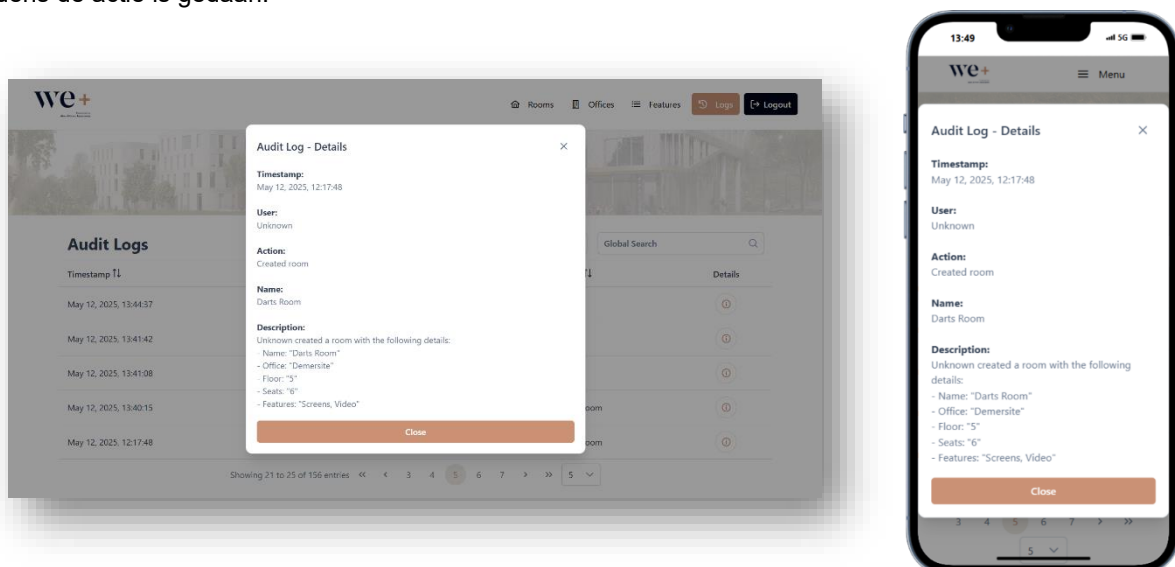
Op deze pagina kan de administrator audit logs bekijken van alle create-, update- en delete-acties die uitgevoerd zijn op ruimtes, kantoren en kenmerken. De logs worden weergegeven in een overzicht waar eenvoudig gefilterd kan worden op datums en gesorteerd kan worden op bijvoorbeeld datum of type bewerking. Zo kunnen specifieke gebeurtenissen snel teruggevonden worden. Via een infoknop kan extra informatie opgevraagd worden over elke bewerking. Omdat er nog geen login is gekoppeld aan de applicatie, wordt de uitvoerder van de actie momenteel weergegeven als unknown behalve als de actie vanuit Outlook komt. Dit komt doordat ze wel in Outlook zijn ingelogd.

Timestamp ↑↓	User ↑↓	Action ↑↓	Name ↑↓	Details
May 12, 2025, 14:25:00	Unknown	Deleted room	ABC Room	ⓘ
May 12, 2025, 14:05:57	Unknown	Deleted feature	VR Utils	ⓘ
May 12, 2025, 14:05:21	Unknown	Updated feature	VR	ⓘ
May 12, 2025, 14:04:32	Unknown	Updated feature	VR Utils	ⓘ
May 12, 2025, 14:04:18	Unknown	Updated feature	VR	ⓘ

Showing 1 to 5 of 156 entries << 1 2 3 4 5 >> 5

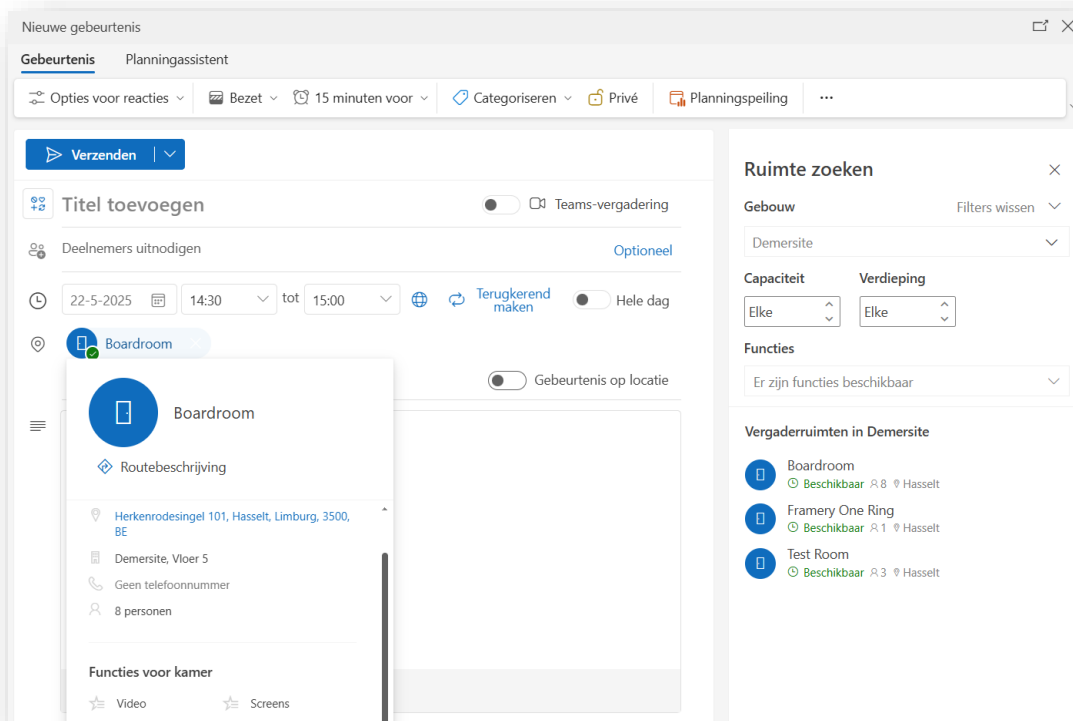
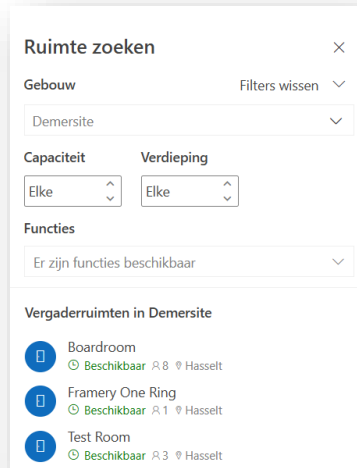
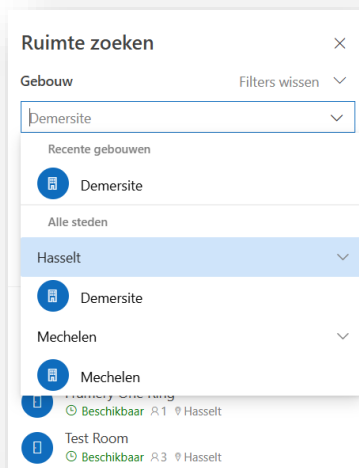
DETAILS BEKIJKEN

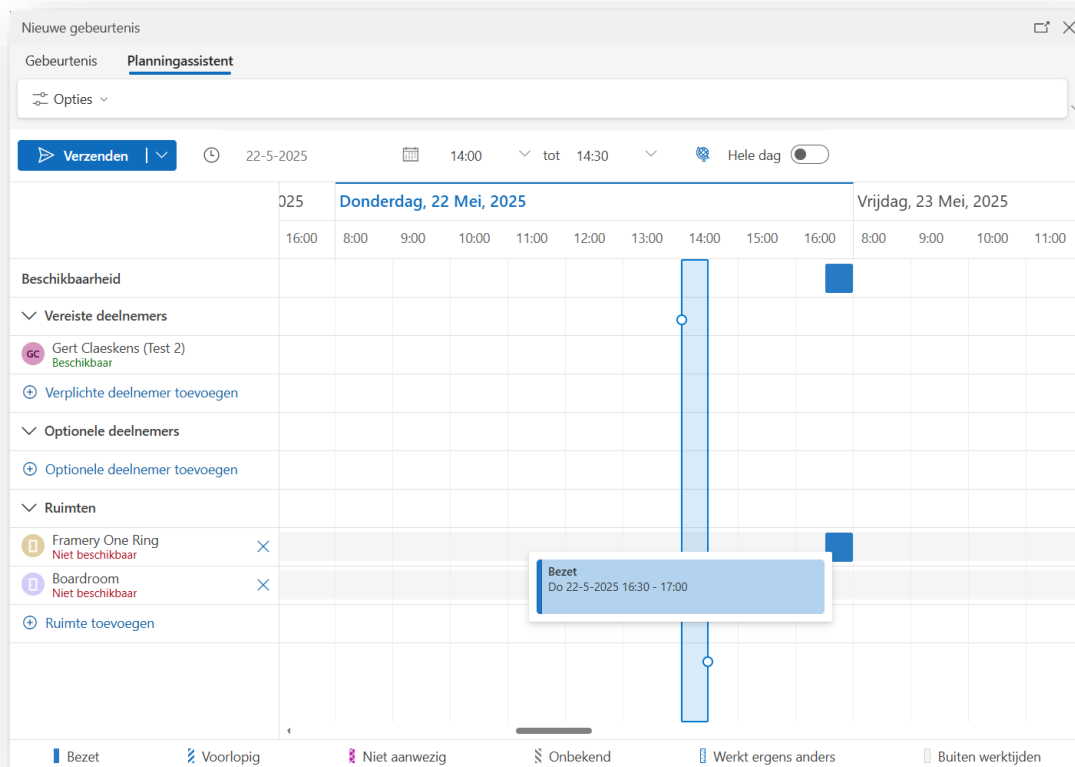
Op deze pagina ziet de administrator gedetailleerde informatie over de gekozen actie. Er wordt uitgelegd wat er precies is aangepast, toegevoegd of verwijderd, wanneer dit is gebeurd en op welk onderdeel (ruimte, kantoor of kenmerk) de actie betrekking heeft. Ook is er een duidelijke beschrijving van wat er tijdens de actie is gedaan.



3.5. Outlook

De applicatie werkt samen met Outlook. Via Outlook kunnen gebruikers ruimtes boeken met de Room Finder. Zo vinden ze gemakkelijk een vrije ruimte. Nieuwe ruimtes worden automatisch aan Outlook toegevoegd. Dit kan soms tot 24 uur duren voordat ze zichtbaar zijn. Als een afspraak verandert, bijvoorbeeld de tijd of de ruimte, past de applicatie dit automatisch aan in de agenda's van alle bedrijven. Alle ruimtes en de bijbehorende afspraken worden ook automatisch gesynchroniseerd tussen alle gekoppelde bedrijven. Zo blijft alles up-to-date.

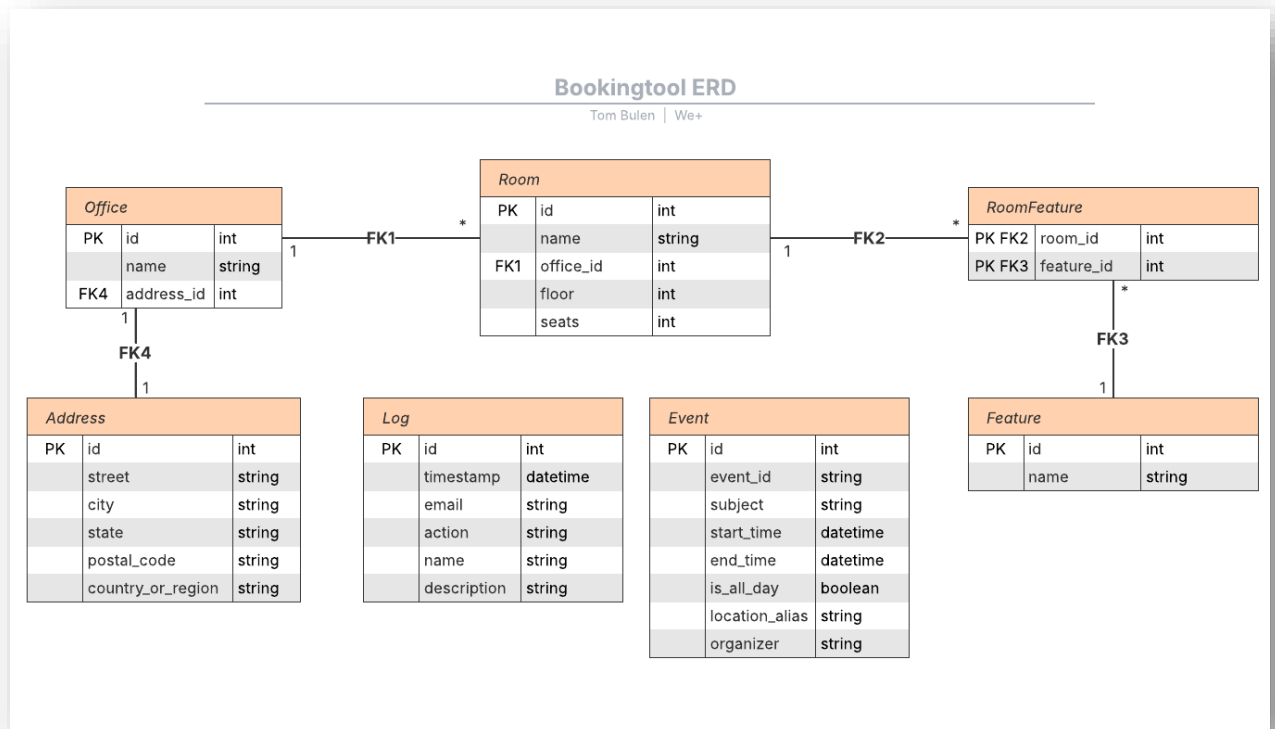




3.6. Database

De applicatie gebruikt PostgreSQL om alle gegevens op te slaan. Denk aan ruimtes, kantoren, kenmerken, vergaderingen en logs. De database is zo opgebouwd dat alles logisch met elkaar verbonden is. Bijvoorbeeld: een ruimte hoort bij een kantoor en kan meerdere kenmerken hebben. De structuur is eenvoudig en kan later makkelijk worden uitgebreid als dat nodig is.

Hieronder zie je het databaseschema, gevolgd door een korte uitleg van elke tabel.



ROOMS TABLE

De tabel Rooms bevat gegevens over vergaderruimtes en bestaat uit velden zoals id, name, office_id, floor en seats. Deze tabel speelt een centrale rol binnen de applicatie voor het beheren van ruimtes. Daarnaast wordt deze tabel gebruikt om vergaderruimtes te synchroniseren met Outlook, aangezien Outlook ruimtebeheer uitsluitend via rooms organiseert.

FEATURE TABLE

De Feature-tabel bevat een unieke id en een naam voor elk kenmerk. Via de tussentabel "RoomFeature" worden deze kenmerken gekoppeld aan ruimtes. Hierdoor kunnen meerdere kenmerken aan verschillende ruimtes worden verbonden en beheerd binnen de applicatie.

ROOMFEATURE TABLE

Ruimtes worden aan kenmerken gekoppeld door de tabel "RoomFeature". Het heeft twee kolommen: feature_id en room_id. Door deze twee kolommen kan een ruimte meerdere kenmerken hebben, en kan een kenmerk aan meerdere ruimtes zijn verbonden. Ook zorgt dit ervoor dat het gebruik van het systeem makkelijker en flexibeler wordt.

OFFICE TABEL

De tabel Offices slaat alle kantoren op binnen de applicatie en bevat onder andere een unieke ID en een naam. Elk kantoor is gekoppeld aan een adres via een verwijzing naar de Address-tabel. Daarnaast wordt de Offices-tabel gebruikt in de Rooms-tabel om aan te geven in welk gebouw een ruimte zich bevindt. Hierdoor kunnen ruimtes logisch geordend en gefilterd worden per kantoorlocatie.

ADDRESS TABEL

De tabel Address houdt de adresgegevens van de kantoren bij. Ze bevat een id als primaire sleutel, en velden zoals street, city, state, postal_code en country_or_region. Elk adres is gekoppeld aan één kantoor en omgekeerd. Dit zorgt voor een duidelijke een-op-een relatie tussen kantoren en hun adressen.

LOG TABEL

De tabel Log registreert alle wijzigingen binnen de applicatie die betrekking hebben op ruimtes, kantoren en kenmerken. Deze bevat een primaire sleutel id, een timestamp (datetime), email, action, name en description. Het veld email geeft aan wie de actie heeft uitgevoerd. Als een gebruiker in de applicatie zelf iets wijzigt, staat hier "Unknown", omdat er geen loginfunctie is. Wanneer een gebruiker via Outlook een ruimte boekt, annuleert of aanpast, wordt het bijbehorende e-mailadres geregistreerd. Het veld action geeft aan of het om een create, update of delete gaat. In name staat de naam van de ruimte of entiteit waarop de actie betrekking heeft, en description bevat een korte uitleg van de wijziging. Deze tabel biedt zo inzicht en controle over alle acties in het systeem.

EVENT TABEL

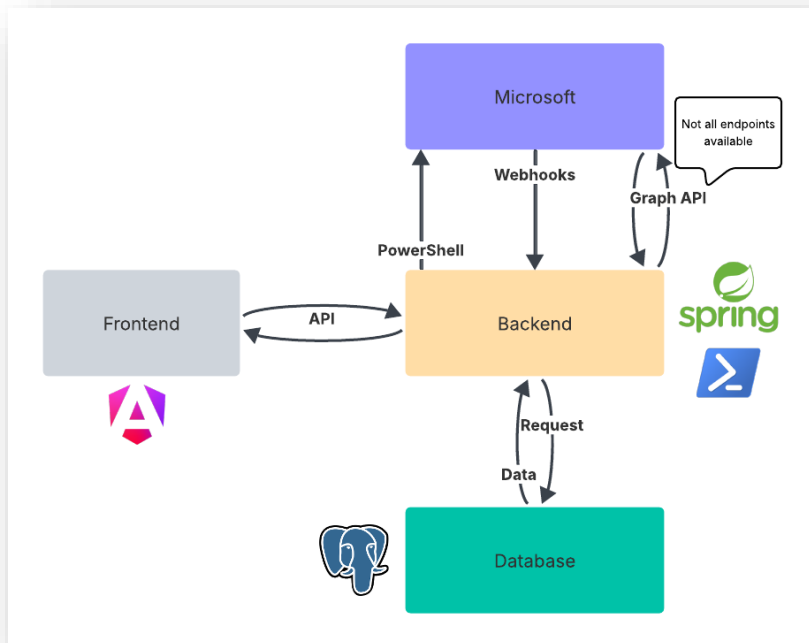
De tabel Event bevat gegevens over vergaderingen die via Outlook aan de applicatie gekoppeld zijn. De primaire sleutel is id, daarnaast wordt het Outlook-specifieke event_id opgeslagen om de koppeling te behouden. Verder bevat de tabel het subject van de afspraak, de start_time en end_time als datetime-waarden, en de is_all_day status. Ook wordt location_alias vastgelegd, wat overeenkomt met de naam van de ruimte waar de afspraak plaatsvindt. Ten slotte wordt de organizer geregistreerd, het e-mailadres van de organisator van de afspraak. Deze tabel maakt het mogelijk vergadering tussen bedrijven te synchroniseren en te beheren binnen de applicatie.

FLYWAY SCHEMA HISTORY TABEL

De tabel flyway_schema_history houdt bij welke migraties al op de database zijn uitgevoerd. Elke keer dat een migratiebestand wordt toegepast, maakt Flyway daar een regel voor aan. Die regel bevat onder andere het versienummer, de naam van het script en of het goed is verlopen. Door deze tabel weet Flyway welke scripts nog moeten worden uitgevoerd en welke al gebeurd zijn. Zo worden migraties niet dubbel uitgevoerd en gebeurt alles in de juiste volgorde. De tabel wordt automatisch beheerd door Flyway en is belangrijk voor een goed verloop van database-updates.

3.7. Werking van de applicatie

In dit onderdeel leg ik uit hoe de applicatie functioneert, aan de hand van een zelfgemaakt schema. Dit schema wordt hieronder weergegeven en laat zien hoe de verschillende onderdelen van de applicatie met elkaar samenwerken. Op basis daarvan geef ik een duidelijke toelichting op de werking van het systeem, van gebruikersinteracties tot de koppeling met Outlook.



FRONTEND EN BACKEND COMMUNICATIE

De frontend vraagt via de API gegevens op bij de backend. De backend zoekt deze in de database op en stuurt ze terug. Andersom gebeurt hetzelfde: als je via de frontend een nieuwe ruimte invoert, gaat dat via de API naar de backend, en daar wordt de ruimte in de database opgeslagen. Zo wisselen frontend en backend continu informatie uit.

SYNCHRONISATIE MET MICROSOFT OUTLOOK

Microsoft biedt geen API om ruimtes te maken of bij te werken, daarom gebruikt de backend PowerShell-scripts om dit in Outlook te regelen. Wanneer een ruimte wordt aangemaakt of aangepast, voert de backend met die scripts de wijzigingen in Outlook door. Houd er wel rekening mee dat het tot 24 uur kan duren voordat Outlook deze aanpassingen oppikt. Dit geldt alleen voor ruimtes. Kantoren en kenmerken worden bij het aanmaken niet met Outlook gesynchroniseerd.

VERWERKING VAN MICROSOFT MELDINGEN VIA WEBHOOKS

Wanneer een ruimte via Outlook wordt geboekt, boeking aangepast of geannuleerd, stuurt Microsoft een melding naar een webhook waarop de backend is geabonneerd. De backend ontvangt deze melding en haalt via de Microsoft Graph API de details van de gebeurtenis op. Vervolgens past de backend deze gegevens ook aan in de agenda's van de andere gekoppelde bedrijven, zodat alle boekingen overal up-to-date blijven.

AANPASSINGEN VAN KANTOREN EN KENMERKEN

Bij het aanmaken van kantoren en kenmerken vindt alleen een opslag in de database plaats. Bij wijzigingen aan deze entiteiten, omdat ze gekoppeld zijn aan een ruimte, worden de aanpassingen via PowerShell ook in Microsoft doorgevoerd om de consistentie te bewaren.

LOGGING VAN GEBEURTENISSEN

Alle CRUD-acties (aanmaken, wijzigen, verwijderen) in de applicatie en belangrijke gebeurtenissen vanuit Outlook, worden opgeslagen in een logtabel. Deze tabel bevat gegevens over de actie, het tijdstip, de betrokken ruimte en, indien beschikbaar, de gebruiker (of "Unknown" wanneer er geen login is). Hierdoor kunnen alle veranderingen eenvoudig worden gevolgd en gecontroleerd. Deze loggegevens kunnen ook worden opgevraagd en bekeken via de frontend van de applicatie.

3.8. Hosting

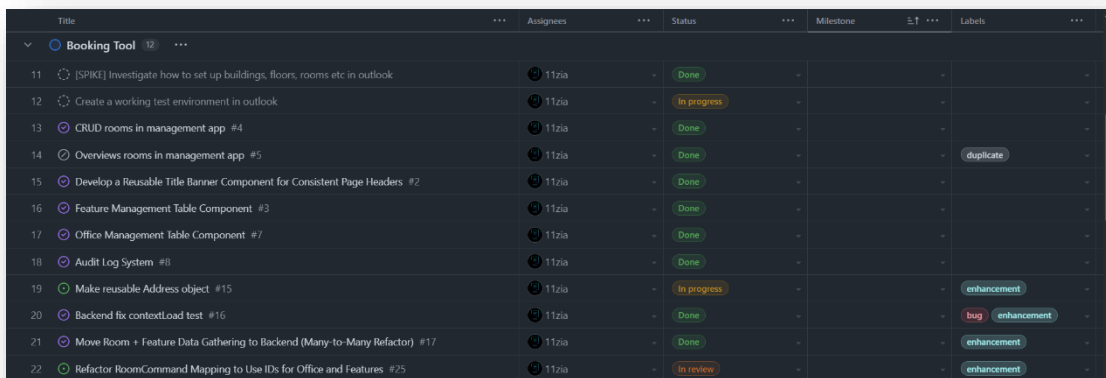
De applicatie wordt op dit moment lokaal gedraaid op de computer van de gebruiker. Met Docker en Docker Compose worden de database (PostgreSQL) en Adminer opgezet in containers. De backend start je via IntelliJ, en de frontend via Node.js. De Microsoftomgeving draait wel in de cloud. Die is nodig om Outlook te gebruiken, zoals het boeken van ruimtes en het bijhouden van agenda's. De applicatie maakt dus lokaal verbinding met Microsoftdiensten in de cloud voor alles wat met Outlook te maken heeft.

4. Projectmanagement

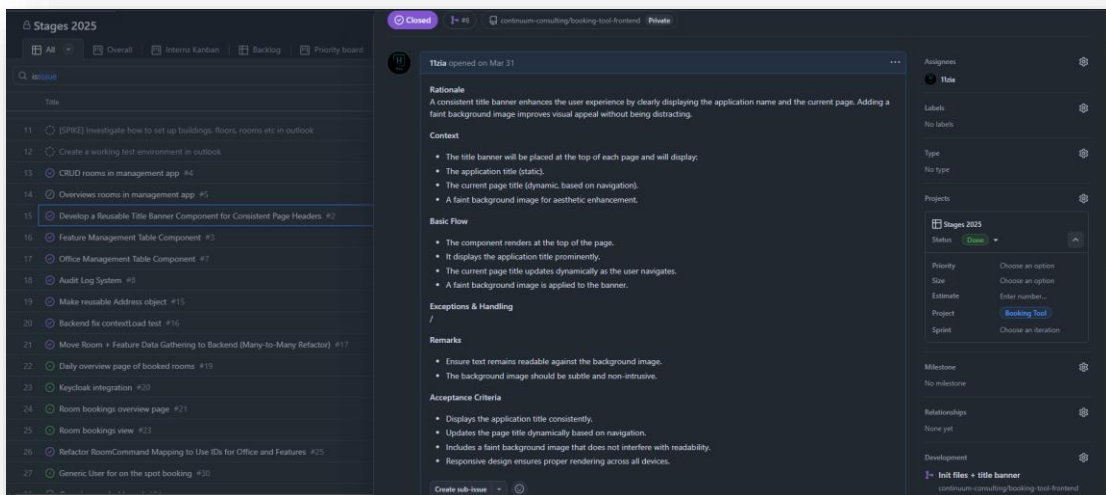
Voor het beheren van het project is gebruikgemaakt van GitHub Projectmanagement. Om overzicht te houden over de taken werd er gebruik van projectboards en issues. Elke taak werd toegevoegd als een issue en kreeg een duidelijke titel, beschrijving en label. Ik werkte in sprints van twee weken, waarbij ik telkens probeerde om een deel van de applicatie af te ronden. Aan het einde van elke sprint was er een kleine demo aan het bedrijf om de voortgang te tonen en feedback te verzamelen door mij.

Tijdens het werken gebruikte ik ook pull requests om nieuwe code voor te stellen. Op die manier kon het bedrijf elke wijziging apart nakijken voordat die werd toegevoegd aan het project. Dit hielp om structuur te houden in de code en fouten te vermijden.

Hieronder zijn twee screenshots toegevoegd: één van het algemene projectbord met de sprintplanning, en één van een specifiek issue waarin een taak beschreven staat.



Title	Assignees	Status	Milestone	Labels
11 [SPIKE] Investigate how to set up buildings, floors, rooms etc in outlook	11zia	Done		
12 Create a working test environment in outlook	11zia	In progress		
13 CRUD rooms in management app #4	11zia	Done		
14 Overviews rooms in management app #5	11zia	Done		duplicate
15 Develop a Reusable Title Banner Component for Consistent Page Headers #2	11zia	Done		
16 Feature Management Table Component #3	11zia	Done		
17 Office Management Table Component #7	11zia	Done		
18 Audit Log System #8	11zia	Done		
19 Make reusable Address object #15	11zia	In progress		enhancement
20 Backend fix contextLoad test #16	11zia	Done		bug enhancement
21 Move Room + Feature Data Gathering to Backend (Many-to-Many Refactor) #17	11zia	Done		enhancement
22 Refactor RoomCommand Mapping to Use IDs for Office and Features #25	11zia	In review		enhancement



Stages 2025

Title banner

Rationale

A consistent title banner enhances the user experience by clearly displaying the application name and the current page. Adding a faint background image improves visual appeal without being distracting.

Context

- The title banner will be placed at the top of each page and will display:
 - The application title (static).
 - The current page title (dynamic, based on navigation).
 - A faint background image for aesthetic enhancement.

Basic Flow

- The component renders at the top of the page.
- It displays the application title prominently.
- The current page title updates dynamically as the user navigates.
- A faint background image is applied to the banner.

Exceptions & Handling

/

Remarks

- Ensure text remains readable against the background image.
- The background image should be subtle and non-intrusive.

Acceptance Criteria

- Displays the application title consistently.
- Updates the page title dynamically based on navigation.
- Includes a faint background image that does not interfere with readability.
- Responsive design ensures proper rendering across all devices.

Assignees

11zia

Labels

No labels

Type

No type

Projects

Stages 2025

Priority

Choose an option

Size

Choose an option

Estimate

Enter number...

Project

Booking Tool

Sprint

Choose an iteration

Milestone

No milestone

Relationships

None yet

Development

Init files + title banner

5. Besluit

Het project had als doel een systeem te maken waarmee vergaderruimtes makkelijk beheerd kunnen worden binnen verschillende organisaties. Belangrijk was dat ruimtes via Outlook geboekt konden worden en dat deze boekingen automatisch worden doorgegeven aan alle gekoppelde bedrijven. Ongeacht waar of door wie de afspraak is gemaakt hebben alle bedrijven dezelfde actuele informatie.

CONCLUSIE

De basisfunctionaliteiten van het systeem werken zoals verwacht namelijk ruimtes kunnen aangemaakt, aangepast en gesynchroniseerd worden met Outlook. Vergaderingen worden correct verwerkt en gedeeld tussen de gekoppelde bedrijven. Tijdens de ontwikkeling zijn enkele problemen opgetreden, maar deze zijn succesvol opgelost binnen de stageperiode. De belangrijkste technische doelen zijn daarmee binnen deze periode behaald.

REFLECTIE

In dit project heb ik veel bijgeleerd. Door dat ik stapsgewijs gewerkt heb aan deze opdracht heb ik ervaringen opgedaan met verschillende tools. Ook heb ik ingezien dat het belangrijk is om een project in te delen in duidelijke stukken. Een goed gestructureerde planning bijhouden is ook zeer belangrijk om tot een mooi resultaat te komen. Tijdens mijn stageperiode heb ik geleerd hoe ik complexe problemen zelfstandig aan moet pakken. Ook heb ik geleerd om tijdig hulp te vragen en in te schakelen wanneer dat nodig was.

Ik ben tevreden met mijn behaald resultaat. Het origineel opgestelde plan is volledig behaald en functioneert zoals bedoeld. Vooral de integratie met Outlook leek een complexe uitdaging, maar ik ben blij dat dit gelukt is en ik heb hier veel van geleerd. Naast het oorspronkelijke plan heb ik ook nagedacht over mogelijke uitbreidingen en optimalisaties voor de toekomst, zoals:

- Het gebruik van WebSockets om PowerShell-scripts op de achtergrond te laten draaien en de status live te updaten in de frontend.
- Een notificatiesysteem om gebruikers beter op de hoogte te houden van systeemgebeurtenissen.
- Een inlog- en uitlogfunctionaliteit voor betere beveiliging en gebruikersbeheer.
- Het automatisch afhandelen van verlopen webhooks.
- Het effectief gebruiken van video-, projector-, audio- en rolstoeltoegankelijkheidsgegevens in Outlook.

Deze extra ideeën zouden het systeem nog completer maken en zijn een mogelijke volgende stap.