



**TRACK**

CMS-DEVELOPMENT

ACADEMIEJAAR: 2017 - 18

AURELIO DECOCK

NEW MEDIA DEVELOPMENT



**track**



**arteveldehogeschool**  
LID VAN DE ASSOCIATIE UNIVERSITEIT GENT





## Discover

Briefing	4
----------	---

## Define

Analyse	5
Specifications	6
Planning	7

## Design

Sitemap	8
Wireframes	9
Wireflow	14
Style guide	15
Visuals	16

## Develop

Code snippets	21
Screenshots	23

## Deliver

Handleiding	24
-------------	----

## Deploy

Deployment guide	27
------------------	----

# DISCOVER

## Briefing

Voor de eindopdracht van CMS-Development werd ons verteld een digitaal platform te ontwikkelen met behulp van drupal en vue.js om het uitlenen van elektrische voertuigen te bevorderen.

### **Het doel van de applicatie.**

De applicatie wordt ontwikkeld om voor zowel de huurder als verhuurder een platform aan te bieden waar de gebruiksvriendelijkheid centraal staat. Indien je als gebruiker zelf een elektrisch voertuig beschikbaar hebt, kan je deze ook makkelijk ter beschikking stellen.

### **Wie behoort tot het doelpubliek?**

Het doelpubliek is voor iedere gebruiker boven de 18 jaar die in bezit is van een rijbewijs B. Specifieke doelgroepen kunnen zijn:

- Eigenaars van 1 of meerdere elektrische voertuigen
- Mensen zonder wagen die voor een korte periode een elektrisch voertuig willen huren.
- Mensen die een elektrisch voertuig willen uittesten voor een bepaalde periode.
- ...

### **Wat is de timing?**

De opdracht moet worden ingediend ten laatste 16 januari 2018.

# DEFINE

## Analyse

Vooraleer we starten met het ontwikkelen van een applicatie gaan we eerst een analyse uitvoeren. Dit houdt in dat we de concurrentie onder de loep nemen, wat we de klant bieden en hoe we dit aanpakken.

### Concurrentie

Het spreekt voor zich dat we de concurrenten even bekijken. Er zijn al enkele platformen die een gelijkaardige service bieden, zoals:

- ev-verhuur.be
- sixt.be
- dockx.be
- ...

### Wat bieden we de klant?

In het geval van deze opdracht is er geen effectieve klant. Maar we gaan er van uit dat dit voor een fictieve klant gemaakt wordt. In dit geval geven we de klant een webapplicatie die bestaat uit drupal 8 voor de backoffice en vue.js voor de frontoffice. Daarnaast wordt er een eigen css-framework geschreven voor de uitbreidbaarheid van de app te bevorderen. De communicatie tussen de front en backoffice gebeurt via een RESTful API die wordt gegenereerd van drupal.

### Hoe pakken we dit aan?

We beginnen bij het ontwikkelen van de UI en de UX. We vertrekken vanuit de logica van een sitemap. Daarna starten we met het opzetten van de ontwikkelomgeving. Eenmaal die op punt staat gaan we de backoffice in orde brengen door de nodige modules en entities te installeren en genereren. Vervolgens gaan we aan de slag met de front office om zo het eindproduct te bekomen.

## Specificaties

Het spreekt voor zich dat een applicatie komt met verschillende specificaties. Hieronder vindt je een lijst met alle functionele, niet-functionele en technische specificaties.

### Functionele specificaties

- Gebruiker kan zich registreren
- Gebruiker kan zich aanmelden
- Gebruiker kan zijn voertuig aanbieden voor verhuur
- Gebruiker kan een voertuig huren
- Gebruiker kan een overzicht raadplegen en filteren van de voertuigen
- Gebruiker kan de boekingen van zijn voertuig raadplegen.

### Niet-functionele specificaties

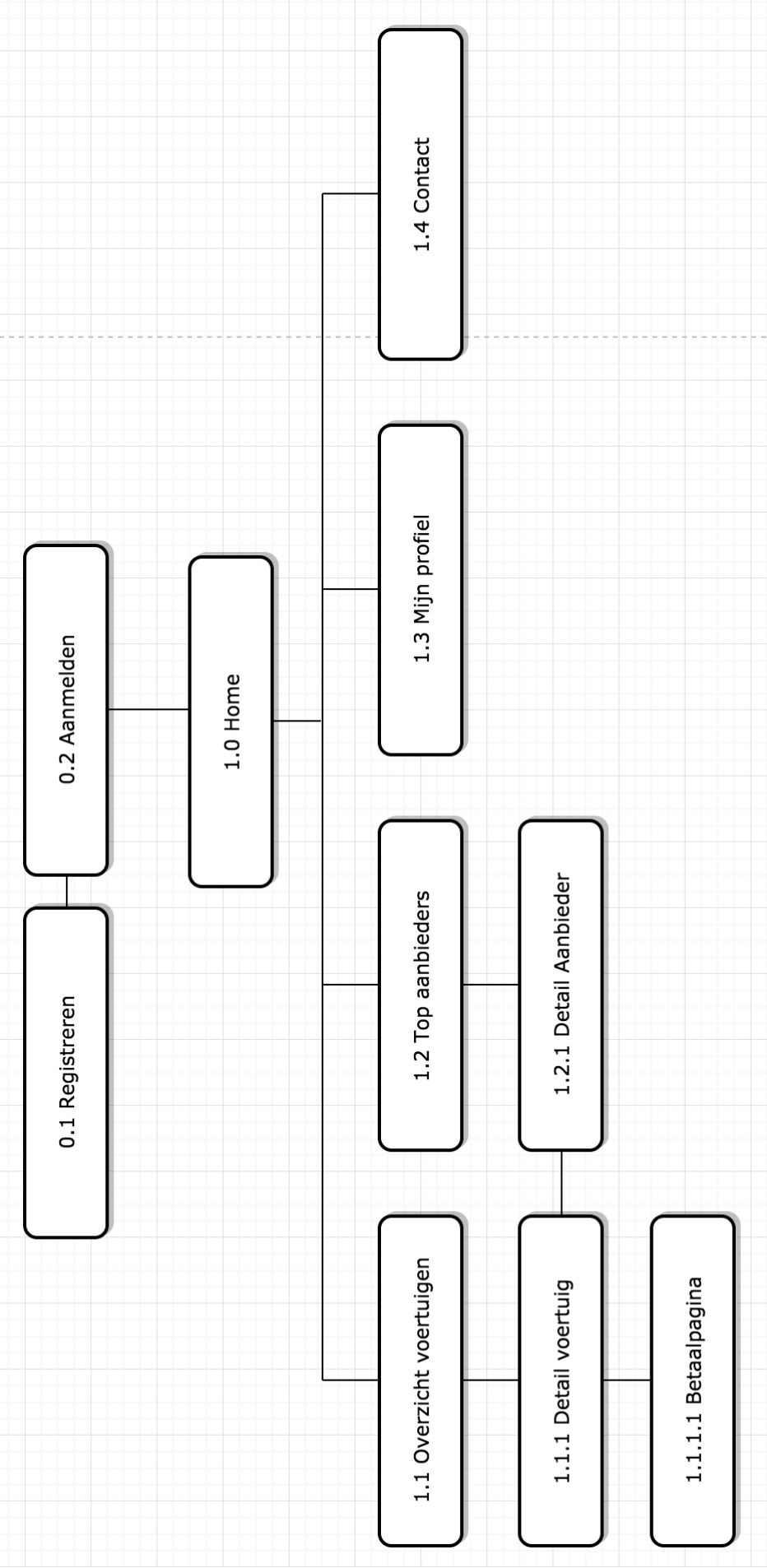
De applicatie moet opgebouwd worden op een gebruiksvriendelijke manier. Dit wil zeggen dat de gebruiker zonder enige moeite de applicatie moet kunnen gebruiken. Maar naast de UX moet de applicatie ook op een kwalitatieve manier gemaakt worden. Daarom besteden we belang aan de netheid van de code en schrijven we die op de best mogelijke manier.

### Technische specificaties

Voor het realiseren van deze applicatie maken we gebruik van volgende technische specificaties:

- Drupal 8
  - (hal)\_JSON
  - API
- Vue.js
  - axios
  - ES6
- Responsive
- Een custom form integreren
- SASS


FASE 1: ONDERZOEKFASE					
Deeltaak	Projectbeschrijving	Start datum	Eind datum	Week 39	Week 40
		26-9-2017	26-9-2017		
FASE 2: ONTWERPFASE					
Deeltaak		Start datum	Eind datum	Week 39	Week 40
	Sitemap	9-10-2017	9-10-2017		
	Wireframes	16-10-2017	16-10-2017		
	Style tiles	16-10-2017	16-10-2017		
	Visual designs	23-10-2017	23-10-2017		
FASE 3: ONTWIKKELFASE 1 (Statische vormgeving)					
Deeltaak	Visual design omzetten in HTML/CSS	Start datum	Eind datum	Week 39	Week 40
		23-10-2017	23-10-2017		
FASE 4: ONTWIKKELFASE 2 (Dynamische vormgeving)					
Deeltaak		Start datum	Eind datum	Week 39	Week 40
	Opzetten ontwikkelomgeving	10-10-2017	10-10-2017		
	Gebruikers aanmaken/inloggen via API	30-10-2017	3-11-2017		
	Voorpagina's ophalen via API + detailpagina's	6-11-2017	10-11-2017		
	detailpagina's + reservatie flow	13-11-2017	17-11-2017		
	Voorpagina's aanbieden	20-11-2017	24-11-2017		
	Profiel + features	27-11-2017	1-12-2017		
	Extra features	4-12-2017	12-1-2018		
FASE 5: ONTWIKKELFASE 3 (Bugfixing en finetuning)					
Deeltaak	Fijnen van bugs	Start datum	Eind datum	Week 39	Week 40
		30-10-2017	12-1-2018		
FASE 6: DEPLOYMENT					
Deeltaak	Eindproduct voorleggen	Start datum	Eind datum	Week 39	Week 40
		12-1-2018	15-1-2018		
SAMENVATTING					
	Onderzoeksfase	25-4-2016	3-5-2016		
	Ontwerpfase	25-4-2016	5-5-2016		
	Ontwikkefase 1	3-5-2016	12-5-2016		
	Ontwikkefase 2	28-4-2016	20-5-2016		
	Ontwikkefase 3	3-6-2016	3-6-2016		
	Deployment	3-6-2016	6-6-2016		





# DESIGN

## Wireframes

 track

**Name**

eg: Jaison Justus

**Email**

eg: yourname@gmail.com

**Username**

eg: jaison.justus

**Password**


.....

**Password confirm**

.....

**SIGN UP**

0.1 Registreren

 track

**Username**

eg: jaison.justus

**Password**

.....

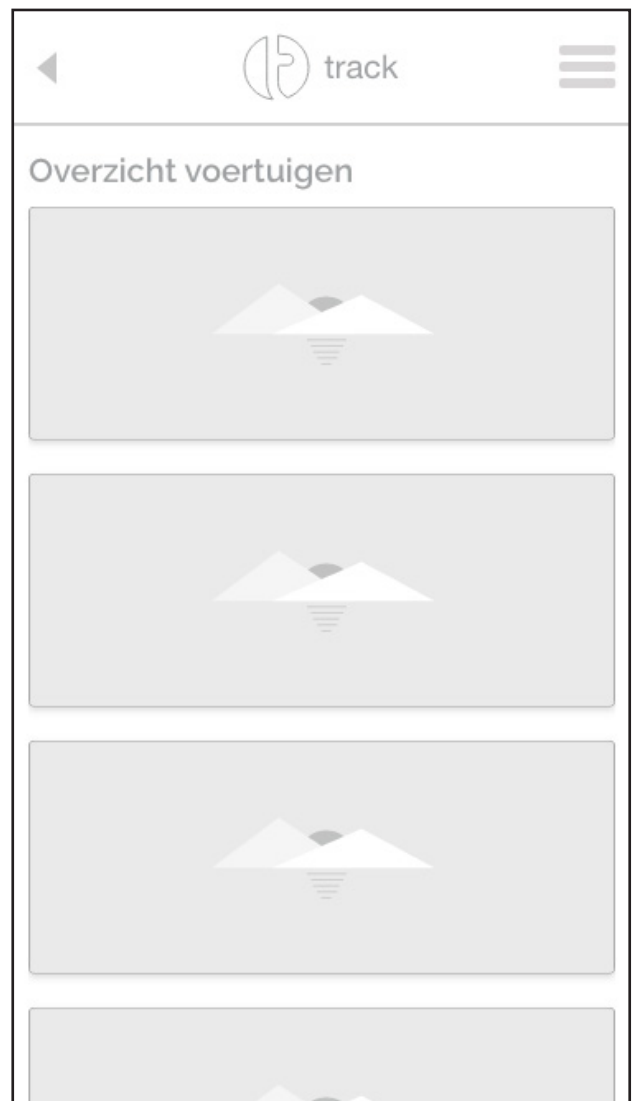
☐ Remeber me on this computer

**SIGN UP** **LOGIN**

0.2 Inloggen



1.0 Home



1.1 Overzicht voertuigen

track

Voertuign naam

Type

Zitplaatsen

Prijs

Over dit voertuig.

Lorem Ipsum is slechts een proeftekst uit het drukkerij- en zetterijwezen. Lorem Ipsum is de standaard proeftekst in deze bedrijfstak sinds de 16e eeuw, toen een onbekende drukker een zet- haak met letters nam en ze door elkaar husselde om een font-catalogus te maken. Het heeft niet alleen vijf eeuwen overleefd maar is ook, vrijwel onveranderd, overgenomen in elektronische letterzetting.

Specificaties

Type	Benzinemotor
Type	Benzinemotor
Type	Benzinemotor
Type	Benzinemotor
Type	Benzinemotor
Type	Benzinemotor
Type	Benzinemotor

HUREN

1.1.1 Detail voertuig

track

Voertuign naam

Start date

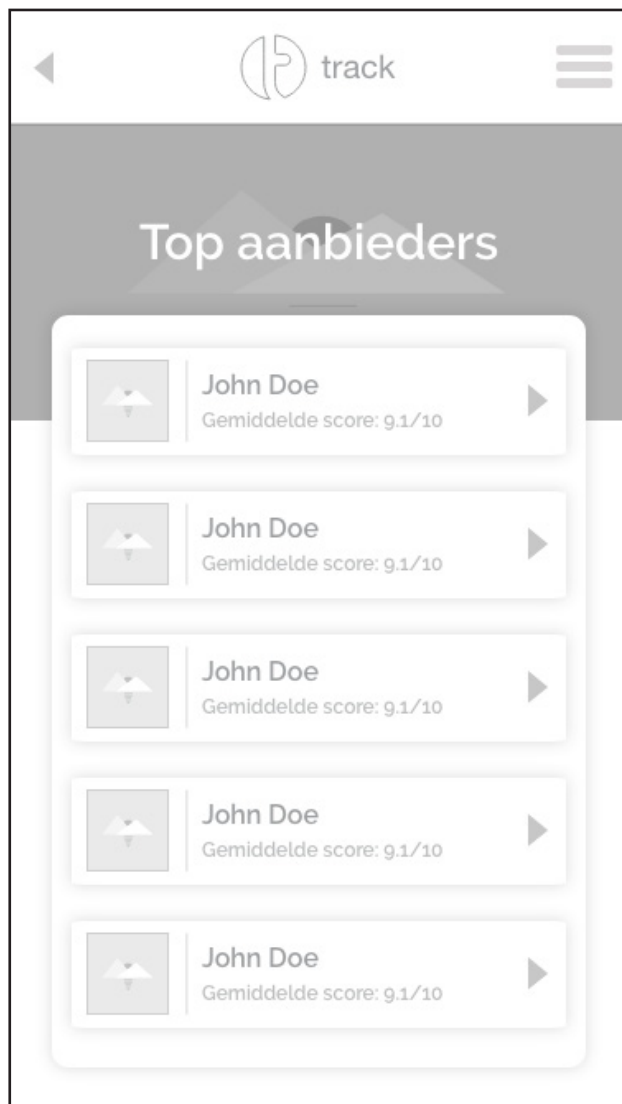
End date

12-12-17

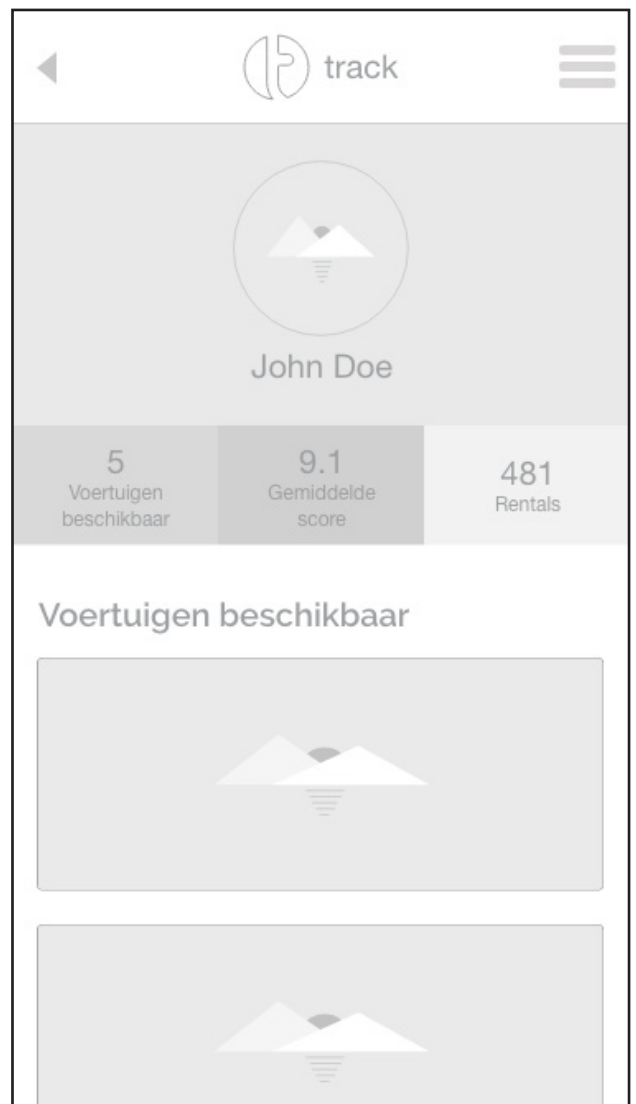
12-12-17

HUREN

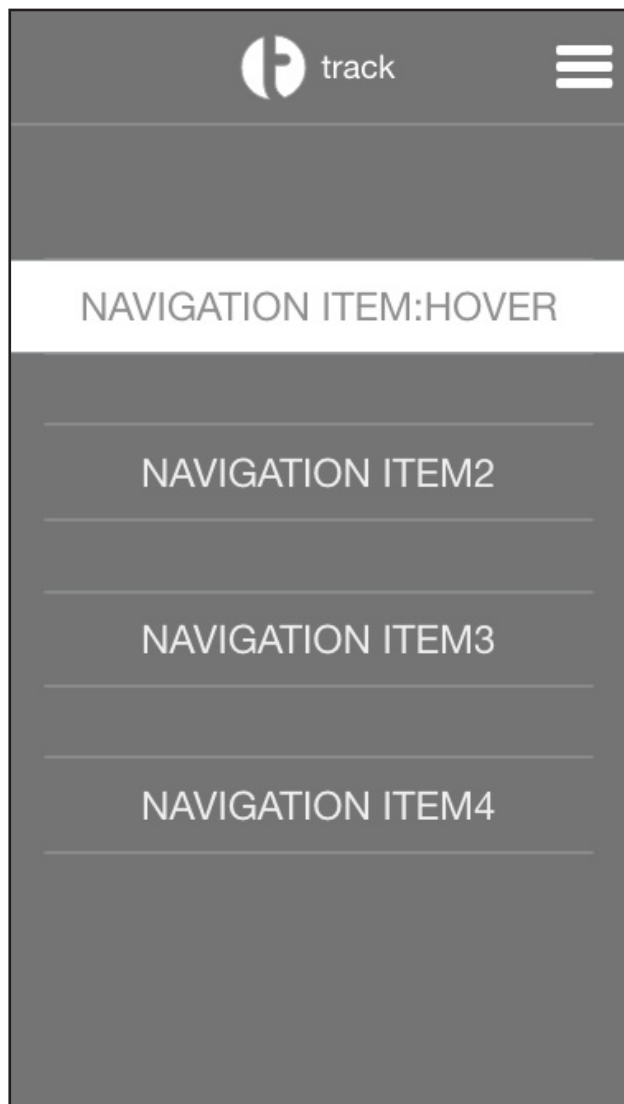
1.1.1.1 Boeking



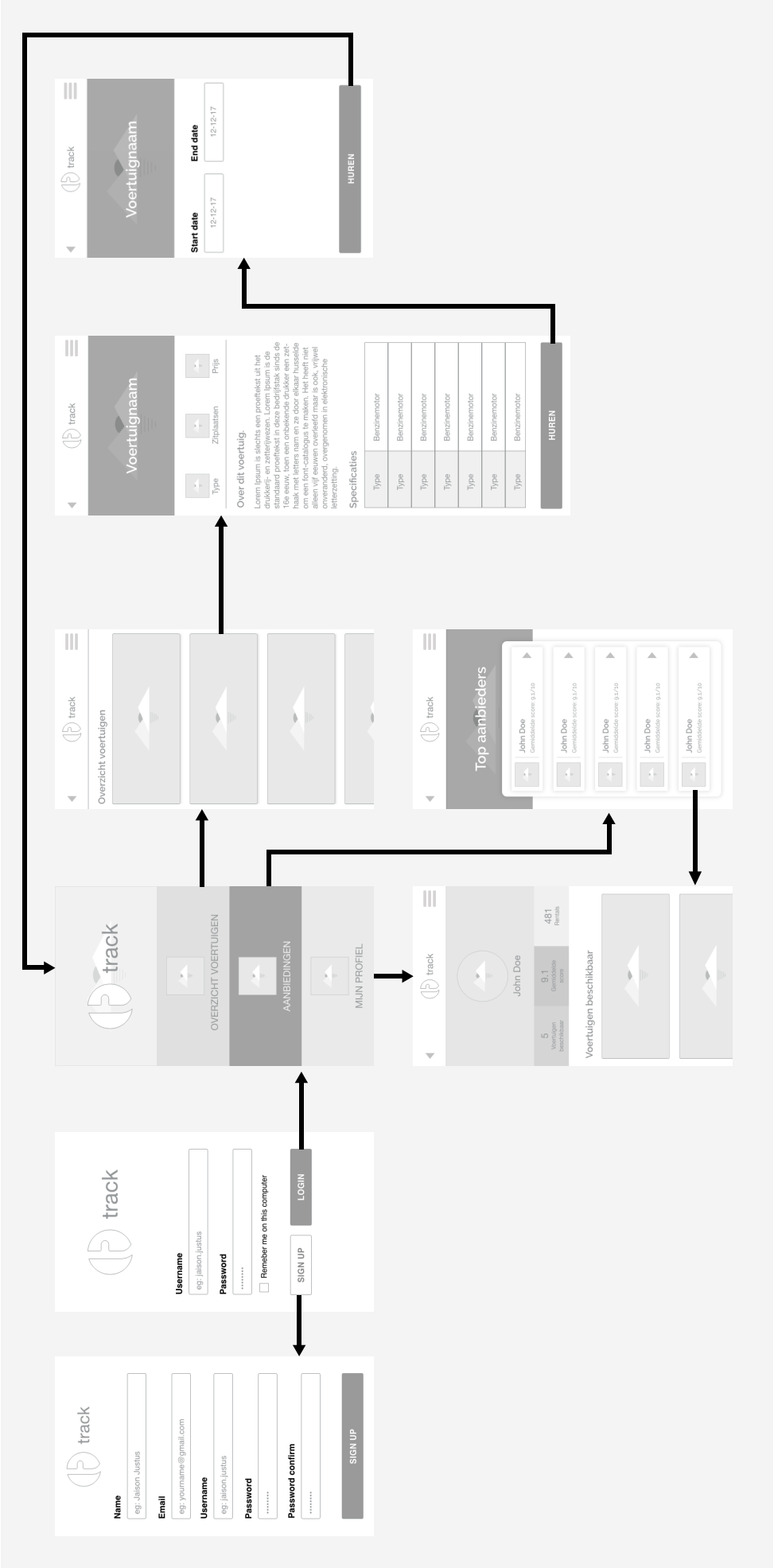
1.2 Top aanbieders



1.2.1 Detail aanbieders



Navigatie



## Style guide

### Logo's



Embleem B/W



Embleem



Logo

# track

### Kleuren

#### Primaire kleur



RGB	251-198 - 85
CMYK	1 - 25 - 74 - 0
HEX	#FAC554

#### Secundaire kleuren



RGB	45 - 45 -45
CMYK	71 - 61 - 57 - 70
HEX	#2D2D2D



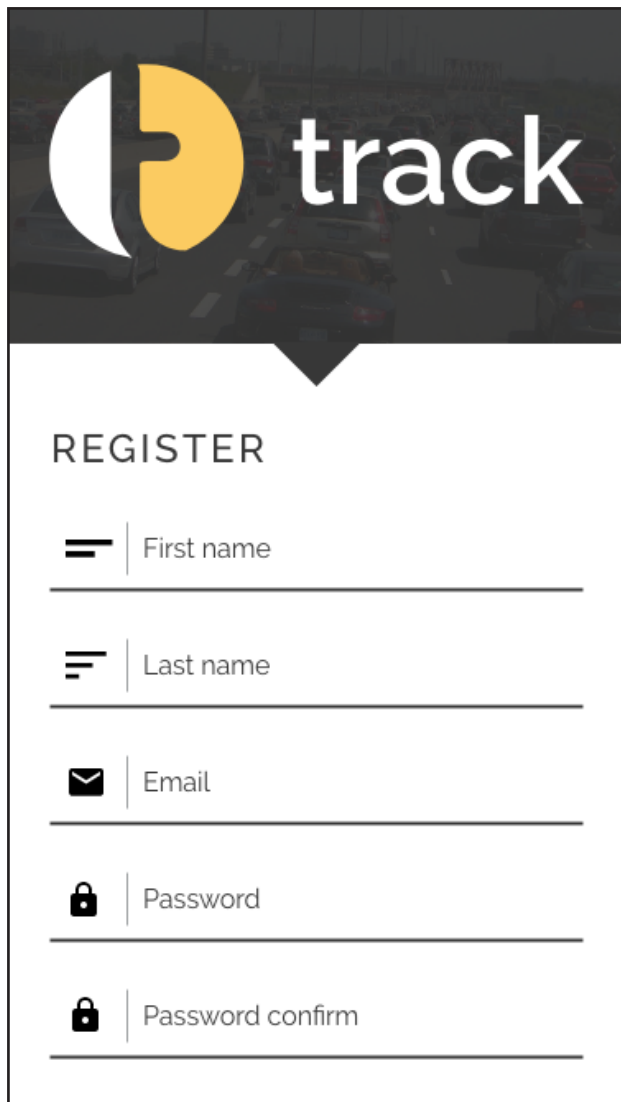
RGB	204 - 204 -204
CMYK	23 - 17 - 18 - 1
HEX	#CCCCCC

### Typografie

Voor de titels en subtitels wordt het font **Raleway** gebruikt. Voor de leestekst een open sans.

# Raleway

# Open sans



The registration screen features a dark header with the 'track' logo. Below the header, the word 'REGISTER' is displayed in a bold, sans-serif font. The form consists of five input fields, each with a small icon to its left: a horizontal line icon for 'First name', a horizontal line icon for 'Last name', an envelope icon for 'Email', a padlock icon for 'Password', and a padlock icon for 'Password confirm'. Each input field is a simple horizontal line.

**track**

**REGISTER**

First name

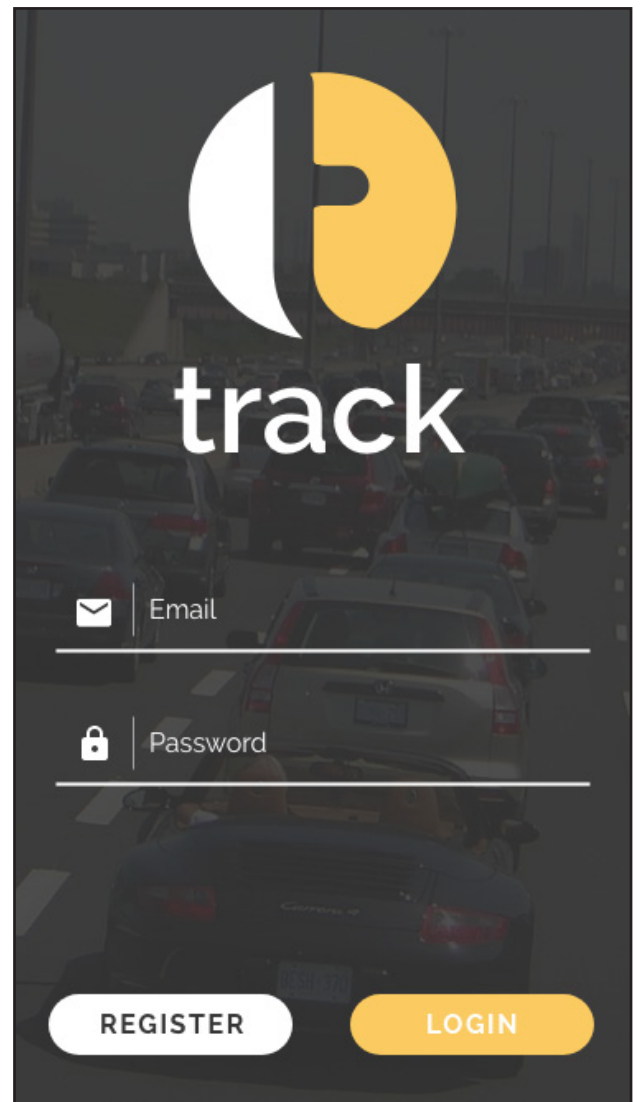
Last name

Email

Password

Password confirm

0.1 Registreren



The login screen features a dark header with the 'track' logo. Below the header, the word 'track' is displayed in a large, bold, sans-serif font. The form consists of three input fields, each with a small icon to its left: an envelope icon for 'Email', a padlock icon for 'Password', and a padlock icon for 'Password confirm'. Each input field is a simple horizontal line. At the bottom of the screen, there are two buttons: a white button with the text 'REGISTER' and a yellow button with the text 'LOGIN'.

**track**

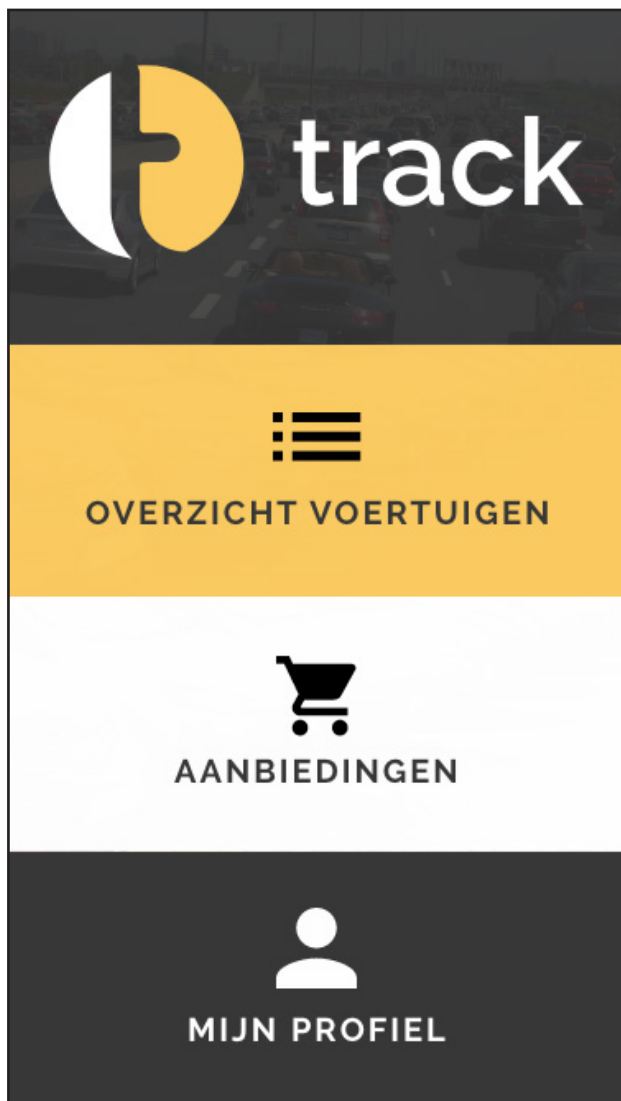
Email

Password

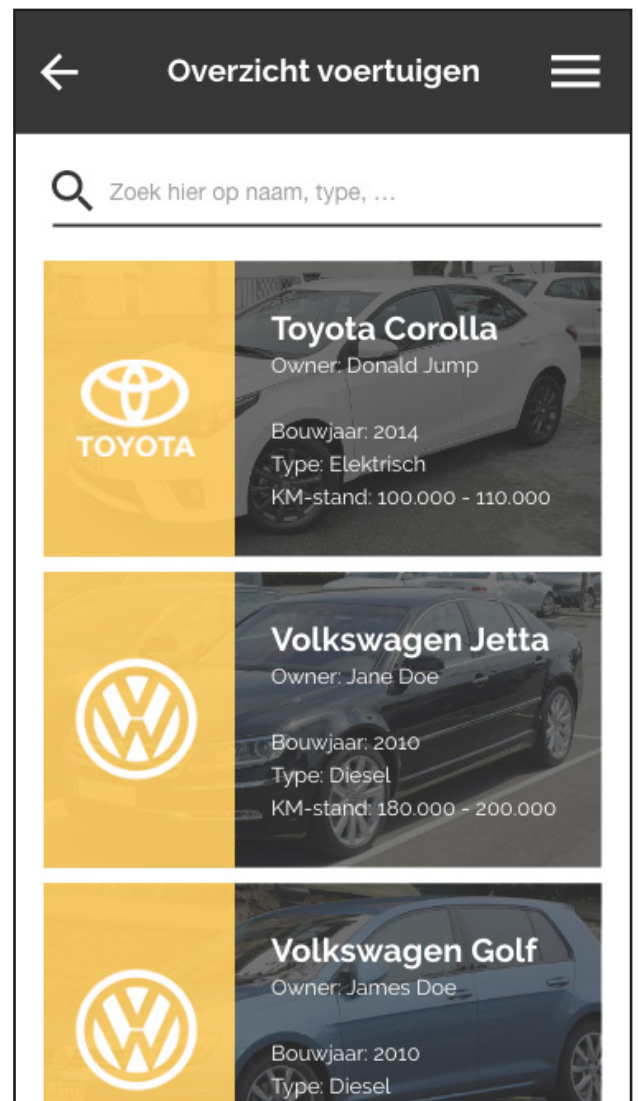
**REGISTER** **LOGIN**

0.2 Login

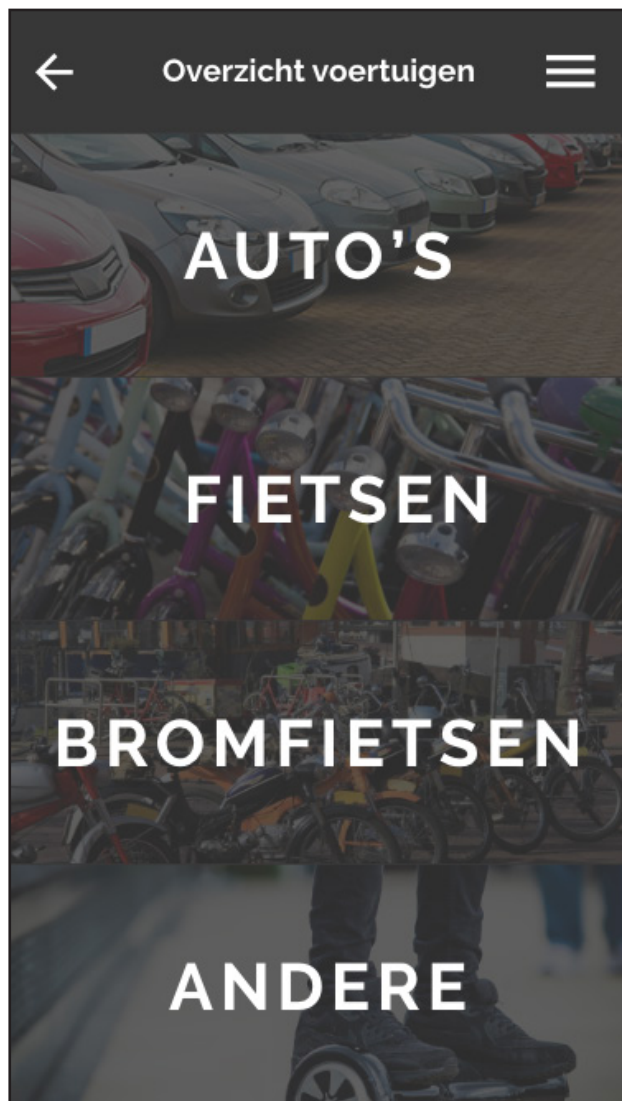




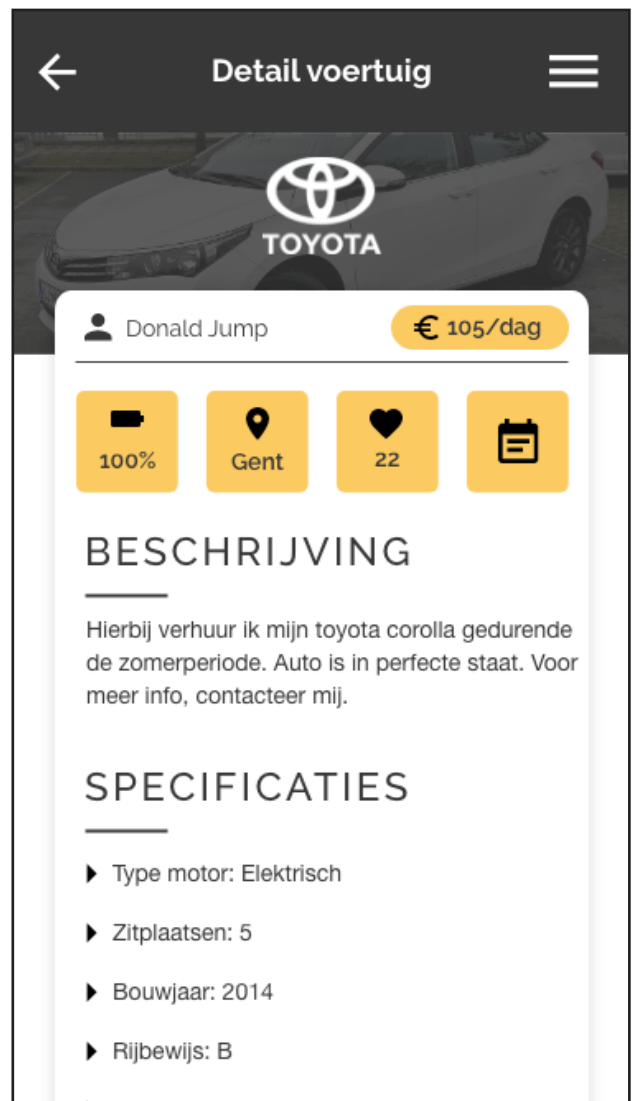
1.0 Home



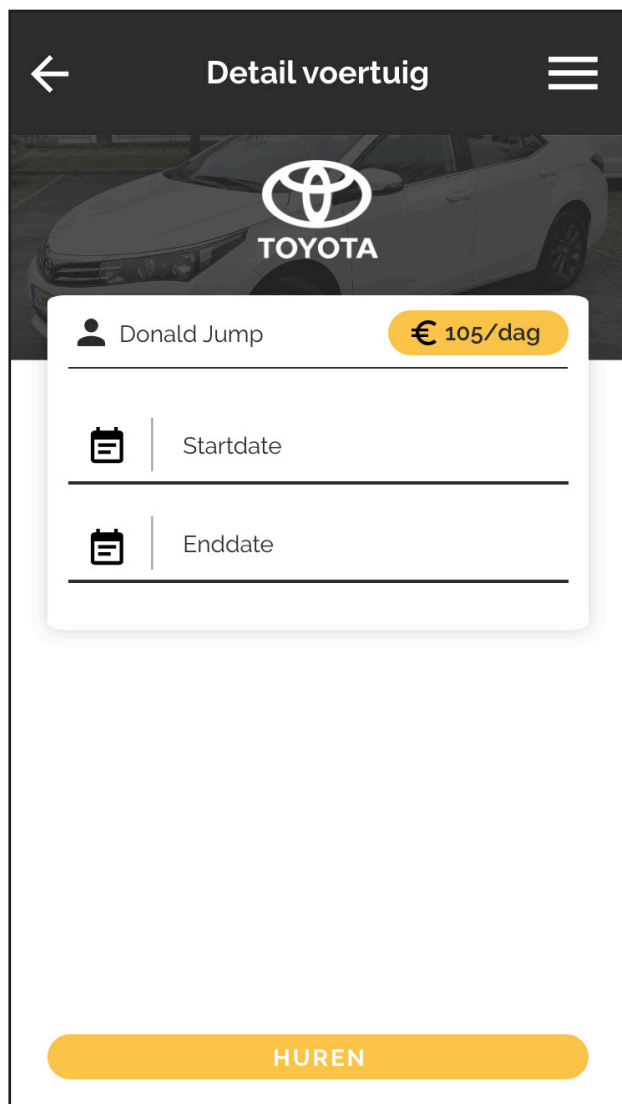
1.1 Overzicht voertuigen



1.1 Overzicht voertuigen (filter)



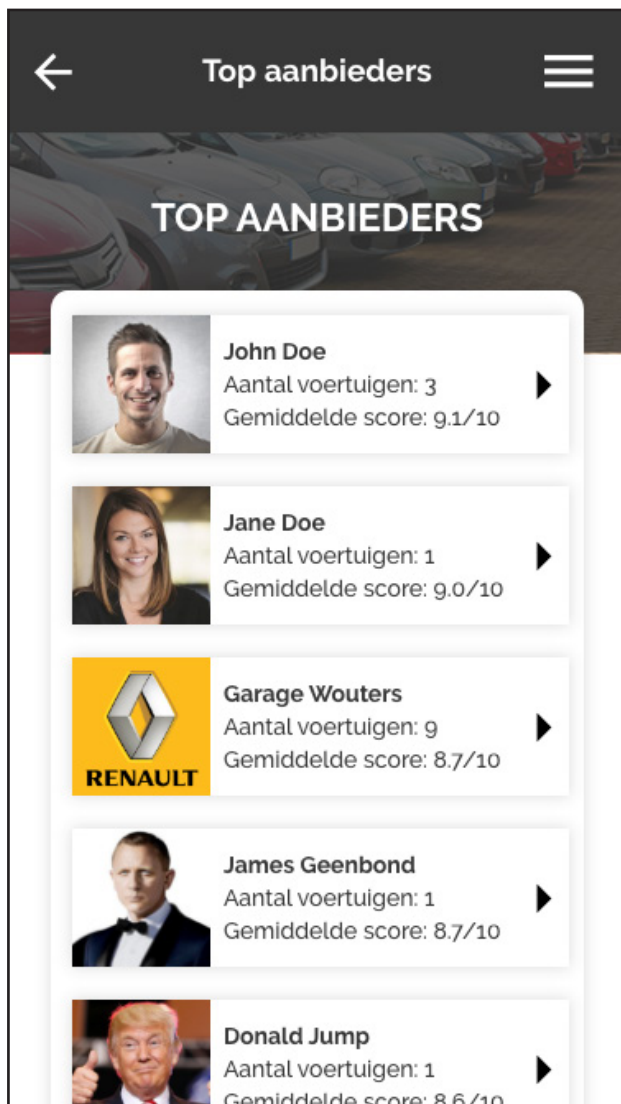
1.1.1. Detail voertuig



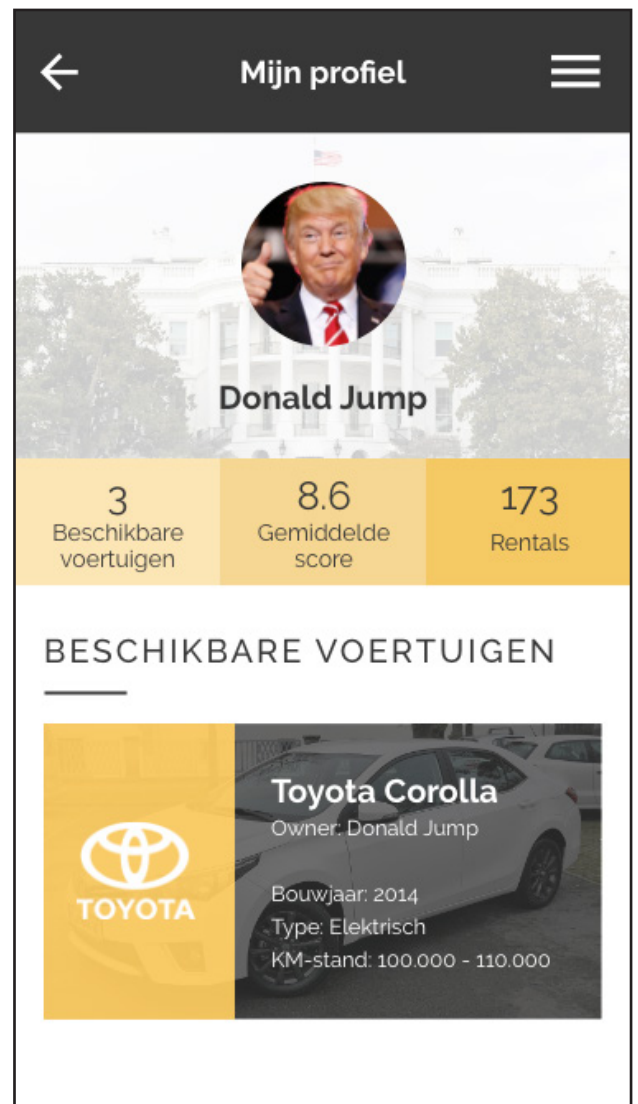
1.1.1.1 Booking



Navigatie



1.2 Top aanbieders



1.3 Mijn profiel

# DEVELOP

## Code snippets

```
MakeBooking() {
  let storage = JSON.parse(localStorage.getItem("Track-User"));
  let self = this;
  axios({
    method: 'POST',
    baseURL: 'http://cmsdev.localhost/',
    url: 'entity/booking',
    data: {
      "name": {"value": storage.current_user.name + " books vehicle #" + self.vehicle.id[0].value },
      "field_booker":{"value": storage.current_user.uid},
      "field_end_date":{"value": self.customFormatter(self.enddate) + "T00:00:00"},
      "field_start_date":{"value":self.customFormatter(self.startdate)+ "T00:00:00"},
      "field_vehicle":{"value": self.vehicle.id[0].value}
    }
  }).then(response => {
    self.UpdateVehicleBookings(response.data);
    //self.$router.push('booking/succes');
  })
  .catch( error =>{
    console.log(error.response);
  });
},

UpdateVehicleBookings(booking) {
  let vehicleId = booking.field_vehicle[0].target_id;
  let newBooking = {
    "target_id": booking.id[0].value,
    "target_type": "booking",
    "target_uuid": booking.uuid[0].value,
    "url": "/track/booking/" + booking.id[0].value
  }
  this.vehicle.field_bookings.push(newBooking);
  if(this.vehicle.changed || this.vehicle.created || this.vehicle.field_comments)
  {
    delete this.vehicle.changed;
    delete this.vehicle.created;
    delete this.vehicle.field_comments;
  }
  let car = this.vehicle;

  axios({
    method: 'PATCH',
    baseURL: 'http://cmsdev.localhost/',
    url: 'track/vehicle/' + vehicleId + '?_format=json',
    data: car
  }).then(response => {
    console.log(response.data);
  }).catch( error => {
    console.log(error);
  })
},
```

```

<paginate name="filteredVehicles" :list="filterByName" :per="5">
  <div v-for="vehicle in paginated('filteredVehicles')" class="vehicle-block">
    <router-link :to="{ name: 'vehicle-detail', params: { id: vehicle.id[0].value }}">
      
      <div class="brand">
        
      </div>
      <div class="vehicle-info">
        <h1 v-if="vehicle.name[0].value">{{vehicle.name[0].value}}</h1>
        <p>€{{ vehicle.field_price[0].value }}</p>
      </div>
    </router-link>
  </div>
</paginate>
<paginate-links for="vehicles" :limit="5" :async="true"></paginate-links>

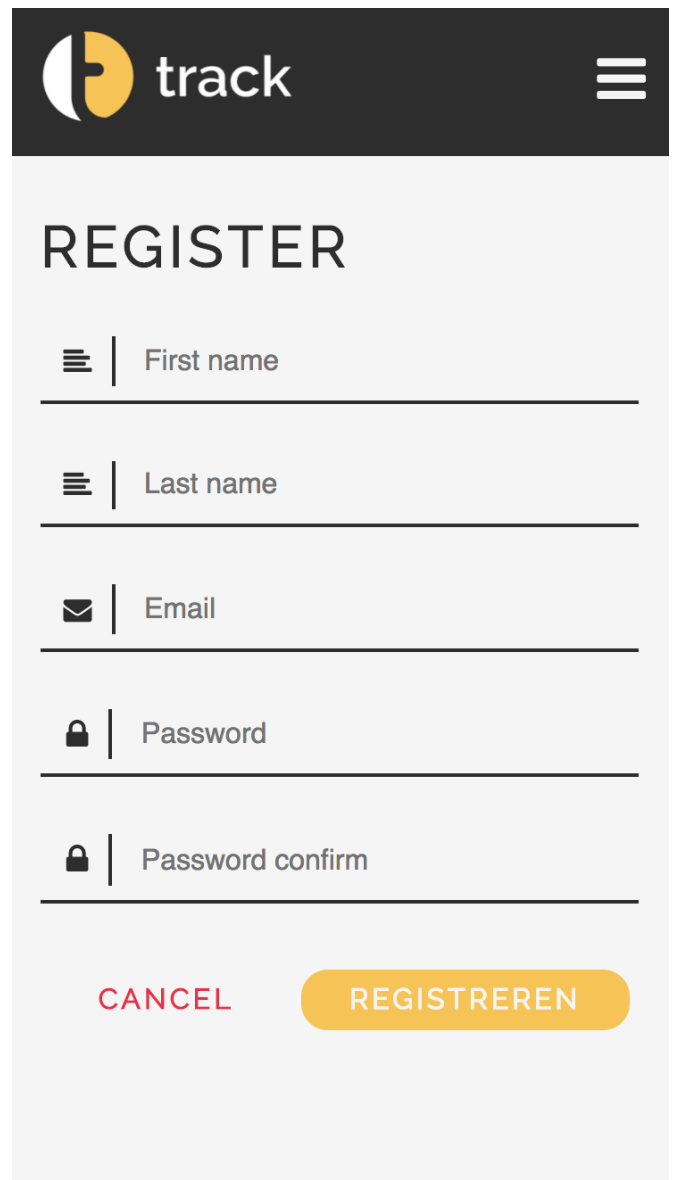
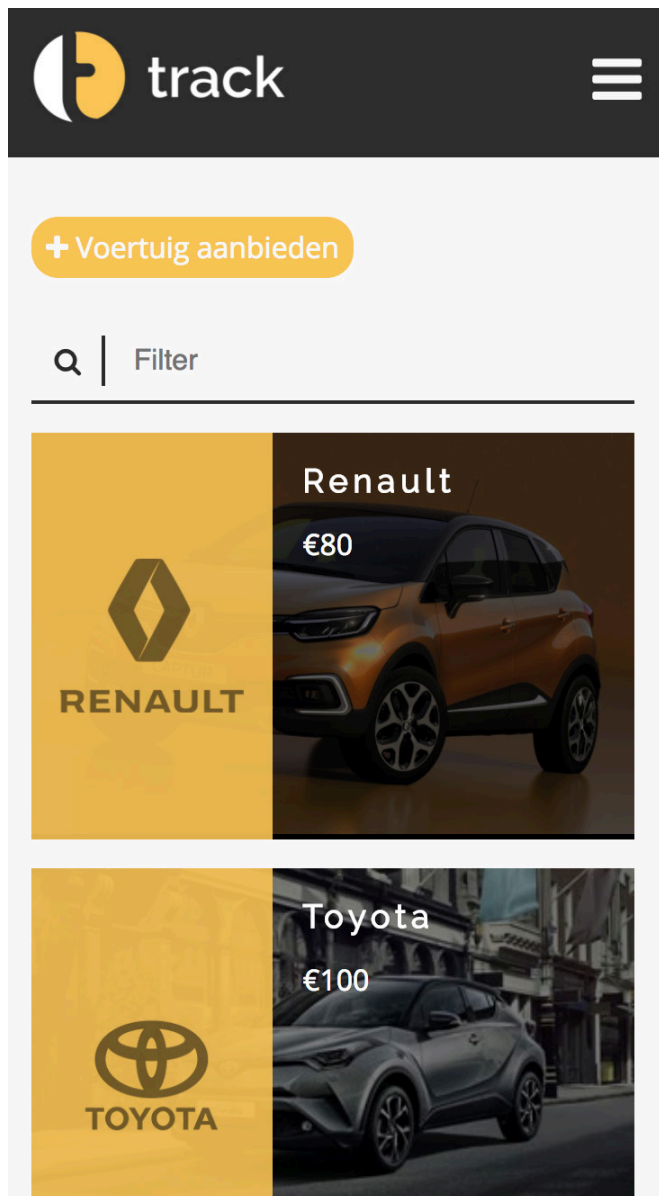
```

```

methods: {
  checkIfUserLogin() {
    if(localStorage.getItem('Track-User')) {
      this.$parent.user = JSON.parse(localStorage.getItem('Track-User'));
      this.$router.push('/home');
    }
  },
  loginUser() {
    let self = this;
    axios({
      method: 'post',
      baseURL: 'http://cmsdev.localhost/',
      url: 'user/login?_format=json',
      data: {
        name: self.mail,
        pass: self.pass
      }
    }).then(response => {
      localStorage.setItem('Track-User', JSON.stringify(response.data));
      self.$parent.user = response.data;
      self.$router.push('home');
    }).catch( error =>{
    });
  },
}

```

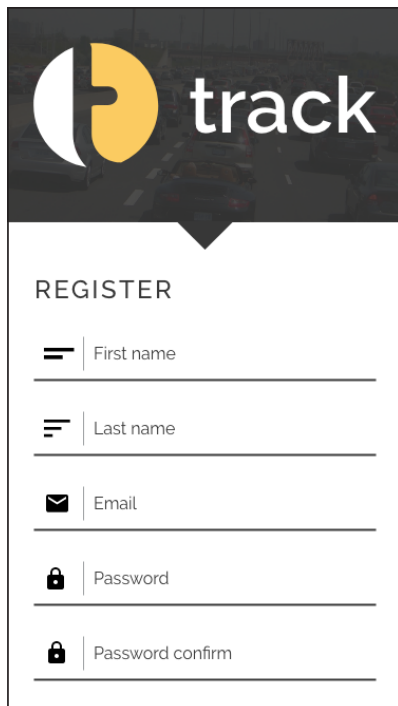
## Screenshots





# DELIVER

## Handleiding



track

REGISTER

First name

Last name

Email

Password

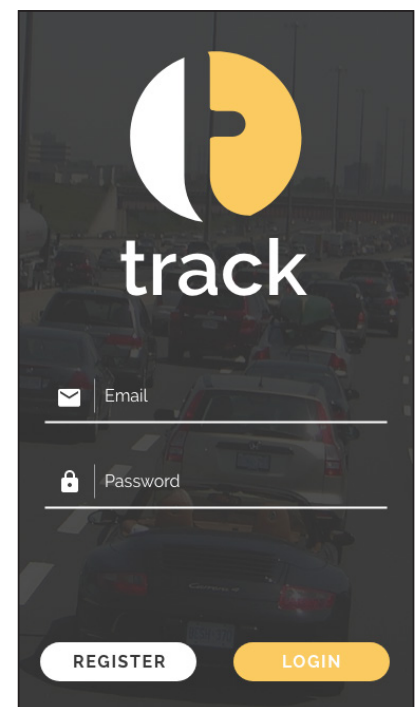
Password confirm

### Stap 1: Registreren

Vul de nodige velden in om uw profiel aan te maken. Eenmaal dit gebeurd is kan je doorgaan naar stap 2 waar je je moet inloggen.

### Stap 2: Inloggen

Na het registreren vraagt de app om je gebruikersnaam (email) en wachtwoord. Als deze overeen komen met de gegevens waarmee je je hebt geregistreerd ben je ingelogd op de applicatie. Zo niet, moet je het opnieuw proberen.



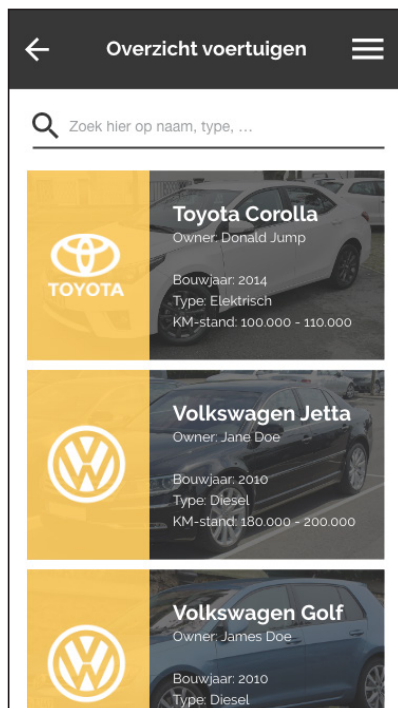
track

Email

Password

REGISTER LOGIN



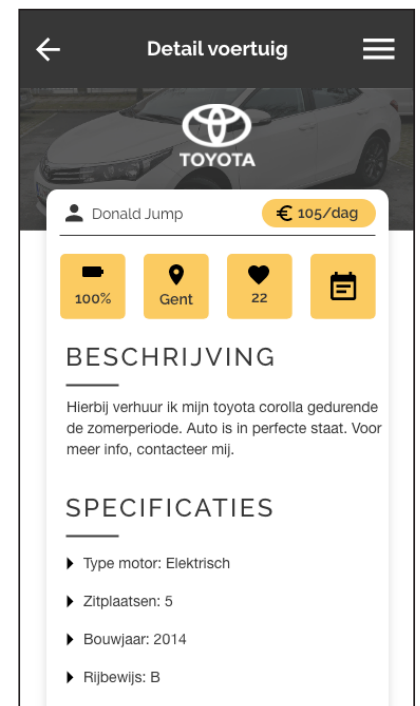


### Stap 3: Voertuig zoeken

Kies uit een volledige lijst voertuigen welk voertuig je het meest aanspreekt. Klik die aan en bekijk de details van dat voertuig.

### Stap 4: Keuze maken

Is dit het voertuig die je zoekt? Klik op de knop "boeken" om uw voertuig vast te leggen voor uw specifieke datum.



The screenshot shows a mobile app interface for car rental. At the top, there is a dark header with a back arrow on the left, the text "Detail voertuig" in the center, and a menu icon on the right. Below the header is a car image with the Toyota logo and the word "TOYOTA" underneath. A white card overlay contains the following information: a user icon and the name "Donald Jump", a yellow button with "€ 105/dag", a calendar icon and the label "Startdate", and another calendar icon and the label "Enddate". At the bottom of the screen is a large yellow button with the text "HUREN".

## Stap 5: Voertuig boeken

Bekijk de beschikbare datums en boek uw voertuig voor uw gewenste datum

# DEPLOY

## Deployment guide

Deze applicatie wordt niet online geplaatst. Om de applicatie lokaal te doen werken gebruik je volgende commands in de terminal (powershell).

```
PS: cd project  
PS: phpserve -Course cmsdev
```

Hiermee krijg je de drupal aan het runnen. Voor de front moet je een tweede terminal venster openzetten, deze hoeft niet met powershell op te starten. Je moet enkel npm (yarn) actief hebben op uw computer. Om de frontend op te starten gebruik je volgende commands:

```
$: cd project/vue  
$: yarn dev
```

Nu zijn beide lokale servers actief en kan je de backend bezoeken op:  
cmsdev.localhost

Voor de frontend bezoek je de url die via de terminal wordt meegegeven, deze is afhankelijk van de poorten die beschikbaar zijn op uw computer. Voorbeeld van een url:  
localhost:8080