

# Afstudeerverslag

Jorik Diks

Organisatie: Carthago ICT  
Opdrachtgever: Dennis Wolters  
Contactpersoon: Jordy Nieuwenhuis  
Mail: [jordy.nieuwenhuis@carthago-ict.nl](mailto:jordy.nieuwenhuis@carthago-ict.nl)

Onderwijsinstelling: Saxion  
Begeleider: Etto Salomons  
Mail: [e.l.salomons@saxion.nl](mailto:e.l.salomons@saxion.nl)

# Informatiepagina

## *Student*

Naam: Jorik Diks  
Bison nummer: 464979  
Email: 464979@student.saxion.nl

## *Opleiding*

Naam: HBO-ICT (B)  
Richting: Software engineering  
School: Hogeschool saxion

## *Bedrijfsbegeleider*

Naam: Jordy Nieuwenhuis  
Email: jordy.nieuwenhuis@carthago-ict.nl

## *Afstudeerbegeleider*

Naam: Etto Salomons  
Email: e.l.salomons@saxion.nl

## **Inleverdatum**

12-06-2022

## Voorwoord

Dit is het afstudeerverslag voor het afstudeerproject van Jorik Diks, dit project loopt van februari 2022 tot juli 2022. Tijdens dit project wordt een onderzoek uitgevoerd en aan een applicatie gewerkt om te aan te tonen dat de 3 beroepsactiviteiten die in de afbeelding hieronder zijn aangekruist op niveau 3 beheerst worden.

		Beroepsactiviteiten				
		manage/control	analyseren	adviseren	ontwerpen	realiseren
Architectuurlagen	gebruikersinteractie				x	
	organisatieprocessen					
	infrastructuur					
	software				x	x
	hardware interfacing					

Figuur 1: Het HBO-i 25-vlaksmodel

De aangekruiste beroepsactiviteiten zijn op het softwareniveau: analyseren, ontwerpen en realiseren. De exacte werkzaamheden die uitgevoerd moeten worden om te bewijzen dat de beroepsactiviteiten beheerst worden op niveau 3 staan beschreven in de HBO-i domeinbeschrijving.

Om te bewijzen dat deze beroepsactiviteiten op het vereiste niveau zijn uitgevoerd staat in het afstudeerverslag beschreven welke onderzoeken ik heb uitgevoerd. Bij ieder onderzoek staat beschreven welke activiteiten zijn uitgevoerd om de onderzoeksvraag te kunnen beantwoorden.

## Samenvatting

Voor mijn afstudeer opdracht werk ik aan Pedalpact.cc. Pedalpact.cc is een zakelijke wiel community die in samenwerking met Carthago is gebouwd. Het doel van de app is om op een leuke en actieve manier afspraken te maken met interessante zakelijke contacten, door middel van pacts. Pacts zijn fietsritten die georganiseerd worden door de gebruiker of door Pedalpact.cc zelf.

Op dit moment is het niet mogelijk om voor Pedalpact.cc te betalen, dit is echter wel nodig om de applicatie te kunnen blijven onderhouden en uit te breiden met nieuwe functionaliteiten. Aan mij is de taak om een freemium model te implementeren. Een freemium model houdt in dat de gebruiker de service gratis of betaald kan gebruiken, betaalde gebruikers hebben meer functionaliteiten dan gratis gebruikers. Het moet ook makkelijker worden voor bezoekers om Pedalpact.cc te kunnen gebruiken, een bezoeker moet niet een account aan hoeven maken om de kaart te zien.

Om het makkelijker te maken voor bezoekers om de website te gebruiken is er een openbare kaart- en pact pagina geïmplementeerd. Deze pagina's zijn te bezoeken zonder dat de gebruiker een account hoeft aan te maken. De gebruiker kan hier alleen kijken naar pacts en gebruikers en kan geen profielen bekijken of meedoen aan pacts. Tijdens het onderzoek heb ik de focus gelegd op de anonimiteit van de gebruiker en te voorkomen dat bezoekers ongevraagd mee kunnen doen aan een pact. Tevens heb ik een gebruikerstest uitgevoerd om de doelstellingen van de publieke pagina's te evalueren vanuit het gebruikersperspectief. Tijdens de testen kwam naar boven dat pacts niet genoeg opvallen op de kaart.

Voor de volgende onderzoeksvraag moeten de pacts meer opvallen op de kaart. Ik heb een onderzoek gedaan waar ik de thinking aloud methode gebruikte voor ideeën om pacts meer op te laten vallen. De beste ideeën heb ik uitgewerkt in designs en per design heb ik gekeken of pacts echt meer opvielen. Uiteindelijk heb ik een combinatie van meerdere designs gebruikt om tot de uiteindelijke oplossing te komen. Dit design heb ik geïmplementeerd in de website.

Toen ik hier mee klaar was viel me op dat de product owner veel tijd kwijt was aan de branches testen om de applicatie te testen voor bugs voordat deze gemerged wordt in master. Mij leek het daarom een goed idee om automatische testen te schrijven, hiermee kan de applicatie automatisch getest worden om zeker te zijn dat er geen functionaliteiten zijn gebroken. Ik ging eerst onderzoeken welke type testen er waren voor de front- en backend. Toen ik hiermee klaar was ging ik per type test kiezen welke test-framework ik ging gebruiken, hiervoor gebruikte ik een rubriek. Met de gekozen test-frameworks ging ik daarna testen schrijven voor de front- en backend.

Nu de bugs zijn opgelost was het tijd om het freemium te implementeren. Ik heb eerst onderzocht welke functionaliteiten worden afgeschermd voor de gratis gebruikers. Ik heb daarna het model geïmplementeerd in de applicatie. Het freemium model is nu geïmplementeerd maar de gebruiker is nog niet op de hoogte van dit abonnement. Om de gebruiker informatie hierover te geven is er een premium informatie pagina geïmplementeerd, hier staan alle voordelen die een premium gebruiker krijgt.

Als laatste onderdeel is de onboarding flow aangepast en een product tour toegevoegd aan de website. Deze aanpassingen zijn gemaakt om de toegankelijkheid van de website te vergroten. De oude onboarding flow was overweldigend dus wordt de gebruiker nu alleen voor de essentiële onderdelen gevraagd. De product tour leidt de gebruiker door de onderdelen van pagina's die als verwarrend kunnen worden ervaren.

## Inhoudsopgave

Informatiepagina .....	1
Voorwoord .....	2
Samenvatting.....	3
1 Inleiding .....	5
1.1 Bedrijfsomschrijving .....	5
1.2 Verslagopbouw .....	5
1.3 Probleem .....	6
1.4 Opdracht.....	6
1.5 Onderzoek .....	7
1.6 Context .....	8
1.6.1 Frontend .....	8
1.6.2 Backend .....	8
1.6.3 Authenticatie .....	8
1.7 User Stories .....	9
2 Projectaanpak.....	10
3 Openbare kaart- en pact pagina.....	11
3.1 Resultaat.....	12
3.2 Implementatie .....	16
3.3 Uiteindelijk resultaat .....	19
3.4 Testen .....	20
3.5 Social sharing image .....	22
3.6 Reflectie.....	24
4 Pact markers meer op laten vallen.....	25
4.2 Definitieve oplossing .....	26
4.3 Implementatie .....	27
4.4 Testen .....	28
4.5 Reflectie.....	29
5 Automatisch testen .....	30
5.1 Reflectie.....	33
6. Premium gebruiker.....	34
6.1 Implementatie .....	35
6.2 Welke nieuwe functionaliteiten hebben betaalde gebruikers in de applicatie? .....	36
6.3 Welke andere bronnen van inkomsten kunnen er komen?.....	37
6.4 Reflectie.....	38
7. Nieuwe onboarding flows voor verschillende type gebruikers en een product tour .....	39

7.1 Implementatie .....	40
7.2 Uiteindelijke resultaat .....	41
7.3 Testen .....	43
7.4 Reflectie .....	45
8. Conclusie .....	46
9. Aanbevelingen .....	47
10. Reflectie .....	48
Bronnen .....	49

## 1 Inleiding

### 1.1 Bedrijfsomschrijving

Carthago ICT B.V is een is een bedrijf dat zich voornamelijk bezighoudt met softwareontwikkeling, door middel van het ontwikkelen van maatwerkenoplossingen, maar ook onderhouden van bestaande softwareoplossingen. Voor de softwareoplossingen wordt er gebruik gemaakt van de nieuwste technologieën zodat ze een stap voor de competitie lopen. Carthago heeft twee locaties, één in Hengelo en één in Veenendaal. Er werken ongeveer 60 werknemers over de locaties verspreid.

Carthago investeert ook graag in innovatieve projecten. Dit is hoe de samenwerking met Pedalpact.cc is ontstaan. De eigenaar van Carthago houdt van fietsen en is in gesprek geraakt met Daan Walet, de bedenker en eigenaar van Pedalpact.cc. Daan is een freelance marketeer en ging daarom naar veel netwerk evenementen om aan opdrachten te komen en om nieuwe contacten op te doen. Het probleem van deze evenementen was volgens Daan dat deze eigenlijk gewoon saai zijn.

Met deze gedachte bedacht hij Pedalpact.cc, dit is een fietsers community platform waar fietsers in heel Nederland pacts kunnen organiseren. Pacts zijn fietsritten waar deelnemers de mogelijkheid hebben om te netwerken terwijl ze hun hobby uitoefenen: fietsen. Inmiddels heeft het platform 1000 gebruikers. Het platform heeft alleen nog geen manier om inkomsten te generen.

Pedalpact.cc is vorig jaar in september uitgekomen met alleen de mogelijkheid om een gratis account aan te maken. Dit jaar is erop of eronder voor Pedalpact.cc. Zonder inkomsten wordt de investering stopgezet vanuit Carthago.

### 1.2 Verslagopbouw

De structuur van dit verslag is chronologisch opgebouwd. In elk hoofdstuk wordt een deelvraag uitgelicht, en worden de bijhorende sub vragen beantwoord. Nadat de deelvraag is beantwoord volgt het resultaat van het onderzoek met eventuele opmerkingen, de implementatie in de applicatie en het uiteindelijke resultaat van de website. Het laatste onderdeel is een reflectie van alle punten die goed gingen, de problemen waar ik tegen aan liep en wat ik hiervan heb geleerd.

### 1.3 Probleem

Pedalpact.cc is een zakelijke wieler community die in samenwerking met Carthago is gebouwd. Het doel van de app is om op een leuke en actieve manier afspraken te maken met interessante zakelijke contacten, door middel van fietsen.

Op dit moment is er geen mogelijkheid om voor deze service te betalen, dit is echter wel nodig om de applicatie te kunnen blijven onderhouden en uit te breiden met nieuwe functionaliteiten.

Daarom is ervoor gekozen om een freemium model toe te passen. Hiermee kan de gebruiker kiezen om te betalen voor de service of deze gratis te blijven gebruiken. Het moet aantrekkelijk zijn om een abonnement af te sluiten. Om een abonnement aantrekkelijk te maken moeten extra functies beschikbaar zijn voor betalende gebruikers die gratis gebruikers niet hebben. Om het freemium model bekend te maken aan de gebruiker is er ook een premium informatie pagina nodig.

Ook moet het platform meer gebruikers zien te bereiken. Dit zal worden gerealiseerd door de pact detail pagina, dit is de pagina waar alle informatie over het pact oftewel de fietsrit wordt beschreven, en de kaart ook zichtbaar te maken als bezoeker. Hierbij is het belangrijk om gevoelige informatie niet te tonen voor deze gebruikers. Het doel van deze openbare pagina's is om de bezoekers te overtuigen om ook een account aan te maken. Om het proces van een account aan te maken zo soepel mogelijk te laten verlopen moet er ook een vernieuwde onboarding flow komen en een product tour om de website uit te leggen aan nieuwe gebruikers.

Door de hierboven beschreven functie toe te voegen zal het verkeer op de website toenemen. Het wordt dan nog lastiger om fietsritten te vinden op de kaart tussen alle profiel markers. Daarom zullen er ook verbeteringen moeten worden gemaakt aan de interface om bepaalde onderdelen zichtbaarder te maken.

### 1.4 Opdracht

Voor Pedalpact.cc dienen meerdere opdrachten uitgevoerd en ingediend te worden. Dit betreffen allemaal individuele opdrachten waar voor sommige gedeeltes eerst groen licht gegeven moet worden door de product owner, Daan Walet. Mijn focus voor het project ligt op het freemium model te implementeren. De onderdelen die hieronder vallen zijn:

- Voeg een openbare pact en kaart pagina toe voor bezoekers
- Verbeter de zichtbaarheid van pacts op de kaart
- Implementeer testen om de functionaliteiten te waarborgen
- Implementeer functionaliteit voor premium gebruikers
- Implementeer een product tour en een vernieuwde onboarding flow voor de nieuwe gebruiker types.

## 1.5 Onderzoek

Per onderzoeksvraag wordt er eerst een onderzoek uitgevoerd en hierna wordt dit geïmplementeerd in de ontwikkelingsfase.

**Hoofdvraag:** hoe kan Pedalpact.cc een freemium model toevoegen aan de applicatie?

Deze hoofdvraag is gekozen omdat alle onderzoek onderdelen uiteindelijk directe impact hebben of worden beïnvloed door het nieuwe freemium model.

Het onderzoek dat uitgevoerd zal worden is opgedeeld in vijf deelvragen. Deze deelvragen zijn allemaal van toepassing op de hoofdvraag van dit onderzoek. De resultaten en implementatie die de deelvragen opleveren zorgen voor de uiteindelijke implementatie van het freemium model.

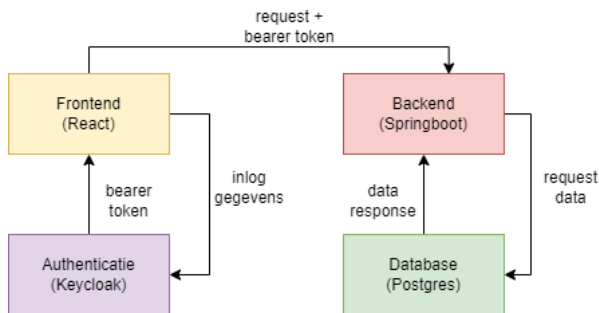
De vijf deelvragen zijn de volgende:

1. Welke gebruikers en pact informatie mag er worden getoond op de openbare pact en kaart pagina?
2. Hoe kan de kaart interface worden aangepast om het zoeken naar fietsritten makkelijker te maken en deze ook meer op te laten vallen?
3. Hoe kan de frontend en backend automatisch worden getest om bestaande en nieuwe functionaliteit te waarborgen?
4. Hoe kunnen de nieuwe gebruikersvormen worden ondersteund met de huidige opzet van de applicatie?
5. Hoe moet de vernieuwde onboarding flow worden van een gratis gebruiker en premium gebruiker en hoe kan er een product tour worden geïmplementeerd?



## 1.6 Context

Pedalpact.cc is een bestaande applicatie, de taal en libraries zijn dus al gekozen. De structuur van de beginsituatie van de applicatie ziet er als volgt uit:



Figuur 2: Structuur beginsituatie applicatie

### 1.6.1 Frontend

#### Taal

Voor het project is gekozen voor Typescript in combinatie met React en SASS. Als ik het project zelf op zou zetten had ik voor React Native of Ionic gekozen. De applicatie is namelijk bedoeld om vooral op de fiets gebruikt te worden, hiervoor wordt altijd een smartphone gebruikt. Bij een native app konden bepaalde functionaliteiten zoals notificaties veel makkelijker en netter worden geïmplementeerd. Het was ook makkelijker geweest om het mobiele ontwerp te maken omdat er gebruik gemaakt kon worden van native componenten.

#### Code styling

Om de code styling te waarborgen wordt er gebruik gemaakt van ESLint 6. Er wordt gebruik gemaakt van de instellingen van Carthago om de code styling hetzelfde te houden als de rest van de projecten. ESLint wordt alleen lokaal gedraaid.

Voor iedere nieuwe React component of pagina moet een nieuwe folder worden aangemaakt. Ieder component of pagina heeft zijn eigen styling pagina samen met een view en business logic class. In de business logic class wordt alle logica uitgewerkt en afgehandeld. De resultaten worden doorgestuurd naar de view class door middel van props.

#### Libraries

Om calls naar de API te maken wordt er gebruik gemaakt van Axios, dit is een node module die het communiceren naar een API makkelijker maakt. De responses worden hierna opgevangen door de al vastgelegde types in de frontend.

### 1.6.2 Backend

#### Taal

Voor het project is gekozen voor Java in combinatie met Springboot. Voor de database is gebruik gemaakt van Postgres. Tijdens het lokaal ontwikkelen wordt er gebruik gemaakt van de “staging database”. Dit is een database die alleen in de testomgeving wordt gebruikt.

### 1.6.3 Authenticatie

Voor de authenticatie in Pedalpact.cc wordt Keycloak gebruikt, dit is een aparte server die de inloggegevens en gebruikers metadata op slaat. Keycloak geeft aan Axios de Bearer token door na authenticatie.

## 1.7 User Stories

Binnen de MoSCoW methode zijn er 4 verschillende prioriteit niveaus hieronder weergegeven van hoge naar lage prioriteit (MoSCoW, 2018).

- Must
- Should
- Could
- Won't/Would/Wish

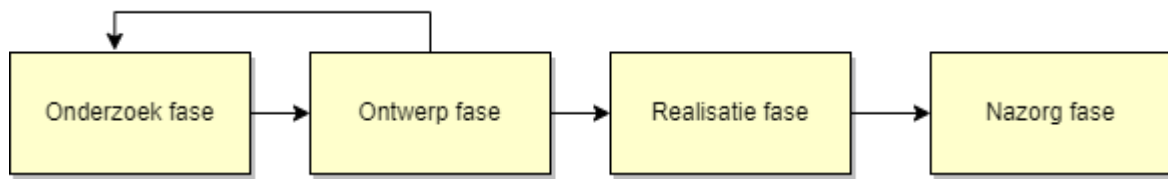
Deze priorisering is tijdens het ontwikkelen van de applicatie toegepast om te bepalen welke requirements wanneer geïmplementeerd worden.

De user stories zijn verzameld na alle deelvragen op te hebben gesteld, per deelvraag zijn dan de user stories opgesteld die nodig zijn om de deelvraag op te kunnen lossen.

De user stories zijn terug te lezen in de meegeleverde bijlage (zie bijlage; UserStories.pdf).

## 2 Projectaanpak

Iedere onderzoeksvraag zal op ongeveer dezelfde manier worden uitgevoerd, zie figuur 3 hieronder.

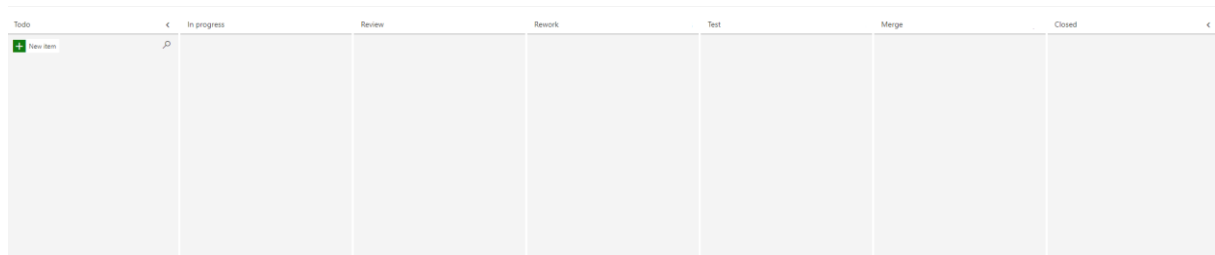


Figuur 3: Onderzoek fase

Per onderzoeksvraag zal eerst onderzoek gedaan worden, dit kan een interview zijn, researchen etc. Hierna neemt de ontwerpfase plaats, bij een functionele onderzoeksvraag begint dit met een hi-fi ontwerp ontworpen in Adobe XD. Bij een technische onderzoeksvraag kan dit een prototype zijn of een technisch ontwerp. Het resultaat van dit ontwerp wordt besproken met de product owner en feedback wordt uitgewerkt. Dan neemt de realisatiefase plaats: hier wordt het product ontwikkeld en wanneer gereed gemerged. Tijdens het mergen worden er soms nog bugs gevonden, dit wordt afgehandeld in de nazorgfase.

Om een overzicht te hebben wat er nog gedaan moet worden om de huidige onderzoeksvraag af te ronden maak ik gebruik van Kanban. Kanban is een methode om kenniswerk-diensten (processen) te definiëren, managen, en verbeteren. De kracht van Kanban zit 'm vooral in het onzichtbare (kenniswerk) zichtbaar maken, en in de kunst om niet alles tegelijk op te pakken. (vanderwardt@agilecrumgroup.nl, 2020). Deze methoden heb ik gekozen omdat ik wel een planning heb (zie bijlage; Planning.pdf) en een backlog maar niet met sprints werk.

De backlog wordt bijgehouden in Azure DevOps, hier heb ik een kolom van TODO, In progress, Review, Rework, Test, Merge en Closed. Voordat een issue gemerged mag worden moet die eerst worden goedgekeurd door de opdrachtgever, Dennis. Als de issue is goedgekeurd wordt deze naar Test gesleept, dan kijkt de product owner, Daan Walet, er naar. De product owner controleert of de oplossing voldoet aan de visuele eisen die hij stelt. Uiteindelijk wordt de issue naar Merge verplaatst en mag die gemerged worden in de master branch. Na het mergen wordt de issue in Closed gezet.



Figuur 4: Azure backlog opzet

Om de voortgang te bespreken vindt er iedere week op de dinsdag een weekly plaats. Hier is de product owner, opdrachtgever, ik en de backend developer aanwezig.

Er vindt ook iedere dag om 12 uur een daily plaats met iedereen uit hetzelfde "team". Ik zit in team Blauw, dit is iedereen die in Veenendaal werkt aan een project vanuit Carthago. De reden dat ik in het team van Veenendaal zit is omdat het project in Veenendaal is opgezet. Hier bespreek je alles wat je hebt gedaan die dag en wat je nog gaat doen.

### 3 Openbare kaart- en pact pagina

Pedalpact.cc heeft nu alleen een kaart die zichtbaar is voor gebruikers. Dit zorgt ervoor dat bezoekers die eerst een beeld van de applicatie willen krijgen voordat ze een account aanmaken dit niet kunnen doen. Ditzelfde geldt voor de pact pagina, als je nu een pact deelt op bijvoorbeeld Facebook kunnen je vrienden niet de pact informatie bekijken voordat ze een account hebben aangemaakt.

De oplossing hiervoor is een openbare kaart- en pact pagina. Dit is een uitgeklede versie van de pagina die een gebruiker ziet, alleen is de privacygevoelige informatie weggelaten. Op de openbare pact pagina moet ook rekening gehouden worden dat een bezoeker niet alle informatie voor een pact ziet waarmee die ongevraagd mee kan doen met een pact.

Voor de openbare pagina's moet ik eerst alle privacygevoelige informatie vinden en beslissen of dit weggelaten kan worden of dat de informatie met aanpassingen getoond kan worden.

Daarnaast moet er ook een social sharing image komen om het delen op social media te verbeteren. Als er nu een link wordt gedeeld van een pact is het niet duidelijk waar dit voor dient en ook niet aantrekkelijk om op te klikken.

Voor mijn onderzoek moest ik eerst uitzoeken uit welke onderdelen de pact en kaart pagina bestaan, om dit te doen maakte ik een screenshot van beide pagina's en ging ik ieder onderdeel die ik kon vinden omcirkelen. Ik kwam er al snel achter dat ieder onderdeel deel kan zijn van drie verschillende onderdeel groepen: persoonsgebonden onderdelen, gebruikersinformatie en pactinformatie.

Persoonsgebonden onderdelen zijn onderdelen die een actie uitvoeren voor de gebruiker. Een voorbeeld voor een persoonsgebonden onderdeel is het aanmelden voor een pact, de gebruiker wordt dan aangemeld voor het pact.

Gebruikersinformatie zijn onderdelen die iets zeggen over andere gebruikers op de website. Een voorbeeld van gebruikersinformatie is de naam van een gebruiker of de profielfoto. Gebruikersinformatie is gevoelig en moet worden afgeschermd voor bezoekers, niet iedere gebruiker wil zijn profielfoto zichtbaar hebben op een openbare pagina.

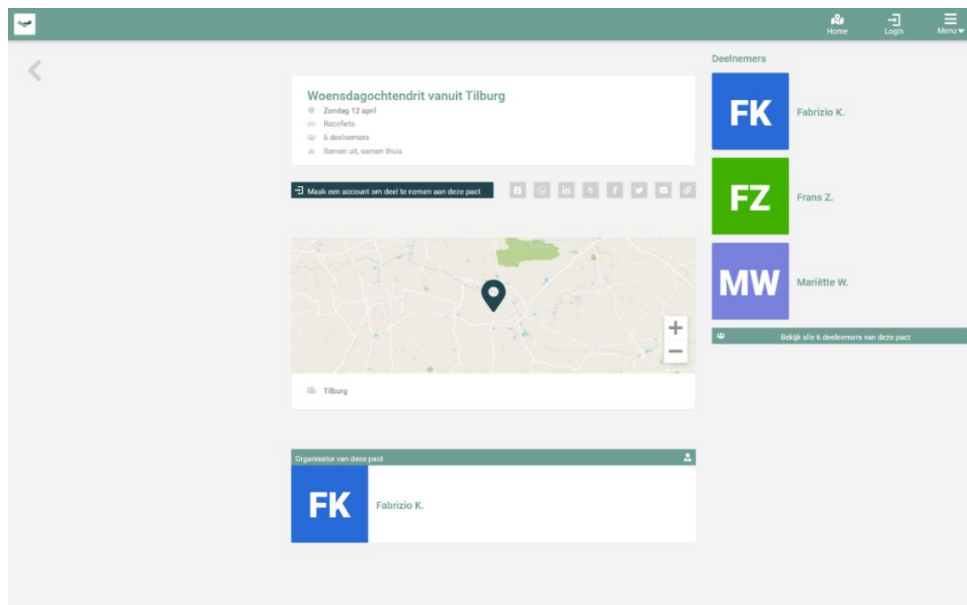
Pactinformatie zijn onderdelen die informatie bieden voor het pact. Een voorbeeld van pactinformatie is de locatie van het pact of de start tijd. Pactinformatie moet ook voorzichtig mee omgegaan worden, het is ongewenst dat bezoekers ongevraagd mee kunnen fietsen met een pact. De informatie die dit mogelijk maakt, bijvoorbeeld de start tijd en de exacte locatie, moet dus verborgen worden voor bezoekers.

Ik ben ieder onderdeel afgegaan en heb de afweging gemaakt of het onderdeel zichtbaar mag zijn voor de bezoeker, ik heb iedere beslissing ook toegelicht in het onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf). Uiteindelijk heb ik een overzichtelijke tabel gemaakt waar ieder onderdeel in staat en of deze zichtbaar is voor de bezoeker, zichtbaar is met aanpassingen of niet zichtbaar is. Voor onderdelen die zichtbaar zijn met aanpassingen heb ik ook genoteerd wat deze aanpassingen zijn. Deze tabel is te zien in het onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf).

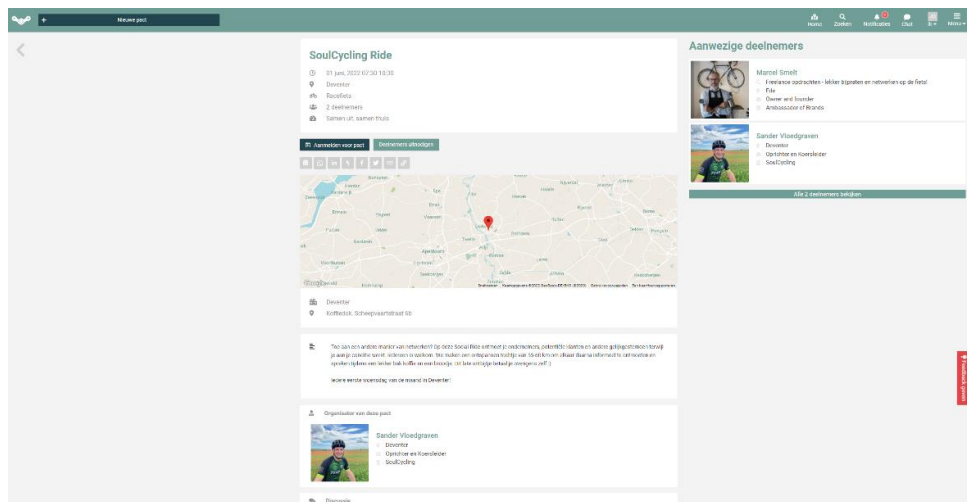
Nu ik alle informatie heb wat er wel en niet getoond mag worden ben ik bezig gegaan met het ontwerp, het resultaat hiervan is hieronder in [3.1 Resultaat](#) te zien.

### 3.1 Resultaat

Ik heb mijn bevindingen toegepast in een vernieuwd design, deze is in figuur 5 te zien



Figuur 5: Openbare pact pagina design



Figuur 6: Pact pagina oude situatie

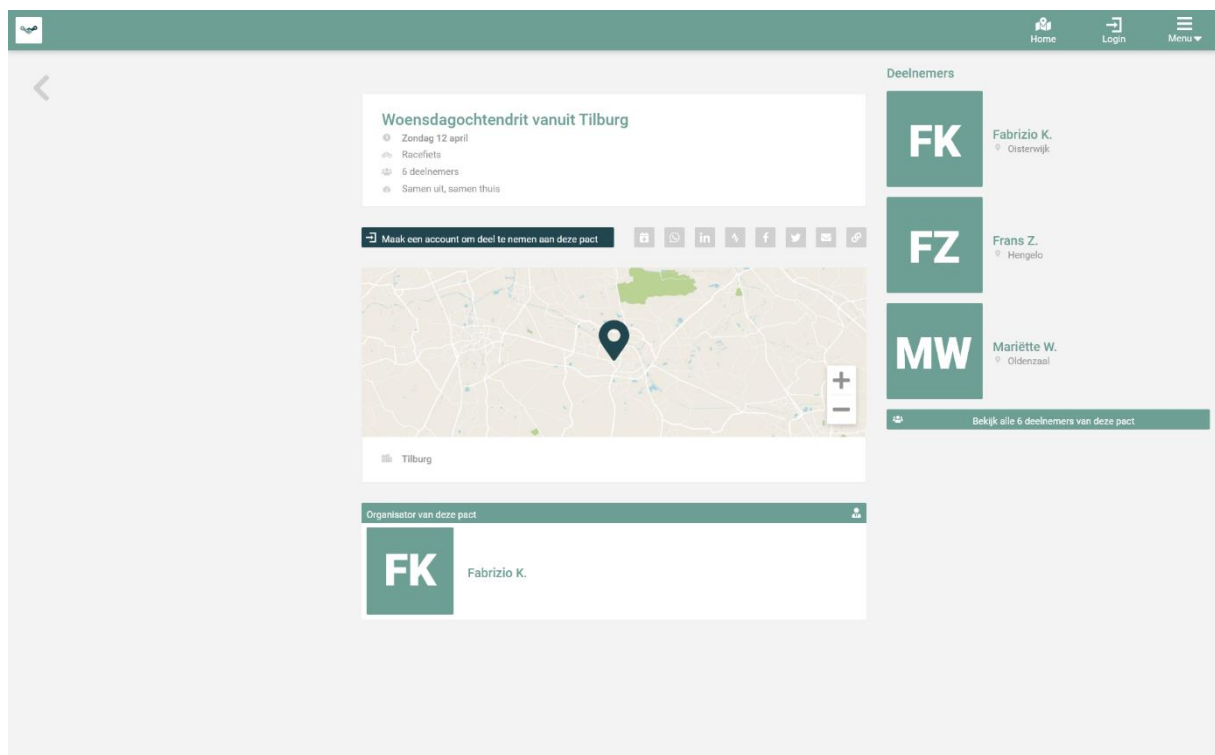
Zoals te zien in figuur 5 is in vergelijking met figuur 6, de oude situatie, alle privacygevoelige informatie weggehaald. Ook alle pact informatie waarmee een bezoeker op zou kunnen dagen bij een pact. Er zijn wel een aantal componenten toegevoegd speciaal voor de openbare pagina:

- Een login knop in de navigatiebalk: Een gebruiker kan namelijk per ongeluk op de openbare pagina terecht zijn gekomen en kan op deze manier makkelijk inloggen.
- Een knop toegevoegd om een account aan te maken, dit om de stap zo klein mogelijk te maken om voor een bezoeker te registreren.
- De gebruikers profielen hebben ook allemaal unieke kleuren, hierdoor zit er meer verschil tussen gebruikers profielen en zijn ze makkelijker uit elkaar te houden.

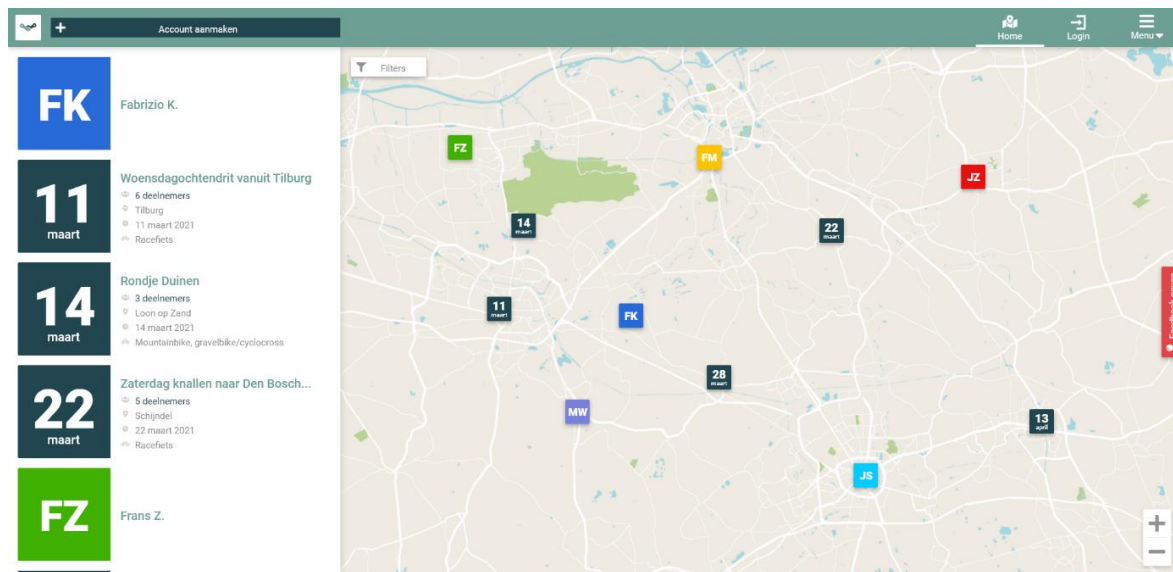
## Feedback UX-Designer

- Goede positionering voor de “deelnemen” knop.
- De plaats van een gebruiker moet ook getoond worden op de openbare pagina's, dit is namelijk cruciale informatie voor een pact.
- Ik zou niet de extra padding toevoegen tussen de rij met buttons en de locatie-rij, en tussen de locatie-rij en de organisator-rij
- De kleurtjes voor de gebruikers zou ik nog even niet doen. Ik ben het ermee eens dat het de kaart opfleurt. Maar enerzijds staan er nog behoorlijk wat veranderingen aan de kaart/lijst op de planning. Die vormen al met al best een ander design dadelijk. Ik vind ook dat de kleuren iets moeten betekenen. En ik had natuurlijk juist het grijze bedacht, met de tegenovergestelde reden.

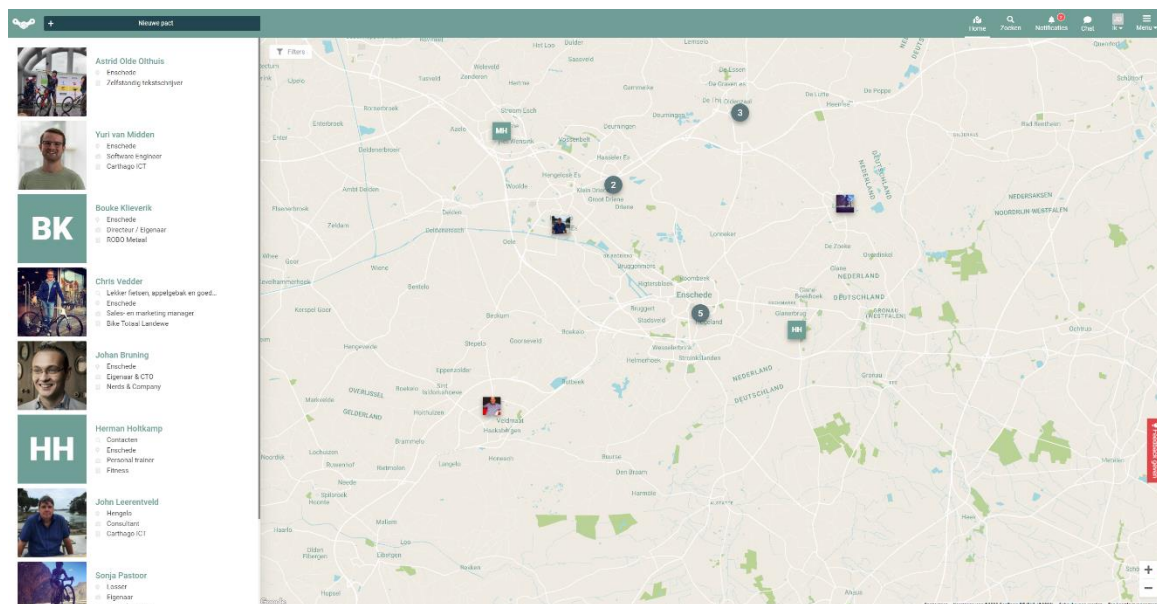
De feedback heb ik weer uitgewerkt in een nieuw design, deze is in figuur 7 te zien.



Figuur 7: Resultaat feedback openbare pact design



Figuur 8: Openbare kaart pagina design



Figuur 9: Kaart pagina oude situatie

Figuur 8 toont het ontwerp van de openbare kaart pagina. In vergelijking met figuur 9, de oude situatie van de kaart, is de privacygevoelige informatie weggehaald.

Ook hier is een login knop toegevoegd in de navigatiebalk.

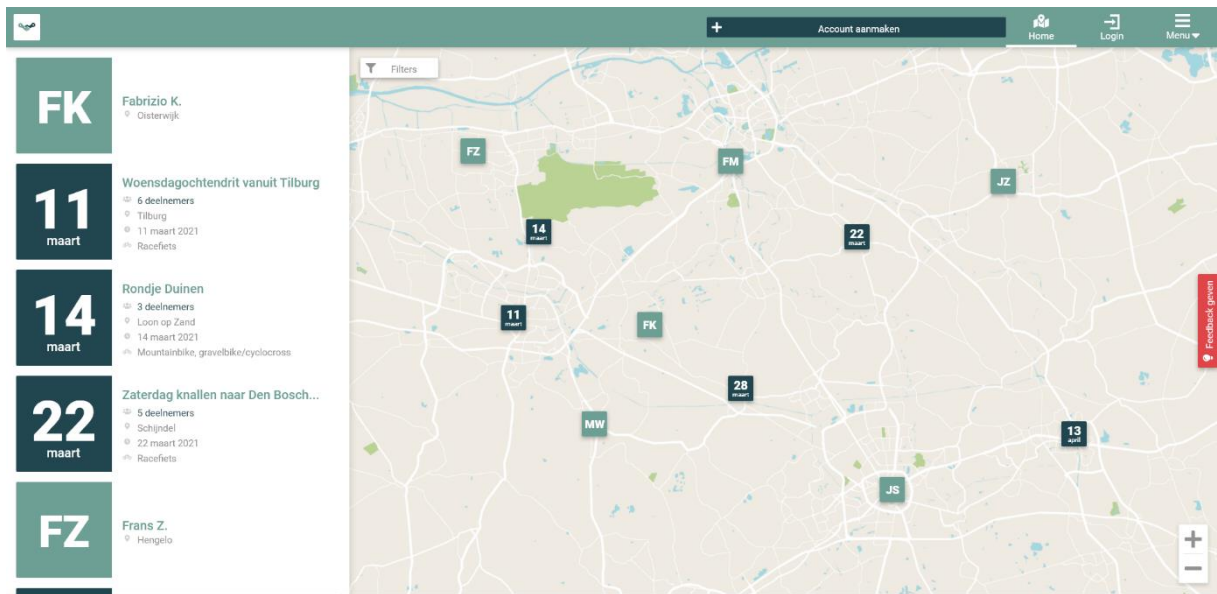
De knop voor een pact aanmaken is vervangen met een knop om een account aan te maken, een bezoeker kan namelijk geen pact aanmaken.

Ook in dit ontwerp zijn de profiel foto's vervangen voor de eerste letter van de voor- en achternaam en heeft elke gebruiker een andere kleur. Wel hebben alle pacts dezelfde kleur, hierdoor is het makkelijker om pacts met gebruikers te scheiden.

## Feedback UX-Designer

- De registratieknop had ik rechts in de header navigatie geplaatst, omdat ie daar vaak staat. Ook wilde ik wat contrast creëren met de 'pact aanmaken' knop, als je eenmaal ingelogd bent.
- Zeer goede toevoeging dat je de login knop in het menu hebt toegevoegd.

De feedback heb ik weer uitgewerkt in een nieuw design, deze is in figuur 10 te zien.

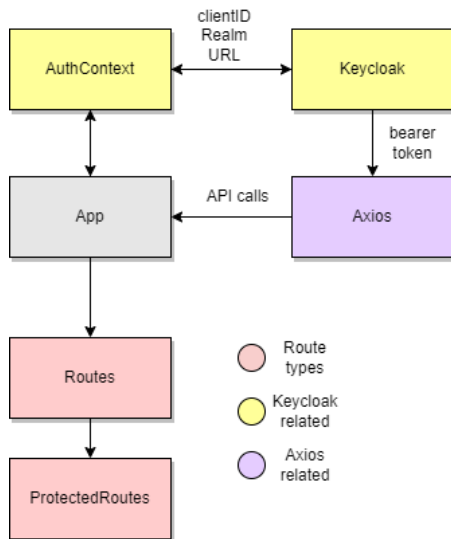


Figuur 10: Resultaat feedback openbare kaart design



### 3.2 Implementatie

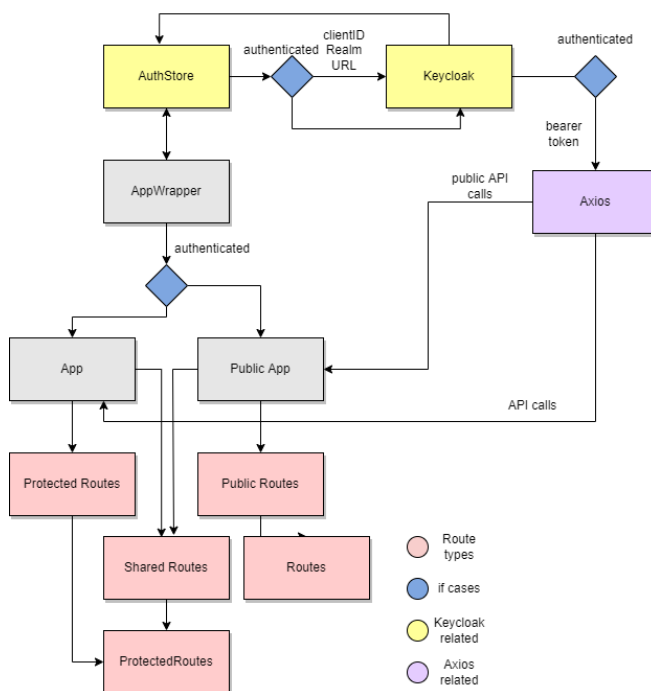
De huidige opzet van de website gaat ervan uit dat een gebruiker altijd ingelogd is wanneer die gebruik maakt van de website. Een bezoeker die nu een pagina probeert te bereiken wordt doorgestuurd naar de login pagina. Er moet dus een onderscheid worden gemaakt van publieke pagina's voor bezoekers en pagina's waar alleen gebruikers bij kunnen. De URL voor de publieke en gebruikers pagina moet wel hetzelfde blijven, anders kunnen pacts niet gedeeld worden. De huidige architectuur is in figuur 11 te zien.



Figuur 11: Architectuur huidige opzet app

De oplossing hiervoor is een aparte PublicApp class aan te maken. De PublicApp bevat dezelfde onderdelen als de App zonder de authenticatie laag en een eigen routes component voor de publieke pagina's, "Public Routes". De kaart en pact pagina gaan via de public routes, deze pagina's hebben dezelfde URL als de beveiligde versie om het delen van pacts makkelijk te maken. De rest van de routes staan in de "Shared Routes", deze wordt gedeeld met de publieke en beveiligde app.

Alle shared routes zijn alleen te bereiken als de gebruiker is ingelogd, als een bezoeker deze pagina probeert te bereiken wordt die verwezen naar de inlogpagina. De authenticatie laag is wel nodig voor de PublicApp om te controleren of de gebruiker is geautoriseerd. Daarom is de authenticatie laag nu een store geworden zodat op elke plek in de applicatie opgevraagd kan worden of de gebruiker is ingelogd. De nieuwe architectuur is te zien in figuur 12.



Figuur 12: Architectuur nieuwe opzet app

Voor de publieke pagina's zijn er nieuwe business logic classes aangemaakt, maar wordt er wel gebruik gemaakt van dezelfde views als de beveiligde pagina's. Dit om dubbele code te voorkomen. De views moesten wel aangepast worden om gebruik te kunnen maken van de nieuwe public types van profiles en pacts. De publieke variant mag namelijk niet alle informatie bevatten die een ingelogde gebruiker wel mag zien. Aan iedere view zit nu een nieuwe prop om te bepalen of het om de publieke variant gaat.

Voor de backend zijn er een aantal nieuwe API-calls aangemaakt voor de bezoeker, deze moet namelijk minder informatie bevatten dan de beveiligde calls. De volgende calls zijn toegevoegd

Type	Call	Response
GET	api/public/pact/:id	<ul style="list-style-type: none"> <li>- Titel</li> <li>- Datum</li> <li>- Plaats</li> <li>- Fiets type</li> <li>- Deelnemers aantal</li> <li>- Samen uit samen thuis?</li> <li>- Organisator</li> <li>- Deelnemers</li> </ul>
GET	api/public/profile/user/:id	<ul style="list-style-type: none"> <li>- Voornaam</li> <li>- Eerste letter achternaam</li> <li>- Latitude profiel marker</li> <li>- Longitude profiel marker</li> <li>- Plaats</li> </ul>
GET	api/public/map?:filter-opties	<ul style="list-style-type: none"> <li>- Pacts die voldoen aan het filter</li> <li>- Profielen die voldoen aan het filter</li> <li>- Aggregate markers (verzameling van profielen) die voldoen aan het filter</li> </ul>
GET	api/public/profile?:filter-opties	Profielen die voldoen aan het filter
GET	api/public/pact?:filter-opties	Pacts die voldoen aan het filter

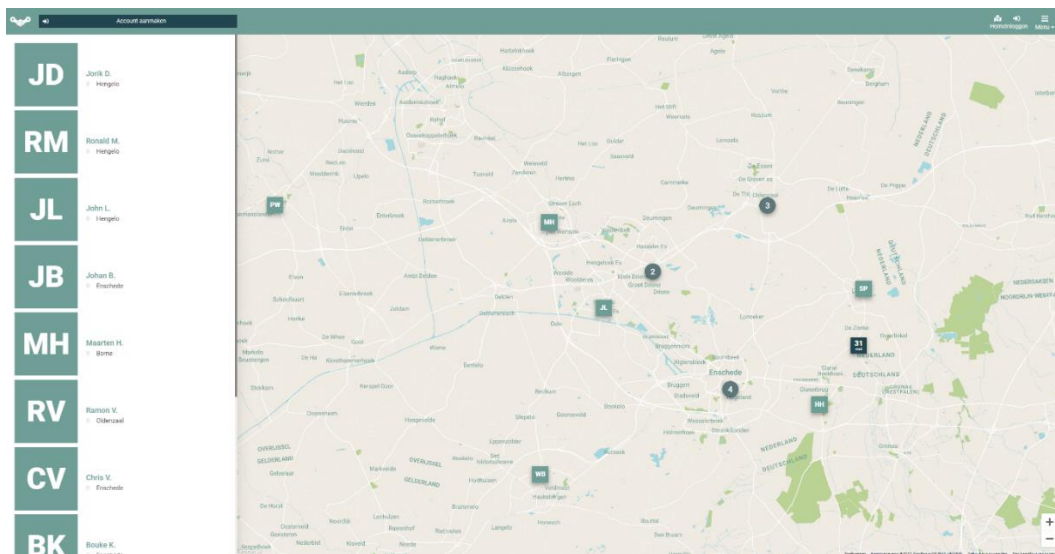
### 3.3 Uiteindelijk resultaat

Het uiteindelijke resultaat is bijna hetzelfde geworden als het ontwerp, er zijn alleen een paar aanpassingen.

De feedback knop is weggehaald, dit is gedaan omdat een bezoeker alleen feedback mag geven via het contact formulier.

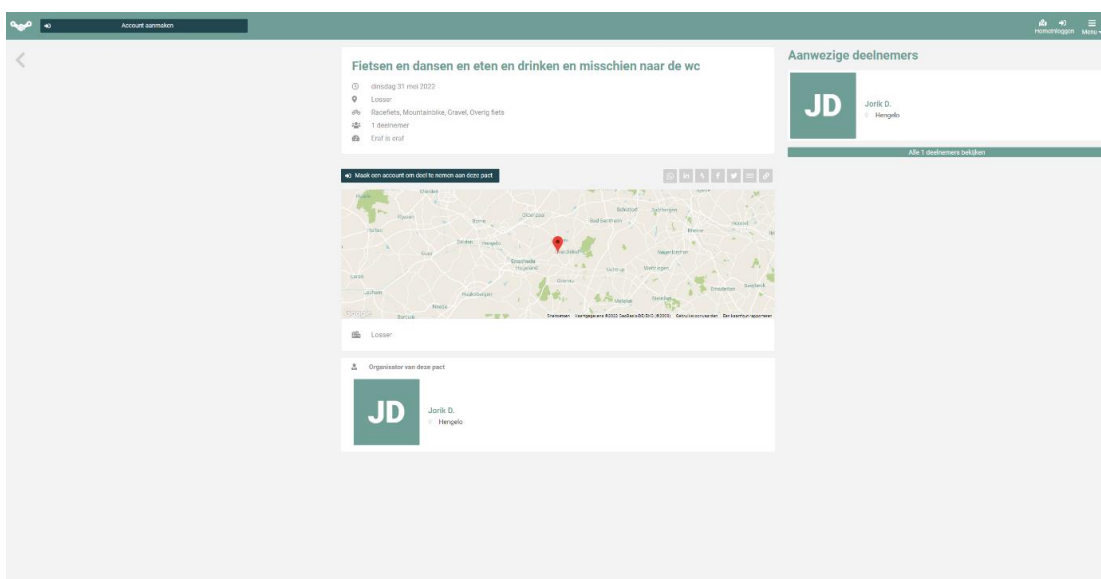
De filter knop is ook weggehaald, omdat een bezoeker niet hoeft te kunnen filteren op het type fiets wat een gebruiker heeft.

De knop voor een account aanmaken is ook terug naar links verplaatst om het design consistent te houden. Het resultaat is te zien in figuur 13



Figuur 13: Uiteindelijke resultaat publieke kaart pagina

Het eindresultaat van de pact informatiepagina is bijna hetzelfde gebleven als het ontwerp. Er zijn enkele kleine aanpassingen. De knop voor een account aanmaken is toegevoegd linksboven, en de social sharing knoppen staan naast elkaar in plaats van uit elkaar. Het resultaat is te zien in figuur 14.



Figuur 14: Uiteindelijke resultaat publieke pact pagina

### 3.4 Testen

Om er achter te komen of het vernieuwde design te begrijpen is voor nieuwe gebruikers heb ik een aantal testen geschreven. Deze testen zijn ontworpen om de doelstellingen van de publieke pagina's te evalueren vanuit het gebruikersperspectief. De volgende testen zijn hiervoor geschreven:

- Ga naar de website, wat is u eerste gevoel als u naar de kaart kijkt?
- Is de kaart volgens u overzichtelijk?
- Ga naar het pact "Fietsen, dansen en drinken".
- Is er volgens u genoeg informatie over dit pact beschikbaar om te weten of het een pact is waar u aan mee zou willen fietsen?
- Stel u zou ook mee willen fietsen met dit pact, is dit genoeg reden om een account aan te maken volgens u?

De antwoorden van de personen staan hieronder vermeld.

#### **Persoon 1:**

- De kaart is een beetje eentonig. Ik zie wel meteen dat 12 juni een pact is. Ik zou de onderscheiding tussen gebruikers en pacts beter maken, bijvoorbeeld een icoon toevoegen.
- Ja nu wel maar misschien niet meer als er heel veel meer gebruikers zijn.
- Ik klik op de marker waar 12 juni op staat, dat is de correcte pact en klik op het plaatje.
- Nee, ik mis wat ik kan verwachten wat voor mensen er mee gaan fietsen. En wat voor route het pact is.
- Ja, ik zou sowieso al een account aanmaken om erachter te komen wat voor deelnemers er zijn.

#### **Persoon 2:**

- Ik heb het gevoel dat je mensen hiermee makkelijk kan vinden.
- Ja, als er teveel mensen op beeld komt wordt het een groep. Ik vind het makkelijk om het verschil te zien tussen gebruikers en pacts en vind het niet onoverzichtelijk dat alle gebruikers dezelfde kleur hebben,
- Ik vind het moeilijk om een losse pact te vinden, misschien handig om een zoekfunctie toe te voegen? Of de gebruikers uit de lijst aan de zijkant te halen.
- Ik vind de termen die gebruikt worden niet allemaal even duidelijk, misschien een idee om bij bijvoorbeeld: samen uit samen thuis een informatie knopje te zetten wat het inhoud. Of als je met je muis over het icoon hooft dat dan de naam wordt getoond. Misschien om het aantal kilometers toe te voegen dat het pact is om een idee te krijgen of het pact voor mij te fietsen is.
- Ja, als ik mee wil fietsen moet ik een account aanmaken dus waarom niet?

**Persoon 3:**

- Ik vind de kaart er netjes uit zien
- Ja, ik vind de kaart overzichtelijk, ik vind het niet meteen duidelijk dat 12 juni een pact betekend.
- Ik vind het moeilijk om er naar te zoeken zonder een zoek functie. Ik ga scrollen door de zijlijst totdat ik het pact tegenkom. Als ik naar de kaart kijk zou ik niet makkelijk een specifiek pact kunnen vinden.
- Ik mis iets van een korte beschrijving wat het pact inhoudt. Misschien een foto van de omgeving?
- Ja, als het mij interesseert waarom niet

**Resultaat**

- Het is moeilijk om pacts te vinden op de kaart.
- De termen die gebruikt worden op de pact detail pagina zijn niet allemaal duidelijk.
- Het is fijn om meer te weten over de route van het pact voor een bezoeker.
- Iedereen zou een account aanmaken als die mee wilde fietsen met het pact.

### 3.5 Social sharing image

Nu bezoekers ook toegang hebben tot Pedalpact.cc is het cruciaal om pacts te kunnen delen op social media. Om het delen effectiever te maken moet er een social sharing image komen, dit is een afbeelding die op social media verschijnt na het plaatsen van de link. Een pact image heeft de volgende informatie nodig:

- De datum van het pact
- De titel
- De locatie
- De vereiste type fiets

Aan mij is de taak om dit uit te werken in HTML. Als dit resultaat voldoet zal de backend developer dit implementeren in de website zodat de correcte tekst wordt getoond voor het gedeelde pact.

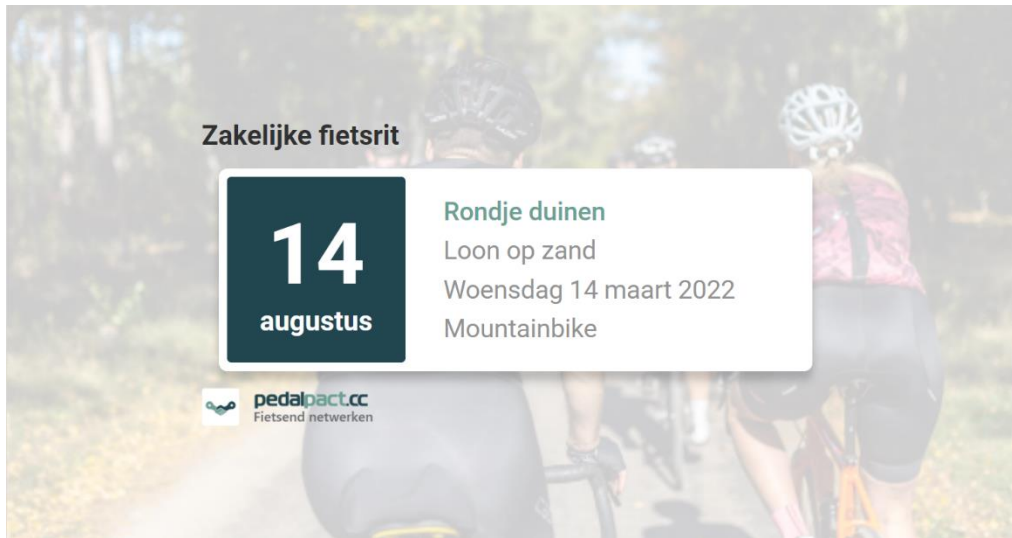
Het ontwerp moet duidelijk leesbaar zijn. Een social sharing image heeft namelijk een vast formaat, dit verschilt per social media app, op Facebook is dit 1200 bij 628 pixels. Op de mobiel ziet zo'n afbeelding er als figuur 15 uit.



Figuur 15: Social sharing image Facebook mobiel voorbeeld

#### Ontwerp

Voor het ontwerp probeerde ik de stijl van de Pedalpact.cc website aan te houden. De card heeft hetzelfde ontwerp als de pact card op de website. De titel moet goed opvallen dus deze heeft een andere kleur, is groter, en heeft een dikker lettertype. De datum heeft hetzelfde ontwerp als op de website. De achtergrond moet niet te veel opvallen dus zit er een witte laag overheen, de afbeelding is afkomstig van de registreer pagina van Pedalpact.cc. Het uiteindelijke ontwerp is figuur 16 te zien. Dit ontwerp is ook goedgekeurd door de product owner, Daan.



Figuur 16: Social sharing image ontwerp

De backend developer heeft dit ontwerp geïmplementeerd en werkt succesvol met Facebook. Een week later heeft de product owner een eigen nieuw ontwerp. Het vorige ontwerp was op de telefoon niet duidelijk te lezen. Dit nieuwe ontwerp moet ik na gaan maken in HTML. Gelukkig kan ik veel van de styling hergebruiken van het vorige ontwerp. Na een aantal revisies is figuur 17 het uiteindelijke ontwerp geworden voor de social media image.



Figuur 17: Social sharing image definitief ontwerp



### 3.6 Reflectie

Het maken van de openbare pact- en kaart pagina bleek meer werk te zijn dan van tevoren gedacht. Het was eerst heel erg mijn weg vinden in de code, erachter komen hoe alle functionaliteiten werkte. Ik had eerst voor alle publieke pagina's nieuwe views en classes aangemaakt maar dit werd niet goedgekeurd door de opdrachtgever, Dennis. De code bevatte namelijk heel veel dubbele code van de beveiligde pagina's. Toen kwam ik erachter dat de views niet veranderen tussen de publieke en afgesloten pagina's behalve dat er bepaalde onderdelen afschermt moesten worden. Uiteindelijk moest een heel groot gedeelte van de applicatie herbouwd worden om publieke pagina's te ondersteunen. Omdat er zoveel herschreven moest worden ontstonden er ook veel bugs die lastig te debuggen waren en dus ook veel tijd kostte om op te lossen.

Er werden ook steeds meer eisen toegevoegd door de product owner waardoor het afmaken van de functionaliteiten langer duurde. Een voorbeeld is het inzoomen op de kaart van de huidige locatie van de bezoeker wanneer die de locatie bepaling accepteert.

Het ontwerpen van de publieke pagina's was leuk om te doen, ik heb al best wat ervaring met ontwerpen door mijn minor en mijn specialisatie. Het echte ontwerp gedeelte mist wel omdat het een bestaand project is waar ik alleen aanpassingen maak aan het huidige ontwerp. Het ontwerp is gemaakt in Adobe XD, daar had ik nog niet eerder in gewerkt en was leuk om te ontdekken hoe het anders werkt dan Figma. Een aantal functies zijn makkelijker gemaakt in Adobe, zoals kleuren pallets toevoegen aan je library. Het was ook leerzaam om mijn gemaakte ontwerp te kunnen bespreken met een UX-designer.

## 4 Pact markers meer op laten vallen

Pedalpact.cc heeft op het moment al meer dan 1000 gebruikers, deze zijn allemaal zichtbaar op de kaart. Gebruikers gebruiken de kaart voornamelijk om pacts te zoeken, deze zijn moeilijk te vinden tussen alle gebruiker markers. Pacts moeten dus meer opvallen op de kaart, er moet een nieuw design komen om beter onderscheid te maken tussen pacts en gebruikers.

Voor het onderzoek gebruik ik de Thinking aloud methode over mogelijke oplossingen voor het meer op te laten vallen van pact markers. De beste oplossingen die ik kon bedenken zijn de volgende:

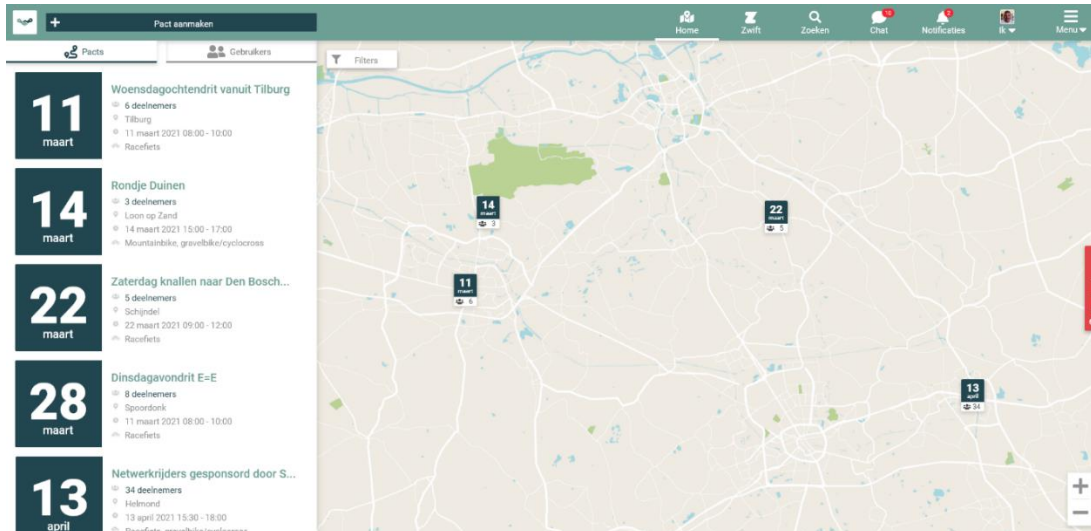
- Gebruiker markers kleiner maken
- Pact markers groter maken
- Gebruiker markers los tonen van pact markers
- Pact markers een prominentere kleur geven
- Meer informatie tonen van een pact

Voor ieder van deze oplossingen heb ik een ontwerp gemaakt, deze zijn te zien in het onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf). Geen van deze oplossingen was op zichzelf genoeg om de pact markers meer op te laten vallen. Ik heb er toen voor gekozen om ze te combineren, de pact markers groter maken, gebruiker markers los tonen van pact markers en meer informatie tonen van een pact. Hierdoor zijn pacts makkelijk te onderscheiden van gebruikers, omdat ze nu los worden getoond. Er zijn veel minder pacts dan gebruikers dus heb ik ze groter gemaakt om meer op te vallen en extra informatie toegevoegd zodat ze nog groter zijn. Deze extra informatie toont het aantal deelnemers aan het pact.

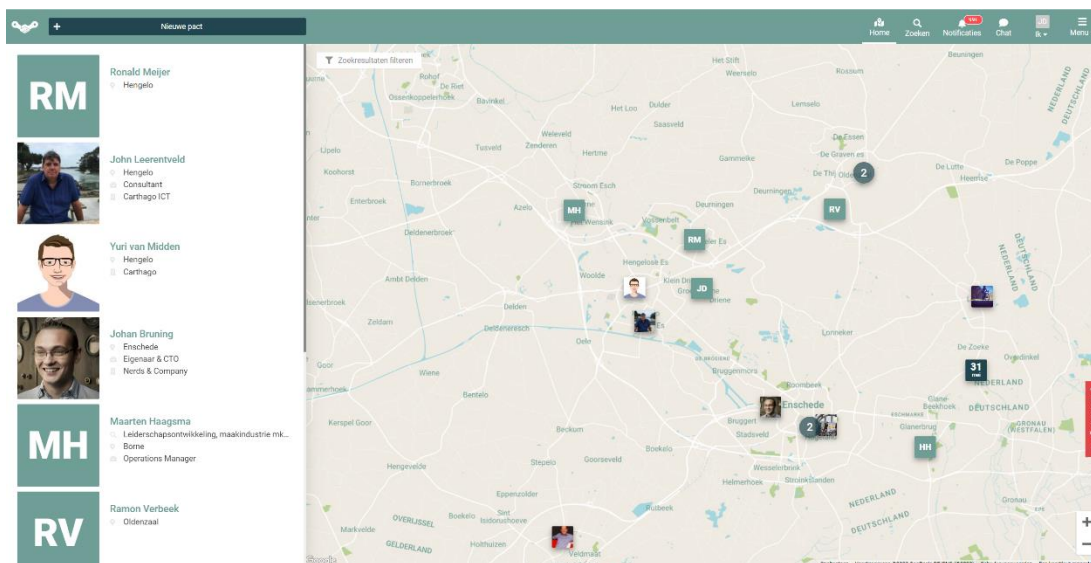
Het uiteindelijke resultaat is in [4.2 Definitieve oplossing](#) te zien.

## 4.2 Definitieve oplossing

In figuur 18 is de oplossing te zien. De oplossing is een combinatie geworden van de pact markers loszetten van de gebruiker markers en meer informatie te tonen van een pact. In vergelijking met de oude situatie, figuur 19, is de kaart veel leger en is het overzicht van pacts duidelijker te zien. Pacts tonen nu ook het aantal deelnemers onder de datum.



Figuur 18: Definitieve oplossing kaart



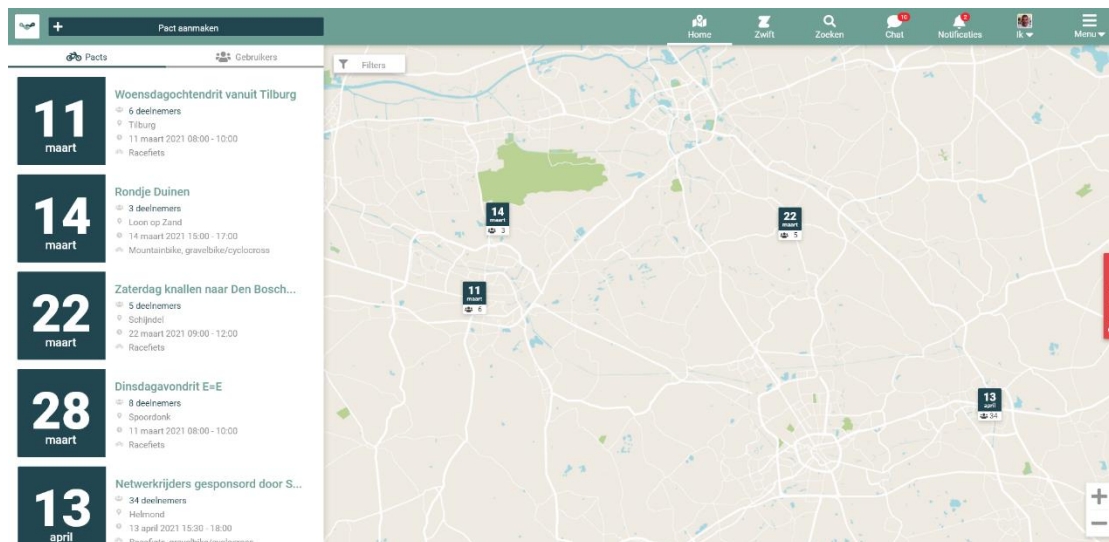
Figuur 19: Oude situatie kaart

### Feedback UX-designer

- Icoon van pacts veranderen naar een fiets en een slag kleiner
- Icoon van gebruikers veranderen naar icoon voor deelnemers en ook een slag kleiner.
- Geen inner-shadow of schaduw achter de knop
- Geen ruimte tussen de twee knoppen

### Resultaat

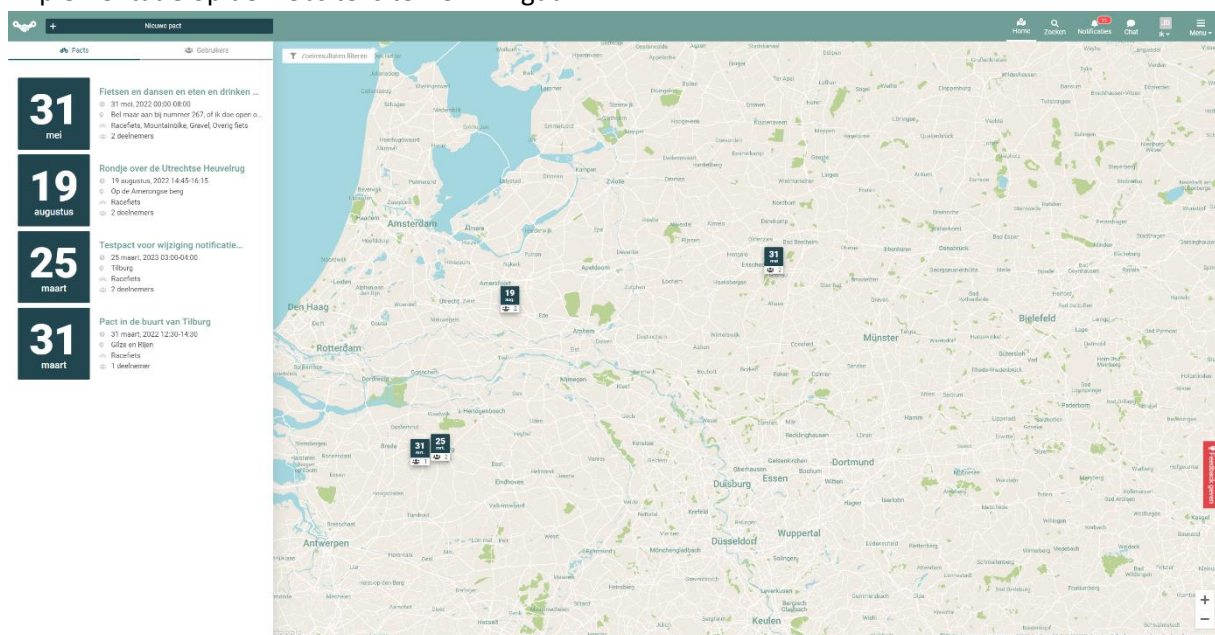
De gekregen feedback heb ik uitgewerkt in het uiteindelijke resultaat, figuur 20. Dit resultaat ga ik implementeren in de website.



Figuur 20: Resultaat feedback kaart design

### 4.3 Implementatie

Om de staat van de knop bij te houden maak ik gebruik van een store. De store gebruikt een enum die “pacts” of “users” kan zijn. Deze state wordt bijgehouden door de kaart om te bepalen of die de pact of gebruiker markers moet tonen. De zij-balk houdt deze state ook bij om te bepalen of die pact of profile cards moet tonen. Het aantal deelnemers wordt bepaald door een call naar de backend te doen om de profielen op te halen die aan het pact zijn gekoppeld. Het resultaat van de implementatie op de website is te zien in figuur 21.



Figuur 21: Resultaat implementatie kaart

#### 4.4 Testen

Om er achter te komen of het vernieuwde design te begrijpen is voor nieuwe gebruikers heb ik een aantal testen geschreven. Deze testen zijn ontworpen om de doelstellingen van het los maken van pacts en gebruikers te evalueren vanuit het gebruikersperspectief. De volgende testen zijn hiervoor geschreven:

- Ga naar de website, wat ziet u nu op de kaart?
- Wat betekent de witte streep onder het pact?
- Toon de gebruikers op de kaart, is de manier om dit te doen logisch?
- Is de kaart overzichtelijker nu de pacts los worden getoond van de gebruikers?

De antwoorden van de personen staan hieronder vermeld.

##### **Persoon 1:**

- Ik zie pacts.
- Hoeveel deelnemers er zijn voor het pact.
- Ik klik op de knop "gebruikers linksboven". Ik vind dit een logische manier om te doen.
- Ja, wanneer zou je gebruikers en pacts tegelijk willen zien?

##### **Persoon 2:**

- Ik zie alleen pacts op de kaart. Ik haal eruit dat dit pacts zijn omdat er het aantal deelnemers en de datum staan.
- Aantal deelnemers
- Ik zie geen gebruikers op de kaart, zitten ze allemaal in het pact? (met een beetje hulp) Ik klik op de gebruikers knop linksbovenin. De manier om dit te doen vind ik logisch dit zou ik ook uit intuïtie doen.
- Ja, ik vind het wel fijn dat het nu in tweeën is gedeeld.

##### **Persoon 3:**

- Ik zie pacts op de kaart. Ik zie dat pacts aan staat bij de knoppen.
- Het aantal deelnemers van het pact.
- Ik klik op de gebruikers knop links bovenin, dit is logisch.
- Ja, het is overzichtelijker geworden. Ik zou wel de pacts groeperen wanneer je verder uitzoomt net zoals gebruikers.

##### **Resultaat:**

- De gebruikers zijn allemaal enthousiast over deze nieuwe functionaliteit en hebben geen opmerkingen wat ze er minder aan vinden.

#### 4.5 Reflectie

Deze onderzoeksvraag kostte minder tijd om te realiseren dan de vorige onderzoeksvraag. De styling was nog wel lastig op mobiel, de elementen gebruiken allemaal absolute waardes waardoor een mijn initiële CSS oplossing, een flexbox zodat alle componenten onder elkaar kwamen te staan, niet wilde werken. Ik heb uiteindelijk gekozen om een padding te gebruiken om de kaart lager te zetten.

Het implementeren van de gebruikers en pacts tab vergde minder tijd omdat ik nu bekend ben met hoe de code in elkaar zit. Waar ik wel lang mee bezig was het goed krijgen van de lijst van gebruikers of pacts die werd geladen. Er zat een vervelende bug in die ervoor zorgde dat deze lijst dubbel werd geladen na het in- of uitzoomen van de kaart. Deze bug heb ik uiteindelijk opgelost, het probleem was dat de lijst niet goed werd geleegd na het opnieuw halen van data.

De product owner is ook heel blij met het resultaat, het maakt een positief verschil voor de gebruikerservaring in zijn ogen.

## 5 Automatisch testen

Er zijn nog geen testen aanwezig in de applicatie. Hierdoor kost het veel tijd om de applicatie testen voor bugs als er een branch wordt gemerged met master. Automatische testen lossen dit probleem op, deze worden tijdens de CI/CD uitgevoerd zodat een branch niet gemerged kan worden als er functionaliteiten kapot zijn gegaan.

In mijn onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf) beschrijf ik hoe ik heb gekozen voor de test-frameworks voor de testen. Voordat ik de test-frameworks koos voor de testen ging ik eerst uitzoeken welke type testen er waren voor de back- en frontend.

Voor de backend koos ik ervoor om alleen API layer testen te schrijven omdat de database al in een release omgeving draait. Voor de frontend heb ik gekozen om E2E (End-to-End) testen en unit testen te schrijven. Hierna ben ik gaan onderzoeken welke test-frameworks er zijn voor de backend. Voor Java was dit schaars en was alleen JUnit echt geschikt, hier waren nog wel veel libraries voor beschikbaar die het mocken en asserts verbeteren.

Voor de frontend waren veel meer opties, ik heb de drie test-frameworks met de meeste potentie met elkaar vergeleken door middel van een rubriek.

Criteria	Onvoldoende (4)	Voldoende (6)	Goed (8)	Zeer Goed (10)
<b>Performance</b>	Het uitvoeren van de testen duurt te lang om efficiënt gebruikt te worden.	Het uitvoeren van de testen gaat vlot maar biedt geen ondersteuning voor parallel testing.	Het uitvoeren van de testen gaat vlot en biedt ondersteuning voor parallel testing.	Het opstarten en uitvoeren van de testen gaat vlot en biedt ondersteuning voor parallel testing.
<b>Community</b>	De framework heeft geen Github pagina of Gitter pagina	De framework heeft een Github of Gitter pagina	De framework heeft een Github en Gitter pagina maar zijn niet zeer actief	De framework heeft een actieve Github en Gitter pagina.
<b>Documentatie</b>	Er is geen of slechte documentatie beschikbaar.	Er is een documentatie beschikbaar maar biedt geen voorbeelden	Er is een uitgebreide documentatie beschikbaar met voorbeelden	Er is een uitgebreide documentatie beschikbaar met voorbeelden en video tutorials.
<b>Mocking</b>	De framework heeft geen ingebouwde mocking functionaliteit	De framework heeft geen ingebouwde mocking functionaliteit maar kan dit wel krijgen d.m.v. een library	De framework heeft ingebouwde mocking functionaliteit maar niet duidelijk gedocumenteerd	De framework heeft ingebouwde mocking functionaliteit die duidelijke is gedocumenteerd
<b>Code</b>	De code is onleesbaar en niet te begrijpen zonder eerst de documentatie te lezen.	De code is leesbaar en te begrijpen zonder documentatie.	De code is leesbaar en biedt een versimpelde manier van debuggen	De code is goed te begrijpen en leesbaar en heeft een uitgebreide debug tool.



## E2E testen

Voor de E2E testen heb ik de volgende drie test-frameworks vergeleken: WebDriverIO, Testcafe en Cypress. Ik heb per test-framework deze rubriek ingevuld en heb ook een toelichting gegeven waarom ik het zoveel punten gaf. Voor de E2E testen kwam ik uit op Cypress, de documentatie en code is het duidelijkst. De community van Cypress is ook het grootste waardoor oplossingen vinden voor problemen makkelijker gaat.

Voor de E2E testen heb zoveel mogelijk functionaliteiten proberen te testen. Ik heb alle update, verwijder en creëer functies getest. De navigatie door de applicatie heb ik ook getest.

## Frontend unit testen

Ditzelfde heb ik gedaan voor de unit testen, hier heb ik de volgende drie test-frameworks vergeleken: Jest, Jasmine en Mocha. Voor de unit testen kwam ik uit op Jest. In vergelijking met de andere frameworks heeft deze meer functionaliteiten en een grotere community. Jest werkt ook goed samen met de React testing library waarmee React componenten gemakkelijker getest kunnen worden.

Ik wilde hierna unit testen gaan schrijven voor de applicatie. Dit bleek lastiger dan gedacht, de componenten in de applicatie zijn allemaal afhankelijk van veel props. Deze props zijn ook weer afhankelijk van andere componenten. De applicatie is een soort kaartenhuis, als alles precies goed is opgesteld werkt de applicatie feilloos maar zodra er één onderdeel wordt veranderd of apart wordt getrokken stort alles in. In mijn aanbevelingen ga ik hier dieper op in.

Unit testen worden voornamelijk geschreven omdat deze snel te implementeren en te testen zijn maar voor dit project zou dan bijna de hele applicatie gemocked moeten worden. Daarom heb ik alleen E2E testen geschreven.

## Backend unit testen

Voor de backend heb ik unit testen geschreven in JUnit. Voor het mocken van services gebruik ik Mockito, dit is de grootste mocking service library voor JUnit. De documentatie hiervan is duidelijk en de code is logisch. Voor assertions gebruik ik AssertJ, deze library maken het gebruik van asserts makkelijker en bieden meer opties dan de standaard assertions die JUnit leveren. Voor de backend unit testen heb ik een aantal nep profielen en pacts aangemaakt. Een voorbeeld van een mock profiel en pact zijn te zien in figuren 22 en 23. Deze mocks gebruik ik in een aantal testen waar een interactie een pact of profiel vereist. Een voorbeeld van een test is te zien in figuur 24, hier gebruik ik ook een aantal mock profielen pacts om een week rapport te sturen.

```
private static Profile user1() {
    Profile profile = new Profile();
    profile.setUserId(USER1_ID);
    profile.setFirstname("Jan");
    profile.setMiddlename("de");
    profile.setLastname("Fietser");
    profile.setSearchQuestion("Nieuwe uitdagingen");
    profile.setPlace("Hengelo");
    profile.setActive(now());
    profile.setCreated(now());
    profile.setLatitude(52.1);
    profile.setLongitude(5.1);
    return profile;
}
```

Figuur 22: Mock Profiel



```

private static Pact pact1() {
    Pact pact = new Pact();
    pact.setId(PACT1_ID);
    pact.setDescriptionShort("Hengelo Tochtje door het platteland");
    pact.setCreated(ZonedDateTime.of( year: 2021, month: 4, dayOfMonth: 1, hour: 0, minute: 0, second: 0, nanos: 0, zone: ZoneId.of("Europe/Amsterdam")));
    pact.setCreatorId(PACT_CREATOR_ID);
    pact.setDuration(150);
    pact.setLatitude(52.25);
    pact.setLongitude(6.79);
    pact.setLocation("Hengelo");
    pact.setLocationDescription("In dat steegje in Hengelo");
    pact.setStartTime(LocalDateDateTime.of( year: 2022, month: 5, dayOfMonth: 1, hour: 0, minute: 0, second: 0, nanos: 0, zone: ZoneId.of("Europe/Amsterdam")));
    pact.setBikes(Transformer.transformBikes(RACE, MTB, GRAVEL));
    return pact;
}

```

Figuur 23: Mock Pact

```

@Test
void sendWeekreport() throws Exception {
    WeeklyStatsMessage message = new WeeklyStatsMessage(LocalDate.of( year: 2022, month: 5, dayOfMonth: 28));
    message.pactIdsUpcomingWeek = new UUID[]{PACT1_ID};
    message.pactIdsOpenInvitation = new UUID[]{PACT2_ID};
    message.pactIdsInteresting = new UUID[]{PACT3_ID};
    message.pactIdsNear = new UUID[]{PACT4_ID};
    message.userIdsNear = new UUID[]{USER3_ID, USER2_ID};
    message.pactAttendanceNew = new PactAttendanceNew[]{new PactAttendanceNew(PACT1_ID, new UUID[]{USER2_ID})};

    usersettings();

    testData.withPactAttendance(testData.pact1, now(), testData.user1, testData.user2);
    testData.withPactAttendance(testData.pact2, now(), testData.user1, testData.user2);
    testData.withPactAttendance(testData.pact3, now(), testData.user2);
    testData.withPactAttendance(testData.pact4, now(), testData.user1);

    sendAndVerify(mailerService::queueWeekReport, filename: "weekReport.html", subject: "Jouw weekoverzicht", message(WEEK_REPORT, message));
}

```

Figuur 24: Week report test

## 5.1 Reflectie

Het onderzoek om uit te vinden welke type testen er waren was makkelijk voor de frontend maar lastig voor de backend. Voor de backend was het lastig om een concreet aantal types te vinden, de types verschilden per website en mening. Ik heb daarom een lijst samengesteld van type testen die bij meerdere website overeen kwamen. Voor de frontend was dit niet nodig, hier was het makkelijk om de verschillende type testen te vinden.

Het onderzoek om de frameworks te kiezen voor het testen liep voorspoedig. Door middel van de rubriek was het makkelijk om een goede onderbouwde reden te geven voor welke frameworks ik ging gebruiken.

Het testen van de backend was moeilijk, ik had nog niet eerder met JUnit gewerkt en ik werkte met een backend die ik zelf niet heb geschreven. Gelukkig waren er veel tutorials te vinden over JUnit, ook op de eigen [website](#) van springboot.

Het testen van de frontend was frustrerend. Ik heb eerder met Cypress gewerkt dus ik wist de basis al en dit proces verliep in ieder vorig project altijd voorspoedig. Met dit project ben ik een paar dagen bezig geweest om tekst te kunnen typen in een textbox. Ik ben de hele Github pagina afgegaan om er achter te komen waarom dit niet wilde werken en ben het hele kantoor afgegaan of mensen wisten hoe dit kwam. Niemand snapte waarom dit niet wilde en alle oplossingen op Github werkten niet. Uiteindelijk lukte het mij om dit op te lossen, de oplossing bleek om langer te wachten. De applicatie was nog aan het laden terwijl Cypress probeerde in de textbox te schrijven. Dit probleem keerde telkens weer terug tijdens het testen schrijven dus duren de testen langer om zeker te weten dat alle elementen op de pagina zijn geladen.

Toen ik dit probleem had verholpen ging de rest van de testen schrijven gelukkig makkelijker, het uitvoeren helaas niet. De testen werkten niet consistent, soms slaagde ze allemaal in één keer en soms faalde er een paar omdat de website net iets langer moest laden dan Cypress wachtte. Uiteindelijk heb ik de timing goed gekregen dat alle testen slaagde.

Terwijl ik bezig was met testen schrijven slaagde er telkens een paar testen niet, ik snapte niet waarom totdat ik de acties zelf uit probeerde te voeren op de website. Ik had een grote bug gevonden in de applicatie waardoor de applicatie nog steeds dacht dat de gebruiker een bezoeker was na het inloggen. Hierdoor werd een lege pagina getoond omdat een bezoeker niet bij de pagina moet kunnen. Deze bug oplossen kostte mij ook weer veel tijd omdat het niet duidelijk was waar dit gebeurde, keycloak gaf aan dat de gebruiker ingelogd was en de applicatie gaf dit ook aan. Uiteindelijk was de oplossing simpel, de pagina laten laden tot dat de authstore is geïnitieerd.

Toen ik dit had opgelost was er nog een grote bug opgedoken, de CSS deed het niet meer in de staging omgeving. Deze omgeving werd gebruikt door de product owner om de applicatie te testen voordat een branch gemerged mocht worden. Ik ging dit probleem proberen te debuggen, de CSS package werd niet aangemaakt omdat de volgorde van de CSS files niet meer klopte. Ik kwam er niet uit dus probeerde ik het samen met een collega op te lossen, hier zijn we een hele dag mee bezig geweest en uiteindelijk kwamen we erachter dat het kwam omdat de app en public app lazy werden geladen. Dit haalde we weg en toen werkte de staging omgeving weer normaal.

Tijdens het testen ben ik dus veel bezig geweest met grote bugs oplossen in de applicatie die op kwam duiken door het testen. De testen bewezen dus meteen hun nut ten behoeve van de applicatie. Ik vond het leuk om de testen te schrijven en een testplan op te stellen.

## 6. Premium gebruiker

Op dit moment heeft de applicatie maar één soort gebruiker, een gratis gebruiker. Voor het freemium model moet er een onderscheid worden gemaakt tussen premium en freemium gebruikers. Freemium gebruikers zijn gebruikers die gratis de applicatie gebruiken en premium gebruikers zijn gebruikers die door middel van een abonnement premium functionaliteiten in de applicatie kunnen gebruiken.

Ten eerste moet er een nieuw datamodel komen in de backend om premium gebruikers te accepteren, dit heb ik opgelost door een extra veld toe te voegen genaamd "isPremium". De backend synchroniseert iedere avond met Keycloak om te controleren of er nieuwe premium gebruikers zijn. Een gebruiker kan ook een proefabonnement starten, de gebruiker kan dan 30 dagen gratis de premium functionaliteiten gebruiken. Deze informatie wordt opgeslagen in een JWT-token, hier wordt bijgehouden of de gebruiker een trial heeft lopen en tot wanneer.

Voor mijn onderzoek ben ik alle functionaliteiten in de applicatie nagegaan en gekeken of dit cruciale functionaliteiten zijn. Cruciale functionaliteiten zijn functionaliteiten die nodig zijn om de applicatie te kunnen gebruiken, een voorbeeld voor Pedalpact.cc zou het meedoen aan een pact kunnen zijn. Dit heb ik uitgewerkt in een tabel dat te zien is in mijn onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf). Deze ingevoingen heb ik gegeven aan de product owner en zijn feedback heb ik weer verwerkt in de tabel.

Naast premium gebruikers moet er worden gekeken naar meer inkomstbronnen, ik ga een aanbeveling schrijven met andere manieren om inkomsten te generen. Door meerdere inkomstbronnen is de toekomst van het platform zekerder. Deze aanbevelingen zijn terug te vinden in het kopje [6.2 Welke nieuwe functionaliteiten hebben betaalde gebruikers in de applicatie?](#)

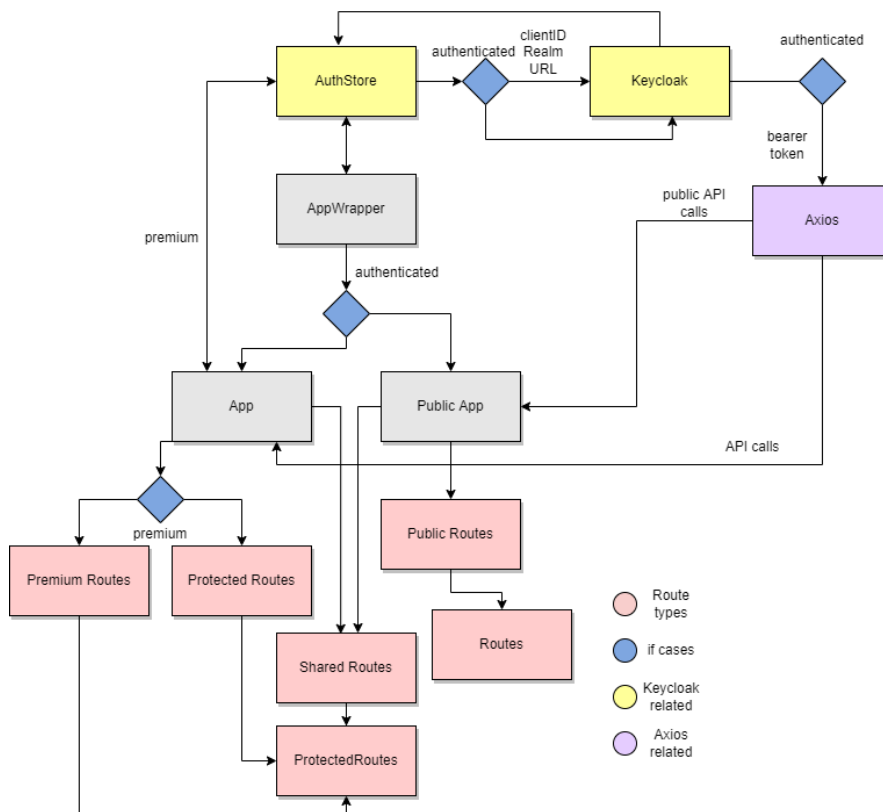
Als laatste ga ik een aanbeveling schrijven over nieuwe exclusieve functionaliteiten voor premium gebruikers. Dit is terug te lezen in het kopje [6.3 Welke andere bronnen van inkomsten kunnen er komen?](#)

## 6.1 Implementatie

Voor premium gebruikers moest er een nieuw veld worden toegevoegd aan de Authstore. De volgende velden zijn toegevoegd: isTrial, isPremium, hasPremiumFunctionalities. Bepaalde functies, zoals een vergrootte profiel marker op de kaart, zijn alleen beschikbaar voor premium gebruikers, dus niet voor gebruikers met een proefabonnement.

Er moest ook een nieuwe route type komen, premium routes. Premium routes zijn routes die alleen beschikbaar zijn voor premium gebruikers. Premium routes is een kind van protected routes, er wordt eerst gecontroleerd of de gebruiker is ingelogd en daarna wordt er gekeken of het een premium gebruiker is.

De nieuwe architectuur van de applicatie is in figuur 22 te zien. De premium routes component is toegevoegd en een nieuwe connectie tussen App en AuthStore waar gevraagd kan worden of de gebruiker premium bezit.



Figuur 25: Nieuwe architectuur applicatie met premium

## 6.2 Welke nieuwe functionaliteiten hebben betaalde gebruikers in de applicatie?

Op dit moment worden alleen bestaande functionaliteiten afgeschermd voor gratis gebruikers. De vraag is of dit voldoende is om de maandelijkse kosten van het abonnement te verantwoorden. Ik ga een aanbeveling schrijven voor nieuwe functionaliteiten exclusief voor premium gebruikers. De functionaliteiten die ik heb bedacht zijn de volgende:

### Premium pacts

Pacts georganiseerd door Pedalpact.cc speciaal voor premium gebruikers. Omdat deze worden georganiseerd door Pedalpact.cc kan er ook, bijvoorbeeld, een lunch gratis worden aangeboden of gratis koffie.



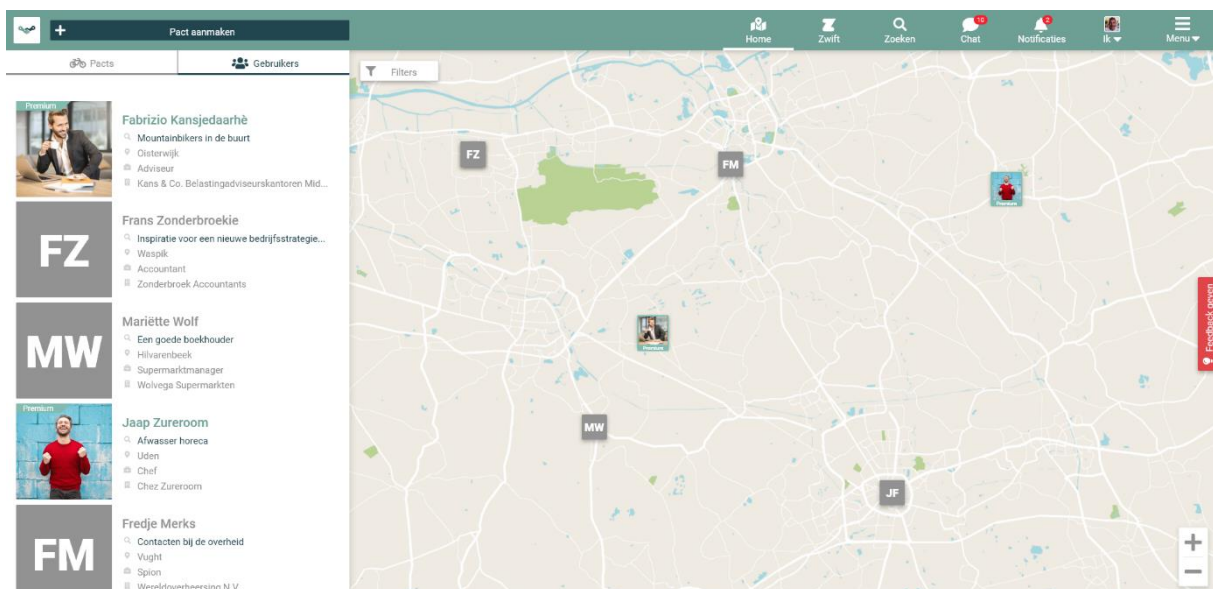
Figuur 26: Premium pact

### Dashboard overzicht

Een dashboard waar de gebruiker een overzicht ziet van de activiteiten van connecties, het totaal aantal meegereden pacts, georganiseerde pacts en interessante pacts in de buurt. Dit dashboard kan worden uitgebreid met meer informatie wanneer de applicatie meer functionaliteiten krijgt.

### Opvallendere Premium gebruikers

Een premium gebruiker kan meer opvallen op de kaart, dit zien gratis gebruikers ook en kunnen daardoor overtuigd worden om premium te halen. Mijn nieuwe ontwerp voor opvallendere premium gebruikers is te zien in figuur 27:



Figuur 27: Resultaat opvallendere premium gebruikers

### Pacts kopellen in de agenda

Premium gebruikers kunnen pacts opslaan in hun agenda naar keuze, op deze manier kun je pacts niet vergeten.

### Groepen aanmaken

Als premium gebruiker ben je een actieve gebruiker van het platform, je hebt waarschijnlijk ook een vaste groep aan mensen met wie je fietst. Premium gebruikers kunnen een groep aanmaken om deze in één keer uit te kunnen nodigen voor pacts.

### 6.3 Welke andere bronnen van inkomsten kunnen er komen?

Om de toekomst van het platform zekerder te stellen moet er ook worden gekeken naar andere inkomst bronnen, naast premium gebruikers. De actieve gebruikersgroep is nu nog te klein om alleen afhankelijk te zijn van het freemium model.

Ik ga een aanbeveling schrijven voor suggesties van andere inkomstbronnen. Ik ben tot deze alternatieve inkomstbronnen gekomen door naar andere gratis applicaties te kijken en te kijken wat zij doen om inkomsten te generen.

Ik heb ook nagedacht over de gebruikersgroep van Pedalpact.cc waar zakelijke gebruikers en enthousiaste fietsers de voornaamste groepen zijn. Waar heeft een zakelijke gebruiker en enthousiaste fietsers behoefte aan en zouden ze eventueel voor betalen?

#### **Bedrijf pacts**

Pacts die worden georganiseerd door bedrijven: het bedrijf betaalt een bedrag aan Pedalpact.cc om dit te mogen organiseren. Tijdens zo'n pact kan het bedrijf recruiters mee laten fietsen om medefietsers enthousiast te krijgen over hun bedrijf en daardoor misschien wel nieuwe werknemers binnen te halen. Voor een extra bedrag kan Pedalpact.cc dit pact ook nog delen op zijn sociale platformen.

#### **Workshop pacts**

Pedalpact.cc kan betaalde pacts organiseren, hier wordt dan een expert uitgenodigd om over een onderwerp te praten. Een gebruiker kan dan voor een bepaald bedrag zich inschrijven om mee te fietsen en te luisteren. Een premium gebruiker mag, of gratis, of met korting meefietsen. Pedalpact.cc kan ook nog voor een lunch zorgen waar de fietsers makkelijker vragen kunnen stellen aan de expert.

#### **Donaties**

Het kan zijn dat gebruikers het platform leuk vinden om te gebruiken maar premium te duur vinden. Een donatie kan dan een interessante optie zijn voor mensen die wel graag het platform willen steunen. Vanaf een bepaald bedrag krijgt een gebruiker een badge naast zijn naam, hoe hoger het bedrag hoe unieker de badge.



*Figuur 28: Donateur badge*

#### **Reclame**

Pedalpact.cc kan reclame banners in gaan zetten op de website voor gratis gebruikers en bezoekers. Premium gebruikers zien uiteraard geen reclames. De reclames kunnen via een extern bureau gebeuren of bedrijven kunnen betalen om hun pact te promoten op de website via reclame.

#### **Merchandise**

Pedalpact.cc kan fietskleding gaan verkopen, dit is ook goed voor reclame buitenaf. Het past ook goed bij het fiets model.

## 6.4 Reflectie

Ik vond het leuk om aan dit onderzoek te werken. Mijn mening over de functionaliteiten kwam zo goed als overeen met de product owner. Het ontwerpen van premium gebruikers vond ik ook heel leuk om te doen, denken over manieren om premium gebruikers meer uit de kaart te laten springen.

Het implementeren van premium was lastig, vooral het synchroniseren van de backend met Keycloak, zoiets had ik nog niet eerder gedaan. Aan de frontend was het implementeren wel te doen, het leek op het implementeren van de bezoeker rol.

Het freemium model wordt helaas niet geïmplementeerd in de release versie. De gebruikersactiviteit van Pedalpact.cc valt tegen waardoor ze dit nog niet willen implementeren dit jaar. De implementatie voor premium staat wel volledig klaar en kan door één klik op een knop worden geïmplementeerd.

## 7. Nieuwe onboarding flows voor verschillende type gebruikers en een product tour

Pedalpact.cc heeft nu ondersteuning voor drie verschillende type gebruikers: bezoekers, freemium gebruikers en premium gebruikers. Er is nu echter alleen een onboarding flow voor de normale, nu freemium, gebruiker. Er moet dus een vernieuwde onboarding flow komen voor de freemium en premium gebruikers. Om de applicatie toegankelijker te maken moet er ook een product tour komen. Dit is een tour die uitlegt hoe de verschillende onderdelen van de website werken, bijvoorbeeld een pact aanmaken. Een tour maakt het makkelijker voor nieuwe gebruikers om de applicatie te gebruiken.

Voor het onderzoek ben ik eerst de oude onboarding flow gaan bekijken, hier bekeek ik wat er aan verbeterd kon worden. Ik kwam erachter dat de oude onboarding flow overweldigend is en niet voldoet aan de principes van de Heuristic Abilities (Nielsen, 1994). Ik besloot het onderdeel waar de gebruiker zijn locatie moet kiezen los te trekken van de huidige onboarding flow. Een gebruiker hoeft namelijk alleen zijn locatie in te vullen om de applicatie te kunnen gebruiken. Hier maakte ik een ontwerp van voor de desktop en mobiele versie, deze liet ik nakijken door de UX-designer. De feedback heb ik verwerkt in een nieuw ontwerp en deze heb ik geïmplementeerd in de website.

Omdat er nu een premium model is heb ik ook een ontwerp gemaakt voor een premium info pagina. Dit is een pagina waar alle functionaliteiten op staan die een premium gebruiker heeft. De gebruiker kan dan kiezen of die een abonnement wilt afsluiten, een proefabonnement wilt starten of dit wil overslaan.

De huidige onboarding pagina blijft bestaan maar zal anders zijn voor de freemium en premium gebruiker. Een freemium gebruiker kan niet alle velden invullen die een premium gebruiker wel mag, dit heb ik uitgewerkt in een tabel en geïmplementeerd in de website.

Hierna ben ik bezig gegaan met de product tour implementeren. Ik gebruik React Joyride om dit te doen. Ik ging eerst onderzoeken hoe React Joyride werkt door de documentatie op hun [website](#) te bestuderen. Nadat ik wist hoe React Joyride werkte ging ik kiezen welke pagina's van de website allemaal product tours vereisten. De belangrijkste pagina's voor uitleg zijn pagina's waar de gebruiker een actie uit kan voeren, als voorbeeld: de pact pagina, een uitleg hoe je aan een pact mee kan doen. Per pagina ga ik de acties af die de gebruiker uit kan voeren en bepalen of hier een uitleg voor nodig is. De afweging voor de keuze is of de actie duidelijk is uitgelegd op de pagina zelf, als dit niet het geval is helpt een product tour hierbij. Deze acties zijn te lezen in de tabel die te vinden is in het onderzoeksrapport (zie bijlage; Onderzoeksrapport.pdf)

Toen ik erover uit was voor welke pagina's ik een product tour wilde maken en voor welke acties een uitleg nodig was ben ik de product tour gaan implementeren. De implementatie is in het kopje [7.1 Implementatie](#) te lezen.

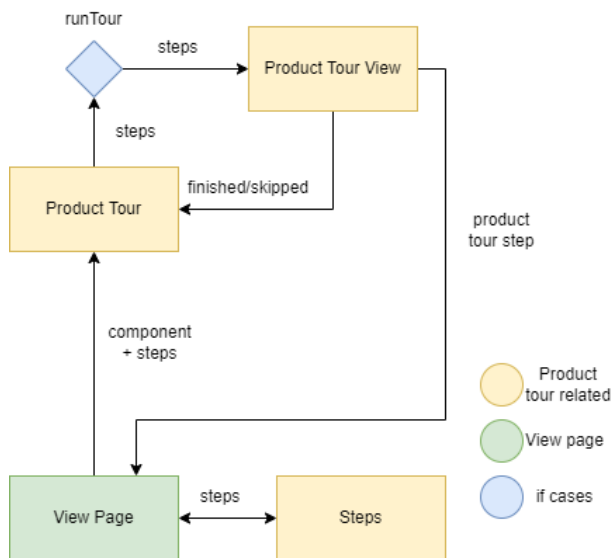


## 7.1 Implementatie

Ik heb een nieuw component genaamd: ProductTour aangemaakt. Dit component staat bovenaan elke view die gebruik maakt van de product tour. De component vereist de component waar deze tour heen verwijst en de stappen die hierbij horen. De stappen worden in de folder van de component opgeslagen, bijvoorbeeld: new-pact, hier staat een bestand genaamd "new-pact.steps". De component is nodig om bij te houden welke tours de gebruiker al heeft afgelegd zodat deze niet na het voltooien opnieuw worden getoond. Welke tours de gebruiker al heeft afgelegd worden bijgehouden in de backend, deze zijn op te halen door de volgende call:

<b>GET/SET</b>	api/usersettings/producttour	<ul style="list-style-type: none"><li>- Show</li><li>- Home</li><li>- Profile</li><li>- Pact</li><li>- Me</li><li>- newPact</li></ul>
----------------	------------------------------	---

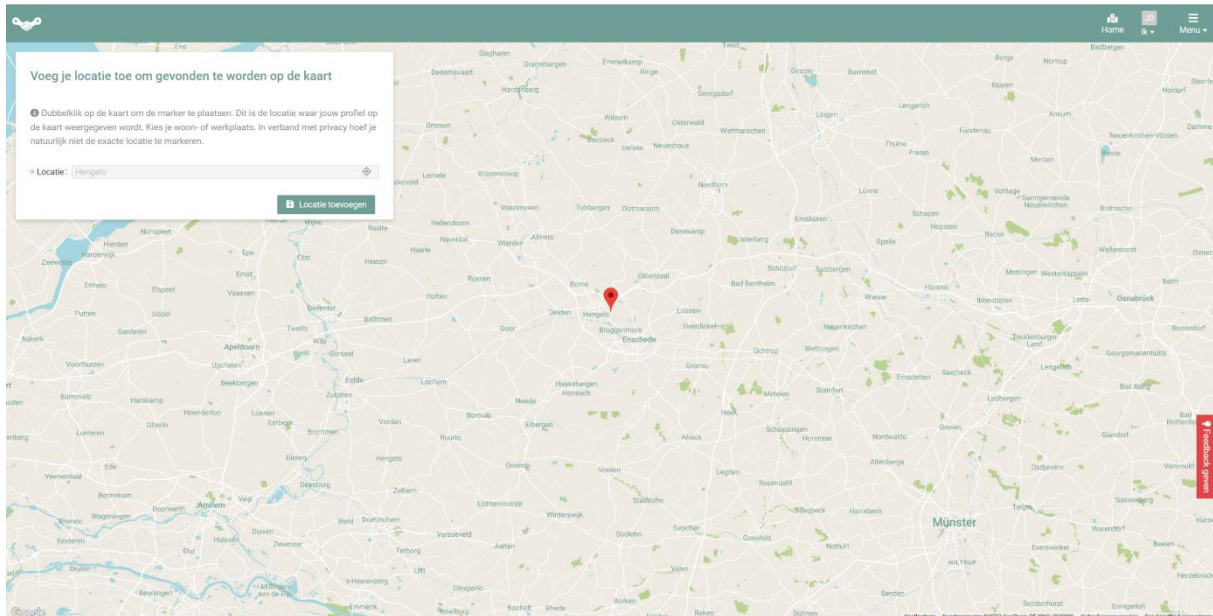
De hierboven beschreven uitleg is te zien in figuur 26.



Figuur 29: Product tour

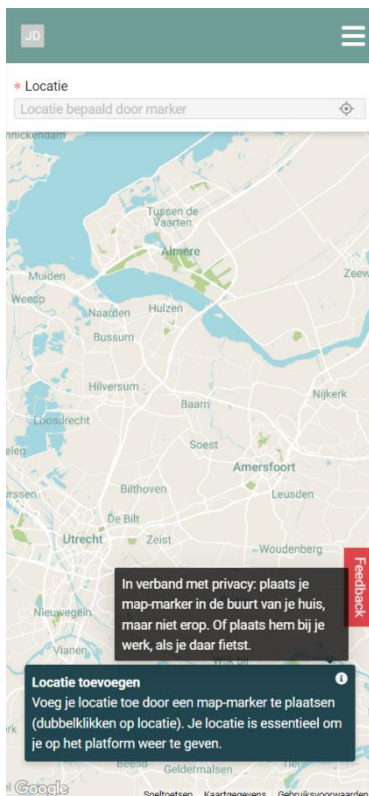
## 7.2 Uiteindelijke resultaat

Voor het uiteindelijke resultaat van de desktop versie van de locatie kies pagina is niet veel veranderd in vergelijking met het ontwerp. De pijl met de tekst “volgende stap” is vervangen door een knop die eerst is uitgeschakeld totdat de gebruiker een locatie kiest. Het resultaat is te zien in figuur 27.



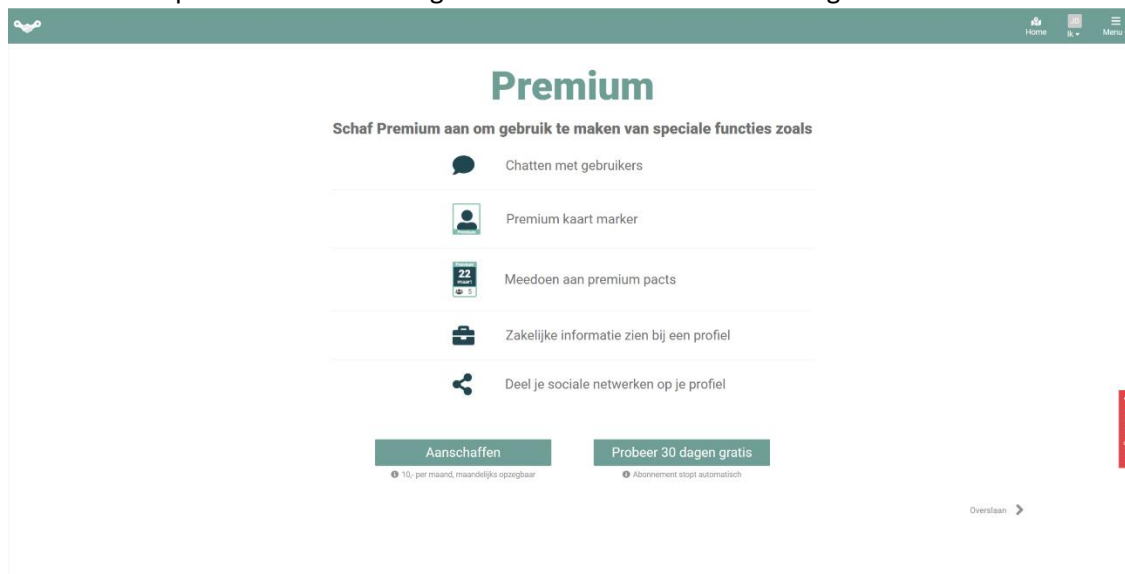
Figuur 30: Uiteindelijk resultaat locatie kiezen desktop

Voor de mobiele versie is er niks veranderd in vergelijking met het ontwerp. Het resultaat is te zien in figuur 28.

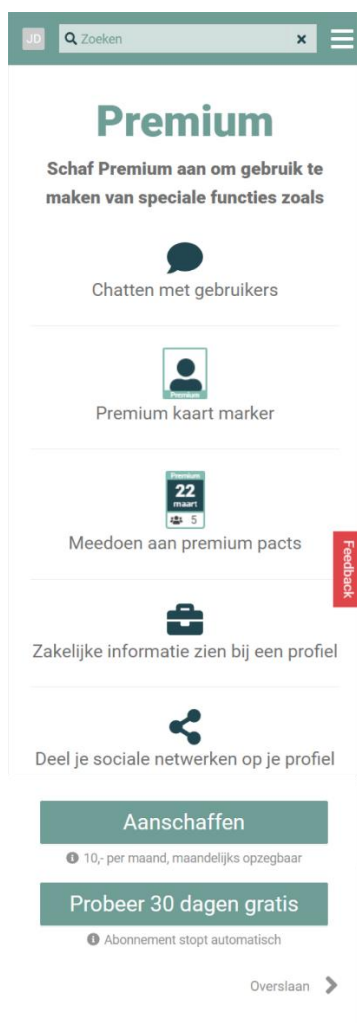


Figuur 31: Locatie kiezen mobiel

Voor de premium info pagina is de pagina precies hetzelfde gebleven als het ontwerp. Het resultaat van de desktop versie is te zien in figuur 29 en de mobiele versie in figuur 30.



Figuur 32: Premium info desktop



Figuur 33: Premium info mobiel

### 7.3 Testen

Om er achter te komen of het vernieuwde design te begrijpen is voor nieuwe gebruikers heb ik een aantal testen geschreven. Deze testen zijn ontworpen om de doelstellingen van het los maken van pacts en gebruikers te evalueren vanuit het gebruikersperspectief. De volgende testen zijn hiervoor geschreven:

- Ga naar de locatie kies pagina, is het duidelijk wat je moet doen?
- Selecteer je locatie door je huidige locatie te gebruiken.
- Ga opnieuw naar de pagina en kies nu je locatie via de kaart.
- Ga naar de premium info pagina, is het duidelijk wat premium allemaal biedt voor opties?
- Worden er opties besproken die niet duidelijk zijn?
- Wordt je enthousiast om premium aan te schaffen door deze informatie pagina? Waarom wel/niet?
- Als je niet overtuigd genoeg bent om premium aan te schaffen, zou u dan kiezen om de proefperiode te starten? Waarom wel/niet?
- Ziet u ook een mogelijkheid om beide opties niet te doen en verder te gaan?

De antwoorden van de personen staan hieronder vermeld.

#### **Persoon 1:**

- Ik denk dat ik hier mijn locatie moet toevoegen? Ik denk dat ik in het tekstveld locatie mijn locatie moet invullen. Ik lees nu dat je een marker plaatsen, nu ik de tekst heb gelezen is het duidelijk.
- Ik zou mijn huis zoeken op de kaart maar dat is niet heel gebruiksvriendelijk. Ik zie geen knop om mijn huidige locatie te gebruiken, naar wat zoeken zag hij de knop naast het tekstblok. Hij moet rood worden om duidelijk te maken dat deze klik baar is.
- Ik zou de kaart verplaatsen naar mijn woonplaats en inzoomen naar mijn huis en daar dubbel op klikken.
- Ja het staat duidelijk aangegeven
- Ik snap niet precies wat er met zakelijke informatie wordt bedoeld bij het profiel.
- Ja, chatten met gebruikers lijkt me leuk en meedoen aan premium pacts.
- Ik haat het als ik eerst mijn bankgegevens in moet vullen voordat ik een proefperiode zou kunnen starten. Als ik zou kunnen klikken op de proefperiode starten zonder dat ik verder gegevens hoeft in te vullen zou ik het doen.
- De overslaan knop indrukken. Ik zou deze wel groter maken want hij valt nu niet echt op.

**Persoon 2:**

- Ja, deze pagina dient ervoor om mijn locatie toe te voegen aan mijn account zodat ik gevonden worden door andere mensen.
- Ik klik op de “ga naar huidige locatie” knop.
- Ik sleep de kaart naar mijn locatie, ik dubbelklik op de kaart zoals de beschrijving beschreef en druk op de knop “locatie toevoegen”
- Ja het is duidelijk wat de opties inhouden
- Als ik denk aan zakelijke informatie denk ik aan Whatsapp of LinkedIn.
- Nee, deze informatie lijkt voor mij niet nodig om de applicatie te kunnen gebruiken. Het enige wat mij interessant lijkt is chatten met gebruikers.
- Zou ik wel doen omdat er staat dat het abonnement automatisch stopt. Hierdoor hoef ik mij geen zorgen te maken om het abonnement stop te zetten.
- Ja, de knop overslaan. De knop valt genoeg op.

**Persoon 3:**

- Ja, de uitleg is helder. Ik zou toevoegen dat je een adres kan intypen.
- Ik klik op het target icoon in het tekstblok.
- Ik sleep de kaart naar mijn locatie en dubbel klik waar ik de marker wil neerzetten.
- Ja behalve de zakelijke informatie bij een profiel.
- De zakelijke informatie bij een profiel.
- Als de applicatie veel werd gebruikt zou ik het overwegen
- Ja, ik zou het wel beëindigen naar de proefperiode behalve als ik het veel gebruik.
- Ja, de knop overslaan rechts onderin.

**Resultaat:**

- Voor alle personen was het niet duidelijk wat zakelijke informatie inhield.
- Bij de proefperiode starten duidelijk neerzetten of bankgegevens nodig zijn.

## 7.4 Reflectie

Het ontwerpen van de locatie kies pagina en premium info pagina was uitdagend, dit waren compleet nieuwe ontwerpen in plaats van een aanpassing op een oude pagina. Omdat het nieuwe ontwerpen waren vond ik het wel heel leuk om te doen omdat ik mijn eigen visie hierin kon verwerken. Ik vond het moeilijk om te bepalen hoe de pagina opgebouwd moest worden, gelukkig hielpen de principes van de Heuristic Abilities (Nielsen, 1994) hier mee.

De product tours waren ingewikkeld om te maken, ik had nog niet eerder gewerkt met de library React Joyride. Gelukkig was de documentatie helder. Door de product tours is het makkelijker geworden voor nieuwe gebruikers om de applicatie te begrijpen en te gebruiken.

Door de nieuwe locatie kies pagina is het proces om een account te maken versoepeld. De gebruiker wordt nu niet meer overspoeld met velden dat die in moet vullen.

Ik hoop dat de premium info pagina verheldering biedt aan gebruikers wat premium inhoudt en mensen overhaalt om premium aan te schaffen.

## 8. Conclusie

Pedalpact.cc is een zakelijke wiel community die in samenwerking met Carthago is gebouwd. Het doel van de app is om op een leuke en actieve manier afspraken te maken met interessante zakelijke contacten, door middel van fietsen.

Op dit moment is er geen mogelijkheid om voor deze service te betalen, dit is echter wel nodig om de applicatie te kunnen blijven onderhouden en uit te breiden met nieuwe functionaliteiten.

Als hoofdvraag is daarom genomen: hoe kan Pedalpact.cc een freemium model toevoegen aan de applicatie?

De deelvragen die nodig zijn om deze vraag te beantwoorden zijn:

### **Welke gebruikers en pact informatie mag er worden getoond op de openbare pact- en kaart pagina?**

De deelvraag is beantwoord door te onderzoeken welke onderdelen privacy gevoelige informatie bevatten. Het antwoord hiervan is geïmplementeerd in de openbaren pact- en kaart pagina.

### **Hoe kan de kaart interface worden aangepast om het zoeken naar fietsritten makkelijker te maken en deze ook meer op te laten vallen?**

De deelvraag is beantwoord door een think-aloud onderzoek uit te voeren en de beste ideeën hiervan uit te werken in een ontwerp. Deze ontwerpen zijn samengevoegd tot één ontwerp en geïmplementeerd.

### **Hoe kan de frontend en backend automatisch worden getest om bestaande en nieuwe functionaliteit te waarborgen?**

De oplossing hiervoor is eerst onderzoeken welke onderdelen van de applicatie getest moeten worden en welke test-frameworks hiervoor beschikbaar zijn. Er is gekozen om voor de backend JUnit te gebruiken en voor de frontend Cypress. De testen draaien automatisch in de CI/CD.

### **Hoe kunnen de nieuwe gebruikersvormen worden ondersteund met de huidige opzet van de applicatie?**

Het antwoord hiervoor is om aan het datamodel van de gebruiker "isPremium" toe te voegen. In de frontend worden voor speciale premium pagina's deze column opgevraagd om te weten of de gebruiker hier bij mag. Welke functionaliteiten afgeschermd moeten worden is bepaald door een onderzoek. Om de gebruiker op de hoogte te stellen van de premium functionaliteiten is er een premium informatie pagina toegevoegd.

### **Hoe moet de vernieuwde onboarding flow worden van een gratis gebruiker en premium gebruiker en hoe kan er een product tour worden geïmplementeerd?**

Het antwoord is om het locatie kiezen onderdeel van de onboarding los te tonen van de onboarding flow, nu kan de gebruiker sneller gebruik maken van de applicatie. Voor de product tour is er gekozen om React-Joyride te gebruiken, dit is een product tour library. Uit een onderzoek is eruit gekomen welke pagina's een product tour vereisten en deze is daar geïmplementeerd.

Uiteindelijk zijn alle deelvragen beantwoord en is de hoofdvraag hiermee ook beantwoord en geïmplementeerd in de applicatie.

## 9. Aanbevelingen

Hieronder heb ik een aantal aanbeveling punten geschreven voor Pedalpact.cc die in mijn optiek nodig zijn om van Pedalpact.cc een succes te maken.

### **Innovatie**

Pedalpact.cc is een leuk concept, er zit zeker potentie in het idee. Er zijn alleen teveel vergelijkbare applicaties die al grote gebruikersgroepen hebben. Pedalpact.cc moet meer innovatieve functionaliteiten krijgen waardoor ze unieker in het fietssegment worden geplaatst. Een voorbeeld kan zijn smart functionaliteiten waarmee je smartwatch gekoppeld kan worden aan de applicatie zodat je je vooruitgang bij kan houden. Een ander voorbeeld is wedstrijden tussen de gebruikers wie de meeste kilometers aan pacts mee kan fietsen in een maand. Dit soort evenementen en functies zorgen ervoor dat gebruikers terug blijven komen op de website ook al zijn er weinig pacts beschikbaar bij hen in de buurt.

### **Advertenties**

Pedalpact.cc heeft een te kleine actieve gebruikersgroep om de kern van het idee uit te kunnen voeren, er zijn te weinig fietsers voor pacts. De mensen die meedoen aan pacts zijn nu steeds dezelfde actieve gebruikers waardoor het netwerk aspect afzwakt. Ik denk dat reclame maken bij bijvoorbeeld fietsevenementen een grote impact zal hebben op de groei van de applicatie.

### **Mobiele applicatie**

Pedalpact.cc heeft nu een mobiele applicatie dat de website toont in de vorm van een Android-applicatie. Deze oplossing is niet erg gebruiksvriendelijk. Als er veel tijd vrij komt zou ik de applicatie omschrijven naar een native applicatie. Bij een native applicatie kun je de native functies gebruiken van het apparaat, notificaties zouden beter werken, de lay out en functies in de applicatie zouden beter werken. Het ombouwen vergt wel veel tijd maar zal in de toekomst wel nodig zijn als de gebruikersgroep toeneemt.

### **Test driven development**

De applicatie wordt nu geschreven zonder dat de functionaliteiten getest worden. Ik heb met mijn onderzoek voor een groot gedeelte van de applicatie testen geschreven maar ik denk dat het belangrijk is dat voor iedere nieuwe functie die wordt toegevoegd testen geschreven worden. Een groot gedeelte van de ontwikkeltijd is nu gebruikt om bugs op te lossen die met testen meteen boven water zouden zijn gekomen.



## 10. Reflectie

De afstudeeropdracht verliep hobbelig. Het begon eigenlijk bij de start, de opdracht was opgesteld door iemand die niet betrokken is bij Pedalpact.cc. Hierdoor kwam ik er tijdens de eerste weekly's al achter dat de belangen van de product owner heel anders waren dan dat er in mijn opdracht was beschreven. Door deze miscommunicatie zijn veel van mijn eerder opgestelde onderzoeksvragen gewijzigd naar nieuwe die meer in de visie paste van de product owner.

Tijdens het begin van het ontwikkelen was ik overweldigd door de applicatie. De structuur van de app is heel anders dan hoe ik het normaal zou maken voor een project. Na een tijdje bezig te zijn geweest met ontwikkelen wende dit wel en kon gelukkig gemakkelijk hulp vragen van mijn bedrijfsbegeleider en de opdrachtgever, Dennis.

Er stond best veel druk achter Pedalpact.cc omdat er nog veel moest gebeuren en er weinig tijd was tot het "wielerseizoen" begon, dit begint vanaf april en loopt tot ongeveer september. Ik vond het moeilijk om hiermee om te gaan omdat ik mij probeerde te focussen om mijn afstudeeropdracht te maken. Gelukkig werden er nieuwe mensen op het project gezet die bezig gingen met features bouwen die los stonden van mijn opdracht.

Ik ben ook veel bezig geweest met bugs fixen, deze hadden prioriteit omdat anders de nieuwe versie van de applicatie niet uitgebracht kon worden. Deze bugs hadden niet te maken met mijn onderdelen van de code maar ik loste ze aan het begin wel elke keer op omdat ik het ook belangrijk vond dat er een nieuwe versie uitgebracht kon worden. Op een gegeven moment was ik te veel bezig met bugs oplossen en kleine nieuwe features in mijn opdrachten verwerken die Daan later bedacht dan door te kunnen gaan met een nieuwe onderzoeksvraag. Dit gaf ik aan bij mijn bedrijfsbegeleider en die zei dat ik me moet focussen op mijn afstudeeropdracht. Hierna ging het beter met mijn voortgang en kon ik mij weer beter focussen op mijn opdracht.

Ik ben tevreden met het eindresultaat, ik heb alle onderdelen die ik wilde bouwen kunnen implementeren. De product owner is ook blij met de functionaliteiten die ik heb ingebouwd in de applicatie. De onderdelen helpen de toegankelijkheid en hopelijk ook de populariteit van de website te verbeteren.

## Bronnen

Lytvynenko, D. (z.d.). *Backend testing: the definitive guide* — *MindK Blog*. MindK. Geraadpleegd op 3 maart 2022, van <https://www.mindk.com/blog/backend-testing/>

Hamilton, T. (2022, 16 april). Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE. Guru99. Geraadpleegd op 3 maart 2022, van <https://www.guru99.com/unit-testing-guide.html>

vanderwardt@agilescrumgroup.nl. (2020, 24 december). *Wat is Kanban? Een volledige uitleg van de Kanban Methode + template*. Agile Scrum Group. Geraadpleegd op 17 februari 2022, van <https://agilescrumgroup.nl/wat-is-kanban-methode/>

Nielsen, J. (1994, 24 april). *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group. Geraadpleegd op 8 juni 2022, van <https://www.nngroup.com/articles/ten-usability-heuristics/>

Gokce, S. (2022, 6 juni). *Top 5 React Onboarding Libraries for Product Tours and Walkthroughs*. UserGuiding. Geraadpleegd op 9 juni 2022, van <https://userguiding.com/blog/react-onboarding-tour/>