



# SLEEPLOG

## Productiedossier

3NMD - CMS Development  
Bachelor in de grafische en  
digitale media

2018 - 2019

<b>DISCOVER</b>	<b>2</b>
Briefing	3
<b>DEFINE</b>	<b>4</b>
Analyse	5
1. Technische scope	5
2. Functionele scope	5
Planning	7
<b>DESIGN</b>	<b>8</b>
Wireflows	9
1. Profiel	9
2. Kalender & Statistieken	10
3. Vrienden	11
4. Tips & tools	11
5. Wall	12
Style Guide	13
Visual Designs	14
1. Profiel	14
2. Kalender	15
3. Statistieken	16
4. Vrienden	16
5. Tips & tools	17
6. Wall	17
7. Bedtijd	18
<b>DEVELOP</b>	<b>19</b>
Code snippets	20
Schermafbeeldingen	21
<b>DELIVER</b>	<b>22</b>
Handleiding Mobile Web App	23
<b>DEPLOY</b>	<b>24</b>
Deployment Guide	25
Bijlage: timesheet	26

# DISCOVER

# Briefing

In het kader van een project rond gezonde slaapgewoontes bij jongeren, wordt vanuit de onderzoeksgroep 'Gezondheidsbevordering' aan de UGent momenteel een interventie ontwikkeld die gezonde slaap wil promoten. Momenteel nemen 7 Vlaamse scholen deel aan dit project (3 als interventieschool, 4 als controleschool). De inhoud van de interventie ter promotie van gezonde slaapgewoontes, wordt samen met de doelgroep bepaald, er wordt in dit project namelijk gedoeld om op maximale wijze participatief te werk te gaan.

Van de jongeren werd vernomen dat ze heel graag een app zouden ontwikkelen, die hen kennis over slaap bijbrengt maar ook stimuleert om hun smartphone/tablet minder te gebruiken voor het slapengaan.

Voor het ontwikkelen beschikt de onderzoeksgroep niet over de juiste kennis, noch over uitgebreide middelen. Daarom zijn ze op zoek naar developers die hen hierbij zouden kunnen helpen.

De interventie gaat van start in januari 2019. Wanneer de interventie na afloop effectief blijkt te zijn (er wordt zowel een effect als procesevaluatie uitgevoerd), zou het de bedoeling zijn om de ontwikkelde interventie (en dus de eventueel ontwikkelde app), wijder te verspreiden over Vlaamse middelbare scholen.

Het doel is om een progressive mobiele web app (PWA) ontwikkelen. Deze draait vanuit de browser, en is gebouwd met behulp van het framework Angular. Alle data van die mobiele app wordt beheerd vanuit een CMS (je hebt hierbij de keuze uit Wordpress of Drupal).

# DEFINE

# Analyse

## 1. Technische scope

De responsive mobile web app is ontwikkeld met behulp van de volgende programma's:

- Adobe Creative Cloud
- MAMP
- Firefox Developer Edition
- Visual Studio Code

Voor de serverside is gebruik gemaakt van de volgende technologieën:

- Drupal 8
- Drupal Console
- PHP

Ten slotte is voor de client side gebruik gemaakt van de volgende technologieën:

- Angular 2
- Typescript
- RESTful API
- Sass
- HTML5
- OAuth 2

## 2. Functionele scope

- **Gebruiker**
  - Profiel
  - Gegevens
    - profielfoto, naam, school, woonplaats, tagline
    - persoonlijk slaapdoel kunnen opstellen
  - Type slaper
  - Login en registreer

- **Vrienden**
  - Lijst van vrienden kan op een interactieve map worden weergegeven  
Locatie is aangeduid met een bed-icon
    - profiel met openbare gegevens
- **Quiz**
  - Via formulier kun je je slaapprofiel bepalen. Resultaat verschijnt op het profiel.
  - Slaapprofiel wordt getoond als icon
- **Slaap Tracking**
  - Dagelijks is er mogelijkheid tot ingeven
    - om hoe laat je opgestaan en gaan slapen bent
    - kwaliteit van je slaap (rating van 0-5)
    - ben je uitgerust (rating van 0-5)
  - Statistieken
    - hoeveelheid slaap gemiddeld per nacht
    - kwaliteit van je slaap gemiddeld per nacht
    - uitgerustheid gemiddeld per nacht
- **Bedtime**
  - Gebruiker kan een herinnering instellen op het moment wanneer hij moet gaan slapen (push notification).
- **Tips & Tools**
  - Categorieën: filmpjes en tips ivm slaaphygiëne
- **Wall**
  - Gebruiker kan foto's uploaden op een openbare tijdlijn
  - Mogelijkheid om foto terug te verwijderen.
  - Sorteermogelijkheid (likes, upload datum en afstand)
  - Overzicht van alle posts
  - Foto kan geliked worden. Totaal aantal likes zijn zichtbaar

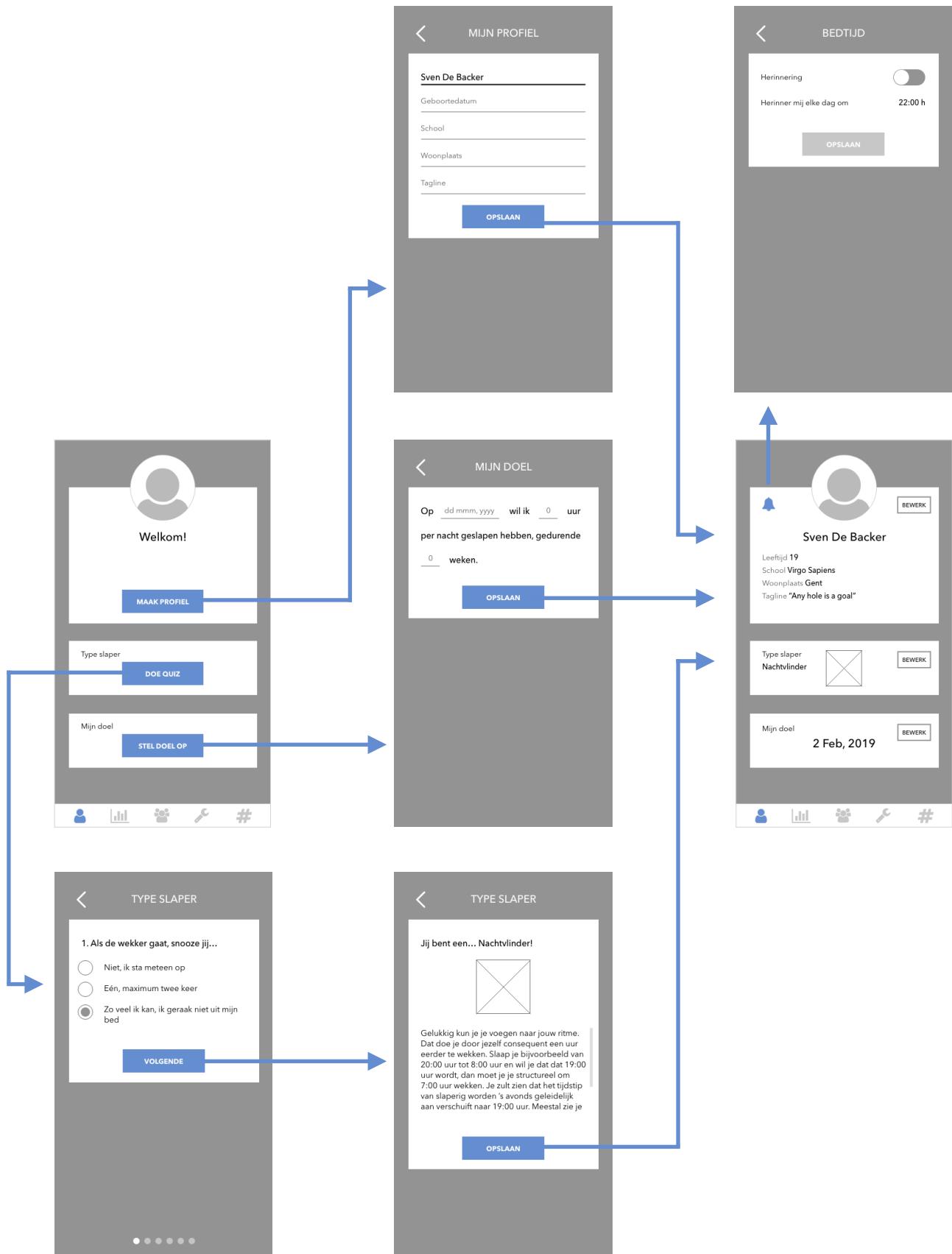
# Planning

	December	Januari
1	concept	App: friends
2	visuals	Dossier samenstellen
3		
4	wireflow	<b>Deadline: dossier</b>
5	style guide	
6	begin implementatie	App: tips & tools
7	theme & Drupal setup	
8		App: wall
9	App: profile info	<b>Deadline: code</b>
10		
11		Onvolledigheden afwerken
12		
13	App: profile goal	Presentatie voorbereiden
14		<b>Presentaties</b>
15		
16		
17	App: profile quiz	
18		
19		
20		
21	App: auth	
22		
23		
24		
25		
26		
27	App: auth / calendar	
28		
29		
30	App: calendar / stats	
31		

# DESIGN

# Wireflows

## 1. Profiel

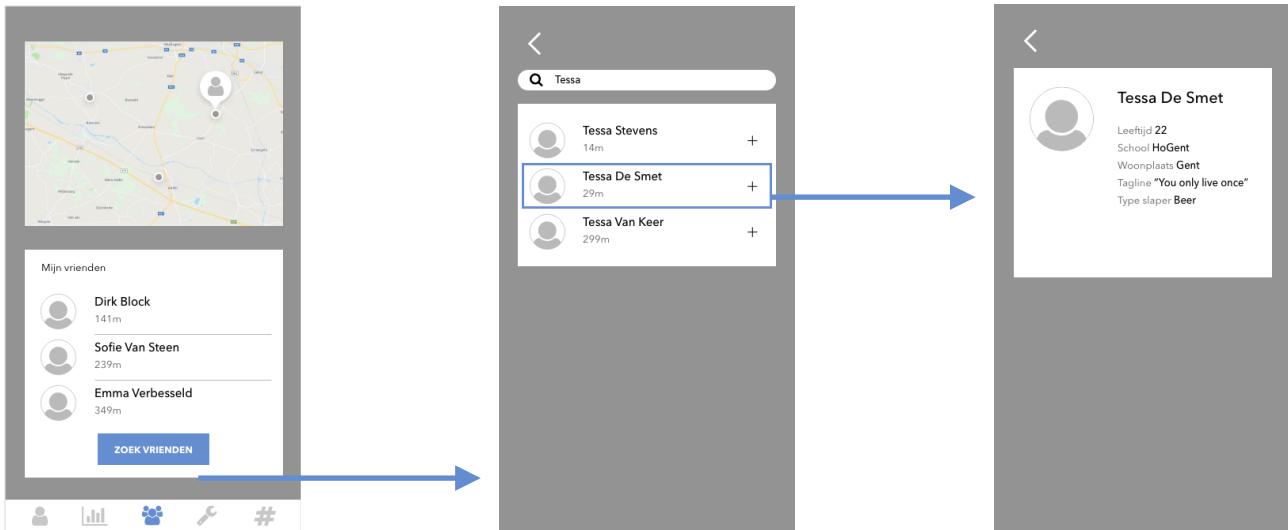


## 2. Kalender & Statistieken

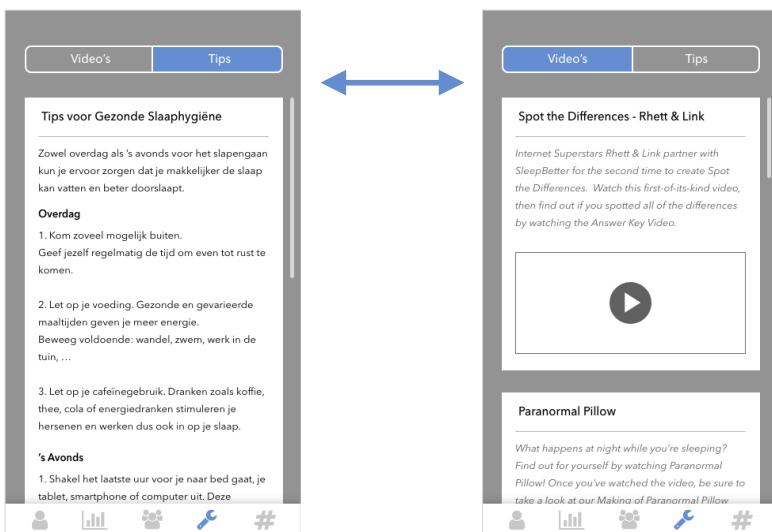
The diagram illustrates the workflow in the Sleeplog CMS:

- Kalender Screen:** Shows a monthly calendar for April 2018. A blue arrow points from this screen to the 'NIEUWE ENTRY' screen.
- NIEUWE ENTRY Screen:** A modal window titled 'NIEUWE ENTRY' for creating a new sleep entry. It includes fields for 'Gaan slapen om' (22:00 h), 'Opgestaan om' (08:00 h), 'Slaap kwaliteit' (5 stars), and 'Uitgerust' (5 stars). An 'OPSLAAN' (Save) button is at the bottom. A blue arrow points from this screen to the 'Stats' screen.
- Stats Screen:** Shows a line graph of sleep duration (in hours) over time. The Y-axis ranges from 0 h to 10 h, and the X-axis shows dates from March 17 to March 26. The graph shows fluctuating sleep durations. Below the graph, tabs for 'Uren slaap' (Sleep hours), 'Slaap kwaliteit' (Sleep quality), and 'Uitgerust' (Awake) are visible. A blue arrow points from the 'Kalender' screen to this 'Stats' screen.

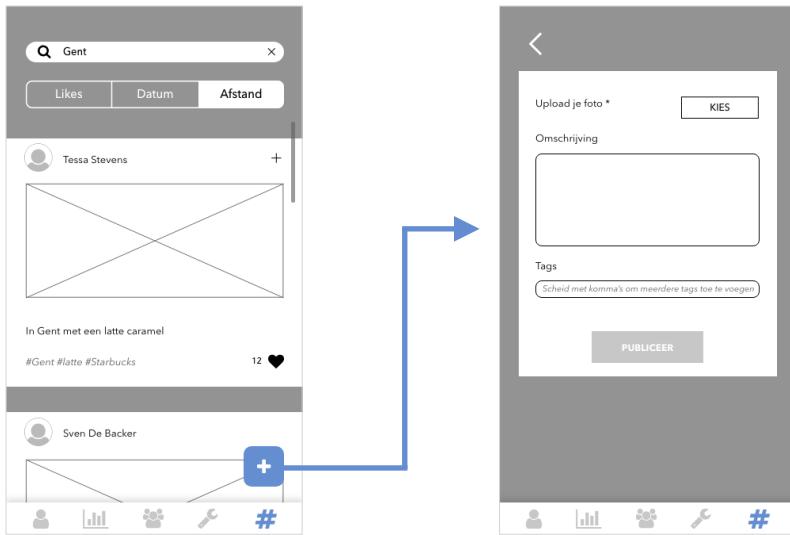
### 3. Vrienden



### 4. Tips & tools



## 5. Wall



# Style Guide

## Sleeplog

### STYLE GUIDE



#### COLORS

#648ED0		#876BD1	
#876BD1		#B5A4E3	
#000000	#707070	#A0A0A0	#FFFFFF

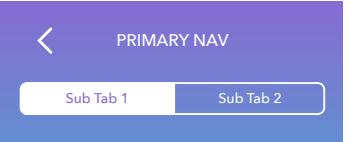
#### BUTTONS



#### TYPOGRAPHY

<b>Aa</b>	Ultra Light
Avenir	Regular
Next	Medium
	Demi Bold
<b>18PX MEDIUM</b>	Header titles (small caps)
15px Medium	Big titles, nav tabs
12px Medium	Buttons
14px Regular	Normal text, input
13px Regular	Placeholders, small labels
13px Regular	Main labels

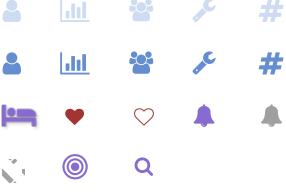
#### NAVIGATION



#### COMPONENTS

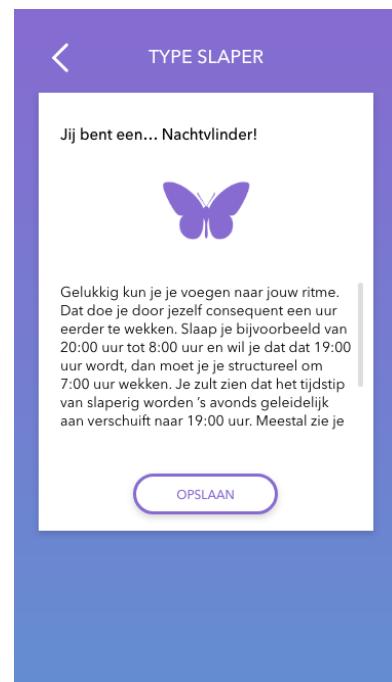
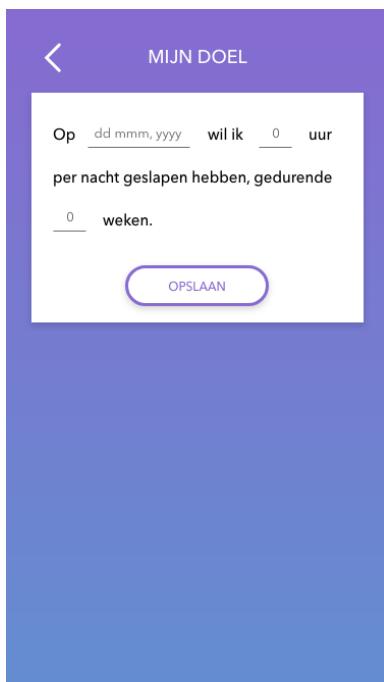
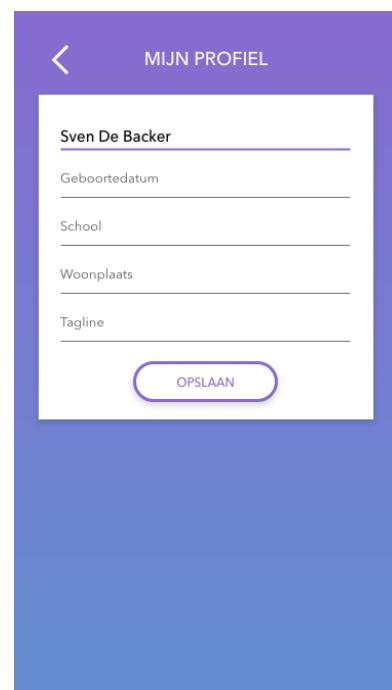
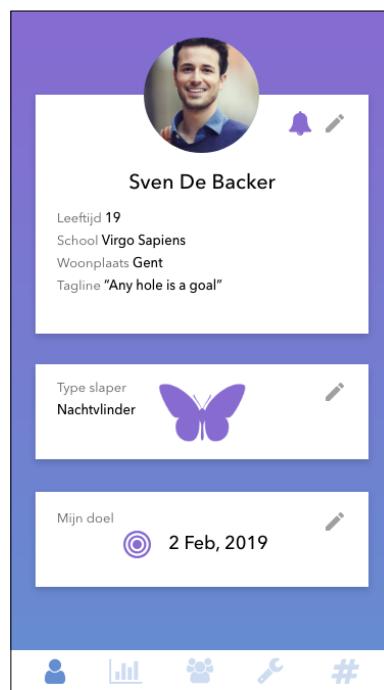
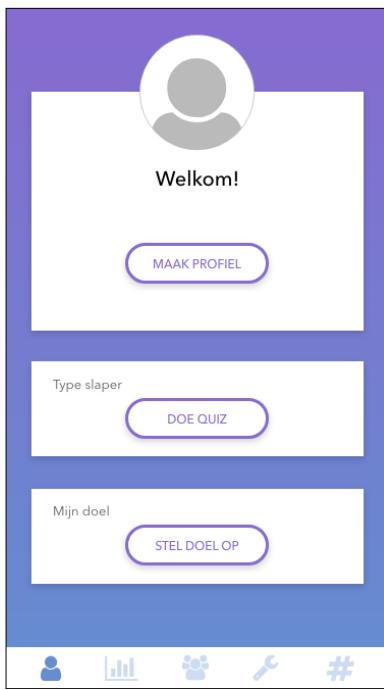
Gaan slapen om	22:00 h
Opgestaan om	08:00 h
Slaap kwaliteit	☆☆☆☆☆
Uitgerust	★★★★☆
Herinnering (aan)	<input checked="" type="checkbox"/>
Herinnering (uit)	<input checked="" type="checkbox"/>
Upload je foto *	<input type="button" value="KIES"/>
Omschrijving	<input type="text"/>
Tags	<input type="text" value="Scheid met komma's om meerdere tags toe te voegen"/>
Zoek...	<input type="text"/>

#### ICONOGRAPHY

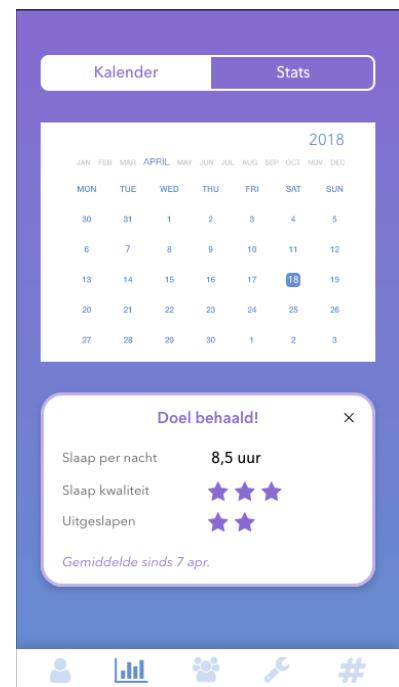
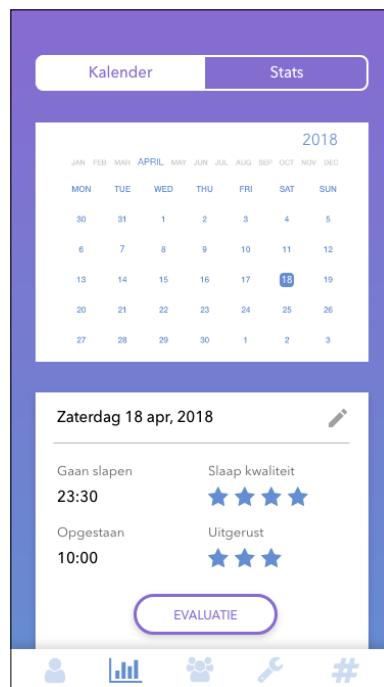
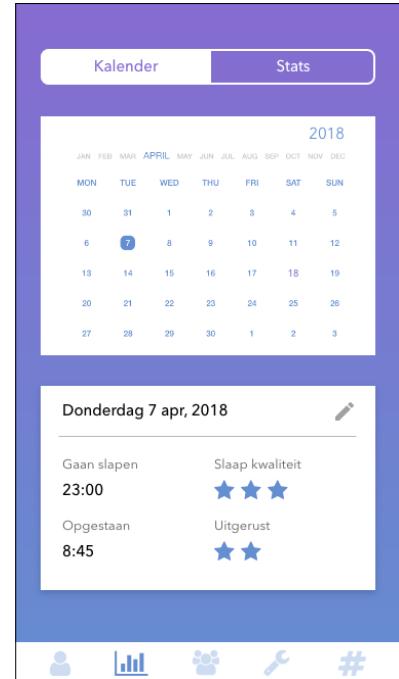
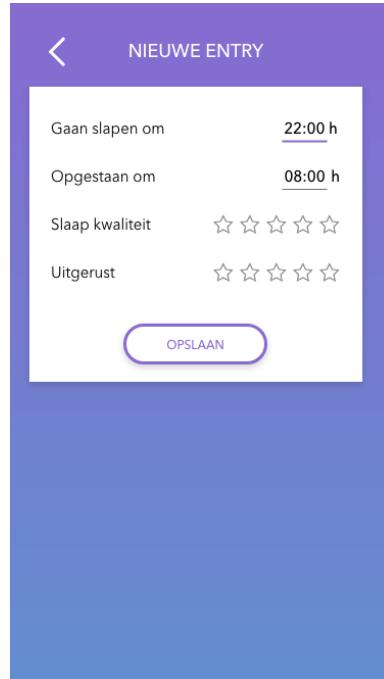
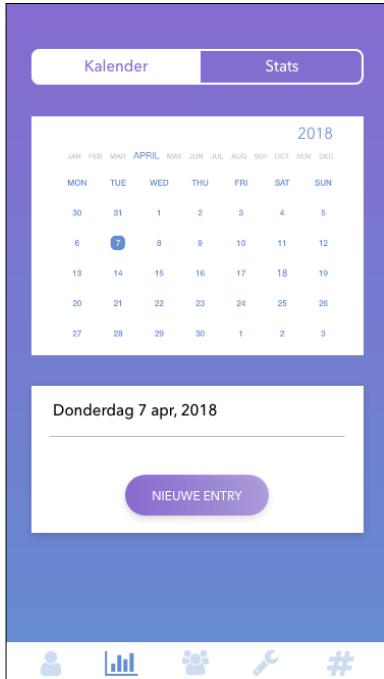


# Visual Designs

## 1. Profiel



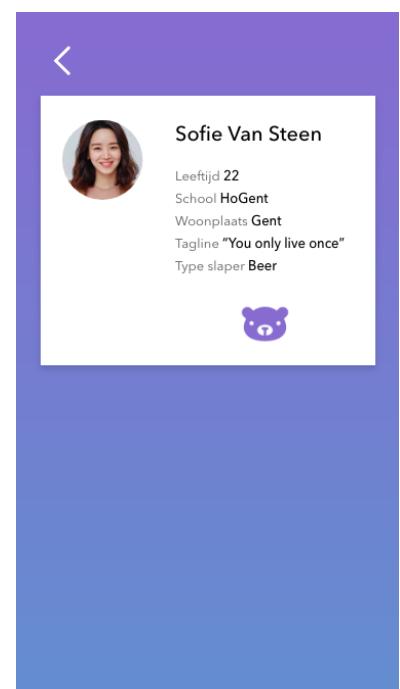
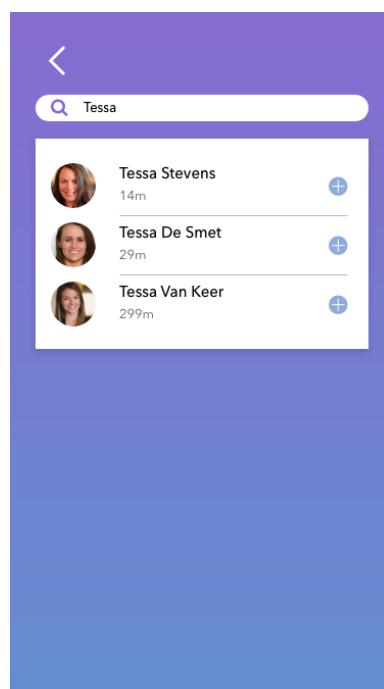
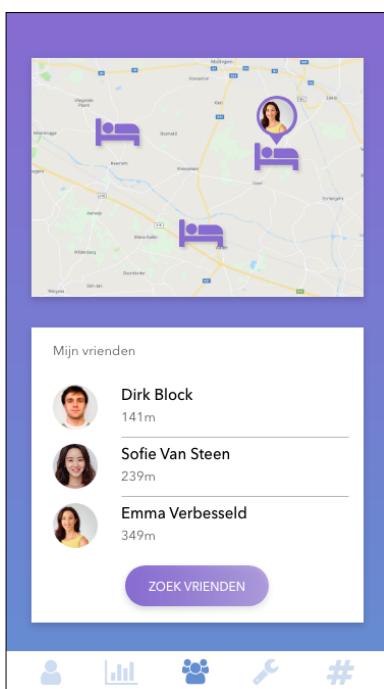
## 2. Kalender



### 3. Statistieken



### 4. Vrienden



## 5. Tips & tools

The image displays two screenshots of the Sleeplog mobile application interface. Both screenshots feature a purple header bar with 'Video's' and 'Tips' buttons. Below the header, there are sections for 'Tips voor Gezonde Slaaphygiëne' and 'Spot the Differences - Rhett & Link'. Each section includes descriptive text and a play button for a video thumbnail. At the bottom of each screen are five blue icons representing user, chart, group, wrench, and hash.

**Tips voor Gezonde Slaaphygiëne**

Zowel overdag als 's avonds voor het slapengaan kun je ervoor zorgen dat je makkelijker de slaap kan vatten en beter dootslaapt.

**Overdag**

1. Kom zoveel mogelijk buiten. Geef jezelf regelmatig de tijd om even tot rust te komen.
2. Let op je voeding. Gezonde en gevarieerde maaltijden geven je meer energie. Beweeg voldoende: wandel, zwem, werk in de tuin, ...
3. Let op je cafeïnegebruik. Dranken zoals koffie, thee, cola of energiedranken stimuleren je hersenen en werken dus ook in op je slaap.

**'s Avonds**

1. Shakel het laatste uur voor je naar bed gaat, je tablet, smartphone of computer uit. Deze

## 6. Wall

The image displays two screenshots of the Sleeplog mobile application interface. The left screenshot shows a 'Wall' feed with posts from users Tessa Stevens and Sven De Backer. The right screenshot shows a 'Post' creation screen with fields for photo upload, description, and tags, and a 'PUBLICER' button.

**Gent**

Likes Datum Afstand

**Tessa Stevens**

In Gent met een latte caramel 😊

#Gent #latte #Starbucks 12 ❤️

**Sven De Backer**

Upload je foto \* KIES

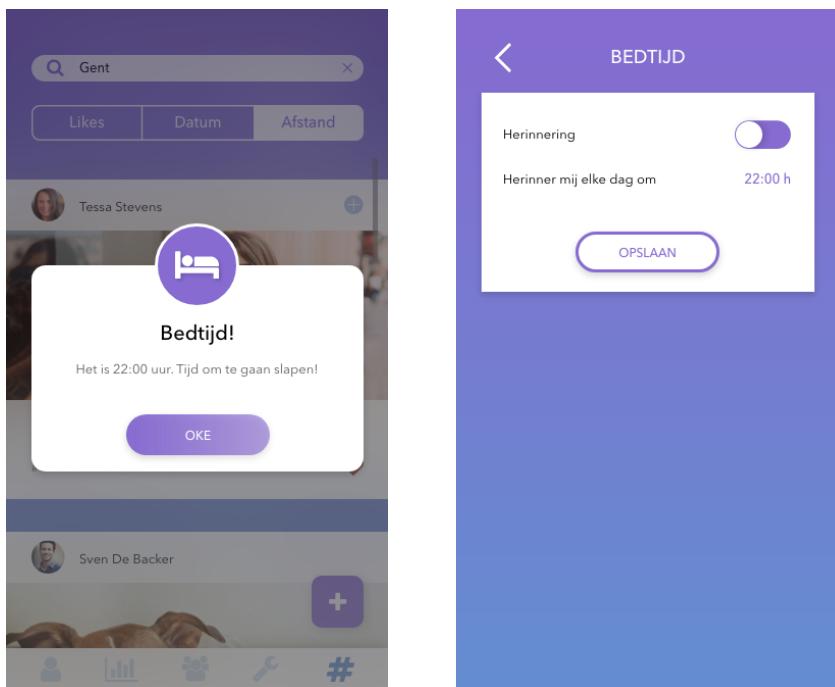
Omschrijving

Tags

Scheid met komma's om meerdere tags toe te voegen

PUBLICER

## 7. Bedtijd



# DEVELOP

# Code snippets

```
<div class="field">
  <label class="label">Email</label>
  <div class="control">
    <input class="input" [(ngModel)]="email" type="text" placeholder="Email" name="email" value="">
  </div>
</div>

<div class="field">
  <label class="label">Wachtwoord</label>
  <div class="control">
    <input class="input" [(ngModel)]="password" type="password" placeholder="Wachtwoord" name="password" value="">
  </div>
</div>

<div class="field">
  <label class="label">Geboortedatum</label>
  <div class="control">
    <input class="input" [(ngModel)]="birthday" type="date" name="birthday" placeholder="dd-mm-yyyy" value="">
  </div>
</div>

<div class="field">
  <label class="label">Gemeente</label>
  <div class="control">
    <input class="input" [(ngModel)]="location" type="text" name="location" placeholder="gemeente" value="">
  </div>
</div>

<div class="field">
  <label class="label">School</label>
  <div class="control">
    <input class="input" [(ngModel)]="school" type="text" name="school" placeholder="school" value="">
  </div>
</div>
```

```
login(logName:string, logPass:string) {
  let formData = new FormData();
  let user_id;
  let user_uid;

  const data = {
    username: logName,
    password: logPass,
    client_id: "3a7b468-55ea-4e65-99c5-09bc8ea061e9",
    client_secret: "root",
    grant_type: "password",
    scope: "user"
  };

  for (let key in data) {
    formData.append(key, data[key]);
  }

  this.userService.login(formData).then(res => {
    let params = new HttpParams();
    params = params.append('filter[name]', this.username);

    this.http.get<JsonObject>(`http://localhost:8888/jsonapi/user/user`, {params: params})
      .subscribe(event => {
        user_id = event.data[0]["id"]
        user_uid = event.data[0]["attributes"]["drupal_internal__uid"]

        localStorage.setItem("uuid", user_id);
        localStorage.setItem("access_token", "Bearer " + res.data.access_token);
        localStorage.setItem("refresh_token", res.data.refresh_token);

        this.createProfile(user_id, user_uid, "Bearer " + res.data.access_token)
      });
  });
}
```

```
<div class="field">
  <label class="label">Profiefoto veranderen</label>
  <div id="upload_button">
    <label>
      <input type="file" (change)="onFileChanged($event)">
      <a>KIES BESTAND</a>
    </label>
    <p>[innerHTML] = "fileName"</p>
  </div>
</div>
```

```
try {
  if (dt != null && nAverage != null) {
    const httpOptionsPatch = {
      headers: new HttpHeaders({
        'Content-Type': 'application/json',
        'Authorization': localStorage.getItem("access_token")
      })
    };
    let body: any =
    {
      type: "goal",
      status: {value: true},
      title: {value: "Goal " + String(dt)},
      field_profile_id: [
        {
          target_id: this.profile_id,
          target_type: "profile",
          target_uuid: this.profile_id,
          url: "/profile/" + this.profile_id
        }
      ],
      field_goal: [{ value: dt }],
      field_achieved: [{value: false}],
      field_average: {value: Math.round( nAverage * 10 ) / 10},
    }
  }

  this.http.post(`http://localhost:8888/node?_format=json`, body, httpOptionsPatch)
    .subscribe(event => {
      console.log(event);
      this.refresh_token();
      this.router.navigate(["profile"]);
    })
  }
  else {
    this.changeMsg("Vul de verplichte velden in!")
  }
}

catch (error) {
  console.error(error);
}
```

```
public async postUser<T>(body: Object): Promise<T> {
  try {
    const axiosResponse = await axios.request<T>({
      method: 'post',
      url: `${this.userApiURL}`,
      headers: {'Content-Type': 'application/json'},
      data: body
    });

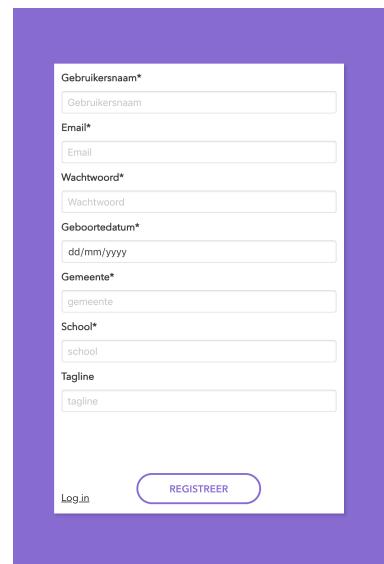
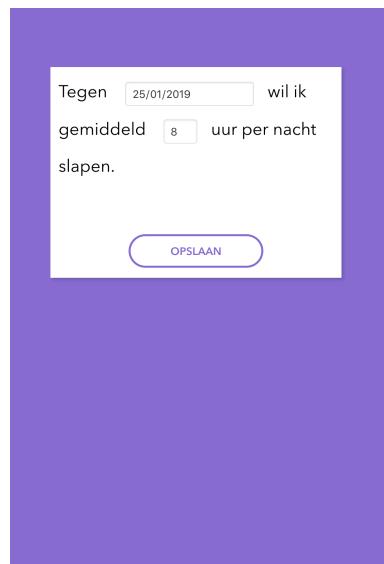
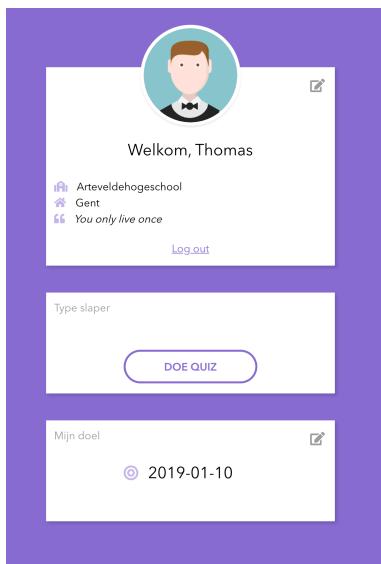
    return( axiosResponse.data );
  } catch (e) {
    return e.response["data"]["message"];
  }
}
```

```
public async getGoals<T>(id: string): Promise<T> {
  try {
    const res = await axios.request<T>({
      method: 'get',
      url: `${this.getGoalURL}?filter[revision_uid.id]=${id}&filter[status]=1`});
    return res.data;
  } catch (error) {
    return Promise.reject(this.handleError(error));
  }
}

public async postGoal<T>(body: Object, token:string): Promise<T> {
  try {
    const axiosResponse = await axios.request<T>({
      method: 'post',
      url: `${this.postGoal}`,
      headers: {
        'Content-Type': 'application/json',
        'Authorization': token
      },
      data: body
    });

    return( axiosResponse.data );
  } catch (e) {
    return e.response["data"]["message"];
  }
}
```

# Schermafbeeldingen



# DELIVER

# Handleiding Mobile Web App

Voordat de app gebruikt kan worden dient de gebruiker zich eerst **in te loggen** of te **registreren**. Bij de registratie kan de gebruiker alvast een aantal persoonlijke gegevens ingeven.

Op het **profiel** worden de gebruikersgegevens weergegeven. Hier kan de gebruiker verdere gegevens zoals zijn **profielfoto** creëren of bewerken. Ook kan hij een **doel** opstellen voor een betere slaap hygiëne. Hiervoor moet hij een gemiddeld aantal uur ingeven dat hij wenst te slapen per nacht. Ook moet hij een datum ingeven waarop hij zijn doel wil bereiken. De gebruiker kan een **herinnering** instellen dat hij elke dag krijgt wanneer het tijd is om te gaan slapen.

Verder kan hij ook een **vragenlijst** invullen dat bepaald welk soort type slaper hij is. Het resultaat wordt daarna weergegeven op het profiel.

Daarnaast moet de gebruiker elke dag gegevens ingeven in de **kalender**. Deze gegevens zijn hoeveel uur hij ongeveer geslapen heeft, hoe diep hij heeft geslapen en hoe uitgeslapen hij zich de dag nadien voelt. Op de dag van de deadline wordt er bepaald of zijn doel wel of niet is bereikt. De gebruiker kan zo elke dag zijn gegevens bekijken. Deze gegevens worden ook weergegeven op een **grafiek** in functie van de laatste 7 dagen.

Daarnaast kan de gebruiker ook zien of er andere gebruikers in de buurt zijn. Dit kan door middel van een **interactieve kaart** waarop de locaties van andere gebruikers te zien zijn. De gebruiker kan het profiel van een andere gebruiker bekijken en omgekeerd.

De gebruiker kan ook terecht bij “**tips & tools**”. Hier worden tips weergegeven die kunnen helpen bij een betere slaap hygiëne. Ook kan de gebruiker informatieve filmpjes bekijken ivm slaap.

Tenslotte kan de gebruiker foto's uploaden op een **tijdlijn** met een eventuele bijschrift en/of hashtag. Deze tijdlijn is openbaar en wordt gedeeld met elke gebruiker. Gebruikers kunnen elkaar foto's liken door op het hart-icoontje te klikken.

# DEPLOY

# Deployment Guide

This project was generated with [Angular CLI](#) version 6.2.4.

## Development server

Run **ng serve** for a dev server. Navigate to **http://localhost:4200/**. The app will automatically reload if you change any of the source files.

## Code scaffolding

Run **ng generate component component-name** to generate a new component. You can also use **ng generate directive|pipe|service|class|guard|interface|enum|module**.

## Build

Run **ng build** to build the project. The build artifacts will be stored in the **dist/** directory. Use the **--prod** flag for a production build.

## Running unit tests

Run **ng test** to execute the unit tests via Karma.

## Running end-to-end tests

Run **ng e2e** to execute the end-to-end tests via Protractor.

## Further help

To get more help on the Angular CLI use **ng help** or go check out the Angular CLI README.

## Bijlage: timesheet

Student Sam De Smedt - CMSDEV		
Datum	Taak	Tijd
5/12/2018	concept uitwerken	5,00
6/12/2018	wireframes/wireflow	3,00
7/12/2018	visuals	3,00
8/12/2018	visuals	2,50
9/12/2018	visuals	3,00
10/12/2018	style guide	4,00
11/12/2018	Drupal setup, angular components	5,00
12/12/2018	profile: components, rest api	6,00
14/12/2018	profile: components, rest api, create data	4,50
15/12/2018	profile: file upload, data fields	4,00
16/12/2018	profile: file upload, patch user info, lay out	10,50
20/12/2018	form validation, error handling, lay out	7,00
23/12/2018	profile: edit, patch	5,00
25/12/2018	auth: login component, access token	9,00
26/12/2018	auth: access token	6,50
28/12/2018	auth: register component, lay out	8,00
29/12/2018	auth: register component, error logging	7,75
1/01/2019	goals: data fields, create, components	9,00
2/01/2019	goals: data fields, edit, lay out	8,00
3/01/2019	quiz: forms, lay out	9,00
4/01/2019	dossier samenstellen	9,00
		<b>Totaal:</b> 128,75