

08-10-2021



JEM-id

# Afstudeerdossier

## Realisatie van een contextuele chat binnen Floriday

School:

Opleiding:

Begeleidend examiner:

Expert examiner:

De Haagse Hogeschool

Software Engineering

Veld, G.E. (Gerda) in 't

Bosch, P (Pascal) van den

Afstudeerbedrijf:

Begeleider:

JEM-id

Zegveld, T.F.C. (Tim)

**Toni van Bergenhenegouwen**

**17063590**

## Referaat

**Naam:** Toni van Bergenhenegouwen

**Student nummer:** 1

**E-mail adres:** I

**Opleiding:** HBO ICT Software Engineering

**Opleidingsinstituut:** De Haagse Hogeschool

**Locatie:** Den Haag

**Afstudeerdatum:** 8 oktober 2021

**SLB'er:** Hoek, J.J. (Cobie) van der

**Begeleidend examiner:** Veld, G.E. (Gerda) in 't

**Expert examiner:** Bosch, P (Pascal) van den

**Afstudeerbedrijf:** JEM-id BV

**Bedrijfsbegeleider:** Zegveld, T.F.C. (Tim)

## Voorwoord

Voor u vindt u mijn afstudeerverslag “Realisatie van een contextuele chat binnen Floriday”. Deze afstudeeropdracht is uitgevoerd bij JEM-id en staat beschreven in dit afstudeerverslag. De ontwikkeling van mijn afstudeerverslag heeft plaatsgevonden van 10 mei 2021 tot en met 8 oktober 2021. De afstudeeropdracht is de meesterproef voor de opleiding HBO-ICT Software Engineering aan de Haagse Hogeschool. Dit dossier is bestemd voor de examinatoren en het afstudeerbedrijf.

In samenspraak met de opdrachtgever, Bas La Graauw en de bedrijfsbegeleider, Tim Zegveld, ben ik tot deze opdracht gekomen. Dankzij de kennis die ik gedurende mijn studie eigen heb gemaakt heb ik deze opdracht succesvol ontwikkeld.

Ik wil graag dit moment gebruiken om mijn examinatoren, bedanken voor het geven van feedback en beantwoorden van onduidelijkheden tijdens het schrijven van dit verslag.

Daarnaast wil ik graag mijn bedrijfsbegeleider bedanken voor de begeleiding en ondersteuning tijdens de ontwikkeling van het eindproduct.

Verder wil ik graag mijn opdrachtgever bedanken voor het faciliteren van deze opdracht en het visuele ontwerp. Tot slot wil ik het afstudeerbedrijf JEM-id bedanken voor de afstudeerplek en tevens ook alle collega's die mij tijdens de periode geholpen hebben waar nodig was.

Het doel van dit verslag is om de lezer te vermaken en beeld te schetsen over wat ik geleerd en uitgevoerd heb tijdens mijn afstudeerperiode. Ik wens u veel leesplezier!

Toni van Bergenhenegouwen

Honselersdijk, 8 oktober 2021

## Inhoudsopgave

1. Inleiding.....	1
2. Verklarende woordenlijst .....	2
3. Organisatieomschrijving .....	4
3.1 Het bedrijf .....	4
4. Opdrachtoomschrijving.....	6
4.1    Aanleiding .....	6
4.2    Probleemstelling .....	6
4.3    Doelstelling .....	6
4.4    Verwacht resultaat.....	6
5. Projectaanpak .....	8
5.1    Ontwikkelmethode .....	8
5.2    Globale planning .....	8
5.3    Werkzaamheden .....	9
6. Sprint 0: Onderzoeksfase .....	12
6.1    Interview .....	12
6.2    Visiedocument .....	13
6.3    Gebruik van Notion .....	16
6.4    Product backlog .....	17
6.5    Real-time chat mogelijkheden .....	18
6.6    Afweging tussen nieuwe of bestaande service.....	19
7. Sprint 1: Koper chat prototype .....	22
7.1    Sprint doel.....	22
7.2    User stories .....	22
7.3    Review.....	25
7.4    Retrospective .....	26
8. Sprint 2: Kweker chat prototype .....	27
8.1    Doel .....	27
8.2    User stories .....	27
8.3    Review.....	31
8.4    Retrospective .....	32
9. Sprint 3: Prototypes finetunen .....	33
9.1    Doel .....	33
9.2    User stories .....	33
9.3    Review.....	36
9.4    Retrospective .....	36

10. Sprint 4: Automatisch berichten ophalen .....	38
10.1 Doel .....	38
10.2 User stories .....	38
10.3 Review .....	40
10.4 Retrospective .....	41
11. Sprint 5: Vooraf gedefinieerde antwoorden voor kweker .....	42
11.1 Doel .....	42
11.2 User stories .....	42
11.3 Review .....	44
11.4 Retrospective .....	44
12. Sprint 6: Contextuele chat voor koper .....	46
12.1 Doel .....	46
12.2 User stories .....	46
12.3 Review .....	49
12.4 Retrospective .....	49
13. Gebruikerstesten .....	51
13.1 Uitvoering van de tests .....	51
13.2 Test resultaten .....	51
13.3 Verwerking van de resultaten.....	52
14. Aanbevelingen .....	53
15. Evaluatie Proces .....	54
16. Evaluatie Product .....	55
17. Verantwoording beroepstaken.....	57
Literatuurlijst.....	59
Bijlagen.....	61

## 1. Inleiding

Binnen de afstudeerperiode heb ik vanuit JEM-id een contextuele chat gerealiseerd voor het digitale platform Floriday. Hiervoor heb ik verschillende tussenproducten opgeleverd die door middel van mijn afstudeerverslag onderling een logische vorm krijgen.

In het afstudeerverslag zal er meerdere malen doorverwezen worden naar bijlagen die aan dit document gekoppeld staan. Het is dan niet per se relevant om de gehele bijlage door te lezen, wel zal hier meer informatie te vinden staan over het desbetreffende onderdeel.

Het is relevant om te beginnen met het lezen van de organisatiestructuur. Op deze manier zal het duidelijk worden hoe de organisatie opgebouwd is en wat mijn rol binnen de organisatie is. Hierna is het relevant om de opdrachtschrijving door te nemen, dit is dan ook meteen wat ik met JEM-id afgesproken heb om op te pakken. Hier is de aanleiding van het afstudeerproject te vinden, evenals de probleemstelling, doelstelling en het te verwachten resultaat.

In hoofdstuk 5 zal ik ingaan op de projectaanpak, die ik eerder in mijn afstudeerplan al deels beschreven heb. In dit hoofdstuk schrijf ik over de ontwikkelmethode Scrum die ik toe zal passen tijdens mijn project. Ook beschrijf ik hier mijn globale planning, werkzaamheden en de risico's binnen mijn project.

Hoofdstuk 7 tot en met 12 zal ingaan op de verschillende sprints met de doelen die ik gesteld heb. Voordat ik hieraan toe kom beschrijf ik in hoofdstuk 6 eerst mijn 'Sprint 0'. Hierin doe ik interviews met de opdrachtgever, waarna ik een visiedocument opstel. Ook schrijf ik hier over het gebruik van Notion, waarin ik mijn Product backlog heb opgesteld. Binnen Sprint 0 maak ik ook verschillende afwegingen. Zo bekijk ik de mogelijkheden voor het real-time chat gevoel en maak ik de afweging tussen een nieuwe of bestaande service.

Vanaf hoofdstuk 7 zal ik per sprint ingaan op het doel, de opgepakte user stories met de werkzaamheden hiervoor, de review en de retrospective. Zo bouw ik iedere sprint voort op wat ik geleerd heb uit de vorige sprint. In hoofdstuk 12 beschrijf ik dan ook mijn laatste sprint, sprint 6.

De sprints sluit ik af met een gebruikerstest, die ik in hoofdstuk 13 beschrijf. Aan de hand van deze gebruikerstesten maak ik nog een paar laatste wijzigingen aan het eindproduct. De uitkomsten van de gebruikerstest neem ik ook onder andere mee in de aanbevelingen, die ik in hoofdstuk 14 beschrijf. Hierin neem ik ook aanbevelingen mee die ik gedurende mijn afstudeertraject ben tegengekomen.

Vanaf hoofdstuk 15 ga ik in op de evaluatie. Hier begin ik met het evalueren op het proces die ik doorlopen heb tijdens mijn afstuderen. Hoofdstuk 16 gaat meer in op de evaluatie op mijn eindproduct, de contextuele chat en de bijbehorende tussenproducten. Tot slot beschrijf ik in hoofdstuk 17 de vooraf gestelde beroepstaken met hierin de werkzaamheden die ik heb gedaan om deze beroepstaken te volbrengen.

## 2. Verklarende woordenlijst

### **JEM-id:**

JEM-id is het afstudeerbedrijf gedurende dit afstudeerproject.

### **Floriday:**

Floriday is een dealmaking platform waar kopers en kwekers van bloemen en planten zaken met elkaar doen. Binnen het Floriday platform is dit afstudeerproject gerealiseerd.

Het platform Floriday is opgesplitst in twee aparte Front-End projecten, de customersportal (voor kopers) en de suppliersportal (voor kwekers).

### **RFH (Royal Flora Holland):**

RFH is het bedrijf wat samen met JEM-id het platform Floriday heeft opgericht. Royal FloraHolland is een coöperatie van kwekers binnen de sierteeltsector. Ze faciliteren onder andere de financiële stromen tussen kwekers en kopers, maar leveren ook services zoals transport en een open veiling van bloemen en planten.

### **Customersportal**

De customersportal is het Front-End project waar kopers van bloemen en planten inkopen kunnen doen bij verschillende kwekers op het platform.

### **Suppliersportal**

De suppliersportal is het Front-End project waar kwekers van bloemen en planten hun voorraad op kunnen beheren en verkopen.

### **Service**

Een service binnen Floriday is een kleine microservice die voornamelijk bestaat uit een API met verschillende endpoints. Elke service heeft zo zijn eigen verantwoordelijkheid en logica. De Front-End projecten en andere clients kunnen deze services aanroepen om zo data op te kunnen halen. De services zijn opgebouwd met de volgende technieken/talen: C#, .NET Core, Entity Framework Core (EF Core).

### **M&C service (Match & Connect service)**

De Match & Connect service is een service binnen Floriday die aangeroepen kan worden door de Front-End projecten en andere clients. Deze service is voornamelijk bedoeld voor de logica en informatie over de connectie tussen de kweker en de koper.

Deze service wordt gebruikt gedurende dit afstudeerproject.

### **Team Purple**

Team Purple is het Back-End ontwikkelteam binnen JEM-id dat zich bezighoudt met alle operationele werkzaamheden rondom de verschillende services.

### **Team Cyan**

Team Cyan is het Back-End ontwikkelteam dat nauw samenwerkt met team Purple. Team Cyan houdt zich ook bezig met de verschillende services, maar focust zich meer op de nieuwe ontwikkelingen.

### **Team Blue**

Team Blue is het Front-End ontwikkelteam dat zich richt op alle ontwikkelingen rondom beide Front-End projecten, de customersportal en de suppliersportal.

**Team Black**

Team Black is het cloud-engineering team dat zich bezighoudt met alle overige IT, security en integratie werkzaamheden.



### 3. Organisatieomschrijving

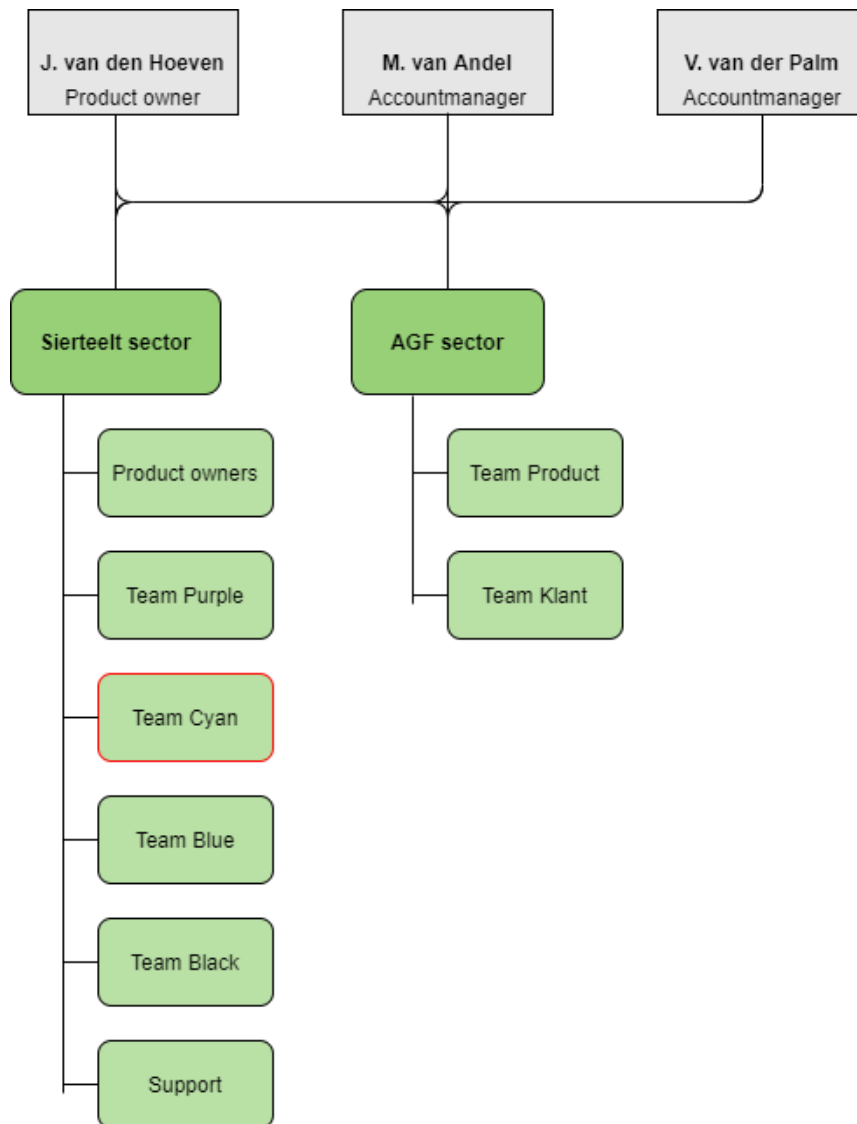
In dit hoofdstuk wordt er een duidelijk beeld geschept van de organisatie, de structuur en de werkzaamheden van de situatie waarin de afstudeeropdracht heeft plaatsgevonden.

#### 3.1 Het bedrijf

JEM-id is een softwarebedrijf dat zich richt op het ontwikkelen en beheren van softwarepakketten voor de sierteelt en AGF(Aardappelen, Groenten, Fruit) sector. JEM-id (JEM-id, 2021) bestaat al meer dan 25 jaar en is opgericht door Jefry en Martijn. Momenteel bestaat het bedrijf uit bijna 50 medewerkers die onderverdeeld zijn in 2 verschillende secties, de sierteelt- en AGF sector.

Gedurende de afstudeerperiode zal er deel uitgemaakt worden van de sierteelt sector. Het grootste project waar de sierteelt sector momenteel mee bezig is, is het platform Floriday. Floriday is in samenwerking met Royal FloraHolland (RFH) ontwikkeld (Royal FloraHolland, 2021). Floriday dient als een centraal platform waar kwekers en kopers van bloemen en planten hun aanbod kunnen beheren, aanbieden en bestellen. De afstudeeropdracht zal verwerkt worden in dit platform.

Om meer duidelijkheid te scheppen over de structuur en hiërarchie van het afstudeerbedrijf is hieronder een Organogram opgesteld.



*Figuur 1: Organogram van afstudeerbedrijf*

JEM-id heeft een relatief platte hiërarchische organisatiestructuur. In Figuur 1: Organogram van afstudeerbedrijf is te zien dat de hiërarchie verdeeld is over 2 sectoren, de sierteelt-en AFG sector. Boven deze lagen staan de 3 werkgevers beschreven die over beide sectoren gaan. Onder de 2 sectoren vallen alle werknemers die verdeeld zijn in multifunctionele teams.

Onder de sierteeltsector vallen de product owners, 4 ontwikkelteams en een support team.

Gedurende de afstudeerperiode zal ik onderdeel uitmaken van team Cyan.

Team Cyan is het ontwikkelteam dat nauw samenwerkt met team Purple. Team Cyan en Purple zijn verantwoordelijk voor de Back-End services. Waarvan team Cyan verantwoordelijk is voor de nieuwe ontwikkelingen en team Purple voor de operationele ontwikkelingen.

Team Blue is het ontwikkelteam dat zich richt op alle ontwikkelingen rondom beide Front-End projecten, de customersportal en de suppliersportal.

Team Black is het technische team dat zich bezighoudt met alle overige IT, security en integratie werkzaamheden.

De AGF sector bestaat uit team Product en team Klant. Team Klant bestaat uit supportmedewerkers en richt zich op de klant. Team Product bestaat uit ontwikkelaars en marketingmedewerkers en richt zich op de producten.

## 4. Opdrachtschrijving

In dit hoofdstuk wordt beschreven hoe de afstudeeropdracht tot stand is gekomen. Hierbij zal de probleemstelling, met bijbehorende doelstelling, beschreven worden, die opgepakt zal worden tijdens het afstudeertraject. Tot slot zal het verwachte resultaat geschetst worden.

### 4.1 Aanleiding

Het online platform Floriday (RFH, 2021) is in samenwerking met JEM-id en Royal FloraHolland opgericht. Floriday is een online platform voor kwekers en kopers van potplanten en snijbloemen. Kwekers kunnen met Floriday hun voorraad, aanbod en orders op één plek beheren. Kopers kunnen met Floriday het aanbod van verschillende kwekers bekijken en bestellen. Al deze processen vinden momenteel digitaal plaats via Floriday waar dagelijks zo'n 4000 kwekers en kopers, verspreid over 33 landen, gebruik van maken.

Doordat het gebruikersaantal van Floriday in een stijgende lijn toeneemt komt het steeds vaker voor dat kwekers en kopers elkaar persoonlijk niet kennen voordat ze zaken met elkaar doen via het platform. Dit maakt het lastig snel contact met elkaar op te nemen wanneer er bijvoorbeeld vragen zijn met betrekking tot het aanbod. Voorheen kende kwekers en kopers elkaar persoonlijk als zij zaken met elkaar deden. Hierdoor namen zij gemakkelijk contact met elkaar op buiten het platform door middel van bijvoorbeeld telefonie, E-mail, WhatsApp of andere communicatiemiddelen.

### 4.2 Probleemstelling

Floriday wordt wekelijks gebruikt door meer dan 4000 kwekers en kopers, verspreid over 33 landen. Op dit moment kan een kweker via Floriday zijn producten aanbieden, die de koper vervolgens kan bestellen. Ook kunnen de betalingen via Floriday plaatsvinden. Wanneer de koper en kweker onderling contact op willen nemen, over het aanbod of een bestelling, gebeurt dit nu via externe kanalen zoals e-mail of telefonisch. Deze exponentiële groei op Floriday zorgt ervoor dat niet iedere koper en kweker elkaar persoonlijk meer kent. Hierdoor is het lastig voor hen om te kunnen communiceren over een specifiek vraagstuk.

### 4.3 Doelstelling

Het doel van deze afstudeeropdracht is dat de onderlinge communicatie tussen koper en kweker digitaal, via het platform, mogelijk wordt. Door dit te digitaliseren hoeven koper of kweker elkaar niet persoonlijk te kennen om de communicatie te starten. Dit communicatiemiddel moet ook helpen bij het stellen van de juiste vragen en het geven van specifieke antwoorden. Zo hoeft er niet zelf een vraag of antwoord gedefinieerd te worden en blijft het gesprek gekaderd.

### 4.4 Verwacht resultaat

Aan het einde van het project is Floriday uitgebreid met een digitaal communicatiemiddel waardoor kopers en kwekers met elkaar kunnen communiceren over vraagstukken. Met de uiteindelijke oplossing wordt de koper geholpen met het stellen van vragen aan de kweker. Ook zal de kweker geholpen worden in het geven van de juiste antwoorden.

Deze oplossing zal succesvol geïntegreerd worden in het Floriday landschap dat bestaat uit verschillende losstaande .NET Core (Microsoft, 2021) services, API's en een tweetal React.JS (Facebook Inc., 2021) Front-End projecten.

Naast het digitale communicatiemiddel voor Floriday zal er aan het einde van dit project ook een afstudeerverslag opgeleverd worden met bijlagen zoals: Plan van Aanpak, Visiedocument, Product Backlog, Ontwerpdokumentatie en Testrapport.

## 5. Projectaanpak

Dit hoofdstuk beschrijft de gekozen projectaanpak wat afstamt van het plan van aanpak (Bijlage 2: Plan van Aanpak) en het afstudeerplan. Met dit plan van aanpak is er een globaal idee ontstaan voor de aanpak van de contextuele chat. Het plan van aanpak is een los document, waardoor hier een samenvatting gegeven wordt van het document.

### 5.1 Ontwikkelmethode

Om op een gestructureerde manier te werk te gaan tijdens mijn afstudeerperiode, is het belangrijk om te werken met een passende ontwikkelmethode.

De sierteelt afdeling binnen JEM-id werkt op basis van een iteratieve ontwikkelmethode met Scrum elementen. Ze houden zich niet aan de volledige Scrum ontwikkelmethode. Wel maken ze gebruik van een Scrum bord en houden ze daily stand-ups. Zo nu en dan wordt er een retrospective gehouden met het team, maar dit vindt niet op een vast moment plaats, aangezien ze niet in Sprints werken.

Wanneer je werkt op de manier van JEM-id is er voor een bepaald team niet een doel. Dit is lastig aangezien je als team en als individu de structuur mist. Ik ben van mening dat de manier van werken van JEM-id niet aansluit bij mijn project en mij als individu. Voor mij is structuur een belangrijk onderdeel omdat ik anders snel het overzicht en het doel uit ogen verlies. Ook is dit een individueel project, waardoor ik als enige persoon met deze manier van werken werk. Om die reden is dit project niet gebonden aan de werkwijze van JEM-id. Daarom is ervoor gekozen om de Scrum methode (Agile Scrum Group, 2021) met wat meer structuur te hanteren.

Zoals aangegeven wordt de afstudeeropdracht individueel uitgevoerd. Desondanks maak ik onderdeel uit van Team Cyan. Met dit team zal ik meedoen aan de daily meet. Echter zal dit team niet deelnemen aan de sprints aangezien dit een individuele opdracht is. Wel beschik ik hierdoor over meerdere aanspreekpunten voor hulp.

Daarnaast vinden er ook de dagelijkse bespreking, sprint planningen, refinements en retrospectives plaats met mijn bedrijfsbegeleider Tim. Verder houd ik aan het einde van elke sprint een review met de opdrachtgever. Hierdoor houd ik mijn bedrijfsbegeleider en de opdrachtgever betrokken bij de voortgang van mijn afstudeeropdracht.

Ik heb ervoor gekozen om te werken in sprints van 2 weken. Hierdoor heb ik genoeg tijd voor alle werkzaamheden binnen een sprint. De werkzaamheden bestaan, naast de genoemde Scrum activiteiten, uit het technisch ontwerpen, realiseren en testen van het product.

Verder wordt er gedurende de afstudeerperiode een product backlog bijgehouden die zal bestaan uit epics en taken. De epics geven een duidelijke beschrijving van wensen van de belanghebbende. Vervolgens worden de epics opgedeeld in uitvoerbare taken.

### 5.2 Globale planning

De indeling van de planning is gebaseerd op de hoeveelheid tijd dat beschikbaar is gesteld voor het afstuderen. Het afstudeertraject duurt 17 werkweken, dus  $17 \times 5 = 85$  werkdagen.

De eerste 2 weken van mijn afstuderen besteed ik aan de voorbereiding van de eerste sprint. Dit wordt gedaan tijdens sprint 0. Hierin heb ik mijn plan van aanpak, visiedocument en mijn initiële

product backlog opgesteld. Daarnaast zijn er belangrijke keuzes afgewogen voor de realisatie van de chat. Zoals de afweging of er in een bestaande of nieuwe service ontwikkeld gaat worden. In het volgende hoofdstuk wordt hier verder op ingegaan.

Vervolgens heb ik ingeschat dat ik voor het grootste gedeelte van de afstudeerperiode bezig ben met het realiseren van het product. Er is gekozen om dit op te delen in 6 sprints van 2 weken. Gedurende deze sprints wordt er ontworpen, ontwikkeld, getest. Daarnaast worden ook de requirements bijgehouden door verschillende Scrum activiteiten uit te voeren zoals een sprint retrospective. Meer hierover in het volgende hoofdstuk.

Verder is er een buffer aan het einde van de afstudeerperiode ingepland die zich volledig leent voor het maken van het verslag of het afronden van het product.

Als laatste wil ik benadrukken dat deze planning iets afwijkt van de planning uit het afstudeerplan. Het opstellen van het plan van aanpak, de initiële backlog en het visiedocument zullen onder sprint 0 vallen.

In Figuur 2: Globale planning is de globale planning te zien.

			Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17
Nr	Taak:	Dagen																	
1	Sprint 0	10																	
	Bedrijfsbezoek																		
2	inplannen	1																	
2 tot 7	Sprint 1	10																	
	Bedrijfsbezoek																		
2 tot 7	Sprint 2	10																	
2 tot 7	Sprint 3	10																	
2 tot 7	Sprint 4	10																	
	Bespreken concept afstudeerdossier																		
2 tot 7	Sprint 5	10																	
	Tussentijds assesment																		
2 tot 7	Sprint 6	10																	
8	Documenteren afstudeerverslag	15																	
	Inleveren afstudeerverslag																		

Figuur 2: Globale planning

### 5.3 Werkzaamheden

Gedurende dit project zullen Back-End en Front-End werkzaamheden uitgevoerd worden. Voor het realiseren van Back-End functionaliteiten worden gedurende dit project een aantal stappen doorlopen.

#### 5.3.1 Technisch ontwerp

Voor de realisatie van het product zal document opgesteld worden die de ontwerpdocumentatie bevat. Dit document zal alle ontwerpdocumenten bevatten die gemaakt en gebruikt zijn bij de realisatie van het eindproduct. De ontwerpdocumentatie is te vinden onder Bijlage 4: Ontwerp documentatie.

De nieuwe functionaliteiten of uitbreidingen worden uitgetekend in een UML klassendiagram. Deze klassendiagrammen bestaan uit entiteiten en de relaties tussen de entiteiten. De entiteiten beschikken zelf over de eigenschappen zoals ze ook in de database komen te staan. Ook beschikken

de entiteiten over methodes, die weer corresponderen met de daadwerkelijke endpoints in de Back-End.

Zoals te lezen is in hoofdstuk 2 Systeemarchitectuur onder Bijlage 4: Ontwerp documentatie, bestaat Floriday uit enorm veel services met ieder hun eigen entiteiten en controllers. Daarnaast is de Back-End toevoeging gedurende dit project een losstaand onderdeel. Door deze redenen wordt er in de UML-diagrammen alleen de entiteiten opgenomen die worden toegevoegd of aangepast.

Normaliter maakt JEM-id amper gebruik van UML-diagrammen of andere ontwerptechnieken. Echter geven de UML-diagrammen tijdens dit project handvatten aan de student bij de realisatie van de Back-End en zijn ze om deze reden ook opgesteld.

### 5.3.1 Front-End

Gedurende de afstudeerperiode zal er gewerkt worden aan de twee losstaande Front-End projecten van Floriday, de customersportal en de suppliersportal. Zie hoofdstuk 2 Systeemarchitectuur onder Bijlage 4: Ontwerp documentatie.

Binnen beide zowel de suppliersportal en de customersportal wordt er gebruik gemaakt van de volgende technieken:

- **React** met react-router-dom (Facebook Inc., 2021)
- **Typescript** zorgt voor type-safe javascript (Microsoft, 2021)
- **Webpack** wordt gebruikt voor bundeling (Webpack, 2021)
  - o **webpack-dev-server** zorgt voor het testen op localhost
- De **material-ui** components vormen de basis van de gehele UI (Material-UI, 2021)
- **okta-react** beheert de log-in flow en de token storage React with react-router-dom (Npmjs, 2021)

### 5.3.2 Back-End

Bij het realiseren van Back-End functionaliteiten worden de volgende technieken gebruikt:

- **C#** (Microsoft, 2021) als programmeertaal
- **.NET Core** (Microsoft, 2021) als framework voor maken van de service
- **Entity Framework Core** (Microsoft, 2021) als Data access layer

JEM-id kiest ervoor om alle Back-End services te schrijven in de C# programmeertaal.

Verder wordt het .NET Core Framework gebruikt bij ontwikkelen van de services en hun API's. Daarnaast wordt het Entity framework Core (EF Core) gebruikt als Data Access Layer. EF Core helpt bij het opstellen van entiteiten die een tabel in de database representeren. Daarnaast gebruikt JEM-id EF Core voor Code First database migraties. (Entity Framework Tutorial, 2021) Op deze manier is het werken met data binnen de Back-End service eenvoudiger gemaakt.

Daarnaast heeft JEM-id eigen Code Conventies opgesteld, Bijlage 7: JEM-id Code conventies, waar iedere ontwikkelaar zich aan moet houden bij het ontwikkelen van Back-End services. Door een generieke werkwijze en Code conventies zorgt JEM-id ervoor dat alle Back-End services van consistente kwaliteit zijn.

Bij het ontwikkelen aan Back-End services binnen Floriday, wordt er tijdens dit project zoveel mogelijk gehouden aan de bovengenoemde werkwijze en Code conventies. Dit zorgt ervoor dat de

consistentie van dit project aansluit op de rest van het platform en zorgt voor een betere overdraagbaarheid na de afstudeerperiode.

### 5.3.3 Testen

Om de werking van de geïmplementeerde endpoints in de Back-End te valideren worden er Unit tests geschreven. De Unit tests zorgen ervoor dat de werking van elke endpoint wordt getest voordat een wijziging live gezet kan worden.

Verder zal er een integratietest geschreven worden om de integratie van het gemaakte product binnen het platform te valideren.



## 6. Sprint 0: Onderzoeksfase

In sprint 0 worden alle voorbereidingen getroffen om hierna te starten met de eerste sprint van de afstudeerperiode. In deze sprint zal de opdrachtgever geïnterviewd worden om daarmee het visiedocument op te kunnen stellen. Aan de hand van het visiedocument wordt vervolgens de initiële Product backlog opgesteld.

Verder worden er in deze sprint belangrijke beslissingen afgewogen die voor het gehele project van belang zullen zijn. Zoals de afweging tussen het ontwikkelen binnen een nieuwe of bestaande service. En worden de mogelijkheden rondom een real-time chat in kaart gebracht.

### 6.1 Interview

Om een visiedocument op te stellen moet de belanghebbende, de behoeftes, het product perspectief en de positionering van het afstudeerproject bepaald en vastgelegd worden. Om dit te kunnen doen is ervoor gekozen om de opdrachtgever te interviewen. De meest ideale situatie zou zijn dat alle belanghebbende geïnterviewd konden worden, maar vanwege de pandemie en vakantie periode waarin de afstudeerperiode plaats vond, was het lastig om kopers en kwekers te spreken.

Er is er daarom voor gekozen om de opdrachtgever te interviewen. Hij is binnen het bedrijf een van de Product Owners en de lead UX designer van Floriday. Hij heeft het meeste contact met alle belanghebbende en gebruikers van het gehele Floriday platform. Door de opdrachtgever met de juiste vragen te interviewen kan alsnog alle benodigde informatie achterhaald worden om het visiedocument op te stellen.

De vragen van het interview zijn opgesteld aan de hand van: Wat, Waarom, Wie, Wanneer en Hoe. In dit interview zijn er zowel open als gesloten vragen gesteld. Hierdoor is er achterhaald wat de reden is voor een contextuele chat, wat voor problemen de chat gaat oplossen, wie het gaat gebruiken, hoe het gebruikt gaat worden en welke functionaliteiten de contextuele chat moet bevatten om te gebruiken.

Het interview is opgenomen. Waarvan een samenvatting is gemaakt die opgestuurd is naar de opdrachtgever ter verificatie. Aan de hand hiervan is het visiedocument opgesteld. Hier onder is de samenvatting van het interview te vinden:

#### Belangrijkste bevindingen

De reden om een chat functie in Floriday te bouwen is omdat er al 1200 kwekers zijn die een eigen winkel omgeving binnen het platform hebben. Binnen deze winkel omgeving kan de kweker zijn aanbod ter beschikking stellen aan de koper. Op dit moment hebben deze kwekers nog geen mogelijkheid om contact te hebben met de kopers via Floriday. De kopers hebben ook geen mogelijkheid om vragen te stellen aan de kweker over het aanbod.

Om het gevoel en idee van een fysieke winkel na te bootsen is ervoor gekozen om een chatfunctie in de winkel omgevingen van de kwekers te implementeren. Op deze manier kunnen zij digitaal contact hebben met de kopers en kunnen de kopers vragen stellen.

Daarnaast heeft Floriday het doel om 100% digitaal te worden, dit houdt in dat de volledige tot stand komen van een bestelling digitaal afgehandeld kan worden. Hier hoort ook de communicatie

omtrent het aanbod bij. De contextuele chat heeft een toegevoegde waarde om het 100% digitaal doel te behalen.

De contextuele chat zal enkel gebruikt worden door de kopers en de kwekers. Waar rekening wordt gehouden met de tijd en ruimte van de kweker. Wanneer zij geen ruimte hebben voor een chat zal deze niet in hun winkelomgeving aangeboden worden.

De contextuele chat lost veel problemen op in de dagelijkse werkzaamheden van de kopers en kwekers. Voor de koper is de drempelwaarde om een vraag te stellen lager, ze kunnen meteen een bericht sturen wat sneller is dan bellen, whatsappen of mailen. Hierbij hoeven ze dus niet te switchen van communicatiemiddel. Naast de connectie tussen de koper en kweker via Floriday, hoeft de koper geen contactgegevens van de kweker te hebben. Op deze manier kunnen er gelijk handelingen plaatsvinden en kunnen ze gelijk zaken met elkaar doen.

Voor de kweker lost de contextuele chat ook veel problemen op. Zo komen informatievragen gestructureerd binnen, is de chat toegankelijker, sneller, tijdbesparend en zijn de werkzaamheden binnen de kwekers organisatie overdraagbaar. De chat haalt de drempelwaarde bij de koper weg om vragen te stellen, zo krijgt de kweker ook sneller vragen binnen vanuit de kopers. Ook hoeft de kweker niet te switchen tussen telefoon, whatsapp en e-mail. Er kunnen verschillende vragen binnen komen en de kweker kan de kopers tegelijk bedienen.

De overige uitkomsten van het interview zijn verwerkt in een visiedocument. Hierin zijn de belanghebbenden beschreven met de bijpassende behoeftes.

## 6.2 Visiedocument

Aan de hand van het interview is er een visiedocument opgesteld, te vinden onder Bijlage 3: Visiedocument. Zoals te lezen is in hoofdstuk 5.1 Ontwikkelmethode wordt er voor dit project gebruik gemaakt van Scrum en is om die reden ook het visiedocument opgesteld. Met het visiedocument wordt de richting, visie, belanghebbenden en behoeftes van het project vastgelegd. Het document helpt om gedurende het project juiste keuzes te maken met de beoogde visie. Nu zullen de belangrijkste punten uit het document samengevat worden.

### Huidige bedrijfsproces

Momenteel is er geen manier voor kopers en kweker om te communiceren met elkaar via het platform. Nu communiceren de kopers en kweker voornamelijk via externe communicatiemiddelen zoals email, whatsapp en/of telefonie. Echter kan dit alleen wanneer beide partijen elkaar al kennen en deze informatie van elkaar hebben. Hierdoor is het lastig om snel een vraag te kunnen stellen bij bijvoorbeeld een nieuwe partner.

### Probleemstelling

<b>Het probleem:</b>	Kopers en kwekers communiceren momenteel nog alleen via externe communicatiemiddelen, zoals e-mail, telefonie of WhatsApp. Doordat het platform steeds meer groeit kennen steeds meer kwekers en kopers elkaar persoonlijk niet meer. Hierdoor kost het tijd om te wisselen tussen verschillende externe communicatiemiddelen en het platform. Daarnaast kunnen ze elkaar lastig bereiken via externe communicatiemiddelen
<b>Dit beïnvloedt:</b>	Kopers en kwekers van bloemen en planten binnen het Floriday platform.
<b>De impact hiervan is:</b>	Doordat er binnen het platform geen manier mogelijk is om vragen te kunnen stellen aan een kweker, worden externe communicatiemiddelen gebruikt. Hierdoor zijn kopers en kwekers meer tijd kwijt aan het stellen en beantwoorden van vragen en het wisselen tussen communicatiemiddelen en het platform. Daarnaast is de drempelwaarde om een vraag te stellen ook hoog, doordat niet alle kwekers en kopers elkaars gegevens hebben om de communicatie te starten.
<b>Een succesvolle oplossing:</b>	<p>Verlaagd de drempelwaarde om vragen te stellen en antwoorden te geven.</p> <p>Bespaard tijd door vragen te stellen en antwoorden geven over contextuele onderwerpen.</p> <p>Bespaard tijd door niet te hoeven wisselen van communicatiemiddel of platform bij het stellen of beantwoorden van vragen rondom het aanbod.</p>

### Alternatieve oplossingen

<b>Alternatieve oplossing:</b>	<b>Voordelen:</b>	<b>Nadelen:</b>
<b>WhatsApp</b>	<ul style="list-style-type: none"> <li>• Veel gebruikt communicatiemiddel</li> <li>• Overal te gebruiken</li> <li>• Te gebruiken in groepen</li> </ul>	<ul style="list-style-type: none"> <li>• Wordt ook voor persoonlijke communicatie gebruikt</li> <li>• Buiten werktijd lastiggevalen worden</li> <li>• Is een extern communicatiemiddel</li> <li>• Communicatie niet vastgelegd binnen het platform</li> </ul>
<b>Email</b>	<ul style="list-style-type: none"> <li>• Kan alleen zakelijk gebruikt worden</li> <li>• Reactie moment zelf bepalen</li> </ul>	<ul style="list-style-type: none"> <li>• Tijd tussen berichten</li> <li>• Geen directe reactie</li> </ul>
<b>Telefonie</b>	<ul style="list-style-type: none"> <li>• Direct iemand aan de lijn</li> </ul>	<ul style="list-style-type: none"> <li>• Huidige activiteiten worden onderbroken</li> <li>• Kan niet meerdere gebruikers tegelijk helpen</li> <li>• Is een extern communicatiemiddel</li> </ul>

		<ul style="list-style-type: none"> <li>Kost meer tijd</li> </ul>
<b>Vrije chat</b>	<ul style="list-style-type: none"> <li>Alle soorten vragen mogelijk</li> <li>Alle soorten antwoorden mogelijk</li> </ul>	<ul style="list-style-type: none"> <li>Alle soorten vragen mogelijk</li> <li>Alle soorten antwoorden mogelijk</li> </ul>

### Belanghebbende

Belanghebbendenrol	Vertegenwoordiger	Betrokkenheid
Kwekers	Product Owner Floriday: Bas La-Graauw	Deze belanghebbende maken gebruik van het Floriday platform voor kwekers om hun aanbod op aan te maken en bij te houden. Kwekers zullen de contextuele chat gebruiken om vragen van kopers te beantwoorden.
Kopers	Product Owner Floriday: Bas La-Graauw	Deze belanghebbende maken gebruik van het Floriday platform voor kopers om aanbod te bestellen bij verschillende kwekers. Kopers zullen de contextuele chat gebruiken om vragen te stellen over het aanbod van kwekers.
Opdrachtgever	Product Owner Floriday: Bas La-Graauw	Deze belanghebbende is de opdrachtgever van de afstudeeropdracht. Verder levert hij het visuele ontwerp.
Ontwikkelaars	Ontwikkel Teams: Blue, Purple, Cyan	De ontwikkelaars werken ook aan het digitale Floriday landschap. Hierdoor zullen aanpassingen of toevoegingen afgestemd moeten worden.

### Productperspectief

Om af te sluiten is het productperspectief dat het beoogde eindproduct een toevoeging zal zijn binnen het digitale Floriday landschap. De kopers kunnen met het beoogde eindproduct communiceren met de kweker wanneer zij direct inkopen doen bij een kweker. Kopers kunnen direct inkopen doen bij een kwekersorganisatie. Hier kan de koper al het aanbod zien van de kweker en artikelen selecteren en bestellen.

De kwekers kunnen met beoogde eindproduct kopers helpen wanneer zij vragen hebben bij het direct inkopen doen.

Om samen te vatten zal het beoogde product:

- De snelheid verhogen waarmee vragen gesteld en antwoorden gegeven kunnen worden
- De drempelwaarde verlagen voor kopers en kweker om contact op te nemen bij het direct inkopen doen
- De taalbarrière weghalen door vertaling van vooraf gedefinieerde vragen en antwoorden
- Meer context aan een bericht gegeven kunnen worden (zoals informatie over een artikel)
- Geïntegreerd worden in het digitale landschap van Floriday
- Visueel omgezet worden op basis van het aangeleverde ontwerp

### 6.3 Gebruik van Notion

Om het afstudeerproces gestructureerd te laten verlopen, is ervoor gekozen om gebruik te maken van de webapplicatie Notion (Notion Labs, 2021). Notion is een applicatie die componenten biedt om eigen systemen te kunnen creëren. Gedurende de afstudeerperiode is Notion gebruikt voor alle Scrum activiteiten, zoals het opstellen en bijhouden van de Product backlog, vaststellen van het sprint goal, het vastleggen van de punten uit de retrospectives en sprint reviews. Notion is daarnaast ook gebruikt voor het vastleggen en bijhouden van beslissingen, keuzes en meetings. Notion heeft geholpen bij een gestructureerd afstudeerproces.

In het onderstaande Figuur 3: Voorbeeld Notion pagina is een voorbeeld te zien van een opgestelde sprint pagina binnen Notion. Zoals te zien is bevat elke Sprint backlog Features met User Stories waaronder taken zijn opgesteld.

**#5 Sprint board (17-07 / 30-07)**

**Info**

Sprint Goal

**Sprint Board**

Product backlog Table view

Estimati...	Name	Sprint	Status	Prioriteit	Platform	Tags	Epic
13	Antwoord: Het is geregeld!	Sprint 5	Done	1. Must have	FE-Suppliersportal	Front-End	Vooraf gedefinieerde antwoorden voor kweker
5	Component aanmaken voor de het antwoord	Sprint 5	Review	1. Must have	FE-Customersportal	Front-End	Antwoord: Het is geregeld!
5	Antwoord versturen bij het selecteren.	Sprint 5	Done	1. Must have	FE-Customersportal	Front-End	Antwoord: Het is geregeld!
13	Antwoord: Moment, we kijken ernaar	Sprint 5	Done	1. Must have	FE-Suppliersportal	Front-End	Vooraf gedefinieerde antwoorden voor kweker
5	Component aanmaken voor de het antwoord	Sprint 5	Done	1. Must have	FE-Customersportal	Front-End	Antwoord: Moment, we kijken ernaar
5	Antwoord versturen bij het selecteren.	Sprint 5	Done	1. Must have	FE-Customersportal	Front-End	Antwoord: Moment, we kijken ernaar
13	Antwoord: Helaas is er op dit moment geen aanbod	Sprint 5	Done	1. Must have	FE-Suppliersportal	Front-End	Vooraf gedefinieerde antwoorden voor kweker
5	Component aanmaken voor de het antwoord	Sprint 5	Done	1. Must have	FE-Customersportal	Front-End	Antwoord: Helaas is er op dit moment geen aan
5	Antwoord versturen bij het selecteren.	Sprint 5	Done	1. Must have	FE-Customersportal	Front-End	Antwoord: Helaas is er op dit moment geen aan
5	Vertaling van chat onderzoeken	Sprint 5	Done	3. Could have		Onderzoek	Vertalen van vragen
1	Gebruikers als tester werven	Sprint 5	Done			Onderzoek	Vertalen van antwoord

+ New

COUNT 11

**#2 Retrospective**

Goed 1 Kon beter 1 Mee te nemen 2

Epics opgedeeld in kleine opknapbare userstories(features) en taken.

Nog steeds minder switchen en meer gefocust blijven bij sprint.

Laatste sprint weer epics opdelen in opknapbare userstories.

Gefocust blijven op het doel

+ New

No Status 0

+ Add a group

+ New page

Figuur 3: Voorbeeld Notion pagina














## 6.4 Product backlog

Zoals hiervoor genoemd is, is ook de Product backlog en alle Sprint backlogs opgesteld en bijgehouden in Notion (Notion Labs, 2021).

De definitieve Product backlog is te vinden onder Bijlage 1: Product backlog.

Binnen de Product backlog wordt er gebruik gemaakt van epics en taken. Hierboven zijn de epics te zien die allemaal een korte beschrijving bevatten van de behoefte in vorm van een userstory. De behoefte is afkomstig uit het visiedocument (Bijlage 3: Visiedocument) waar de behoefte van de belanghebbende beschreven staat. De user stories zijn allemaal op dezelfde manier opgesteld, zoals bijvoorbeeld:

*Als kweker, wil ik antwoord kunnen geven op vragen van kopers, zodat ik de koper kan helpen bij het plaatsen van een bestelling.*

Prioriteit	Name	User Story
1. Must have	 <u>Vraag stellen als koper</u>	Als koper, wil ik een vraag in vrije tekst kunnen stellen aan een kweker op zijn winkel omgeving, zodat ik antwoord in vrije tekst kan krijgen op mijn vraag.
1. Must have	 <u>Antwoord geven als kweker</u>	Als kweker, wil ik antwoord kunnen geven op vragen van kopers, zodat ik de koper kan helpen bij het plaatsen van een bestelling.
1. Must have	 <u>Vooraf gedefinieerde vraag voor koper</u>	Als koper, wil ik een vooraf gedefinieerde vraag kunnen kiezen, zodat ik sneller een vraag kan stellen door de vraag niet zelf op te hoeven stellen.
1. Must have	 <u>Vooraf gedefinieerde antwoorden voor kweker</u>	Als kweker, wil ik kunnen kiezen uit vooraf gedefinieerde antwoorden, zodat ik snel antwoord kan geven zonder zelf het antwoord op te hoeven stellen.
1. Must have	 <u>Chat overzicht voor kweker</u>	Als koper, wil ik een overzicht hebben van alle chats, zodat ik in een oogopslag kan zien welke chats er open staan.
1. Must have	 <u>Integratie in het platform</u>	Als ontwikkelaar van het Floriday platform, wil ik dat de contextuele chat geïntegreerd wordt binnen het platform zonder dat het impact heeft op huidige functionaliteiten, zodat de rest van het platform operationeel blijft.
2. Should have	 <u>Instelling om chat te gebruiken</u>	Als kweker, wil ik in kunnen stellen of ik de chat functionaliteit wil gebruiken, zodat ik kan bepalen of ik wel of niet kopers te woord wil/kan staan.
3. Could have	 <u>Beschikbaarheid inzien</u>	Als kweker en als koper, wil ik zien of iemand online is, zodat ik weet of ik een snelle reactie kan verwachten.
2. Should have	 <u>100% digitaal</u>	Als opdrachtgever, wil ik alle communicatie digitaal vastleggen over de totstandkoming van een bestelling, zodat het het 100% digitaal doel gehaald kan worden.
2. Should have	 <u>Vertalen van vragen</u>	Als koper, wil ik dat de vooraf gedefinieerde vragen vertaald worden, zodat de koper mijn vraag kan begrijpen.
2. Should have	 <u>Vertalen van antwoorden</u>	Als kweker, wil ik dat de vooraf gedefinieerde antwoorden vertaald worden, zodat de koper mijn antwoorden kan begrijpen.
1. Must have	 <u>Afstudeerverslag</u>	Als student, wil ik mijn afstudeerverslag schrijven, zodat ik mijn afstudeerperiode kan beschrijven door middel van een verslag.
2. Should have	 <u>Live chat</u>	Als koper en als kweker, wil ik live kunnen chatten, zodat snel en gemakkelijk berichten kan communiceren zonder handmatig de nieuwste berichten op te halen.

Figuur 4: Product backlog Epics

In het bovenstaande figuur (Figuur 4: Product backlog Epics) zijn de Epics te zien die opgesteld zijn binnen Notion op basis van het Bijlage 3: Visiedocument, dat onder andere bestaat uit de behoefte van de belanghebbende.

Voor de prioritering is er gekozen voor de MoSCoW methode (projectmanagementsite, 2021). In tegenstelling tot de prioritering methode hoog-midden-laag, heeft MoSCoW de mogelijkheid om requirements af te bakenen van het project (won't have now). Deze afbakening helpt om rekening te houden met de toekomst en het hebben van een compacte scope. Daarnaast wordt er met MoSCoW ook bepaald welke requirements essentieel zijn voor het project (must have), welke ook belangrijk zijn (should have) en welke requirements nog opgepakt kunnen worden wanneer daar tijd voor is (could have).

Vervolgens worden de epics opgedeeld in kleine uitvoerbare taken die opgepakt kunnen worden tijdens een sprint. Deze taken bevatten geen user story aangezien die al beschreven staat bij de overkoepelende epics, maar bevatten wel vaak een beschrijving. Hieronder, in Figuur 5: Voorbeeld Sprint backlog, is een voorbeeld van een Sprint backlog te zien.

Estimati...	Name	Epic	Sprint	Status	Prioriteit	Tags	Platform
16	Integratie test in API-tester	Integratie in het platform	Sprint 4	Done	2. Should have		
8	Test flow van supplier	Integratie test in API-tester	Sprint 4	Done	2. Should have	Tests	API-tester
8	Test flow van customer	Integratie test in API-tester	Sprint 4	Done	2. Should have	Tests	API-tester
17	Berichten automatisch ophalen en weergeven in chat	Live chat	Sprint 4	Done	1. Must have		BE-Match & Connect
5	Automatisch ophalen van berichten onderzoeken.	Berichten automatisch ophalen en weergeven in	Sprint 4	Done	1. Must have	Onderzoek	
2	UML diagram uitbreiden met ophalen van laatste berichten	Lezen van berichten vastleggen	Sprint 4	Done	1. Must have	Documentatie	
5	Endpoint maken die laatste berichten ophaalt sinds een bepaald tijdstip	Berichten automatisch ophalen en weergeven in	Sprint 4	Done	1. Must have	Back-End	BE-Match & Connect
5	Endpoint maken die laatste berichten ophaalt sinds een bepaald tijdstip	Berichten automatisch ophalen en weergeven in	Sprint 4	Done	1. Must have	Back-End	BE-Match & Connect
12	Lezen van berichten vastleggen	Live chat	Sprint 4	Done	3. Could have		BE-Match & Connect
5	Bepalen wanneer een bericht nieuw is of gelezen is onderzoek	Lezen van berichten vastleggen	Sprint 4	Done	3. Could have	Onderzoek	
2	UML diagram uitbreiden met lezen van berichten	Lezen van berichten vastleggen	Sprint 4	Done	3. Could have	Documentatie	
5	Back-End uitbreiden met lezen van berichten	Lezen van berichten vastleggen	Sprint 4	Done	3. Could have	Back-End	BE-Match & Connect

Figuur 5: Voorbeeld Sprint backlog

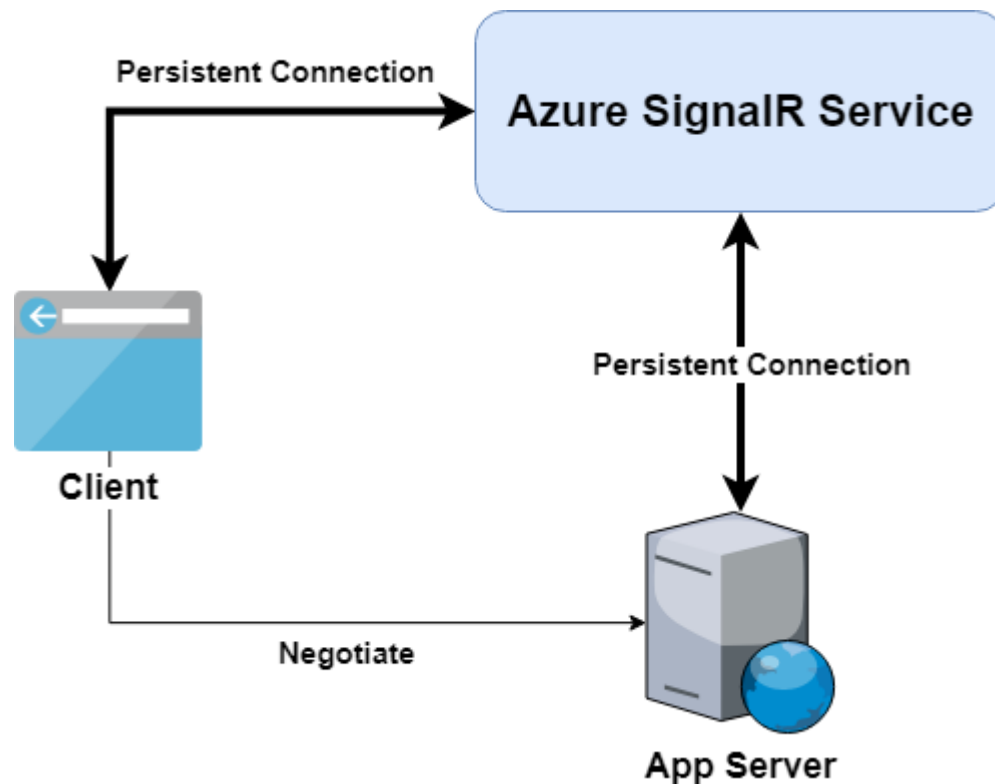
Aangezien het project een individuele opdracht is en de taken individueel worden opgepakt, is er gekozen om een gemakkelijke en voor mij als individu, werkbare naamgeving en beschrijving van de taken op te stellen. De epics daarentegen hebben wel een goede beschrijving in de vorm van een user story om de behoefte van de belanghebbende te waarborgen.

## 6.5 Real-time chat mogelijkheden

De meest veelvoorkomende chats zijn real-time chats. Dit houdt in dat berichten real-time verstuurd en opgehaald kunnen worden. Het belangrijkste van een real-time chat is dat het snel, betrouwbaar en simpel moet zijn voor de gebruikers. Gebruikers zouden niet een actie moeten uitvoeren om nieuwe berichten op te halen.

Om dit te implementeren zijn er verschillende mogelijkheden zoals bijvoorbeeld SignalR (Microsoft, 2021) of FireBase (Google, 2021). SignalR en FireBase zijn event based technieken, waar SignalR onderdeel is van het ASP.NET framework (Microsoft, 2021), waar de microservices binnen het Floriday landschap ook in zijn opgebouwd. FireBase is een losstaand framework die dan in zijn geheel geïmplementeerd zou moeten worden. In dit geval is SignalR de betere keus voor een real-

time chat omdat dit al in het ASP.NET framework zit. Figuur 6: Werking SignalR laat de werking van SignalR zien.



Figuur 6: Werking SignalR

Na het onderzoeken van beide technieken is er geïnformeerd bij het team die gaat over de architectuur, security en implementatie van nieuwe technieken binnen Floriday, genaamd team Black. Zij gaven aan dat er al eerder een poging is gedaan om event based technieken in een andere service te implementeren, dit zonder succes vanwege security issues. Daarnaast gaven ze aan dat ze eerst deze issues willen verhelpen voordat ze SignalR of FireBase in een service kunnen implementeren.

Met dit in gedachte is ervoor gekozen om tijdens de eerste sprint te starten met endpoints aan te maken, in een service binnen het digitale Floriday landschap. Hiermee wordt ervoor gezorgd dat de chat niet afhankelijk is van event based technieken voor het verwerken van berichten. Bij het aanroepen van deze endpoints kunnen berichten verstuurd, opgeslagen en opgehaald worden. Hierdoor zijn de hoofdfunctionaliteiten van de chat alsnog mogelijk. Wanneer er in de toekomst alsnog een event based techniek geïmplementeerd kan worden, kunnen deze events het live updaten van berichten mogelijk maken door deze endpoints aan te roepen.

## 6.6 Afweging tussen nieuwe of bestaande service

Floriday bestaat uit verschillende services die allemaal hun eigen functionaliteiten hebben. Deze structuur is te vergelijken met de microservices architectuur (Microsoft, 2019). Iedere microservice heeft zo zijn functionaliteiten en verantwoordelijkheden. Elke service kan daardoor los van elkaar functioneren en zijn niet afhankelijk van elkaar. Daarnaast zijn er 2 Front-End projecten, 1 voor de koper en 1 voor de kweker, die beide communiceren met deze services.



De chat toevoeging moest ergens binnen dit landschap geplaatst worden. De overwegingen waren een nieuwe microservice of een toevoeging op de bestaande microservice te ontwikkelen waar de Front-End projecten gebruik van kunnen maken.

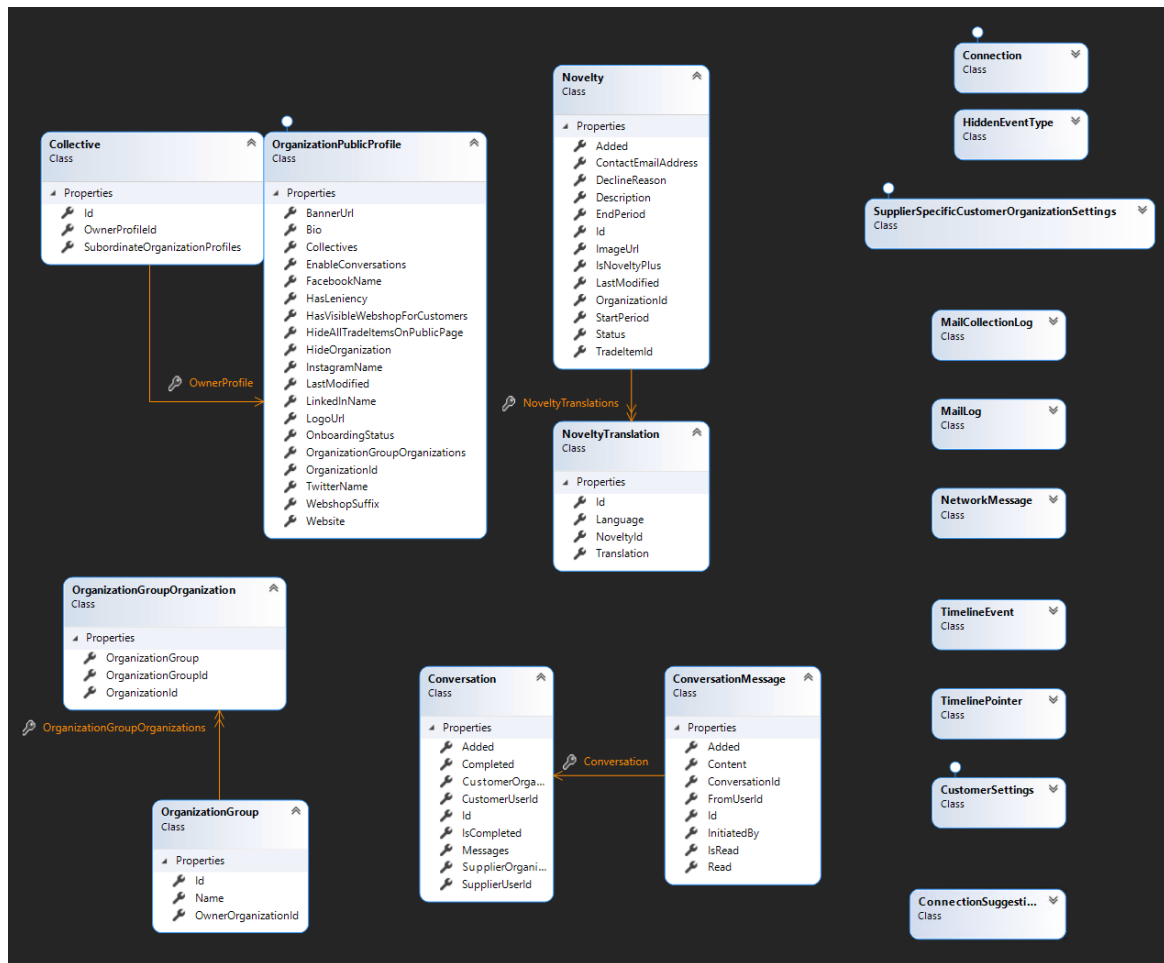
<b>Nieuwe microservice</b>	<b>Voordelen:</b> <ul style="list-style-type: none"> <li>- Losstaande en onafhankelijke service</li> </ul> <b>Nadelen:</b> <ul style="list-style-type: none"> <li>- Tijdrovend</li> <li>- Afhankelijkheid bij andere drukbezette teams</li> <li>- Minder tijd daadwerkelijk product</li> </ul>
<b>Bestaande microservice</b>	<b>Voordelen:</b> <ul style="list-style-type: none"> <li>- Weinig tot geen integratie tijd</li> <li>- Meer tijd daadwerkelijk product</li> <li>- Geen afhankelijkheid bij andere teams</li> </ul> <b>Nadelen:</b> <ul style="list-style-type: none"> <li>- Niet volledig losstaande en onafhankelijke service</li> </ul>

De uiteindelijke keuze, voor de toevoeging van de chat omgeving, is gevallen op een bestaande microservice. De logica van de chat wordt aan de bestaande M&C service ontwikkeld. De voordelen van een bestaande microservice voor dit project uitwegen de voor- en nadelen van de keuze voor een nieuwe service.

De M&C (Match & Connect) is een vrij nieuwe, al bestaande, service. De service wordt, zoals te zien in de onderstaande figuur (Figuur 7: Match & Connect datastructuur), gebruikt voor de connectie en de functionaliteiten daaromheen, tussen kopers en kwekers.

De M&C service is dan ook de perfecte plek om de contextuele chat in op te bouwen aangezien het voor een koper enkel mogelijk is om direct inkopen te doen bij een kweker waar zij al een connectie mee hebben. Wanneer de chat in de M&C service opgebouwd wordt kunnen kopers ook enkel chatten met kwekers waarmee zij al een connectie hebben.

Figuur 7: Match & Connect datastructuur geeft de huidige datastructuur aan van de Match & Connect service. Houdt hierbij rekening dat deze diagram gegenereerd is nadat het eindproduct is opgeleverd, vandaar dat ook de Conversatie en ConversationMessage classes erin staan.



Figuur 7: Match & Connect datastructuur

Ook is de M&C service al opgebouwd met een API die ervoor zorgt dat de endpoints door de UI benaderd kunnen worden. Tot slot is de M&C service al volledig geïntegreerd in het Floriday landschap. Door geen eigen service op te bouwen voor dit project is er onnodige tijd bespaard gebleven voor het project en de rest van het team.

## 7. Sprint 1: Koper chat prototype

Door sprint 1 zal er een begin worden gemaakt aan het eindproduct, de contextuele chat. Aan het begin van de sprint is er een doel opgesteld. Voorafgaande aan de sprint is er een Sprint backlog gemaakt, op basis van het de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij sprint ingegaan worden op de realisatie werkzaamheden rondom het chat prototype voor de koper. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 7.1 Sprint doel

*Sprint 1 heeft als doel een prototype van de contextuele chat te ontwikkelen voor de kopers. Met het prototype moet het mogelijk zijn voor kopers om een bericht naar een kweker te versturen via een chat venster. Dit venster moet geïntegreerd worden binnen het bestaande scherm waarin kopers direct inkopen kunnen doen bij een kweker.*

De student heeft vanuit de visie van het project en de belangrijkste requirements het sprint doel voorgesteld aan de opdrachtgever. Vanuit dit overleg is er bepaald dat het versturen en ontvangen van berichten het meest belangrijke element voor een chat is. Ook wordt het sprint doel en de punten in de sprint bepaald door de punten uit de review en retrospective van de vorige sprint.

De keuze is dan ook geweest voor deze eerste sprint, om aan de slag te gaan om het versturen en ophalen van berichten voor de koper mogelijk te maken. Het visuele ontwerp voor de customersportal (Bijlage 4: Ontwerp documentatie, hoofdstuk 5.1 Visueel ontwerp customersportal) stond al klaar en hier kon meteen mee gewerkt worden. Het visuele ontwerp van de kwekers portal daarentegen nog niet.

Desondanks ervoor gekozen is om voor deze sprint enkel te focussen op de Front-End voor de koper, wordt er wel een goede basis neergezet voor de Back-End voor zowel de koper als de kweker. Het versturen van berichten werkt twee kanten op en betreft beide partijen.

### 7.2 User stories

Om het sprintdoel voor de eerste sprint te voltooien zijn er Epics opgesteld met ieder een user story. De onderstaande epics zijn afkomstig van de behoefte van de belanghebbende, afkomstig uit het visiedocument (Bijlage 3: Visiedocument). Deze epics zijn grote globale onderwerpen die tijdens dit project voltooid moeten worden, deze zitten verspreid over de sprints. Het is dan niet zo dat een Epic voltooid wordt tijdens deze sprint. Onder deze Epics zijn taken aangemaakt waarmee het sprintdoel gehaald kan worden. De taken zijn hier niet vermeld aangezien de Epics meer zeggen over welke onderwerpen er opgepakt worden.

Titel	User story	Prioriteit
Vraag stellen als koper	Als koper, wil ik een vraag in vrije tekst kunnen stellen aan een kweker op zijn winkel omgeving, zodat ik antwoord in vrije tekst kan krijgen op mijn vraag.	Must have
Antwoord geven als kweker	Als kweker, wil ik antwoord kunnen geven op vragen van kopers, zodat ik de koper kan helpen bij het plaatsen van een bestelling.	Must have

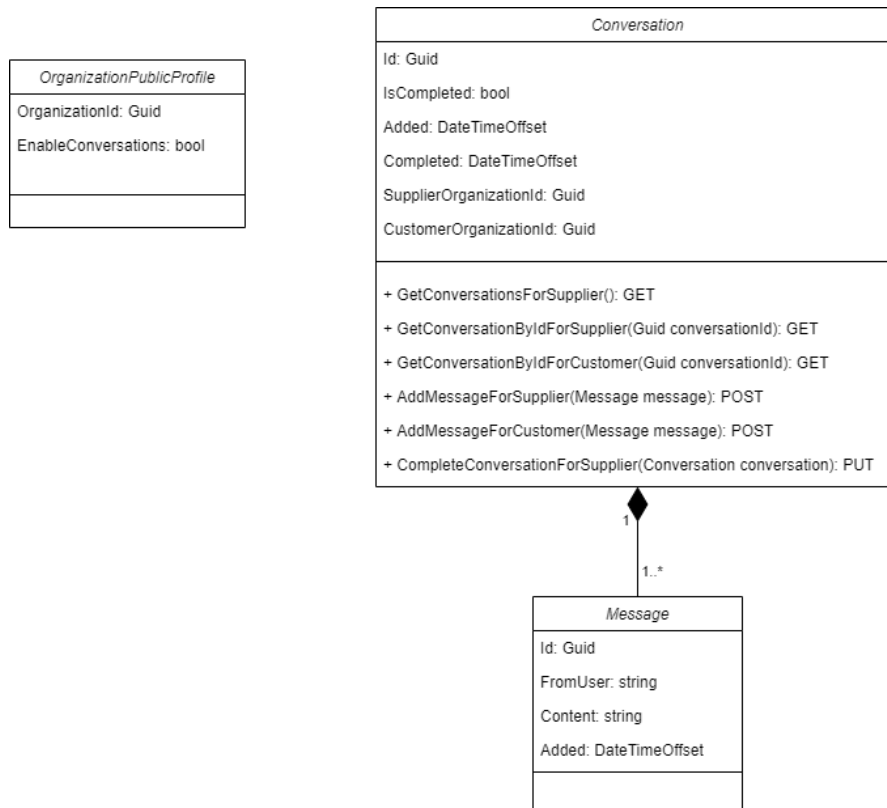
100% digitaal	Als opdrachtgever, wil ik alle communicatie digitaal vastleggen over de totstandkoming van een bestelling, zodat het 100% digitaal doel van Floriday gehaald kan worden.	Should have
Instelling om chat te gebruiken	Als kweker, wil ik in kunnen stellen of ik de chat functionaliteit wil gebruiken, zodat ik kan bepalen of ik wel of niet kopers te woord wil/kan staan.	Should have

Zoals te lezen in het sprint doel stond deze sprint in het teken van het ontwikkelen van een prototype van de contextuele chat. De bovenstaande epics zijn opgesplitst voor de koper en de kweker. De koper moet een vraag kunnen stellen en de kweker moet hier antwoord op kunnen geven. Deze twee punten kunnen wel op dezelfde manier aangevlogen worden. Om dit te kunnen verzorgen moeten er überhaupt berichten verstuurd kunnen worden.

### Ontwerp

Met dit bovenstaande in gedachte is er een begin gemaakt aan de ontwerp documentatie, door een UML-diagram te maken van de Back-End toevoeging op de M&C service. Dit diagram helpt bij het ontwikkelproces waarin de entiteiten en endpoints gerealiseerd worden.

Zoals te zien is in het visuele ontwerp, onder Bijlage 4: Ontwerp documentatie hoofdstuk 5.1 Visueel ontwerp customersportal, is het uiteindelijk de bedoeling dat een koper een conversatie kan starten wanneer hij op het *Direct inkopen* scherm zit van een specifieke kweker. Dit doet hij door op de knop stel een vraag te drukken, wat vervolgens een chat venster opent. De conversatie wordt enkel gestart wanneer de koper een vraag heeft verstuurd. De berichten worden in de database opgeslagen en binnen de conversatie van de kweker weer getoond.



Figuur 8: Sprint 1 Back-End UML-diagram

Figuur 8: Sprint 1 Back-End UML-diagram is te zien dat de entiteiten Conversation en Message gehanteerd worden. Het is namelijk belangrijk voor zowel de koper als de kweker dat ze berichten naar elkaar kunnen versturen, maar deze berichten wel onder een conversatie hangen. Uiteraard kunnen kopers wel meerdere conversaties hebben lopen met verschillende kwekers en vice versa.

Naast de entiteiten Conversation en Message is ook de bestaande class, genaamd OrganisationPublicProfile, opgenomen. Bij deze class is er een eigenschap toegevoegd, genaamd EnableConversations. Deze eigenschap gaat ervoor zorgen dat de functionaliteiten rondom de contextuele chat alleen gebruikt worden door kwekers die dit willen gebruiken.

### Back-End realisatie

Na het maken van het UML-diagram is dit gerealiseerd in de M&C service. Dit diagram en de realisatie hiervan binnen de M&C microservice is opgesteld volgens de Code Conventies van JEM-id, te vinden onder Bijlage 7: JEM-id Code conventies. Ondanks er binnen Floriday voor gekozen is om de endpoints voor de koper en kweker te splitsen is er bij het realiseren van de endpoints voor gezorgd dat er generieke methodes worden gebruikt om dubbele code te voorkomen. De endpoints van de koper en kweker maken hierdoor beide gebruik van dezelfde logica binnen de generieke methode.

Binnen de Back-End bestond er al een endpoint die ervoor zorgde dat de eigenschappen binnen de OrganisationPublicProfile gewijzigd konden worden. Er is ervoor gezorgd dat de EnableConversations eigenschap hier ook in meegenomen kon worden. Bij de realisatie van de endpoints is er rekening mee gehouden dat wanneer iemand de EnableConversations eigenschap, niet ingesteld heeft, hij deze functionaliteiten niet kan gebruiken.

Bij de realisatie is er ook voor gekozen om Code First (Entity Framework Tutorial, 2021) te werken. Dit houdt in dat er eerst classes zijn aangemaakt binnen de service. Deze classes staan gelijk aan de modellen die er in de database moeten komen te staan. Dit zorgt ervoor dat de gehele database structuur automatisch gegenereerd wordt, door middel van migraties en gelijk staat aan de entiteiten binnen in je service. Fijn hieraan is dat dezelfde database structuur altijd opnieuw te genereren is, ongeacht wat er met de database gebeurt.

Om de functionaliteit van de endpoints te valideren zijn er Unit tests opgesteld voor de endpoints. Daarnaast wordt de code altijd eerst gereviewd voordat het live mag.

### Front-End realisatie

Tijdens deze sprint was enkel het visuele ontwerp van de customersportal beschikbaar. Dit ontwerp is in de Front-End gerealiseerd met de bijbehorende interacties.

De realisatie van het visuele ontwerp voor het antwoord geven als kweker, zal in een volgende sprint opgepakt worden. Dit ontwerp was namelijk nog niet ontworpen door de opdrachtgever Bas.

De epic 'Vraag stellen als koper' zal in volgende sprints verder afgerond worden door de gemaakte endpoints te implementeren binnen de Front-End van de customersportal. Hierdoor zal de connectie tussen de Front-End en de Back-End gerealiseerd worden.

## 7.3 Review

Sprint 1 had als doel om een prototype van de contextuele chat te ontwikkelen voor de kopers. Om dit goal te behalen zijn er verschillende epics aan deze sprint gekoppeld. Voor dit project zijn er grote epics opgesteld die ieder een user story bevatten. Deze epics zijn verdeeld over meerdere sprints en zijn hierom na deze sprint nog niet volledig behaald.

De epics gekoppeld aan de eerste sprint:

- Vraag stellen als koper
- Antwoord geven als kweker
- 100% digitaal
- Instelling om chat te gebruiken

Dankzij het maken van het UML-diagram, is het gelukt om de Back-End uit te bereiden met de entiteiten en endpoints die nodig zijn rondom het versturen en ophalen van berichten. Hiermee is er voor zowel de epic 'Vraag stellen als koper' als 'Antwoord geven als kweker' een goed begin gemaakt. Waarbij ook de Front-End voor de 'Vraag stellen als koper' epic staat.

Daarnaast is het voor de epic '100% digitaal' gelukt om de onderlinge communicatie tussen koper en kweker in de database vast te leggen. Hiermee is deze epic behaald.

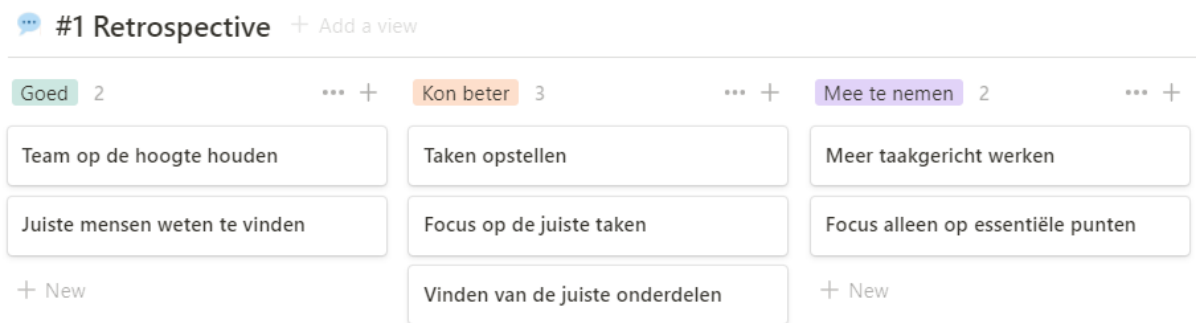
Ook is de epic 'Instellingen om chat te gebruiken' afgerond binnen deze sprint. Dit is gedaan door de eigenschap EnableConversation toe te voegen en de functionaliteiten rondom het versturen en ophalen van berichten en conversaties te beveiligen op basis van deze eigenschap.

In de review is gesproken over het net niet behalen van het sprintgoal. Het is gelukt om de Back-End service in te richten om zo berichten te kunnen versturen en te ontvangen. Ook zijn er voor de endpoints van deze functionaliteiten Unit tests geschreven. Verder is het uiterlijk van de chat in de customersportal gerealiseerd. Wat nog ontbreekt is de koppeling tussen de Back-End en Front-End.

Met de opdrachtgever is tijdens de review afgesproken om volgende sprint de koppeling tussen de Back-End en Front-End te realiseren. Ook zal de opdrachtgever het visuele ontwerp van de chat in de suppliersportal aanleveren waardoor er volgende week een zelfde soort prototype gemaakt wordt voor de kweker als voor de koper. Desondanks was de opdrachtgever tevreden met het huidige resultaat van de sprint.

## 7.4 Retrospective

Aan het eind van de sprint is er een retrospective gehouden met de bedrijfsbegeleider. Uit de retrospective zijn de volgende punten gekomen:



Figuur 9: Sprint 1 Retrospective

### Kon beter

Aan het begin van de sprint heb ik moeite gehad met het opstellen van taken op basis van de epics, hier was ik zoekende wanneer iets een taak is. Ook had ik moeite met een taak kort en bondig te beschrijven zodat die duidelijk voor mij was.

Verder werd ik tijdens mijn eerste sprint overdunderd door de omvang van alle aspecten van het digitale Floriday landschap (Bijlage 4: Ontwerp documentatie hoofdstuk 2 Systeemarchitectuur). Floriday heeft veel verschillende microservices, Front-End projecten en koppelingen. Hierdoor had ik moeite met het vinden van de juiste onderdelen en kon ik mij minder goed focussen op de essentiële sprint taken.

Omdat ik nog niet over alle domeinkennis beschik, ben ik veel aan het wisselen tussen verschillende taken en onderzoeken. Ik wil alles weten en begrijpen, wat ervoor zorgt dat ik veel tegelijkertijd wil doen. Ik vind het lastig om gefocust te blijven op de essentiële taken voor deze sprint.

### Goed

Ondanks ik nog niet over alle domeinkennis beschik, wist ik wel de juiste personen te vinden waar ik de benodigde informatie van kon krijgen. Hierdoor stelde ik de juiste vragen aan personen die mij daar het meeste bij kon helpen. Het team was continue op de hoogte van mijn voortgang door de daily-meet.

### Mee te nemen

Voor de volgende sprint wil ik taakgerichter gaan werken en alleen focussen op de essentiële taken van de sprint. Hierdoor zal de volgende sprint efficiënter en taakgerichter verlopen.

## 8. Sprint 2: Kweker chat prototype

Aan het begin van sprint 2 is er een doel opgesteld. Voorafgaande de sprint is er een Sprint backlog gemaakt, op basis van de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij deze sprint ingegaan worden op de realisatie werkzaamheden van het chat prototype voor de kweker. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 8.1 Doel

*Sprint 2 heeft als doel een prototype te ontwikkelen van de contextuele chat voor de kwekers. Met het prototype kunnen alle chats inzichtelijk gemaakt worden voor de kweker door middel van een chat overzicht. Daarnaast moet het mogelijk zijn voor kwekers om berichten te versturen naar kopers om zo hun vragen te kunnen beantwoorden*

Deze sprint doel is opgesteld op basis van de resultaten en openstaande punten van de vorige sprint, samen met de punten uit de review. Zoals beschreven in de review van sprint 1 (7.3 Review) is er gewerkt aan het chat prototype voor de customersportal, waar het visuele ontwerp al beschikbaar voor was. Daarnaast was de Back-End ook al ingericht met endpoints die zowel de kopers als de kwekers kunnen gebruiken voor het versturen en ophalen van berichten en conversaties.

De logische volgende stap is om een prototype binnen de suppliersportal te realiseren. Hiervoor moet een chatoverzicht met alle openstaande conversaties en het chat venster zelf gerealiseerd worden. Nu het visuele ontwerp van de suppliersportal (Bijlage 4: Ontwerp documentatie, Hoofdstuk 5.2) aangeleverd is door de opdrachtgever, kan de Front-End hiervan gerealiseerd worden. Zoals beschreven in de review van sprint 1 (7.3 Review) zal in deze sprint het doel zijn om de koppeling tussen de Back-End en beide Front-End projecten te realiseren, om zo de chat daadwerkelijk functioneel te maken.

### 8.2 User stories

Titel	User story	Prioriteit
Vraag stellen als koper	Als koper, wil ik een vraag in vrije tekst kunnen stellen aan een kweker op zijn winkel omgeving, zodat ik antwoord in vrije tekst kan krijgen op mijn vraag.	Must have
Antwoord geven als kweker	Als kweker, wil ik antwoord kunnen geven op vragen van kopers, zodat ik de koper kan helpen bij het plaatsen van een bestelling.	Must have
Chat overzicht voor kweker	Als kweker, wil ik een overzicht hebben van alle chats, zodat ik in een oogopslag kan zien welke chats er open staan.	Must have
Integratie in het platform	Als ontwikkelaar van het Floriday platform, wil ik dat de contextuele chat geïntegreerd wordt binnen het platform zonder dat het impact heeft op huidige functionaliteiten, zodat de rest van het platform operationeel blijft.	Must have

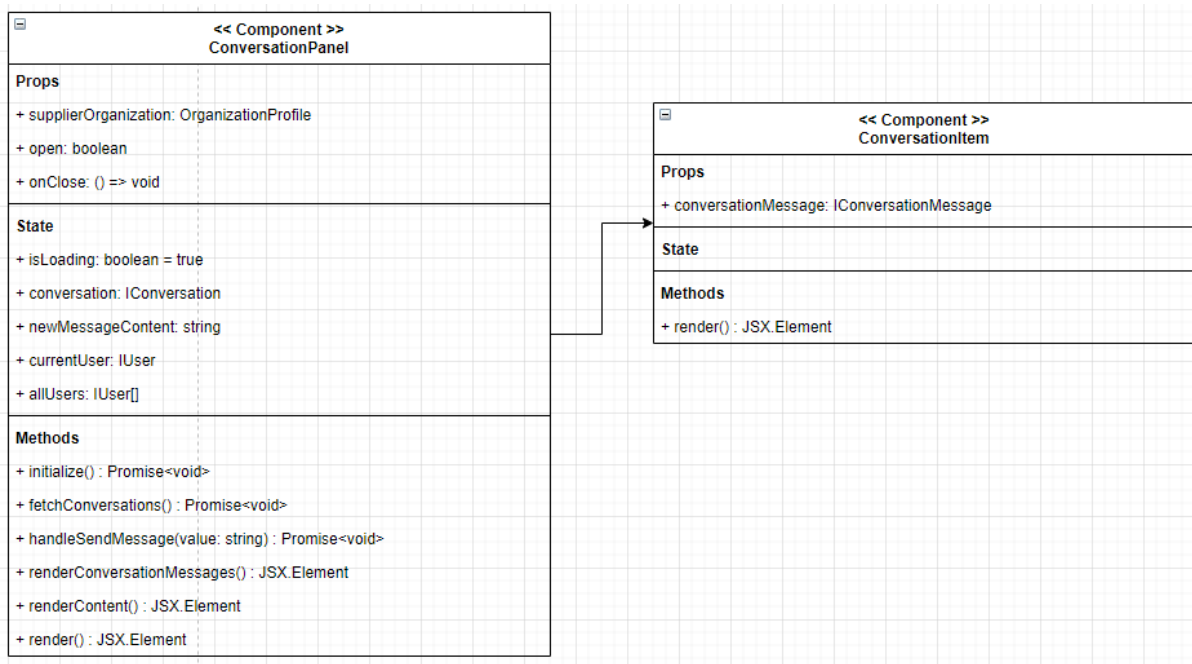


Zoals te lezen in het sprint doel wordt er deze sprint verder gewerkt aan de openstaande epics van de vorige sprint: 'Vraag stellen als koper' en 'Antwoord geven als kweker'. Bij het prototype voor de kweker hoort ook een overzicht van de chat schermen, om deze reden zal er gewerkt worden aan het 'Chat overzicht voor kweker'. Tot slot zal er gewerkt worden aan 'Integratie in het platform' omdat het van belang is dat er geen conflict ontstaat tussen de al gebouwde onderdelen.

### Vraag stellen als koper

In hoofdstuk 7. Sprint 1: Koper chat prototype is er een UML-diagram gemaakt waarin het ophalen en versturen van berichten voor de koper en kweker in verwerkt is. Dit UML-diagram is gerealiseerd en getest in de Back-End. Daarnaast is het visuele ontwerp, gemaakt door de opdrachtgever Bas, gerealiseerd van de customersportal.

In deze sprint is de epic 'Vraag stellen als koper' verder opgepakt door de Front-End van de customersportal te koppelen aan de endpoints binnen de M&C Back-End service. Vorige sprint was daar helaas nog geen tijd voor. Door middel van de gegenereerde API client, zoals genoemd in hoofdstuk 5.3.1 Front-End, kunnen de endpoints uit de Back-End services aangeroepen worden door de customersportal.



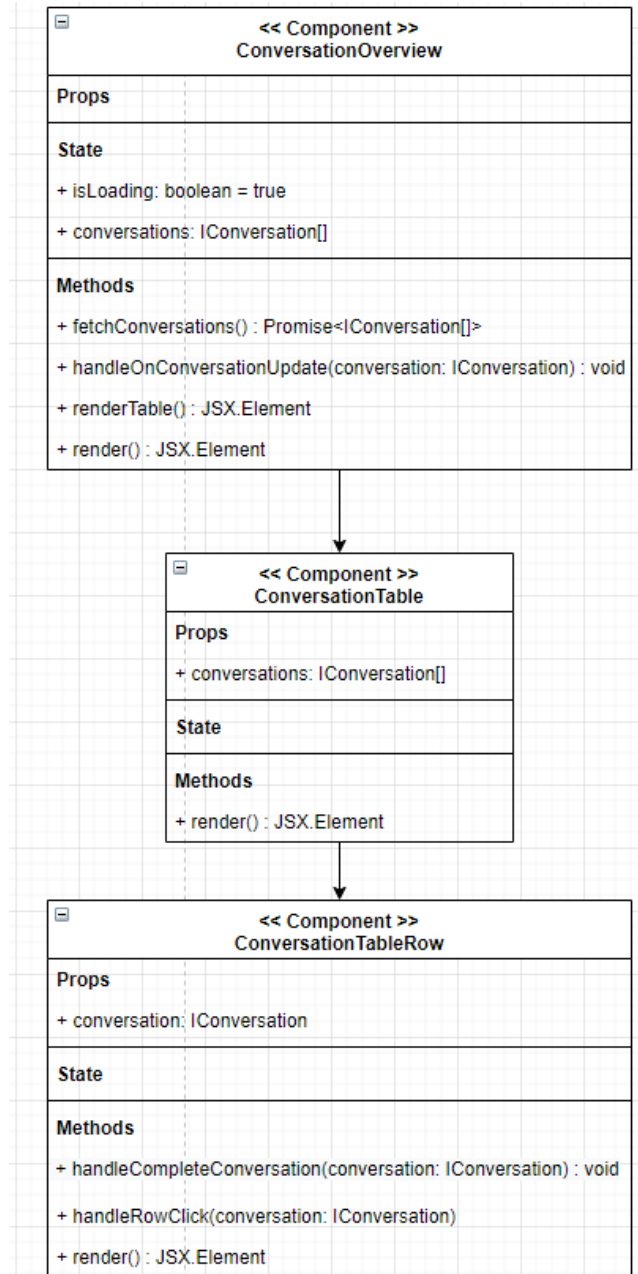
Figuur 10: React componenten diagram, Conversation Panel

Verder is er, zoals te zien is in Figuur 10: React componenten diagram, Conversation Panel, meer structuur gebracht aan de React componenten binnen de Front-End van de customersportal.

### Chat overzicht voor kweker

Voor de kweker is het van belang dat er een overzicht is van alle actieve conversaties, zo kan hij in een oogopslag zien welke conversaties er open staan.

Zoals te zien is in het visuele ontwerp (Bijlage 4: Ontwerp documentatie, Hoofdstuk 5.2), ontvangt de kweker een vraag vanuit de koper. Wanneer een vraag binnenkomt vanuit de koper wordt er een nieuwe conversatie aangemaakt en wordt deze getoond in het overzicht. In dit overzicht kan de kweker alle benodigde informatie zien van een conversatie. Wanneer de kweker op een conversatie in het overzicht klikt gaat hij naar het daadwerkelijke chat scherm.



Figuur 11: React componenten diagram, Conversation Overview

In bovenstaand Figuur 11: React componenten diagram, Conversation Overview is de componenten structuur van het conversatie overzicht voor de kweker te zien. Deze bestaat uit verschillende React componenten. Het conversatie overzicht is opgebouwd uit de volgende componenten:

ConversationOverview, ConversationTable en ConversationTableRow. Binnen het ConversationOverview worden de conversaties opgehaald, die weer gedistribueerd worden in de ConversationTable. Voor het renderen van de specifieke rij in de tabel is ook een apart component aangemaakt, ConversationTableRow. De ConversationTable had dit ook kunnen doen, alleen toont

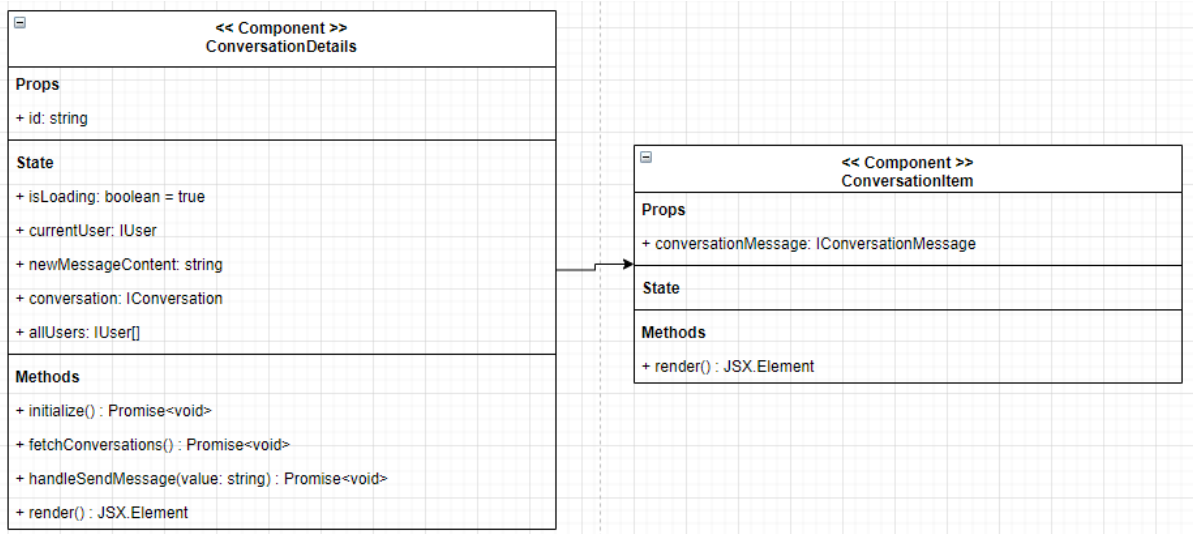
het overzichtelijker wanneer dit in een los component gebeurt. Dit zorgt ervoor dat alle componenten enkel hun eigen verantwoordelijkheden hebben.

Om ervoor te zorgen dat de componenten niet extra verantwoordelijk zijn voor het ophalen en verwerken van data, is er een logic file aangemaakt. Deze logic file bevat alle logica om de endpoints van de Back-End aan te roepen om zo de data, betreft de conversaties en berichten, op te kunnen halen en te verwerken.

### Antwoord geven als kweker

Net zoals voor de epic 'Vraag stellen als koper' is er voor deze epic 'Antwoord geven als kweker' in de vorige sprint een UML-diagram gemaakt, gerealiseerd en getest in de Back-End.

In deze sprint is het visuele ontwerp (Bijlage 4: Ontwerp documentatie, Hoofdstuk 5.2) voor de suppliersportal wel beschikbaar. Deze is dan ook in de Front-End van de suppliersportal gerealiseerd. Zoals te lezen is in *Chat overzicht voor kweker*, kan na het selecteren van een conversatie in het overzicht, antwoord gegeven worden door de kweker. Het uiterlijk daadwerkelijke chatvenster is voor zowel de koper als de kweker gelijk. Het enige verschil is dat het chatvenster voor de koper geïntegreerd zit in het *Direct inkopen scherm* en het chatvenster voor de kweker een volledige pagina is. Verder is het uiterlijk en de functionaliteit van het chat venster gelijk voor beide partijen.



Figuur 12: React componenten diagram, Conversation Details

In bovenstaand Figuur 12: React componenten diagram, Conversation Details is de componenten structuur van het conversatie scherm voor de kweker te zien. Dit conversatie scherm bestaat uit de componenten ConversationDetails en ConversationItem.

Het ConversationDetails component omvat alles omtrent een specifieke conversatie tussen een koper en kweker. Hiermee is te zien met wie de conversatie is, welke berichten binnen de conversatie zijn en is er een onderdeel waarin berichten getypt en verstuurd kunnen worden. Om ervoor te zorgen dat de berichten binnen een conversatie altijd op een structureerde manier getoond worden, is er een ConversationItem component aangemaakt.

Net zoals de ConversationOverview wordt ook voor ConversationDetails gebruik gemaakt van hetzelfde logic file. Deze logic file bevat alle logica om de endpoints van de Back-End aan te roepen om zo de data, betreft de conversaties en berichten, op te kunnen halen en te verwerken.

### Integratie in het platform

Deze sprint is de student achter een probleem gekomen dat betrekking heeft met de naamgeving van de entiteiten zoals ze opgesteld zijn in de vorige sprint.

Zoals genoemd in de vorige sprint is er een UML-diagram opgesteld. In dit diagram is er voor de duidelijke naamgeving van de entiteiten gekozen voor *Conversation* en *Messages*. Voordat dit geïmplementeerd werd in de Back-End is het ontwerp en de uiteindelijke code gereviewd door meerdere collega's. Nadat alles gereviewd en goedgekeurd was is het geïmplementeerd doorgezet naar de staging omgeving.

Gedurende deze sprint is de koppeling tussen de Back-End en Front-End gerealiseerd en is er een functionele chat ontstaan. Bij de realisatie van deze koppeling is er ontdekt dat de naam *Message* in de Front-End projecten al vaker gebruikt werd door externe libraries, ondanks er geen overeenkomst was in de Back-End.

Om deze reden is er gekozen voor een andere naamgeving van de *Message* entiteit, namelijk *ConversatioMessage*. Voor JEM-id is dit een gebruikelijke naamgeving voor entiteiten bij een 1 op veel relatie. Door deze aanpassing is het naamgeving probleem verlopen tussen de libraries en de nieuwe entiteiten.

## 8.3 Review

Sprint 2 had als doel om de chat functionaliteit aan de kwekers kant te realiseren. Om dit doel te behalen zijn er verschillende epics aan deze sprint gekoppeld. Net zoals sprint 1 zijn dit grote epics die verdeeld zijn over meerdere sprints. Deze zijn hierom na deze sprint nog niet volledig behaald.

De epics gekoppeld aan de tweede sprint:

- Vraag stellen als koper
- Antwoord geven als kweker
- Chat overzicht voor kweker
- Integratie in het platform

Het sprint doel voor de tweede sprint is behaald, de functionaliteit aan de kwekers kant is gerealiseerd. Samen met de koppeling tussen de Back-End en Front-End voor zowel de customersportal als de suppliersportal. Ook is de Front-End voor de suppliersportal gemaakt, het chat overzicht en chatvenster zijn omgezet en functioneel gemaakt.

Tijdens de review is er, naast het behalen van het sprint doel, gesproken over verschillende bugs die tot het licht zijn gekomen. Doordat tijdens het plannen van deze sprint het werk onderschat is, is er voornamelijk gewerkt aan het neerzetten van de functionaliteiten. Hierdoor was er minder tijd voor het finetunen en is het resultaat nog niet helemaal gelijk aan het visuele ontwerp.

In de volgende sprint zal er gefinetuned worden waar geen tijd voor was in deze sprint. Ook zullen alle bugs worden weggewerkt zodat de chat naar behoren werkt. Het resultaat zal hierbij ook gelijk worden getrokken aan het visuele ontwerp.

## 8.4 Retrospective



Figuur 13: Sprint 2 Retrospective

### Goed

In sprint 2 is er veel werk afgekomen, voor zowel de customersportal als suppliersportal. Er kunnen berichten verstuurd en ontvangen worden. De Front-End voor beide portalen zijn op een paar kleine punten na afgerond. Ik heb meer taak gerelateerd gewerkt dan vorige sprint, hierdoor is er ook meer werk afgekomen.

### Kon beter

Gedurende deze sprint merkte ik dat ik naast de taken in de sprint weinig tijd over was voor het schrijven van mijn verslag. Hier zal ik dan ook de volgende sprint wat tijd voor meerekenen.

Zoals te lezen is onder *Integratie in platform* van Sprint 2, was er een probleem met de naamgeving, wat resulteerde in een bug die deze sprint opgelost moest worden. Ook hier was oorspronkelijk geen tijd voor ingerekend.

### Mee te nemen

Bij de volgende sprint zal ik voor een betere sprintplanning zorgen, zodat ik zeker weet dat ik de werk load aan kan. Hierdoor zal er ook rekening worden gehouden met mogelijke extra taken en discussies. Met het plannen van de volgende sprint zal ik de taken gaan inschatten zodat ik inzicht heb ik hoeveel ik op kan pakken tijdens sprint 3. Dit wil ik gaan doen volgens de Fibonacci sequence (Muldoon, 2013).

## 9. Sprint 3: Prototypes finetunen

Aan het begin van sprint 3 is er een doel opgesteld waar tijdens de sprint aan gewerkt wordt. Voorafgaande de sprint is er een Sprint backlog gemaakt, op basis van de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij deze sprint ingegaan worden op het finetunen van de prototypes uit de vorige sprints. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 9.1 Doel

*Sprint 3 heeft als doel om de huidige chat te verbeteren, verifiëren en te testen. Vanuit sprint 2 staat er al heel wat, maar er moet ook zeker worden geweten dat wat er nu staat, ook werkt. Zodat er na deze sprint een beeld is van wat er staat en wat er nog gedaan moet worden om aan de wensen uit het visiedocument te voldoen.*

Zoals benoemd in de review van sprint 2, onder hoofdstuk 8.3 Review, zal deze sprint draaien om het finetunen en oplossen van bugs. Zo werken de hoofd functionaliteiten van de chat maar moeten deze nog gefinetuned worden. Ook moet het visuele ontwerp pixel perfect omgezet worden.

### 9.2 User stories

Titel	User story	Prioriteit
Vraag stellen als koper	Als koper, wil ik een vraag in vrije tekst kunnen stellen aan een kweker op zijn winkel omgeving, zodat ik antwoord in vrije tekst kan krijgen op mijn vraag.	Must have
Antwoord geven als kweker	Als kweker, wil ik antwoord kunnen geven op vragen van kopers, zodat ik de koper kan helpen bij het plaatsen van een bestelling.	Must have
Chat overzicht voor kweker	Als koper, wil ik een overzicht hebben van alle chats, zodat ik in een oogopslag kan zien welke chats er open staan.	Must have
Integratie in het platform	Als ontwikkelaar van het Floriday platform, wil ik dat de contextuele chat geïntegreerd wordt binnen het platform zonder dat het impact heeft op huidige functionaliteiten, zodat de rest van het platform operationeel blijft.	Must have

Aangezien de werkzaamheden voor deze sprint verder gaan op sprint 2, zal er hier gewerkt worden aan dezelfde epics. Deze zullen dan ook in deze sprint afgewerkt worden.

#### Verschillen in visuele ontwerpen recht trekken

Deze sprint staat in het teken van de prototypes binnen de Suppliersportal en Customersportal te finetunen. Er zijn een aantal verschillen geconstateerd tussen het ontwerp voor de chat vensters van de koper en de kweker. Voorafgaand aan de sprint is er voorgesteld aan de opdrachtgever om deze verschillen recht te trekken zodat beide type gebruikers dezelfde gebruikerservaring hebben tijdens de conversatie.

Deze verschillen gaan voornamelijk over het uiterlijk van het chat venster. In het visuele ontwerp voor de Customersportal hadden de berichten van de koper een groen uiterlijk en de berichten van de kweker een neutraal uiterlijk. Hierdoor was het verschil tussen eigen berichten en de berichten van een ander duidelijk. Bij het visuele ontwerp van de kweker hadden beide berichten een neutraal uiterlijk. Om de chat functionaliteit consistent te houden voor beide gebruikers, is er voorgesteld om het verschil recht te trekken. Voor zowel de customersportal als suppliersportal zijn eigen berichten nu groen en berichten van een ander neutraal.

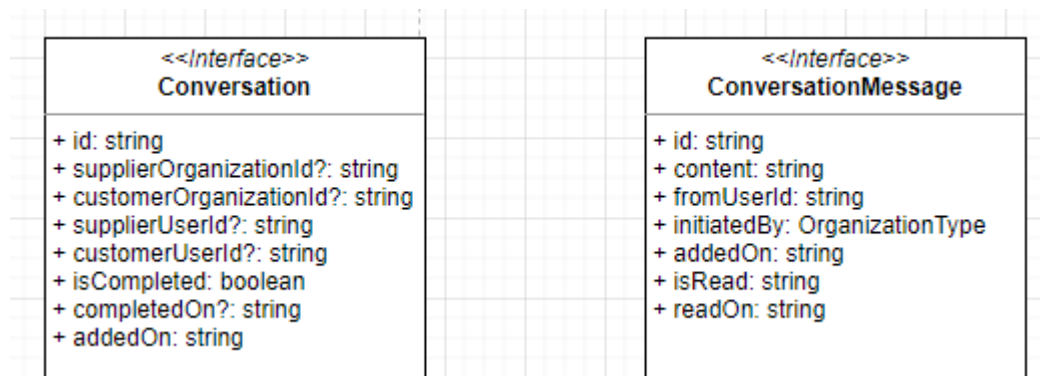
### Front-End datastructuur

In deze sprint is er nog meer structuur gebracht aan de Front-End van zowel de customersportal al de suppliersportal.

Voor de logica van de Conversation en ConversationMessages is er een Logic file aangemaakt, waar de endpoints in worden aangeroepen en waar de data naar de juiste objecten omgezet wordt. Hierdoor kunnen de UI componenten de juiste data opvragen door een methode aan te roepen binnen de logic file.

Door de logica van de Conversation en ConversationMessages op te splitsen wordt dubbele code voorkomen. Verder zijn de componenten zo opgesplitst dat ze maar een verantwoordelijkheid hebben, en eventueel uitgebreid kunnen worden zonder veel aanpassingen nodig te hebben. Bovendien wordt er gebruik gemaakt van interfaces en abstracte classes en objecten. Dit alles zorgt ervoor dat de DRY en SOLID (Likness, 2009) ontwikkel principes nagestreefd worden.

Echter was het uitlezen van de juiste data in de UI nog een uitdaging. Er werd namelijk gebruik gemaakt van Back-End interfaces om objecten in de Front-End aan te maken. Deze objecten werden dan direct gebruikt om de UI te vullen. Dit bracht problemen met zich mee.



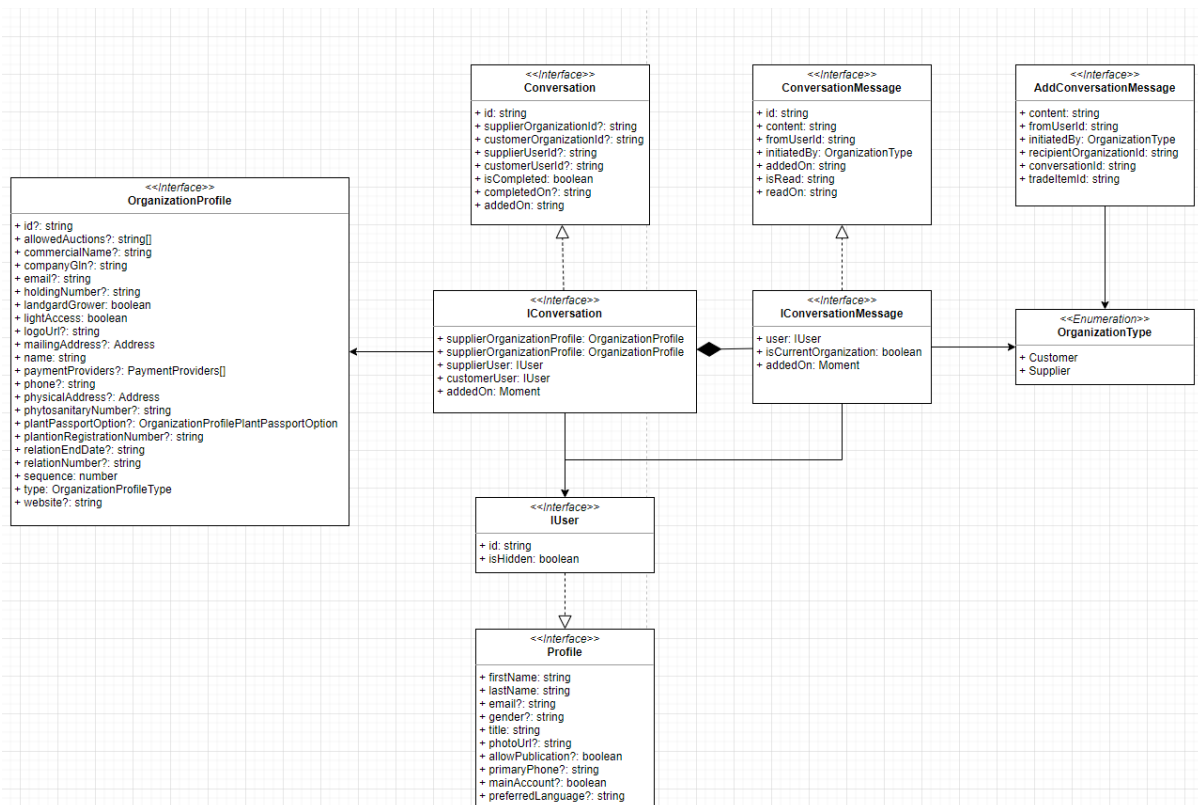
*Figuur 14: Interfaces vanuit Back-End*

In Figuur 14: Interfaces vanuit Back-End zijn de Conversation en ConversationMessage interfaces te zien. Deze interfaces zijn direct afkomstig van de Back-End entiteiten. Zoals te zien is bevatten deze interfaces vrijwel alleen maar platte data, zoals bijvoorbeeld suppliersOrganizationId, customerUserId en addedOn. De UI heeft natuurlijk meer nodig dan alleen een OrganizationId, zo is er bijvoorbeeld ook de naam en het logo nodig van een organisatie.

Normaliter zouden de OrganizationId's en UserId's als foreign key's in de database staan. Daarmee zou alle data betreft de Organizations en Users mee opgevraagd kunnen worden door middel van JOIN's. Echter maakt Floriday gebruik van een microservice architectuur en staat daarom de data betreft de Organizations en Users in een andere microservice. Vandaar dat ze opgeslagen worden als

losse Id's in de match & connect service. Gelukkig kan de Front-End wel de benodigde data van de Organizations en Users ophalen door de desbetreffende endpoints aan te roepen van de services.

Om ervoor te zorgen dat de UI altijd toegang heeft tot de benodigde data zijn er extra interfaces aangemaakt.



*Figuur 15: Uitgebreide Front-End interfaces*

Zoals in Figuur 15 te zien is, zijn de `IConversation` en `IConversationMessage` interface toegevoegd. Deze interfaces overerven van de oorspronkelijke interfaces uit de Back-End.

De nieuwe IConversation interface vervangt de OrganizationId's voor OrganizationProfile objecten en de UserId's voor de IUser objecten. Zoals in het figuur te zien is, hebben IConversation objecten toegang tot alle data van de OrganizationProfile en IUser. IUser overerft ook weer van de Profile interface wat ervoor zorgt dat er nog veel meer data ter beschikking is van een User.

De `IMessage` interface vervangt ook de `UserId` voor een `IUser` object. Daarnaast vervangt de `isCurrentOrganization` de `InitiatedBy` eigenschap. Dit maakt het gemakkelijk voor de UI om het onderscheid tussen berichten te renderen.

Verder is de het type van de AddedOn eigenschap voor beide interfaces vervangen van een string naar een Moment object. Met Moment.js (Moment.js, 2021) kan er een consistente weergave gegeven worden van de tijd. Bijvoorbeeld, in plaats van een ruwe tijdafdruk zoals: *2021-01-01T00:00:00+00:00*, kan er met een Moment object gemakkelijk een afdruk gegeven worden als: *Vandaag om 12:00*, of *14-06 om 12:00*.

### Overige werkzaamheden

In deze sprint vond ook het bedrijfsbezoek plaats, waarin het eerste deel van het afstudeerdossier af moest zijn. Om die reden is daar meer tijd aan besteed.



Als laatste heeft de student de wekelijkse livestream gehost van Floriday, genaamd de FloridayByDay. Hier presenteren twee collega's een onderwerp waar hij/zij mee bezig is via een livestream. Deze livestream wordt bekeken door de gehele organisatie. Tijdens deze livestream heeft de student de afstudeeropdracht en de voortgang hiervan gepresenteerd. Helaas kon dit niet voorzien worden, aangezien elke week een tweetal collega's aangesteld worden. De voorbereiding en uitvoering hiervan heeft daarom tijd van de sprint ingenomen.

### 9.3 Review

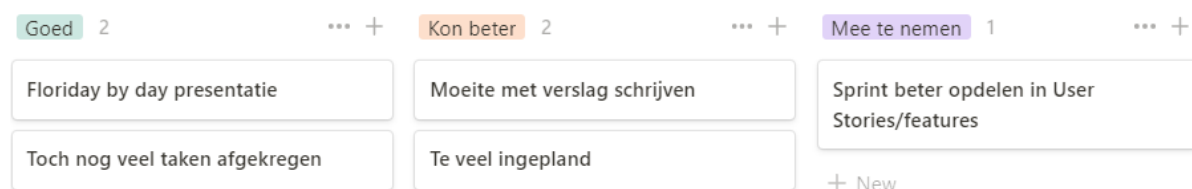
Sprint 3 had als doel om de huidige chat te verbeteren, verifiëren en te testen. Zodat er na de sprint een beeld was van wat er staat en wat er nog gedaan moet worden om aan de wensen uit het visiedocument te voldoen. Na sprint 3 zijn de grotere epics, die ook benoemd staan in sprint 1 en 2, volledig behaald. Het is nu mogelijk voor zowel de koper als de kweker beide berichten te kunnen versturen, te ontvangen en antwoord te geven. Ook zijn alle unit testen beschreven.

De epics gekoppeld aan de derde sprint:

- Vraag stellen als koper
- Antwoord geven als kweker
- Chat overzicht voor kweker
- Integratie in het platform

In de review is gesproken over het behalen van het sprint doel van sprint 3. De opdrachtgever is blij met het neer zetten van de hoofd functionaliteiten, alleen wordt het gevoel van een echte chat nog gemist. Om het gevoel van een real-time chat te creëren moeten nieuwe berichten automatisch getoond worden op het scherm. Dit is een goede en belangrijke volgende stap voor het project.

### 9.4 Retrospective



Figuur 16: Sprint 3 Retrospective

#### Goed

De hoofdfunctionaliteiten voor de chat zijn voltooid. Door dit te voltooien heb ik ook meteen alle taken afgerond die ik deze sprint had ingepland.

Iedere week heeft Floriday een Floriday by Day, hier presenteren twee collega's een onderwerp waar hij/zij mee bezig is via een livestream. Deze livestream wordt bekeken door de gehele organisatie. Deze week werd ik uitgenodigd om mijn afstudeerproject en de voortgang ervan te presenteren bij de Floriday by Day. Ik heb nog nooit voor zoveel mensen een livestream gedaan, daarom vond ik het ook spannend om dit te doen. Desondanks ging het onwijs goed en is het door collega's ook goed ontvangen. Ze vonden het leuk om op deze manier meer te weten te komen over mijn project en wisten nu zelfs collega's buiten mijn team om ook goed waar ik mee bezig was.

### **Kon beter**

Gedurende deze sprint merkte ik, door het uitschrijven van vorige sprints, dat ik steeds meer moeite heb met het schrijven van mijn verslag. Het vastleggen hiervan is iets wat voor mij erg lastig is, wel heb ik hier een goede manier in gevonden. Tijdens dit project maak ik gebruik van het programma Notion (Notion Labs, 2021). Hierin documenteer ik mijn volledige sprint en leg ik meetings, gebeurtenissen en beslismomenten vast.

Ook had ik deze sprint te veel taken ingepland. Zo ben ik niet toegekomen aan het schrijven van een integratietest, deze zal meegenomen worden naar sprint 4.

### **Mee te nemen**

Zoals de afgelopen drie sprints gemerkt heb werk ik met te grootte user stories die gedefinieerd zijn als epics. Iedere sprint heb ik verschillende taken onder de epics gehangen, die iedere sprint afkwamen. Alleen heb ik hierbij niet de epic volledig af kunnen ronden aangezien deze te groot zijn.

In sprint 3 ben ik al gestart met het beter inplannen van de taken door middel van de Fibonacci sequence (Muldoon, 2013). Om het probleem van te grootte epics te verhelpen, wil ik vanaf sprint 4 de epics opdelen in kleinere user stories die binnen 1 sprint opgepakt kunnen worden. Hierdoor zal ik vanaf sprint 4 onderscheid maken tussen epics, features en taken. Op deze manier kan ik meerdere features inplannen en ook daadwerkelijk in een sprint afronden.

## 10. Sprint 4: Automatisch berichten ophalen

Aan het begin van sprint 4 is er een doel opgesteld waar tijdens de sprint aan gewerkt wordt. Voorafgaande de sprint is er een Sprint backlog gemaakt, op basis van de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij deze sprint ingegaan worden op het automatiseren van het ophalen van berichten. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 10.1 Doel

Sprint 4 heeft als doel om de bestaande chat functie te herzien op basis van het automatisch ophalen van berichten. Het automatisch ophalen van berichten zal de gebruikers het gevoel van een real-time chat geven.

Zoals te lezen is in hoofdstuk 6.5 Real-time chat mogelijkheden, is het implementeren van een event based techniek, zoals SignalR (Microsoft, 2021), nog niet mogelijk. Desondanks wordt er in deze sprint gekeken naar andere mogelijkheden, om op die manier alsnog het real-time chat gevoel te realiseren.

Daarnaast gaat er ook gekeken worden naar de functionaliteit om het lezen van berichten vast te leggen. Dit is een functionaliteit die in andere real-time chats gebruikt wordt, gebruikers zijn er hierdoor ook aan gewend. Wanneer deze functionaliteit geïmplementeerd kan worden zal de kans groot zijn dat de gebruiker het gevoel van een real-time chat meer ervaart.

### 10.2 User stories

# Estimati...	Prioriteit	Name	User Story
16	2. Should have	Integratie test in API-tester	Als ontwikkelaar, wil een integratietest schrijven, zodat ik de integratie van de gemaakte endpoints kan valideren.
17	1. Must have	Berichten automatisch ophalen en weergeven in chat	Als kweker of koper, wil ik dat de berichten binnen een conversatie automatisch opgehaald en weergegeven worden, zodat ik zelf geen extra actie hoeft te voeren om de berichten op te halen en weer te geven.
12	3. Could have	Lezen van berichten vastleggen	Als kweker of koper, wil ik zien of mijn berichten gelezen zijn, zodat ik het gevoel van een real-time chat krijg en weet dat iemand mijn bericht heeft gelezen.

Figuur 17: Sprint 4, Features in Sprint backlog

In deze sprint is er gewerkt met het opstellen van kleinere features in plaats van epics. In de afgelopen drie sprints ervaarde ik veel moeite met het afronden van epics. De reden hiervoor was dat de epics grote overkoepelende punten zijn, die niet in een sprint afgerond kunnen worden. In deze sprint is er voor het eerst gewerkt met het opsplitsen van epics in kleinere features, waar weer kleinere taken onder hangen.

## Real-time chat

Momenteel zien de koper en kweker enkel nieuwe berichten na het verversen van de pagina. Voor de gebruikerservaring ontbreekt hier het gevoel van een real-time chat. Een real-time chat zou de gebruiker het gevoel moeten geven dat de UI automatisch geüpdatet wordt wanneer er een nieuw bericht binnen gekomen is. Om dit gevoel te kunnen creëren is er wederom gekeken of een event based techniek zoals SignalR (Microsoft, 2021) mogelijk was. Deze overweging is te lezen in 6.5

Real-time chat mogelijkheden. De implementatie hiervan is nog steeds niet mogelijk. Wel zou dit in de toekomst de manier zijn om het gevoel en werking van een real-time chat te kunnen realiseren. Ook zal deze oplossing het aantal calls naar de database verminderen, omdat er enkel om specifieke data gevraagd wordt wanneer er iets nieuws is.

Desondanks het niet mogelijk was om met een event based techniek zoals SignalR te werken, is er toch gezocht naar een alternatieve oplossing om het gevoel van een real-time chat na te bootsen voor de gebruiker. Er is ervoor gekozen om door middel van een interval om de paar seconden de nieuwe berichten op te halen. Deze interval is geïmplementeerd in de customersportal en suppliersportal.

Conversation	
Id: Guid	
SupplierOrganizationId: Guid	
CustomerOrganizationId: Guid	
SupplierOrganizationUserId: Guid	
CustomerOrganizationUserId: Guid	
Added: DateTimeOffset	
Completed: DateTimeOffset	
IsCompleted: bool	
+ GetConversationsForSupplier(Guid: supplierOrganizationId, bool?: showCompleted, bool?:showForCurrentUser): GET + GetConversationByIdForSupplier(Guid: conversationId): GET + GetCurrentConversationForCustomer(Guid: customerOrganizationId, Guid: supplierOrganizationId): GET + CompleteConversationForSupplier(Conversation: conversation): PUT - GetConversationMessages(Guid: organizationId, OrganizationType: organizationType, Guid: conversationId, DateTimeOffset?: since): PUT + GetConversationMessagesForSupplier(Guid: supplierOrganizationId, Guid: conversationId, DateTimeOffset?: since): PUT + GetConversationMessagesForCustomer(Guid: customerOrganizationId, Guid: conversationId, DateTimeOffset?: since): PUT	

Figuur 18: Conversation class uitbreiding

Zoals te zien in bovenstaand Figuur 18: Conversation class uitbreiding zijn dit de GetConversationMessagesForSupplier en de GetConversationMessagesForCustomer endpoints. Om ervoor te zorgen dat alleen de nieuwe berichten opgevraagd worden en de oude berichten niet dubbel opgevraagd worden zijn deze endpoints aangemaakt. Binnen deze endpoints staat een parameter 'since' die ervoor zorgt dat alleen nieuwe berichten worden opgevraagd.

De oplossing van deze interval met nieuwe endpoints zorgt voor het gevoel van een real-time chat en vermindert de load en hoeveelheid calls die uitgevoerd worden op de database.

### Lezen van berichten vastleggen

Om het gevoel van interactie in de real-time chat te creëren is er een nieuwe feature voorgelegd aan de opdrachtgever. Namelijk het lezen van berichten kunnen vastleggen. Dit zou betekenen dat wanneer er een bericht gelezen wordt (door koper of kweker) hier een indicatie van bij komt te staan. Deze feature is iets wat gebruikers ook herkennen vanuit sociaal media platformen zoals WhatsApp.

Aan de class ConversationMessage is de eigenschap HasRead toegevoegd. Ook is er een SetConversationMessage endpoint aangemaakt voor de koper en de kweker. Wanneer de koper of kweker actief is in het conversatie scherm worden de berichten op gelezen gezet. Door deze toevoeging wordt het gevoel van een interactieve real-time chat verbeterd.

### Integratietest

In het afstudeerplan is aangegeven dat er een integratietest gedaan zou worden. Binnen het digitale Floriday landschap is er een service opgezet waarin enkel testen staan die de endpoints en de functionaliteiten van andere services testen. De tests binnen deze service worden automatisch uitgevoerd wanneer er een nieuwe versie van een andere service live wordt gezet. De tests valideren de werking van de externe services alsof een externe client deze ook zou benaderen.

Deze service bleek de ideale plek te zijn om de functionaliteiten van de nieuwe endpoints en de integratie hiervan te testen en te valideren.

Helaas bleek deze service alleen endpoints van services te kunnen benaderen die bedoeld zijn voor externe softwareleveranciers. Floriday biedt natuurlijk Front-End projecten aan voor de koper en de kweker, maar daarnaast biedt Floriday ook specifieke functionaliteiten direct aan externe softwareleveranciers. Er zijn namelijk bepaalde kopers en kwekers organisaties die al een eigen softwarepakket hebben en dus niet gebruik willen maken van de Front-End projecten, maar wel gebruik willen maken van bepaalde functionaliteiten van Floriday binnen hun eigen softwarepakket.

Om die reden is ervoor gekozen om de integratietest niet meer uit te voeren. Daarvoor in de plaats is er wel voor elke endpoint een Unit test geschreven om alsnog de werking van de endpoints te kunnen valideren. Ook zal er in de volgende sprints gekeken worden naar het uitvoeren van gebruikerstesten om zo de gebruikerservaring te kunnen testen en valideren.

## 10.3 Review

Sprint 4 had als doel om berichten automatisch op te halen om zo het gevoel van een real-time chat te creëren. Dit is de eerste sprint waar gewerkt wordt met features in plaats van epics. Om het sprint doel te behalen zijn verschillende features aangemaakt die in deze sprint opgepakt zijn.

De features die in deze sprint zijn opgepakt:

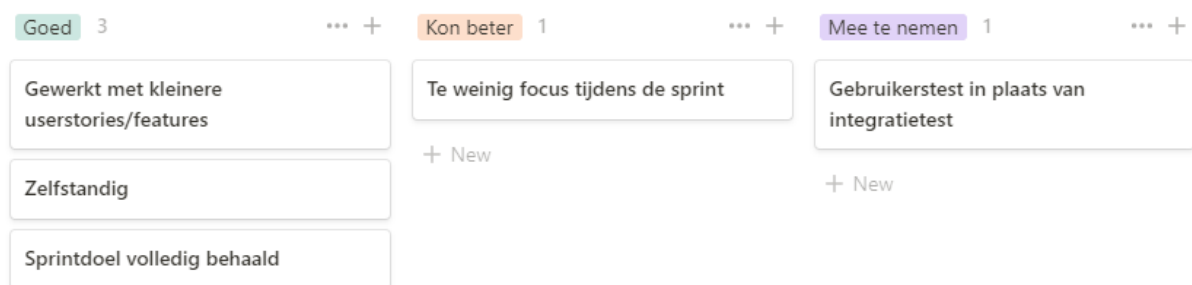
- Berichten automatisch ophalen en weergeven in chat
- Lezen van berichten vastleggen
- Integratie test in API-tester

Het sprint doel, het automatisch ophalen van berichten en het creëren van een real-time chat, is deze sprint behaald. Tijdens de review met de opdrachtgever is gesproken over de manier waarop

het deze sprint gelukt is om automatisch berichten op te halen, door middel van de interval. De opdrachtgever was tevreden met deze huidige oplossing, de gebruiker heeft het gevoel van een real-time chat en de server wordt niet overbelast. Ook voegt de functionaliteit van het vatsleggen van de gelezen berichten veel toe aan het real-time gevoel van de chat. Desondanks deze feature niet direct vanuit de opdrachtgever is gekomen, is hij wel erg tevreden deze toevoeging.

Zoals te lezen is in *Integratie test in API-tester* is dit niet gelukt. De API tester waar de Integratie test geschreven moest worden bleek uiteindelijk niet geschikt hiervoor. Ook is er niet veel integratie geweest omdat er gebruik wordt gemaakt van een bestaande service. Daarnaast is de werking van elke endpoint getest door middel van unittests. Een logische stap zou zijn om de werking van de chat en de vooraf gedefinieerde wensen te valideren. Er is dan ook met de opdrachtgever afgesproken hiervoor een usability test te doen.

## 10.4 Retrospective



Figuur 19: Sprint 4 Retrospective

### Goed

Nadat ik in sprint 1 tot en met 3 heb gewerkt met grote epics, was sprint 4 de eerste sprint waar ik epics opdeelde in kleine features. Hierna wees ik taken toe aan deze features. Voor mij werkte dit veel beter, ik kon binnen een sprint ergens naartoe werken en had ook een voldaan gevoel na het behalen van de features. In de sprints hiervoor had ik dit voldane gevoel enkel na de derde sprint, toen ik alle epics afgerond had.

Per sprint heb ik gemerkt dat ik steeds zelfstandiger word. Dit is dan ook deze sprint extra opvallend aangezien ik zelf een feature heb bedacht waar de opdrachtgever in eerste instantie niet zelf mee gekomen is. Wel draagt deze feature bij aan het gevoel van een real-time chat en was de opdrachtgever hier erg blij mee. Ook is het sprintdoel deze sprint volledig behaald.

### Kon beter

Deze sprint heb ik gemerkt dat ik minder focus had dan andere sprints. Dit kwam doordat ik een aantal dagen ziek was en weinig energie had aangezien ik ook nog in quarantaine zat.

### Mee te nemen

Zoals te lezen in de review kon de integratietest niet plaatsvinden. Wel is er afgestemd om een usability test uit te voeren met gebruikers. Op deze manier worden de uiteindelijk gemaakte functionaliteiten getest en gevalideerd met echte gebruikers.

Zoals te lezen in de review van deze sprint ga ik gebruikerstesten uitvoeren in plaats van een integratietest.

## 11. Sprint 5: Vooraf gedefinieerde antwoorden voor kweker

Voorafgaand aan deze sprint is er een doel opgesteld waar tijdens de sprint aan gewerkt wordt. Voorafgaande de sprint is er een Sprint backlog gemaakt, op basis van de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij deze sprint ingegaan worden op het implementeren van de contextuele chat oplossing voor de kweker. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 11.1 Doel

*Sprint 5 heeft als doel om de het contextuele deel van de chat te realiseren voor de kweker, door het inbouwen van vooraf gedefinieerde antwoorden.*

Dit sprint doel is opgesteld aan de hand van de visie van dit project. De visie was namelijk dat de chat meer moest kunnen dan enkel het chatten. Ook zou volgens deze visie de koper en kweker geholpen moeten worden met het stellen van vragen en geven van antwoorden. Om die reden is ervoor gekozen vooraf gedefinieerde vragen en antwoorden in te bouwen. Op deze manier kunnen kopers en kwekers snel en efficiënt een conversatie aangaan.

In deze sprint zal enkel gefocust worden op het antwoord geven vanuit de kweker.

### 11.2 User stories

#	Estimati...	Prioriteit	Name	User Story
13		1. Must have	Antwoord: Moment, we kijken ernaar	Als kweker, wil ik snel een koper kunnen antwoorden dat de vraag in behandeling nemen, zodat ik een veelvoorkomend antwoord snel kan geven en niet zelf hoeft op te stellen.
13		1. Must have	Antwoord: Helaas is er op dit moment geen aanbod	Als kweker, wil ik snel kunnen antwoorden dat er geen aanbod beschikbaar is, zodat ik een veelvoorkomend antwoord snel kan geven en niet zelf hoeft op te stellen.
13		2. Should have	Antwoord: Het is geregeld!	Als kweker, wil ik snel een koper kunnen antwoorden dat het geregeld is, zodat ik een veelvoorkomend antwoord snel kan geven en niet zelf hoeft op te stellen.
5		3. Could have	Vertaling van chat onderzoeken	Als opdrachtgever en als ontwikkelaar, wil ik de mogelijkheden rondom het vertalen in kaart hebben, zodat ik de juiste keuze kan maken voor het vertalen.

Figuur 20: Sprint 5, Features in Sprint backlog

Sprint 5 is opgedeeld in verschillende features. Deze features zijn opgesteld op basis van de verschillende voor gedefinieerde antwoorden die de kweker kan geven. Deze antwoorden zijn voortgekomen vanuit de behoeftes van de belanghebbenden, aan de hand van het visiedocument.

Aangezien de koper de chat kan benaderen bij het Direct inkopen scherm, zullen de meeste vragen gaan over het aanbod of een artikel. De antwoorden die de kweker kan geven op zulke vragen zullen de strekking hebben dat het geregeld is, dat de vraag in behandeling is en dat er helaas geen aanbod beschikbaar is.

Ook is er aan de hand van het visiedocument de behoefte naar voren gekomen om de vragen en antwoorden binnen een chat te vertalen. Om vast te kunnen stellen wat de mogelijkheden zijn voor het vertalen van berichten, is er binnen deze sprint hier een feature voor opgenomen.

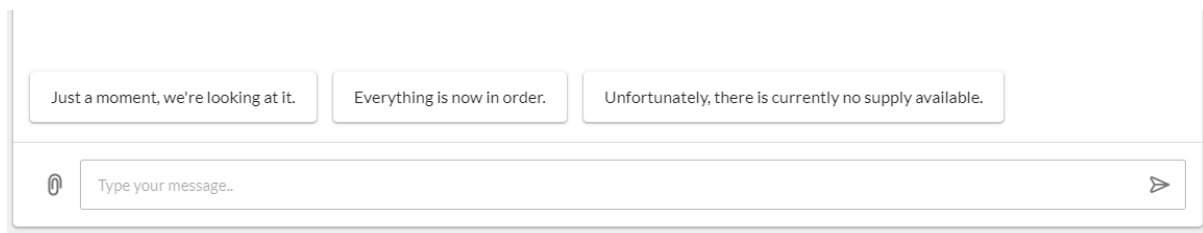
### Front-End uitbereiding

Om de features van deze sprint te realiseren is ervoor gekozen om de voor gedefinieerde vragen in de Front-End op te nemen. Dit is gedaan door een extra Component te creëren, genaamd ConversationAnswers die het renderen van de voorgestelde vragen afhandelt. Daarnaast is er een interface bijgemaakt die structuur geeft aan een vooraf gedefinieerd antwoord. Dit wordt weergegeven door middel van de diagrammen onder Bijlage 4: Ontwerp documentatie, Hoofdstuk 3.

### Werking voor gedefinieerde vragen

De koper start een conversatie met de kweker door een vraag te stellen. Wanneer de kweker de koper nog niet beantwoord heeft, worden er voorgestelde antwoorden getoond. Door op een van de voorgestelde antwoorden uit de selectie te klikken wordt direct het antwoord verstuurd. Voor de kweker bespaart dit een hoop tijd wanneer hij vaak dezelfde antwoorden geeft op vragen.

In het onderstaande Figuur 21: UI voorgestelde vragen zijn de voorgestelde vragen te zien voor de kweker. De gehele UI is te zien onder Bijlage 4: Ontwerp documentatie, Hoofdstuk 5.



*Figuur 21: UI voorgestelde vragen*

### Vertaling

Er is gekeken naar verschillende mogelijkheden om vertaling mogelijk te maken. Eerst is er gekeken naar de Google Vertalen API (Google, 2021). Hiermee zou je binnen een app direct gebruik kunnen maken van de vertaalfunctie van Google. Dit heeft als voordeel dat je gebruik kunt maken van veel verschillende talen en snel verschillende content binnen een app kunt vertalen. Wel brengt dit het nadeel met zich mee dat de Google Vertalen API geld kost. Dit is voorgelegd aan de opdrachtgever, maar door het nadeel van deze oplossing is hier niet voor gekozen.

Helaas zijn er geen andere betrouwbare vertaal oplossingen. Maar beschikt de browser zelf wel al over een automatische vertaalfunctie. Dit zou de gebruiker van de browser wel zelf aan moeten geven. Dit is minder betrouwbaar dan de Google API, maar een kosteloos prima alternatief.

Deze opties zijn voorgelegd aan de opdrachtgever. Door het nadeel van de Google API is er niet voor deze oplossing gekozen. Het vertalen via de browser is voor nu een prima oplossing aangezien de chat eerst beschikbaar wordt gesteld door Early access.

Binnen Floriday worden de talen Nederlands, Duits en Engels gebruikt. Floriday vertaalt de meeste statische content zelf. Dit wordt gedaan door gebruik te maken van de i18n package (NPMJS, 2021). Er is voor gekozen om de vooraf gedefinieerde vragen en antwoorden te vertalen door gebruik te maken van dezelfde package. Hierdoor krijgt de gebruiker van de chat de vooraf gedefinieerde



vragen en antwoorden in zijn of haar eigen taal. Wel brengt dit het nadeel mee dat de partij waarmee je praat antwoorden krijgt in een andere taal. Hier zou de automatische vertaling van de browser weer van pas komen.

### 11.3 Review

Sprint 5 had als doel om het contextuele deel van de chat te realiseren voor de kweker. Om dit sprint doel te behalen zijn er verschillende features aangemaakt die in deze sprint zijn opgepakt.

De features die in deze sprint zijn opgepakt:

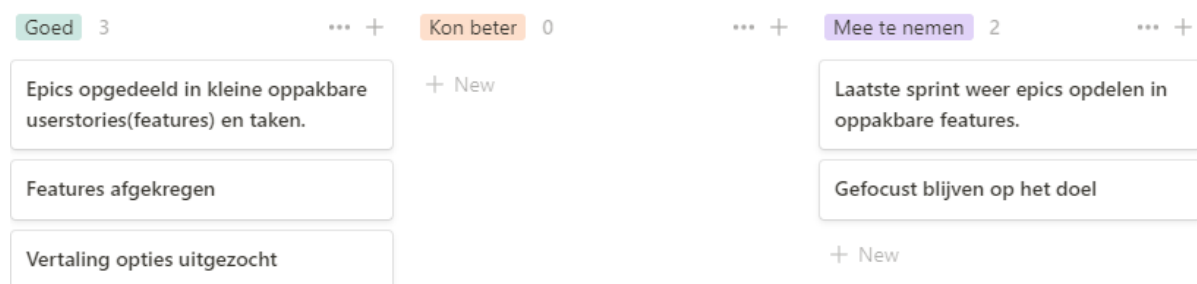
- Antwoord: Moment, we kijken ernaar
- Antwoord: Helaas is er op dit moment geen aanbod
- Antwoord: Het is geregeld
- Vertaling van chat onderzoeken

Het sprint doel van sprint 5 is behaald. Tijdens de review met de opdrachtgever is gesproken over het verwerken van de vooraf gedefinieerde antwoorden. Ook is er gesproken over dat de keuze voor de realisatie naar wens is. Met maar een klik kan er antwoord worden gegeven vanuit de kweker naar de koper.

De vooraf gedefinieerde antwoorden zijn momenteel enkel maar getoond wanneer de kweker nog geen bericht heeft verstuurd binnen de conversatie. Tijdens de review is naar voren gekomen dat het wenselijk is dat na het versturen van een bericht de vooraf gedefinieerde vragen alsnog geselecteerd moeten kunnen worden. Stel de kweker heeft geantwoord dat de vraag in behandeling is genomen, zou hij hierna het antwoord moeten kunnen geven dat het aanbod toch niet beschikbaar is. Op dit moment kan dit nog niet en zal daarom in de volgende sprint gekeken worden naar een oplossing voor dit probleem.

De opdrachtgever is tevreden met de oplossing die er nu betreft het vertalen voor berichten staat. Tijdens de early access kan nog gekeken worden hoe dit in de praktijk in zijn werk gaat. Met de opdrachtgever is gesproken over een mogelijke oplossing betreft het vertalen van berichten. De student is namelijk met een idee gekomen om door zowel de koper als kweker een voorkeurstaal in te kunnen laten stellen. Op deze manier zouden beide partijen van elkaar kunnen weten in welke taal zij het liefst willen communiceren. Deze oplossing gaat binnen dit project niet gerealiseerd worden, maar is zeker iets om op langer termijn mee te nemen.

### 11.4 Retrospective



Figuur 22: Sprint 5 Retrospective

### **Goed**

Net zoals sprint 4 heb ik ook deze sprint gekozen voor het opdelen van de epics in kleine, kleine User Stories/ features. Ik ben er heel blij mee dat ik inmiddels een goede werkwijze en structuur heb gevonden. Het voelt goed om de afgelopen sprints wel features af kan ronden, in plaats van epics die na de sprint nog open blijven staan. In deze sprints zijn dan ook alle features afgerond.

Daarnaast ben ik tevreden met het initiatief dat ik genomen heb voor de vertaal mogelijkheden. Hierdoor heb ik verschillende methodes in kaart kunnen brengen en de opdrachtgever kunnen adviseren over passende oplossingen.

### **Kon beter**

Voor deze sprint vond ik het lastig om iets op te schrijven wat beter kon gaan. In de afgelopen sprints heb ik veel procesverbeteringen doorgevoerd en deze sprint voelde daardoor erg goed.

### **Mee te nemen**

Om het succes van het project door te zetten is het van belang om de laatste sprint te werken met dezelfde werkwijze. Namelijk het opdelen van de epics in kleine features. Dit is dan ook iets wat ik in de laatste sprint mee zal nemen. Ik merk nog steeds aan mijzelf dat ik de neiging heb om te veel tegelijkertijd te doen. De focus die ik afgelopen sprint heb wil ik ook meenemen naar de volgende sprint, om op deze manier bij het doel te blijven en hiervoor te gaan.

## 12. Sprint 6: Contextuele chat voor koper

Voorafgaand aan deze sprint is er een doel opgesteld waar tijdens de sprint aan gewerkt wordt. Voorafgaande de sprint is er een Sprint backlog gemaakt, op basis van de Product backlog, met hierin user stories die binnen deze sprint opgepakt moeten worden om het doel te behalen. Er zal bij deze sprint ingegaan worden op het implementeren van de contextuele chat oplossing voor de koper. Tot slot wordt de sprint geëvalueerd door middel van een sprint review en retrospective.

### 12.1 Doel

*Sprint 6 heeft als doel om de het contextuele deel van de chat te realiseren voor de koper, door het inbouwen van vooraf gedefinieerde vragen.*

Sprint 6 is de laatste officiële sprint. In sprint 5, hoofdstuk 11. Sprint 5: Vooraf gedefinieerde antwoorden voor kweker zijn we begonnen met het contextuele deel van de kweker en in deze chat zal dit ook voor de koper gerealiseerd worden. Ook zal hierbij het openstaande punt uit de review van sprint 5 opgepakt worden, het altijd kunnen inzetten van de voor gedefinieerde vragen en antwoorden.

Aangezien dit de laatste officiële sprint is, zal er alles aan gedaan worden om de chat als een echte chat te laten voelen.

### 12.2 User stories

# Estimati...	🕒 Prioriteit	📄 Name	☰ Sprint	☰ User Story
13	1. Must have	💠 Vraag: Geen aanbod zichtbaar	Sprint 6	Als koper, wil ik snel kunnen aangeven dat er geen aanbod beschikbaar is, zodat ik een veelvoorkomende vraag snel kan stellen en niet zelf hoeft op te stellen.
13	2. Should have	💠 Vraag: Ik ben opzoek naar...	Sprint 6	Als koper, wil ik snel de vraag kunnen stellen dat ik op zoek ben naar een artikel, zodat ik een veelvoorkomende vraag snel kan stellen en dit meer context geeft bij mijn vraag.
13	2. Should have	💠 Vraag: Ik heb een vraag over een artikel	Sprint 6	Als koper, wil ik snel de vraag kunnen stellen dat ik een vraag heb over een artikel, zodat ik een veelvoorkomende vraag snel kan stellen en dit meer context geef bij mijn vraag.

Figuur 23: Sprint 6, Features in Sprint backlog

In de vorige sprint is het contextuele onderdeel van de chat voor de kweker gerealiseerd. Hier is ervoor gekozen om de kwekers de optie te geven om de meest voorkomende antwoorden op vragen van kopers direct te kunnen geven.

Deze sprint zal het contextuele onderdeel voor de koper gerealiseerd worden. Er is voor gekozen om dezelfde structuur aan te houden voor de vragen van kopers als de antwoorden van kwekers. Daarentegen heeft de koper wel andere antwoorden dan de vragen van de kweker. Uit de gesprekken met de opdrachtgever en op basis van het visiedocument zijn de meest voorkomende vragen van de kopers bij het direct inkopen doen: Ik zie geen aanbod, Ik ben op zoek naar een artikel

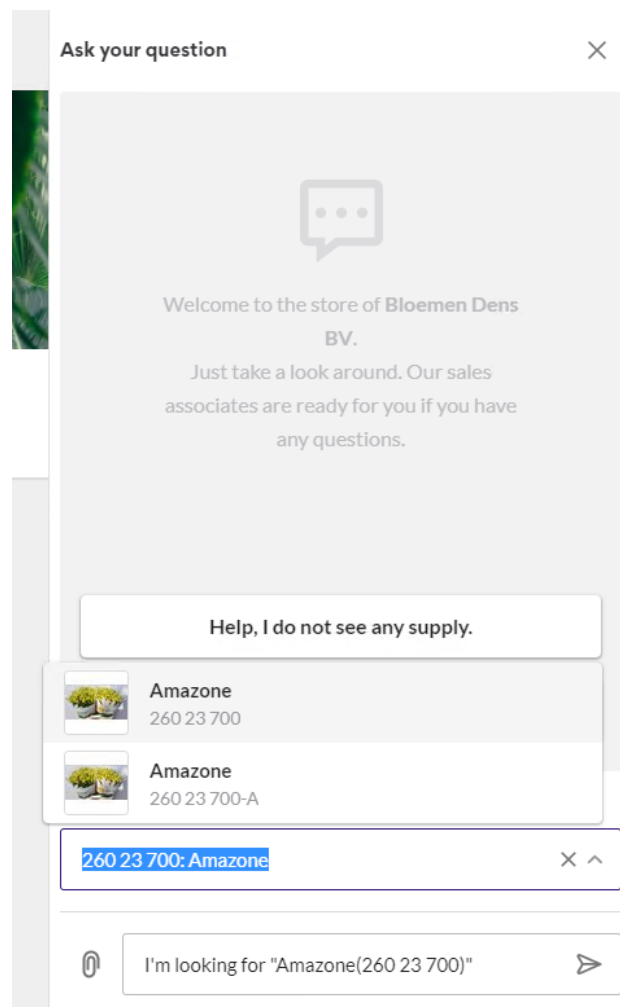
en Ik heb een vraag over een artikel. Deze voorgestelde vragen worden op dezelfde manier getoond als bij de voorgestelde antwoorden voor de kweker.

### Trade item meegeven aan bericht

Aangezien twee voorgestelde vragen over een artikel gaan is het selecteren van de vraag niet voldoende. Er moet op een manier een artikel geselecteerd en meegegeven kunnen worden bij een bericht.

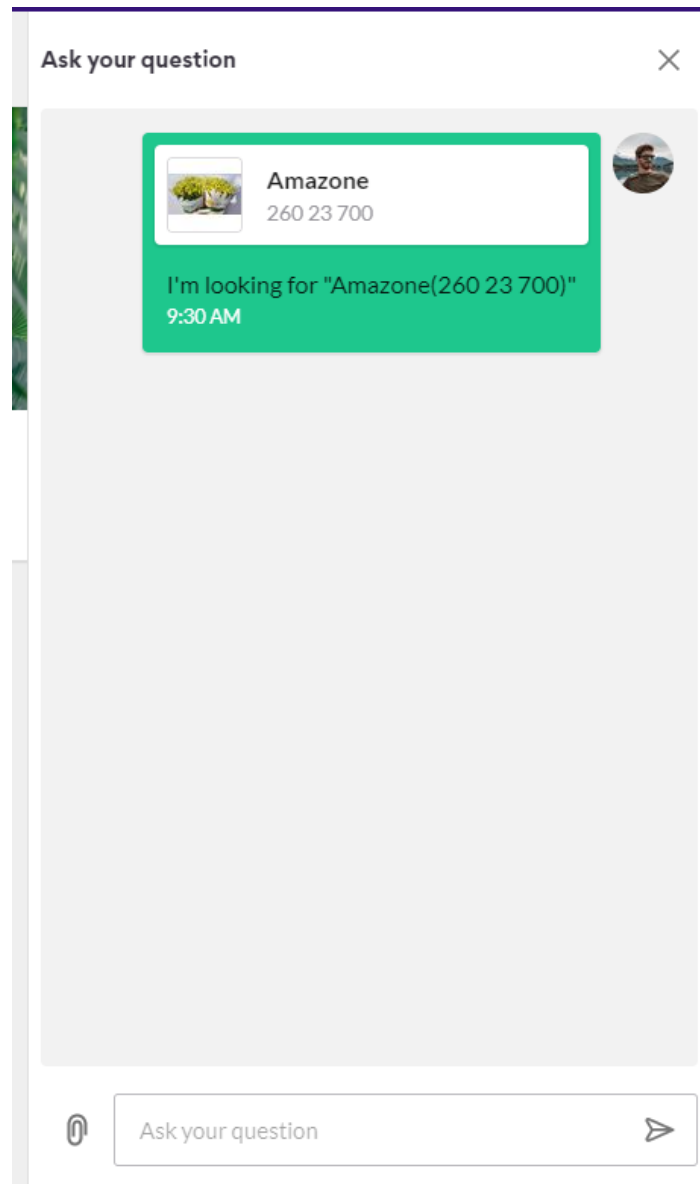
Daarom is ervoor gekozen om de `TradeltemId` op te nemen binnen de `ConversationMessage`. Dit zorgt ervoor dat een artikel toegevoegd kan worden aan een bericht. Dit is inzichtelijk gemaakt door middel van een UML diagram, die te vinden onder Bijlage 4: Ontwerp documentatie hoofdstuk 3.1. In deze diagram is te zien dat de `IConversationMessage` interface over het gehele `Tradeltem` object beschikt ten opzichte van alleen de `TradeltemId`.

Zoals benoemd in Hoofdstuk 5.3.1 Front-End maakt Floriday gebruik van een eigen React componenten set die in veel situatie ingezet kunnen worden. Door middel van een component uit deze set is het mogelijk gemaakt dat de koper naast het typen van een bericht, ook de optie kan hebben om een artikel te selecteren van een desbetreffende kwekersorganisatie op basis van artikelcode of naam. Zie het onderstaande figuur: Figuur 24: UI Selecteer artikel.



Figuur 24: UI Selecteer artikel

Bij het selecteren van een artikel wordt de artikelcode toegevoegd aan het bericht en kan hier ook nog een eigen bericht bij geschreven worden. Wanneer er een bericht verstuurd wordt met een artikel is ervoor gezorgd dat het voor zowel de koper als kweker duidelijk is om welk artikel het gaat.



*Figuur 25: UI Bericht met artikel*

Zoals te zien is in bovenstaand Figuur 25: UI Bericht met artikel wordt het artikel met foto en naam getoond in een box die meegegeven wordt aan het bericht. Wanneer hierop geklikt wordt zal het artikel in een dialog openen met alle specificaties.

#### Heropenen voor gedefinieerde vragen en antwoorden

Vanuit sprint 5 is het punt gekomen dat vooraf gedefinieerde vragen en antwoorden enkel maar getoond worden wanneer de koper of kweker nog geen bericht heeft verstuurd binnen de

conversatie. Voor de koper is het extra belangrijk dat hij tijdens een conversatie terug kan naar de voorgestelde vragen, omdat het toevoegen van een artikel iets is wat vaker voor kan komen binnen de conversatie.

Door naast het invoerveld binnen de UI een extra knop toe te voegen, is het mogelijk gemaakt om hiermee de vooraf gedefinieerde vragen opnieuw in beeld te brengen. Deze oplossing is voor zowel de customersportal als de suppliersportal gerealiseerd.

In de ontwerpdocumentatie (Bijlage 4: Ontwerp documentatie) staan alle definitieve ontwerpen die de structuur van het eindproduct in kaart brengt.

### 12.3 Review

Sprint 6 had als doel om het contextuele deel van de chat te realiseren voor de kweker. Om dit te behalen zijn er verschillende features opgepakt.

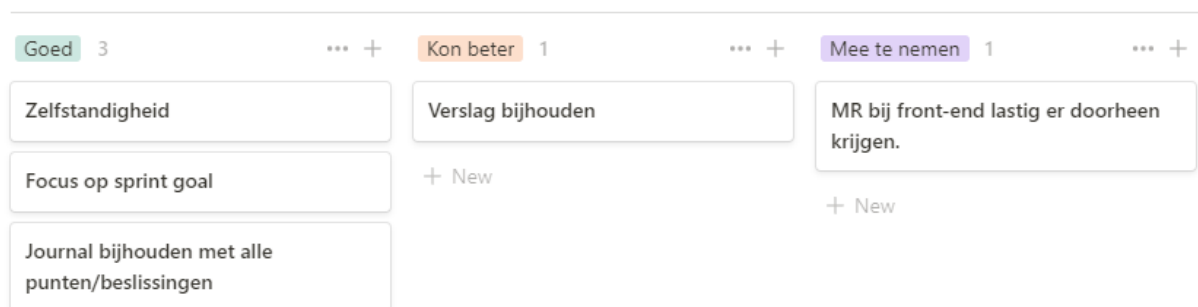
De features die in deze sprint zijn opgepakt:

- Vraag: Geen aanbod zichtbaar
- Vraag: Ik ben opzoek naar...
- Vraag: Ik heb een vraag over een artikel.

Het sprint doel van sprint 6 is behaald, de opdrachtgever is erg tevreden met het resultaat. Aan het einde van deze sprint is het gelukt om een voor zowel de koper als de kweker een contextuele chat te realiseren, met het gevoel van een real time chat.

Tijdens de review is er gesproken over het uitvoeren van gebruikerstesten met een koper en een kweker. Er is met een koper en een kweker een moment ingepland om deze testen uit te voeren. Door middel van deze testen willen we de gebruikers ervaring testen en de vastgestelde behoeftes valideren.

### 12.4 Retrospective



Figuur 26: Sprint 6 Retrospective

#### Goed

In deze laatste sprint is er de gehele tijd focus geweest op het sprint goal. Om deze te behalen heb ik de features en taken zorgvuldig opgesteld, opgesplitst en ingeschat. Daarnaast ben ik in de laatste sprint erg zelfstandig te werk gegaan en heb ik met een voldaan gevoel de laatste sprint afgesloten.

Door het gehele project heen heb ik een journal bijgehouden binnen Notion waarin ik alle beslismomenten, discussies en meetings bij heb gehouden. Dit helpt mij enorm met het afronden van mijn verslag.

### **Kon beter**

Ook in sprint 6 is weer gebleken dat ik moeite heb met het schrijven van mijn verslag. Dankzij het bijhouden van de journal heb ik voor het schrijven veel handvatten voor mijzelf gecreëerd.

### **Mee te nemen**

Ook deze sprint heb ik gemerkt dat het lastig was om de merge requesten er doorheen te krijgen van de Front-End door team blue. De kwaliteit van de Front-End binnen team blue ligt erg hoog. Aangezien dit project de eerste keer was dat ik React heb gebruikt was het voor de eerste sprints erg lastig om direct aan hun hoge kwaliteitseisen te voldoen. Nu is het met de laatste sprint gelukt om kwalitatief goede code te leveren voor de Front-End. Daarmee is het ook gelukt om het eindproduct live te kunnen zetten voor early access.

## 13. Gebruikerstesten

Na sprint 6 (hoofdstuk 12. Sprint 6: Contextuele chat voor koper) stond er een contextuele chat met real time chat ervaring voor zowel de koper als de kweker. Om de gebruikerservaring te testen en de vastgestelde behoeftes te valideren, is erop zoek gegaan naar een koper en kweker om een gebruikers test mee uit te voeren.

In sprint 4 (hoofdstuk 10. Sprint 4: Automatisch berichten ophalen) was het de bedoeling om een integratietest uit te voeren, zoals beschreven in dit hoofdstuk bleek dit niet mogelijk te zijn. Aangezien er voor dit project weinig integratie mogelijk bleek te zijn en de gebruikerservaring een belangrijk onderdeel van het succes bevat, is ervoor gekozen om gebruikerstesten uit te voeren.

De student heeft een minor “Usable App Development” gevolgd aan De Haagse Hogeschool. Tijdens deze minor heeft de student ook gebruikerstesten uit moeten voeren om de gebruikerservaring van de app te testen. Deze kennis is tijdens deze afstudeeropdracht toegepast bij het uitvoeren van de gebruikerstesten.

### 13.1 Uitvoering van de tests

Voorafgaande aan de tests is er voor zowel de koper als de kweker een gebruikerstest opgesteld die bestaat uit:

- Een procedure
- Het doel
- Pre-vragen
- Casus met taken
- Post-vragen

De gebruikerstest voor de koper is te vinden in “Bijlage 5: Koper gebruikerstest” en voor de kweker in “Bijlage 6: Kweker gebruikerstest”. Deze test is enkel door de student uitgevoerd met een koper en een kweker. Hierbij fungeerde de student als de interviewer en de observant. De tests zijn op de locaties van de koper en kweker uitgevoerd om zo in de realiteit van de gebruiker te blijven.

Tijdens het uitvoeren van de test is zowel het scherm als de audio opgenomen. Helaas was het niet mogelijk om hierbij ook de video op te nemen van de respondent. Op deze manier kon achteraf het testrapport opgesteld worden met de input van de respondent.

### 13.2 Testresultaten

#### **Kweker**

De kweker had weinig moeite met het uitvoeren van de taken uit de casus. Hierdoor blijkt (in ieder geval voor deze kweker) de gebruikerservaring succesvol te zijn.

Wel had de kweker moeite met het terug navigeren naar het conversatie overzicht nadat hij in de catalogus aanbod had aangemaakt. Dit kwam voornamelijk doordat de kweker aangaf weinig gebruik te maken van de ‘Mijn winkel’ omgeving, waar het conversatie overzicht terug te vinden is.

Door middel van de pre en post-vragen is er geconstateerd dat de kweker de contextuele chat een fijne feature vindt waar hij graag mee wil werken. Wel kan hij hier pas gebruik van gaan maken als ook zijn kopers naar deze digitale oplossing gaan. Ook is er door deze test gevalideerd dat de kweker de vooraf gedefinieerde antwoorden ook de meest voorkomende zijn.



Tijdens de test met de kweker liep hij tegen het scenario aan waar hij graag een artikel mee zou willen geven aan een bericht, net zoals de koper dit kan.

#### **Koper**

Ook de koper had weinig moeite met het uitvoeren van de gebruikerstest. Uit de test is gebleken dat de koper de manier van het versturen en ontvangen van berichten snel en simpel vond. Precies wat een chat zou moeten zijn.

Door het beantwoorden van de pre en post-vragen door de koper, is er gevalideerd dat ook dit de meest voorkomende vragen zijn voor een koper. De koper ervaarde het als prettig dat hij snel de meest voorkomende vragen kon stellen door deze te selecteren. En ook later in de chat deze vragen weer terug kon vinden. Het meesturen van een artikel aan een bericht werd ook positief ontvangen.

De koper gaf wel aan dat hij het fijn zou vinden om te kunnen zien of de kweker online is en zijn vragen direct kan beantwoorden.

### **13.3 Verwerking van de resultaten**

Aan de hand van de test resultaten van de test met de kweker, is ervoor gezorgd dat ook de kweker een artikel mee kan sturen met een bericht.

Ook is de opdrachtgever op de hoogte gebracht dat de kweker moeite had met het terugvinden van het conversatie overzicht. Er is aan de opdrachtgever aanbevolen om na dit project meerdere gebruikerstesten uit te voeren om te valideren of andere kwekers hetzelfde probleem ervaren.

Verder is er aan de hand van de test met de koper een aanbeveling gedaan voor een nieuwe functionaliteit binnen de chat. Namelijk het in kunnen zien of koper en/of kweker online is.

## 14. Aanbevelingen

Na het succesvol ontwikkelen van een contextuele chat binnen het Floriday platform, het uitvoeren van gebruikerstesten, het observeren van andere chat applicaties en gesprekken met collega's wil ik graag aanbevelingen doen, om de gebruiksvriendelijkheid en het gevoel van een real-time chat nog meer te verbeteren.

### **SignalR**

Als eerste wil ik graag aanbevelen om toch event-based technieken zoals SignalR te implementeren. Dit kan ervoor zorgen dat het versturen en tonen van berichten ook echt real-time werkt. Zoals te lezen is in "6.5 Real-time chat mogelijkheden" heb ik tijdens die sprint en gedurende het project naar de mogelijkheden gekeken rondom een real-time chat, maar waren deze afgeraden door Team Black.

Ik ben nog steeds van mening dat ik voor dit project de juiste keuze heb gemaakt, maar bij een volgend soort gelijk project, waar het digitale landschap minder restricties biedt, zou ik ervoor kiezen om direct te ontwikkelen met SignalR.

### **NoSQL- of Real-Time database**

Als tweede wil ik graag aanbevelen om een NoSQL- of real-time database te overwegen. Zoals te zien is in Bijlage 4: Ontwerp documentatie hoofdstuk 3.2, bestaan de klassen uit veel platte data. Dit komt onder andere door de keuze van de Microservice architectuur van Floriday, maar ook doordat er bij de Conversaties en Messages veel platte data opgeslagen moet worden. Dit is nu natuurlijk binnen de Match & Connect service ontwikkeld waar een SQL database aan hangt. Echter zouden de data en de functionaliteiten zich perfect lenen voor het gebruik van een NoSQL- of Real-time database.

### **Gebruikerstesten**

Als derde wil ik graag aanbevelen om meer gebruikerstesten uit te voeren met andere kopers en kwekers. Er is nu namelijk met 1 koper en 1 kweker getest. Om echt een goed beeld te kunnen schetsen van de gebruikerservaring raad ik aan om met meerdere kopers en kwekers soortgelijke gebruikerstesten uit te voeren. Dit zorgt ervoor dat er nog meer feedback ontvangen kan worden van verschillende gebruikers.

### **Interactieve chat**

Als vierde wil ik aanbevelen om de gebruiker aan te tonen wanneer de andere partij aan het typen is. Hierdoor blijft de conversatie interactief.

### **Notificaties**

Als vijfde wil ik aanbevelen om notificaties als feature toe te voegen. Dit gaat ervoor zorgen dat de gebruiker direct op te hoogte gebracht wordt van een nieuw bericht. Ook gaat dit ervoor zorgen dat er sneller gereageerd wordt op een bericht.

## 15. Evaluatie Proces

Aan het begin van mijn afstuderen ben ik begonnen met het opstellen van een plan van aanpak, een visiedocument en daarna met een Product backlog.

Aan het begin ging de Product backlog niet helemaal lekker. Maar na sprint 3 heb ik hier een andere werkwijze voor aangepakt. Ik heb toen namelijk de epics opgesplitst in kleine features, waar ik weer taken aan kon hangen. Daarnaast ben ik ook toen begonnen met inschattingen maken voor hoeveel werk elke feature en taak is. Door mijn afstuderen heen heb ik uiteindelijk een goede en fijne werkwijze gevonden, die voor mij erg goed werkte. Dit was ook zeker terug te zien in mijn productiviteit en het werk wat ik afkreeg.

Tijdens mijn afstuderen heb ik gebruik gemaakt van Scrum. Dit heb ik ingezet door iedere sprint een goal te geven, gebruik te maken van een ingeschatte en geprioriteerde backlog, sprint reviews te doen met de opdrachtgever en sprint retrospectives te doen met de bedrijfsbegeleider. Voor dit project werkte de Scrum werkwijze erg fijn voor mij. En naarmate de tijd werd ik ook steeds beter in het inzetten van Scrum voor mijn afstuderen.

Met het ontwikkelen heb ik ook een proces gevolgd. Per sprint heb ik namelijk eerst technisch ontwerp gemaakt, wanneer dat nodig was. Vervolgens heb ik de Back-End ontwikkeld volgens de Code First werkwijze en heb ik de Code Conventies vanuit JEM-id gevolgd. Bij het ontwikkelen heb ik voor iedere gemaakte endpoint unit tests geschreven/bijgewerkt. Daarna ontwikkelde ik de Front-End. Deze manier van werken gaf mij veel overzicht.

Ook ben ik blij met de keuze voor het vervangen van de integratietest door de gebruikerstesten. Voor het ontwikkelen van een chat is de gebruikerservaring enorm belangrijk. Als er niet prettig en logisch aanvoelt voor de gebruiker, zullen de gebruikers er nooit gebruik van maken. Daarentegen ben ik wel van mening dat de gebruikerstesten al eerder in dit project hadden moeten plaatsvinden. Bij een volgend project zou ik extra een gebruikerstest eerder in het traject houden, om zo eerder feedback te ontvangen en te verwerken in het project.

## 16. Evaluatie Product

Voordat ik begon aan het implementeren van mijn opdracht wilde ik de volgende producten opleveren:

1. Contextuele chat
2. Plan van aanpak
3. Visiedocument
4. Product Backlog
5. Ontwerp documentatie
6. Testplan
7. Testrapport

Alle producten zijn succesvol opgeleverd. Echter hebben het testplan en testrapport een andere vorm dan initieel bedacht was. Zoals al eerder te lezen is, heb ik ervoor gekozen om in plaats van een integratietest gebruikerstesten uit te voeren en op te leveren. Voor de gebruikerstesten zijn er gebruikersscenario's opgesteld en uitgevoerd met een koper en kweker, de resultaten hiervan zijn ook in deze documenten meegenomen. Deze documenten dienen als vervanging voor het testplan en testrapport.

De contextuele chat is naar verwachting succesvol opgeleverd. De Match & Connect service is uitgebreid met de logica en endpoints voor alle functionaliteiten rondom de conversaties en berichten. Daarnaast zijn de Front-End projecten van de Suppliersportal en de Customersportal uitgebreid met de contextuele chat.

De contextuele chat is een werkend product en is opengezet voor early access om verder te testen met meerdere gebruikers.

De opdrachtgever is enorm tevreden met het product, net zoals de gebruikers uit de gebruikerstest.

Door het plan van aanpak op te stellen heeft dit mij op voorhand geholpen over werkwijzen en aanpak hoe ik het project aan zou vliegen. Het visiedocument heeft mij gedurende de afstudeerperiode geholpen om de visie centraal te stellen.

Door middel van Notion heb ik alle Scrum activiteiten (zoals het goal, sprintplanning, retrospectives en reviews). Ook heb ik hierin de Product backlog op een gestructureerde wijze bij kunnen houden door gebruik te maken van geprioriteerde epics, features en taken. Op voorhand van iedere sprint heb ik aan de hand van de Product backlog items een sprint backlog gemaakt. Vanaf sprint 3 ben ik taken ingaan schatten waardoor ik met iedere sprint exact wist hoeveel werk ik aankon.

In mijn afstudeerplan bedacht ik enkel een ontwerpdocumentatie te maken met een UML-diagram van de Back-End. Het uiteindelijke ontwerpdocumentatie bevat veel meer dan alleen een UML-diagram van de Back-End. Namelijk ook: systeemarchitectuur, klassendiagrammen voor zowel de Front-End als Back-End, Use Case diagram voor gebruikersfuncties en visuele ontwerpen.

Vanuit school heb ik geleerd om de Back-End door middel van UML-diagrammen in kaart te brengen. Gezien de Front-End opgebouwd is als een React single page application en zelf veel logica, classes, interfaces en componenten bevat, heb ik die kennis gebruikt om deze structuur in kaart te brengen.

Door middel van gebruikerstesten heb ik de contextuele chat op gebruikerservaring kunnen testen. Deze bestaat uit een casus en gebruikers scenario's met taken die de gebruiker uit moet voeren. Dit heb ik geleerd bij de minor Usable App Development. Ik heb een gebruikerstest met een koper en een kweker uitgevoerd. Hier heb ik ook pre en post-vragen gesteld waarmee ik al vastgestelde

behoeftes uit het visiedocument kon valideren. De belangrijke informatie/punten hieruit heb ik verwerkt in het eindproduct en als aanbeveling gedaan.

Gedurende mijn afstuderen voelde ik mij steeds zekerder over het uiteindelijke eindproduct, de contextuele chat. Ik ben dan ook erg tevreden over de progressie die ik geboekt heb en de groei die ik naarmate de tijd heb doorgemaakt. Zowel de opdrachtgever als de gebruikers waren erg onder de indruk van het product. Het doet mij dan ook goed om te horen dat ze de contextuele chat graag in gebruik willen nemen. Ik kijk er dan ook naar uit om bij JEM-id aan de slag te gaan om met de teams onder andere aan dit product verder te werken.

## 17. Verantwoording beroepstaken

### **A1: Analyseren van probleemdomein & opstellen van probleemstelling:**

Het probleemdomein heb ik geanalyseerd door interviews te houden met de opdrachtgever en door in sprint 0 kritische beslissingen uit te zoeken. Aan de hand hiervan heb ik de probleemstelling opgesteld. Dit heb ik gedaan door een visiedocument op te stellen waarmee ik de Product backlog heb opgesteld.

Ondanks ik op afstand heb moeten werken en daardoor aan het begin geen direct contact met de gebruikers kon hebben, heb ik toch de juiste mensen weten te vinden. Hierbij heb ik voornamelijk met de teams (Blue voor opstellen en ontwikkelen van de Front-End, Purple voor de Back-End en Black voor architectuur en integratie) kunnen schakelen en de bedrijfsbegeleider en opdrachtgever. Hierbij is de opdrachtgever tevens ook Product Owner en lead designer van Floriday. Hij staat hierdoor in nauw contact met de gebruikers van het platform.

### **Gc: Kritisch, onderzoekend en methodisch werken:**

Gedurende de afstudeerperiode heb ik kritisch naar mijzelf en mijn werkwijze gekeken. Hierbij ben ik mijzelf ook vaak tegengekomen aan het begin. Ik werk namelijk beter met een goede structuur en aan het begin vond ik dit nog lastig te vinden. Hierbij heb ik mij onderzoekend opgesteld naar de methode die JEM-id gebruikt voor hun werkwijze. Gezien dit voor mij niet voldoende structuur zou bieden heb ik een eigen versie van Scrum gehanteerd.

De Product backlog bood mij veel structuur, deze heb ik opgesteld met epics, features en taken. Die ik door middel van de MoSCoW geprioriteerd heb. Ook heb ik vanaf sprint 3 hier allemaal een effort value aangegeven om zo kritisch naar mijzelf te kijken en in te schatten hoeveel werk ik aankon.

Tijdens het project ben ik onderzoekend te werk gegaan om constant te kijken naar betere oplossingen. Zo daagde ik mijzelf uit om altijd voor het beste te willen gaan. Hier heb ik onderzoek gedaan naar SignalR om het gevoel van een real-time chat te stimuleren. Ondanks hier niet voor gekozen is heb ik andere oplossingen aangedragen en nieuwe features gerealiseerd die de opdrachtgever eerder niet voor ogen had.

De code die ik opleverde heb ik altijd door 1 of meerdere collega's laten reviewen. Zij hebben hier kritisch naar gekeken en feedback gegeven. Hierdoor was het aan mij de taak om ook weer kritisch naar mijn eigen werk te kijken. En op deze manier dit ook mee te nemen in de rest van mijn afstudeertraject.

### **Gf: Leren leren:**

Tijdens de afstudeerperiode heb ik veel nieuws geleerd. Zo heb ik gebruik gemaakt van nieuwe programmeertalen, frameworks en manieren van werken. Ook heb ik mij moeten aanpassen aan de verschillende werkwijzen van de teams. Zo werkte bijvoorbeeld team Blue (Front-End) weer anders dan team Purple (Back-End).

Vanuit school heb ik handvatten meegekregen om de datastructuur van de Back-End in kaart te brengen. Deze manier heb ik toegepast op het in kaart brengen van de Front-End structuur. Aan deze diagrammen heb ik een eigen draai gegeven zodat ze in lijn zijn met de verwachtingen van een UML-diagram, maar wel passend zijn voor de Front-End structuur.

Door het afstudeerproces heb ik gezien dat met de ontwikkelmethode van JEM-id zij een gezamenlijk doel en structuur missen. Ik wil bekijken of de Scrum methode die tijdens dit project is gehanteerd beter aan kan sluiten bij JEM-id. Zo hoop ik ook hun in de toekomst iets te leren wat ik zelf uit deze periode geleerd heb.

### **A3: Informatie beoordelen, waarderen & prioriteren:**

Aan het begin van de afstudeerperiode heb ik interviews gedaan met de opdrachtgever. De bevindingen hieruit zijn meegenomen in een visiedocument. Ook heb ik een product backlog opgesteld die geprioriteerd is en voorgelegd is aan de opdrachtgever.

Na iedere sprint heb ik door middel van reviews en retro's meer informatie opgehaald bij de opdrachtgever. Deze informatie is vervolgens gewaardeerd en in sommige gevallen weer geprioriteerd binnen de volgende sprint.

Aan het einde van de sprints heb ik een gebruikerstest gedaan waar veel informatie uitgehaald is. Deze informatie is wederom beoordeeld, gewaardeerd en geprioriteerd. Zo heb ik bepaald of iets direct op moest worden gepakt, of als aanbeveling moest worden opgenomen.

### **C1: Ontwerpen van software:**

Aan het begin van de afstudeerperiode was het idee om enkel een UML-diagrammen te ontwerpen voor de Back-End realisatie. De ontwerp documentatie heb ik uitgebreider opgesteld door hier ook de Back-End, Front-End, gebruikersfuncties, architectuur en interfaces in kaart te brengen. Dit heeft mij zowel geholpen bij het ontwikkelen van het eindproduct, maar ook bij het schrijven van mijn afstudeerverslag.

### **D1: Realiseren van software:**

Door binnen de micro service architectuur van Floriday de Match & Connect service uit te breiden met de entiteiten en endpoints, om zo de Customersportal en Suppliersportal te voorzien van data,

Binnen de microservice architectuur van Floriday is de Match & Connect service uitgebreid met entiteiten en endpoints. Hiermee is de Customersportal en Suppliersportal voorzien van data. Deze portals zijn verder uitgebreid met alle schermen en functionaliteiten om de contextuele chat te realiseren voor zowel de koper als de kweker.

Voor de realisatie van de Back-End heb ik voornamelijk gewerkt met C# en .NET Core. Voor de Front-End realisatie heb ik gewerkt met React, Typescript en JavaScript.

### **D2: Testen & evalueren:**

Binnen het project heb ik uiteindelijk de integratietest uitgesloten. Hiervoor in de plaats heb ik een gebruikerstest gedaan met zowel de koper als de kweker, om zo de gebruikerservaring in kaart te brengen. Deze testen zijn opgebouwd volgens de werkwijze die ik tijdens mijn minor Usable App Development heb geleerd.

De gebruikerstest paste uiteindelijk beter bij deze opdracht dan een integratietest, aangezien er weinig integratie nodig was door de keuze van een bestaande service. Daarentegen paste de gebruikerstest perfect bij deze opdracht omdat het belangrijk is bij een chat dat de gebruikerservaring optimaal is en dit ook gevalideerd wordt.

Verder heb ik voor de endpoints unit testen geschreven om de functionaliteit en werking hiervan te testen.

## Literatuurlijst

- Agile Scrum Group. (2021). *Wat is Scrum*. Opgehaald van Agile Scrum Group: <https://agilescrumgroup.nl/wat-is-scrum-methode/>
- Entity Framework Tutorial. (2021). *What is Code-First?* Opgehaald van EntityFrameworkTutorial.net: <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>
- Facebook Inc. (2021). *Home pagina*. Opgehaald van React: <https://reactjs.org/>
- Google. (2021). *Cloud Translation*. Opgehaald van Cloud Google Web Site: <https://cloud.google.com/translate>
- Google. (2021). *Firebase Realtime Database*. Opgehaald van Firebase Web Site: <https://firebase.google.com/docs/database>
- JEM-id. (2021). *Home pagina*. Opgehaald van JEM-id Web Site: <https://www.jem-id.nl/nl>
- Likness, J. (2009). *SOLID and DRY principles*. Opgehaald van Code Project Web Site: <https://www.codeproject.com/Articles/36712/SOLID-and-DRY>
- Material-UI. (2021). *Home pagina*. Opgehaald van Material-UI: <https://material-ui.com/>
- Microsoft. (2019). *Microservicearchitectuur*. Opgehaald van Microsoft Docs Web Site: <https://docs.microsoft.com/nl-nl/azure/architecture/guide/architecture-styles/microservices>
- Microsoft. (2021). *.NET documentation*. Opgehaald van Microsoft Docs Web Site: <https://docs.microsoft.com/en-us/dotnet/>
- Microsoft. (2021). *C# documentatie*. Opgehaald van Microsoft Docs Web Site: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- Microsoft. (2021). *Entity Framework Core*. Opgehaald van Microsoft Docs Web Site: <https://docs.microsoft.com/en-us/ef/core/>
- Microsoft. (2021). *Home pagina*. Opgehaald van TypeScript: <https://www.typescriptlang.org/>
- Microsoft. (2021). *Real-time ASP.NET with SignalR*. Opgehaald van Microsoft Docs Web Site: <https://dotnet.microsoft.com/apps/aspnet/signalr>
- Moment.js. (2021). *Home pagina*. Opgehaald van Momentjs Web Site: <https://momentjs.com/>
- Muldoon, N. (2013). *Why do high performing Scrum teams use story point estimation?* Opgehaald van velocitycounts: <http://www.velocitycounts.com/2013/05/why-do-high-performing-scrum-teams-tend-to-use-story-point-estimation/>
- Notion Labs. (2021). *Home pagina*. Opgehaald van Notion Web Site: <https://www.notion.so/>
- NPMJS. (2021). *i18n*. Opgehaald van NPMJS Web Site: <https://www.npmjs.com/package/i18n>
- Npmjs. (2021). *Okta React SDK*. Opgehaald van NPMJS: <https://www.npmjs.com/package/@okta/okta-react>
- projectmanagementsite. (2021). *MoSCoW*. Opgehaald van Projectmanagement Web Site: <https://projectmanagementsite.nl/moscow/#.YVwUyZpBwuU>



RFH. (2021). *Home pagina*. Opgehaald van Floriday Web Site: <https://www.floriday.io/>

Royal Flora Holland. (2021). *Over ons*. Opgehaald van Royalfloraholland Web Site:  
<https://www.royalfloraholland.com/over-ons>

Webpack. (2021). *Home pagina*. Opgehaald van Webpack: <https://webpack.js.org/>

## Bijlagen

Alle bijlagen zijn opgeleverd in een extern bijlagen document. Dit hoofdstuk verwijst enkel naar de bijlagen die in het externe document zijn toegevoegd.

Bijlage 1: Product backlog

Bijlage 2: Plan van Aanpak

Bijlage 3: Visiedocument

Bijlage 4: Ontwerp documentatie

Bijlage 5: Koper gebruikerstest

Bijlage 6: Kweker gebruikerstest

Bijlage 7: JEM-id Code conventies

Bijlage 8: Evaluatie bedrijfsmentor