Europcar

Europcar Self Service Check-in

Opdrachtbeschrijving

Europcar is aan het nadenken over de mogelijkheid om mensen de optie te bieden de gehele check-in procedure zelf te doen (borg, verificatie etc) zodat ze alleen maar bij het ophalen van een auto een code hoeven te scannen en dan de sleutels krijgen,

dat betekent dus niet meer wachten aan de balie maar snel de auto op kunnen halen.

User stories / functionaliteiten

Jullie gaan aan de slag met de userflow van een gebruiker die begint bij de verificatie procedure tot en met de gereserveerde auto ophalen. De gebruiker heeft hierbij al een reservering geplaatst en kan nu zelfstandig de check-in procedure doorlopen. Hierbij horen in ieder geval:

- ID verificatie
- Borg betalen
- Sleutel ontvangen

Data en/of content

Voor de gevraagde functionaliteit is nog geen API ontwikkeld. Als er behoefte is aan informatie (bijv over lokaties en type auto's e.d.) dan kunnen we die aanleveren.

Werkwijze en planning

In de eerste week van de meesterproef is een virtuele briefing door Q42. Wekelijks wordt de voortgang getoond en de volgende stappen besproken. Bij sommige voortgangsgesprekken zal de klant, Europcar, ook aanhaken.

Briefing:
Maandag 17 mei 16:00

Contactpersoon

Sanne Swagerman, sanne@q42.nl, 06 3300 9091

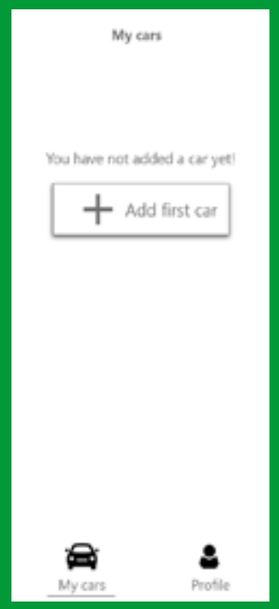
Leerdoelen

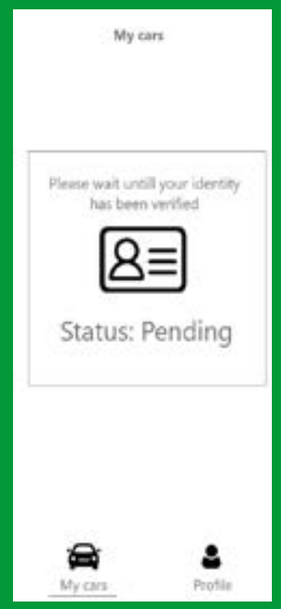
In dit project wil ik graag een volledig werkend eindproduct maken, waar zowel de ik als de opdrachtgever blij mee zijn. Daarbij wil ik aandacht besteden aan de volgende zaken:

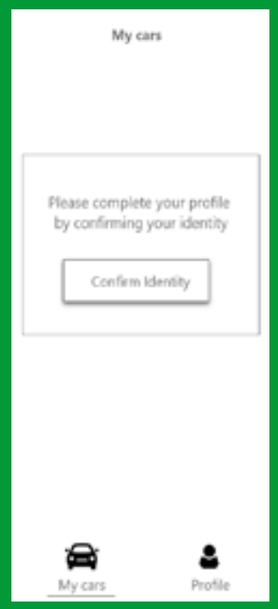
- Nette code, server en client side.
- Juiste documentatie
- Passend vooronderzoek
- Aandacht besteden aan UX en UI
- Een goede planning aanhouden voor mijzelf, waaronder: Uit-schrijven van sprints, bijhouden in bijvoorbeeld een Trello bord

Ontwerp V1


Europcar
Email
Password
Login
Need an account?
Register

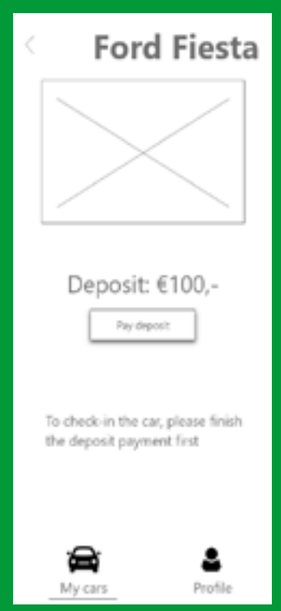

My cars
You have not added a car yet!
Add first car
My cars
Profile

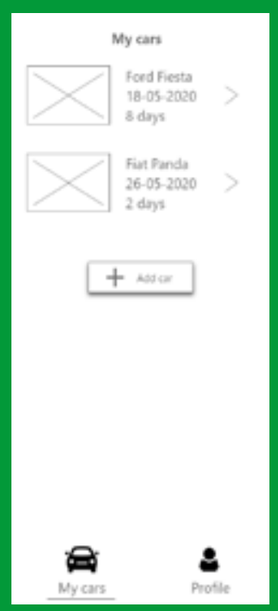

My cars
Please wait untill your identity has been verified
Status: Pending
My cars
Profile

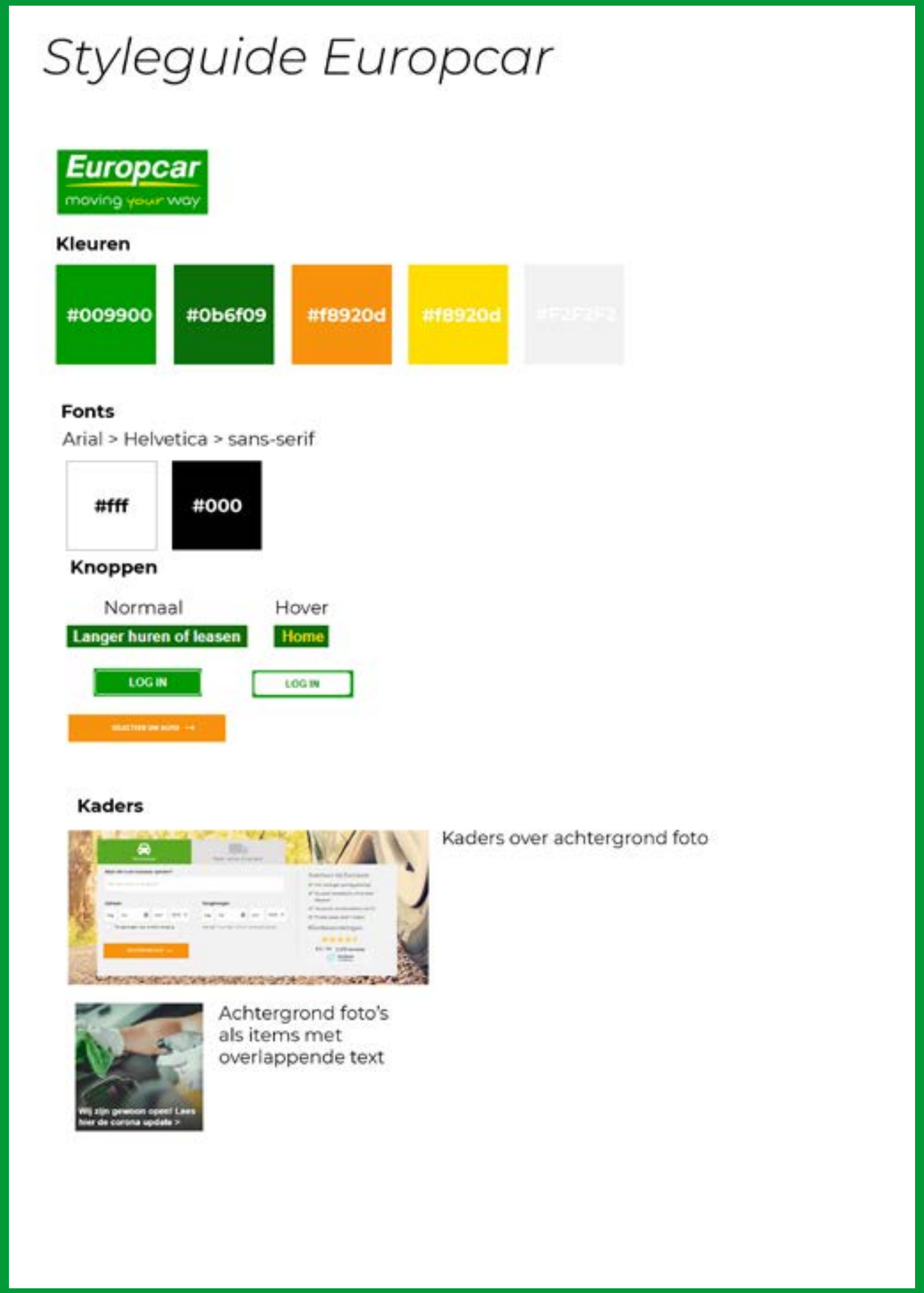

My cars
Please complete your profile by confirming your identity
Confirm Identity
My cars
Profile


Ford Fiesta
Deposit: €100,-
Paid
My cars
Profile


Ford Fiesta
Deposit: €100,-
Paid
My cars
Profile


Ford Fiesta
Deposit: €100,-
Pay deposit
To check-in the car, please finish the deposit payment first
My cars
Profile


My cars
Ford Fiesta
18-05-2020
8 days
Fiat Panda
26-05-2020
2 days
Add car
My cars
Profile



Meeting 1

Vragen en antwoorden

Europcar (introductie)
Om wat voor type voertuigen gaat het? (personenauto's, busjes, vrachtwagen)
Kunnen gebruikers meerdere auto's huren op 1 bestelling?
Wat is het verschil tussen de Nederlandse Europcar website, en die van andere landen in europa?

EuropCar.nl is een Nederlandse franchise onderneming, auto's kunnen per stuk worden gereserveerd, sommige klanten hebben een abonnement bij EuropCar om hier voordelig mee uit te komen (zij huren vaak wekelijks een auto).

Klant/Doelgroep:
Wat is de doelgroep? (mensen uit NL, toeristen)

Nederlandse gebruikers.

EuropCar technieken:
In hoeverre werken wij het inlog en betaalproces uit?
Welke betaalmiddelen zijn mogelijk, en hoe wordt dit geregeld? (Volgens website alleen creditcard?)
Wat houd de ID verificatie in? (wat voor documenten, hoe toevoegen)
Hoe authenticeren we gebruikers (reserveringsnummer/kenteken/inloggen), en hoe verifiëren we de identiteitsdocumenten?
Hoe linken we auto's aan gebruikers indien inloggen?

Authenticatie: gebruikersaccount d.m.v. login. Het aanmaken van een account is voor deze case overbodig. Dit wordt door EuropCar opgepakt. Het valideren van een rijbewijs is cruciaal, het is van belang om dit op een veilige wijze op te slaan. Denk hierbij ook in welke fase van het proces dit wordt gevraagd & gevalideerd. Het is expliciet valideren van een rijbewijs. Idin is een tool die hier gebruikt voor kan worden

Criteria:
Rijbewijs van belang
Minimale leeftijd: 21, wordt 23
Vrachtwagens > vrachtwagen-rijbewijs
Specifieke voertuigen (vaak duurdere auto's) gebaseerd op leeftijd

Er wordt geboekt per klasse, reserveringsnummer is hier van belang.

Logistieke vragen:
Waar ontvangen mensen een sleutel? Is dit op alle locaties op dezelfde manier te herkennen? Is het beschikbaar op alle locaties?
Tot hoe ver van te voren is het mogelijk om aan te melden?
Moet er een huurovereenkomst worden getekend?
Wat als er meerdere bestuurders zijn?
Kunnen we iets bedenken om het contactmoment bij het halen van de sleutels over te slaan?

Huidige situatie:
Bij balie melden voor check-in.

Toekomst:
Uitdenken hoe je contactloos sleutel wordt opgehaald. Reserveren en betalen gebeurd online, fysiek enkel reservering controleren en sleutel afgeven. Het scannen van een QR code kan hier een oplossing voor bieden misschien.

Gebruikers moeten binnen een bepaalde periode zich in-checken. Zo is bijvoorbeeld 2 maanden van te voren te lang, en zitten ze eerder te denken aan rond 1 a 2 weken.

Fysiek de huurovereenkomst ondertekenen is op dit moment nog de way to go.

Hoofdbestuurder moet worden geverifieerd. Ook de extra bestuurders moeten worden gecontroleerd worden op leeftijd en rijbewijs.

Algemeen:
Wanneer houden we onze wekelijkse voortgangsgesprekken?
Voorkeur voor platform en tech stack?
Ontwerpen we voor mobiel, desktop, of complete plaatje?
Houden we de styleguide van Europcar aan, of kunnen jullie de nieuwe styleguide misschien delen?
Ontwerpen we ook een dashboard voor Europcar, waar de reserveringen binnen komen?

Idealiter wordt het geïntegreerd in de website. Een responsive variant is hiervan dus vereist.

De styleguide is gemaakt door Frabicque, deze wordt met ons gedeeld zodat we in dezelfde stijl kunnen doorwerken.

Tech-stack: Vrije keuze hierin. Huidige stack: ASP.net en Vue.

Borg reserveren op creditcard in de check-in.

Aantekeningen

Europcar wil self check-in aanbieden. Mogelijk een paal waarbij je iets kunt scannen en vervolgens een sleutel krijgt en vervolgens kunt wegrijden. Q42 is bezig met het aanpassen van de website van europcar.

1. Europcar nederland is een franchise van het Franse moederbedrijf. bieden allerlei soorten auto's aan, busjes en vrachtauto's. Meerdere auto's in een keer is niet mogelijk. Er zijn wel mensen die heel vaak auto's huren, bijvoorbeeld 100x per jaar (ongeveer 1-2 keer per week). Kan via een vast contract. Het boeken van huurauto's voor het buitenland verloopt via de internationale website.

2, Buitenlandse boekingen komen binnen via de internationale site. De reservering wordt afgehandeld door de nederlandse vestiging. Voor nu de doelgroep binnen Nederland houden.

3, Voor deze opdracht is het goed om er vanuit te gaan dat mensen een account hebben. Het inlog en registratieproces valt buiten de scope van de opdracht.

4. Het controleproces is een van de belangrijkste stukken van de case, hoe haal je dit op en hoe ga je deze informatie opslaan? Verschil tussen wie je bent, welke leeftijd (minimaal 21, wordt 23) en dat je ook daadwerkelijk een rijbewijs hebt. Voor het huren van een vrachtwagen heb je een aanvullend rijbewijs nodig. Bij bepaalde type voertuigen gelden andere leeftijdseisen. Verificatie kan wellicht via een externe dienst als Idin.

5. Gebruikers hebben een reserveringsnummer. De implementatie die nu wordt gemaakt bewaard automatisch de auto's die je hebt gehuurd in je account.

6. Het proces gebeurt nu nog aan de balie. Er is nog niet goed uitgedacht hoe het proces van check-in gaat verlopen en wat de vorm daarvan is. Wordt mogelijk een persoonlijk proces bij een apart loket voor de mensen die al ingecheckt zijn.

7. Wil liever niet dat mensen te ver van tevoren gaan inchecken. Bijvoorbeeld binnen 7 dagen zou kunnen.

8. Tekenen van een huurovereenkomst zal voor nu nog aan de balie worden gedaan. Gaat erom dat er al veel dingen kunnen worden ingevuld.

9. Er is een hoofdbestuurder. Als er andere bestuurders worden toegevoegd zullen ook die allemaal moeten worden gecheckt volgens hetzelfde proces.

10. Idealiter is het een integratie voor de website (web-only is prima dus).

11. Er is een stylesheet gemaakt die met ons kan worden gedeeld (kleuren, typografie).

12. Werken momenteel in .NET en Vue. Gaat voornamelijk om het concept, hoeven ons niet te beperken door de techniek.

13. Wordt nu gebruik gemaakt van worldline payments. Er wordt een reservering gemaakt op je credit card (geen geld afgeschreven) - dit is een aantal dagen geldig. Ook mogelijkheid toevoegen om de borg contant bij de balie af te rekenen.

14. Het controleproces zal voor alle reserveringen steeds opnieuw moeten worden uitgevoerd, maar wellicht kan dat in de toekomst worden verbeterd.

Debriefing

Probleemstelling

Het incheckproces van Europcar Nederland kan al gauw 15 minuten duren. Europcar wilt een mogelijkheid aanbieden om de gehele check-in zelf te regelen. Denk hierbij aan het verifiëren, valideren en betalen van de borg, zodat er op locatie enkel een code hoeft worden gescand, en de huurovereenkomst ondertekend wordt. De gebruiker kan vervolgens snel op pad.

Klantomschrijving

Europcar Nederland is een franchisenemer van Europcar International en is één van de grootste autoverhuurbedrijven van Nederland. Europcar voorziet in diverse mobiliteitsbehoeften van haar klanten. Zo kun je bij Europcar terecht voor het huren of leasen van auto's, zowel zakelijk als particulier. Het wagenpark van Europcar loopt uiteen, van personenauto's en bestelwagens tot verhuis- en vrachtauto's.

Ook biedt Europcar het GoforGreen label aan, waaronder uitsluitend hybride auto's vallen.

De klanten van Europcar zijn divers, zo zijn er onder meer gebruikers die sporadisch een auto huren, omdat ze deze nodig hebben, gebruikers die wekelijks of maandelijks een auto huren, of zakelijke bedrijven die een contract hebben met Europcar en op basis hiervan de auto's huren.

Opdracht omschrijving

Een gebruiker reserveert vaak ruim van te voren een auto om te huren. Het inchecken van de gebruiker gaat tot op heden altijd fysiek, op locatie bij Europcar. Dit kost al gauw zo'n 15 minuten per gebruiker.

Maak een digitaal incheckproces voor Europcar, zodat er weinig tot geen fysieke interactie nodig is voor het ophalen van de huurauto. Het incheckproces bestaat uit het inloggen van de gebruiker, het inchecken van de reservering, bevestigen van de identiteit en rijbewijs, en het reserveren van de borg op de desbetreffende creditcard.

Design challenge

Hoe kunnen wij ervoor zorgen dat de klanten die een auto hebben gereserveerd bij Europcar Nederland, zichzelf gemakkelijk en vermakelijk(?) online kunnen inchecken, zodat zij hun gereserveerde auto snel kunnen ophalen bij de desbetreffende locatie.

Stakeholders

EuropCar
Voor EuropCar is het van belang dat de applicatie de volgende problemen oplost:
Gebruiker moet gemakkelijk online kunnen inchecken
Gebruiker moet snel kunnen reserveren
Gebruiker moet reservering kunnen doorgeven/ laten scannen op locatie

Q42
Voor Q42 is het van belang om een doordachte applicatie uit te denken:
De applicatie moet een correcte userflow hebben bij het inchecken
De applicatie moet een verbeterde gebruikerservaring zijn
De applicatie moet verschillende edge-cases bevatten

Users
Voor de gebruikers is het van belang dat we een werkende applicatie ontwikkelen met de volgende functionaliteiten:

- Gebruikers moeten het proces overzichtelijk en gemakkelijk doorlopen
- Gebruikers moeten gemakkelijk online kunnen inchecken
- Gebruikers moeten hun rijbewijs online kunnen laten verifiëren
- Gebruikers moeten de borg online kunnen betalen/laten reserveren

Oplevering CMD
Design Rationale
Product biografie
Individuele reflectie op het project
Werkende tool/blije klanten

Plan van aanpak

Werkwijze
Ik wil gaan werke in kleine sprints. Alle stappen wil ik gaan opdelen in kleine requirements en features. Dan wil ik steeds aan 1 feature tegelijk werken en zo de lijst afwerken van belangrijk naar onbelangrijk.

Tools
Ik wil Trello gaan gebruiken om mijn sprints op te delen en voor mezelf duidelijk te krijgen. Dit zorgt voor structuur en duidelijkheid voor mezelf, maar ook voor de opdrachtgever zodat deze het proces zou kunnen volgen.

Trello bord:
https://trello.com/b/M7rEyn08/europcar-check-in

Planning

Week 1
- Eerste ontwerp maken
- Opzet server maken
- Simpele frontend features in code omzetten

Week 2
- Flow applicatie moet te doorlopen zijn
- Login feature maken
- Homepage maken

Week 3
- Id verificatie maken
- Borg betaling maken

Week 4
- Offline gebruik applicatie
- Repsonive maken website

Week 5
- UX interacties
- puntjes op de i

User scenario

Een Man wil met zijn vrouw drie dagen naar Brabant op vakantie vanuit Friesland, maar heeft zelf geen auto. Hij huurt een auto op Europcar.nl en maakt hier een account aan. Vervolgens opent hij de online-checkin van Europcar waar hij inlogt met dit account.

Na het inloggen komt hij op een overzichtspagina waar hij zijn reservering terug vindt. Op de detailpagina van zijn reservering kan hij een online checkin procedure beginnen vanaf 7 dagen voor aanvang.

Allereerst wordt er wat informatie gecontroleerd, daarna dient de man zijn rijbewijs te verifieren. Hij maakt hier een foto van, samen met een foto van zijn eigen gezicht ter controle.

Wanneer zijn identiteit gecontroleerd is kan de man de borg betalen doormidden van een creditcard, waarna de check-in compleet is.

Na de check-in krijg de man een QR code die hij bij het ophalen van de auto laat scannen, waarna de man de sleutels van de auto in ontvangst neemt.

Requirements

User requirements
- De gebruiker moet kunnen inloggen met een Europcar account.
- De gebruiker moet een overzicht van zijn reserveringen kunnen zien.
- De gebruiker moet online zijn reservering (auto) kunnen inchecken vanaf ongeveer 1 week voor aanvang.
- De gebruiker moet hier zijn rijbewijs kunnen opsturen van alle bestuurders om deze te verifiëren.
- De gebruiker moet zijn borg kunnen betalen.
- De gebruiker moet het inchecken op ieder moment kunnen annuleren om deze later af te maken.
- De gebruiker dient reeds afgeronde stappen niet opnieuw te hoeven voltooien.
- De gebruiker moet een QR code te zien krijgen om te kunnen tonen.

Technische requirements
- De app moet deels offline te gebruiken zijn (vooral QR code bekijken).
- De app moet responsive zijn vanaf mobiel tot desktop.
- De app dient aan te sluiten op de styleguide.

Styleguide V2

Europcar

Fabrique

Color styles

Text

- Aa Primary
- Secondary
- Disabled

Element

- Divider
- Input border
- Input border (hover)

Background

- Level 0
- Level 1
- Level 2

Tint

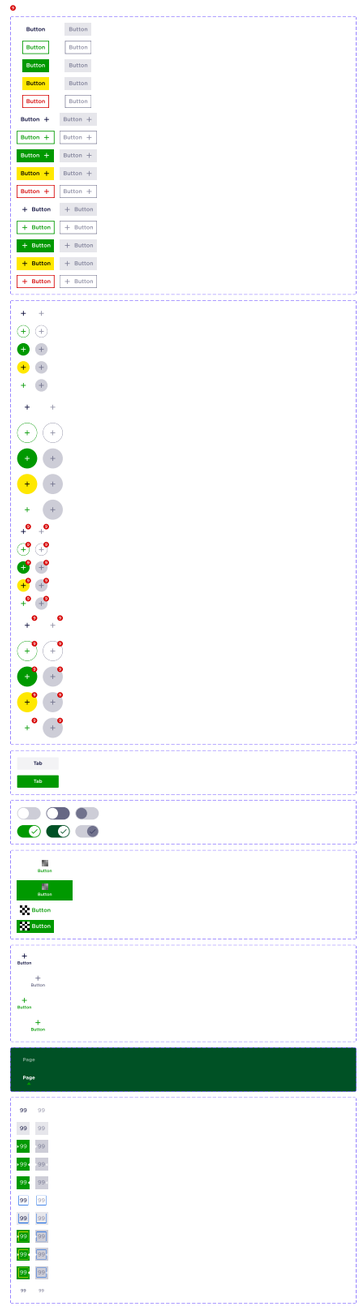
- Aa Secondary 500
- Aa Secondary 600
- Aa Secondary 700
- Aa Secondary 800
- Aa Primary
- Aa Accent
- Error
- Error (transparent)
- Warning
- Positive

Gradient

- Input mask

Dark	Light
5	5
10	10
50	50
100	100

Buttons



Code style

Code consistency

function naamgeving
functions krijgen namen die aangeven wat de functie doet.

Post requests zullen de actie als naam hebben, bijvoorbeeld:
checkin()
login()

Get request functies zullen de naam van de pagina krijgen, bijvoorbeeld:
overviewPage()
detailPage()

Util functies zullen aangeven wat de functie uitvoert, zoals:
findUser()
findCar()
checkLogin()

Hierbij zullen boolean returns een naam krijgen waarbij de vraag true of false kan zijn. Zoals:
userValidates()
userExists()

Code style

Om mijn code style te checken zal ik gebruik maken de npm package ESlint. Met de volgende style:
- Enkele quotes : ‘‘
- Geen semicolons: ‘;’

Code comments

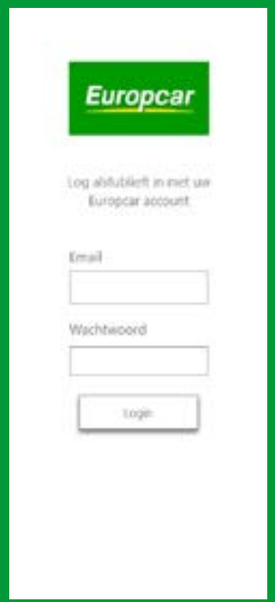
voor documentatie van mijn code ga ik gebruik maken van JSDocs, een teamgenoot heeft mij hieraan geïntroduceerd en het is mij opgevallen dat ik dit vaak tegen gekomen ben. JSdocs maakt gebruik van de volgende code comments structuur:

```
/**
 * Represents a book.
 * @constructor
 * @param {string} title - The title of the book.
 * @param {string} author - The author of the book.
 */
function Book(title, author) {
}
```

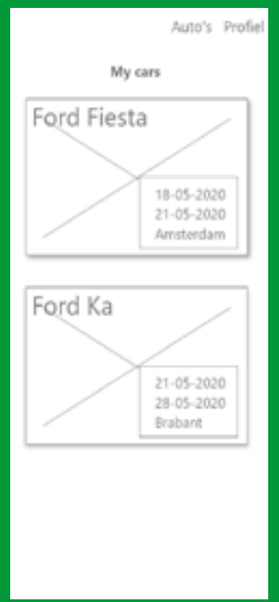
Alle functies die op deze manier van commentaar voorzien zijn kunnen uiteindelijk automatisch gedocumenteerd worden door het systeem van JSDocs, die maakt hier dan automatisch een documentatie boek van.

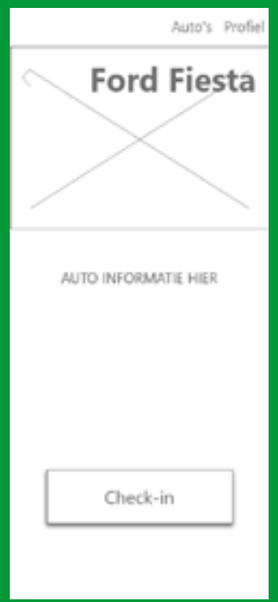
Naast de automatische documentatie functie, vind ik dat het voor een duidelijke code structuur zorgt die duidelijkheid bied over de werking van een functie en zijn parameters.

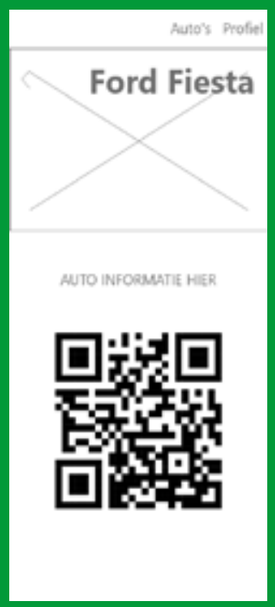
Mockup V2


Europcar
Log alstublieft in met uw Europcar account
Email
Wachtwoord
Login


Auto's Profiel
My cars
Je hebt momenteel geen auto reserveringen.
Maak reservering


Auto's Profiel
My cars
Ford Fiesta
18-05-2020
21-05-2020
Amsterdam
Ford Ka
21-05-2020
28-05-2020
Brabant


Auto's Profiel
Ford Fiesta
AUTO INFORMATIE HIER
Check-in


Auto's Profiel
Ford Fiesta
AUTO INFORMATIE HIER

Userdata

Om de applicatie te bouwen heb ik gebruikersdata nodig. Hiervoor heb ik een object gemaakt bestaande uit een logische inhoud van gebruikersdata, namelijk:
- Account gegevens (zoals inlog en naam)
- Reserveringen
- Reserveringsstatus (is borg betaald, is id reeds geverifieerd)

UserObject

```
{
      name: ‘Sam Slotemaker’,
      birthDate: ‘06-01-2000’,
      email: ‘sam2’,
      password: ‘password’,
      reservations: [
         {
            id: ‘090909’,
            car: ‘Ford’,
            model: ‘Ka’,
            from: ‘09-01-2020’,
            to: ‘12-01-2020’,
            imgUrl: ‘/style/images/fordKa.jpg’,
            checkedIn: false,
            drivers: [
               {
                  name: ‘Sam’,
                  validated: false
               },
            ],
            depositPayed: false
         }
      ]
}
```

Login systeem

Het login systeem maak ik werkend doormiddel van een simpele check of het ingevulde email gelijk is aan een van die van de gebruikers, en vervolgens wil ik kijken of het wachtwoord van deze gebruiker overeen komt.

Dit vergt de volgende zaken:
- Een hash package om wachtwoorden te encrtypen en te controleren (dit is nodig wanneer deze in een database staan)
- Een cookiepackage die de login status kan bijhouden
- Middleware die checkt of de gebruiker is ingelogd of niet, zo niet, zal deze moeten inloggen.

Passport
http://www.passportjs.org/

Met passport kan ik wachtwoorden encrypten en tegen elkaar checken.

Cookie-session
Met coockie-session kan ik een cookie maken voor de gebruiker en deze aanpassen wanneer er in en uitgelogd wordt. De cookie slaat ook het gebruikers-id op zodat ik hier later de reserveringen mee kan ophalen.

```
app.post('/login', (req, res) => {
    const user = findUser(req.body.email)

    if (userValidates(user, req.body.password)) {
        req.session.isLoggedIn = true;
        req.session.userID = user.email;
        res.redirect('/cars')

    }
    else {
        res.render('login.ejs', { title: 'login', error: true })
    }
})
```

Middleware functie
Met deze geschreven middleware functie kan ik op iedere get request checken of de gebruiker nog moet inloggen. Zo ja, dan verwijs ik deze door naar de inlog.

```
export function checkLogin(req, res, next) {
    if (req.session.isLoggedIn) {
        next()
    } else {
        console.log('redirect 1');
        res.redirect('/')
    }
}
```

Check-in

Ik wil van de checkout een formulier maken met veschillende fieldsets, waar je doormiddel van progressive disclusure doorheen kunt gaan. Een gebruiker is niet verplicht alle velden in te vullen, dus een veld kan ook overgeslagen worden. Dit wil ik expliciet vermelden.

Borg betalen
Het betalen van de borg gaat via creditcard is ons aangegeven, dit werkt met een soort ‘aanvraag’ op de creditcard, die vervalt wanneer er geen borg betaald hoeft te worden.

Om het de UX fijn en overzichtelijk te houden ga ik gebruik maken van *The anatomy of a credit card form* (2015).

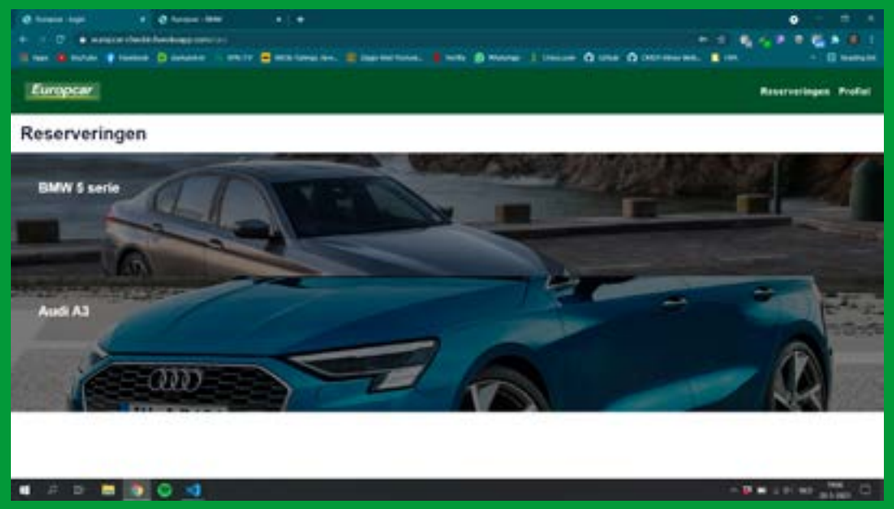
Een creditcard heeft de volgende velden nodig:
- Creditcardhouder zijn/haar naam
- Vervaldatum
- Beveiligingscode
- Creditcard nummer

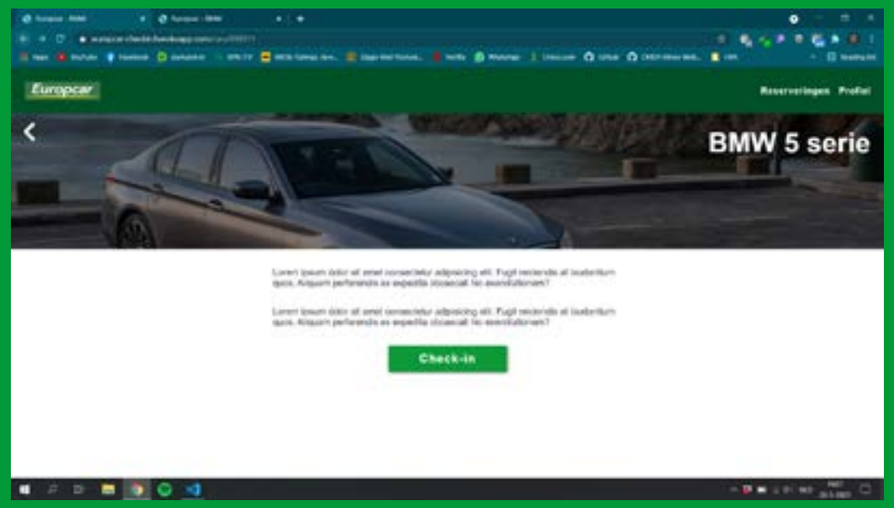
Meeting 2

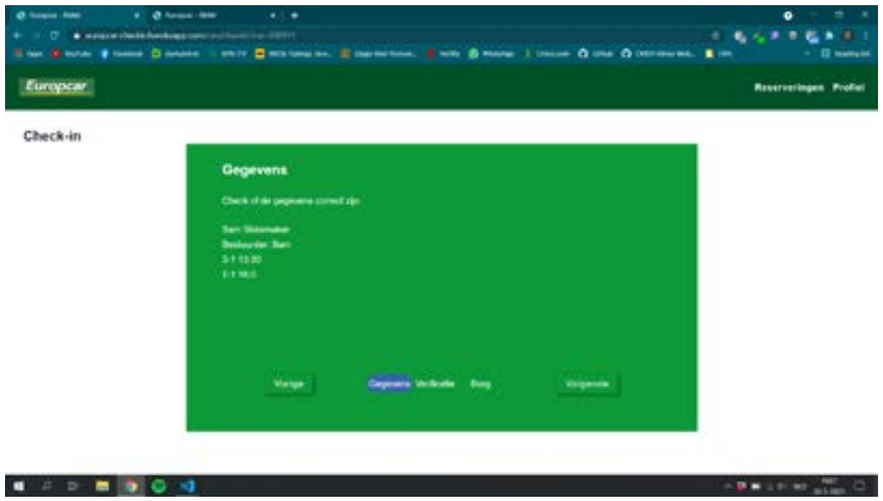
Voorbereiding

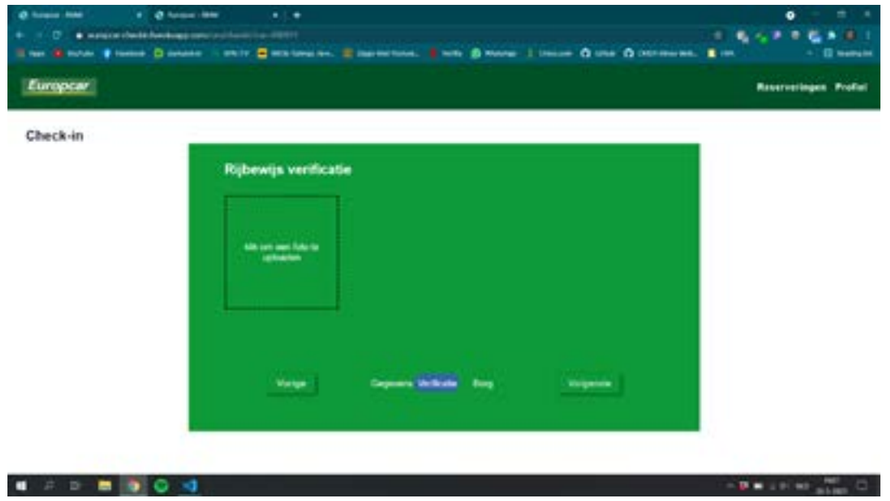
Voor de tweede meeting heb ik een aantal zaken waar ik graag wat meer duidelijkheid over wil hebben.

- Ik wil graag samen het userObject checken met gebruikers-data
- Ik wil graag samen even naar de applicatie flow kijken.
- Ik wil graag samen even nadenken over hoe de ID-verificatie afgehandeld kan worden
- Ik wil graag samen kijken naar hoe de betaling afgehandeld kan worden.
- Ik wil graag tips over de code structuur van de applicatie).
- Ik wil graag weten of ik mijn inlog op een juiste manier afhan-del.
- Ik wil graag samen kijken of een profielpagina nodig is.

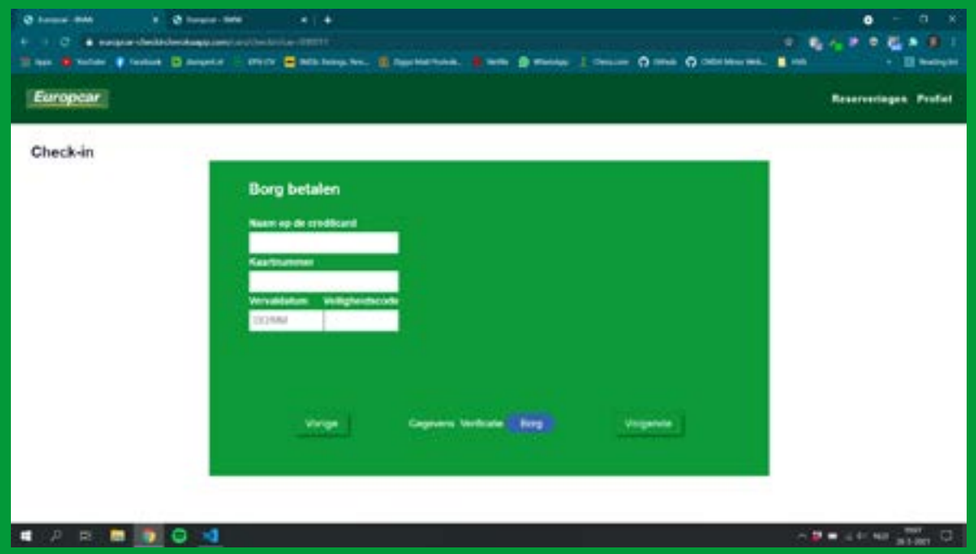

Europcar
Reserveringen Profiel
Reserveringen
BMW 5 serie
Audi A3

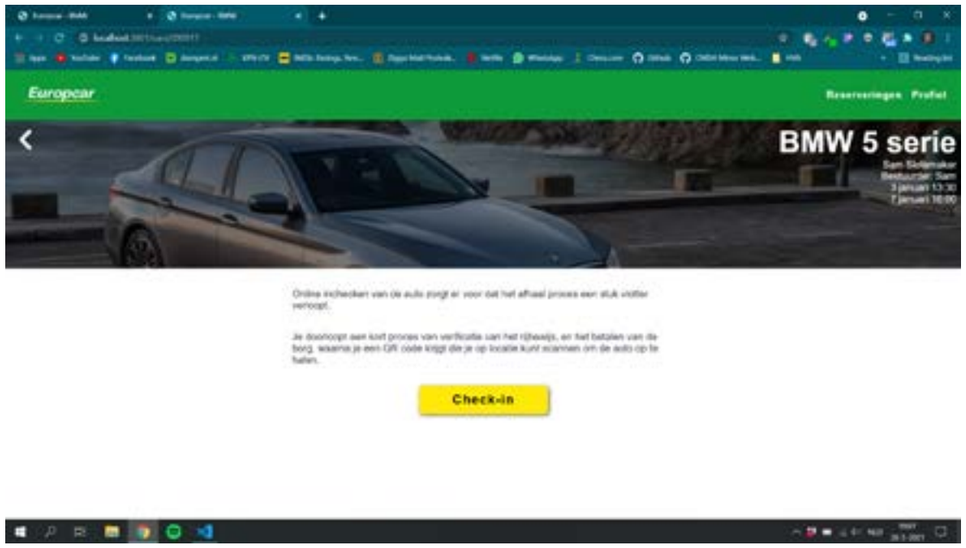

Europcar
Reserveringen Profiel
BMW 5 serie
Check-in

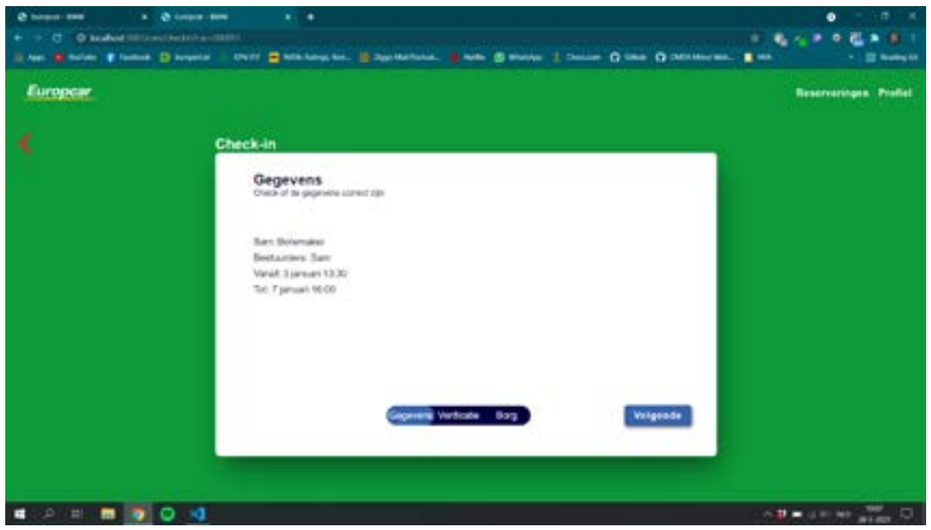

Europcar
Reserveringen Profiel
Check-in
Gegevens

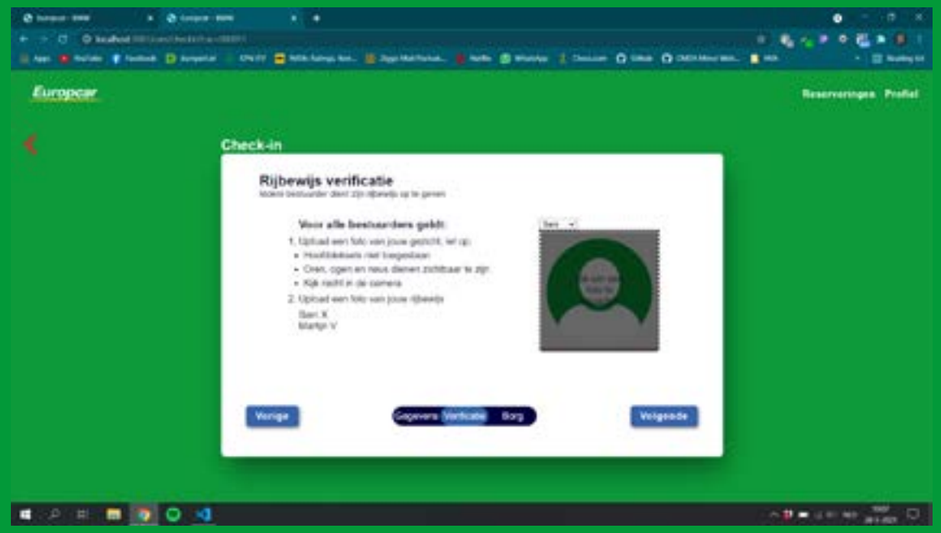

Europcar
Reserveringen Profiel
Check-in
Rijbewijs verificatie

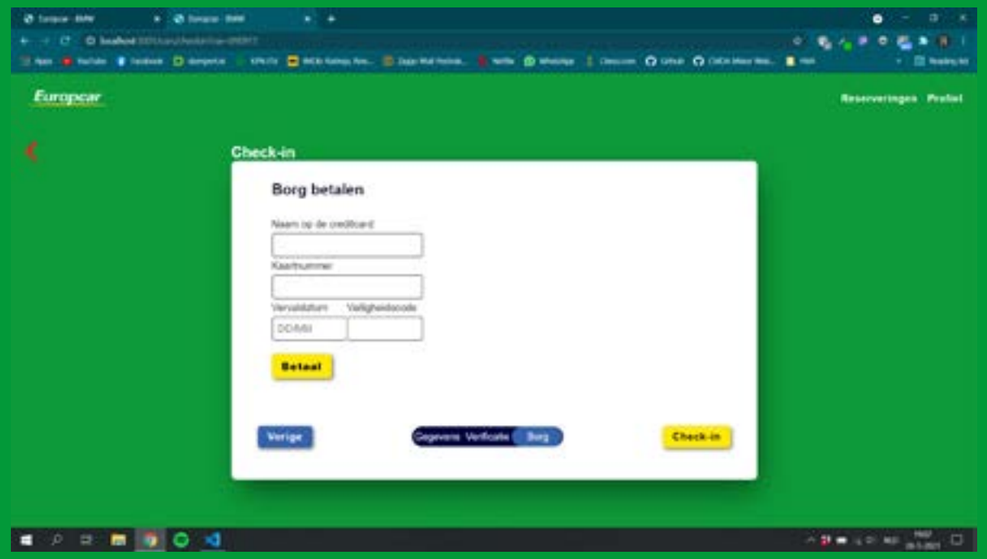
Vormgeving wissel








Europcar
Reserveringen Profiel
Check-in
Rijbewijs verificatie
Voor alle bestuurders geldt:
Upload een foto van jouw gezicht, let op:
Upload een foto van jouw rijbewijs
Gegevens Verificatie Borg
Vorige
Volgende


Europcar
Reserveringen Profiel
Check-in
Borg betalen
Betaal
Vorige
Gegevens Verificatie Borg
Check-in

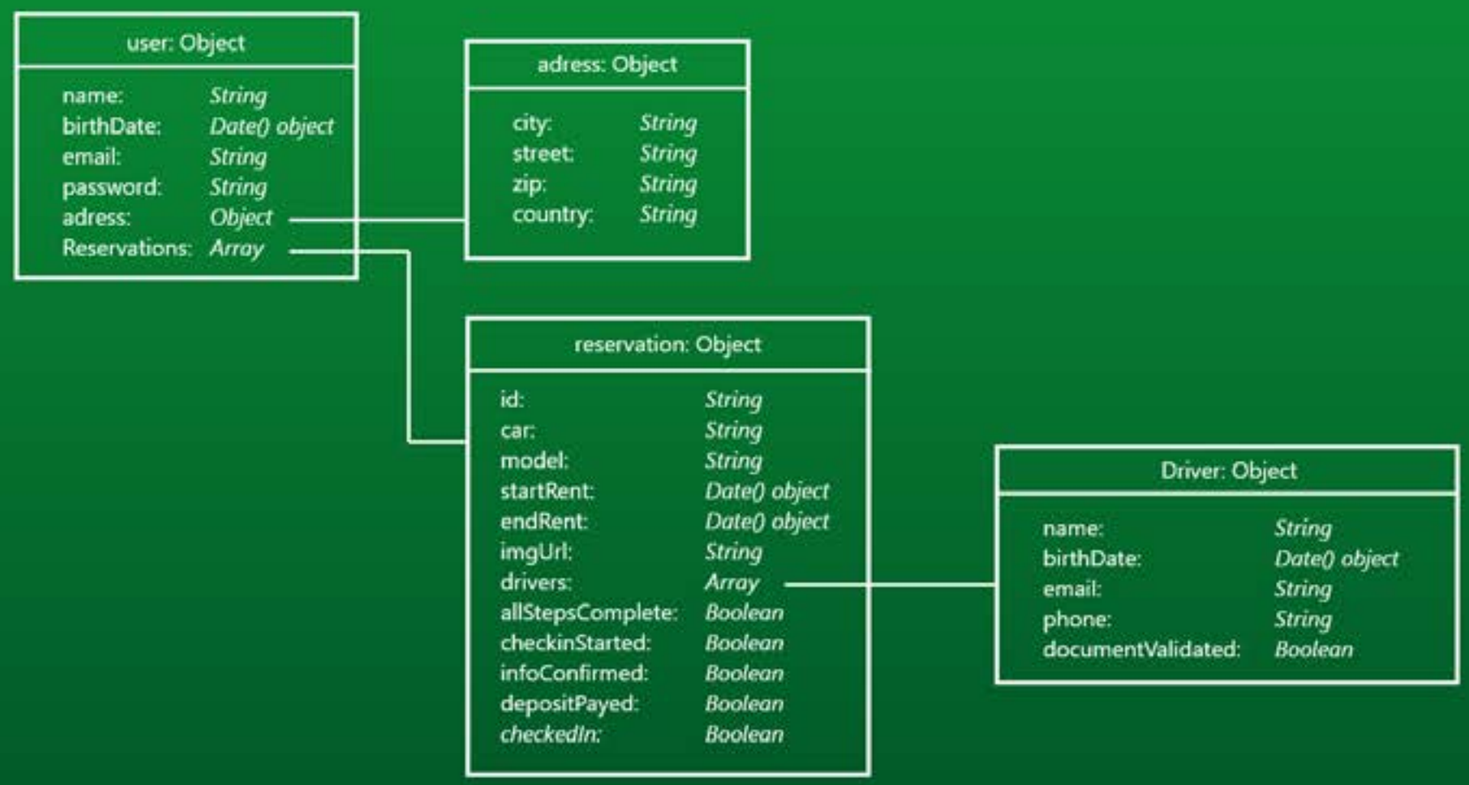
Meeting 3

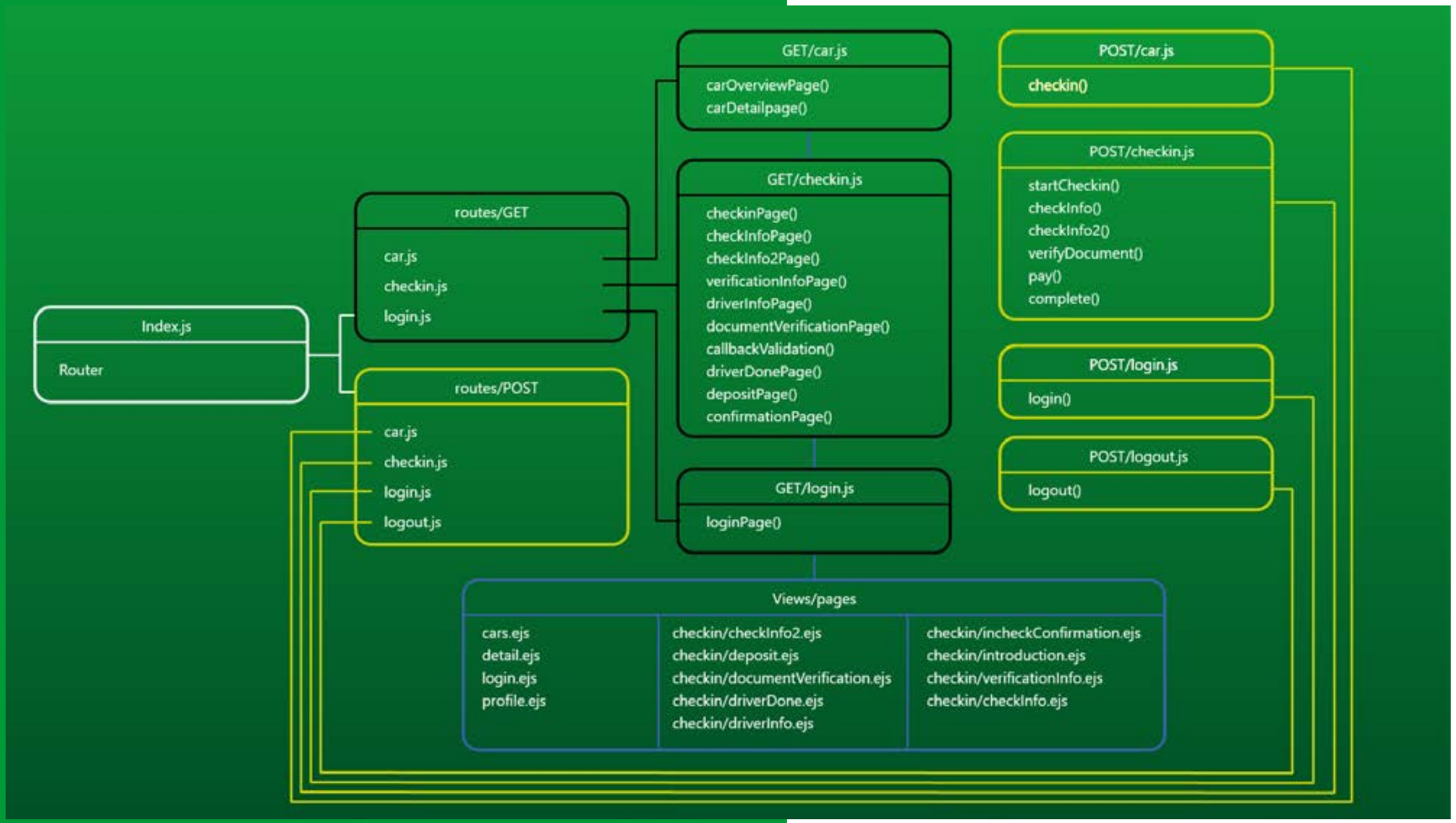
Voor de derde meeting wil ik samen met de opdrachtgever en met Q42 de flow van de checkin bekijken, een daarbij wil ik checken of de flow logisch is en wat de mogelijkheiden zijn voor de betaal en verificatie mogelijkheden.

Meeting aantekeningen

- Gebruikers zouden ook hun adres moeten kunnen verifieren wellicht.
- Gebruikers zouden in elke volgorde hun verificatie moeten kunnen voltooien.
- Adressen zouden niet geverifieerd te hoeven worden wanneer een gebruiker binnen een bepaalde tijd huurt, documenten zouden wel altijd geverifieerd moeten worden (denk zelf aan de logica).
- Maak de gebruikers duidelijk welke stappen wel of niet voltooid zijn.

Data stuctuur




Index.js
Router
routes/GET
car.js
checkin.js
login.js
routes/POST
car.js
checkin.js
login.js
logout.js
GET/car.js
carOverviewPage()
carDetailpage()
GET/checkin.js
checkinPage()
checkInfoPage()
checkInfo2Page()
verificationInfoPage()
driverInfoPage()
documentVerificationPage()
callbackValidation()
driverDonePage()
depositPage()
confirmationPage()
GET/login.js
loginPage()
POST/car.js
checkin()
POST/checkin.js
startCheckin()
checkInfo()
checkInfo2()
verifyDocument()
pay()
complete()
POST/login.js
login()
POST/logout.js
logout()
Views/pages
cars.ejs
detail.ejs
login.ejs
profile.ejs
checkin/checkInfo2.ejs
checkin/deposit.ejs
checkin/documentVerification.ejs
checkin/driverDone.ejs
checkin/driverInfo.ejs
checkin/incheckConfirmation.ejs
checkin/introduction.ejs
checkin/verificationInfo.ejs
checkin/checkInfo.ejs

Test 1

Testplan
De testpersoon dient een auto in te checken vanaf het login-scherm, waarbij alleen de inloggegevens een gegeven zijn.

Tijdens de test zal ik geen verdere informatie geven noch helpen wanneer de testpersoon vastloopt.

Ik zal tijdens de test op de muisbewegingen letten, kijken waar de ogen van de testpersoon als eerst op gericht zijn en waar handelingen anders gaan dan ik verwacht had.

De test is afgelopen wanneer de testpersoon het scherm met de QR code voor zich heeft.

De test bestaat uit de volgende delen:

Inloggen
Auto incheck starten
Een auto inchecken

Resultaten

Test 1
Inlog
Inloggen ging zonder problemen

Incheck starten
Een reservering kiezen en het starten van het incheck proces ging soepel, de inleidende tekst werd gelezen.

Een auto inchecken
Het inchecken van de auto verliep opzich prima, de volgorde van de stappen was logisch en de formulieren werden gevonden, echter:

- De gebruiker wilde zijn gegevens kunnen wijzigen
- Foto uploaden en maken werkte goed, alleen werd maar 1 bestuurder voltooid.
- Foto's werden ook niet verstuurd, enkel geupload en gemaakt, de gebruiker kilte niet op de upload knop.

Belangrijk hierin is:
Geef de gebruiker explicieter de mogelijkheid om beide gebruikers in te checken, zorg er voor dat deze niet vergeten worden.
Zorg er voor dat de gebruiker niet zomaar naar de volgende stap gaat, zonder dat de identiteit expliciet geverifieerd wordt.

Test 2
Inlog
Inloggen ging zonder problemen

Incheck starten
Een reservering kiezen en het starten van het incheck proces ging soepel, de inleidende tekst werd gelezen.

Een auto inchecken
Er werd deze keer wel op de upload knop geklikt na het uploaden van een foto, alleen werd wel de tweede bestuurder volledig genegeerd.

De tester probeerde via de progressie balk naar het volgende onderdeel te klikken.

De gebruiker wilde de qr code naar zichzelf kunnen mailen.

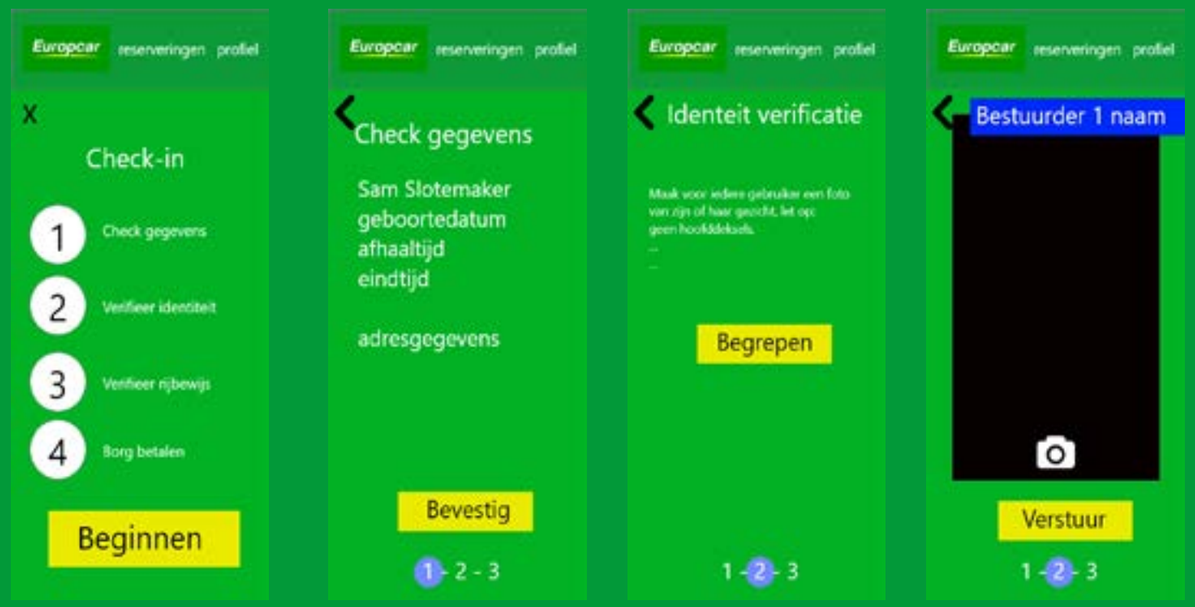
Herontwerp plan

Na het testen blijkt dat het checkin proces nog lang niet logisch genoeg is. De testers voltooiden de check-in beide niet vlekkeloos, waardoor er tijdens het afhalen een probleem zou kunnen ontstaan.

Dit zit hem denk ik in de volgende aspecten:
- Te weinig informatie vooraf
- Te weinig feedback naar de gebruiker
- Een onlogische plaatsing en hierarchie van knoppen
- Overige bestuurders zitten verstopt achter een dropdown
- Volledig voltooien van de verificatie zit verstopt achter een extra knop waar niet op gedrukt werd.

Ik denk dat ik het formulier minder ‘vrijwillig’ moet maken, en dat ik de bestuurders niet meer moet verstoppen. Ik wil er voor zorgen dat het voltooien van de verificaties als de ‘primary action’ voelt, in plaats van de ‘volgende’ knop. Dit zou kunnen door geen gebruik te maken van een grote volgende knop, maar 1 call to action die de actie voltooid en een ‘overslaan’ knop die niet direct opvalt.

Herontwerp flow



Herontwerp styles

Feedback

Maak 1 stap van de verificatie, ipv 2.
Verifieer per gebruiker, in plaats van per stap: Laat een gebruiker meteen zijn foto maken en rijbewijs uploaden, in plaats van beide gebruikers per stap.

Geef de gebruiker meer feedback over de stappen, introduceer de stappen in de inleiding en gebruik misschien iconen in plaats van getallen.

Geef de gebruiker informatie over hoeveel borg er betaald moet worden.


Europcar
Check-in
Check gegevens
Persoon verificatie
Borg betalen
Iedere stap kan worden overgeslagen om op locatie af te ronden.
Beginnen


Europcar
Check gegevens
Sam Slotemaker
06-01-2000
12:00 10 februari 2021
12:00 10 februari 2021
Westerweg 120
Heiloo
1922 DS
Nederland
Wijzigen
Bevestig


Europcar
Check gegevens
Borg 400,-
Verzekering 67,1-
Totaal 6421
Bevestig


Europcar
Bestuurder 1:
Sam Slotemaker
Check of de volgende informatie klopt bij de hierboven genoemde bestuurder en begin vervolgens de verificatie
Sam Slotemaker
06-01-2000
0612233440
Westerweg 120
Heiloo
1922 DS
Nederland
Beginnen
Overslaan


Europcar
Borg betalen
Betaalmethode
Betalen
Overslaan


Europcar
Borg betalen
Betalen
Overslaan


Europcar
Incheck afronden
Check gegevens
Persoon verificatie
Borg betalen
Afronden


Europcar
Persoons verificatie
Maak een foto van je volledige gezicht, let op:
• Draag geen hoofddeksels.
• Oren, ogen en neus zijn zichtbaar.
• Kijk recht in de camera.
Begrepen
Overslaan


Europcar
Sam Slotemaker
Verstuur
Overslaan


Europcar
Sam Slotemaker
Verifieer rijbewijs
Verstuur
Overslaan


Europcar
Bestuurder 1:
Sam Slotemaker
Het verificatieproces van bestuurder 1 is afgerond. Mochten er stappen zijn overgeslagen, kunnen deze op locatie worden afgehandeld.
Verder


Europcar
Martijn Tromp
Rijbewijs ontvangen
Verstuur
Overslaan


Europcar
Betaling voltooid
Verder
Overslaan


Europcar
Martijn Tromp
Verstuur
Overslaan

Meeting aantekeningen

Handig om een gebruiker vooraf te laten zien wat er nodig is, zoals een rijbewijs
Misschien moet het iets duidelijker welke stappen het zijn, (in plaats van 1 – 2 – 3 – 4)
Wil je per onderdeel verifiëren of per gebruiker?
ID verifiëren met stribe

ID verificatie

Stripe
Om rijbewijzen te verifieren wil ik gebruik van maken van stripe. Stripe is een externe service die dit zou kunnen regelen, waardoor het niet nodig is om deze functionaliteit volledig werkend te maken binnen mijn applicatie.

In principe zou je met Stripe enkel een button maken in de applicatie, die een post request doet naar de server. Binnen de server wordt vervolgens een session aangemaakt voor Stripe en die wordt teruggestuurd naar de client. Vervolgens handel je het verifiëren van het document af in een pop-up van Stripe. Stripe stuurt uiteindelijk een response terug naar de applicatie die laat weten of het proces geslaagd is. Dit zou er voor zorgen dat de gebruiker eigenlijk niet hoeft te wachten op de verificatie van het rijbewijs.

Werkt niet

Ik heb uiteindelijk geen idee waar de fout zit. Stripe is beschikbaar in Nederland en ik heb het werkend gezien. Helaas.

Page transitions

Highway
Ik wil page transitions gaan maken doormiddel van een library, hiervoor heb ik de library Highway gevonden. Highway is een library die de oude pagina en de nieuwe pagina in een bestand samenvoegt, en hiermee een transitie tussen beiden kan maken.

Gsap
Om de animatie van de transitie netjes te maken, wil ik gebruik maken van gsap. Gsap is een library die een timeline creëert die animaties kan chainen.

Test app
Ik heb een testapplicatie gemaakt om beide libraries te testen voor de pagina transities.

Er wordt simpelweg een Class aangemaakt, met een in() en out() method. De in method zorgt voor de display, en de out method zorgt voor het verbergen. Binnen deze methods maak ik de gsap timeline aan.

Binnen de gsap timeline heb ik gekozen voor een simpele opacity switch.

In de html moet er een ‘data-router-wrapper’ op de container staan en een ‘data-router-view’ op de container van de wisselende content.

```
// page transitions
class Fade extends highway.Transition {
    in({ from, to, done }) {
        const tl = new TimelineLite();
        tl.fromTo(from, { opacity: 1 }, {
            opacity: 0, duration: .3, onComplete: function () {
                from.remove()
                done()
            }
        })
    }
    out({ from, done }) {
        done();
    }
}

const H = new highway.Core({
    transitions: {
        default: Fade
    }
})
```

Helaas werkt dit enkel in mijn test applicatie waar ik naar html bestanden link, en krijg ik de library niet werkend voor een node. js applicatie waar ik templates serveer.

Wellicht kan ik het uiteindelijk nog voor elkaar krijgen door het gewoon vanilla te doen met wat delays op de buttons.

Herbruikbare templates

```
<nav>
    <ul>
        <li><a href="<%= backUrl %>">
                <%- include('./icons/back.ejs') %>
            </a></li>
        <li><img src="/style/images/europcar_logo.png" alt=""></li>
        <li>
            <a href="">

                <%- include('./icons/close.ejs') %>
            </a>
        </li>
    </ul>
</nav>
```

```
let backUrl = `/cars/checkin/verificationInfo?car=${car.id}`

res.render('checkin/driverInfo', { title: 'check-in', car, user, driver, driverNumber, backUrl, status })
```

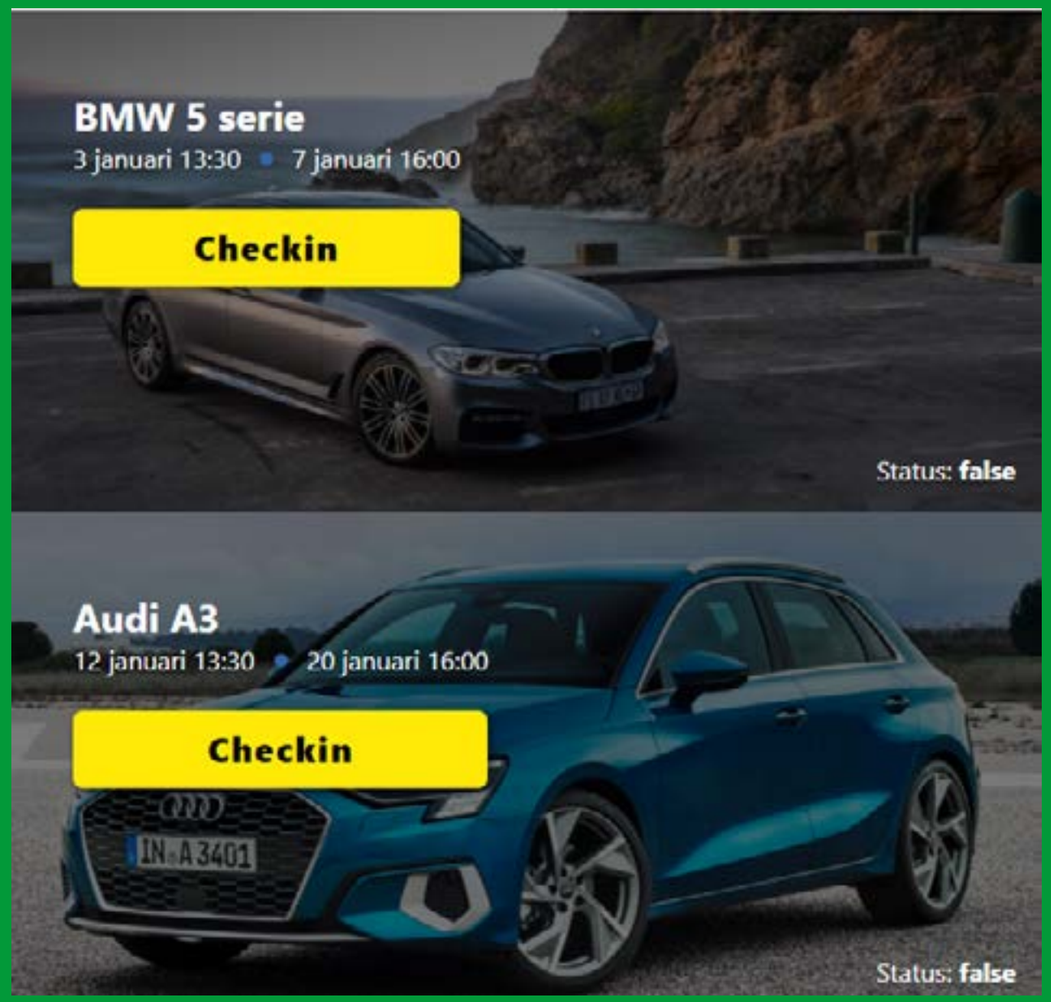
Ieder checkin scherm kan nu dezelfde navigatie gebruiken, waarbij het enige veranderende component de terugknop is, die naar het vorige scherm moet verwijzen. Deze geef ik dus mee aan de render functie, waarbij deze in de template gerendert wordt.

Hetzelfde principe kan ik nu voor de progressie bar gebruiken, waar de class dynamisch is. De class geeft nu aan in welke status het bolletje verkeert en hiermee welke kleur deze moet zin

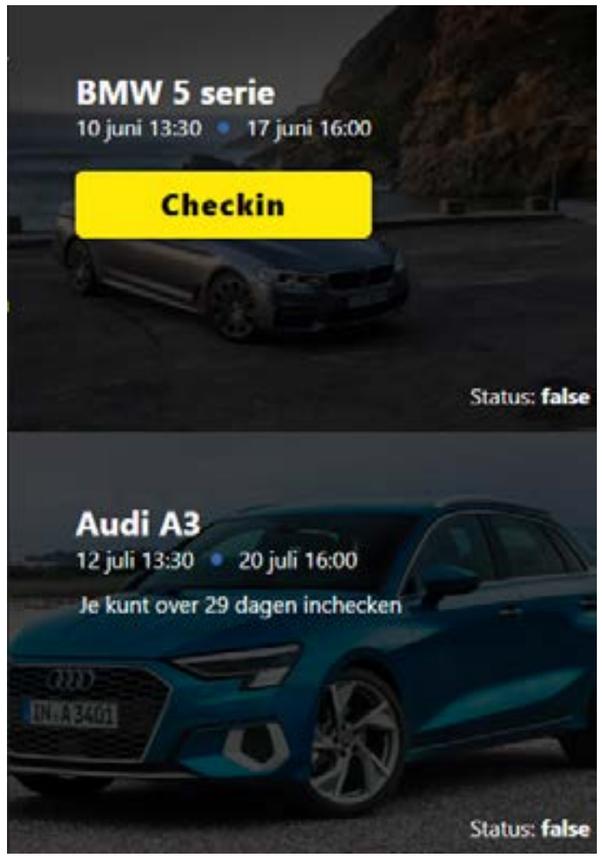
```
<section class="checkin__progress_bar">
    <ul>
        <li class="<%= status.infoStatus %>">
            <%- include('./icons/notes.ejs') %>
        </li>
        <li class="<%= status.verifyStatus %>">
            <%- include('./icons/id.ejs') %>
        </li>
        <li class="<%= status.paymentStatus %>">
            <%- include('./icons/credit-card.ejs') %>
        </li>
    </ul>
</section>
```

```
let status = { infoStatus: 'done', verifyStatus: 'doing', paymentStatus: 'blanc' }
```

Vertrektijden op homescherm en duidelijke call to action



Logica toegevoegd dat een gebruiker 7 dagen voor aanvang kan inchecken



Berekenen van tijd tussen datums

```
/**
 * returns day difference from current date to given date
 * @param {object} date - JS date object - newDate()
 */
export function getDateDifference(date) {
    const now = new Date();
    const diffTime = Math.abs(now - date);
    const diffDays = Math.ceil(diffTime / (1000 * 60 * 60 * 24));
    return diffDays
}
```

```
let dateDifferences = reservations.map(car => {
    return getDateDifference(car.startRent)
})
```

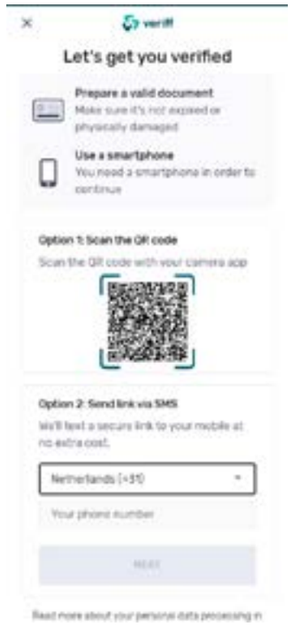
```
<% if(dateDifferences[index] < 8){ %>
<a class="button" href="/cars/<%= car.id %>">Checkin</a>
<% } else { %>

<strong>Je kunt over <%= dateDifferences[index] - 7 %> dagen inchecken</strong>
<% } %>
```

Veriff

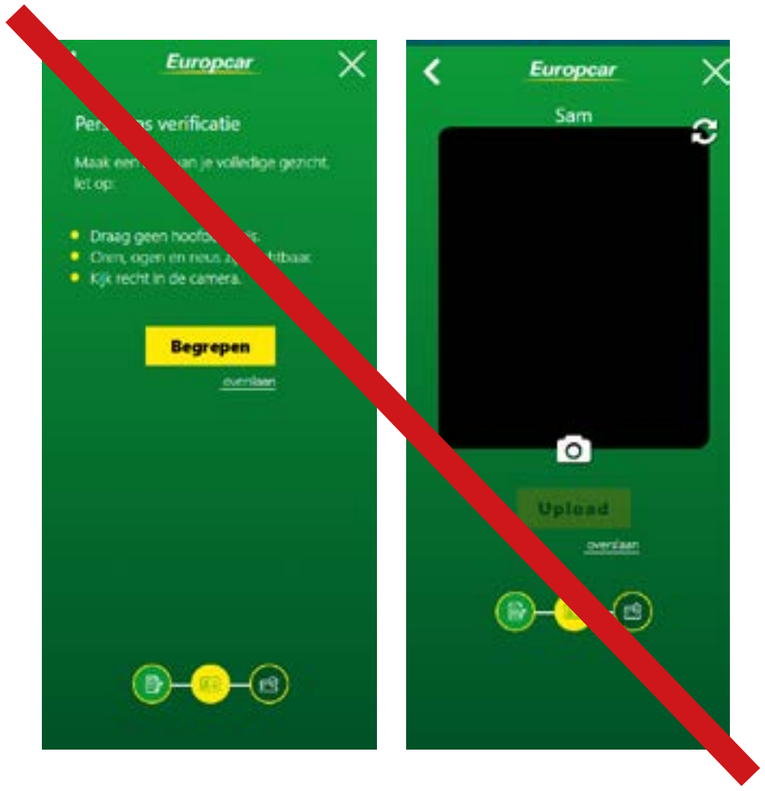
Veriff is eenzelfde soort service als Stribe, het kan gebruikt worden om identiteit te verifieren. Het fijne aan veriff is hier ook dat het tegelijkertijd om een selfie vraagt en deze met Artificial Intelligence tegen het gegeven rijbewijs checkt. Je kunt ook zelf aangeven welke documenten gebruikers mogen gebruiken.

Er kan simpelweg een connectie gemaakt worden door devolgende code in een script tag van de pagina te zetten.

```
const veriff = Veriff({
    host: 'https://stationapi.veriff.com',
    apiKey: '02dfbf41-b5c9-4411-8e17-5d44332046a8',
    parentId: 'veriff-root',
    onSession: function (err, response) {
        window.veriffSDK.createVeriffFrame({ url: response.verification.url });
    },
});
veriff.setParams({
    person: {
        givenName: ' ',
        lastName: ' '
    },
    vendorData: ' '
});
veriff.mount({
    submitBtnText: 'Verifieer jezelf',
    loadingText: 'Aan het laden...'
});
```



Omdat de foto ook binnen veriff gemaakt wordt, is hiervoor geen plaats meer nodig binnen mijn flow, dat elimineert dus de de volgende schermen.



Meeting aantekeningen

Q42 was erg tevreden over wat er nu stond.
Klein feedback puntje was dat de kleine letters mischien wat groter konden.

Tekst dat je wordt doorgestuurd naar een externe partij kan duidelijk!

Terugknop

De knop knop was een erg vervelend aspect. Omdat deze dynamisch moet zijn wanneer de app dynamisch is.

De terugknop link nu standaard naar de vorige pagina in de flow, zodat deze altijd te gebruiken is.

Wanneer de gebruiker javascript aan heeft staan linkt deze doormiddel van de History API naar de laatst bezochte pagina, zodat de terugknop dynamisch werkt.

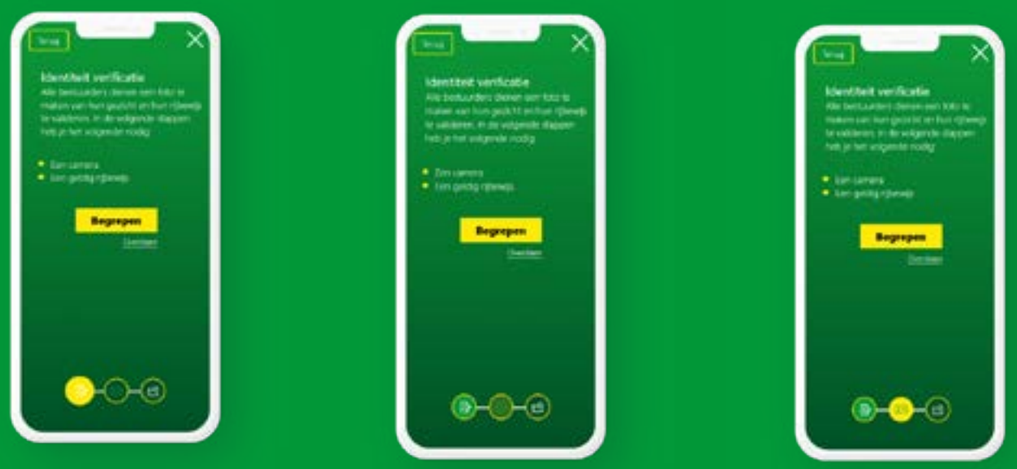
```
//change all backbuttons on checkin to lastvisited page
const checkinBackButton = document.querySelector('.checkin__back_button')
if (checkinBackButton) {
    checkinBackButton.href = '#'
    //go back in history when back is clicked
    checkinBackButton.addEventListener('click', () => {
        window.history.back()
    })
}
```

Animaties

Ik ga de progress bar een animatie geven zodat er feed-back gegeven wordt aan de gebruiker wanneer een stap voltood is.

Ook wil ik de page transitions nog een keer proberen met vinilla javascript.

De afgeronde stap wordt nu iets groter waarna deze groen wordt, de te beginnen stap wordt daarna lang-zaam geel.



Vanilla transition

opzich was dit niet enorm ingewikkeld. Ik zet nu sim-pelweg een delay op de pagina, wanneer de pagina een a met href heeft wordt de pagina na de delay doorge-stuurd en wanneer dit niet het geval is wordt het formu-lier gesubmit.

Op het laden van de pagina staat gewoon een animatie, dus deze wordt standaard uitgevoerd op pagina's met het script.

```
//handle fadeout when javascript is on
function handleNextPage(event) {
    event.preventDefault();
    container.classList.add('fade-out')

    setTimeout(() => {
        if (event.target.href) {
            window.location = event.target.href
        }
        else {
            form.submit()
        }
    }, 400)
}

nextButton.addEventListener('click', handleNextPage)
```

ReadMe

Bij het schrijven van een readMe heb ik de volgende structuur gemaakt waar ik het over wil hebben.:

installation - hoe installeer je de app
debriefing - Waarvoor is de app gemaakt en waarom
Code structure - hoe werkt de structuur van de code
Data - op welke data is de applicatie gebouwd
Npm packages - welke packages zijn er gebruikt en wat doen ze

de readme is hier te vinden
https://github.com/SamSlotemaker/europcar-check-in-#readme

Jsdoc document

Met de command:

json src -r

Kan ik van alle javascript bestand een HTML bestand genereren uit de gemaakte code comments. De -r zorgt er voor dat alle geneste mappen ook meegenomen worden.

Om de modules te structureren heb ik aan ieder bestand nog een comment toegevoegd om de modules aan te geven.

```
/**
 * module beschrijving.
 * @module moduleNaaam
 */
```

het document is hier te bekijken
https://samslotemaker.github.io/europcar-check-in/docs/code-Doc/index.html

Evaluatie

Proces

Het proces begon redelijk soepel. Ik vond het een leuke opdracht en had al gauw wat ideeën in mijn hoofd. De ideeën kwamen misschien iets te makkelijk, waardoor ik gelijk ben begonnen met bouwen, hierdoor was de applicatie eigenlijk nog niet volledig uitgedacht en klopte er hier en daar een aantal zaken simpelweg niet. Hierdoor moest ik terug naar de tekentafel en eigenlijk praktisch opnieuw beginnen, maar ik ben er wel van overtuigd dat dit er voor heeft gezorgd dat het product uiteindelijk vollediger en bruikbaarder is geworden.

Doordat ik het project in mijn eentje heb moeten doen, heb ik in ieder geval in alle aspecten nieuwe dingen moeten doen, waardoor ik ook enorm veel nieuwe dingen heb geleerd. Wel denk ik dat er ook veel nadelen aan zitten, zo werkte ik bijvoorbeeld volledig op mijn eigen manier, zonder andere oplossingen aan te horen van andere programmeurs. Hierdoor is er eigenlijk nooit zekerheid om mijn oplossing te slimste/beste is, doordat er simpelweg geen tegenspraak is.

Waar ik uiteindelijk graag meer aandacht aan zou willen besteden is de UX. Er zijn nog genoeg zaken die wellicht niet helemaal duidelijk zijn of beter hadden gekund, met name de feedback naar de gebruiker is waar ik denk ik een steek heb laten liggen. Dit geldt zowel voor feedback op de handelingen, als feedback over het incheck proces: Wat gebeurt er met jouw data? Waarom is jouw data nodig? Wat nu verder? Etc. Dit zorgde er echter wel voor dat ik meer tijd heb kunnen besteden aan de realisatie van het product, waardoor ik meer naar voornaamste doel, het leren developen, toe heb kunnen werken.

Vakken

Eigenlijk denk ik dat vrijwel alle vakken in dit project naar boven zijn gekomen, afgezien van de real-time events bij Real TIme Web.

Web app from Scratch
Bij WAFS heb ik geleerd hoe je code opsplitst in modules, iets waar ik me enorm op heb gefocust in dit project om alles overzichtelijk te houden.

CSS to the rescue
Mijn specifieke opdracht bij CSSttR had niet bijster veel met layouts te maken, maar heeft er wel voor gezorgd dat ik bepaalde CSS properties ben gaan begrijpen en gebruiken. Met name de CSS variabelen komen enorm goed van pas in een wat groter project.

Progressive web apps
Bij PWA hebben we leren wreken met Node.js, express en template engines, eigenlijk de hele basis van mijn applicatie dus. Wat ik graag nog toe had willen passen was de service worker, om de applicatie offline beshikbaar te maken. Hier ben ik helaas niet meer aan toe gekomen omdat het geen prioriteit had.

Browser technologies
BT heeft er voor gezorgd dat ik de applicatie volledig progressive enhanced wilde laten zijn. De applicatie werkt volledig zonder JavaScript en waar javascript echt nodig is, bijvorobeeld de validatie, krijgt de gebruiker een melding dat deze stap niet mogelijk is.

Humand Centered Design
HCD is niet specifiek teruggekomen, omdat er geen exclusive design gemaakt wordt, maar het heeft er wel voor gezorgd dat ik in ieder geval heb nagedacht over de semantiek en het gebruik van de juiste HTML elementen, zodat de site toegankelijk blijft.

Real time web
De essentie van RTW is dan misschien niet volledig terug gekomen in de meesterproef, maar de RTW applicatie bestaat wel voor een groot deel uit dezelfde componenten. Hierdoor was RTW eigenlijk een kleine voorbereiding op de Meesterproef.

Bronnen

Veriff, van https://www.veriff.com/
Gsap, van https://greensock.com/gsap/
Barba, van https://barba.js.org/
Stripe, van https://stripe.com/en-nl
Express docs, van https://expressjs.com/en/starter/installing.html
History API, van https://developer.mozilla.org/en-US/docs/Web/API/History_API
JsDocs, van https://jsdoc.app/about-getting-started.html
Sessions, van https://www.npmjs.com/package/cookie-session
audi A3, van https://www.besthdwallpaper.com/audi/audi-a3-sport-back-receptie-dt_nl-53511.html
bmw 5 serie, van https://wall.alphacoders.com/big.php?i=809574&lang=Dutch