

# **MobDev II**

## **Visuele impact NMD**

**Door Wouter Vlaeyen, Owen De Waele, Bruno De Coene**  
**Docent Phillipe De Pauw**

# Inhoud

**Discover**

**Define**

**Design**

**Develop**

**Demo**



# GDM Gent

Join GDM Gent.

Let's build cool things together

See more



20

19

# Discover

**Visuele  
impakt  
GDM**

**Website  
Webapp**

**MERN -  
stack**

**Server**

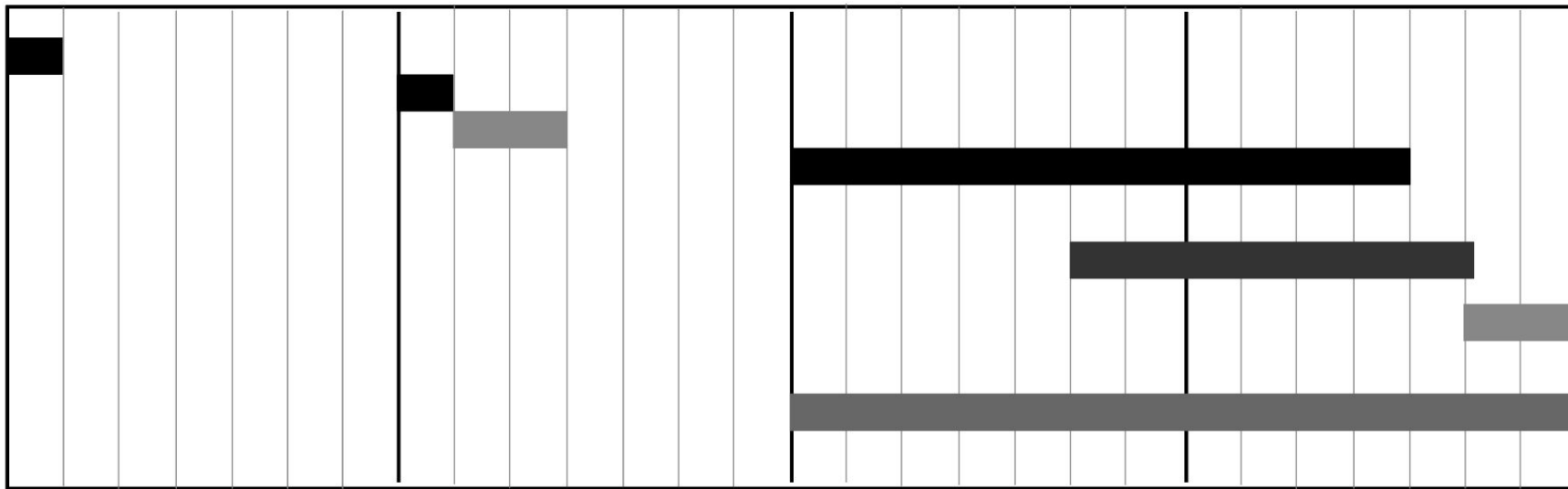
**Database**

**Client**

# Define



# Planning



Template  
downloaden  
+ configuratie

Configuratie afwerken  
API calls / structure

Inladen van  
dynamische  
content en layout

Testing

Inladen content /  
laout

Authenticatie


Testing

Inladen content /  
laout

Authenticatie

Dossier

# **Persona sheets**



## Thomas

Job: Student      Locatie: Gent, België

**Beschrijving**  
 Thomas is een 19 jarige student uit Gent. In zijn vrije tijd houdt hij zich vooral bezig met sport en film, zijn twee grote passies. Nu de middelbare school achter de rug is, denkt hij er aan om een opleiding bij Artevelde te beginnen.

**Noten | Doelen**  
 – Een opleiding vinden die voor hem geschikt is  
 – Creativiteit blijven combineren met werk.

**Uitdagingen | Frustraties**  
 – Veel verschillende scholen en opleidingen  
 – Niet veel tijd om een keuze te maken  
 – Persoonlijke projecten nemen veel tijd in

### Skillset

Algemene kennis	Video skills
83%	86%



## Jennifer

Job: Receptioniste      Locatie: Wetteren, België

**Beschrijving**  
 Jennifer is een 42 jarige receptioniste uit Wetteren. Haar zoon begint binnenkort aan een hogere opleiding. Ze heeft gehoord dat hij misschien een opleiding GDM wil doen, dus gaat ze op zoek naar meer info over deze opleiding.

**Noten | Doelen**  
 – Op een makkelijke en duidelijke manier informatie vinden  
 – Creativiteit blijven combineren met werk.

**Uitdagingen | Frustraties**  
 – Computergebruik kan wel te niet strand  
 – Geen idee waar te zoeken

### Skillset

Algemene kennis	Internet skills
88%	52%



## Marco

Job: Grafisch Designer      Locatie: Antwerpen, België

**Beschrijving**  
 Marco is een freelance grafisch designer. Na zijn opleiding aan Artevelde is hij begonnen bij een klein bedrijf in Antwerpen, maar is al snel als freelancer begonnen. Als oud-student kijkt hij regelmatig nog eens wat er binnen de opleiding gebeurt.

**Noten | Doelen**  
 – Up to date blijven met nieuws binnen de opleiding  
 – Anderen informeren via getuigenissen

**Uitdagingen | Frustraties**  
 – Geen duidelijk systeem om getuigenissen achter te laten  
 – Enkel nieuws voor oud-studenten in overvolle inbox

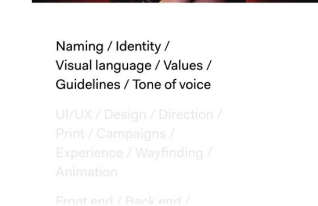
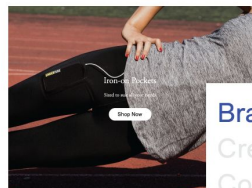
### Skillset

Algemene kennis	Internet skills
74%	92%

**Scenario's**

**Deze worden uitgebreid besproken  
in het dossier**

**Moodboard**



James Green  
Director



Charles Fisher: A Year at Chaptr in Digital Marketing.

2 days ago

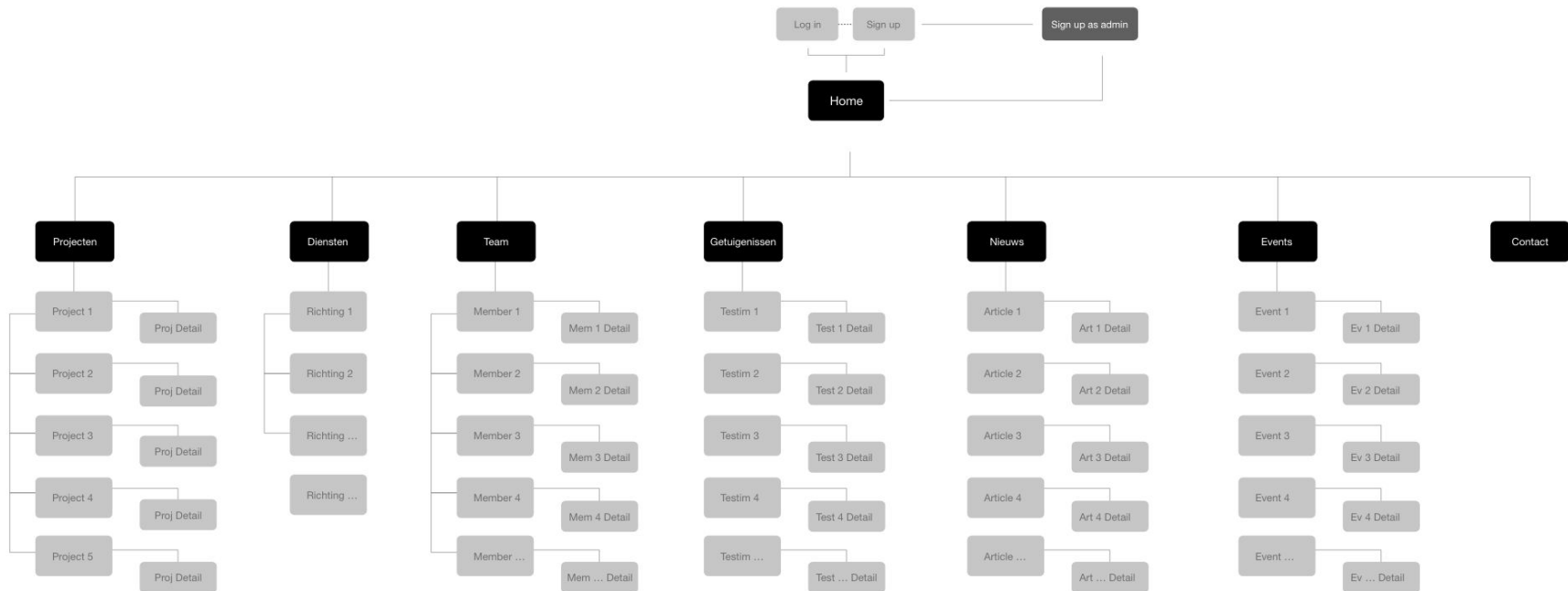
# Moodboard



# Design

# Sitemap

# Sitemap



# Wireframes



**Deze worden uitgebreid  
weergegeven in het dossier**

# Style Tiles

## Colors



## Fonts

Gill Sans Std is our main font  
This is our main paragraph text

**“A quote looks like this”**

### Section title

PROJECT TITLE

# Style Tiles

## Buttons

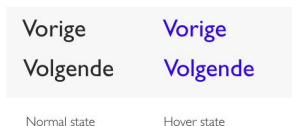
### Main call to action



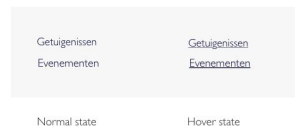
### Secondary call to action



### Main link button

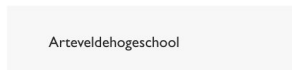


### Secondary link button

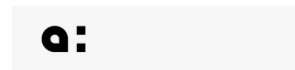


## Logo

### Text logo



### Logo



# **Visual Designs**



**Deze werden uitgebreid verwerkt  
in het dossier**

# Develop

# **Code snippets server**

```

import mongoose from 'mongoose';
import mongoosePaginate from 'mongoose-paginate';
import slug from 'slug';

const { Schema } = mongoose;

const ProjectSchema = new Schema(
  {
    title: { type: String, required: true, max: 128 },
    synopsis: { type: String, required: true, max: 1024 },
    body: { type: String, required: true },
    slug: {
      type: String, lowercase: true, unique: true, required: true,
    },
    creator: { type: String, max: 128 },
    published_at: { type: Date, required: false },
    deleted_at: { type: Date, required: false },
    categoryId: { type: Schema.Types.ObjectId, ref: 'Category', required: false },
  },
  {
    toJSON: { virtuals: true },
    toObject: { virtuals: true },
  },
  {
    timestamps: { createdAt: 'created_at', updatedAt: 'updated_at' },
  }
);

ProjectSchema.methods.slugify = function () {
  this.slug = slug(this.title);
};

Complexity is 3 Everything is cool!
ProjectSchema.pre('validate', function (next) {
  if (!this.slug) {
    this.slugify();
  }
  return next();
});

ProjectSchema.virtual('id').get(function () { return this._id });
ProjectSchema.virtual('category', {
  ref: 'Category',
  localField: 'categoryId',
  foreignField: '_id',
  justOne: true,
});
ProjectSchema.virtual('images', {
  ref: 'ProjectImage',
  localField: '_id',
  foreignField: 'projectId',
  justOne: false,
});

ProjectSchema.plugin(mongoosePaginate);
export default mongoose.model('Project', ProjectSchema)

```

```

import { Project } from '../database';
import { APIError, handleAPIError } from '../../utilities';

```

Complexity is 14 You must be kidding

```
class ProjectController {
```

Complexity is 14 You must be kidding

```
index = async (req, res, next) => {
```

```

  try {
    const { limit, skip } = req.query;
    let projects = null;

    if (limit && skip) {
      const options = {
        page: parseInt(skip, 10) || 1,
        limit: parseInt(limit, 10) || 10,
        populate: ['images', 'category'],
        sort: { created_at: -1 },
      };
      projects = await Project.paginate({}, options);
    } else {
      projects = await Project.find().populate(['category', 'images']).sort({
        created_at: -1 }).exec();
    }

    if( projects === undefined || projects === null) {
      throw new APIError(404, 'Collection for projects not found');
    }

    return res.status(200).json(projects);
  } catch (err) {
    return handleAPIError(500, err.message || 'Some error occurred while retrieving posts', next);
  }
};

```

Complexity is 10 It's time to do something...

```
show = async (req, res, next) => {
```

```

  try {
    const { id } = req.params;
    const item = await Project.findById(id).populate(['category', 'images']).exec();

    if (item === undefined || item === null) {
      throw new APIError(404, `Project with id ${id} not found`);
    }
    return res.status(200).json(item);
  } catch (err) {
    return handleAPIError(err.status || 500, err.message || 'Something went wrong while retrieving posts', next);
  }
}

```

```

import { ProjectController } from '../controller';

const projectController = new ProjectController();

const initializeEndpoints = (parentRouter, authService) => {
  /**
   * @swagger
   * /api/v1/projects:
   *   get:
   *     tags:
   *       - Projects
   *     description: Returns all projects
   *     produces:
   *       - application/json
   *     responses:
   *       200:
   *         description: An array of projects
   */
  parentRouter.get('/projects', projectController.index);
  /**
   * @swagger
   * /api/v1/projects/create:
   *   get:
   *     tags:
   *       - Project
   *     description: Returns specific viewmodel such as projects
   *     produces:
   *       - application/json
   *     responses:
   *       200:
   *         description: Create project
   */
  parentRouter.get('/projects/create', projectController.create);
  /**
   * @swagger
   * /api/v1/projects/{id}:
   *   get:
   *     tags:
   *       - Project
   *     description: Returns specific project
   *     produces:
   *       - application/json
   *     parameters:
   *       - name: id
   *         description: Project id
   *         in: path
   *         required: true
   *         type: string
   *     responses:
   *       200:
   *         description: Get project by id
   */
  parentRouter.get('/projects/:id', projectController.show);
  /**
   * @swagger
   * /api/v1/projects:
   *   post:

```

```

Complexity is 6 It's time to do something...
initializeGithubStrategy = () => {
  passport.use(new GithubStrategy(
    {
      clientID: config.auth.github.clientID,
      clientSecret: config.auth.github.clientSecret,
      callbackURL: "http://127.0.0.1:8080/api/v1/login/github/redirect",
    },

```

Complexity is 5 Everything is cool!

```
function(accessToken, refreshToken, profile, done) {
```

Complexity is 4 Everything is cool!

```

User.findOne({ 'githubProvider.id': profile.id }).then((currentUser) => {
  if(currentUser) {
    done(null, currentUser);
  } else {
    new User({
      email: profile.emails[0].value,
      githubProvider: {
        id: profile.id,
        token: accessToken,
      }
    }).save().then((newUser) => {
      done(null, newUser);
    });
  }
});
// User.findOrCreate({ githubId: profile.id }, function(err, user) {
//   return cb(err, user);
// });
});
}

```

Complexity is 10 It's time to do something...

```

initializeLocalStrategy = () => {
  passport.use(new LocalStrategy({
    usernameField: 'email',
  },

```

Complexity is 9 It's time to do something...

```

async (email, password, done) => {
  const user = await User.findOne({email});
  if (!user) { return done(null, false); }

```

Complexity is 5 Everything is cool!

```

return user.comparePassword(password, (isMatch) => {
  if(isMatch) {
    return done(null, false);
  } else {
    return done(null, user);
  }
});
}
});
// passport.use(new LocalStrategy(

```

Filter

HistoryCollectionsAPIsBETA

Save ResponsesClear all

Today

- GET http://127.0.0.1:8080/api/v1/projects
- GET http://127.0.0.1:8080/api/v1/course
- GET http://127.0.0.1:8080/api/v1/team
- GET http://127.0.0.1:8080/api/v1/
- GET http://127.0.0.1:8080/api/v1/
- GET http://127.0.0.1:8080/api/v1/
- GET http://127.0.0.1:8080/api/v1/
- GET http://127.0.0.1:8080/api/v1/course

Yesterday

- GET http://127.0.0.1:8080/api/v1/course
- PUT http://127.0.0.1:8080/api/v1/events/5d0227e23c0841634dada321
- GET http://127.0.0.1:8080/api/v1/events/5d0227e23c0841634dada321
- GET http://127.0.0.1:8080/api/v1/events/
- PUT http://127.0.0.1:8080/api/v1/events/5d039e70d3692c71afe5fadc
- PUT http://127.0.0.1:8080/api/v1/events/5d039e70d3692c71afe5fadc
- GET http://127.0.0.1:8080/api/v1/events/5d039e70d3692c71afe5fadc
- GET http://127.0.0.1:8080/api/v1/events

GET http://127.0.0.1:8080/api/v1/projectsPUT http://127.0.0.1:8080/api/v1/ev...GET http://127.0.0.1:8080/api/v1/pr...+...

No Environment

http://127.0.0.1:8080/api/v1/projects

GEThttp://127.0.0.1:8080/api/v1/projectsSendSave

ParamsAuthorizationHeaders (9)BodyPre-request ScriptTestsCookiesCodeComments

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

BodyCookiesHeaders (9)Test ResultsStatus: 200 OKTime: 121 msSize: 6.74 KBDownload

PrettyRawPreviewJSON

```
1 {
2   "_id": "5ceba948864a754dacef2be5",
3   "title": "Thrive",
4   "synopsis": "Erasmusproject: Exchange Knowledge for Future Innovation",
5   "body": "Gedurende 3 jaar zal de opleiding 'Grafische en Digitale Media' van de Arteveldehogeschool samen werken met 5 buitenlandse partners in het kader van Erasmus+ aan het project 'EKFI' (Exchange Knowledge for Future Innovation). Dit project heeft als doel om een platform te ontwikkelen om vooral innovatief lesmateriaal uit te wisselen tussen verschillende organisaties in de creatieve en crossmediale sector, zoals bv. secundaire en hogescholen, bedrijven & sectororganisaties. Het EKFI-project is gestart met een kick off meeting begin december 2018 in Schiphol in Nederland. Meer info over dit project is te vinden op EFKI-project. Het onderzoeksteam is op zoek naar Vlaamse bedrijven en onderwijsinstellingen die interesse hebben in het uitwisselen van digitaal lesmateriaal in het domein van crossmedia (print & multimedia), marketing en design. Heb je interesse, neem dan snel contact met Inge Sintobin mail us.",
6   "categoryId": "5cd016742bbe276aa09b340d",
7   "slug": "this-is-a-test-project",
8   "_v": 0,
9   "id": "5ceba948864a754dacef2be5",
10  "category": {
11    "_id": "5cd016742bbe276aa09b340d",
12    "name": "impedit",
13    "description": "Omnis qui vero.",
14    "slug": "impedit",
15    "_v": 0,
16    "id": "5cd016742bbe276aa09b340d"
17  },
18  "images": [
19    {
20      "_id": "5d028ef65f37a7ebdc7a668f",
21      "title": "Een uil foto",
22      "path": "https://cdn.dribbble.com/users/1994/screenshots/4410997/attachments/1002768/308_d.png",
23    }
24  ]
25 }
```

BootcampBuildBrowse

**Code  
snippets  
client**

```
class Main extends Component {
  render() {
    return (
      <div className="App">
        <Switch>
          <RouteWithLayout
            exact
            path="/"
            layout={PageLayout}
            component={HomePage}
          />
          <Redirect from="/home" to="/" />
          <RouteWithLayout
            exact
            path="/news"
            layout={PageLayout}
            component={NewsPage}
          />
          <RouteWithLayout
            exact
            path="/news/:id"
            layout={PageLayout}
            component={NewsDetailPage}
          />
          <RouteWithLayout
            exact
            path="/projects"
            layout={PageLayout}
            component={ProjectsPage}
          />
          <RouteWithLayout
            exact
            path="/projects/:id"
            layout={PageLayout}
            component={ProjectDetailPage}
          />
          <RouteWithLayout
            exact
            path="/events"
            layout={PageLayout}
            component={EventsPage}
          />
          <RouteWithLayout
            exact
```



```
/*
Import extenal libraries
*/
import React, { Component } from "react";

/*
Import internal libraries
*/
import Api from "../../services";
import GeneralNav from "../../components/header/general-nav/GeneralNav";
import GridWrapper from "../../components/structural-components/grid-wrapper/GridWrapper";
import Footer from "../../components/footer";
import BlankDiv from "../../components/styled-components/blank-div/BlankDiv";
import Title from "../../components/text-components/title/Title";
import OverlayMenu from "../../components/structural-components/overlay-menu/OverlayMenu";

/*
Import stylesheet
*/
import "./EventsPage.scss";
```

```

class EventsPage extends Component {
  state = {
    events: [],
    pagination: {
      limit: 5,
      page: 1,
      pages: 1,
      total: 1
    }
  };

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

  loadEvents = pageIndex => {
    console.log(pageIndex);
    Api.findAllEvents({ limit: 3, skip: pageIndex })
      .then(data => {
        console.log(data.docs);
        const prevElements = this.state.events;
        const newElements = [...prevElements, ...data.docs];
        this.setState(prevState => ({
          ...prevState,
          events: newElements,
          pagination: {
            limit: data.limit,
            page: data.page,
            pages: data.pages,
            total: data.total
          }
        }));
      })
      .catch(error => {
        console.log(error);
      });
  };

  goToProjectDetailPage = id => {
    this.props.history.push(`/events/${id}`);
  };

  toggleMenu = e => {
    e.preventDefault();
    this.setState(state => ({ showMenu: !state.showMenu }));
  };
}

```

```

render() {
  const { pagination, events } = this.state;
  this.items = this.state.events.map((item, key) => (
    <div key={item.id} className="event-wrapper">
      <h2 className="primary-subtitle">{item.title}</h2>
      <div className="event-border" />
      <div className="image-wrapper">
        <div className="event-image-color-layer" />
        <a href={"/events/" + item.id}>
          <img src={item.image} alt="" className="event-image" />
        </a>
      </div>
      <a href={"/events/" + item.id} className="view-event-button">
        Bekijk event
      </a>
    </div>
  ));

  return (
    <React.Fragment>
      <OverlayMenu menustate={this.state.showMenu} />
      <GridWrapper style="html-wrapper">
        <GridWrapper style="main-page-wrapper">
          <GeneralNav
            toggleMenu={this.toggleMenu}
            menustate={this.getParentState()}
          />
          <BlankDiv style="blank-div-lg" />
          <Title style="section-title" text="Events" />
          <div className="event-row-container">{this.items}</div>
          <BlankDiv style="blank-div-lg" />
          <Footer />
        </GridWrapper>
      </GridWrapper>
    </React.Fragment>
  );
}

```

```
export default EventsPage;
```

# Demo

**Bedankt voor uw aandacht**