

ArteTech

(Headless) Symfony & React



arteveldehogeschool
LID VAN DE ASSOCIATIE UNIVERSITEIT GENT

**CMSDEV
2019-20**

Jan Temmerman

New Media Development

**Bachelor in de grafische en digitale media
Arteveldehogeschool**

Inhoud

Discover 5

Briefing 6

Briefing 7

Define 8

Planning 9

Analyse 10

Design 11

Sitemap 12

Wireframes 13

Wireflow 14

Style Guide 15

Visual Designs 16

Develop 20

Code Snippets 21

Schermafbeeldingen 25

Deploy 28

Deployment Guide 29

Discover

Briefing

Introductie

Jouw klant ARTE-TECH, een technisch onderhoudsbedrijf komt met de vraag om de opvolging van hun werken en opvolging van het personeel, aannemers, onderaannemers, ... te optimaliseren.

Activiteiten van de klant

ARTE-TECH biedt de dienst aan om technisch geschoolde mensen gedurende een bepaalde periode ter plaatse naar bedrijven te sturen om daar gevraagde onderhoudswerken uit te voeren. Zowel tijdens een shutdown als voor het dagelijks onderhoud zorgen ze ervoor dat de juiste mensen snel op locatie zijn om assistentie te bieden of volledig autonoom een machine te repareren.

Probleemstelling

ARTE-TECH beschikt enerzijds over eigen personeel, maar beschikt anderzijds ook over een aantal onderaannemers (vergelijkbaar met freelancers) die ze - in naam van ARTE-TECH - kunnen uitscharen naar bedrijven. De opvolging hiervan is moeilijk want iedereen heeft een andere manier van werken.

De freelancers werken met eigen werkbonnen (startuur, einduur, activiteiten van de dag, verplaatsingen, ...).

Personeel houdt dan weer alle uren en activiteiten op papier bij en laat die wekelijks ondertekenen door hun klant (het bedrijf waar ze onderhoud doen).

Elke week verzamelt de manager van ARTE-TECH alle paperassen, werkbonnen en e-mails en stelt op basis daarvan de factuur op.

Maar er geraken al eens papieren, e-mails verloren en sommige zaken worden vergeten door gebrek aan uniformiteit.

Er is een oplossing nodig.

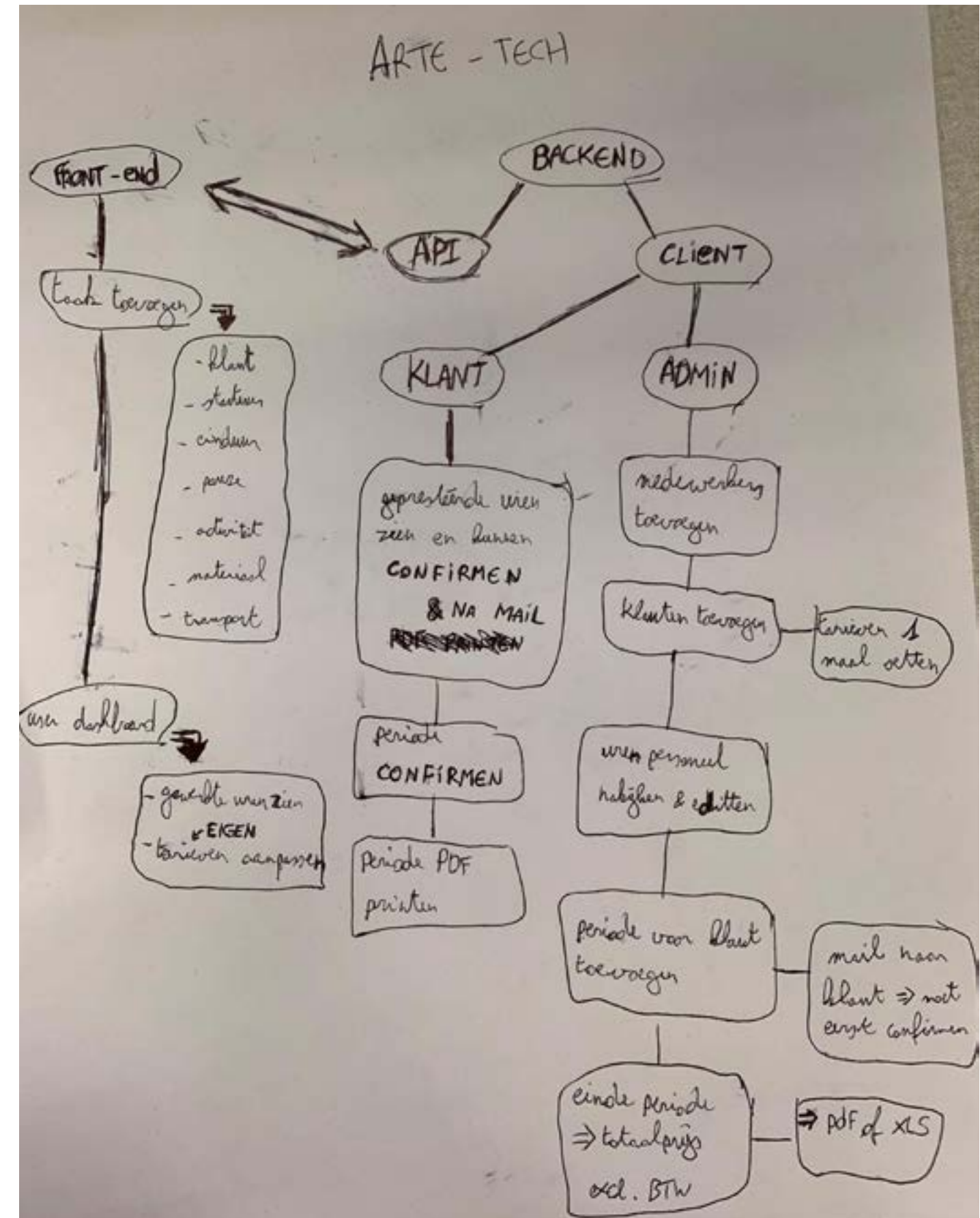
Opdracht

De klant heeft weinig kaas gegeten van computers, maar beseft wel dat zijn manier van werken een stuk gemakkelijker zou kunnen gemaakt worden door het maken van een webapplicatie. Hij had zelf het volgende idee:

Een all-in-one webapplicatie waarbij hij de uren niet meer zelf hoeft in te vullen, maar reeds gebeurd is door het personeel / onderaannemers die hij uitscharen.

Visuele presentatie

Voor mezelf maakte ik een visuele presentatie van hoe de opdracht er uit zag.



Define

Analyse

Planning

Planning

Mijn planning

Ik maak meestal geen plannings, maar werk volgens een volgorde die ik het meest gepast vind.

Opdracht begrijpen

Eerst probeer ik ervoor te zorgen dat ik de opdracht goed begrijp. De manier waarop hangt af van de complexiteit van de opdracht. Bij kleine opdrachten volstaat het om deze gewoon te lezen. Bij complexere opdrachten zoals deze maak ik een visuele voorstelling om mezelf te helpen doorheen de realisatie ervan.

Database

Bij opdrachten zoals deze, degen die een database gebruiken, vind ik het belangrijk te starten met het maken van een database model. Dit omdat dit meestal toch het eerste is wat je moet doen en omdat ik dan graag de hele database in 1 keer aanmaak, met alle tabellen vanaf het begin. Zodat ik later geen conflicten meer krijg.

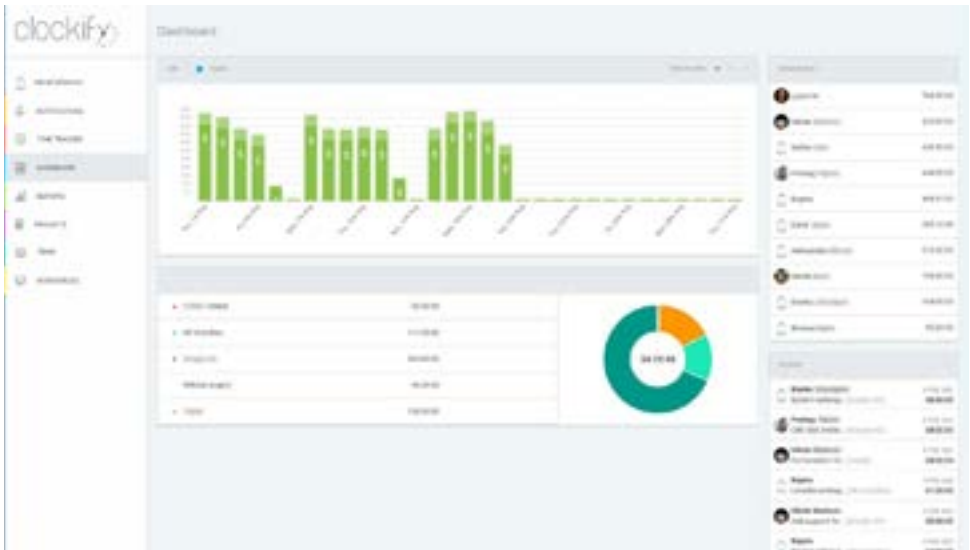
Coderen

Het coderen begin ik, zoals hierboven gezegd, met het aanmaken van de database en de tabellen. Wanneer dit gelukt is begin ik de controllers en pagina's aan te maken.

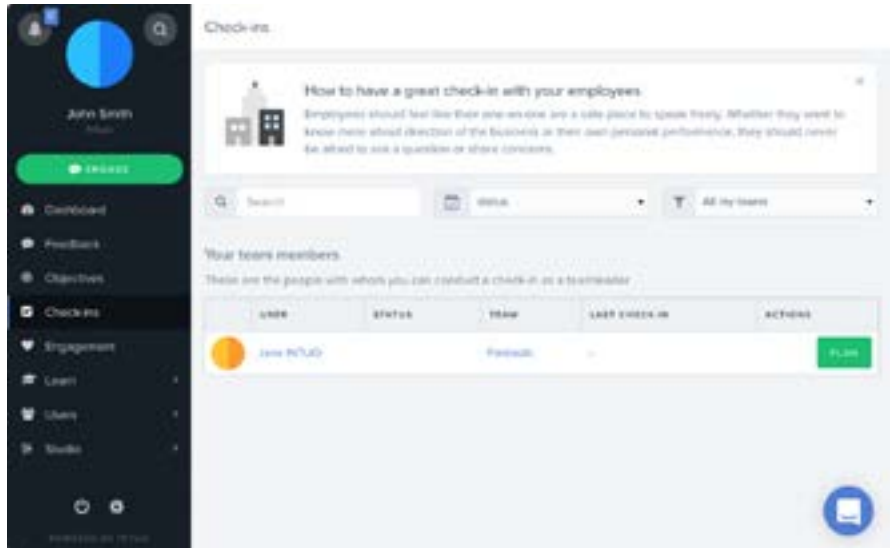
Concurrentie

Er zijn veel verschillende werkenemer management systemen, maar geen een die precies werkt zoals onze klant (ArteTech) wil.

Clockify, hier geven werknemers hun uren in en wordt er een mooi overzicht gemaakt voor de admins. Geen optie voor de klant om het werk te zien dat voor hem verricht is.



Intuo, bij dit pakket geldt eigenlijk hetzelfde verhaal, werknemers managen zonder optie voor de klant om iets te kunnen bekijken.



Design

Sitemap

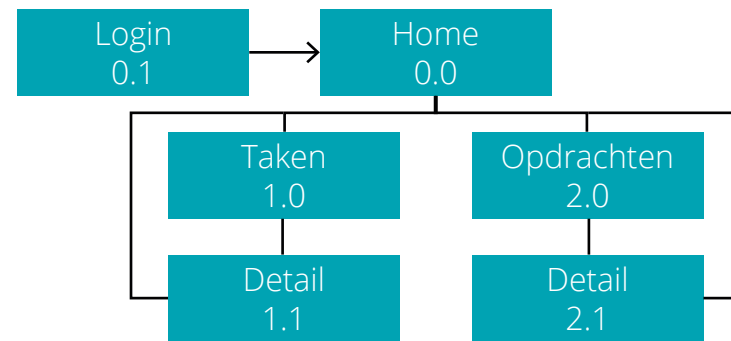
Wireframes

Wireflow

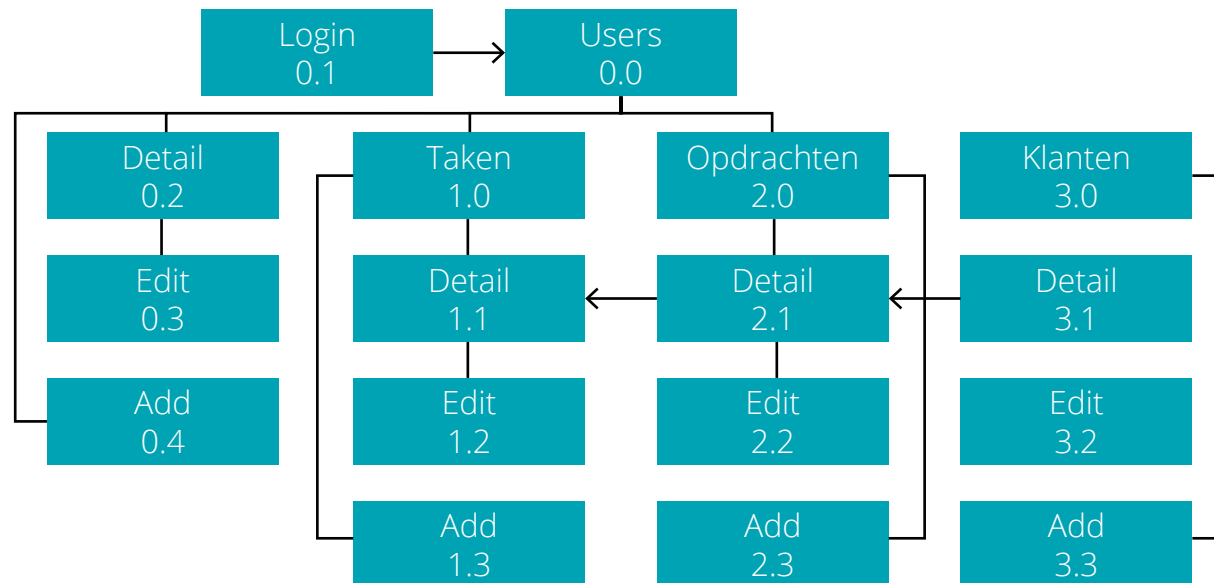
Style Guide

Visual Designs

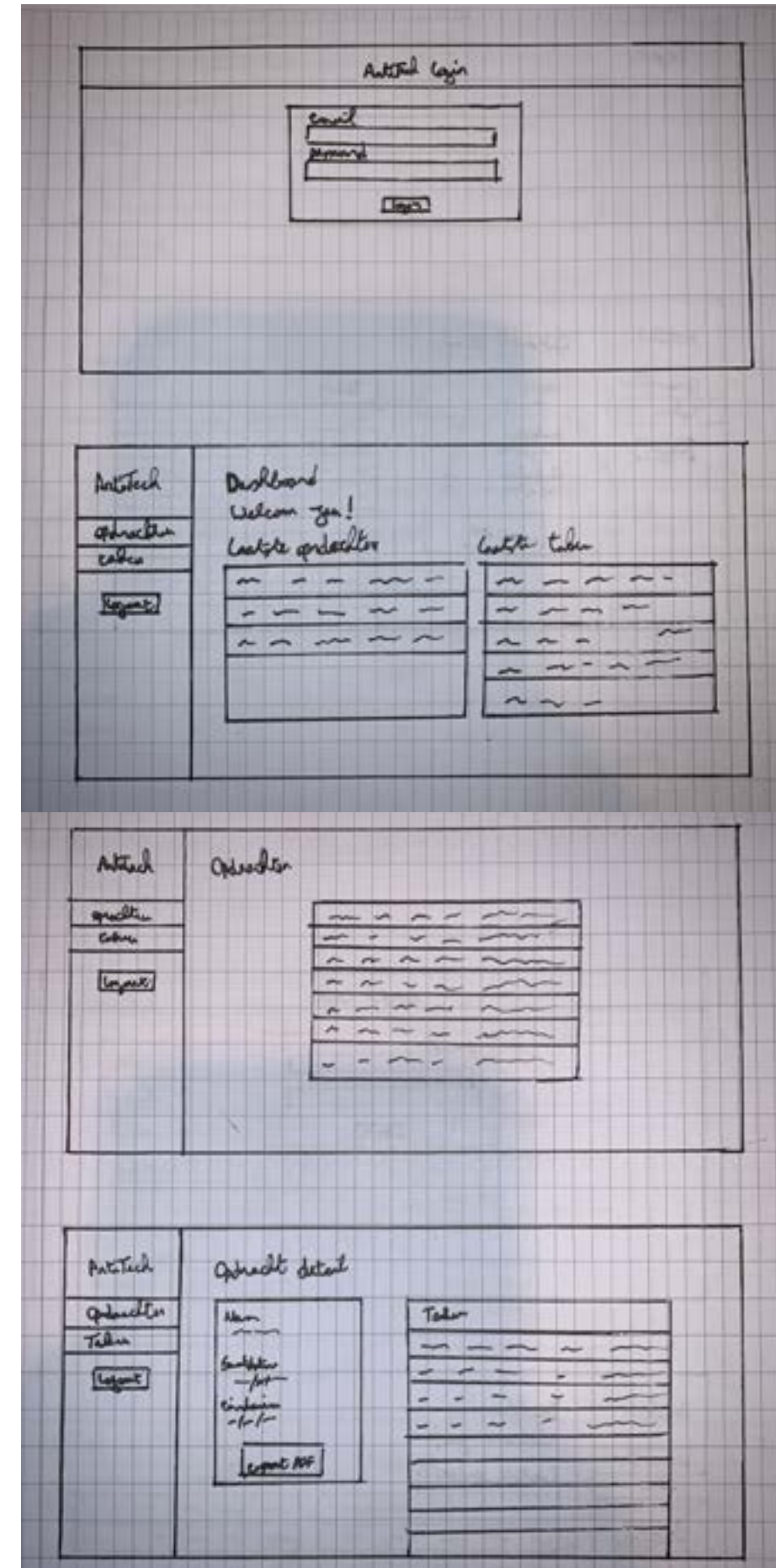
Website klant gedeelte



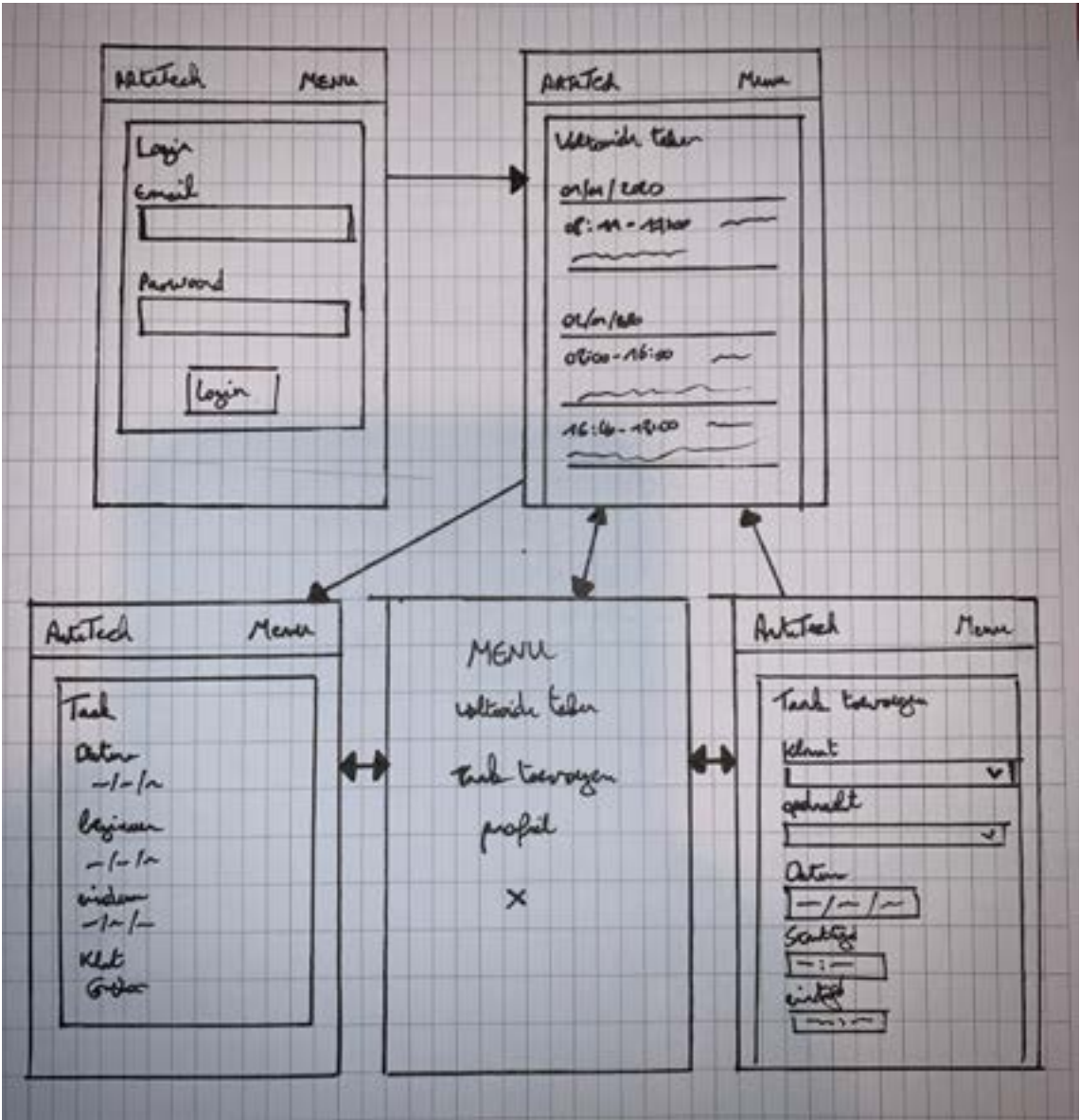
Website admin gedeelte



Website klant gedeelte



Mobile web app



Kleurenpalet

#080b11

#00a3b3

#f7faff

#04727e

#233641

Fonts

Open Sans

Open sans

Open Sans

Buttons

Button

Menu's

Menu

Voltooid Taken

Taak Toevoegen

Profiel

X

ArteTech

Gebruikers

Klanten

Opdrachten

Voltooid Taken

Mobile web app

The mobile app screens for ArteTech are as follows:

- Please sign in:** A login screen with fields for Email and Password, and a Sign in button.
- Voltooide Taken:** A list of completed tasks. Each task entry includes a date, time range, location, and a brief description.

Date	Time	Location	Task
02/01/2020	08:20 - 17:35	G-Build	bouten aangedraait
30/12/2019	22:50 - 22:50	Ibize	zv
07:00 - 18:00	Ibize	Vijzen gedraaid	
13:30 - 17:54	Ibize	ihbib	
07:15 - 17:30	G-Build	gyproc geplaatst	
01/01/2020	07:15 - 19:30	Ibize	jbjb
30/12/2019	00:52 - 00:52	Ibize	jbjb
- Taak:** A detailed view of a task.

Datum: 02/01/2020

Gewerkt Van Tot: 08:20 - 17:35

Gewerkt Voor: G-Build

Werk Verricht: bouten aangedraait

Materiaal Gebruikt: bouten

Lengte Pauze: 01:00

Km's afgelegd: 555
- Menu:** A central menu screen with options: Voltooide Taken, Taak Toevoegen, and Profiel.
- Taak toevoegen:** A form to add a new task.

Klant: Select...

Opdracht: Select...

Datum: 02/01/2020

Start tijd: 23:03

Eind tijd: 23:03

Lengte Pauze: Select...

Uitgevoerde activiteiten:

Materialen:
- Profiel:** A user profile screen showing:

Naam: Hannelore Temmerman

Email: hannelore@gmail.com

Status: employee

Logout button

Admin Website

The Admin Website interface for ArteTech includes:

- ArteTech Login:** A login screen with fields for Email and Password, and a Login button.
- Gebruikers:** A user management section with a sidebar menu (Gebruikers, Klanten, Opdrachten, Voltooide Taken) and a main area with a 'Gebruiker Toevoegen' button.
 - Admins:** List of administrators: Jan Temmerman.
 - Clients:** List of clients: Jan Temmerman, Anna De Langhe.
 - Freelancers:** List of freelancers: Jan Temmerman, Wouter Vlaeyen.
 - Employees:** List of employees: Jan Temmerman.

Admin Website

ArteTech

Gebruikers

Klanten

Opdrachten

Voltooide Taken

Opdrachten

Opdracht Toevoegen

Opdracht	Looptijd	Klant	Details
Johnny needs your help	02/01/2020 - 24/01/2020	ibite	Meer...
Make some fat meatballs	01/01/2020 - 06/02/2020	ibite	Meer...
Deuren plaatsen	22/02/2019 - 12/04/2019	ibite	Meer...
Bureau opbouw	21/12/2019 - 31/01/2020	ibite	Meer...
Bouw keuken	31/01/2020 - 22/03/2020	G-Build	Meer...
Renovatie hq	11/12/2019 - 31/01/2020	G-Build	Meer...

ArteTech

Gebruikers

Klanten

Opdrachten

Voltooide Taken

Opdrachten

Opdracht Details

Info Aangepassen

Opdracht

Bureau opbouw

Looptijd

21/12/2019 - 31/01/2020

Klant

ibite

Uurloon

20 euro/uur

Transport Vergoeding

0.3 euro/km

Totaal aantal uren gewerkt

25.08

Totaal aantal km te vergoeden

8 km

Totale Prijs (op dit moment)

€564

Exporteer PDF

Datum	Uur	Werknaam	Klant	Opdracht	Details
Fri 21/12/2019	18:50 - 20:15	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...
Mon 30/12/2019	00:52 - 00:52	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...
Wed 01/01/2020	07:15 - 19:30	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...
Mon 30/12/2019	13:30 - 17:54	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...
Mon 30/12/2019	07:00 - 18:00	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...
Mon 30/12/2019	22:00 - 22:00	Hannelore Tenkerman	ibite	Bureau opbouw	Meer...

Admin Website

ArteTech

Gebruikers

Klanten

Opdrachten

Voltooide Taken

Opdracht Toevoegen

Vul De Velden In

Klant

G-Build

Startdatum

01-01-2111

Einddatum

01-01-2111

Opdracht

Uurloon

20 euro/uur

Transport Vergoeding

0.3 euro/km

Opslaan

Develop

Code Snippets

Schermafbeeldingen

Functie in de PageController waar we alle statuses ophalen en de twig file renderen.

```
/**
 * @Route("/users", name="users")
 * @param Request $request
 * @return Response
 */
public function users(Request $request)
{
    $statuses = "";
    $isUnauthorized = false;

    if($this->isAdmin()) {
        $repository = $this->getDoctrine()->getRepository( persistentObject: UserStatus::class);
        $statuses = $repository->findAll();
    } else {
        $isUnauthorized = true;
    }

    return $this->render( view: 'page/users.html.twig', [
        'statuses' => $statuses,
        'title' => 'Gebruikers',
        'isUnauthorized' => $isUnauthorized,
    ]);
}
```

In onderstaande twig file tonen we alle users gegroepeerd per status.

```
{% extends 'base.html.twig' %}

{% block title %}{{ title }}{% endblock %}

{% block body %}

<div class="heading_container">
    <h1>{{ title }}</h1>
    <a class="button-add" href="{{ path('addUser') }}">Gebruiker Toevoegen</a>
</div>

<div class="content-container">
    {% for status in statuses %}
        <div class="card">
            <h2>{{ status.status|capitalize }}</h2>
            {% for user in status.users %}
                <p>{{ user.firstName }} {{ user.lastName }}</p>
            {% endfor %}
        </div>
    {% endfor %}
</div>

{% endblock %}
```

Hieronder zien we een functie van de ApiController die alle periodes terug geeft met de Serializer.

```
/**
 * @Route("/api/periods/getAll", name="api_periods_getAll")
 * @return Response
 * @method GET
 * @throws AnnotationException
 * @throws ExceptionInterface
 */
public function getPeriods()
{
    $periodsRepository = $this->getDoctrine()->getRepository( persistentObject: Period::class);
    $period = $periodsRepository->findAll();

    $classMetadataFactory = new ClassMetadataFactory(
        new AnnotationLoader(
            new AnnotationReader()
        )
    );

    $norm = [ new DateTimeNormalizer(), new ObjectNormalizer($classMetadataFactory)];
    $encoders = [new JsonEncoder()];
    $serializer = new Serializer($norm, $encoders);

    $jsonContent = $serializer->normalize(
        $period,
        format: 'json', ['groups' => ['period_safe', 'hourlyRate_safe', 'transportRate_safe', 'company_safe']]
    );

    $jsonContent = json_encode($jsonContent);

    return new Response($jsonContent, status: 200, ['Content-Type' => 'application/json']);
}
```

Dit is een fetch naar de Symfony api vanuit de React app.

```
const fetchTasks = () => {
    const token = localStorage.getItem('bearer')
    const user = JSON.parse(localStorage.getItem('user'))

    fetch('http://localhost:8000/api/tasks/getFromUser', {
        method: 'POST',
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json',
            'Authorization': 'Bearer ' + token,
        },
        body: JSON.stringify({
            id: user.id,
        })
    })
    .then(response => response.json())
    .then(data => {
        if(data.code)
            handleBadJWT()
        else {
            setTasks(data.reverse())
            setIsFetching(false)
        }
    })
    .catch(error => console.error(error))
}
```

Wanneer de JWT lokaal verwijderd wordt of expired is wordt onderstaande functie aangeroepen.

```
const handleBadJWT = () => {
    alert("Session timed out. Please log in.")
    localStorage.removeItem('bearer')
    localStorage.removeItem('user')
    history.push('/login')
}
```

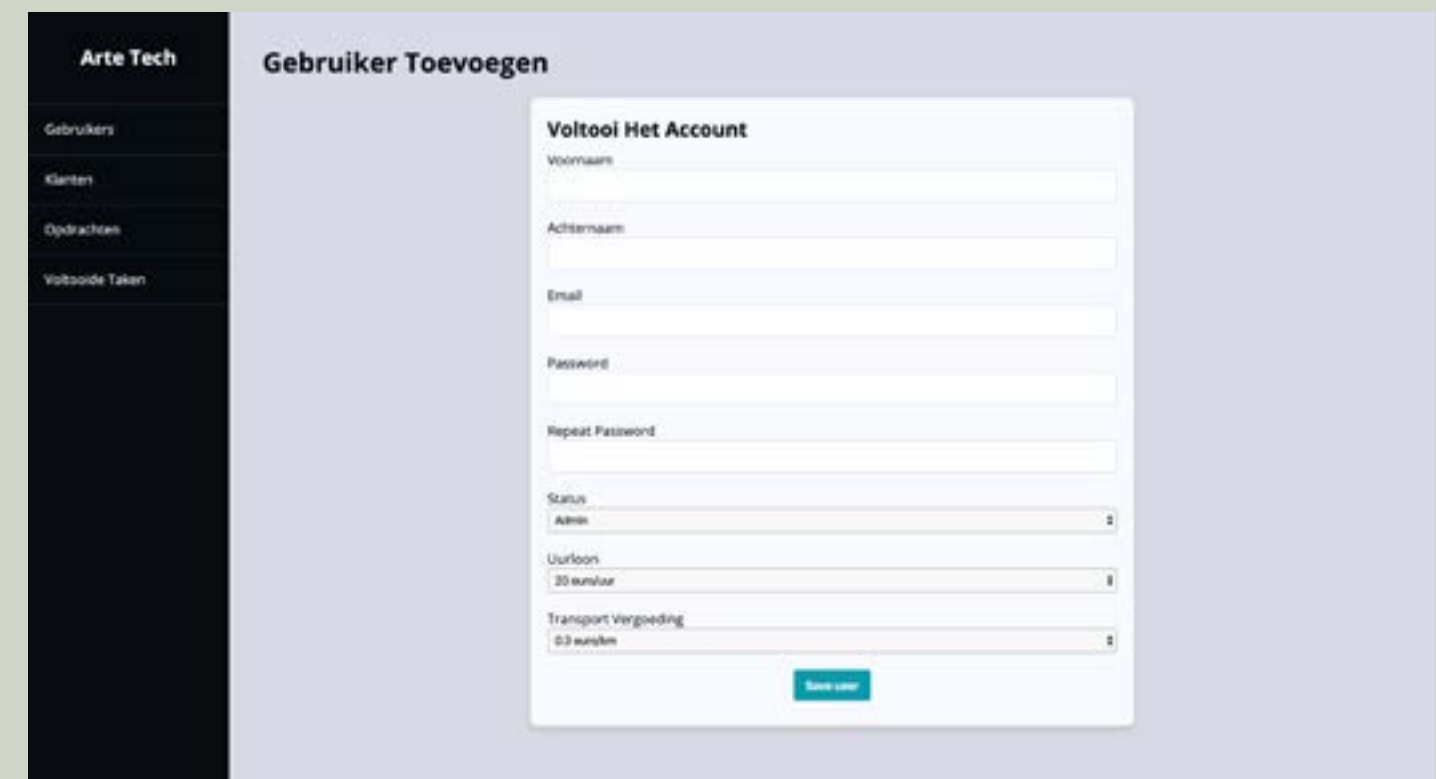
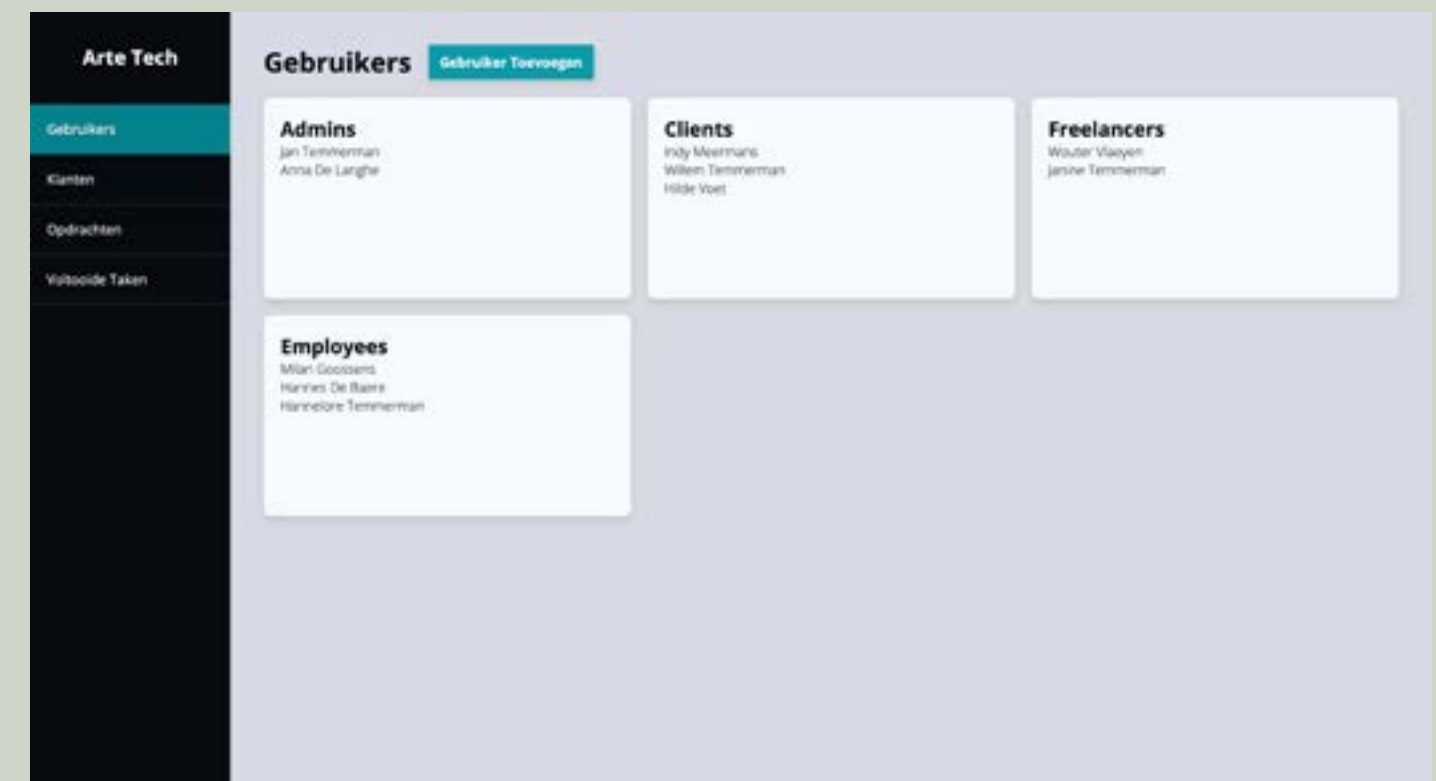

Dit is de functie binnen de React app die een nieuwe taak post naar de database door middel van de api.

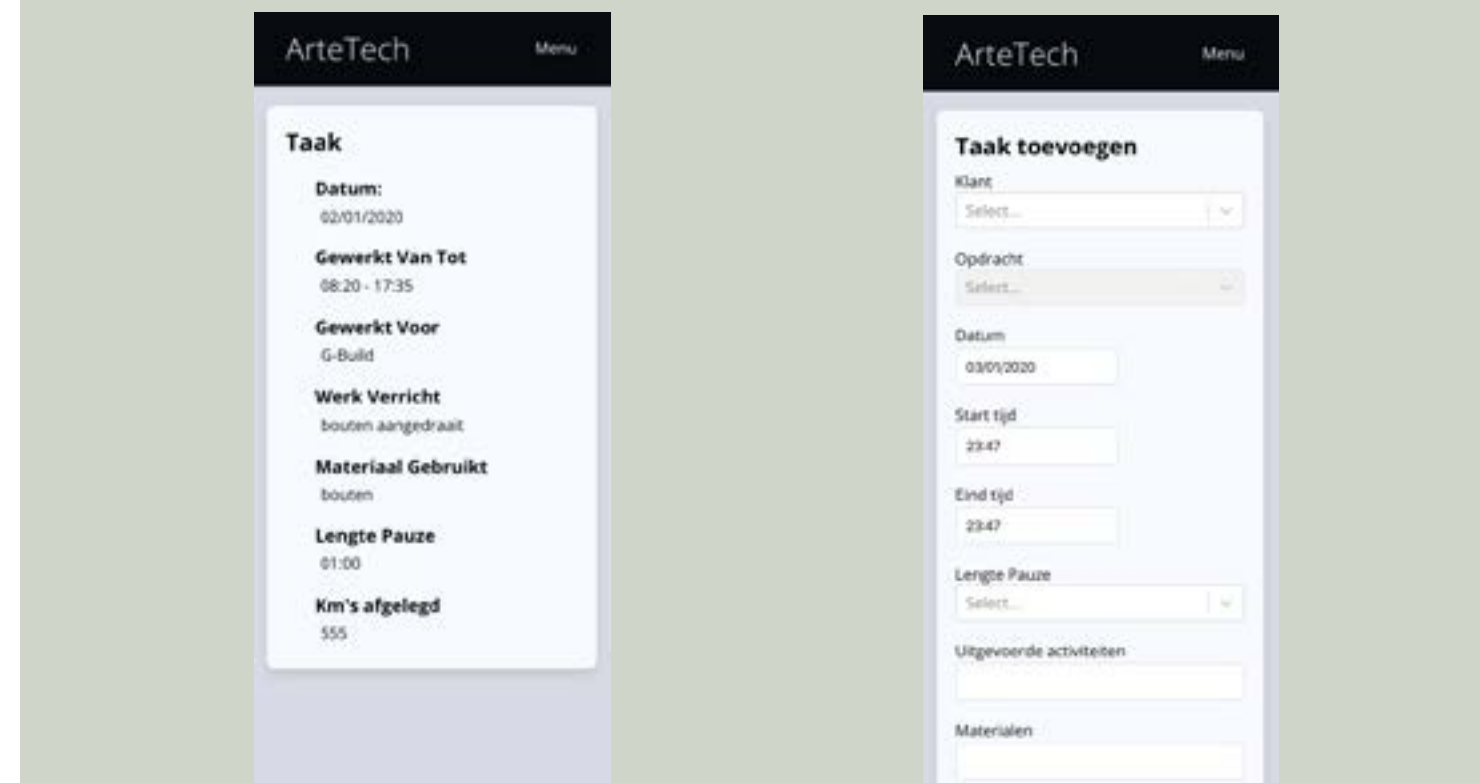
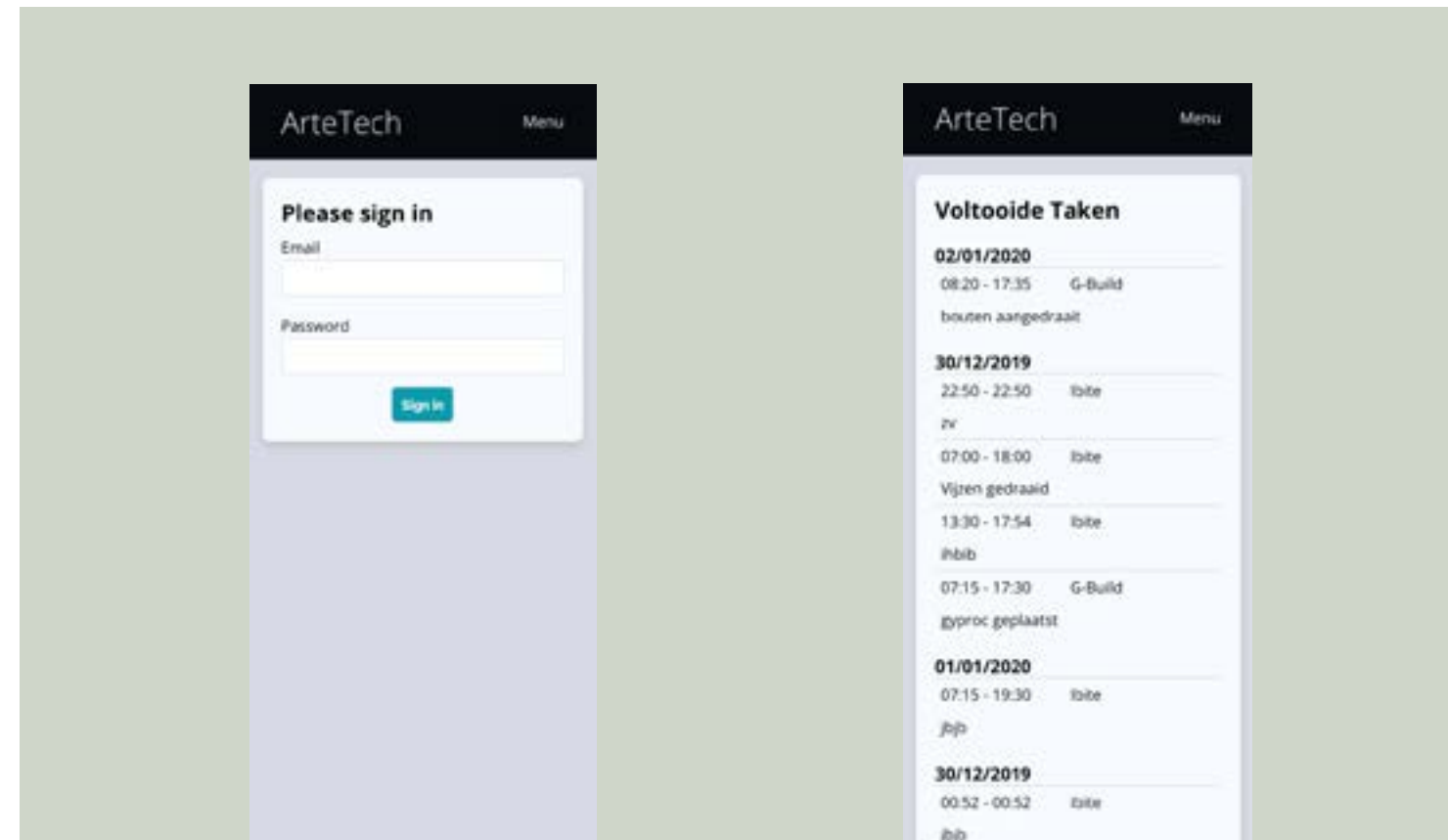
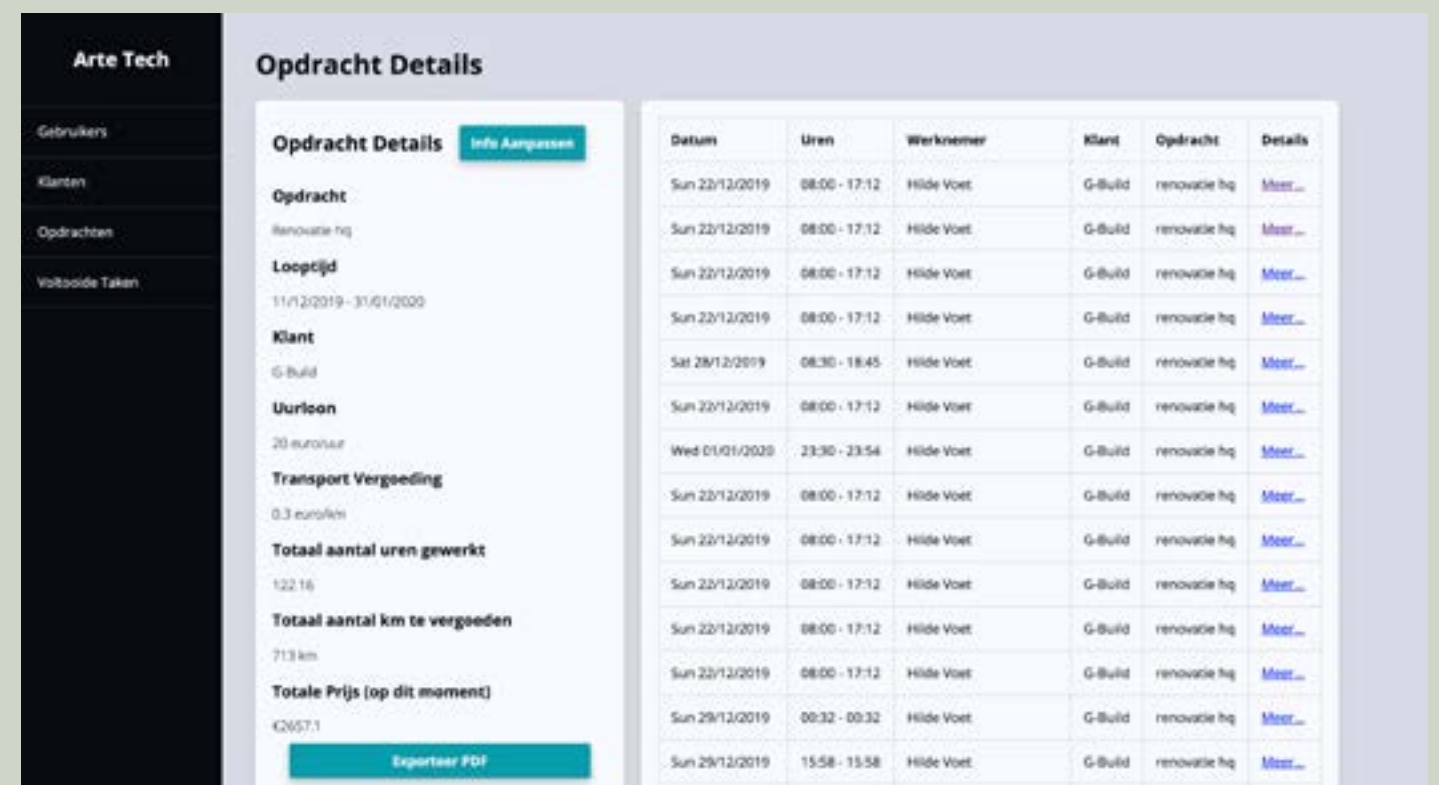
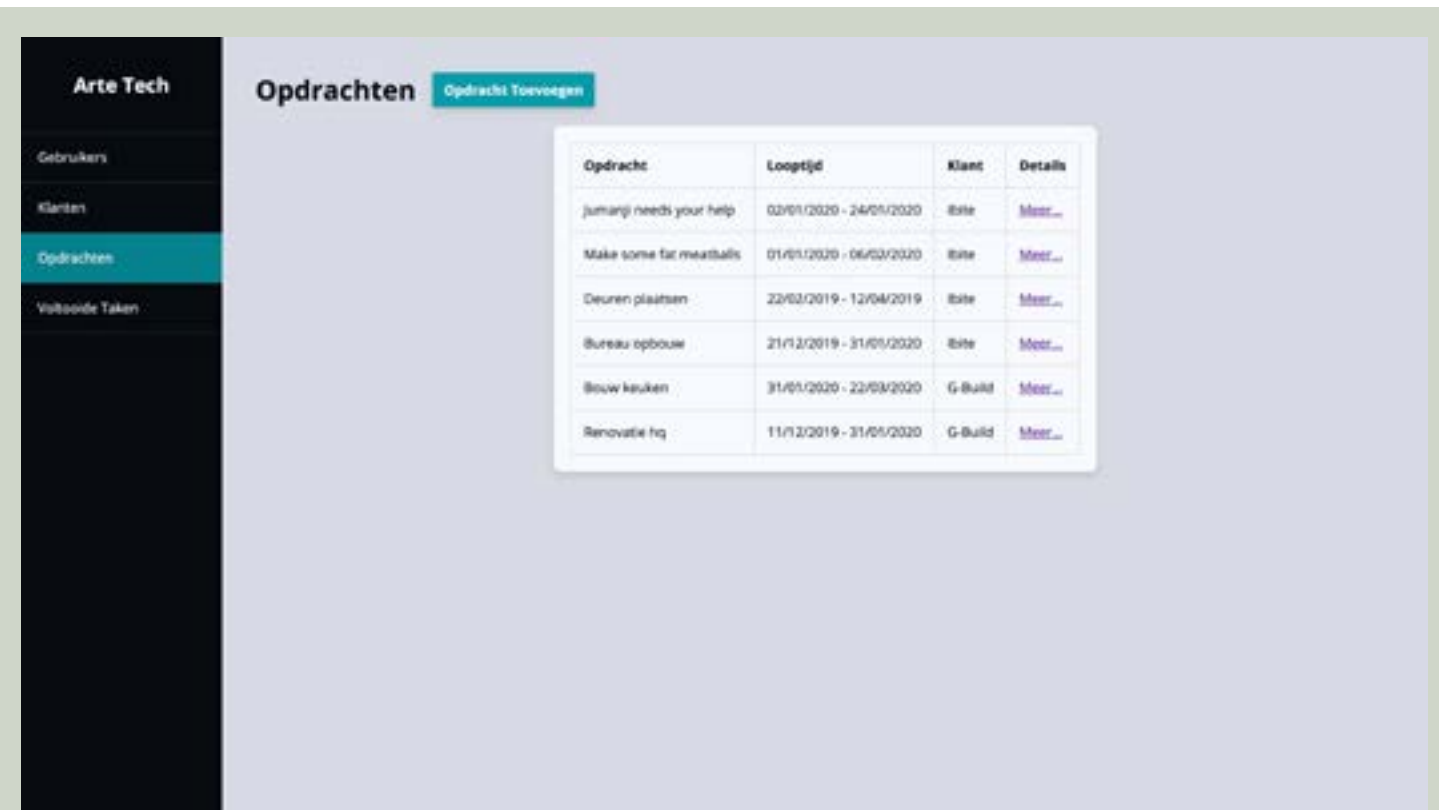
```
const postTask = () => {
  const token = localStorage.getItem('bearer')
  const user = JSON.parse(localStorage.getItem('user'))

  const data = {
    'employee_id': user.id,
    'period_id': selectedPeriod.value,
    'pause_id': selectedPause.value,
    'date': format(date, "yyyy-MM-dd", { awareOfUnicodeTokens: true }),
    'time': {
      'start': format(startTime, "HH:mm:ss", { awareOfUnicodeTokens: true }),
      'end': format(endTime, "HH:mm:ss", { awareOfUnicodeTokens: true })
    },
    'activities_done': activity,
    'materials_used': materials,
    'km': parseInt(km)
  }

  fetch('http://localhost:8080/api/tasks/setTask', {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
      'Authorization': 'Bearer ' + token,
    },
    body: JSON.stringify(data)
  })
  .then(response => response.json())
  .then(data => {
    if(data.code)
      handleBadJWT()
    else {
      if(data.status === "400")
        setErrorContent(<p className="error">{data.message}</p>)
      else {
        setErrorContent("")
        history.push('/')
      }
    }

    console.log(data)
  })
  .catch(error => console.error(error))
}
```





Deploy

Deployment Guide

Deployment Guide

Voer onderstaande commando's uit in de terminal na het navigeren naar de map van het project en na het starten van de MAMP server

- cd artetech
- Composer install
- php bin/console doctrine:migrations:migrate
- symfony server:start
- cd..
- cd app
- yarn install
- yarn start