

A photograph showing a woman from behind, wearing a red and white patterned dress, looking at a large painting in a museum. The painting depicts a scene with many figures, including several nude women in a pool of water. The woman is standing in front of a glass partition.

MOBILE DEVELOPMENT

# FIND YOUR PATH

SIMON DEMEYERE - ANNA DE LANGHE

# INHOUDTAFEL

## DISCOVER

Briefing	5
Wat wil de opdrachtgever?	6
Wie hoort tot het doelpubliek?	6
Welke informatie is beschikbaar?	6
Wat is de timing?	8

## DEFINE

Planning	11
Activiteiten	12
Geschatte tijdsinvestering	12
Personas / Scenario's	13
Journey map	14
Moodboard	15

## DESIGN

Musea	17
Detail	18
home	18
SiteMaps	18
Wireframes	19
Visuals	21
Mockup's	23

5

6

6

6

8

11

12

12

13

14

15

17

18

18

18

19

21

23

## DEVELOP

Routes voor Companies	25
companies in seeder	26
Companies in Controller	27
Companies in schema	28
Routes in client	29
Mappen structuur in client	30
Sass Musea	31
cards	31
Musea Card in Web	32
Musea detail in web	33
Musea list renderen in web	34
Naviagatie op Mobile	35
Musea card op mobile	36

## DELIVER

Handleiding	39
1. Musea	40
2. About	40
3. Contact	40
4. Account	41
5. Login Mobile	41
6. Musea	44
7. Menu	45
8. Overzicht bestellingen	45
9. Ticket valideren	46
10. Profiel	46
	47

## DEPLOY



DISCOVER

BRIEFING

# BRIEFING

## WAT WIL DE OPDRACHTGEVER?

Onze opdracht is AppsForGhent met als focus Hack the Museum, dus een app gericht op Gentse musea. Met de opening van het Industriemuseum en de terugkeer van het Huis van Alijn bevestigen de Gentse musea nog maar eens hun levendigheid. En dit wordt ook digitaal doorgetrokken, wat duidelijk zichtbaar is in hun publicatie van hun open data. Denk hierbij aan informatie over de collectie, tijdelijke tentoonstellingen, foto's, evenementen enzovoort.

## WIE HOORT TOT HET DOELPUBLIEK?

De applicatie die zal gecreëerd worden voor grote groepen, bijvoorbeeld een school met jonge studenten.

## WELKE INFORMATIE IS BESCHIKBAAR?

### Functionele specificaties

De functionele eisen zijn volledig afhankelijk van de noden van de frontoffice, en indien nodig ook de frontend van de backoffice. Wordt besproken in samenwerking met de docenten en opdrachtgever.

### Technische specificaties

#### Server

De server wordt geschreven met JavaScript (esnext) in Node.js (Links to an external site.)

Links to an external site. met het Express (Links to an external site.)Links to an external site. framework. Voor authenticatie en autorisatie gebruiken we het Passport (Links to an external site.)Links to an external site. framework. Binnen Passport zullen we minimaal een local-strategy, jwt-strategy (Links to an external site.)Links to an external site. en een extra strategy (Links to an external site.)Links to an external site. (bv: facebook-strategy, google-strategy, twitchtv-strategy, ...) aanbieden.

De server is voorzien van een API (version controlled). Voor ieder collectie uit de database maken we een corresponderende controller, schema, validatie en routes. De volgende actiemethoden zijn minimaal in de controller aanwezig: index (alle documenten uit een collectie tonen, inclusief pagination), show (GET), create, store (POST), edit, update (PUT), delete (DELETE), softdelete (PATCH) en softundelete (PATCH).

Alle API-endpoints worden gedocumenteerd via de OpenAPI-specificatie (Links to an external site.)Links to an external site.m.b.v swagger-js-doc en swagger-ui-express (of ReDoc). Beveilig de swagger-ui-express en dus ook de API-endpoints met JWT. De API is afgeschermd voor niet-bevoegden en voldoet aan de normale eisen voor beveiliging.

Zorg voor een globale foutafhandeling, logging en custom 404 en Error pages (m.b.v. ejs (Links to an external site.)Links to an external site.template engine).

De client wordt geïnjecteerd in de root route: `http://{your domain}:{your port}/`. De client wordt niet gerenderd op de server, maar wel de informatie in de <head>. Op deze manier wordt de app SEO-friendly en ook om te book-

marken via OpenGraph e.d.. SSR (Links to an external site.)Links to an external site. is ook mogelijk, maar combineren met PWA (Links to an external site.)Links to an external site. is niet eenvoudig.

### Database

We gebruiken MongoDb als databasesysteem. Deze databank moet gehost worden op mLab (Links to an external site.)Links to an external site. of MongoDb (Links to an external site.)Links to an external site.. Gebruik op de campus of binnen het netwerk van de Arteveldehogeschool de wifi "ArteveldeHS Open".

Voor een team dat bestaat uit:

- \* 1 lid, moeten minimaal 7 collecties (tabellen) aanwezig zijn, exclusief users en roles.
- \* 2 leden, moeten minimaal 9 collecties (tabellen) aanwezig zijn, exclusief users en roles..
- \* 3 leden, moeten minimaal 11 collecties (tabellen) aanwezig zijn, exclusief users en roles.

De volgende relaties tussen documenten (en dit zonder embedded documents) moeten minstens eenmaal voorkomen:

- \* 1..1 (één op één)
- \* 1..N (één op veel)
- \* N..N (veel op veel)

De volgende Unit Tests (Mocha, Chai, Sinon) moeten minimaal aanwezig zijn voor een team dat bestaat uit:

- \* 1 lid: tests voor een minimaal 1 controller, tests voor de auth controller, tests voor minimaal 1 mongoose model
- \* 2 leden: tests voor een minimaal 2 controllers, tests voor de auth controller, tests voor minimaal 2 mongoose models

- \* 3 leden: tests voor een minimaal 3 controllers, tests voor de auth controller, tests voor minimaal 3 mongoose models

### Client

De cliënt die geïnjecteerd wordt in de server is een PWA React-client met gedeeltelijke SSR (volledig kan ook). De cliënt bevat ook state management via Redux (Links to an external site.)Links to an external site..met Redux middleware (Thunk, Sagas of Epics) . Het router-systeem dat we zullen gebruiken binnen React is react-router (Links to an external site.)Links to an external site..

Styling wordt binnen React toegevoegd via scss en / of via styled components (Links to an external site.)Links to an external site..

De volgende Unit Tests (Jest, fetch-mock, expect, Sinon) moeten minimaal aanwezig zijn voor een team dat bestaat uit:

- \* 1 lid: tests voor een minimaal 1 component, tests voor 1 redux store en middelware
- \* 2 leden: tests voor een minimaal 2 componenten, tests voor 2 redux stores en middelware
- \* 3 leden: tests voor een minimaal 3 componenten, tests voor 3 redux stores en middelware

## Werkwijze

Na het Algemeen Concept doorloop je deze stappen in volgorde:

\* Requirements

De opgegeven minimumeisen vul je zelf verder aan in functie van het concept.

\* Server

» Data Modeling

Deze stap is gemeenschappelijk met de backoffice en API.

» Coderen

- Functionaliteiten programmeren in Node.js (Express + Mongoose).

- (Migrations)

- Seeds

- Models

- Controllers

- Routes

- Tests runnen.

\* Client

» Coderen

- Functionaliteiten programmeren in React.

- Components

- Pages

- Routes

- Tests runnen.

## Mappenstructuur

- \* app (folder)
- \* docs (folder for GitHub pages)
  - » assets (folder)
    - css (folder)
    - images (folder)
    - js (folder)
    - presentation (folder)
    - academische-poster.pdf
    - dossier.pdf (Maak een digitaal productiedossier dat alle stappen in de Workflow (Links to an external site.)  
Links to an external site. documenteert)
    - screencast.mp4
  - » index.html (landing page)

## WAT IS DE TIMING?

ten laatste op 16-06-2019 om 23:59

- \* Via GitHub. Geef rechten aan docent Philippe De Pauw
- \* Via Canvas. Zip het geheel en uploaden via <https://arteveldehogeschool.instructure.com/courses/3681/assignments/18784>



# DEFINE

PLANNING - PERSONAS - SCENARIOS - JOURNEY MAPS - MOODBOARD

DESIGN

IDEATE

RESEARCH

# PLANNING

## ACTIVITEITEN

- \* Voorstudies
  - » De briefing horen waarin de eisen worden vermeld.
  - » Inschattingen maken over de tijdsduur
  - » Startdocument maken
  - » Inspiratie opdoen op het internet in verband met het design.
  - » De sitemap in elkaar steken
  - » Wireframes schetsen op basis van de sitemap
  - » Van de wireframes een visueel concept maken.
- \* Alle functies in de website plaatsen
  - » Login systeem/ registreren
  - » Navigatie maken
  - » Musea API
  - » Content aan MongoDB geven
  - » tickets bestellen (orders)
  - » Detailpagina's museums
  - » CSS aan geven

## GESCHATTE TIJDSINVESTERING

- \* 1 week ontwerpen en voorstudies maken.
- \* 4 weken concentreren op de functies en met Mern-stack leren werken
- \* 1 week stijlen en laatste bijwerkingen toevoegen

Elke dag ongeveer 6 uren gerekend.

**Totale tijdsinschatting: 6 weken ( 252 uren)**

# PERSONA'S / SCENARIO'S

## Persona Eddie Russell



Gedreven flexibel Georganiseerd

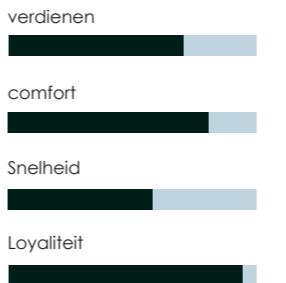
### Gegevens

Tel: 208-812-6623  
E-mail: eddie.russell99@example.com  
Geboortedatum: 9/2/1968  
Adress: 4081 Bus Ave  
Job: Leerkracht Middelbare school  
status: Familie  
instresses: autosport  
inkomen: 2200 maand

### Tijdsbesteding

Spendeert het meeste tijd aan:  
Websites: Facebook, musea.be, gent.be  
Apps: WhatsApp, twitch, Candy Crush  
Sport: tennis, fitness  
resterende tijd: aandacht schenken aan kennis opdoen.  
werk veel op: laptop, smartphone, tablet

### Motivatie



### Goals/Frustrations

- Goals:**
- Werkt hard voor de **centen**. Hij wilt graag zijn dochters een mooie toekomst geven.
  - Wilt graag meer leren over verschillende zaken.** Van geschiedenis tot moordzaken oplossen
  - Zijn **studenten** de tijd van hun leven geven
- Frustrations**
- Mensen die werk **niet serieus** beschouwen.
  - Geen goede klik** met de studenten
  - Studenten die niet beseffen hoe leuk het is een museum uitstap te doen**

### Scenario

Eddie is een leerkracht die graag veel bijleert. Hierdoor vind hij niets beter dan een museum bezoeken met studenten. Het is natuurlijk niet altijd evident om in groep te gaan naar een museum en al de ticketen aan de studenten uit te delen. Soms verliest een studenten een ticket. En al die studenten zijn toch maar bezig op die gsm dat ze geen besef hebben van hoe interessant het rondom hen kan zijn.

## Persona Judith Crawford



vrijgevig creatief Pasief

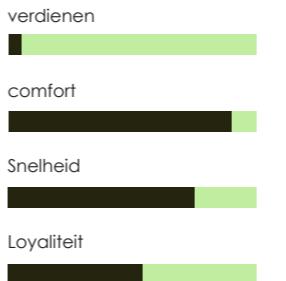
### Gegevens

Tel: 265-876-4885  
E-mail: judith.crawford69@example.com  
Geboortedatum: 1/7/1998  
Adress: 4960 Cavour Ave  
Job: Student  
status: Single  
instresses: Movie Make-Up Artist  
studie: Producer

### Tijdsbesteding

Spendeert het meeste tijd aan:  
Websites: mijn Dinar, twitter, instagram  
Apps: Canvas, Snapchat, Youtube  
Sport: Yoga  
resterende tijd: dictielessen, viool, studeren  
werk veel op: computer, smartwatch, smartphone

### Motivatie



### Goals/Frustrations

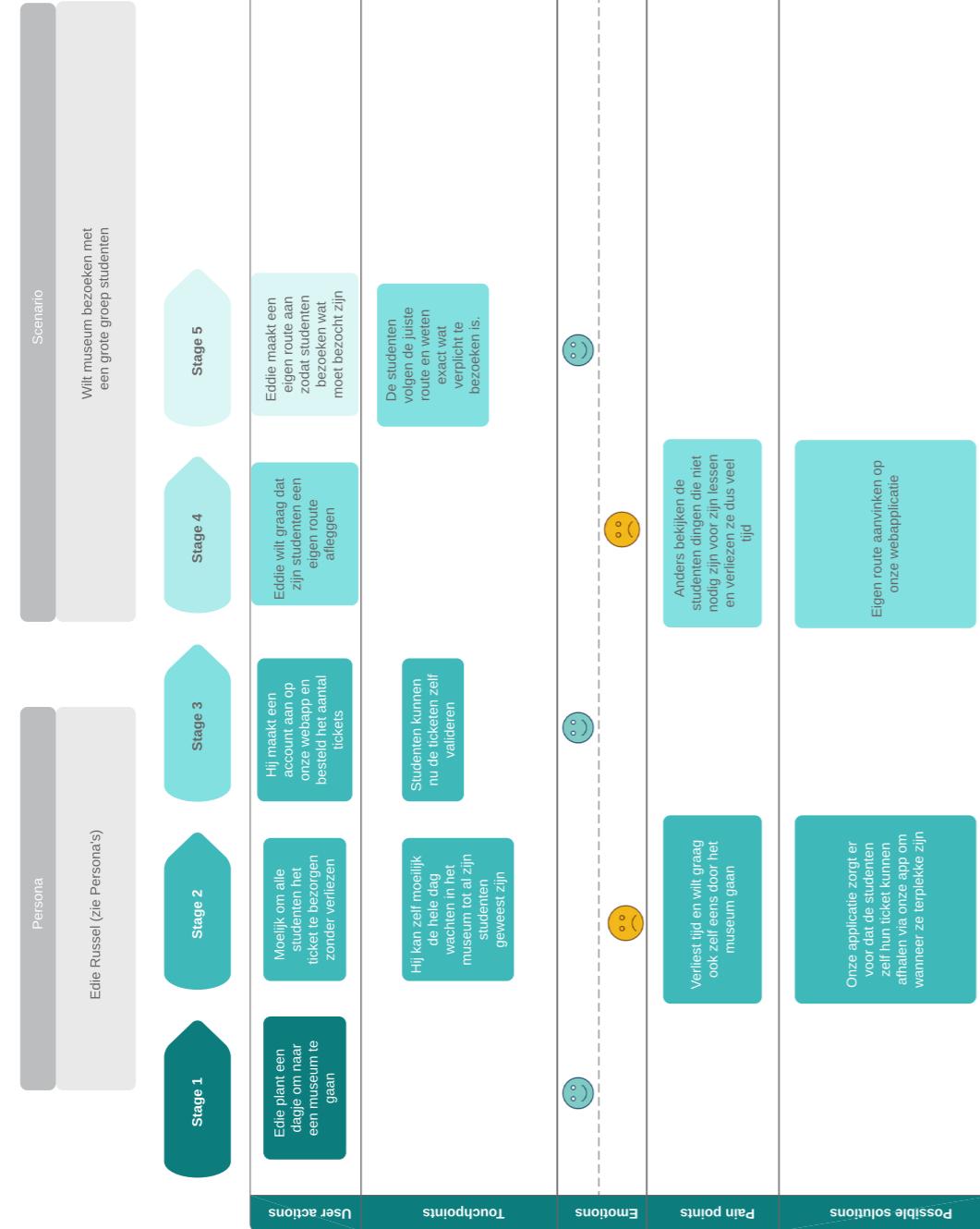
- Goals:**
- Goede studies met als doel een goede job.
  - Make-Up Artist worden in eigen gemaakte film.
  - Toneel voeren.
  - goede woning
  - gezin hebben
- Frustrations**
- Vakken studeren waar ze eigenlijk niet in geïnteresseert
  - Tijd verliezen door uitstappen

### Scenario

Judith studeert nog en weet heel goed wat ze wilt worden later. Het enige probleem is dat ze bij de batchlor steeds vakken krijgt die niet belangrijk zijn voor haar doel later. Zo moet ze continue museums bezoeken van haar docent. Ze vindt dat dit veel tijdsverlies heeft omdat ze nooit snel haar weg terug vind in een museum. Aan ticketen geraken is een groot probleem, en tegen dat ze gevonden heeft wat belangrijk is heeft ze het museum al 10 keer doorlopen. Ze vindt het heel vervelend om zoveel tijd te verliezen. Ze had wel wat betere dingen kunnen doen in het weekend.

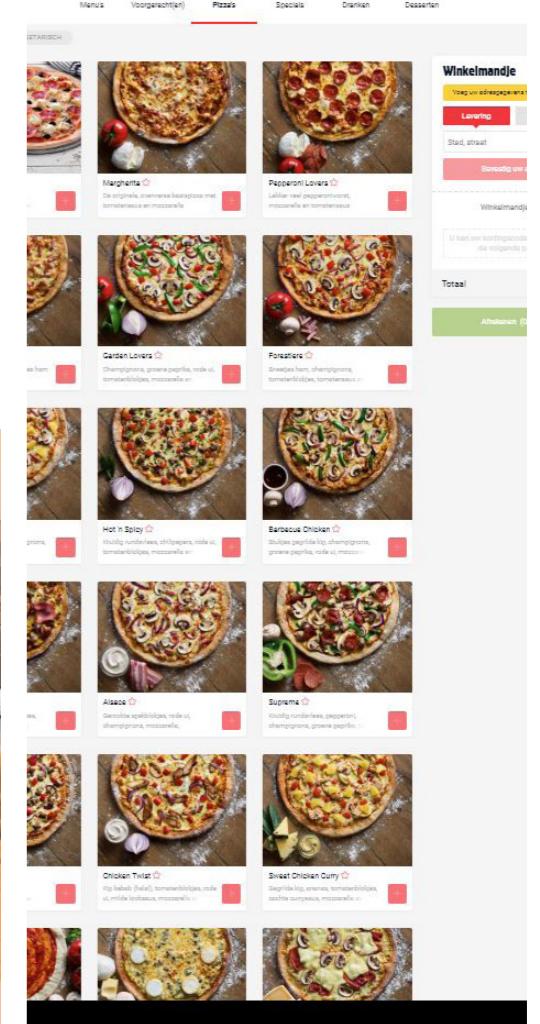
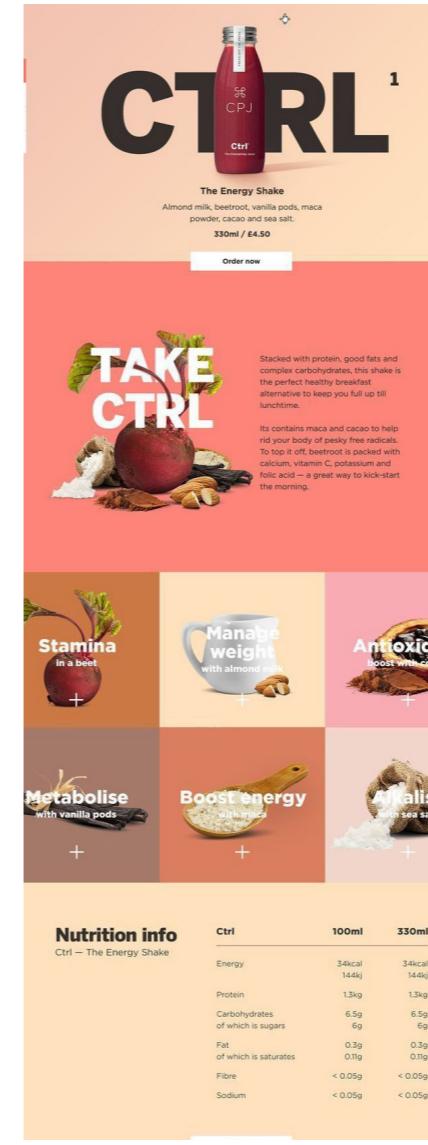
## Customer Journey Map

Anna De Langhe | June 15, 2019



## JOURNEY MAP

## MOODBOARD

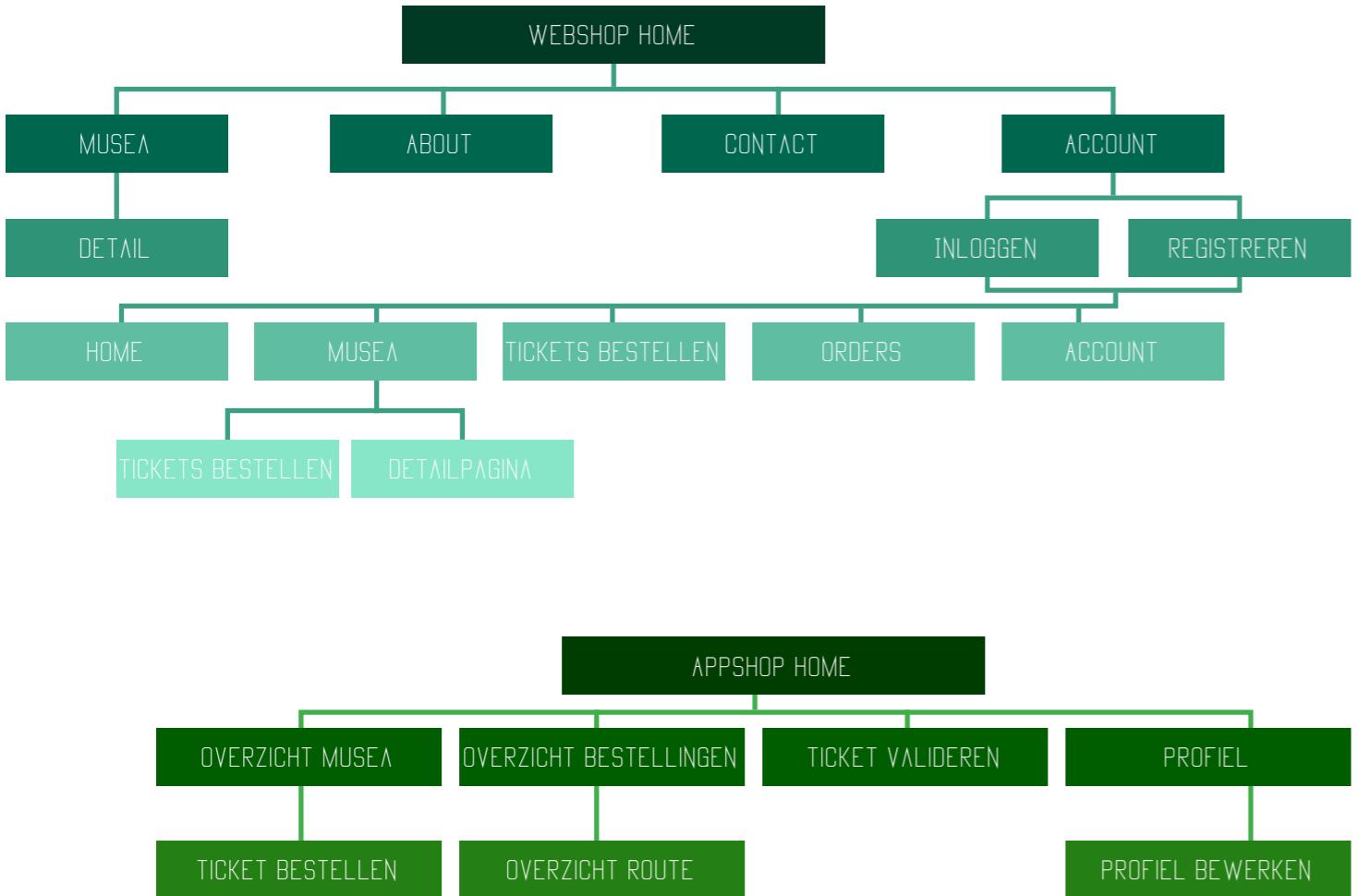


# DESIGN

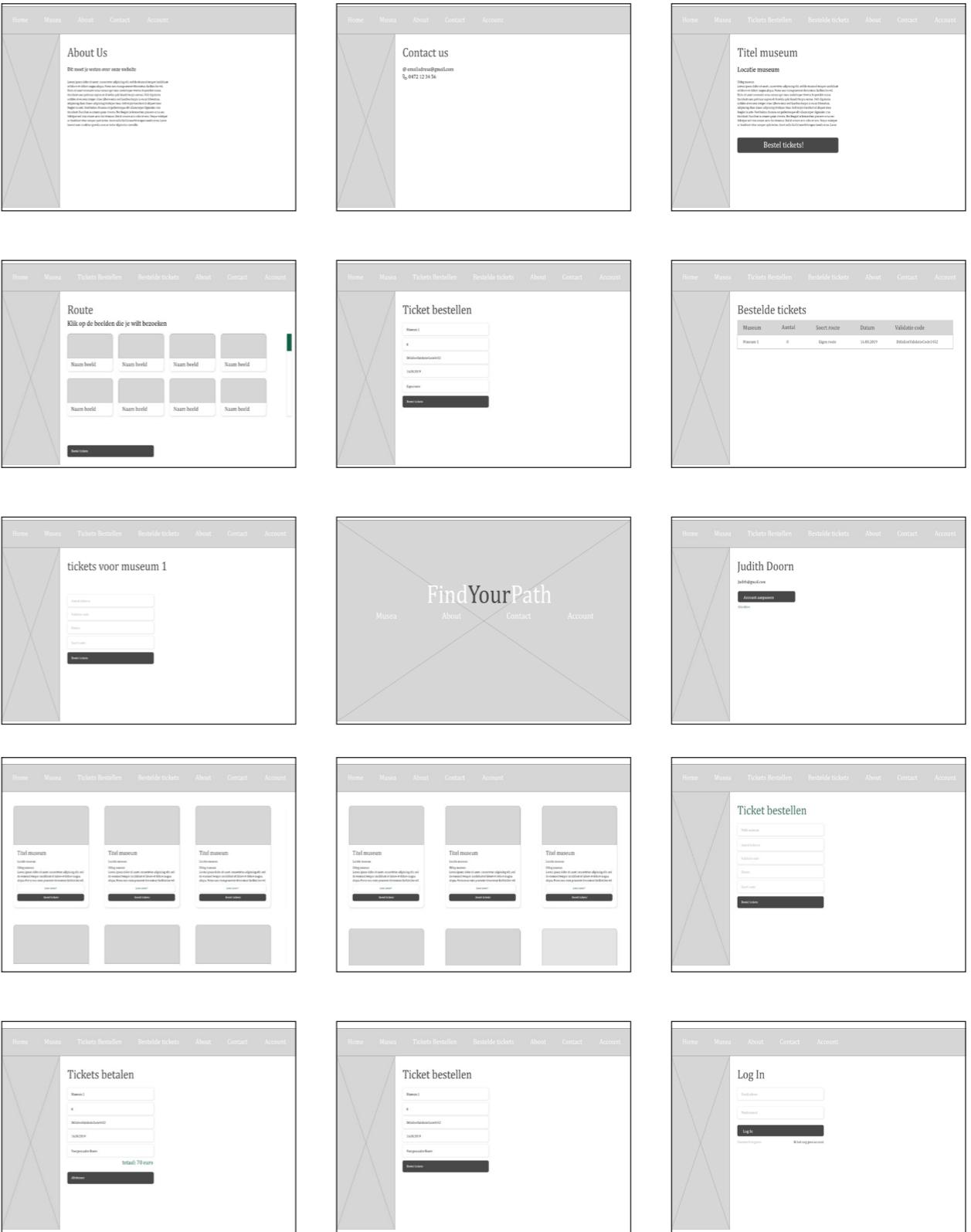
SITEMAP - WIREFRAMES - VISUALS - MOCKUPS



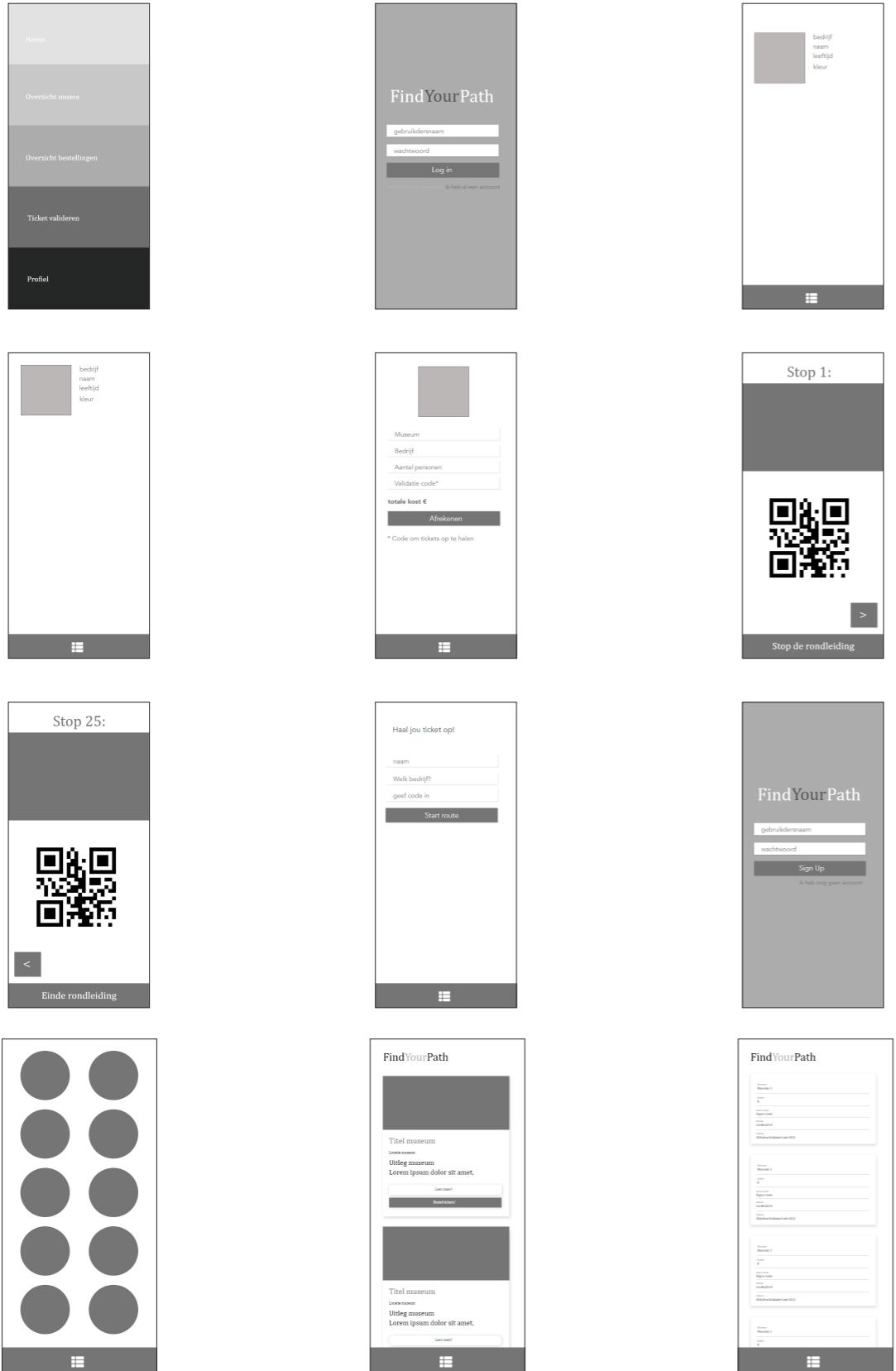
# SITEMAPS



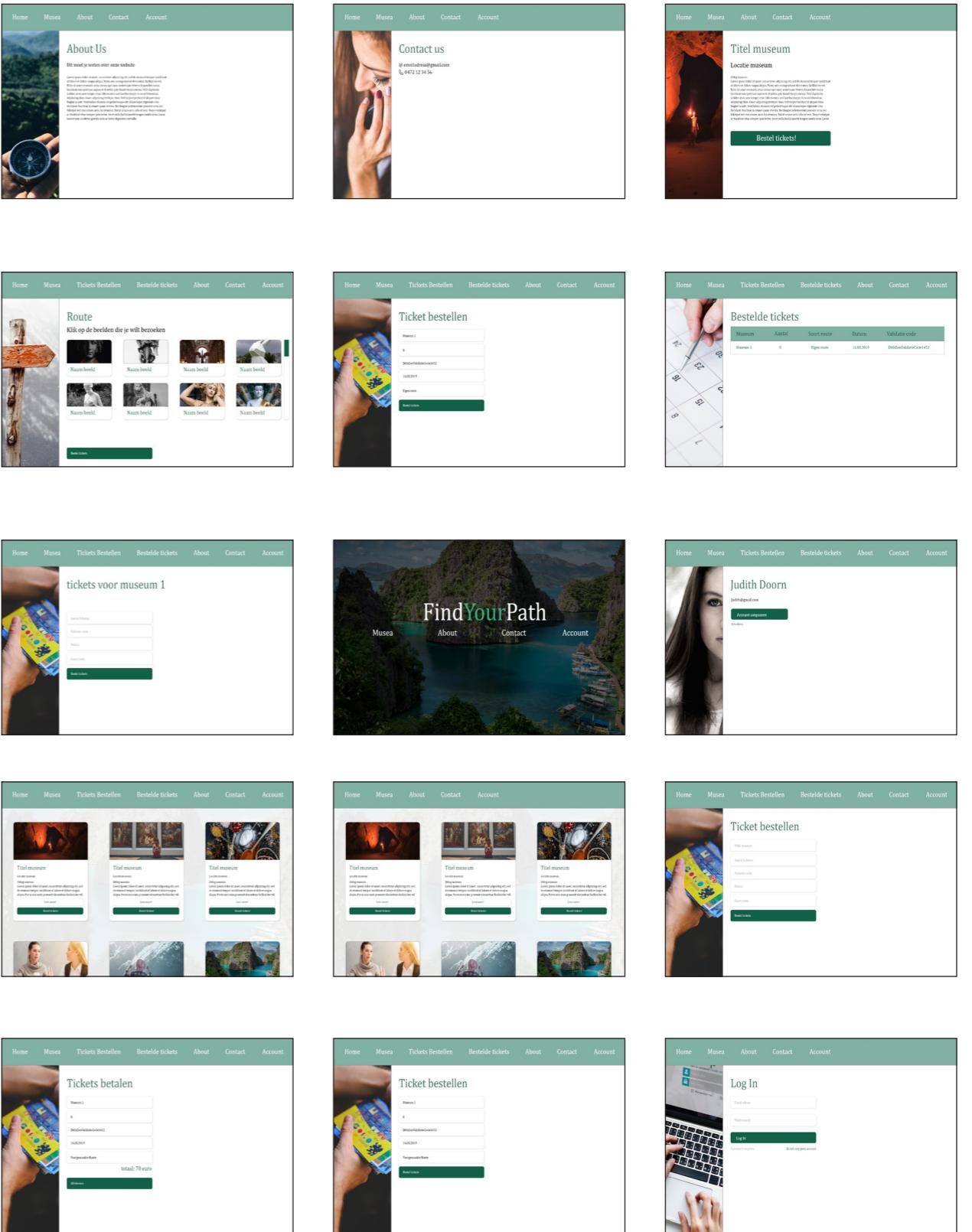
# WIREFRAMES - WEB



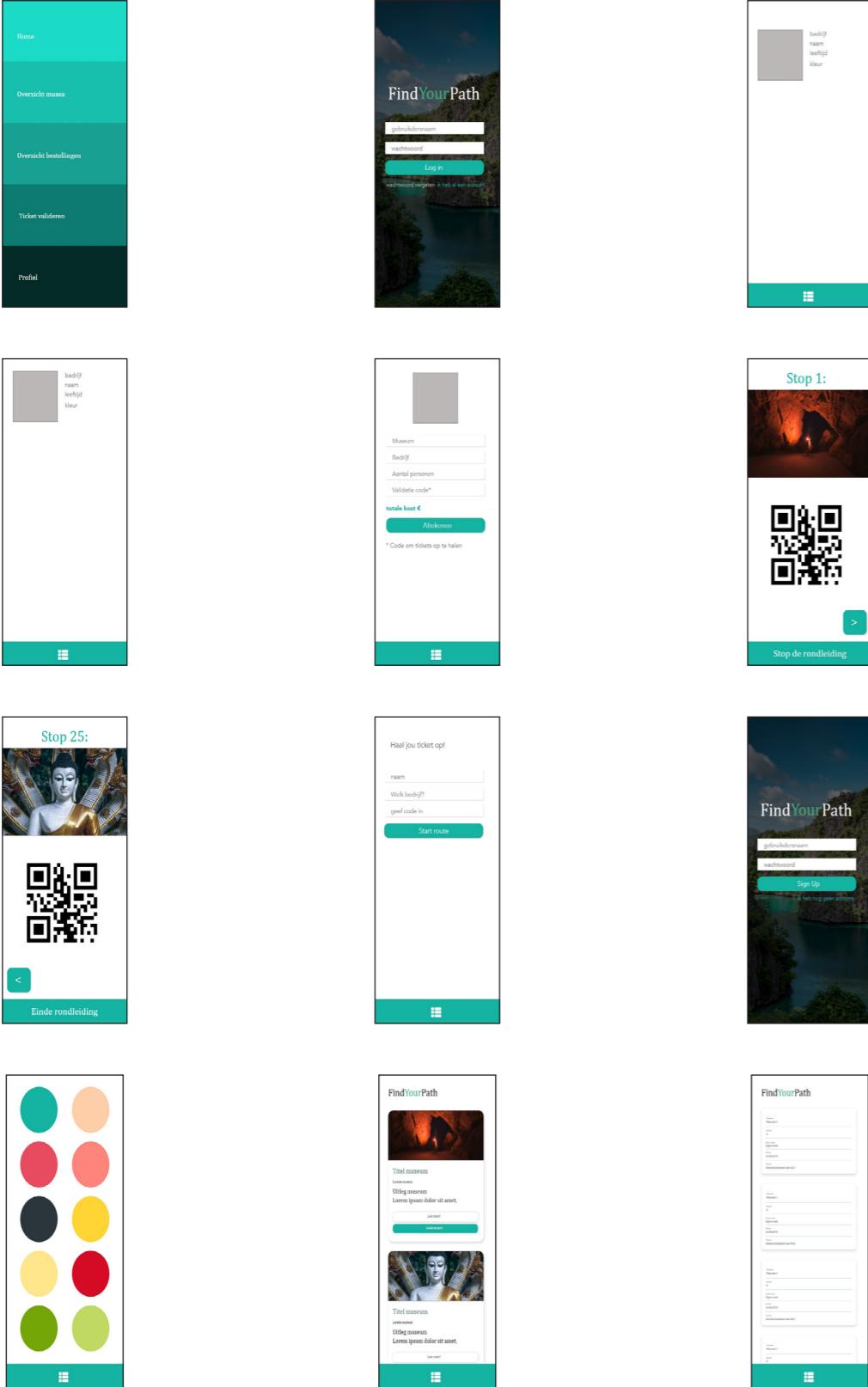
## WIREFRAMES - MOBILE



## VISUALS - WEB



## VISUALS - MOBILE



22

## MOCKUP'S



23

# DEVELOP

SNIPPETS

```
g _config['code'],
image_src
('hex2rgb') ) {
$hex_str, $return_string = false, $separator =
eg_replace("/[^0-9A-Fa-f]/", '', $hex_str); //array();
$x_str) == 6 ) {
= hexdec($hex_str);
['r'] = 0xFF & ($color_val >> 0x10);
['g'] = 0xFF & ($color_val >> 0x8);
['b'] = 0xFF & $color_val;
n($hex_str) == 3 ) {
[r'] = hexdec(str_repeat(substr(
[g'] = hexdec(str_repeat(substr(
[b'] = hexdec(str_repeat(substr(
node($se
```

## ROUTES VOOR COMPANIES

```
1  /*
2   Import the internal libraries:
3   - CategoryController
4   */
5 import {
6   CompanyController,
7 } from '../controller';
8
9 // Create instance of CategoryController otherwise you can't use it
10 const companyController = new CompanyController();
11
12 // authService --> can be used as parameter
13 const initializeEndpoints = (parentRouter) => {
14 /**
15  * @swagger
16  * /api/v1/categories:
17  *   get:
18  *     tags:
19  *       - Categories
20  *     description: Returns all categories
21  *     produces:
22  *       - application/json
23  *     responses:
24  *       200:
25  *         description: An array of categories
26  */
27 parentRouter.get('/companies', companyController.index);
28 /**
29  * @swagger
30  * /api/v1/categories/create:
31  *   get:
32  *     tags:
```

## COMPANIES IN SEEDER

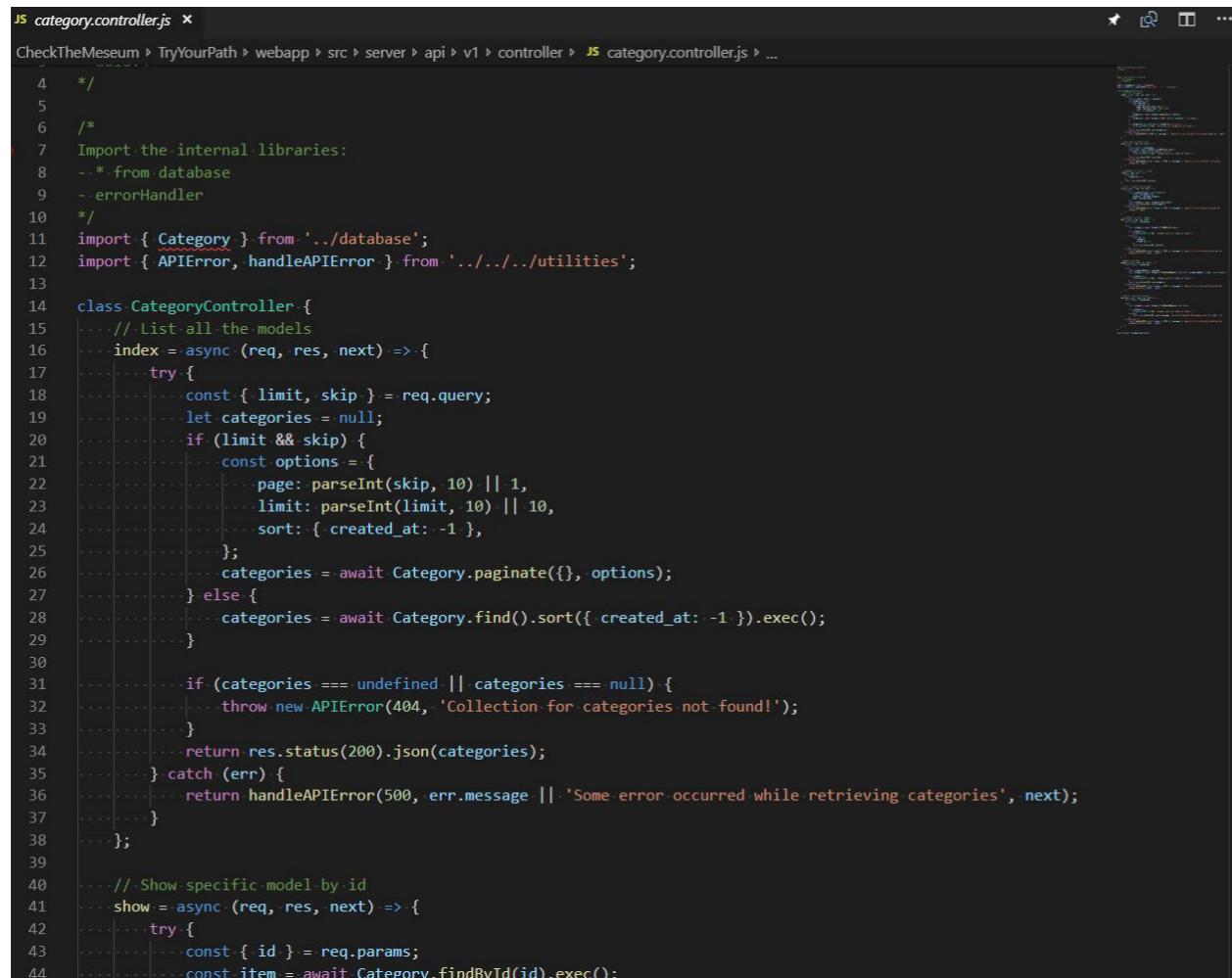
```
companyCreate = async (name, description) => {
  const companyDetail = {
    name,
    description,
  };
  const company = new Company(companyDetail);

  try {
    const newCompany = await company.save();
    this.companies.push(newCompany);

    logger.log({ level: 'info', message: `Company created with id: ${newCompany.id}` });
  } catch (err) {
    logger.log({ level: 'info', message: `An error occurred when creating a company: ${err}` });
  }
}

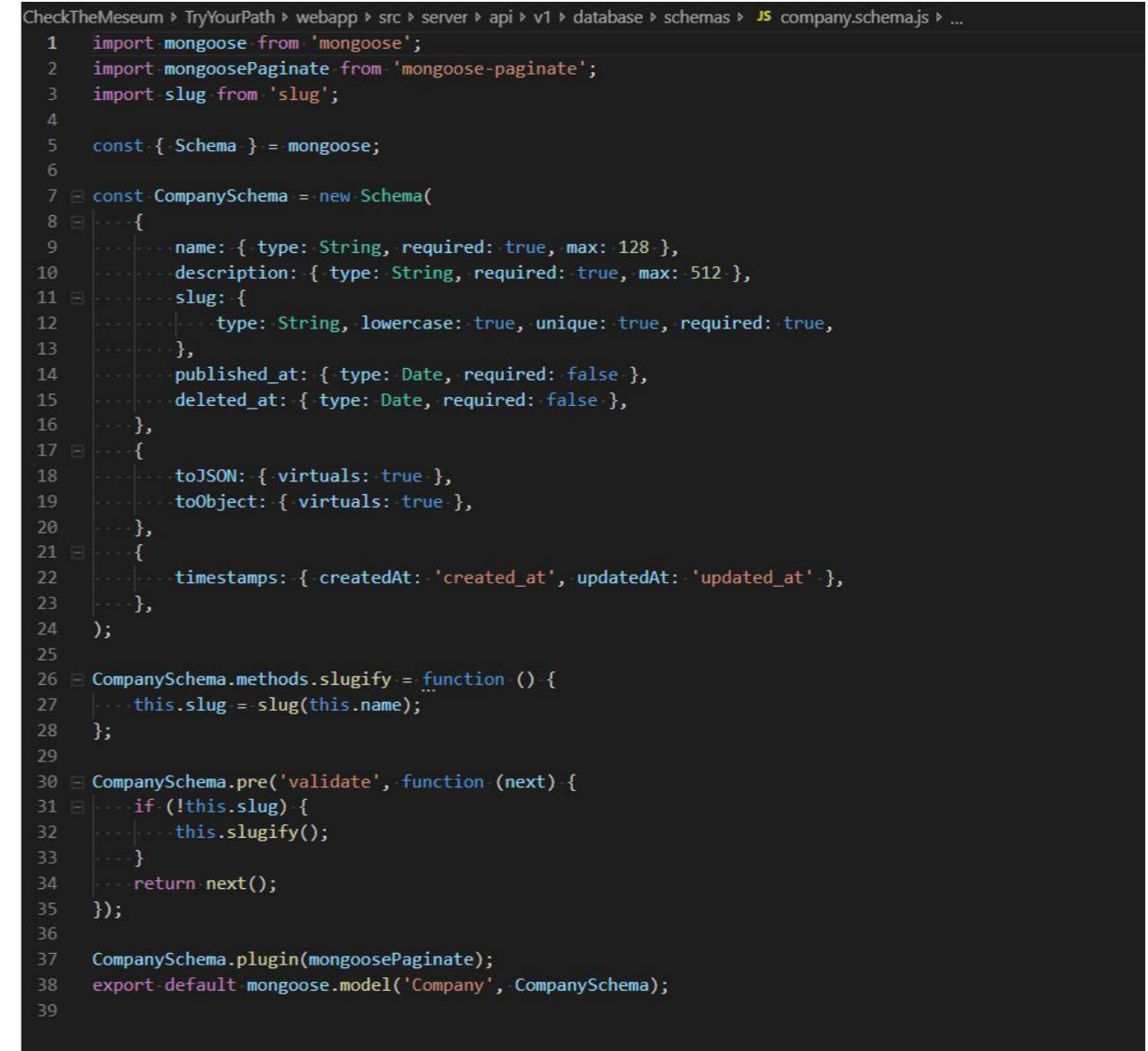
createCompanies = async () => {
  await Promise.all([
    (async () => this.companyCreate(faker.company.companyName(), faker.lorem.sentence()))(),
    (async () => this.companyCreate(faker.company.companyName(), faker.lorem.sentence()))(),
  ]);
}
```

## COMPANIES IN CONTROLLER



```
JS category.controller.js x
CheckTheMeseum > TryYourPath > webapp > src > server > api > v1 > controller > JS category.controller.js > ...
4  */
5  /*
6   * Import the internal libraries:
7   - * from database
8   - errorHandler
9   */
10 import { Category } from '../database';
11 import { APIError, handleAPIError } from '../../../../../utilities';
12
13 class CategoryController {
14   // List all the models
15   index = async (req, res, next) => {
16     try {
17       const { limit, skip } = req.query;
18       let categories = null;
19       if (limit && skip) {
20         const options = {
21           page: parseInt(skip, 10) || 1,
22           limit: parseInt(limit, 10) || 10,
23           sort: { created_at: -1 },
24         };
25         categories = await Category.paginate({}, options);
26       } else {
27         categories = await Category.find().sort({ created_at: -1 }).exec();
28       }
29     }
30     if (categories === undefined || categories === null) {
31       throw new APIError(404, 'Collection for categories not found!');
32     }
33     return res.status(200).json(categories);
34   } catch (err) {
35     return handleAPIError(500, err.message || 'Some error occurred while retrieving categories', next);
36   }
37 }
38
39 // Show specific model by id
40 show = async (req, res, next) => {
41   try {
42     const { id } = req.params;
43     const item = await Category.findById(id).exec();
44   }
45 }
```

## COMPANIES IN SCHEMA



```
CheckTheMeseum > TryYourPath > webapp > src > server > api > v1 > database > schemas > JS company.schema.js > ...
1 import mongoose from 'mongoose';
2 import mongoosePaginate from 'mongoose-paginate';
3 import slug from 'slug';
4
5 const { Schema } = mongoose;
6
7 const CompanySchema = new Schema({
8   name: { type: String, required: true, max: 128 },
9   description: { type: String, required: true, max: 512 },
10  slug: {
11    type: String, lowercase: true, unique: true, required: true,
12  },
13  published_at: { type: Date, required: false },
14  deleted_at: { type: Date, required: false },
15
16 },
17 {
18   toJSON: { virtuals: true },
19   toObject: { virtuals: true },
20 },
21 {
22   timestamps: { createdAt: 'created_at', updatedAt: 'updated_at' },
23 },
24 );
25
26 CompanySchema.methods.slugify = function() {
27   this.slug = slug(this.name);
28 };
29
30 CompanySchema.pre('validate', function(next) {
31   if (!this.slug) {
32     this.slugify();
33   }
34   return next();
35 });
36
37 CompanySchema.plugin(mongoosePaginate);
38 export default mongoose.model('Company', CompanySchema);
39
```

## ROUTES IN CLIENT

```
class Main extends Component {
  render() {
    return (
      <div>
        <Switch>
          <RouteWithLayout exact path="/" layout={ PageLayout } component={ HomePage }/>
          <Redirect from="/home" to="/" />
          <RouteWithLayout exact path="/musea" layout={ PageLayout } component={ NewsPage }/>
          <RouteWithLayout exact path="/musea/:id" layout={ PageLayout } component={ PostDetailPage }/>
          <RouteWithLayout exact path="/order-ticket" layout={ PageLayout } component={ OrderTicketPage }/>
          <RouteWithLayout exact path="/order-ticket/:id" layout={ PageLayout } component={ OrderTicketDetailPage }/>
          <RouteWithLayout path="/login" layout={ LoginLayout } component={ LoginPage }></RouteWithLayout>
          <RouteWithLayout path="/admin" layout={ AdminLayout } component={ AdminPage }></RouteWithLayout>
        </Switch>
      </div>
    );
  }
}
```

## MAPPEN STRUCTUUR IN CLIENT

```
▶ components
  ▶ musea-detail
  ▶ musea-list
  ▶ musea-list-paged
  ▶ post-detail
  ▶ posts-list
  ▶ posts-list-paged
  ▶ ticket-detail
  ▶ layouts
  ▶ pages
    ▶ home
    ▶ login
    ▶ news
    ▶ order-ticket
    ▶ order-ticket-detail
    ▶ overview-ticket
    ▶ post-detail
```

## SASS MUSEA CARDS

```
[card {
  width: 400px;
  margin-bottom: 50px;
  background-color: white;
  color: $font-color;

  -webkit-box-shadow: 0 10px 6px -6px #777;
  -moz-box-shadow: 0 10px 6px -6px #777;
  box-shadow: 0 10px 6px -6px #777;

  &__title {
    margin-top: -10px;
  }

  &__sub-title {

  }

  &__header {

  }

  &__footer {

  }

  &__image {

  }

  &__content {
    margin: 15px;
    padding: 15px;
  }

  &__description {
    overflow-y: scroll;
    height: 100px;
  }

  &__btn {
    background-color: $main-color;
    text-align: center;
    padding: 15px;
    border-radius: 7px;
    margin-top: 30px;

    > a {
      text-decoration: none;
      color: $background_color;
    }
  }
}
```

## MUSEA CARD IN WEB

```
class MusealistsPaged extends Component {
  readTicketsHandler = (ev, id) => {
    ev.preventDefault();
    if (typeof this.props.onReadTickets === 'function') {
      this.props.onReadTickets(id);
    }
  }
  readMoreHandler = (ev, id) => {
    ev.preventDefault();
    if (typeof this.props.onReadMore === 'function') {
      this.props.onReadMore(id);
    }
  }
  loadMoreHandler = (ev, pageIndex) => {
    ev.preventDefault();
    if (typeof this.props.onLoadMore === 'function') {
      this.props.onLoadMore(pageIndex);
    }
  }
  render() {
    const { musea, pagination } = this.props;
    return (
      <React.Fragment>
        {musea && musea.map( (museum, index) => (
          <article key={ museum.id } className={classNames("museum--small")}>.....
            <div className="card">
              <div className="card_header">
                
              </div>
              <div className="card_content">
                <h2 className="card_title">{ museum.name }</h2>
                <p>{ museum.location }</p>
                <p>{ museum.description }</p>
                <div className="card_btn">
                  <button className="order-btn" onClick={(ev) => this.readTicketsHandler(ev, museum.id)}>Bestel Tickets</button>
                </div>
                <button onClick={(ev) => this.readMoreHandler(ev, museum.id)}>More</button>
              </div>
            </article>
        ))}
        {musea && pagination.page < pagination.pages ? <button onClick={(ev) => this.loadMoreHandler(ev, pagination.page + 1)}>Meer laden</button>:''}
      </React.Fragment>
    );
  }
}
```

## MUSEA DETAIL IN WEB

```
/*
gdm-1819-simodeme2 libraries
*/
import React, { Component } from 'react';
import Parser from 'html-react-parser';
import classNames from 'classnames';

/*
Styling
*/
import './MuseumDetail.scss'

class MuseumDetail extends Component {
  render() {
    const { data: museum } = this.props;

    return (
      <React.Fragment>
        {museum ? (
          <article key={ museum.id } className={classNames("museum--large")}>....
            <h1 className="museum_title">{ museum.name }</h1>
            <div className="museumSynopsis">{ museum.location }</div>
            <div className="museumSynopsis">{ museum.description }</div>
          </article>
        ) : '<p>LOADING</p>'}
      </React.Fragment>
    );
  }
}

export default (MuseumDetail);
```

## MUSEA LIST RENDEREN IN WEB

```
class PostDetailPage extends Component {
  state = {
    museum: null,
  };

  componentWillMount() {
    this.loadPost(this.props.match.params.id);
  }

  loadPost = (id) => {
    Api.findOnePost(id)
      .then((data) => {
        this.setState(prevState => ({
          ...prevState,
          museum: data
        }));
      })
      .catch((error) => {
        console.log(error);
      });
  }

  render() {
    const { museum } = this.state;
    console.log(museum);
    return (
      <React.Fragment>
        <MuseumDetail data={museum}>/>
      </React.Fragment>
    )
  }
}
```

## NAVIGATIE OP MOBILE

```
const TabNavigator = createMaterialBottomTabNavigator({
  Home: { screen: HomeScreen,
    navigationOptions: {
      tabBarLabel: 'Home',
      tabBarIcon: ({ tintColor }) => (
        <View>
          <Icon style={[{color: tintColor}]} size={25} name={'ios-home'}/>
        </View>
      )
    }
  },
  Profile: { screen: ProfileScreen,
    navigationOptions: {
      tabBarLabel: 'Profile',
      tabBarIcon: ({ tintColor }) => (
        <View>
          <Icon style={[{color: tintColor}]} size={25} name={'ios-person'}/>
        </View>
      ),
      activeColor: '#f60c0d',
      inactiveColor: '#f65a22',
      barStyle: { backgroundColor: '#f69b31' },
    }
  },
});
```

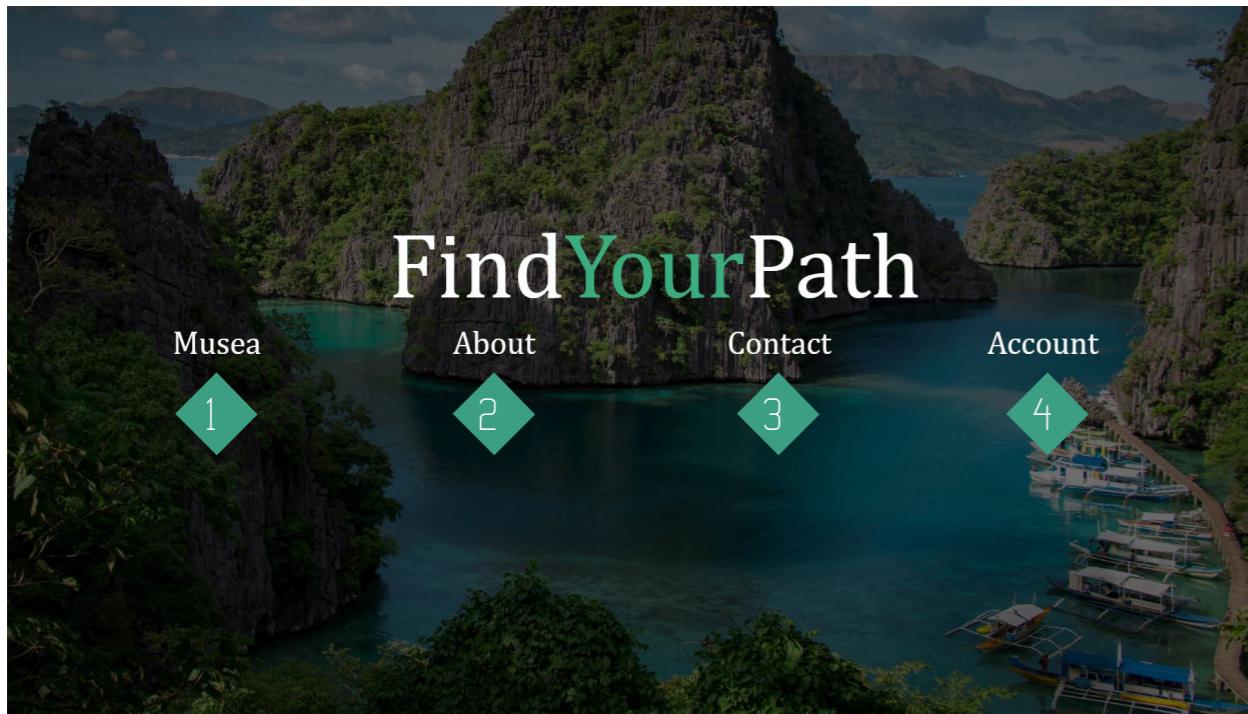
## MUSEA CARD OP MOBILE

```
class Card extends React.Component {
  render () {
    return (
      <View style={styles.card}>
        <Image source={require('../assets/placeholder_250_150.jpg')}/>
        <Text style={styles.card__title}>SMAK</Text>
        <Text style={styles.card__description}>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla a
        </Text>
        <Button
          onPress={() => this.props.navigation.navigate('Order')}
          title="Bestel tickets"
          color="#841584"
          accessibilityLabel="Learn more about this purple button"
        />
      </View>
    );
  }
}
```



**DELIVER**

HANDLEIDING



## HANDLEIDING

### 1. MUSEA

Als je op nummertje 1 drukt zal je terecht komen bij een lijst van musea. Daar zal je kunnen klikken op "Meer Lezen" of op "Tickets bestellen".

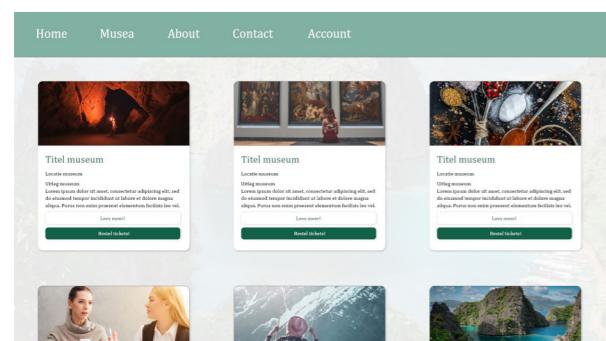
Als je niet bent ingelogd zal je bij het drukken op "Tickets bestellen", doorgestuurd worden naar de login pagina.

### 2. ABOUT

Nummer 2 is een about pagina waar je alle informatie zal terug kunnen vinden over de website en de app van FindYourPath.

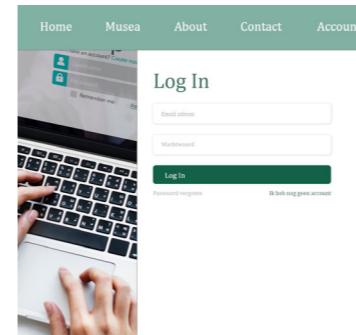
### 3. CONTACT

Bij contact vind je het emailadres en telefoonnummer van de verantwoordelijke van de website.

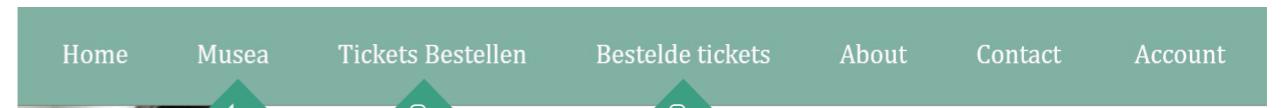


40

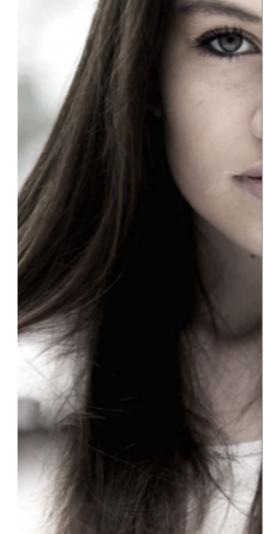
### 4. ACCOUNT



Bij account zal je terug je als je nog niet ingelogd bent terecht komen op de login pagina. Mocht je nog geen account hebben kan je onderaan zien staan "Ik heb nog geen account". Als je daar op klikt kom je terecht bij de Sign Up pagina. Eens je ingelogd bent kom je op de account pagina.



Judith Doorn



### 1. Musea tickets

Wanneer je dan naar musea gaat en dan op "Ticket bestellen" drukt zal je naar deze pagina gerefereerd worden.

Je zal alle gegevens in moeten geven zoals de dag wanneer je wilt bezoeken, een validatie code indien er meerdere mensen komen, bedrijfsnaam en het aantal ticketen. Dan nog belangrijk is of je een eigen route wilt instellen of toch de voorgemaakte route van het museum.



41

Als je kiest voor eigen route dan krijg je alle plaatsen in het museum te zien en zal je moeten aanduiden wat je wilt zien.

Route

Klik op de beelden die je wilt bezoeken

Naam beeld Naam beeld Naam beeld Naam beeld  
Naam beeld Naam beeld Naam beeld Naam beeld

Bestel tickets

Als je dan drukt op tickets bestellen of je had gekozen voor voorgeselecteerde route dan kom je op de volgende pagina.

Tickets betalen

Museum 1  
8  
DitIs Een ValidatieCode1452  
16.08.2019  
Voorgemaakte Route  
totaal: 70 euro  
Afrekenen

## 2. Ticket bestellen

Home Musea Tickets Bestellen Bestelde tickets About Contact Account

### Ticket bestellen

Welk museum  
Aantal ticketen  
Validatie code  
Datum  
Soort route  
Bestel tickets

Als je direct naar tickets bestellen gaat dan weet je al exact welk museum je wilt bezoeken. Dan hoef je alleen nog maar de naam van het museum in te geven. Daarna verloopt het weer verder zoals bij nummer 1

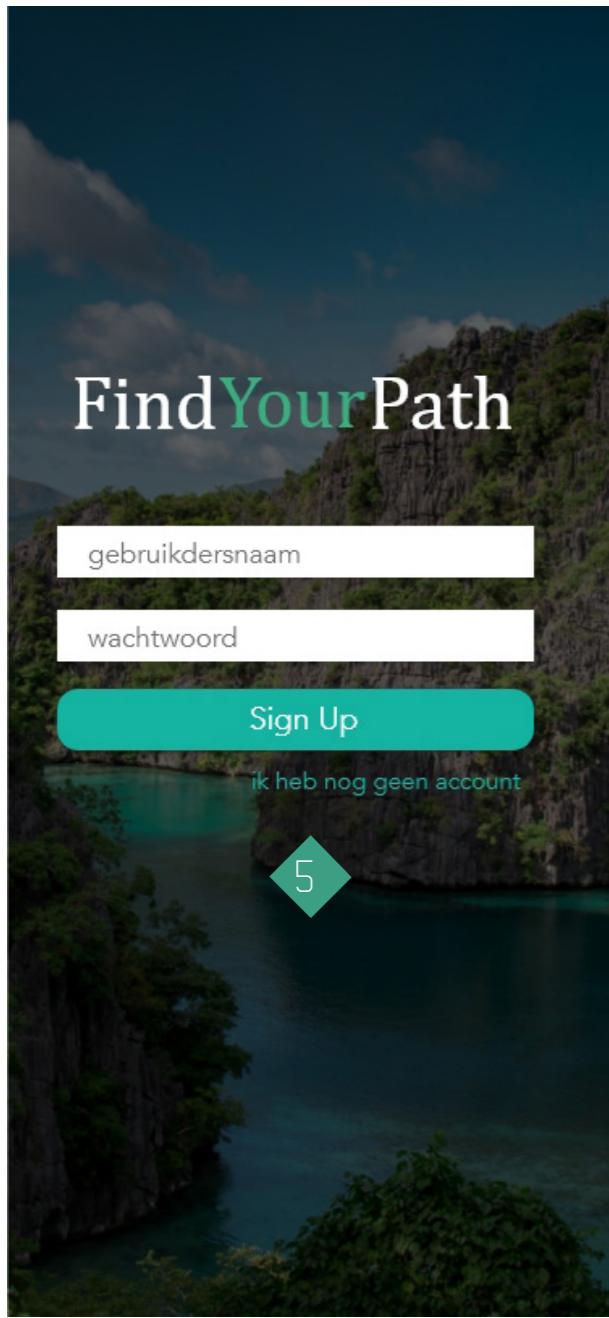
## 3. Bestelde tickets

Uiteindelijk kan je nog al jou bestellingen terug vinden, en zien of ze allemaal gevalideerd zijn of niet.

Home Musea Tickets Bestellen Bestelde tickets About Contact Account

### Bestelde tickets

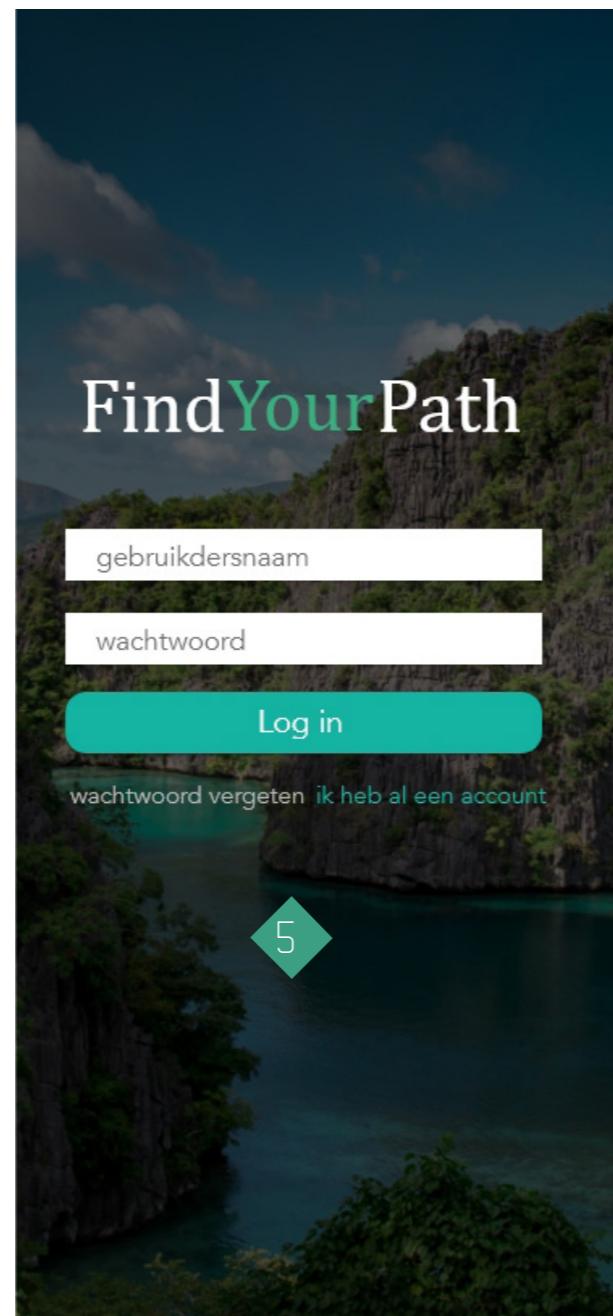
Museum	Aantal	Soort route	Datum	Validatie code
Museum 1	8	Eigen route	16.08.2019	DitIs Een ValidatieCode1452



## 5. LOGIN MOBILE

Daarnaast hebben we de mobile versie. Met mobile is het de bedoeling dat je rondleidingen zal krijgen doorheen het museum en jou ticket kan valideren van de gene die jou ticket heeft voorzien.

Het begint met het login scherm. Ook heb je het Sign up scherm mocht je nog geen account hebben.



44

## FindYourPath



Titel museum

Locatie museum

Uitleg museum

Lorem ipsum dolor sit amet,

Lees meer!

Bestel tickets!



Titel museum

Locatie museum

Uitleg museum

Lorem ipsum dolor sit amet,

Lees meer!



7

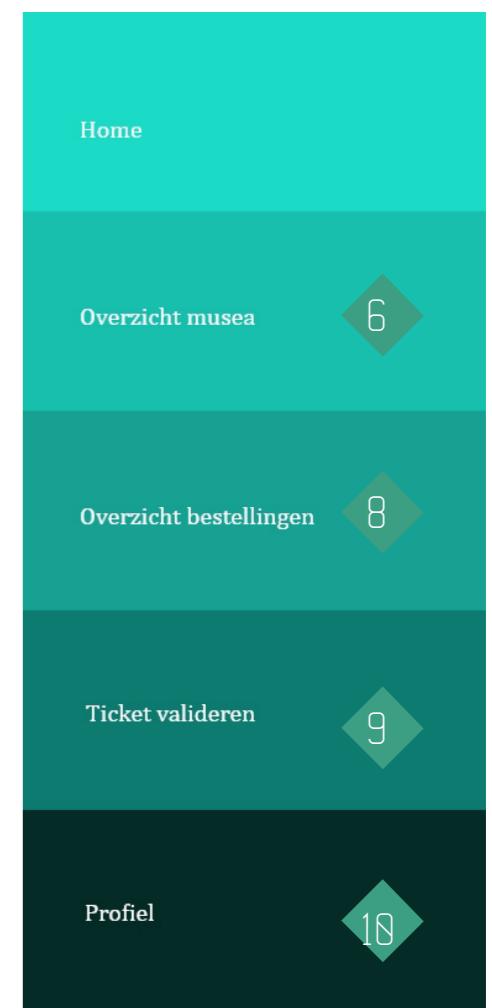
## 6. MUSEA

Hier kan je terug het scherm zien met alle musea en krijg je nog de mogelijkheid om toch nog ticketen te bestellen. Dit zal ongeveer hetzelfde zijn als op de web.

## 7. MENU

Als je onderaan het menu klikt zal een uitklapmenu krijgen waar je verschillende opties krijgt.

- \* Overzicht musea
- \* overzicht bestellingen
- \* tickets valideren
- \* profiel



45

## 8. OVERZIJKT BESTELLINGEN

Hier zie je ook weer een overzicht van jou bestellingen wanneer je bent ingelogd, met alle details. Bedrijfsnaam, aantal tickets (die naar nul gaan naarmate deze gevalideert zijn), datum van het bezoek en de validatie code.

Museum	Museum 1
Aantal	8
Soort route	Eigen route
Datum	16.08.2019
Datum	DitIsEenValidatieCode1452

## 9. TICKET VALIDEREN

Hier moet je jou eigen naam ingeven, het bedrijf en een validatie code om jou ticket op te halen en de route te starten.

Museum	Museum 1
Aantal	8
Soort route	Eigen route
Datum	16.08.2019
Datum	DitIsEenValidatieCode1452

Haal jou ticket op!

naam
Welk bedrijf?
geef code in

Start route

Wanneer je de route start moet je lopen naar het eerst volgende beeld, daar kan je de inhoud lezen over het beeld.

Na het lezen kan je naar het volgende gaan. Eventueel kan je de route stoppen. Op het einde van de route kan je eventueel een nieuw kleur vrijspelen.



## 10. PROFIEL

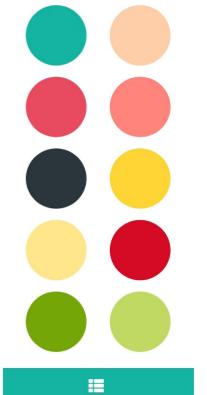
Bij profiel kan je jou gegevens zien. Eventueel een foto toevoegen aan je profiel en een kleur selecteren voor jou app.

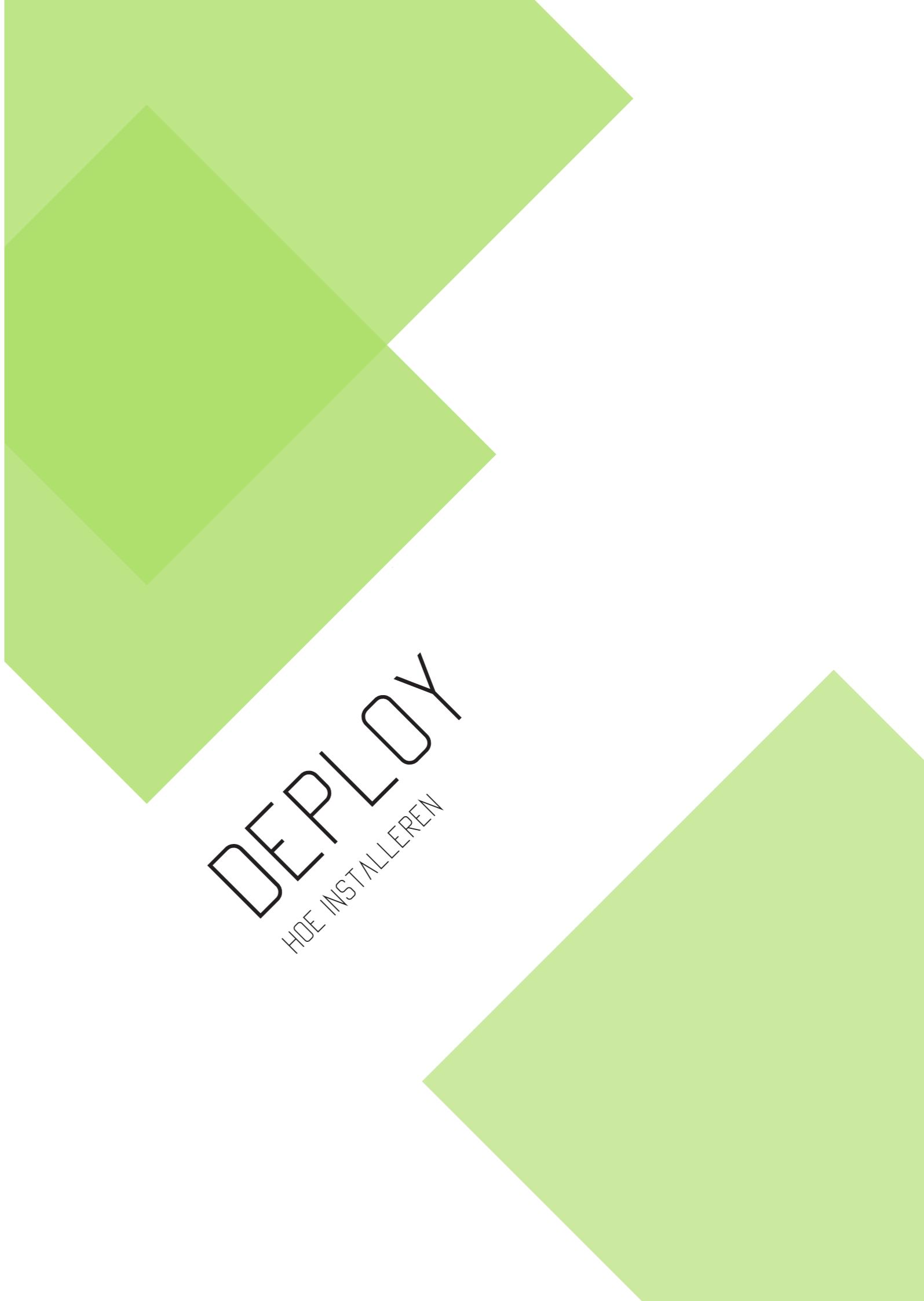
### Naam beeld

Uitleg museum  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Purus non enim praesent elementum facilisis leo vel. Enim sit amet venenatis urna cursus eget nunc scelerisque viverra. Imperdiet massa tincidunt nunc pulvinar sapien et. Gravida quis

>

Stop de rondleiding





**DEPLOY**

HOE INSTALLEREN

Template for the course Mobile Development II specialisation New Media Development in the department Graphical and Digital Media within Artevelde University College Ghent.

Template consists of:

- \* Node.js
- \* Express
- \* React

## Getting Started

### Environment settings

Create a .env-file under the root of the project with the following contents:

- \* NMD\_BASELINE='Like Graphics Love Code'
- \* NODE\_ENV={your node environment} (Development, Production or Staging)
- \* NODE\_SERVER\_HOSTNAME={your ip-adres or domainname} (ex.: 127.0.0.1, 192.168.0.6)
- \* NODE\_SERVER\_PORT={your port for the server} (ex.: 8080)
- \* MONGODB\_CONNECTION={your mongodb connection string}
- \* SKIP\_PREFLIGHT\_CHECK=true
- \* AUTH\_BCRYPT\_SALT={your salt value for passwords} (ex.: 10)
- \* AUTH\_JWT\_SECRET={your JWT secret} (ex.: gdm\_nmd\_mobdev2)
- \* AUTH\_JWT\_SESSION={your JWT session true or false}
- \* AUTH\_FACEBOOK\_CLIENT\_ID={your Facebook Client id}
- \* AUTH\_FACEBOOK\_CLIENT\_SECRET={your Facebook Client secret}

Create a .env-file under the src/client folder following contents:

- \* SKIP\_PREFLIGHT\_CHECK=true
- \* SASS\_PATH=node\_modules:src

### Installing

Under the root execute:

- \* yarn install

### Scripts

Running the Express-server in development

Under the root execute:

- \* yarn server:start

Running the React-client in development

Under the root execute:

- \* yarn client:start

Running the Express-server and React-client in development

Under the root execute:

- \* yarn start:all

The React-client must be build in order to visit the default route: <http://{your domain}:{your port}/>. Run yarn client:build to create a build folder under the root of the client folder.

Running the tests for Express-server.

Under the root execute:

- \* yarn server:test

Check eslint for the server. Under the root execute:

- \* yarn server:lint

Running the React-app. Under the root execute:

- \* yarn client:start

Building the React-app. Under the root execute:

- \* yarn client:build

This command creates a directory build under the client folder. After building, the build folder will be copied to the dist\client folder (under the root of the project).

Running the tests for React-client. Under the root execute:

- \* yarn client:test

### API Specification

#### Urls

- \* [Swagger Ui for Express](<http://{your domain}:{your port}/api/v1/docs/>)
- \* [ReDoc](<http://{your domain}:{your port}/docs/>)

### Client

#### Urls

##### Public

- \* Home: <http://{your domain}:{your port}>
- \* News: <http://{your domain}:{your port}/news>
- \* Post Detail: <http://{your domain}:{your port}/news/{post id}>

#### Admin

- \* Home: <http://{your domain}:{your port}/admin>
- \* Blog List: <http://{your domain}:{your port}/admin/blogs>
- \* Blog Create: <http://{your domain}:{your port}/admin/blogs/create>
- \* Blog Edit: <http://{your domain}:{your port}/admin/blogs/{blog id}/edit>
- \* Categories List: <http://{your domain}:{your port}/admin/categories>
- \* Category Create: <http://{your domain}:{your port}/admin/categories/create>

- \* Category Edit: <http://{your domain}:{your port}/admin/categories/{category id}/edit>
- \* Posts List: <http://{your domain}:{your port}/admin/posts>
- \* Post Create: <http://{your domain}:{your port}/admin/posts/create>
- \* Post Edit: <http://{your domain}:{your port}/admin/posts/{post id}/edit>

### Built With

- \* Node.js
- \* Express
- \* React

### Authors

- \* Philippe De Pauw - Waterschoot | drdynscript
- \* Simon Demeyere - 2MMP-NMD
- \* Anna De Langhe - 2MMP-NMD

### License

This project is licensed under the Apache License - see the LICENSE file for details

