AFSTUDEERVERSLAG

Versienummer	1.0
Versiedatum	8-6-2022
Versietype	Final

Verbree, Anna

ORDINA

VOORWOORD

Voor u ligt het afstudeerverslag van mijn afstudeerstage. Deze afstudeerstage is uitgevoerd bij Ordina. Het verslag is geschreven voor het afstuderen aan de opleiding Informatica aan de Hogeschool Leiden. In de periode van februari t/m juni 2022 ben ik bezig geweest met dit afstudeerproject.

Dit afgelopen halfjaar heb ik onderzoek gedaan naar frameworks voor JavaScript, TypeScript en C#. Ook heb ik aan een activiteitenapplicatie gewerkt die met één van deze frameworks werkt en geschreven is in TypeScript en .NET. Verder heb ik veel geleerd over het gebruik van de verschillende tools van het Microsoft Azure platform, zoals hoe pipelines opgezet kunnen worden in Azure DevOps en hoe een database opgezet kan worden in Azure Portal.

Met de hulp van mijn begeleiders Wilco Morren, Stuart van der Lee en Jan ter Schure heb ik de eisen van de opdracht opgesteld. Stuart en Jan hebben mij hierbij geholpen om te bepalen welke functionaliteiten voor de applicatie belangrijk waren. Hun feedback en perspectieven hebben mij verder geholpen om mijn verslag beter te maken.

Daarnaast wil ik Masha Steinmann, Noor van Ewijk en Lise Voorburg bedanken voor alle hulp tijdens mijn stage en hun enthousiasme over de activiteitenapplicatie. Voor al hun tips en inzichten wil ik graag Mark van der Hart en Ahmet Karatas bedanken. Ook wil ik graag alle collega's bedanken voor de aangeboden hulp.

Ik wil ook graag de trainers bedanken van de trainingen die ik gevolgd heb. Deze trainingen hebben mij veel tools geboden om mijn stage positief af te ronden.

Als laatste wil ik graag mijn docentbegeleider Charlotte Prins-Ipema en alle andere mensen die mij feedback gegeven hebben bedanken.

Ik wens u veel leesplezier toe,

Alphen aan den Rijn, 14 juni 2022

Anna Verbree

1 SAMENVATTING

Ordina is een groot bedrijf met veel medewerkers, waarvoor er veel activiteiten georganiseerd worden. Deze activiteiten worden op veel verschillende plaatsen opgeslagen, waardoor informatie over deze activiteiten lastig te vinden is. Mijn doel is daarom om een overzicht van alle activiteiten te genereren.

Om dit te realiseren is besloten dat er een activiteitenapplicatie gemaakt moest worden. Hiervoor is ten eerste onderzocht welk framework het beste gebruikt kan worden voor de front-end. Met behulp van een analyse zijn daarna de requirements opgesteld. Op basis van deze requirements zijn verschillende ontwerpen gemaakt voor de applicatie. Deze ontwerpen waren de basis voor het realiseren van de applicatie.

Uiteindelijk is er een activiteitenapplicatie gemaakt die bestond uit een backend geschreven in .NET Core en een front-end geschreven in TypeScript. Het uitgevoerde onderzoek gaf aan dat Angular het beste framework voor de front-end was, waardoor dit framework ook gebruikt is in de applicatie. Voor de applicatie zijn er ook unit testen geschreven die automatisch uitgevoerd worden.

Aan het einde van deze afstudeerstage is een activiteitenapplicatie gerealiseerd die een overzicht levert van alle activiteiten die op de MTech afdeling van Ordina georganiseerd worden. Hierdoor is het vinden van informatie over deze activiteiten veel makkelijker geworden. In de toekomst kan de applicatie uitgebreid worden om activiteiten van andere afdelingen toe te voegen.

2 INHOUDSOPGAVE

3 INLEIDING

Ordina is een IT-dienstverlener in Nederland, België en Luxemburg. Aangezien het een groot bedrijf is, worden er veel activiteiten georganiseerd. Voor al deze activiteiten die georganiseerd worden was er geen algemeen overzicht met alle informatie over de activiteiten.

Het probleem dat plaatsvond was dat de gebruikers veel moeite moesten doen om erachter te komen wanneer activiteiten plaatsvinden. Voor organisatoren had dit als gevolg dat zij ongemerkt activiteiten inplanden op momenten dat er al andere activiteiten plaatsvonden. Voor de deelnemers had dit als gevolg dat zij niet makkelijk in konden zien voor welke activiteiten zij zich opgegeven hadden. Het doel van deze afstudeerstage was daarom om een activiteitenapplicatie te ontwikkelen die al deze problemen oplost. Hiervoor moest eerst aan de hand van een onderzoek bepaald worden welk framework aan de front-end van de applicatie gebruikt zou worden. De applicatie zou uiteindelijk de activiteiten van de MTech afdeling tonen, met in de toekomst de mogelijkheid om dit uit te breiden naar andere afdelingen.

Dit document bevat ten eerste in het kort een inleiding van het project. Ten tweede zijn in de aanloop de afstudeerorganisatie en de uitgangspunten van het project beschreven. Verder is hier te vinden wat voor aanpak het project had en wat de verwachtingen waren.

Na de aanloop zijn de achtergronden beschreven. Hierin wordt beschreven wat de aanleiding van het project was en wat de probleem- en doelstelling waren. Verder wordt hier uitgelegd wie de stakeholders waren en wat het verwachte resultaat van dit project was.

Vervolgens is het daadwerkelijke proces beschreven. Hierin wordt ten eerste de daadwerkelijke aanpak uitgelegd. Daarna wordt van de verschillende fasen van het project uitgelegd hoe deze doorlopen waren.

Aan het einde volgt er een reflectie op het gehele afstudeerproces. Hierin wordt ook uitgelegd hoe de competenties aangetoond zijn.

4 AANLOOP

4.1 Afstudeerorganisatie

Binnen Ordina zijn er veel afdelingen. Eén van deze afdelingen is de MTech afdeling. Op deze afdeling wordt voornamelijk met Microsoftproducten gewerkt. Zo wordt er gewerkt met Azure en geprogrammeerd in .NET. Op deze afdeling is de afstudeerstage uitgevoerd. Omdat deze afdeling gespecialiseerd is in het werken met Microsoftproducten was afgesproken dat de backend van de activiteitenapplicatie in .NET Core geschreven zal worden. Dit had namelijk als gevolg dat er goede begeleiding gegeven kon worden, aangezien .NET van Microsoft is. Het gevolg hiervan is dat problemen met .NET makkelijk opgelost kunnen worden.

De begeleiding tijdens deze stage is uitgevoerd door 3 begeleiders: Jan, Stuart en Wilco. De dagelijkse begeleiding werd uitgevoerd door Jan en Stuart. Wilco was verantwoordelijk voor de algemene begeleiding. Daarnaast was hij ook mijn manager binnen Ordina.

Voor de begeleiding is er in Microsoft Teams een team opgezet, waarin er vragen gesteld konden worden en overlegd kon worden over het project. Dit team was alleen bereikbaar voor mij en mijn begeleiders.

Tijdens deze afstudeerstage waren er nog twee andere stagiairs die een afstudeerstage uitvoerden op de MTech afdeling. Eén van deze stagiairs voerde zijn stage uit in Nieuwegein, de andere werkte in Groningen. Binnen Microsoft Teams is er ook een team opgezet voor mij, mijn begeleiders en de andere stagiairs. In dit team konden algemene vragen gesteld worden over het afstuderen binnen Ordina en kon er eventueel feedback uitgewisseld worden tussen de stagiairs.

Samen met de stagiairs en begeleiders is er om de week op vrijdag een voortgangsmeeting gehouden waarin de voortgang van de projecten besproken werd en er feedback gekregen kon worden.

De eerste tijd van de stage is voornamelijk thuisgewerkt. In de loop van het project is met de begeleiders en andere stagiairs besloten om elke woensdag op kantoor te werken. Later is er ook besloten om voor de voortgangsgesprekken naar kantoor te komen.

4.2 Uitgangspunten

Mijn afstudeerstage heeft plaatsgevonden bij Ordina over een periode van 20 lesweken. Het doel van deze afstudeerstage was het aantonen van de vaardigheden die geleerd zijn tijdens de studie Informatica aan de Hogeschool Leiden. De vaardigheden zijn aan te tonen met verschillende competenties, namelijk de A- en B-competenties. De A-competenties bestaan uit Onderzoek, Leren leren, Professioneel werken en Innovatie.
De gekozen B-competenties waren Software analyseren, Software ontwerpen en Software realiseren.

Om deze competenties aan te tonen hebben er verschillende werkzaamheden plaatsgevonden. Ten eerste is er een onderzoek uitgevoerd. Hiervan zijn de resultaten vastgelegd in een onderzoeksrapport. Verder was er een analyse van de huidige situatie met betrekking tot activiteiten binnen Ordina.

De werkzaamheden die het meeste tijd hebben gekost tijdens deze stage waren het ontwerpen en realiseren van software.

Er is voor het uitvoeren van de genoemde werkzaamheden rekening gehouden met de planning zoals te zien in interne bijlage 2 Planning.

Zoals te zien is in de planning waren er ook een aantal deadlines. Dit zijn vaste momenten waarop iets ingeleverd moet worden aan school of wanneer er een evenement plaatsvindt. Deze deadlines stonden vanaf het begin vast.

Aan het einde van het afstudeerproces moesten de volgende producten opgeleverd worden:

- Afstudeerplan
- Onderzoeksrapport
- Agile backlog
- Activiteitenapplicatie
- Test rapportage
- Afstudeerverslag

4.3 Verantwoording aanpak

Het project is uitgevoerd volgens de agile methodiek. Bij agile is het belangrijk om goed met veranderingen om te gaan. Tijdens dit project werden de ontwerpen dan ook steeds waar nodig bijgewerkt. De documenten die opgeleverd moesten worden waren bepaald op basis van het RUP software development proces zoals beschreven in het boek RUP OP MAAT (Collaris & Dekker, 2011).

Tijdens het ontwikkelen van de applicatie is gewerkt volgens de Scrum methode. Hierbij zijn sprints van 2 weken toegepast. De manier waarop Scrum toegepast is wijkt iets af van de standaardmethode. Dit heeft te maken met het feit dat het Scrum team voor dit project maar uit 1 persoon (ikzelf) bestond. Dit had als gevolg dat de daily Scrums niet nodig waren. De sprintplanning gebeurde ook niet in een overleg. Het inplannen van de volgende sprint gebeurde steeds rond het einde van de vorige sprint. Nadat deze planning vaststond werd deze besproken in de sprint review. De sprint reviews vonden plaats aan het einde van elke sprint en hadden de vorm van een voortgangsmeeting. Hierbij waren ook de andere stagiairs op de afdeling aanwezig en rapporteerden zij ook over hun voortgang.

De algemene voortgang en werkzaamheden tijdens het project werden bijgehouden in de vorm van een wekelijks logboek. In dit logboek werd per week kort weergegeven wat de belangrijke data, deadlines en werkzaamheden waren.

4.4 Eigen verwachtingen

Voor dit project waren er twee verwachtingen. Ten eerste werd er verwacht dat er dingen veranderen met betrekking tot de aanpak. Zoals eerder beschreven is er volgens de agile methodiek gewerkt, waarbij er veel op veranderingen gefocust wordt. Ten tweede werd er verwacht dat er in de loop van het project veranderingen zullen komen in de eisen van de applicatie. Dit heeft dan ook weer een gevolg op de aanpak.

5 ACHTERGRONDEN

5.1 Aanleiding

Voor Ordina is het natuurlijk van groot belang dat de teams goed kunnen samenwerken. Binnen de teams werken vaak verschillende soorten mensen met verschillende soorten verantwoordelijkheden. Om deze mensen goed samen te laten werken is er een goede teamcultuur nodig. Een goede teamcultuur bestaat uit vertrouwen, een gezamenlijk doel en betrokkenheid (Ordina, 2022).

Een goede teamcultuur ontstaat niet altijd vanzelf. Mede om deze reden organiseert Ordina veel activiteiten voor haar werknemers. Denk hierbij aan bijvoorbeeld trainingen om de samenwerking te verbeteren, maar ook team gerelateerde uitjes om de werknemers dichter bij elkaar te brengen.

Ordina is natuurlijk een groot bedrijf, wat inhoudt dat er veel verschillende teams zijn. Als elk team iedere maand een activiteit organiseert, ontstaan er al snel veel activiteiten. Natuurlijk zijn er ook activiteiten die georganiseerd worden voor het hele bedrijf en activiteiten georganiseerd door de verschillende communities binnen Ordina. Kortom, er vinden veel activiteiten plaats binnen Ordina.

5.2 Stakeholders

Voor dit project waren er een aantal stakeholders. De activiteitenapplicatie is geschreven voor de MTech afdeling van Ordina, met in de toekomst de mogelijkheid tot uitbreiden naar andere afdelingen. Dit betekent dat alle medewerkers van de MTech afdeling stakeholders zijn.

De verdeling van stakeholders was als volgt:

Rol	Vertegenwoordigers	Betrokkenheid
Opdrachtgever	Wilco Morren Stuart van der Lee Jan ter Schure	Besluiten maken over keuzes en veranderingen met betrekking tot de opdracht.
Opdrachtnemer	Anna Verbree	Is verantwoordelijk voor het uitvoeren van de opdracht.
Gebruikers	MTech medewerkers	De toekomstige gebruikers van de applicatie.

5.3 Probleemstelling

Veel teams hebben een eigen plek waar informatie gedeeld wordt. Zo heeft een team bijvoorbeeld vaak een eigen team in Microsoft Teams. De communities delen hun informatie vaak op Connect, dit is het Intranet voor de werknemers van Ordina. Op Connect is er een overzicht van

informatie voor werknemers te vinden, zoals het laatste nieuws en links naar andere belangrijke websites. Onder deze informatie vielen ook de activiteiten die georganiseerd worden. Dit had als gevolg dat als je wilde weten welke activiteiten een team organiseerde, je vaak helemaal moest navigeren naar de locatie waar dat team informatie deelt. In andere gevallen moest er zelfs naar de informatie gezocht worden. Hierdoor was het makkelijk om interessante activiteiten over het hoofd te zien.

Een ander probleem dat hier ontstond had te maken met wanneer de activiteiten georganiseerd werden. Omdat er geen algemene plaats was waar alle georganiseerde activiteiten te vinden waren, kon het gebeuren dat bijvoorbeeld een team een activiteit organiseert op het moment dat een community een activiteit organiseert. Als iemand dan in dat team zat en tegelijkertijd lid was van de community, kon hij niet deelnemen aan allebei de activiteiten. Dit kon dan weer zorgen voor een lagere opkomst bij de beide activiteiten.

5.4 Doelstelling

Het doel van mijn afstudeerstage was om een overzicht te creëren van alle georganiseerde activiteiten. Hierbij ging het ten eerste om een overzicht van alle activiteiten binnen de MTech afdeling, met daarbij de mogelijkheid om later nieuwe afdelingen toe te voegen.

Het doel van het overzicht was natuurlijk om de eerdergenoemde problemen op te lossen. Het eerste belangrijke doel daarbij was om de gebruiksvriendelijkheid te verbeteren. Hieronder vallen gebruiksvriendelijkheid voor de deelnemers van activiteiten en gebruiksvriendelijkheid voor de organisatoren van activiteiten. Voor werknemers moest het makkelijker worden om in te zien welke activiteiten er georganiseerd worden. De organisatoren moesten in kunnen zien op welke momenten er al activiteiten georganiseerd worden, voordat zij hun eigen activiteit inplannen.

Een ander doel was om het voor een werknemer duidelijker te maken voor welke activiteiten hij zich ingeschreven heeft. In het overzicht moest dus een plek komen waar een overzicht is van de activiteiten waar de werknemer zich al voor ingeschreven heeft. Ook zouden deze activiteiten in de agenda van de werknemer toegevoegd moeten kunnen worden.

Als laatste was het ook belangrijk dat er makkelijk gecommuniceerd kon worden tussen de deelnemers en de organisator van een activiteit. De deelnemers moesten op een makkelijke manier vragen over activiteiten kunnen stellen. Ook moesten de organisatoren hun deelnemers op de hoogte kunnen houden van bijvoorbeeld wijzigingen of informatie over de activiteiten.

5.5 Verwacht resultaat

Het verwachte resultaat van deze opdracht was de oplevering van een applicatie die voldoet aan de eerdergenoemde doelstellingen. Het ging hierbij om een webapplicatie. Zoals beschreven zou de backend van deze applicatie in .NET Core geschreven worden. De vorm van de front-end was nog niet bepaald. Dit zou gebeuren aan de hand van het onderzoek.

De applicatie moest de genoemde problemen oplossen en de algemene ervaring van de gebruikers verbeteren. Zo was het de bedoeling dat er een koppeling met Outlook kwam, waardoor de activiteiten automatisch in de agenda van de gebruiker terechtkwamen.

6 HET PROCES

6.1 Aanpak

Het doel van de afstudeerstage was om een activiteitenapplicatie te maken. Om op een effectieve wijze tot dit product te komen is het hele project uitgevoerd met een specifieke aanpak. Ten eerste is er begonnen met het uitvoeren van het onderzoek. Dit onderzoek bepaalde de architectuur van de front-end van de applicatie.

Na het onderzoek was het van belang om de specifieke details van het product vast te stellen. Ten eerste was het onderzoek een input. Verder is er overleg geweest met de stakeholders van het project om tot de requirements te komen. De rest van de requirements zijn door mijzelf bepaald.

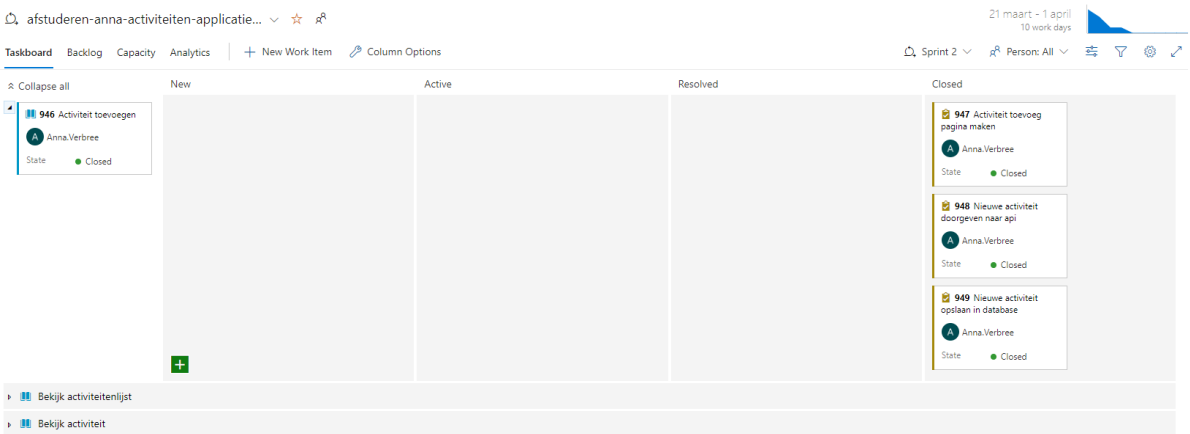
Na het opstellen van de requirements was de volgende stap om te beginnen met de ontwerpfase. In de ontwerpfase zijn er de volgende UML-diagrammen gemaakt:

- Use-case diagrammen;
- Klassediagrammen
- Activiteitendiagrammen;
- Sequence-diagrammen

Deze diagrammen zijn in de loop van het project gemaakt als onderdeel van de sprints. Hierbij is er gewerkt in sprints van 2 weken.

De eerste week van de sprint werd gebruikt om de diagrammen te maken die betrekking hadden op de user story's die in de sprint gerealiseerd zouden worden. Als gevolg hiervan werd bijvoorbeeld het klassediagram elke sprint aangevuld. Ook zijn in deze week de schermontwerpen gemaakt. Als laatste is in deze week het testplan opgesteld en verder aangevuld.

De tweede week werd voornamelijk gebruikt voor het realiseren van de user story's. De user story's zijn voor elke sprint vastgesteld in Azure DevOps en onderverdeeld in taken. Hieronder volgt een voorbeeld van een sprint in DevOps.



Figuur 1 Azure DevOps pagina van sprint 2

De user story's en taken die niet voltooid waren aan het einde van de sprint werden meegenomen naar de volgende sprint. In dit geval werd dit ook doorgegeven aan de stakeholders.

Behalve het realiseren van de user story's werden in deze week ook de integratie testen en unit testen uitgevoerd. Deze laatste vonden automatisch plaats door middel van een pipeline.

Aan het einde van elke sprint was er een voortgangsmeeting met de stakeholders. Bij deze meeting waren ook de twee andere stagiairs van de MTech afdeling aanwezig. Hier was de mogelijkheid om vragen te stellen en eventueel de andere stagiairs feedback te geven. Tijdens de voortgangsmeeting is steeds getoond op welke punten de applicatie veranderd was. Zo kon hier feedback op gegeven worden. Als gevolg hiervan kon op tijd bijgestuurd worden wanneer er iets niet goed was. Verder werden op dit moment ook andere nuttige tips gegeven. Een voorbeeld hiervan was een uitleg over een mogelijke manier om de branches van het project in te stellen in DevOps.

De DevOps omgeving voor dit project is alleen gebruikt om de voortgang van de applicatie bij te houden. Om een overzicht te creëren voor de andere taken die uitgevoerd zijn in het project zijn er andere tools gebruikt.

Ten eerste was het belangrijk dat er bijgehouden werd waar elke dag aan gewerkt werd. Hiervoor zijn Trello-borden gebruikt. Een bord zag er als volgt uit:

Figuur 2 Trello-bord voor het onderzoeksverslag

In dit voorbeeld is het Trello-bord voor het onderzoeksverslag te zien. Het bevat de kopjes Te doen, Bezig en Klaar. De taken worden steeds van kopje naar kopje verplaatst wanneer de voortgang verandert.

De Trello-borden werden elke dag bijgewerkt. Aan begin van elke week werden de taken die klaar waren verwijderd van het bord, om zo het overzicht te behouden.

Behalve de taken die op een bepaald moment uitgevoerd werden bij te houden was het ook belangrijk om vast te leggen welke taken in het verleden uitgevoerd waren. Om deze reden is er ook een logboek bijgehouden in OneNote. Dit zag er als volgt uit:

Figuur 3 Logboek van week 5

Zoals te zien is het logboek per week bijgehouden. Hierdoor is er per logboek item meer informatie vastgelegd. Ten eerste worden de deadlines en afspraken voor de week vastgelegd. Daaronder is aangegeven welke werkzaamheden van toepassing waren voor die week. Vervolgens wordt in de memo verteld hoe de week verlopen is en eventueel belangrijke informatie. Als laatste wordt er nog een overzicht getoond van de deadlines en afspraken voor de volgende week.

6.2 Onderzoek

Het afstuderen begon met het doen van onderzoek naar front-end frameworks. Het doel van dit onderzoek was om te bepalen welke vorm de front-end van de applicatie zou hebben. Zoals eerder beschreven was de vorm van de backend al bepaald. Voor de front-end moest daarentegen nog bepaald worden welke programmeertaal en frameworks het beste gebruikt konden worden. Voordat de beste programmeertaal gekozen kon worden moest eerst bepaald worden wat het beste framework was voor de applicatie.

Voor dit onderzoek was de volgende hoofdvraag opgesteld:

Welk framework kan Ordina het beste gebruiken voor de front-end van de activiteitenapplicatie?

Om deze vraag te kunnen beantwoorden zijn de volgende deelvragen opgesteld:

- **Welke front-end frameworks zijn er beschikbaar voor JavaScript/TypeScript en C#?**
- **Welk framework biedt de meeste en beste functionaliteiten en eigenschappen voor de activiteitenapplicatie?**
- **Welk framework biedt de beste performance voor de activiteitenapplicatie?**

De eerste verwachtingen waren dat React of Vue het beste framework zou zijn. Als gevolg hiervan zou de programmeertaal TypeScript gebruikt moeten worden. Na overleg met de

stakeholders van het project bleek er nog een ander alternatief mogelijk. Het bleek dat C# ook een framework heeft die voor de front-end gebruikt kon worden, genaamd Blazor. Dit maakt het mogelijk om ook .NET voor de front-end te gebruiken.

In deze paragraaf wordt het verloop van het onderzoek in het kort beschreven. Een volledig verslag van het onderzoek is te vinden in bijlage 2 Onderzoeksrapport.

Tijdens het vooronderzoek is gekeken wat een framework precies is. Daarnaast was het belangrijk om te bepalen wat voor soorten frameworks er zijn en wat voor soort framework nodig is voor de applicatie.

Het daadwerkelijke onderzoek bestond uit drie fasen. In de eerste fase is er een lijst met frameworks opgesteld. Het bleek hier dat er zoveel frameworks zijn voor JavaScript en TypeScript, dat gezien de beschikbare tijd het aantal frameworks beperkt moest worden. Dit is gedaan door het opstellen van de volgende eisen:

1. Het framework is voor JavaScript/TypeScript;
2. De laatste stabiele versie van het framework is niet later dan 4 jaar geleden uitgegeven;
3. De officiële website van het framework noemt het een framework.

Door deze eisen toe te passen wordt ervoor gezorgd dat alle frameworks voor het onderzoek geschreven zijn voor de juiste programmeertalen, ze goed bijgehouden worden door hun makers en dat het ook daadwerkelijk frameworks zijn.

In de tweede fase is gekeken welk functionaliteiten en kenmerken de frameworks hadden. Hier zijn specifiek naar punten gekeken die van belang zijn voor de activiteitenapplicatie. Dit waren de volgende punten:

- Licentie
- Kosten
- Programmeertaal
- Routing functionaliteit
- Soort data-binding
- Volledigheid van de officiële guide
- Aantal stars op GitHub
- Testing tools
- Type architectuur
- Localization
- Form validatie
- Hooks

In bijlage 2 Onderzoeksrapport wordt in detail besproken waarom deze punten gekozen zijn. Aangezien ze allemaal van belang zijn voor de applicatie zou het beste framework zoveel mogelijk van deze punten toepassen. Voor de metingen in deze fase zijn de resultaten vastgelegd in een Excel bestand. Deze resultaten zijn als tabel als bijlage van het onderzoeksrapport opgenomen.

De nodige informatie voor deze fase is ten eerste opgezocht op de officiële handleidingen van de frameworks. In een aantal gevallen was hier geen informatie te vinden over bepaalde functionaliteiten. In deze gevallen is naar andere bronnen gekeken. Zo was deze informatie vaak wel te vinden op de officiële forums van de frameworks of op Stack Overflow.

In de derde fase van het onderzoek is gekeken naar benchmarks van de frameworks. In deze benchmarks werd de performance van de frameworks gemeten op 3 verschillende punten. Dit waren de volgende drie punten:

1. Duur in milliseconden
2. Opstartstatistieken in lighthouse met mobiele simulatie
3. Geheugentoewijzing in MB's

Voor elk van deze punten werden er verschillende operaties uitgevoerd. Dit is allemaal in detail beschreven in bijlage 2 Onderzoeksrapport in paragraaf 3.2 De onderzoeksmethode. Hier is verder beschreven wat het belang is van deze benchmarks voor de applicatie.

Tijdens deze fase bleek dat niet alle frameworks benchmarks hadden. Van die frameworks is dus niet bekend hoelang de opstarttijd is. Het is hierbij mogelijk dat deze opstarttijd heel lang is, wat niet goed zou zijn voor de activiteiten applicatie. Om deze reden is besloten dat de frameworks die niet in de benchmarks voorkwamen niet geschikt waren voor de activiteitenapplicatie.

Nadat deze fasen uitgevoerd waren zijn de resultaten van fase twee en drie met elkaar vergeleken. Hier kwam uit elke fase een ander framework als beste naar voren. Hierbij is er gekeken wat belangrijker is: meer functionaliteiten of een betere performance. Uiteindelijk is besloten dat het belangrijker is dat er meer functionaliteiten zijn, zodat zoveel mogelijk gewenste functionaliteiten gerealiseerd kunnen worden. Dit betekende dat Angular of SAP OpenUI5 het beste framework was. Aangezien er geen benchmarks waren voor SAP OpenUI5 is geconcludeerd dat Angular het beste framework is voor de applicatie.

6.3 Analyse

Nadat het framework gekozen was moesten de requirements opgesteld worden. De eerste stap hiervoor was een gesprek met de stakeholders. Uit dit gesprek kwam een aantal requirements die de stakeholders graag wilden zien. Daarna heb ik zelf een groot aantal requirements bedacht. Vervolgens is er goed gekeken welke van deze requirements echt belangrijk waren. Dit bracht het aantal requirements terug tot een hoeveelheid die realistisch was voor de duur van deze afstudeerstage.

Nadat de requirements bepaald waren, is er gekeken naar de prioritering. De requirements die vanuit de stakeholders kwamen waren van het grootste belang, waardoor deze een hogere prioriteit kregen.

Een belangrijke requirement betrof de koppeling met de Outlook accounts. Gebruikers van de applicatie moeten via de applicatie een activiteit toe kunnen voegen aan hun Outlook agenda. Verder zouden de stakeholders graag zien dat er mails verstuurd worden als een activiteit aangepast wordt of als er een bericht op de pagina van een activiteit geplaatst wordt.

Voor deze requirement is er een kort onderzoek uitgevoerd. Hierbij is gekeken hoe de requirement het beste gerealiseerd kan worden. Uit dit onderzoek kwam de Microsoft Graph api als beste naar voren. Met Microsoft Graph kan de applicatie Microsoft-accounts gebruiken om onder andere evenementen toe te voegen aan de agenda en mails te sturen (Microsoft, 2022). Het nadeel van Microsoft Graph is dat het tijd kost om dit te leren. Verder heeft het gebruik van

Microsoft-accounts een gevolg voor het testen. Om de applicatie goed te kunnen testen moet er toegang zijn tot meerdere accounts, wat niet mogelijk is wanneer er Microsoft-accounts gebruikt worden. Ik heb namelijk alleen toegang tot mijn eigen Microsoft-account.

In overleg is uiteindelijk een andere manier gekozen, namelijk iCalendar bestanden. Door iCalendar bestanden gebruiken kan ook een nieuw evenement toegevoegd worden aan de Outlook agenda. Er is dan verder geen koppeling met Outlook accounts. Als gevolg hiervan wordt het sturen van mails functionaliteit weggelaten.

Het doel van de activiteitenapplicatie is dat het een overzicht van alle activiteiten bevat. In de oorspronkelijke situatie was er namelijk geen overzicht. Er was ook geen algemene plaats waar activiteiten opgeslagen werden in een database. Sommige activiteiten waren enkel vastgelegd in een Excel bestand.

6.4 Ontwerp

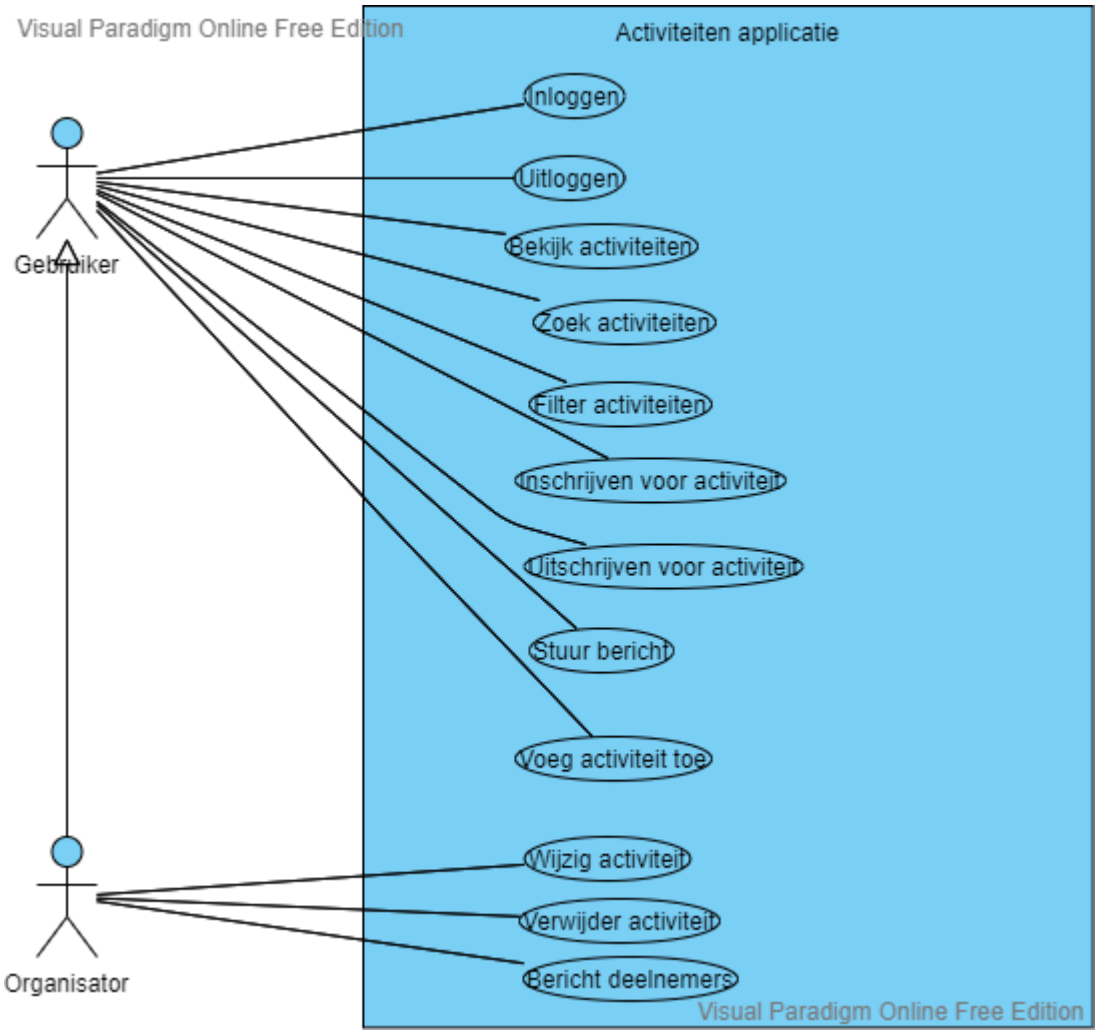
Nadat het onderzoek klaar was is de ontwerpfase begonnen. Zoals beschreven in de vorige paragraaf was al besloten dat de backend in .NET geschreven zou worden. Op basis van het resultaat van het onderzoek is verder besloten om Angular in de front-end te gebruiken, wat als gevolg heeft dat de front-end verder in TypeScript geschreven zal worden.

De start van het ontwerp was een meeting met de stakeholders, waarin de volgende requirements werden gedefinieerd:

- De applicatie bevat een overzicht van events/activiteiten.
- Gebruikers moeten per mail een bevestiging krijgen nadat zij zich opgegeven hebben voor een activiteit.
- De organisator van een activiteit moet een update kunnen versturen.
- De applicatie moet zich houden aan de huisstijl van Ordina.
- De applicatie moet alle activiteiten van de MTech afdeling bevatten, maar in de toekomst uitgebreid kunnen worden met andere afdelingen.
- Er moet gefilterd kunnen worden op afdeling.
- De applicatie moet scalable zijn.
- De activiteiten moeten een type hebben, zoals fun of cursus.

Behalve deze requirements heb ik zelf nog requirements opgesteld. De volledige lijst van requirements is te vinden in bijlage 3 Software Architectuur Document. Dit document bevat verder ook nog informatie over de architectuur van de applicatie. Er is hier bijvoorbeeld ook nagedacht over de kwaliteit van de applicatie. Om de kwaliteit te waarborgen moesten er een aantal kwaliteitseisen opgesteld worden. Deze worden in dit verslag meer in detail besproken in paragraaf 6.5 Realisatie.

Op basis van de requirements was de volgende stap om de use cases te bepalen. Deze zijn in detail beschreven in bijlage 4 Use Case Model en bijlage 5 Use Case Specification. Het use case diagram ziet er als volgt uit.



Figuur 4 Use case diagram

In dit diagram is ook te zien dat er 2 gebruikers zijn: de normale gebruiker en de organisator. Je ziet dat de organisator alle use cases van de normale gebruiker heeft plus nog een aantal extra activiteiten. Hieruit is op te maken dat de organisator een speciaal soort gebruiker is. Iedere gebruiker kan ook een organisator worden. Dit gebeurt zodra de gebruiker een activiteit toegevoegd heeft. In dat geval kan de gebruiker de use cases van de organisator uitvoeren voor deze activiteit.

De uiteindelijke lijst van use cases die opgesteld waren aan het begin van het project was als volgt:

Code	Use Case Naam	Prioritering
UC1	Inloggen	M
UC2	Uitloggen	M
UC3	Bekijk activiteiten	M
UC4	Zoek activiteiten	S
UC5	Filter activiteiten	S
UC6	Inschrijven voor een activiteit	M
UC7	Uitschrijven voor een activiteit	M

UC8	Stuur bericht	S
UC9	Voeg activiteit toe	M
UC10	Wijzig activiteit	M
UC11	Verwijder activiteit	S
UC12	Bericht deelnemers	S
UC13	Bekijk activiteit	M
UC14	Bekijk agenda	S
UC15	Bekijk ingeschreven activiteiten	S
UC16	Bekijk georganiseerde activiteiten	S

Voor elk van deze use cases zijn ook use case beschrijvingen gemaakt. Een voorbeeld van een beschrijving is als volgt.

Use-case naam	Bekijk activiteiten
Goal in context	Een lijst met activiteiten inzien.
Preconditions	De gebruiker is ingelogd.
Successful End Condition	Een lijst met activiteiten wordt getoond.
Failed End Condition	Er worden geen activiteiten getoond.
Actor	Gebruiker
Trigger	De gebruiker wil een lijst met activiteiten zien.
Main Flow	1. De gebruiker klikt op 'Home'. 2. Het systeem opent de homepagina. 3. Het systeem haalt de activiteiten op uit de database. 4. Het systeem toont de activiteiten op de pagina.
Extensions	3.1a Het systeem kan geen activiteiten ophalen uit de database. 3.1b De melding 'Er kunnen geen activiteiten opgehaald worden.' wordt op het scherm getoond.

Figuur 5 Use case beschrijving

Deze use case beschrijving gaat over de use case 'Bekijk activiteiten'. Het beschrijft alle eisen waar de use case aan moet voldoen. De informatie over de use cases is uiteindelijk verdeeld over 2 documenten: het use case model en de use case specification.

In het use case model is een korte beschrijving van elke use case te vinden. Elke use case heeft een eigen code en prioritering gekregen. Verder benoemt dit document ook nog welke actors er in de applicatie zijn. Er zijn weinig wijzigingen in dit document geweest in de loop van het project. Welke use cases er allemaal waren voor het project is namelijk al vrij vroeg bepaald. Het is wel een aantal keer voorgekomen dat er enkele nieuwe use cases toegevoegd zijn. Bijvoorbeeld bij het verwerken van de 'bekijk activiteit' functionaliteit bleek dat hier nog geen use case voor geschreven was.

Op basis van het use case model is daarna de use case specification geschreven. Dit document gaat meer in detail op de use cases in. Elke use case wordt hierin uitgewerkt. Ten eerste

is er voor elke use case een activiteitendiagram gemaakt. Dit activiteitendiagram gaat over het basisscenario van de use case. Dit basisscenario en de bijbehorende alternatieve scenario's en foutscenario's van de use cases worden ook gegeven. De use case specification is een document dat veel veranderde in de loop van het project. Het opzetten van het document is gebeurd voor de eerste sprint. Hierbij zijn ook de scenario's van alle use cases geschreven. De activiteitendiagrammen daarentegen zijn steeds geschreven in de sprint waarin de use case uitgevoerd werd. Het document is dus elke sprint aangevuld.

Het is natuurlijk ook belangrijk dat de applicatie goed werkt. Daarom zijn voor alle use cases ook testen geschreven. Deze testen zijn vastgelegd in bijlage 7 Test Plan. In dit document is alle belangrijke informatie met betrekking tot het uitvoeren van de testen te vinden. Zo wordt hierin bijvoorbeeld aangegeven wat voor soort testen er uitgevoerd worden en wat voor tools daarvoor nodig zijn. Voor elke use case is er vervolgens minimaal 1 test geschreven. Deze testen zijn geschreven op basis van de scenario's die te vinden zijn in de use case specification. Als voorbeeld worden hier de testen voor de use case 'Bekijk activiteiten' getoond. Voor deze use case zijn de 2 volgende testen geschreven:

Test Case ID	3	Test Case Beschrijving	Bekijk activiteiten	Versie	1.0
Datum Uitgevoerd		Resultaat			
Voorwaarden	De gebruiker is ingelogd				
Stap #	Test Stap	Verwachtte Resultaat	Werkelijke Resultaat	Status (Geslaagd/Gefaald)	Notities
1	Klik op [Home]	Het systeem toont de homepagina met een lijst met activiteiten.			

Tabel 1 Testplan voor bekijk activiteiten

Test Case ID	3.1	Test Case Beschrijving	Bekijk activiteiten – ophalen activiteiten mislukt	Versie	1.0
Datum Uitgevoerd		Resultaat			
Voorwaarden	De gebruiker is ingelogd				
Stap #	Test Stap	Verwachtte Resultaat	Werkelijke Resultaat	Status (Geslaagd/Gefaald)	Notities

1	Klik op [Home]	Het systeem toont de homepagina met de melding 'Er kunnen geen activiteiten opgehaald worden'			

Tabel 2 Testplan voor bekijk activiteiten foutscenario

De eerste testcase heeft hier betrekking op het basisscenario van de use case. De tweede test heeft betrekking op het foutscenario.

Naast de uml diagrammen zijn er ook schermontwerpen gemaakt. Deze zijn te vinden in bijlage 6 Schermontwerpen. Ten eerste is er hiervoor bepaald welke pagina's er nodig zijn. Vervolgens is nagedacht over welke informatie op deze pagina's getoond moet worden. Voor de meeste schermen zijn er meerdere ontwerpen gemaakt. Vervolgens is voor elke pagina gekeken welk scherm het beste is op basis van de volgende eigenschappen:

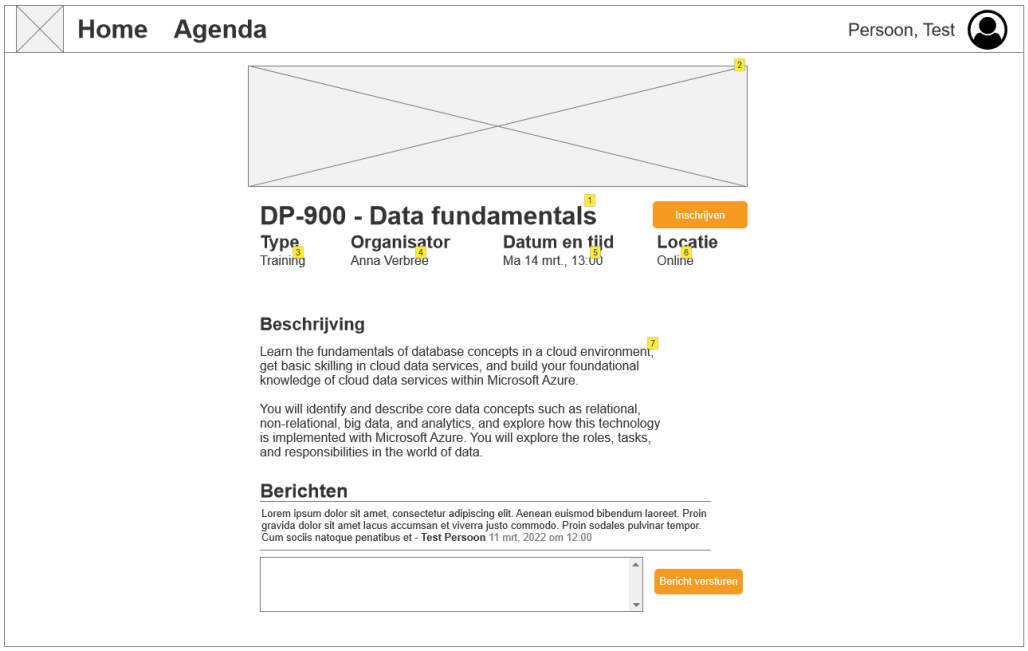
- Overzichtelijk
- Begrijpbaar
- Aantrekkelijk

Uiteindelijk is er voor elke pagina één definitief ontwerp uitgekozen.

Alle schermen moeten voldoen aan de huisstijlregels van Ordina, die een aantal eisen aan applicaties stellen. Ten eerste moet er een specifiek lettertype gebruikt worden en wordt er 'je' gezegd i.p.v. 'u'. Ook zijn er eisen over kleuren. Zo worden de kleuren oranje en blauw veel gebruikt.

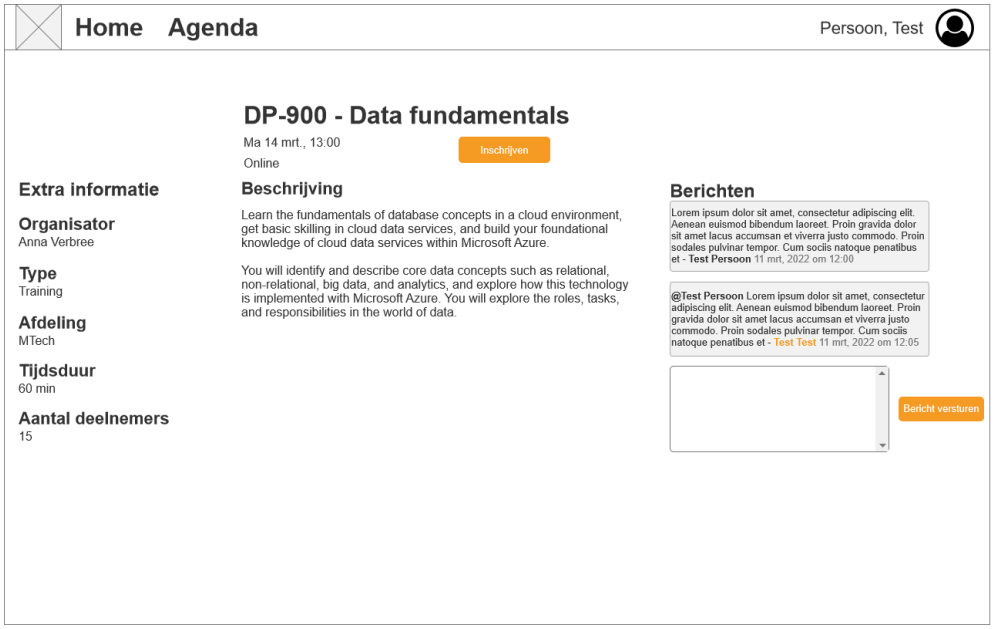
Een belangrijke eis die aan de applicatie gesteld was, was dat deze responsive moest zijn. De applicatie moet er dus goed uit zien op beeldschermen van verschillende grootten, zoals laptop schermen en telefoonschermen. Om dit voor elkaar te krijgen is er voor elk scherm een standaard ontwerp gemaakt, maar ook een ontwerp voor kleinere schermen.

Een van de schermen waar het van groot belang was om rekening te houden met groottes van de beeldschermen was het activiteit scherm. Het oorspronkelijke ontwerp van dit scherm zag er als volgt uit:



Figuur 6 Eerste ontwerp activiteit scherm

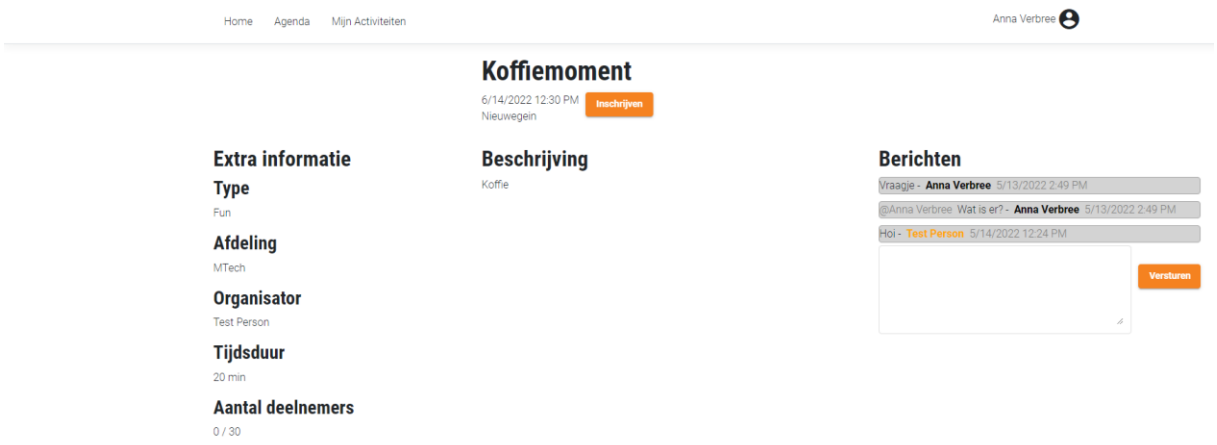
Zoals je kan zien op dit scherm heeft een activiteit veel informatie. Dit neemt veel ruimte in op het scherm. Tijdens een van de voortgangsmeetings is ook als feedback gegeven dat het activiteit scherm toch niet heel duidelijk is. Onder andere was hier het advies om de afbeelding kleiner te maken of weg te halen. Ook is er een tip gegeven om de informatie, zoals het type en de organisator, ergens onder elkaar op de pagina te tonen. Aan de hand van deze informatie is later een nieuw ontwerp gemaakt. Dit ontwerp ziet er als volgt uit:



Figuur 7 Definitief ontwerp voor activiteit scherm

Dit nieuwe ontwerp is tijdens een voortgangsmeeting besproken met de stakeholders. Zij hebben dit ontwerp goedgekeurd en gaven nog wat feedback. Zo waren bijvoorbeeld de kopjes extra informatie en berichten lager geplaatst dan het kopje beschrijving. Hierop werd als feedback gegeven dat dit meer op één lijn moet liggen. Deze feedback is ook in het ontwerp verwerkt.

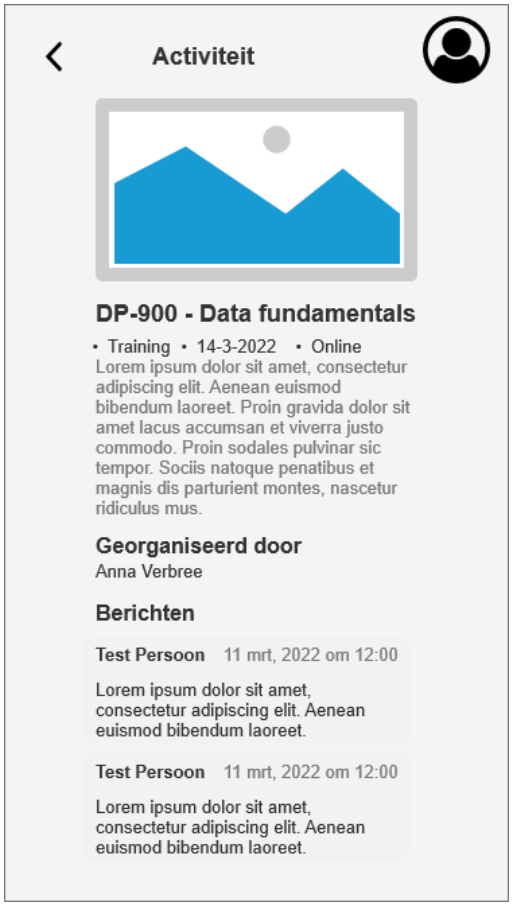
Het uiteindelijke activiteit scherm zag er als volgt uit:



Figuur 8 Definitieve activiteit scherm

Zoals te zien is komen alle onderdelen van het schermontwerp terug in het uiteindelijke scherm.

Om deze informatie op deze manier op een telefoonscherm te tonen zou de tekst heel klein gemaakt moeten worden. Dit heeft een negatief effect op de leesbaarheid. Daarom is het volgende ontwerp gemaakt voor kleinere schermen:



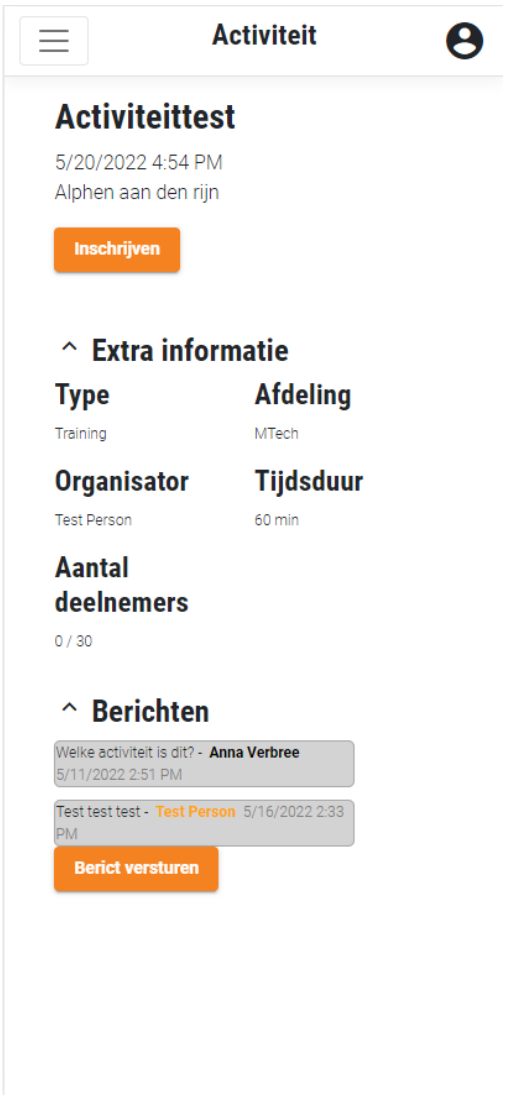
Figuur 9 Ontwerp activiteit scherm voor kleine schermen

Dit ontwerp is veel duidelijker voor kleine schermen. Een aantal van de kopjes zijn weggehaald, zodat alleen de belangrijke informatie te zien is.

Net als voor het standaard activiteit scherm werd hier als feedback gegeven dat deze manier van informatie weergeven niet helemaal duidelijk is voor de gebruikers. Als reactie op deze feedback is uiteindelijk het volgende ontwerp gemaakt:

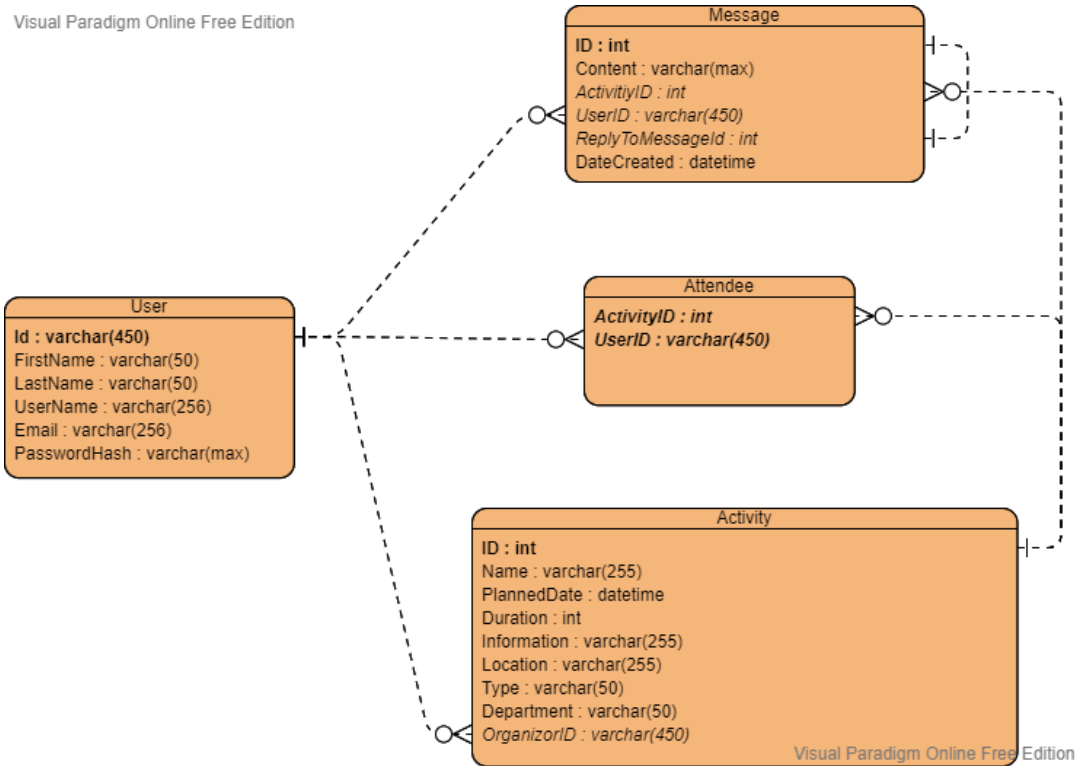
Figuur 10 Nieuw ontwerp activiteit scherm voor kleine schermen

De belangrijkste wijziging hierbij is het weghalen van de afbeelding. Hierdoor ontstaat er veel meer ruimte om de informatie van de activiteit weer te geven. Dit schermontwerp is uiteindelijk ook goedgekeurd door de stakeholders. Het uiteindelijke scherm zag er als volgt uit:



Figuur 11 Definitieve activiteit scherm voor kleine schermen

Voor de applicatie moesten de activiteiten ook ergens opgeslagen worden. Het probleem met activiteiten binnen Ordina was onder andere dat er geen algemene database was waarin alle activiteiten vastgelegd waren. Daarom is er nagedacht over op welke manier de data opgeslagen moest worden voor de activiteitenapplicatie. Om er zeker van te zijn dat de database alle informatie over de activiteiten bevat die nodig is, is ervoor gekozen om een nieuwe database op te zetten voor de applicatie. Aangezien er in Ordina veel met Azure gewerkt wordt, is ervoor gekozen om in Azure een SQL-database op te zetten. Hiervoor is er ten eerste een ER diagram opgezet, op basis waarvan de uiteindelijke database is opgezet. Dit diagram ziet er als volgt uit:



Figuur 12 ER diagram activiteitenapplicatie

Dit diagram gaat er van uit dat de users ook in de database opgeslagen worden. Het oorspronkelijke idee was dat de applicatie de Ordina Outlook accounts zou gebruiken. In dit geval hoeven de email en het wachtwoord van de gebruiker natuurlijk niet in deze database opgeslagen worden. In dat geval zou de user tabel alleen de id en email van de gebruikers bevatten. Een andere mogelijkheid zou zijn om niet gebruik te maken van user id's, maar in plaats daarvan de email van de gebruikers te gebruiken, aangezien deze altijd uniek is.

In de loop van het project is uiteindelijk besloten om de applicatie niet te koppelen aan de Outlook accounts, wat als gevolg had dat de gebruikers toch in de database opgeslagen worden.

De activiteiten die voor de start van dit project aangemaakt zijn moet natuurlijk ook overgezet worden naar de nieuwe database. Aangezien alle activiteiten op verschillende plaatsen en manieren opgeslagen zijn, is dit niet makkelijk in een keer over te zetten. De beste manier om de activiteiten over te zetten is door dit handmatig in de applicatie te doen.

6.5 Realisatie

Naast het ontwerpen en analyseren van de applicatie is er ook tijd besteed aan het realiseren van de applicatie. Dit gebeurde parallel aan het ontwerpen. Zoals beschreven zou de eerste week van elke sprint gebruikt worden om aan de ontwerpen te werken en zou de tweede week gebruikt worden om te realiseren. Dit was in de praktijk iets anders dan gepland. Het bleek dat de ontwerpen vaak in een paar dagen al klaar waren. Als gevolg hiervan was de verdeling vaak: een halve week ontwerpen en anderhalve week realiseren. In feite is er dus de meeste tijd besteed aan realiseren.

Doordat elke sprint begon met ontwerpen kon de realisatie hier makkelijk op aansluiten. Een aantal belangrijke inputs voor de realisatie waren de use case beschrijvingen en de activiteiten diagrammen.

De eerste stap van de realisatie was het opzetten van de database. Op basis van het gemaakte ER diagram is er een SQL-database opgezet in Azure. De backend van de applicatie spreekt vervolgens deze database aan met Entity Framework. Dit maakte het mogelijk om data toe te voegen en te verwijderen zonder query's te schrijven. In plaats daarvan is er gebruik gemaakt van een database context. Voor het toevoegen van een nieuwe activiteit kon dan gewoon de add functie aangeroepen worden op de context. Dit heeft als gevolg dat de code korter is en dat er minder snel fouten gemaakt kunnen worden. De functie in de backend voor het toevoegen van een activiteit ziet er als volgt uit:

```
[Authorize]
[HttpPost("add")]
1 reference | Anna Verbree, 55 days ago | 1 author, 4 changes
public Activity Add(Activity activity)
{
    var currentUserId = HttpContext.User.FindFirstValue(ClaimTypes.NameIdentifier);

    activity.OrganizorID = currentUserId;
    activity.AttendeeList = new List<Attendee>();

    _activityRepository.AddActivity(activity);
    _activityRepository.SaveChanges();
    return activity;
}
```

Figuur 13 Toevoegfunctie aan de backend

De eerste tabellen in de database waren in Azure toegevoegd door SQL-statements uit te voeren, waarna de modellen aangemaakt zijn in de applicatie. Dit wordt ook wel een database first aanpak genoemd. Later is dit omgedraaid en zijn eerst de modellen aangemaakt. Met behulp van migrations zijn vervolgens de tabellen voor deze modellen toegevoegd aan de database. Deze aanpak wordt code first genoemd. Hieronder volgt een voorbeeld van een migration:

```
1 reference | Anna Verbree, 28 days ago | 1 author, 1 change
public partial class AddMessageDateCreated : Migration
{
    0 references | Anna Verbree, 28 days ago | 1 author, 1 change
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.AddColumn<DateTime>(
            name: "DateCreated",
            table: "Message",
            type: "datetime2",
            nullable: false,
            defaultValue: new DateTime(1, 1, 1, 0, 0, 0, 0, DateTimeKind.Unspecified));

        migrationBuilder.AddColumn<int>(
            name: "ReplyTo",
            table: "Message",
            type: "int",
            nullable: true);
    }

    0 references | Anna Verbree, 28 days ago | 1 author, 1 change
    protected override void Down(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropColumn(
            name: "DateCreated",
            table: "Message");

        migrationBuilder.DropColumn(
            name: "ReplyTo",
            table: "Message");
    }
}
```

Figuur 14 Migration voor message tabel

Bij de realisatie is natuurlijk ook rekening gehouden met de kwaliteit van de applicatie. Hiervoor zijn de ISO-25010 kwaliteitseisen geraadpleegd. De toepassing van deze kwaliteitseisen zijn in detail beschreven in het Software Architectuur Document en zullen daarom in dit verslag in het kort beschreven worden.

Er is voor de applicatie rekening gehouden met de volgende 3 kwaliteitseisen:

- Bruikbaarheid
- Betrouwbaarheid
- Beveiliging

Voor de bruikbaarheid is het van belang dat de gebruikers van de applicatie goed gebruik kunnen maken van de applicatie. Het moet duidelijk voor ze zijn hoe zij de applicatie moeten gebruiken.

Voor de betrouwbaarheid is het van belang dat de applicatie blijft werken in het geval van software fouten. Dit wordt gerealiseerd met foutafhandeling. Het moet ook duidelijk zijn voor de gebruiker wat er mis is. Een voorbeeld hiervan is te zien bij het inloggen. Als het wachtwoord dat de gebruiker invoert niet correct is moet dit duidelijk gemaakt worden aan de gebruiker. Om dit te bereiken heb ik ten eerste de backend een foutmelding laten terugsturen in het geval het wachtwoord of de email verkeerd ingevoerd is. Dit ziet er als volgt uit:

```
[HttpPost("login")]
3 references | Anna Verbree, 17 days ago | 1 author, 2 changes
public async Task<IActionResult> Login([FromBody] UserLogin userLogin)
{
    var user = await _userManager.FindByNameAsync(userLogin.Email);

    if (user == null)
    {
        return NotFound();
    }

    if (!await _userManager.CheckPasswordAsync(user, userLogin.Password))
    {
        return Unauthorized();
    }

    var signingCredentials = _jwtHandler.GetSigningCredentials();
    var claims = _jwtHandler.GetClaims(user);
    var tokenOptions = _jwtHandler.GenerateTokenOptions(signingCredentials, claims); ;
    var token = new JwtSecurityTokenHandler().WriteToken(tokenOptions);

    return Ok(new { IsAuthSuccesful = true, Token = token });
}
```

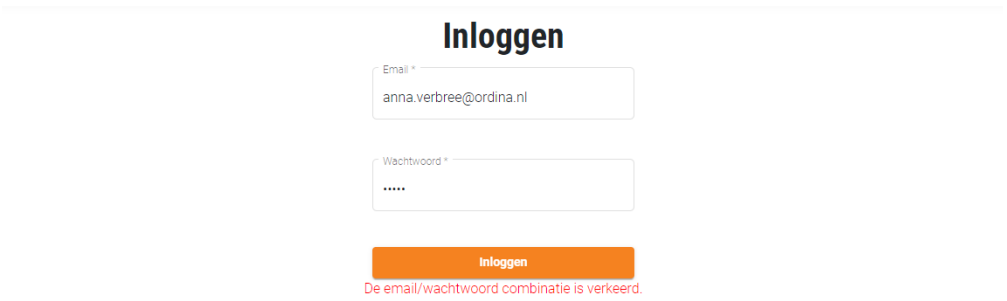
Figuur 15 Inlogfunctie aan de backend

Je ziet in dit voorbeeld dat als de wachtwoord check mislukt er een unauthorized error teruggestuurd wordt aan de front-end. Deze error geeft in dit geval aan dat de gebruiker geen toegang heeft tot dit account, aangezien het verkeerde wachtwoord opgegeven is. De front-end moet dan vervolgens deze foutmelding afhandelen en een bericht op het scherm tonen. Dit ziet er als volgt uit in de front-end:

```
this.authService.login(this.loginForm.value).subscribe(result => {
  localStorage.setItem("token", result.token);
  this.authService.sendAuthStateChangeNotification(result.isAuthSuccesful);
  this.isLoading = false;
  this.router.navigateByUrl('/');
}, error => {
  this.hasError = true;
  if (error.status == 404) {
    this.errorMessage = "Er is geen account met dit emailadres.";
  } else {
    this.errorMessage = "De email/wachtwoord combinatie is verkeerd.";
  }
  this.isLoading = false;
  console.log(error);
});
```

Figuur 16 Inlogfunctie aan de front-end

Je ziet hier dat er gecontroleerd wordt of er een error teruggestuurd wordt door de login functie. In het geval dat er een error is wordt er ook gekeken wat voor error dit is. De unauthorized error die de backend terugstuurt heeft als statuscode 400. De errormessage is dan “De email/wachtwoord combinatie is verkeerd.”. Op het scherm ziet dit er als volgt uit:



Figuur 17 Foutmelding voor verkeerd wachtwoord

Zoals je hier op het scherm ziet wordt de foutmelding getoond onder de inlogknop. Op deze manier is duidelijk voor de gebruiker wat er mis is gegaan en blijft de applicatie nog steeds werken.

Voor de beveiliging is het van belang dat gebruikers niet elkaars informatie kunnen bereiken. Dit wordt gedaan door elke gebruiker een eigen account te geven. Elk account bevat een lijst met activiteiten waar de specifieke gebruiker zich voor ingeschreven heeft en een lijst van activiteiten die de gebruiker aangemaakt heeft. Deze lijsten zijn niet door andere gebruikers in te zien. De hele applicatie, op de inlogpagina na, is alleen zichtbaar voor ingelogde gebruikers.

Een ander punt van veiligheid waar rekening mee gehouden is, is cross-site scripting (XSS). XSS maakt het mogelijk om client-side scripts te injecteren in web pagina's (Wikipedia, 2022). Op deze manier kan mogelijk kwaadwillende JavaScript code uitgevoerd worden. Dit is natuurlijk niet gewenst.

Angular beschermt standaard tegen XSS. Het behandelt alle waarden als niet vertrouwd. Niet vertrouwde code wordt niet uitgevoerd (Google, 2022). In de activiteitenapplicatie is informatie vaak op het scherm getoond doormiddel van interpolatie. Een voorbeeld van interpolatie op de activiteit pagina is als volgt:

```
<div *ngIf="showInfo" fxLayout.lt-sm="row wrap">
  <div fxLayout="column" class="info-box">
    <h3>Type</h3>
    <p>{{activity.type}}</p>
  </div>

  <div fxLayout="column" class="info-box">
    <h3>Afdeling</h3>
    <p>{{activity.department}}</p>
  </div>

  <div fxLayout="column" class="info-box">
    <h3>Organisator</h3>
    <p>{{organizorName}}</p>
  </div>

  <div fxLayout="column" class="info-box">
    <h3>Tijdsduur</h3>
    <p>{{activity.duration}} min</p>
  </div>

  <div fxLayout="column" class="info-box">
    <h3>Aantal deelnemers</h3>
    <p>{{activity.attendeeList.length}} / {{maxAmountAttendees}}</p>
  </div>
</div>
```

Figuur 18 Interpolatie op de activiteit pagina

Zoals je ziet in deze afbeelding is er zes keer interpolatie toegepast. Dit is aangegeven met accolades. Hierbinnen worden de waardes getoond die bij de variabelen horen. In het geval een van de waarden een script bevat wordt dit automatisch weggehaald door Angular. Ditzelfde geldt voor html code.

Bij de realisatie is er ook rekening gehouden met toekomstige aanpassingen. Voor het afstudeerproject is besloten dat de applicatie is voor activiteiten voor de MTech afdeling. Hierbij was het wel belangrijk dat er in de toekomst meerdere afdelingen gebruikt moeten kunnen worden in de applicatie. Om deze reden is ervoor gekozen om bij het aanmaken van een activiteit de afdeling in te voeren met een drop down. Voor nu is MTech de enige afdeling in de drop down, maar in de toekomst kunnen hier meer afdelingen aan toegevoegd worden. Deze toekomstige aanpassingen kunnen verder mogelijk gemaakt worden door nieuwe afdelingen toe

te voegen aan de database. In dit geval zouden de mogelijke afdelingen voor een activiteit opgehaald moeten worden uit de database, wat als gevolg heeft dat dit altijd up to date is.

Tijdens de realisatiefase is er ook tijd genomen voor het uitvoeren van testen. In de eerste ontwerpfase was hiervoor een testplan opgesteld, dat elke ontwerpfase uitgebreid is met testscenario's. Aan het einde van elke sprint zijn deze testen uitgevoerd en samengebracht in testrapporten in Excel. Een voorbeeld van een uitgevoerde test is als volgt.

Test Case ID	1	Test Case Beschrijving	Inloggen	Versie	1.0
Datum Uitgevoerd	13-4-2022	Resultaat	Geslaagd		
Voorwaarden	De gebruiker is niet ingelogd				
Stap #	Test Stap	Verwachtte Resultaat	Werkelijke Resultaat	Status (Geslaagd/Gefaald)	Notities
1	Voer de inloggegevens in.				
2	Klik op [Inloggen]	Het systeem logt de gebruiker in en toont de homepagina.	Zie verwachtte resultaat	Geslaagd	

Test Case ID	1.1	Test Case Beschrijving	Inloggen - inloggen mislukt	Versie	1.0
Datum Uitgevoerd	13-4-2022	Resultaat	Gefaald		
Voorwaarden	De gebruiker is niet ingelogd				
Stap #	Test Stap	Verwachtte Resultaat	Werkelijke Resultaat	Status (Geslaagd/Gefaald)	Notities
1	Voer de verkeerde inloggegevens in.				
2	Klik op [Inloggen]	De verkeerd ingevoerde gegevens worden rood omlijnd.	Foutmelding 401	Gefaald	Er wordt niet op het scherm getoond wat verkeerd is.

Figuur 19 Testen voor inloggen

Zoals je ziet zijn deze testen gebaseerd op de use case beschrijvingen en komen ze overeen met de testen in het testplan. Elk testrapport bevatte uitgevoerde testen die van belang waren voor de nieuwe functionaliteiten van de betreffende sprint. Vooraan elk testrapport stond een overzicht van het aantal uitgevoerde testen en hoeveel er geslaagd waren. Dit zag er bijvoorbeeld als volgt uit:

Uitgevoerd	Geslaagd		14
	Gefaald		1
Use Case	% Geslaagd	% Gefaald	
Inloggen	50%	50%	
Uitloggen	100%	0%	
Inschrijven voor activiteit	100%	0%	
Zoek activiteiten	100%	0%	
Filter activiteiten	100%	0%	
Bekijk activiteiten	100%	0%	
Voeg activiteit toe	100%	0%	
Bekijk activiteit	100%	0%	

Figuur 20 Overzicht uit testrapport

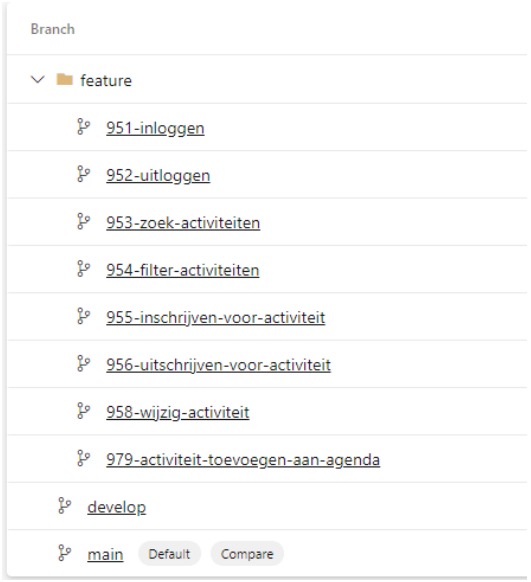
Deze testrapporten bevatten alleen de resultaten van de handmatige testen. Dit is niet het enige soort test dat uitgevoerd is. Er waren namelijk ook unit testen geschreven. Deze testen zijn gebruikt om de functionaliteiten te testen. Hiervoor zijn er unit testen geschreven voor de front-end en backend. Hieronder volgt een voorbeeld van een unit test.

```
[Fact]
0 references | Anna Verbree, 35 days ago | 1 author, 1 change
public void GetRegisteredActivitiesForUser_ReturnsActivities()
{
    // Arrange
    var activityRepositoryMock = new Mock<IActivityRepository>();
    activityRepositoryMock
        .Setup(r => r.GetRegisteredActivitiesForUser("1"))
        .Returns(GetTestActivities());

    var user = new ClaimsPrincipal(new ClaimsIdentity(new Claim[]
    {
        new Claim(ClaimTypes.Name, "1"),
        new Claim(ClaimTypes.NameIdentifier, "1"),
        new Claim("custom-claim", "example claim value"),
    }, "mock"));

    var activityController = new ActivityController(activityRepositoryMock.Object);

    activityController.ControllerContext = new ControllerContext()
    {
        HttpContext = new DefaultHttpContext() { User = user }
    };

    // Act
    var activities = activityController.GetRegisteredActivitiesForUser();

    // Assert
    activityRepositoryMock.Verify(r => r.GetRegisteredActivitiesForUser("1"));
    Assert.Equal("Online", activities[0].Location);
    Assert.Equal("Nieuwegein", activities[1].Location);
    Assert.Equal(2, activities.Count);
}
```

Figuur 21 Unit test voor het ophalen van geregistreerde activiteiten voor gebruiker

Met behulp van een pipeline in Azure DevOps werden deze testen steeds automatisch uitgevoerd als de nieuwe code naar de develop branch gepusht werd.

In Azure DevOps was ook de git repository te vinden waarin alle geschreven code opgeslagen is. Om ervoor te zorgen dat bij het werken aan een nieuwe functionaliteit er geen bugs geïntroduceerd werden in de al werkende functionaliteiten, is er met verschillende branches gewerkt. De branches hadden de volgende structuur:



Figuur 22 Git branches

In dit bovenstaande voorbeeld zijn ten eerste de develop en main branch te zien. Verder zijn er een aantal feature branches. Voor elke use case is een feature branch gemaakt waarin deze use case gerealiseerd werd. Wanneer de use case voltooid was, werd de branch gemerged

met de develop branch. Dit zorgde er vervolgens voor dat de pipeline gestart werd. Deze pipeline zag er als volgt uit:

```yaml
trigger:
- main
- develop

pool:
  vmImage: ubuntu-latest

steps:
  
Settings
- task: Npm@1
  inputs:
    command: 'install'
    workingDir: '$(Build.SourcesDirectory)/ActiviteitenApplicatie/ClientApp'

# perform unit-tests
- script: |
    npm test
  workingDirectory: '$(Build.SourcesDirectory)/ActiviteitenApplicatie/ClientApp'
  displayName: 'perform unit tests'

Settings
- task: PublishTestResults@2
  inputs:
    testResultsFormat: 'JUnit'
    testResultsFiles: '**/TESTS*.xml'
  displayName: 'publish unit test results'

Settings
- task: DotNetCoreCLI@2
  inputs:
    command: 'test'
    projects: '**/*Tests/*.csproj'
    arguments: '--configuration $(buildConfiguration) --collect "Code coverage"'
```

Figuur 23 Pipeline voor develop branch

Deze pipeline werd specifiek gebruikt voor de geschreven unit testen. Hij begint met het uitvoeren van de unit testen voor de front-end. Vervolgens voert hij de unit testen voor de backend uit. Het resultaat van deze testen legt hij vast in een xml bestand. Als laatste haalt hij de code coverage op. In het geval een van de testen faalde zorgde dat ervoor dat de pipeline ook faalde. Als dit gebeurde moest de code zo aangepast worden dat de testen wel slaagden en werd de pipeline opnieuw uitgevoerd.

Een andere pipeline is verantwoordelijk voor het builden van de applicatie en het publishen van een build artifact. Deze pipeline wordt automatisch uitgevoerd op het moment dat er code naar de main branch gepusht wordt. Het voltooien van de pipeline is een trigger voor de release pipeline.

De release pipeline is verantwoordelijk voor het deployen van de applicatie op de opgezette Azure server. Hiervoor gebruikt de pipeline de eerder gemaakte build artifact.

In paragraaf 6.4 Ontwerp is besproken welke use cases opgesteld zijn. Hieronder volgt deze lijst van use cases opnieuw, met in het groen aangegeven welke use cases uiteindelijk voltooid zijn.

Code	Use Case Naam	Prioritering
UC1	Inloggen	M
UC2	Uitloggen	M
UC3	Bekijk activiteiten	M
UC4	Zoek activiteiten	S

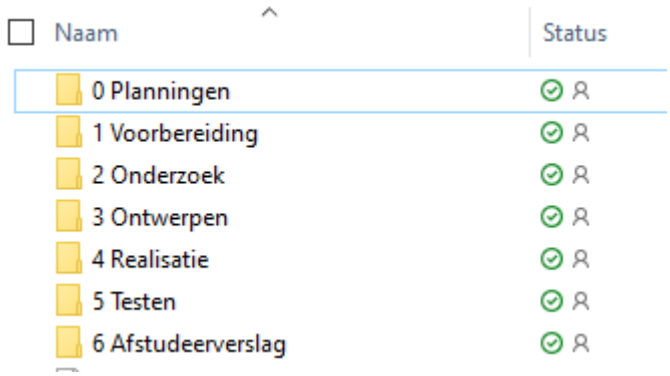
UC5	Filter activiteiten	S
UC6	Inschrijven voor een activiteit	M
UC7	Uitschrijven voor een activiteit	M
UC8	Stuur bericht	S
UC9	Voeg activiteit toe	M
UC10	Wijzig activiteit	M
UC11	Verwijder activiteit	S
UC12	Bericht deelnemers	S
UC13	Bekijk activiteit	M
UC14	Bekijk agenda	S
UC15	Bekijk ingeschreven activiteiten	S
UC16	Bekijk georganiseerde activiteiten	S

Zoals te zien zijn op twee na alle use cases voltooid. Use case 12 is uiteindelijk achtergelaten, omdat besloten was om de applicatie niet te verbinden met de Ordina accounts. Aangezien de applicatie geen toegang heeft tot de accounts, kan het geen mails sturen. Om deze reden is ook use case 6 veranderd. Als onderdeel van deze use case wordt de activiteit toegevoegd aan de agenda van de gebruiker. Oorspronkelijk zou dit automatisch gaan, omdat de applicatie gekoppeld zou zijn aan de Ordina accounts. Uiteindelijk is ervoor gekozen om iCalendar bestanden te gebruiken. Als gevolg is het toevoegen aan de agenda niet automatisch, maar moet de gebruiker het iCalendar bestand openen om de activiteit toe te voegen.

7 REFLECTIE

7.1 Algemeen

Het afstudeerproject is grotendeels zonder problemen verlopen. Naar mijn idee heeft het feit dat ik al eerder een poging heb gedaan om af te studeren hier een grote invloed op gehad. Hierdoor wist ik wat ik kon verwachten tijdens mijn stage. Ook heeft dit mij veel geleerd over hoe ik een overzicht kon bewaren tijdens mijn stage. Zo heb ik bijvoorbeeld geleerd hoe ik het beste de structuur waarin ik mijn bestanden opsla kon indelen. Dit zag er als volgt uit:



Figuur 24 De gebruikte mappenstructuur

Zoals te zien zijn de mappen per fase van het project ingedeeld. De cijfers in de namen van de mappen zijn gebruikt om aan te geven op welke volgorde de mappen getoond moeten worden.

Een ander voorbeeld is het bijhouden van een logboek. Hoe ik dit toegepast heb beschrijf ik in paragraaf 6.1 Aanpak.

Iets anders dat mij geholpen heeft bij het uitvoeren van mijn afstudeerstage was het feit dat Ordina trainingen organiseert voor stagiairs. Deze trainingen gingen allemaal over onderwerpen die nuttig waren voor mijn afstudeerstage. De trainingen waar ik het meest aan gehad heb gingen over focus en presenteren. De focus training leerde mij hoe ik het beste mijn focus kon verbeteren. Dit heb ik veel toe kunnen passen tijdens de hele stage. De training over presenteren leerde mij onder andere hoe ik het beste een presentatie in kan delen en wat de beste houding voor het presenteren is. Dit is iets dat ik goed kan gebruiken voor mijn eindpresentatie.

Tijdens de uitvoer van het project heb ik behalve aan mijn begeleiders ook de applicatie laten zien aan andere werknemers van de MTech afdeling. Deze mensen waren over het algemeen erg enthousiast, wat als gevolg had dat ik ook steeds enthousiaster werd.

7.2 Proces

Voordat ik begon aan deze afstudeerstage heb ik gekeken hoe ik het beste onderzoek uit kon voeren. Hiervoor heb ik een boek over onderzoek gelezen dat ik eerder tijdens mijn studie gebruikt had (Verhoeven, 2014). Dit heeft mij heel veel geholpen met het opzetten van het onderzoek. Het hielp vooral bij het opzetten van het onderzoeksplan en onderzoeksrapport.

Tijdens mijn afstudeerproces zijn er een aantal momenten geweest waarop het proces afweek van de planning. Dit kwam als eerste voor bij het onderzoek. Voor het onderzoek had ik drie weken ingepland. Uiteindelijk had ik vier weken nodig. Deze verlenging had een aantal redenen. Ten eerste moest kwam ik tijdens het vooronderzoek begrippen tegen die ik nog niet kende. Hiervan moest ik de betekenis nog opzoeken en vastleggen, wat tijd kostte.

Ten tweede was het vaak lastig om informatie te vinden over de functionaliteiten van de frameworks. De handleidingen bevatte vaak niet de nodige informatie, waardoor er extra tijd nodig was om andere websites te zoeken die wel deze informatie bevatten.

De langere duur van het onderzoek had verder niet heel grote gevolgen op de rest van de afstudeerstage. De oorspronkelijke planning was ruim genoeg om de rest van de ingeplande taken een week op te schuiven.

De tweede afwijking van de planning kwam naar voren bij het uitvoeren van mijn sprints. Zoals ik in het proces beschreven had was het hier de bedoeling dat ik in de eerste week van elke sprint mij bezig zou houden met het maken van ontwerpen en het schrijven van testen. In de eerste sprint is dit goed gegaan. Voor de tweede sprint heb ik het ontwerpen en schrijven van testen overgeslagen. In deze sprint had ik mij volledig gefocust op de realisatie en het maken van unit testen. Dit had uiteindelijk een vrij groot effect op de derde sprint. Aan het begin van de derde sprint kwam ik erachter dat mijn ontwerpen niet volledig waren. Dit had als gevolg dat ik een gedeelte van de sprint bezig geweest ben met het bijwerken van de ontwerpen en ik dus minder tijd had voor wat voor die sprint ingepland was. Ook ben ik hier bezig geweest met het schrijven en uitvoeren van testen. Een aantal van deze testen bleken uiteindelijk niet te slagen, omdat ik de use case beschrijvingen niet goed genoeg bekeken had tijdens de vorige sprint. Zo kwam ik erachter dat ik bijvoorbeeld geen foutmeldingen geïmplementeerd had die op het scherm getoond moesten worden als activiteiten niet opgehaald konden worden.

Deze tweede afwijking had de grootste impact. Ik merkte hier dat door de ontwerpen en testen op het einde te maken ik snel dingen vergat. Hiervan heb ik geleerd dat het toch heel belangrijk is om te beginnen met het maken van ontwerpen. Deze les heb ik in de derde sprint geleerd, waardoor ik dit in deze sprint en de erop volgende sprints goed heb kunnen toepassen.

7.3 Sprints

In deze paragraaf zal ik korte reflecties schrijven op elke sprint. In totaal zijn er 8 sprints uitgevoerd. Sprint 1 begint in de zesde week van de afstudeerstage. De periode daarvoor zal ik hier beschrijven als Sprint 0.

Sprint 0 (1 feb – 4 mrt)

Sprint 0 begon met het inwerken. Het was even wennen om aan de slag te gaan. De eerste twee weken heb ik mij beziggehouden met het schrijven van het afstudeerplan en het opstellen van de requirements.

Vanaf de derde week is het onderzoek begonnen. Hiervoor had ik oorspronkelijk drie weken ingepland. Dit bleek uiteindelijk toch iets te weinig tijd te zijn. Uiteindelijk ben ik vier weken bezig geweest met het onderzoek.

Sprint 1 (7 mrt – 18 mrt)

Deze sprint begon met het maken van UML-diagrammen. Hierbij ben ik de hele eerste week bezig geweest. De tweede week begon met het maken van schermontwerpen.

Na het opstellen van alle ontwerpen kon ik voor het eerst beginnen met realiseren. Omdat ik in het begin van de tweede week nog schermontwerpen aan het maken was had ik hier minder tijd voor dan gepland. Ik kwam er ook achter dat het ingewikkeld was om de applicatie op te zetten. Het goed opzetten van de git repository kostte meer tijd dan verwacht. Verder had ik nog geen ervaring met Azure, waarin ik de database wilde opzetten. Deze beiden dingen hadden als gevolg dat ik niet alle user story's afkreeg deze sprint.

Sprint 2 (21 mrt – 1 apr)

In deze sprint ben ik verdergegaan met de user story's van vorige sprint. Nu het opzetten van de git repository en de database gedaan was, merkte ik dat ik veel sneller voortgang maakte met de user story's. Aan het einde van de sprint ben ik ook bezig geweest met de unit testen. Ik had in het verleden voornamelijk unit testen geschreven in Java, wat als gevolg had dat ik in het begin veel naar de Angular documentatie heb moeten kijken. Gelukkig ben ik hier niet veel tijd aan kwijt geraakt. Aan het einde van de sprint had ik alle ingeplande functionaliteiten af.

Sprint 3 (4 apr – 15 apr)

Aan het begin van deze sprint kwam ik erachter dat ik vorige sprint de ontwerpen niet bijgewerkt had. Ook had ik de handmatige testen niet geschreven en uitgevoerd. Dit had als gevolg dat ik veel meer tijd kwijt was aan het ontwerpen. Hierbij kwam ik er ook achter dat ik vergeten was om de juiste foutmeldingen te implementeren. Ik miste dus eigenlijk stukken van de geplande functionaliteit. Dit heeft me geleerd hoe belangrijk het is om ontwerpen te maken.

De tweede week van deze sprint ben ik bezig geweest met de realisatie. Dit had ik al vrij snel af. Voor deze sprint had ik dus eigenlijk te weinig ingepland. De rest van de sprint ben ik bezig geweest met testen. Hierbij kwam ik erachter dat een paar gedeelten van de applicatie niet meer werkten. Dit heb ik deze sprint weer werkend gekregen. Aan het einde van de sprint waren ook weer alle geplande functionaliteiten klaar.

Sprint 4 (18 apr – 29 apr)

Deze sprint ben ik goed begonnen met de ontwerpen. Dit was in een paar dagen klaar, waardoor ik meer tijd had voor de nieuwe functionaliteiten. Voor deze sprint heb ik mij voornamelijk gefocust op het responsive maken van de applicatie. Het was belangrijk dat de pagina's er goed uitzagen op beeldschermen van verschillende grootten. Het maken van schermontwerpen is iets dat ik vaak lastig vind. Dit was nu minder het geval. Ik had van een andere stagiair een tip gekregen over een website voor inspiratie, waar ik uiteindelijk heel veel aan gehad heb.

Voor deze sprint had ik ook ingepland om de applicatie aan Outlook te koppelen. Dit bleek lastiger dan ik verwacht had. De applicatie moet namelijk toegang hebben tot de Ordina accounts om bij de agenda te komen en mails te kunnen sturen. Mijn idee was om dit te doen met Microsoft Graph. Dit heeft veel mogelijkheden, maar is niet iets wat ik heel snel zou kunnen leren. Na onderzoek kwam ik erachter dat er andere manieren zijn om activiteiten toe te voegen aan de agenda van een gebruiker. Een van deze mogelijkheden is doormiddel van iCalendar bestanden. Als een gebruiker dit bestand downloadt en opstart, wordt automatisch de agenda van de gebruiker geopend. Hiervoor hoeft de applicatie geen toegang te hebben tot de Ordina accounts.

Na overleg met de stakeholders is uiteindelijk besloten om de koppeling met de Ordina accounts buiten de scope van het project te laten vallen vanwege tijdsgebrek. Het toevoegen van de activiteiten aan de agenda zal gedaan worden met iCalendar bestanden. Het versturen van mails zal verder ook buiten de scope vallen.

Dit is iets wat ik zelf heel jammer vindt. Enkel heb ik nog geen ervaring met Microsoft Graph, waardoor ik de tijd zou moeten nemen om hier mee leren om te gaan. Die tijd is er helaas niet.

Door iCalendar te gebruiken heb ik alsnog alle ingeplande functionaliteiten van deze sprint voltooid.

Sprint 5 (2 mei – 13 mei)

Deze sprint verliep alles ook heel goed. Het grootste gedeelte van de ingeplande functionaliteiten was in de eerste week klaar. De rest was klaar in de tweede week. Bij het uitvoeren van de handmatige testen kwam ik erachter dat er gedeelten van de applicatie niet meer werkten, waarop ik deze problemen opgelost heb. Aan het einde van de sprint werkte alles en waren de ingeplande functionaliteiten klaar.

Vanaf deze sprint heb ik de focus van mijn werkzaamheden ook meer op het afstudeerverslag gelegd.

Sprint 6 (16 mei – 27 mei)

Voor deze sprint had ik besloten dat ik het grootste gedeelte van de tijd zou besteden aan het afstudeerverslag. Daarom had ik voor de sprint niet veel nieuwe functionaliteiten ingepland. Het belangrijkste voor de applicatie was dat het activiteit scherm aangepast werd zodat het bij het nieuwe ontwerp paste. Dit kostte niet heel veel tijd.

Daarnaast ben ik bezig geweest om wat meer unit testen te schrijven, zodat de kwaliteit van de code beter gegarandeerd kon worden en de code coverage beter werd. Dit ging ook goed.

Sprint 7 (30 mei – 10 juni)

In sprint 7 zijn er geen nieuwe functionaliteiten gerealiseerd. De belangrijkste werkzaamheid in deze sprint was het werken aan het afstudeerverslag.

Naast het schrijven aan het afstudeerverslag heb ik mij weer beziggehouden met het schrijven van unit testen.

Sprint 8 (13 juni – 17 juni)

Het begin van sprint 8 heb ik mij beziggehouden met het voltooien van dit afstudeerverslag. Daarnaast is er een begin gemaakt aan de presentatie die gegeven zal worden tijdens de afstudeerzitting.

Na het inleveren van dit afstudeerverslag zal ik mij tijdens deze sprint focussen op de presentatie en het schrijven van de laatste unit testen.

7.4 Product

Vanaf het begin van de afstudeerstage had ik veel ideeën voor de activiteitenapplicatie. Op basis van deze ideeën zijn er uiteindelijk een aantal user story's opgesteld. Deze user story's

hebben aan de hand van Moscow een prioriteit gekregen. Het grootste gedeelte van deze user story's zijn ook daadwerkelijk voltooid. Een overzicht van de voltooide user story's is te vinden in paragraaf 6.5 Realisatie.

Het realiseren van de applicatie is ook goed verlopen. De meeste functionaliteiten waren eerder af dan verwacht. Een enkel moment liep ik vast met een functionaliteit. Dit was op bij het koppelen van de applicatie aan de Outlook accounts. Dit bleek lastiger te zijn dan oorspronkelijk verwacht. Hier ben ik dan ook wat langer mee bezig geweest dan verwacht. Uiteindelijk is in overleg besloten om dit op een andere manier te implementeren. Als gevolg hierdoor is het niet mogelijk om berichten via de mail te versturen.

Om de kwaliteit van de applicatie te waarborgen heb ik gestreefd om minimaal 80% van de applicatie te testen met unit testen. Dit heeft mij, samen met de handmatige testen, ook geholpen om in de gaten te houden of alle functionaliteiten nog werken.

7.5 Competenties

Een van de doelen van mijn afstudeeropdracht was om een aantal competenties aan te tonen. Dit waren A- en B-competenties.

7.5.1 A-competenties

De A-competenties waren als volgt:

Onderzoek
Een groot gedeelte van mijn afstudeerstage was onderzoek. Ten eerste heb ik onderzoek gedaan naar het beste framework voor de activiteitenapplicatie. Hiervoor had ik 3 weken ingepland. Uiteindelijk had ik hier meer tijd voor nodig. Voordat ik begon aan het onderzoek heb ik eerst een vooronderzoek gedaan om erachter te komen wat frameworks precies zijn en wat voor soort frameworks ik nodig had.

Op sommige punten was het lastiger dan verwacht om de frameworks met elkaar te vergelijken. Zo wilde ik oorspronkelijk zelf benchmarks maken. Helaas had ik met maar een paar van de frameworks ervaring, wat dit niet mogelijk maakt. Uiteindelijk had ik besloten om bestaande benchmarks te vergelijken. Hier bleek vervolgens dat er niet voor alle frameworks benchmarks waren. Mijn besluit hierbij was om te concluderen dat de frameworks waar geen benchmarks van waren niet geschikt waren.

Leren leren
In het begin van de afstudeerstage voelde ik mij nog een beetje onzeker over de opdracht. Tegen de tijd dat ik kon beginnen met programmeren was deze onzekerheid verdwenen. De positieve feedback die ik kreeg heeft me veel zelfvertrouwen gegeven.

Tijdens mijn stage heb ik ook een aantal trainingen gevolgd binnen Ordina. Deze trainingen gingen bijvoorbeeld over focus en presenteren. Vooral deze laatste training vond ik heel zinvol en gaf mij meer zelfvertrouwen voor mijn eindpresentatie.

Ook heb ik de mogelijkheid gehad om trainingen te volgen binnen de MTech afdeling. Dit waren Microsoft-trainingen die heel nuttig waren voor het uitvoeren van mijn opdracht. Zo heb ik een training gevolgd over Azure. Hier had ik nog nooit mee gewerkt. Aan de hand van deze training en het opzoeken van informatie op het internet heb ik wel het gevoel gekregen dat ik wist hoe ik Azure het beste kon gebruiken voor mijn opdracht.

Professioneel werken

Tijdens mijn afstudeerstage heb ik gewerkt volgens de planning die ik in mijn afstudeerplan opgesteld had. Deze planning is op twee wijzigingen na niet veranderd. De eerste wijziging was het verlengen van de tijd voor het onderzoek. Als gevolg hiervan heb ik ook de ingeplande taken die daarna volgden een week opgeschoven. Ten tweede kwam ik er op gegeven moment achter dat mijn planning eindigde voordat ik klaar was met mijn stage. Er waren nog 2 weken die ik niet ingepland had. Dit had als gevolg dat ik een extra sprint in kon plannen.

Zoals ik al eerder beschreven had heb ik verder volgens de agile methodiek gewerkt. Hierbij heb ik in meerdere sprints gewerkt die elk 2 weken duurden. Hiervoor was mijn planning om mij de eerste week bezig te houden met het maken van ontwerpen. Later kwam ik erachter dat ik niet zo veel tijd nodig had hiervoor. Als gevolg hiervan had ik dus eigenlijk extra tijd voor het realiseren, wat erg fijn was.

Om de twee weken heb ik een voortgangsmeeting gehad samen met de begeleiders en de twee andere stagiairs die op mijn afdeling stageliepen. In deze meetings heb ik steeds verteld en laten zien waar ik mee bezig geweest was in de huidige sprint en waar ik in de komende sprint mee aan de slag zou gaan. Hier heb ik ook steeds om feedback gevraagd en feedback gegeven aan de andere stagiairs.

Innovatie

Tijdens mijn stage heb ik op verschillende momenten zelf oplossingen moeten bedenken voor problemen. Een van de eerste voorbeelden hiervan is het onderzoek dat ik gedaan heb. Hierbij heb ik bedacht en onderzocht wat het beste framework was voor de activiteitenapplicatie.

Later in het project liep ik tegen problemen aan bij de koppeling met de Outlook accounts. Uiteindelijk heb ik hiervoor een andere oplossing bedacht: het gebruik van iCalendar bestanden. Dit was na overleg met de stakeholders goedgekeurd.

7.5.2 B-competenties

Behalve de A-competenties had ik zelf ook 3 B-competenties uitgekozen die ik wilde aantonen. Dit waren de volgende competenties:

Software analyseren

Een groot onderdeel van het analyseren was het doen van onderzoek. Ik heb naar verschillende aspecten onderzoek gedaan, ten eerste naar het meest geschikte framework. Dit is te vinden in paragraaf 6.2 Onderzoek en paragraaf 6.3 Analyse. In deze paragrafen beschrijf ik hoe dit

onderzoek aangepakt is en wat de resultaten waren. Verder heb ik ook nog kort onderzocht hoe ik het beste de applicatie aan de Ordina accounts kan koppelen. Een volledig verslag van het frameworks onderzoek is te vinden in bijlage 2 Onderzoeksrapport.

Na het onderzoek heb ik de requirements opgesteld. Hiervoor heb ik rekening gehouden met de eisen van de stakeholders. Verder heb ik zelf ook requirements opgesteld. Dit is beschreven in paragraaf 6.3 Analyse. Een volledig overzicht van alle requirements is te vinden in bijlage 3 Software Architectuur Document in hoofdstuk 4 Architecturele eisen. Op dezelfde locaties heb ik ook de acceptatiecriteria beschreven.

Als laatste onderdeel van de analyse heb ik gekeken naar de integratie en migratie van de applicatie. Hier was het vooral belangrijk om te kijken hoe de bestaande activiteiten overgeplaatst konden worden naar de applicatie. Dit is beschreven in paragraaf 6.3 Analyse. Een volledige uitleg over de integratie en migratie is te vinden in bijlage 3 Software Architectuur Document in hoofdstuk 8 Integratie en migratie.

Software ontwerpen

Het maken van de ontwerpen voor de applicatie is gebeurd volgens de RUP-methode. Het plan was om vóór de eerste sprint de nodige documenten op te zetten. Tijdens de eerste week van elke sprint heb ik vervolgens steeds de documenten bijgewerkt.

Meerdere malen heb ik ook om feedback gevraagd en gekregen feedback verwerkt. Een voorbeeld hiervan is dat ik het ontwerp van het activiteit scherm heb aangepast aan de hand van de gegeven feedback. Dit is in detail beschreven in paragraaf 6.4
Ontwerp. Het maken van de schermontwerpen was voor mij op sommige momenten wat lastig. Vaak heb ik moeite om te bedenken hoe ik mooie schermontwerpen kan maken. Een van de feedbackmomenten kreeg ik een tip voor een goede website voor inspiratie voor schermontwerpen. Deze website heeft mij veel inspiratie gegeven, waardoor ik ook meer zelfvertrouwen kreeg in het maken van schermontwerpen.

Ik heb ook een teststrategie opgesteld voor de activiteitenapplicatie. Dit is in het kort beschreven in paragraaf 6.4 Ontwerp. De volledige teststrategie is te lezen in bijlage 7 Test Plan in hoofdstuk 3 Teststrategie. Deze testen waren voor mij een houvast voor de realisatie van de applicatie. Zo heb ik hierin beschreven hoe het systeem moet reageren bij foutscenario's.

Als laatste is heeft de applicatie ook een architectuur. Deze beschrijf ik in het kort in paragraaf 6.4 Ontwerp. De architectuur is in detail uitgelegd in bijlage 3 Software Architectuur Document. Het lastigste hiervan was om te bepalen wat de packagestructuur van het project was en hoe de database eruit moest zien. Het lastige van de packagestructuur was voornamelijk dat ik dit nog niet eerder gemaakt had. Het lastige van de database was dat deze in de loop van het project meermaals grote aanpassingen had.

Software realiseren

Het realiseren van de activiteitenapplicatie is gebeurd aan de hand van verschillende ontwerpen. Om ervoor te zorgen dat de software goed paste bij de ontwerpen zijn er aan het eind van elke sprint testen uitgevoerd. Het verloop van deze sprints is beschreven in paragraaf 6.5 Realisatie.

Het vertalen van het ontwerp naar de realisatie verliep grotendeels zonder problemen. Een paar keer kwam het voor dat ik vergeten was om foutmeldingen correct op de pagina te tonen, maar dit gebeurde in de loop van het project steeds minder.

Om de kwaliteit van de applicatie te waarborgen is er ook rekening gehouden met een aantal kwaliteitsaspecten. Deze aspecten zijn beschreven in paragraaf 6.5 Realisatie. Ook zijn ze in detail beschreven in bijlage 3 Software Architectuur Document in hoofdstuk 3 Kwaliteitsaspecten.

Deze bijlage bevat verder ook een uitleg van de architectuur. Hierbij is onder andere een diagram weergegeven voor de database. De architectuur van de klassen is weergegeven in een klassendiagram.

Een belangrijke manier om de kwaliteit van de software te waarborgen was het uitvoeren van meerdere testen. Er zijn twee soorten testen uitgevoerd voor de applicatie. Deze worden beschreven in paragraaf 6.5 Realisatie. In dit paragaaf wordt ook beschreven hoe de testresultaten opgeslagen werden. Zoals ook hier beschreven worden de unit testen automatisch uitgevoerd door een pipeline. Een gedetailleerde beschrijving van de testen is te vinden in bijlage 7 Test Plan.

Het lastige van de unit testen was dat ik nog niet eerder unit testen geschreven had voor deze programmeertalen. Als gevolg heb ik meermaals online op moeten zoeken hoe ik dit het beste aan kon pakken. Gelukkig was hierover genoeg documentatie te vinden. Het voordeel voor de unit testen voor de backend was ook dat deze gedeeltelijk vergelijkbaar waren met unit testen voor Java.

Het beschikbaar stellen van de applicatie heeft plaatsgevonden met behulp van een pipeline. De unit testen worden automatisch uitgevoerd met een pipeline. Een andere pipeline is verantwoordelijk voor het builden van de applicatie. Verder is er een release pipeline die de applicatie deployt op de server. Deze Pipelines worden beschreven in paragraaf 6.5 Realisatie.

8 BIBLIOGRAFIE

Collaris, R.-A., & Dekker, E. (2011). *RUP OP MAAT.* Den Haag: Academic Service.

Dribbble. (2022). *Dribbble - Discover the World's Top Designers & Creative Professionals*. Opgehaald van Dribbble: https://dribbble.com/

Google. (2022, Februari 28). *Angular - Security*. Opgehaald van Angular: https://angular.io/guide/security#preventing-cross-site-scripting-xss

Microsoft. (2022). *Outlook calendar API overview - Microsoft Graph | Microsoft Docs*. Opgehaald van Microsoft: https://docs.microsoft.com/en-us/graph/outlook-calendar-concept-overview

Ordina. (2022). *High performance teams*. Opgehaald van Connect: https://ordinaweb.sharepoint.com/sites/PortfolioBusinessdevelopment/SitePages/High-performance-teams.aspx

Verhoeven, N. (2014). *Wat is onderzoek?* Den Haag: Boom Lemma.

Wikipedia. (2022, Juni 7). *Cross-site scripting*. Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Cross-site_scripting

9 EXTERNE BIJLAGEN

Voor dit project zijn er meerdere producten opgeleverd aan Ordina en de Hogeschool Leiden. Deze producten worden als externe bijlagen ingeleverd bij dit afstudeerverslag. Dit zijn de volgende externe bijlagen:

Bijlage 1: Onderzoeksplan
Bijlage 2: Onderzoeksrapport
Bijlage 3: Software Architectuur Document
Bijlage 4: Use Case Model
Bijlage 5: Use Case Specification
Bijlage 6: Schermontwerpen
Bijlage 7: Test Plan

Bijlage C: Afstudeerplan
Bijlage D: Adviesformulier afstudeerplan
Bijlage E: Adviesformulier voortgang
Bijlage F: Tussenevaluatie bedrijfsbegeleider
Bijlage G: Feedbackformulier afstudeerverslag
Bijlage H: Eindevaluatie bedrijfsbegeleider

BIJLAGE 1 AFSTUDEERVOORSTEL

1.1 Organisatorische aspecten

Mijn gegevens zijn:

Anna Verbree
Klepperman 38
2401 GJ Alphen aan den Rijn

E-mail: s1094208@student.hsleiden.nl
Telefoon: 06 36 31 34 05

Mijn specialisatie is Software Engineering en mijn slb'er is Ron Arts.

Het bedrijf waar ik ga afstuderen heet Ordina. Dit bedrijf zit in Nieuwegein, Groningen, Eindhoven en Amsterdam. De locatie waar ik ga afstuderen is in Nieuwegein. Het adres is:

Ringwade 1
3439 LM Nieuwegein

Mijn bedrijfsbegeleider is Wilco Morren.
Hij is bereikbaar op het emailadres wilco.morren@ordina.nl.
Telefonisch is hij bereikbaar op +31 610390930.

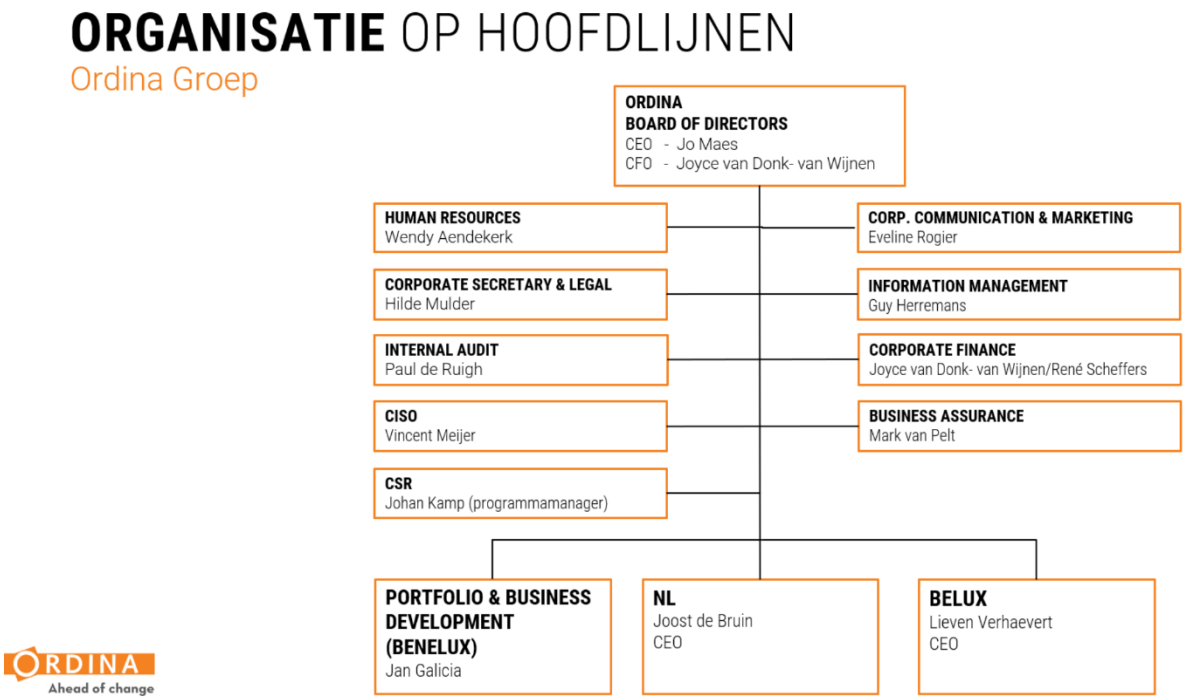
1.2 De context en achtergrond van het afstudeerbedrijf

De afstudeeropdracht zal plaatsvinden bij Ordina. Het bedrijf heeft 2.650 medewerkers en heeft vier locaties in Nederland, drie in België en één in Luxemburg. Ordina is de grootste onafhankelijke IT-dienstverlener in de Benelux.

De visie van Ordina is om doormiddel van IT hun klanten te helpen de veranderingen in de wereld voor te blijven, zodat zij voorbereid zijn op de uitdagingen die de toekomst gaat brengen. Hun missie is om samen met deze klanten duurzaam te innoveren. Om dit te verwezenlijken werkt Ordina met de volgende werkwijze: korte sprints met een flexibele inzet van specialisten gericht op snel resultaat boeken.

Binnen het bedrijf is er een mensgerichte cultuur, er is een participatieve instelling en teamwork is één van de belangrijkste aspecten. Ordina wil namelijk steeds meer met high performance teams gaan werken in plaats van alleen mensen te detacheren.

Onderstaande afbeelding geeft een beeld van de structuur van de organisatie.



FIGUUR 1 ORGANISATIESTRUCTUUR ORDINA

De afstudeeropdracht wordt uitgevoerd binnen de unit Microsoft Solutions (MTech) welke in het deel NL zit in de afbeelding hierboven en onder Operations valt. Dit houdt in dat zij werken met Microsoft-producten, zoals Azure en .NET. De medewerkers van MTech worden gedetacheerd naar de klant. Dit betekent dat de medewerkers vrijwel altijd bij de klant op locatie werken. De medewerkers die fysiek op de MTech afdeling bij Ordina zitten hebben momenteel geen opdracht. Zij zijn op zoek naar een opdracht of zijn bezig met het behalen van bepaalde certificaten. Eén van de Business Unit Managers van deze afdeling is Wilco Morren en hij zal de afstudeeropdracht begeleiden.

1.3 De probleem- of kans-beschrijving

Binnen Ordina vinden er vaak activiteiten plaats voor de werknemers. In de huidige oplossing is er op het intranet een klein plekje waar er een overzicht is van de activiteiten die plaats zullen vinden. Hier zijn alleen de bedrijfsbrede activiteiten te vinden. Het probleem dat hier plaats vindt is dat er geen goed overzicht is van alle activiteiten die georganiseerd zijn. De activiteiten zijn te vinden op verschillende locaties, wat als gevolg heeft dat de gebruiker veel moeite moet doen om erachter te komen welke activiteiten wanneer plaatsvinden.

Een ander probleem wat hier ontstaat is dat er zo meerdere activiteiten op hetzelfde moment ingepland kunnen worden. Als de activiteiten namelijk op meerdere plaatsen te vinden zijn, kan er niet makkelijk gekeken worden of er al iets op een bepaald moment ingepland is.

Als laatste kunnen de gebruikers niet makkelijk inzien voor welke activiteiten zij zich opgegeven hebben en wanneer die plaatsvinden. Dit zorgt ook weer voor een grotere kans dat mensen zich opgeven voor een activiteit, maar uiteindelijk niet op komen dagen. Verder is er ook een kans dat de werknemers zich opgeven voor activiteiten die op hetzelfde moment plaatsvinden.

1.4 Bedrijfsdoelstelling

Met het maken van de Activiteitenapplicatie wil Ordina verschillende doelen bereiken. Ten eerste is er een organisatorisch doel. Door alle activiteiten op een centrale plaats in de applicatie te plaatsen zal het makkelijker worden om nieuwe activiteiten in te plannen. De gebruiker kan dan in één oogopslag zien op welke momenten er al activiteiten ingepland zijn en zijn eigen activiteit inplannen op een moment waar nog geen activiteit is. Als gevolg hiervan kunnen er ook meer mensen naar de activiteit komen, omdat ze geen keuze hoeven te maken tussen meerdere activiteiten.

Een andere doelstelling is de vergroting van de gebruiksvriendelijkheid. Het maken van een applicatie waarin alle activiteiten op één centrale plek staan, maakt het veel makkelijker voor gebruikers om de activiteiten in te zien. Doordat gebruikers verder de mogelijke activiteiten kunnen sorteren op team en afdeling, kunnen gebruikers ook heel makkelijk de activiteiten vinden waar zij in geïnteresseerd zijn.

Als laatste zal het hierdoor ook mogelijk zijn om de activiteiten via de telefoon of tablet in te zien.

1.5 Concrete opdrachtomschrijving

Mijn opdracht is om een Activiteitenapplicatie te maken. In deze applicatie krijgt de gebruiker een overzicht te zien van alle activiteiten die ingepland zijn en wanneer ze plaatsvinden. Hierbij wordt het ook mogelijk om de activiteiten per team en afdeling te zien. Gebruikers kunnen verder ook nieuwe activiteiten voegen. Vervolgens kunnen gebruikers zich ook opgeven voor ingeplande activiteiten, waarna ze deze activiteit desgewenst aan hun Outlook agenda kunnen toevoegen. Als laatste zal ik een chatoptie toevoegen zodat het ook mogelijk is voor gebruikers om vragen te stellen aan de organisator van een activiteit.

De backend van deze applicatie ga ik realiseren in Microsoft .NET. De vorm van de front-end zal bepaald worden aan de hand van een onderzoek naar frameworks die voldoen aan de eisen van de applicatie en de organisatie. In dit onderzoek zal ik verschillende frameworks met elkaar vergelijken, om zo tot het meest geschikte framework te komen. De front-end moet schaalbaar zijn.

Ik begin met het opstellen van ontwerpen, waarmee ik daarna de applicatie ga bouwen aan zowel de backend als front-end zijde.

Ook zal ik, om de kwaliteit van de applicatie te waarborgen, verschillende testen uitvoeren op de applicatie.

1.6 Planning

Week	Werkzaamheden	Op te leveren producten
1	Afstudeerplan opstellen, requirements bepalen, analyseren huidige applicatie, bestuderen gebruikte technieken	
2	Afstudeerplan opstellen, interview stakeholders, frameworks onderzoek	Afstudeerplan (aan school)
3	Frameworks onderzoek	

4	Frameworks onderzoek	Backlog Onderzoeksverslag
5	Sprint 1 ontwerp	
6	Sprint 1 realisatie en testen	Sprint review en demo, detailontwerp
7	Sprint 2 ontwerp	
8	Sprint 2 realisatie en testen	Sprint review en demo, detailontwerp, test-verslag
9	Sprint 3 ontwerp	
10	Sprint 3 realisatie en testen	Sprint review en demo, detailontwerp, test-verslag
11	Sprint 4 ontwerp	
12	Sprint 4 realisatie en testen	Sprint review en demo, detailontwerp, test-verslag
13	Sprint 5 ontwerp	
14	Sprint 5 realisatie en testen	Sprint review en demo, detailontwerp, test-verslag
15	Sprint 6 ontwerp	Pre-final versie afstudeerverslag (aan school)
16	Sprint 6 realisatie en testen	Sprint review en demo, detailontwerp, test-verslag
17	Sprint 7 rework, testen en documenteren	Definitieve versie afstudeerverslag (aan school), sprint review, demo en testverslag, Activiteitenapplicatie incl. ontwerp/documentatie
18	Voorbereiden voor afstudeerzitting	
19	Voorbereiden voor afstudeerzitting	
20	Afstudeerzitting	

Vanaf week 3 zal ik wekelijks het afstudeerverslag gaan bijwerken.

1.7 Deliverables

Er zijn meerdere producten die ik uiteindelijk op zal leveren aan Ordina en school. Deze producten zijn:

- Afstudeerverslag
- Activiteitenapplicatie
- Ontwerpdocumenten
    - UML-diagrammen
    - Functioneel/technisch ontwerp
    - Schermontwerpen
- Technische documentatie (eventueel als onderdeel van het technisch ontwerp)
- Onderzoeksverslag
- Testverslag
- Sprint reviews

1.8 Verdediging A-Competenties

De eerste A-Competentie waar ik aan zal werken tijdens mijn afstudeerstage is **Onderzoek**. Zoals al eerder in mijn opdrachtomschrijving beschreven zal ik onderzoek doen naar verschillende frameworks. Hierbij zal ik de verschillende features van de frameworks vergelijken aan de hand van eisen die gelden voor de activiteitenapplicatie.

Het resultaat van dit onderzoek zal ik inleveren in de vorm van een verslag en bepaalt de uiteindelijke implementatie van de Activiteitenapplicatie, en wordt opgenomen in het afstudeerverslag.

De A-Competentie **Leren leren** zal ik aantonen door het leren omgaan met verschillende nieuwe technieken en tools, zoals Azure en verschillende mogelijke frameworks.

De A-Competentie **Professioneel werken** toon ik aan door met een gedegen onderzoek te beginnen en de resultaten van het onderzoek te gebruiken als basis voor het ontwerp van de applicatie. Gedurende mijn afstudeeropdracht zal ik regelmatig contact houden met de opdrachtgever over de voortgang. Denk daarbij ook aan tussentijdse presentaties waarbij de opdrachtgever en eventueel andere belanghebbenden of geïnteresseerden de mogelijkheid heeft feedback te geven. Ik sta open voor alle feedback en zal deze waar mogelijk meenemen in mijn opdracht.

De A-Competentie **Innovatie**: Als uit mijn onderzoek blijkt dat een techniek, tool of framework, welke nog niet gebruikt wordt binnen het stagebedrijf, het meest geschikt is voor het realiseren van de applicatie of een deel ervan, dan zal ik proberen de opdrachtgever te overtuigen om dit toe te passen.

1.9 Aan te tonen B-Competenties

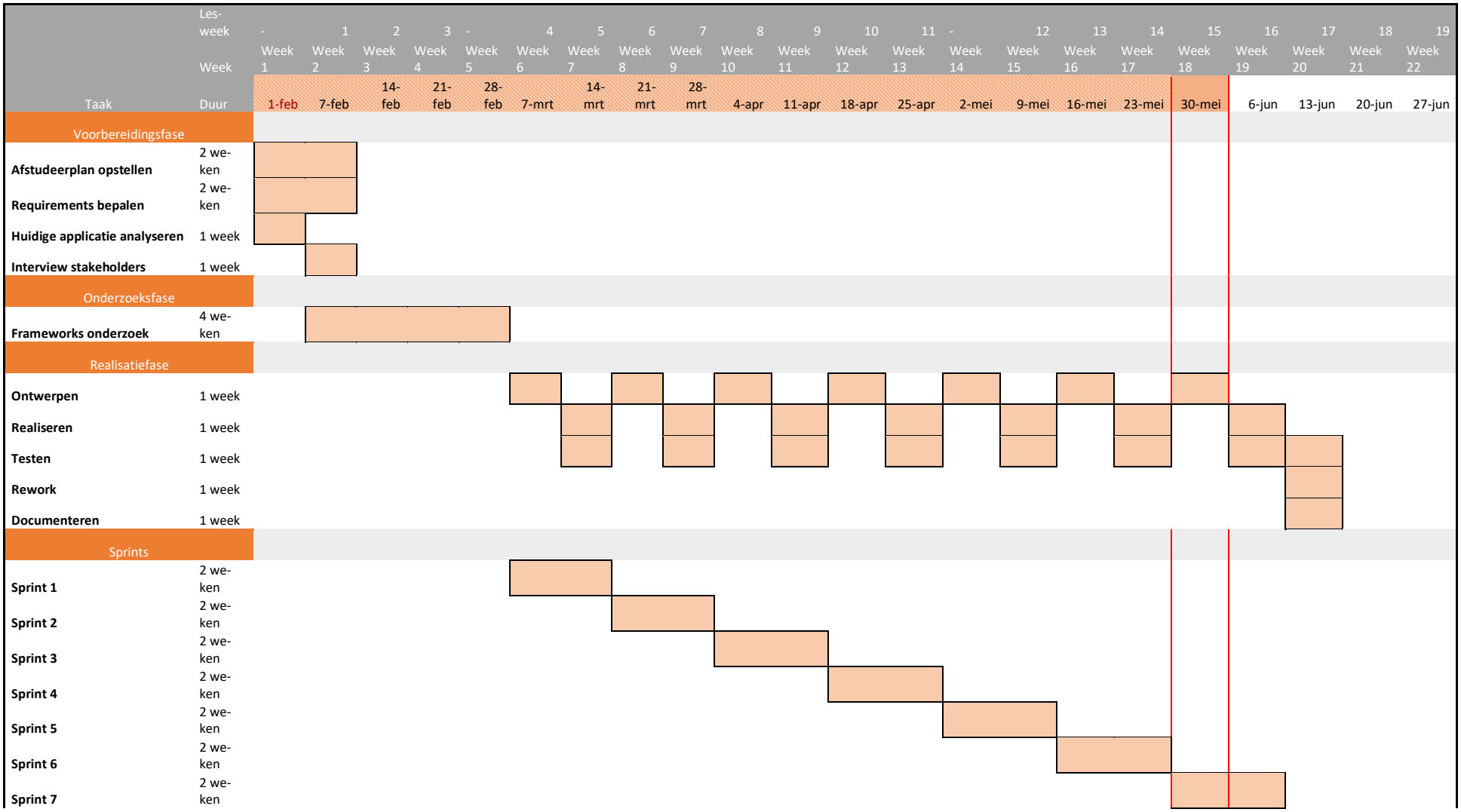
Ik heb de volgende drie B-Competenties gekozen.

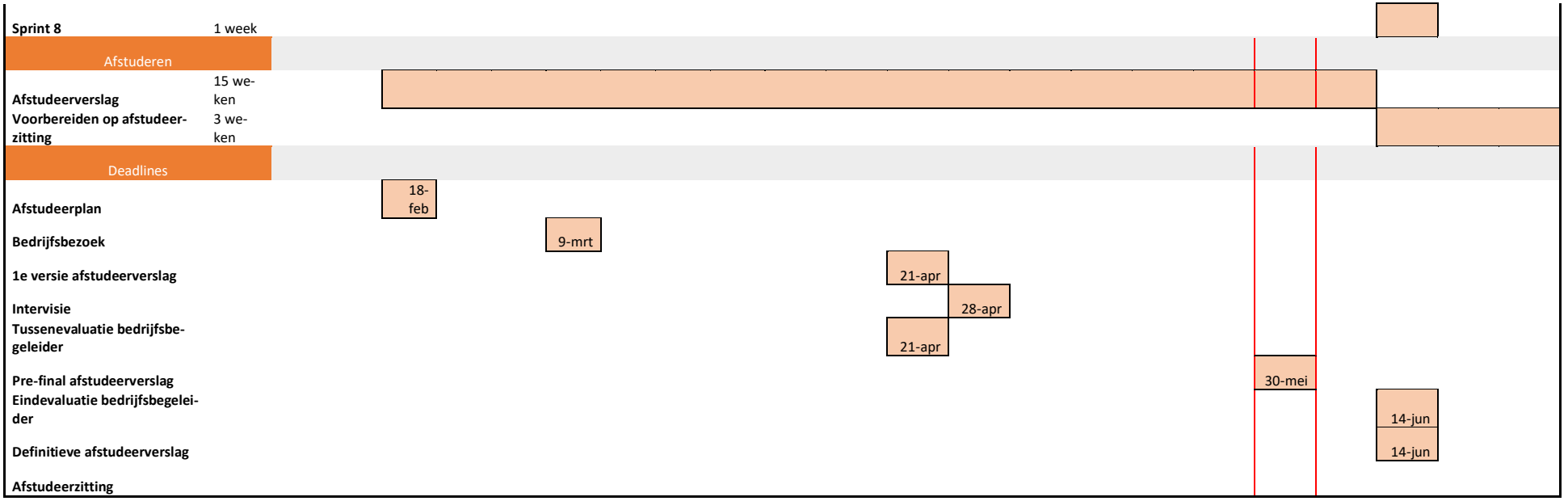
De eerste competentie is **software analyseren**. Ik zal voor mijn project moeten analyseren wat de precieze requirements zijn voor de activiteiten. Deze requirements zullen het uitgangspunt zijn voor mijn onderzoek naar het beste framework voor de front-end. Ook zal ik kijken hoe de huidige applicatie het beste gemigreerd kan worden naar de nieuwe applicatie. Hierbij zal ik natuurlijk ook kijken naar een eventuele migratie van de database.

De tweede competentie is **software ontwerpen**. Ik zal verschillende ontwerpen, zoals een functioneel/technisch ontwerp, UML-ontwerpen en schermontwerpen maken voordat ik begin met het realiseren van de applicatie. Ook zal ik verschillende testscenario’s en een teststrategie opstellen.

De derde competentie is **software realiseren**. Ik zal de activiteitenapplicatie bouwen op basis van het ontwerp volgens de kwaliteitsrichtlijnen van Ordina. De applicatie zal integreren met Outlook, zodat gebruikers activiteiten toe kunnen voegen aan hun agenda. Verder zal de applicatie mogelijk geïntegreerd worden met de applicatie van een andere stagiair binnen Ordina. Met behulp van de opgestelde testscenario’s en de teststrategie zal ik testen uitvoeren voor de applicatie. Hierbij probeer ik zo goed als mogelijk bij relevante veranderingen aan de applicatie deze testen opnieuw uit te voeren. Ook zal ik de applicatie laten testen door (toekomstige) gebruikers en waar nodig ook hun feedback verwerken. Aan het eind van mijn opdracht zal ik de applicatie gedocumenteerd opleveren aan Ordina.

BIJLAGE 2 PLANNING




Sprint 8
1 week
Afstuderen
Afstudeerverslag
15 weken
Voorbereiden op afstudeerzitting
3 weken
Deadlines
Afstudeerplan
18-feb
Bedrijfsbezoek
9-mrt
1e versie afstudeerverslag
21-apr
Intervisie
28-apr
Tussenevaluatie bedrijfsbegeleider
21-apr
Pre-final afstudeerverslag
30-mei
Eindevaluatie bedrijfsbegeleider
14-jun
Definitieve afstudeerverslag
14-jun
Afstudeerzitting