

Mobile Development I

Kevin Leemans

2017 - 2018



arteveldehogeschool

LID VAN DE ASSOCIATIE UNIVERSITEIT GENT

Arteveldehogeschool
Campus Mariakerke
Grafische & Digitale Media

Inhoud

1.	Discover	3
	1. Briefing	3
2.	Define	4
	1. Analyse	4
	2. Planning	9
	3. Ideaboard	10
3.	Design	11
	1. Sitemap	11
	2. Wireframes	12
	3. Wireflow	15
	4. Style Tile	16
	5. Visual Designs	17
4.	Develop	20
	1. Code	20

1. Discover

1. Briefing

Request For Proposal

Ga opzoek naar een systeem waarbij het portfolio van studenten van een opleiding getoond kan worden aan de wereld. Het systeem moet gebruiksvriendelijk en aantrekkelijk zijn om nieuwe studenten aan te trekken.

Benodigde functionaliteiten:

- Bezoeker
 - Portfolio
 - Overzicht werkstukken
 - Detail van werkstuk
 - Opleiding
 - Specialisaties
 - Blog
 - Detail van artikels
 - Contact
 - Social media
 - Registreren
- Geregistreerde gebruiker
 - Werkstukken
 - Like
 - Aantal views
 - Toevoegen aan favorieten
 - Delen op social media
 - Student
 - Volgen
- Extra
 - Profielpagina van student met overzicht van werkstukken
 - Reageren op een werkstuk

2. Define

1. Analyse

GDM Gent Trots



Goed:

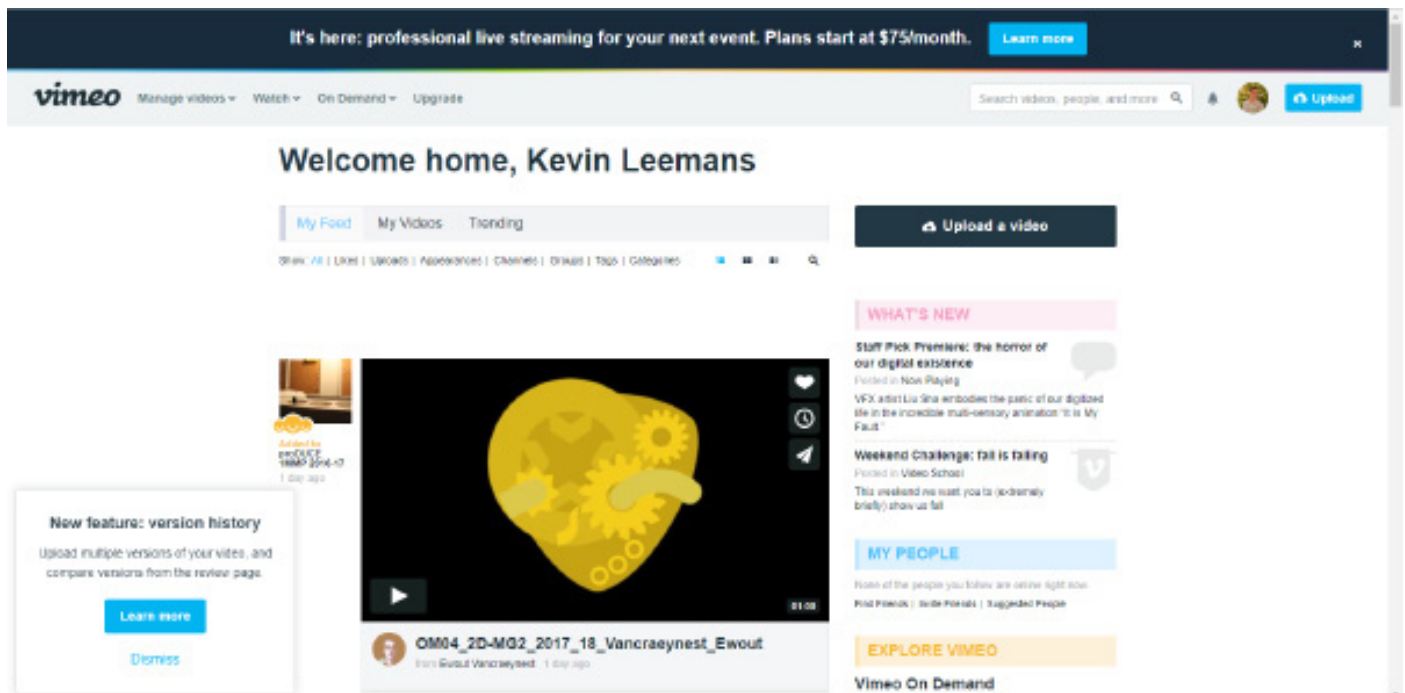
- Eerste wat je ziet op de website is een inleidingsfilmpje. Vind dit wel tof
- Ze splitsen interne en externe projecten van de studenten. Misschien wel een goed idee om dit te implementeren in ons werkstuk
- Ook leuk dat je op de home pagina misschien al wat projecten kan zien die "uitgelicht" staan

Minder goed:

- Deze button popt uit over het volledige scherm. Is goed voor mobile maar voor desktop misschien wat overdreven
- Kan niet specifiek zoeken naar een student
- Kan je niet aanmelden of registreren



VIMEO (Web-App)



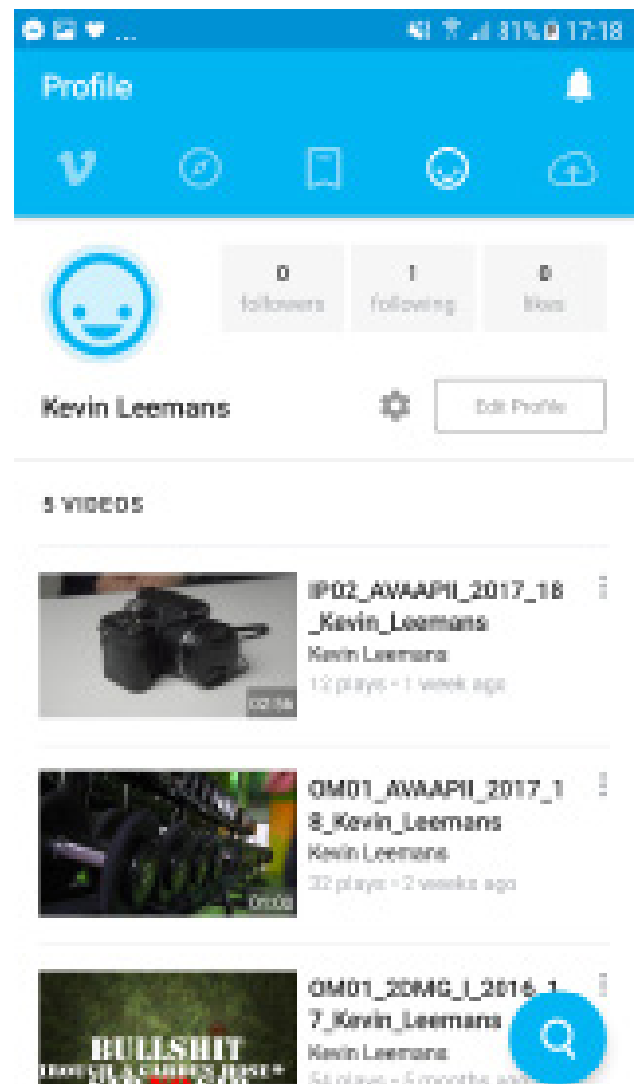
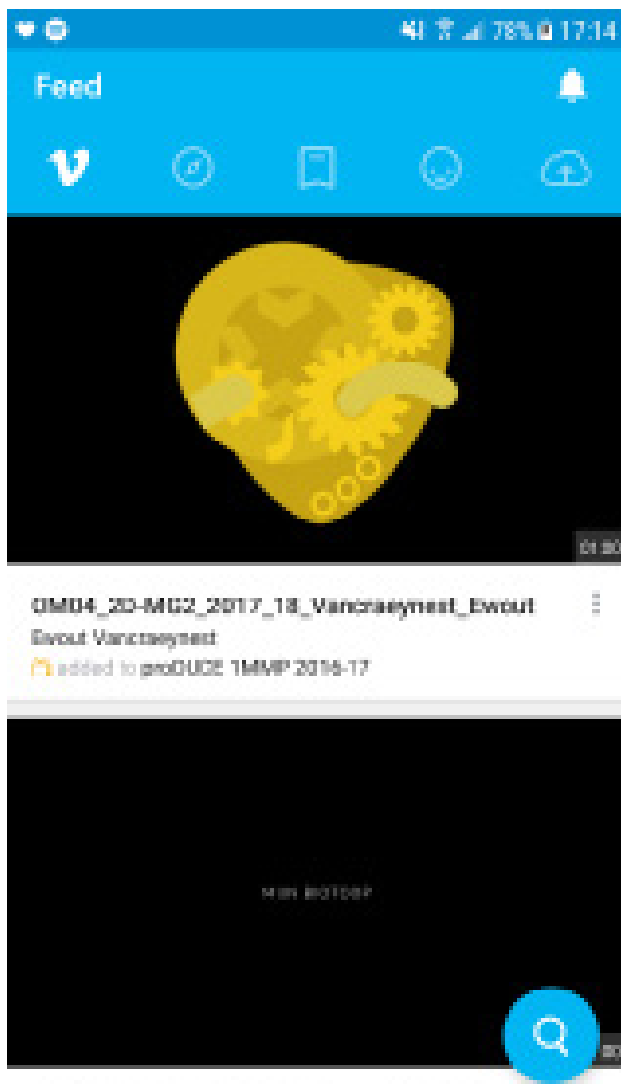
Goed:

- Als je aangemeld bent staan er op de homepagina video's (projecten) die bij jouw groep horen
- Search bar die u mensen, groepen laat zoeken
- Gemakkelijk mensen toevoegen

Minder goed:

- Moeilijk voor mobile te maken / te druk (zij hebben wel een app)

VIMEO (Mobile-App)



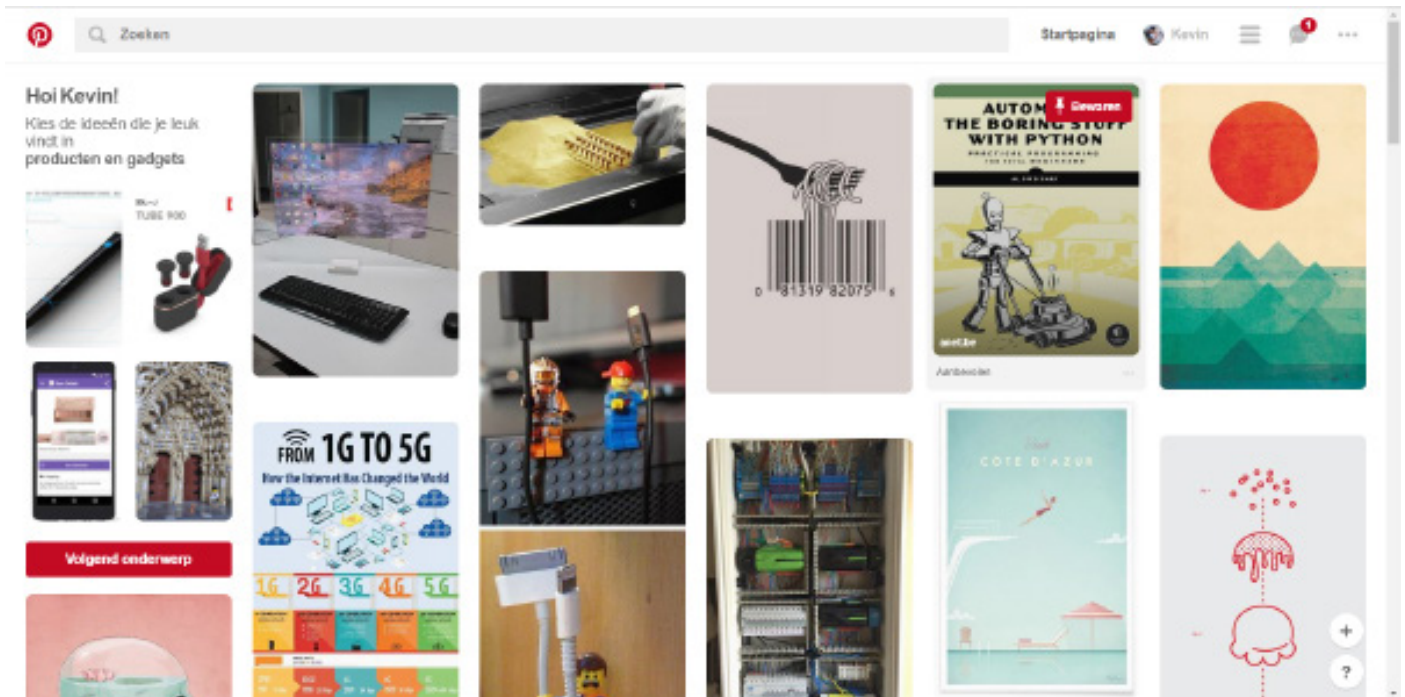
Goed:

- Mooie duidelijke structuur
- Eerste pagina weer met projecten die bij jouw groepen horen.
- Leuke navigatiebalk met pictogrammen
- Makkelijk te zien wie jou volgt en wie jijzelf volgt
- Al uw projecten mooi opgelijst

Minder goed:

- Pictogrammen van de navigatiebalk zijn niet super duidelijk

Pinterest (web-app)



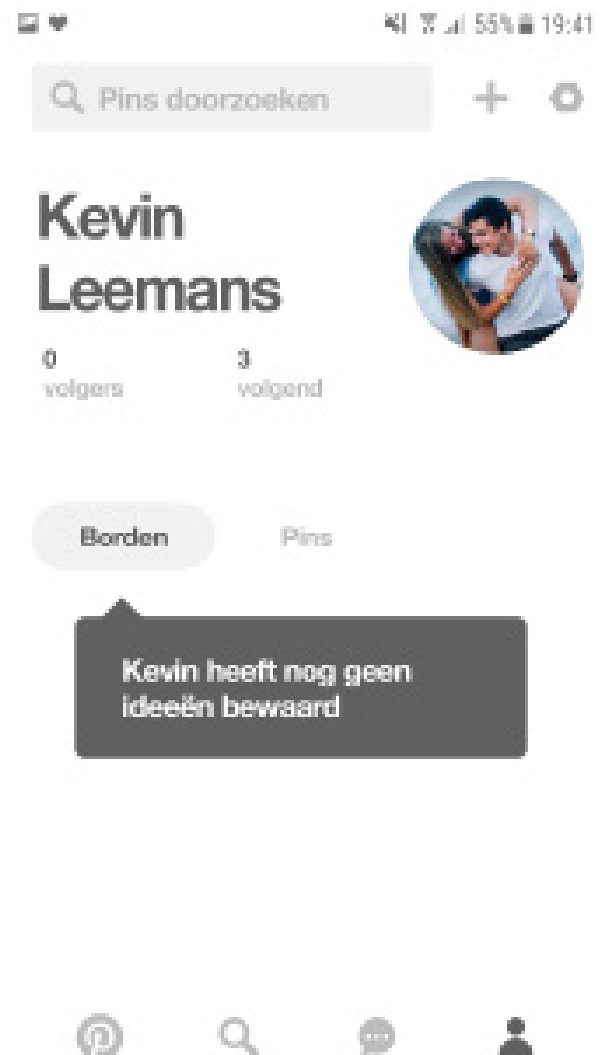
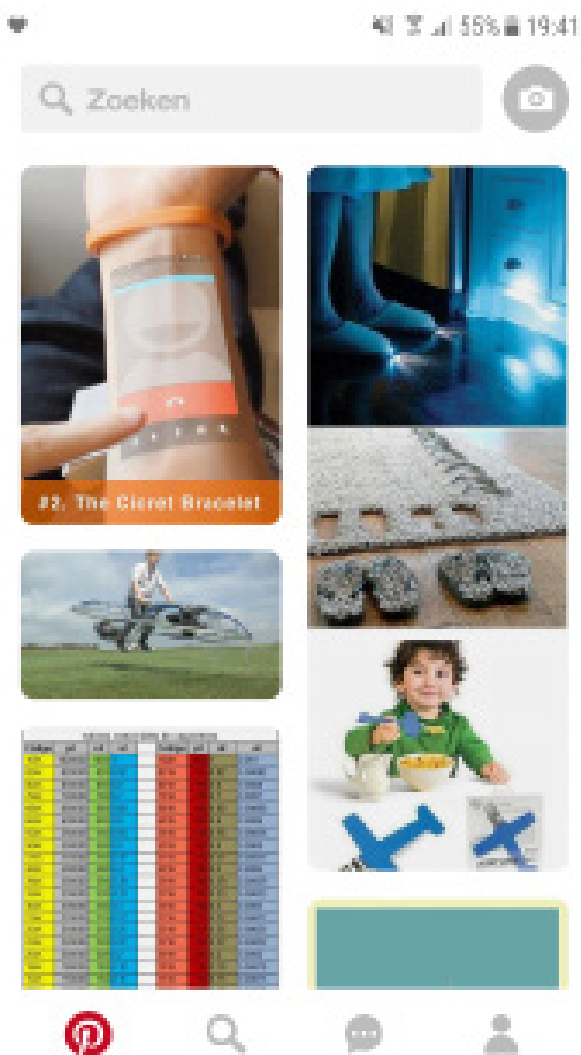
Goed:

- Meldingen die in de navigatiebar komen
- Duidelijke pictogrammen

Minder goed:

- Misschien wat grote searchbar
- Knop voor bestanden toe te voegen mag iets opvallender

Pinterest (mobile-app)



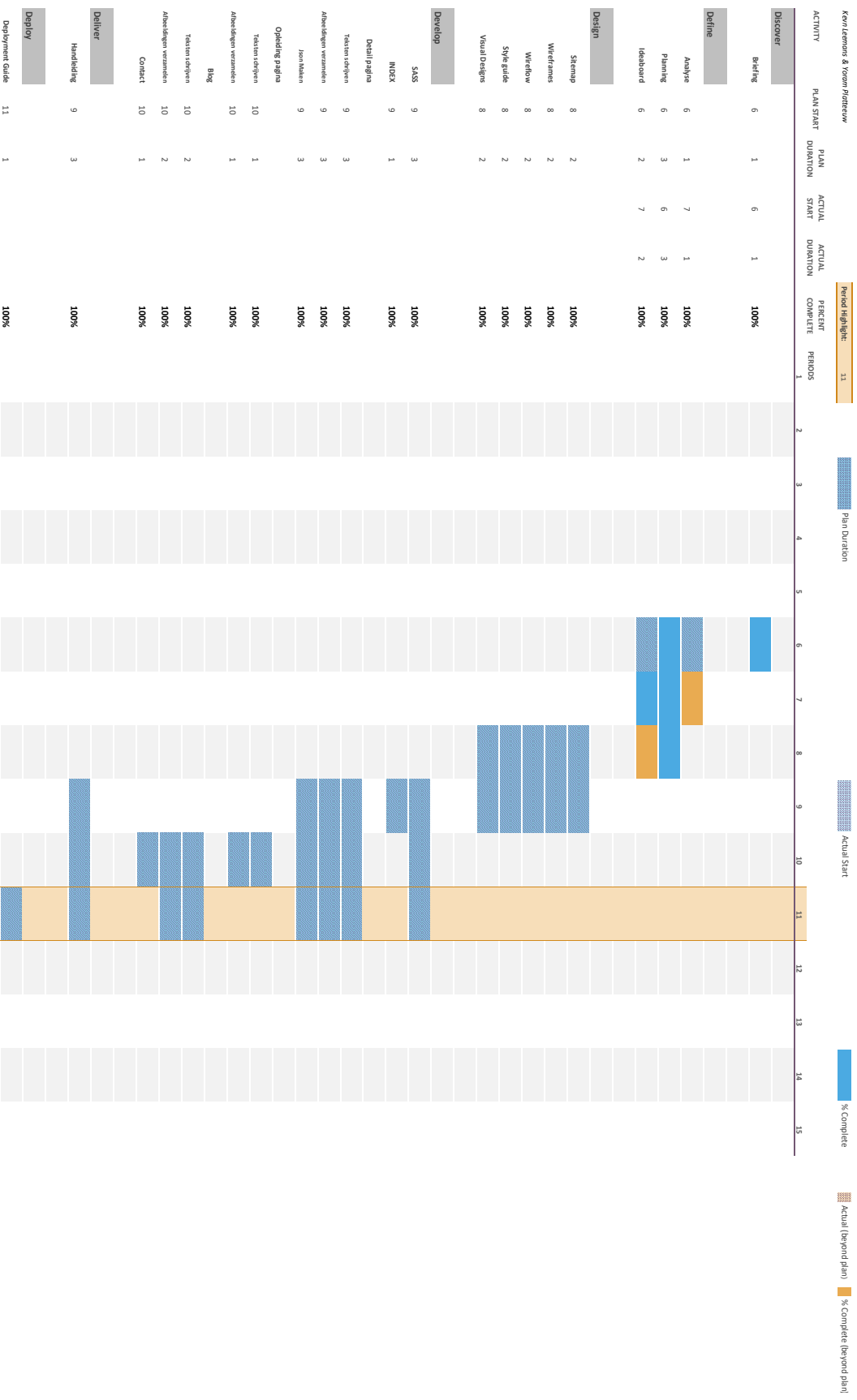
Goed:

- Mooie design, user friendly
- Navigatie bar heeft duidelijke pictogrammen
- Makkelijk te zoeken naar pins

Minder goed:

- /

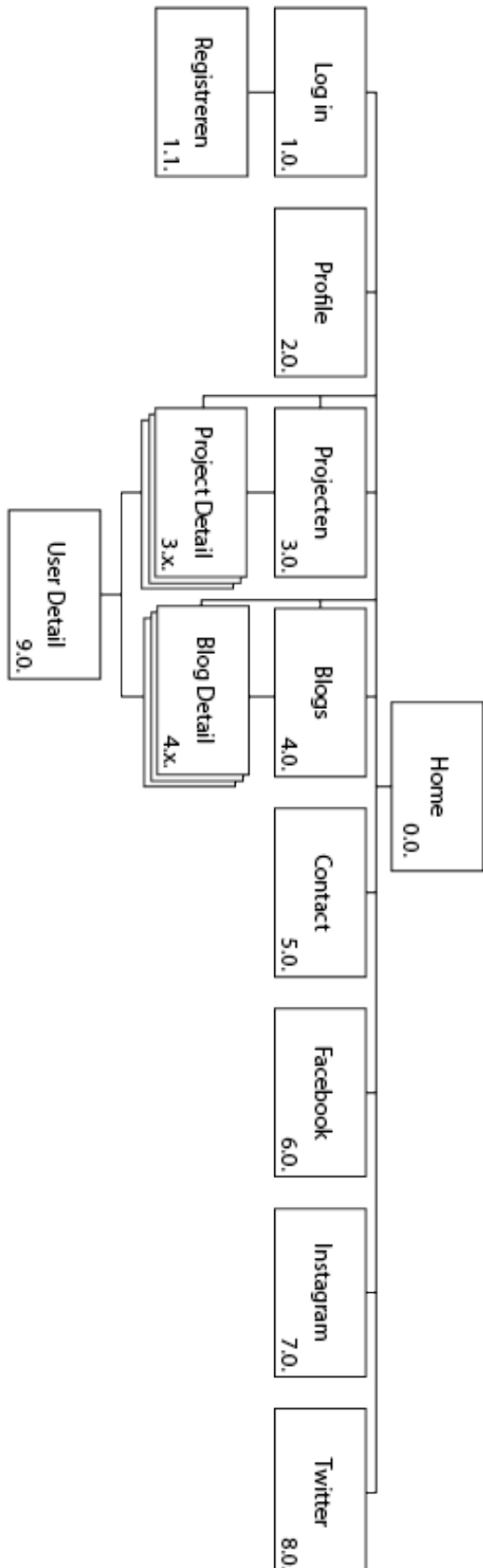
Werkstuk Mobdev 1



The collage features a variety of digital design elements. At the top left, there's a search bar with a magnifying glass icon. Below it, a sign-in form with fields for email and password, and a 'SIGN IN' button. To the right, a social media profile for 'Kate Winslet' with a cartoon avatar, a bell icon, and a clock icon. Below the profile, there's a 'Theme' section with a 'Share this APP' button and a 'Night mode' toggle. To the right of the profile, a color palette with various shades of blue, green, and yellow. Below the palette, a stylized logo for 'stylus' in a cursive font. To the right of the logo, a login form with fields for 'your@yours.com' and 'Password', and buttons for 'LOGIN' and 'CREATE AN ACCOUNT'. Below the login form, a portfolio grid with various images and text. At the bottom right, a social media profile for 'Jody Wisternoff' with a circular avatar, a 'FOLLOWING' button, and a 'MESSAGE' button. The entire collage is set against a light blue background with a subtle pattern of dots.

3. Design

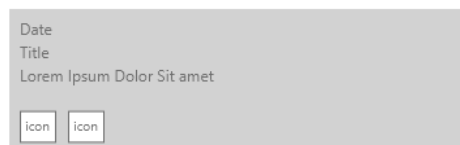
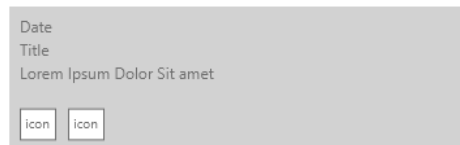
1. Sitemap



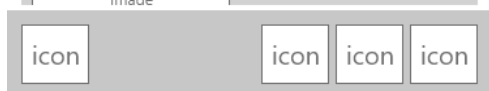
2. Wireframes



Blog



Projects



Log in

A login form with a light gray background. It features two input fields: 'Email' and 'Password'. Below the fields are two dark gray buttons labeled 'Log in' and 'Registrieren'.

Registrieren

A registration form with a light gray background. It features six input fields: 'NAME', 'FIRSTNAME', 'EMAIL', 'STUDENT NUMBER', 'DAY OF BIRTH (DD/MM/YYYY)', 'PASSWORD', and 'REPEAT PASS'. Below the fields is a dark gray button labeled 'Registrieren'.



Projects

Title

Image

icon

icon

Title

Image

icon

icon

Title

Image

icon

icon



Title

Description

Image

Image

icon

icon

icon

Author

>



Blog

Date

Title

Lorem Ipsum Dolor Sit amet

icon

icon

Date

Title

Lorem Ipsum Dolor Sit amet

icon

icon

Date

Title

Lorem Ipsum Dolor Sit amet

icon

icon

Date

Title

Lorem Ipsum Dolor Sit amet

icon

icon



Title

Lorem Ipsum Dolor Sit amet

icon

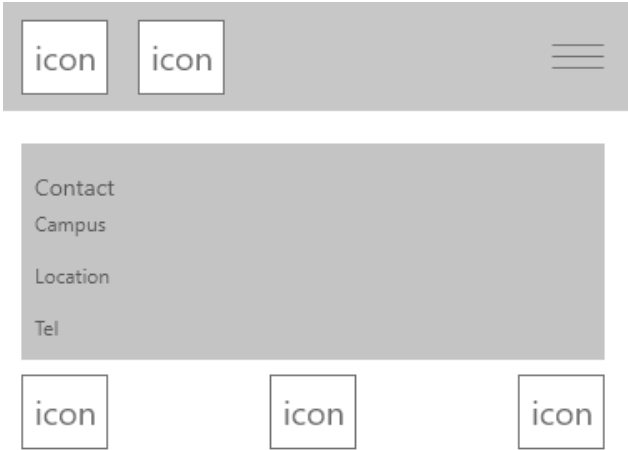
icon

icon

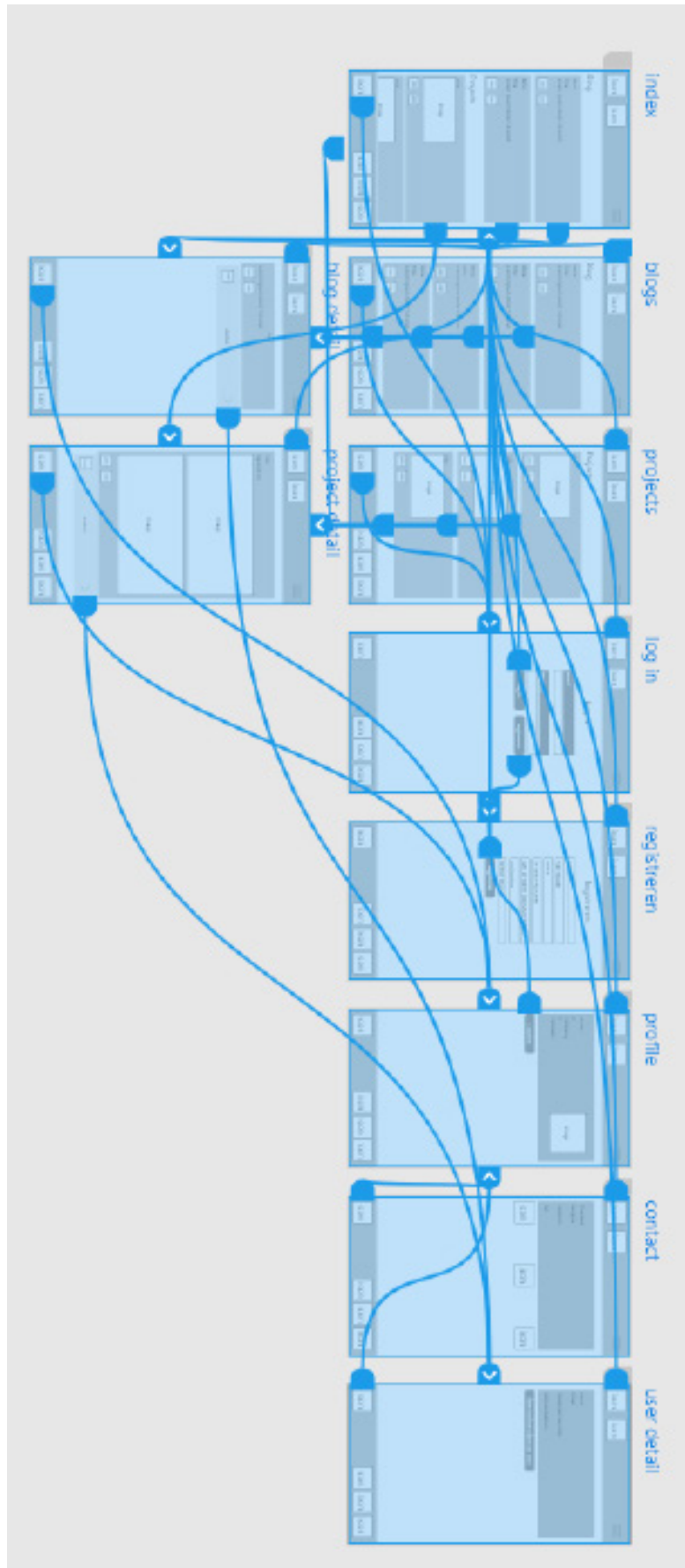
Author

>

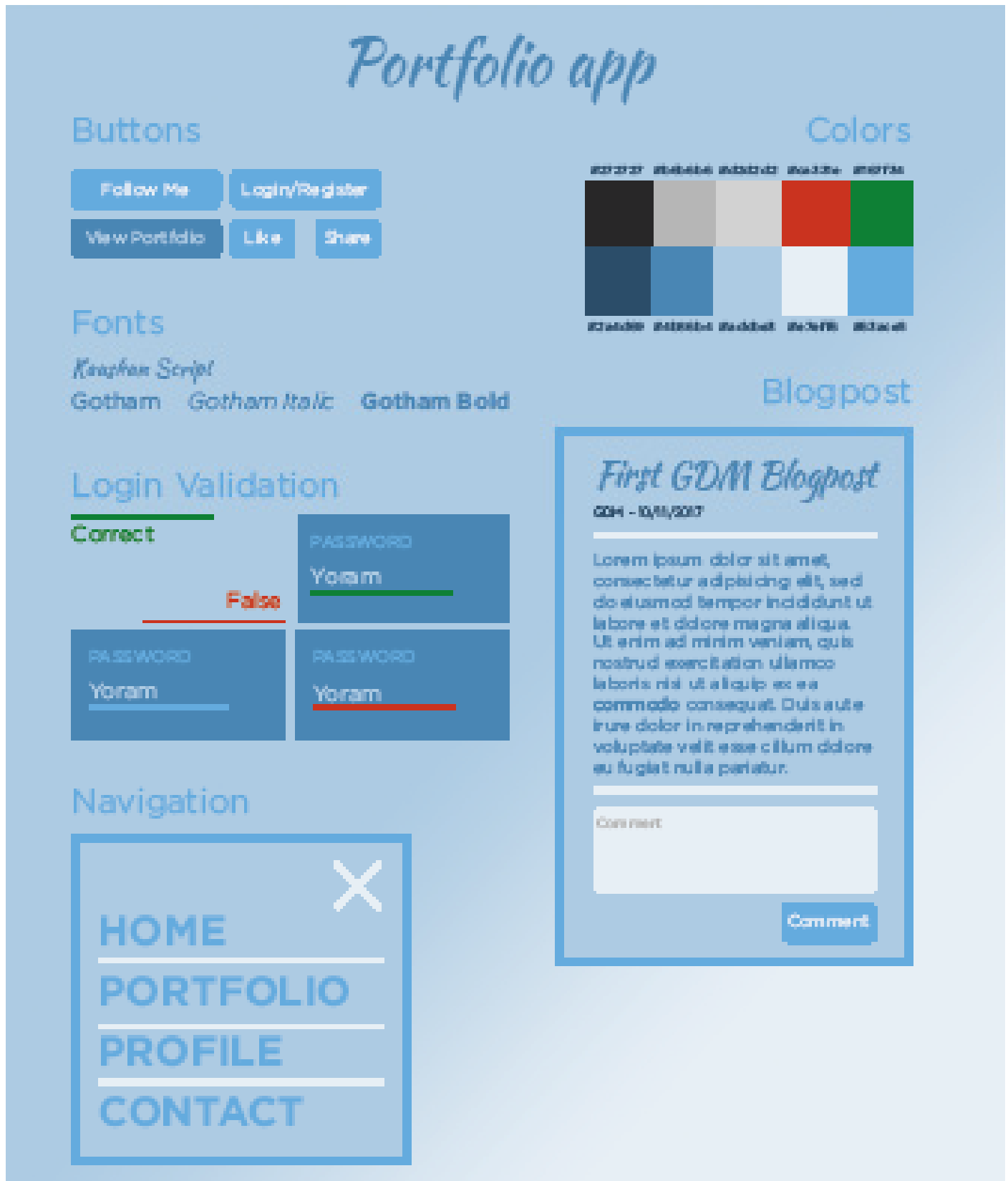




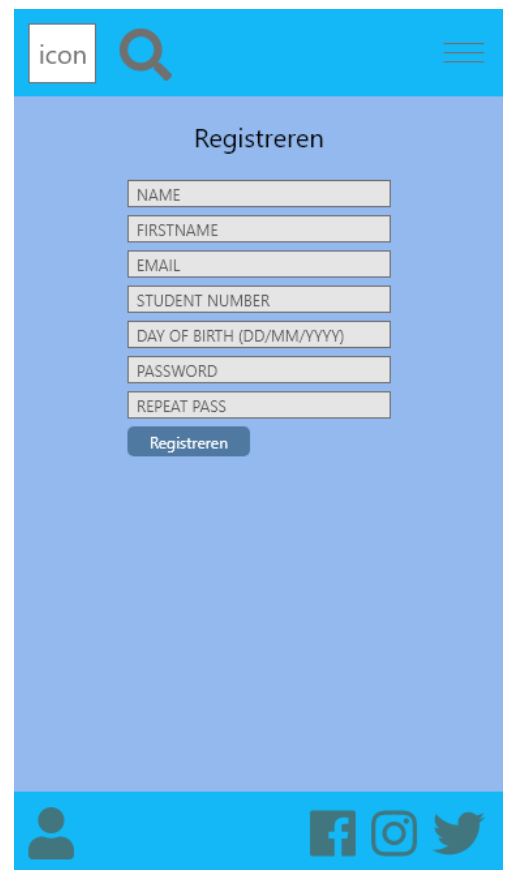
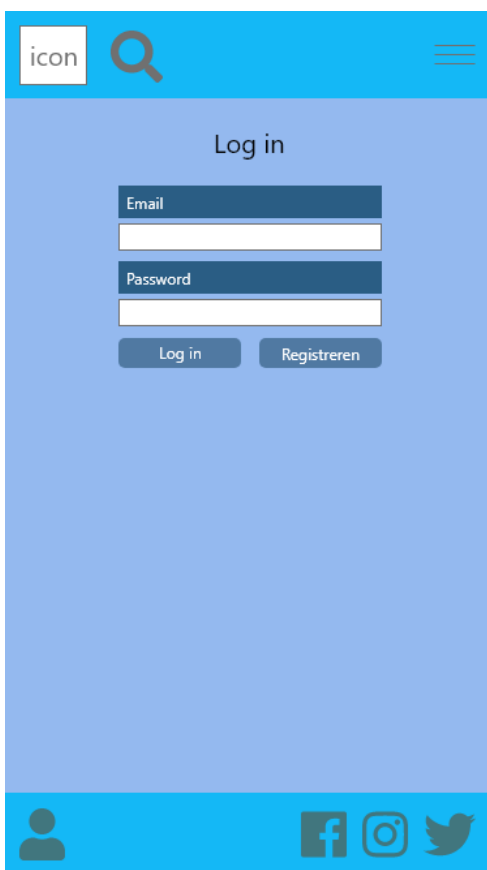
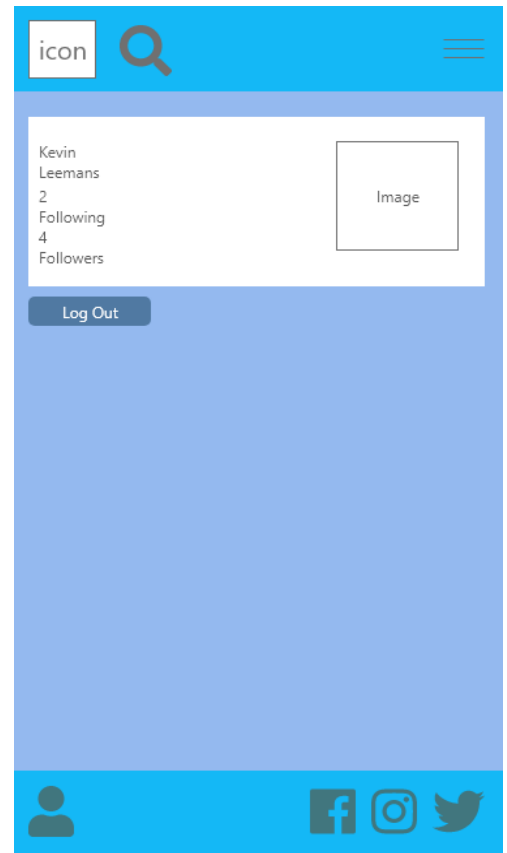
3. Wireflow

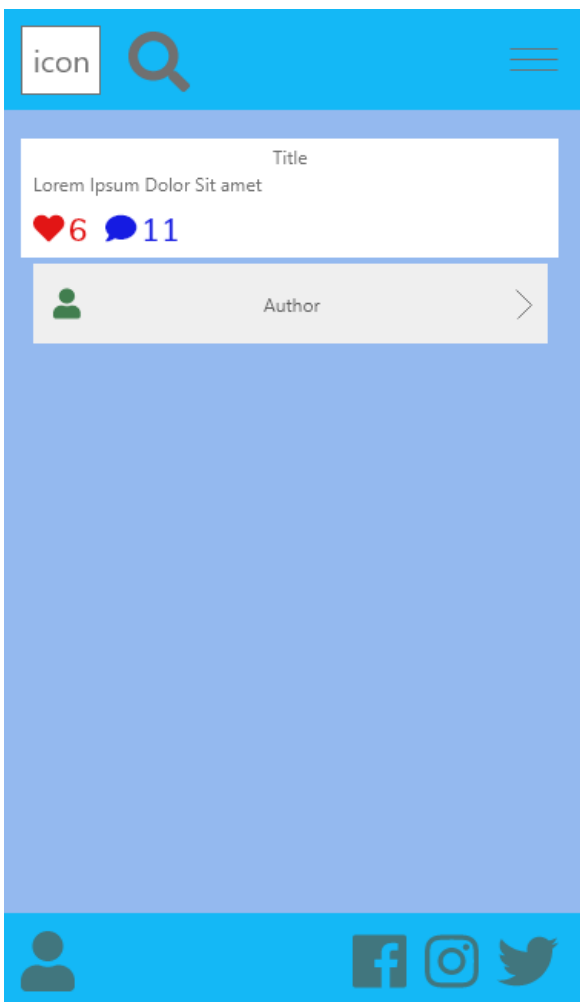
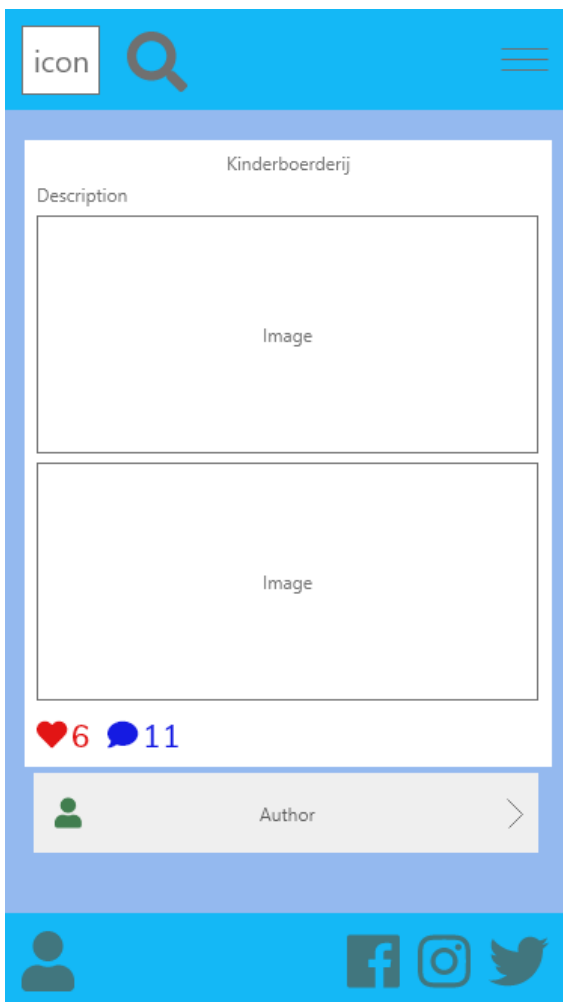
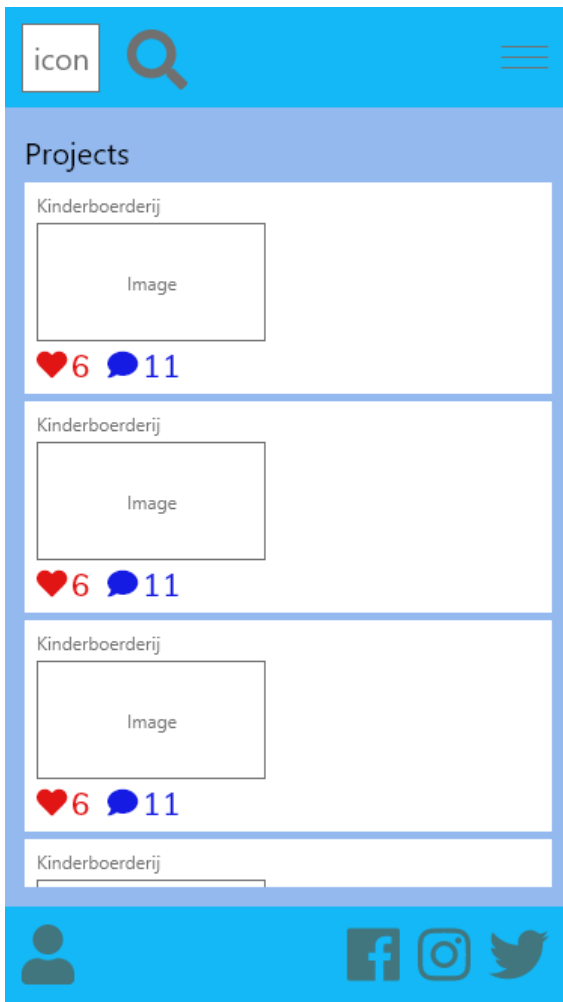


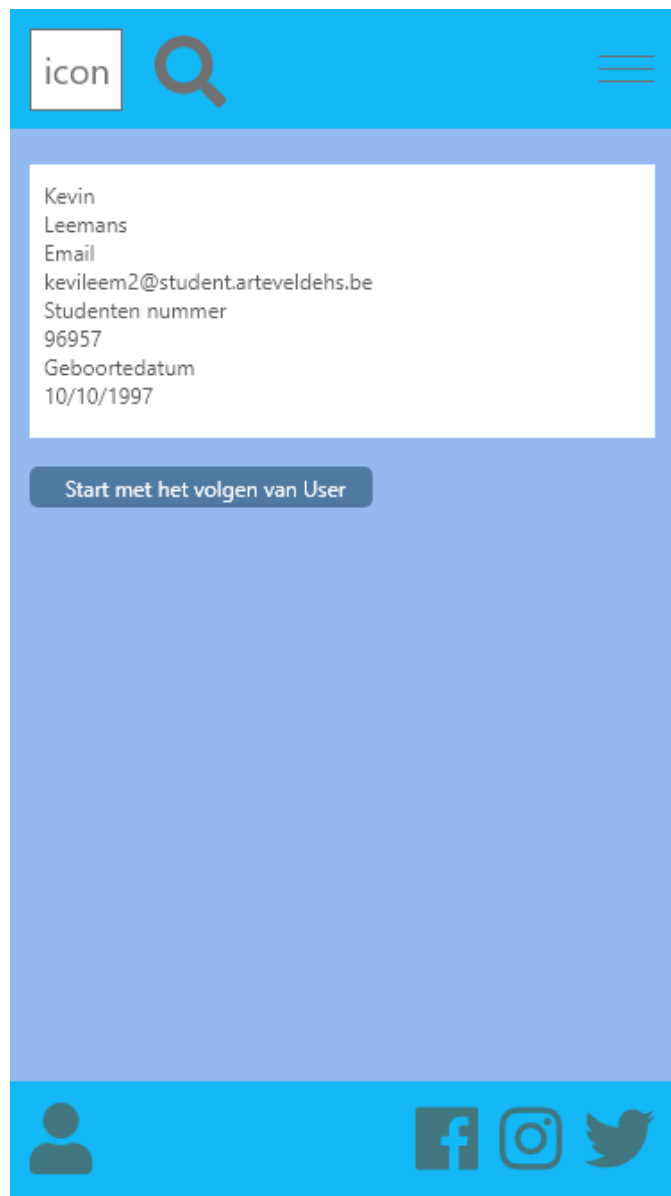
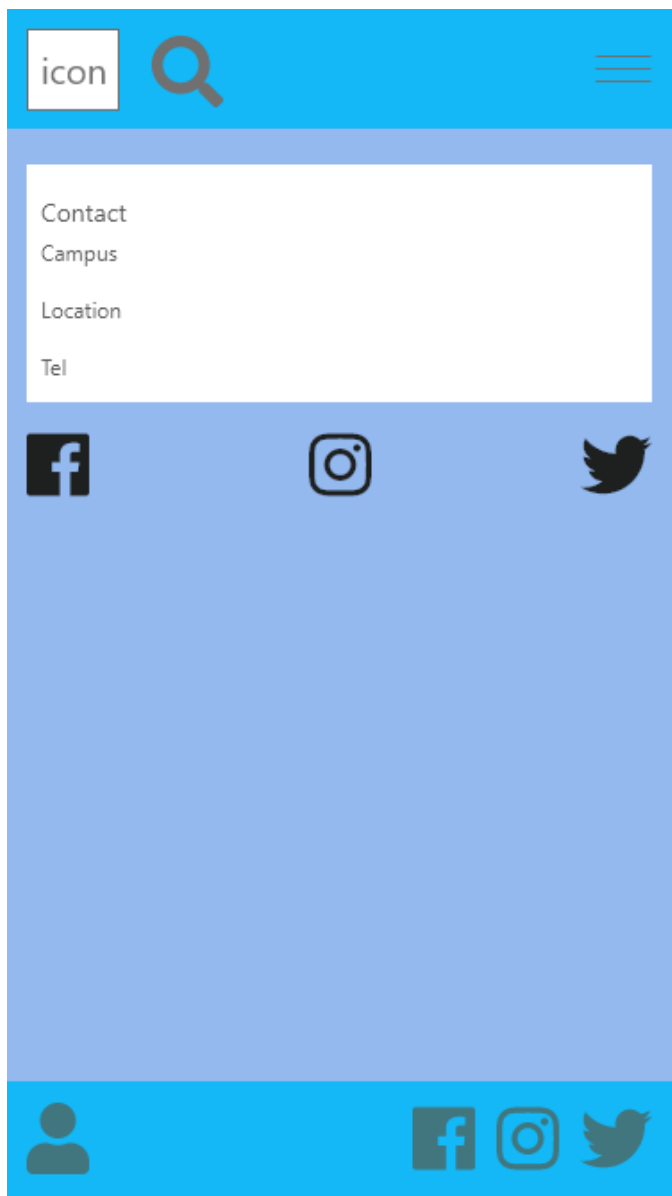
4. Style Tile



5. Visual Designs







4. Develop

1. Code

```
/*
load in the projects from the firebase database with parameters:
{numprojects} which is the amount of projects want to be shown, -1 is all of the projects
{col} for the amount of space you want the projects to hold
*/
loadProject (numprojects = -1, col = 6) {
  let query = this.database.ref('/projects');
  if (numprojects > 0) {
    query = this.database.ref('/projects').limitToFirst(numprojects);
  }
  this.renderProject(query, 'child_added', col);
}

/*
load in the posts from the firebase database with parameters:
{numposts} which is the amount of posts want to be shown, -1 is all of the posts
{col} for the amount of space you want the posts to hold
*/
loadPost (numPosts = -1, col = 6) {
  let query = this.database.ref('/posts');
  if (numPosts > 0) {
    query = this.database.ref('/posts').limitToFirst(numPosts);
  }
  this.renderPost(query, 'child_added', col);
}
```

```
/*
The full rendering of the login page
*/
loginContent () {
  const loginButton = document.querySelector('.loginBtn');
  loginButton.addEventListener('click', (ev) => {
    let email = document.querySelector('#loginEmail').value;
    let password = document.querySelector('#loginPassword').value;
    this.auth.signInWithEmailAndPassword(email, password).catch(error => {
      const errorCode = error.code;
      const errorMessage = error.message;
      console.log(errorCode);
      console.log(errorMessage);
    })
    .then(user => {
      if (user) {
        window.location.pathname = '/index.html';
      }
    });
  });
}
```

```

/*
The full rendering of the profile page
*/
profileContent () {
  const firstname = document.querySelector('.firstname');
  const surname = document.querySelector('.lastname');
  const following = document.querySelector('.following-amount');
  const followers = document.querySelector('.followers-amount');
  const logOutBtn = document.querySelector('.logOutBtn');
  logOutBtn.addEventListener('click', (ev) => {
    console.log('lol');
    this.auth.signOut();
    window.location.pathname = '/login.html';
  });
  this.auth.onAuthStateChanged(firebaseUser => {
    if (firebaseUser) {
      let query = this.database.ref(`/users/${firebaseUser.uid}`);
      query.on('value', (snap) => {
        const user = snap.val();
        const userfn = user.firstname;
        const usern = user.name;
        const userFollowing = user.following;
        const userFollowers = user.followers;
        firstname.innerHTML = userfn;
        surname.innerHTML = usern;
        following.innerHTML = userFollowing;
        followers.innerHTML = userFollowers;
      });
    }
  });
}

// routing for the app
if (pathname === '/' || pathname === '/index.html') {
  this.indexContent();
}
if (pathname === '/projects.html') {
  this.projectsContent();
}
if (pathname === '/blog.html') {
  this.postsContent();
}
if (pathname === '/profile.html') {
  this.profileContent();
}
if (pathname === '/login.html') {
  this.loginContent();
}
if (pathname === '/register.html') {
  this.registerContent();
}
if (pathname === '/contact.html') {
  this.contactContent();
}
if (pathname === '/post.html') {
  this.postDetailPage();
}
if (pathname === '/list.html') {
  this.listContent();
}

```

```
// Initialize the app with the firebase connection
constructor () {
  this.dbConfig = {
    apiKey: 'AIzaSyCIiUvrS0Pk8-przW0cGsw0C0K3nAPbXk',
    authDomain: 'mobdev1-aa6fd.firebaseio.com',
    databaseURL: 'https://mobdev1-aa6fd.firebaseio.com',
    projectId: 'mobdev1-aa6fd',
    storageBucket: 'mobdev1-aa6fd.appspot.com',
    messagingSenderId: '789886567615'
  };
  // Use firebase database and authentication
  firebase.initializeApp(this.dbConfig);
  this.database = firebase.database();
  this.auth = firebase.auth();
}
// The full initialization of the app
init () {
  // global variable for the pathname of the url
  const pathname = window.location.pathname;

  // watch if the user is logged in or not
  this.auth.onAuthStateChanged(firebaseUser => {
    // user is logged in
    if (firebaseUser) {
      console.log(firebaseUser);
    } else {
      console.log('Not logged in.');
```

```
});

/*
get the user from the database by using the id
*/
getUserByNameById (id, firstname) {
  this.database.ref('/users/' + id).once('value', (snap) => {
    |   firstname(snap.val());
  });
}
/*
follow a user by the id
*/
followById (id, name, el) {
  this.auth.onAuthStateChanged(user => {
    if (user) {
      const currentUid = user.uid;
      let updates = {};
      updates[`/followers/${id}/${currentUid}`] = true;
      updates[`/following/${currentUid}/${id}`] = true;
      this.database.ref().update(updates);
      el.innerText = `Je volgt ${name}`;
      return true;
    } else {
      el.innerText = `Je moet je aanmelden om ${name} te volgen`;
      return false;
    }
  });
}
}
```

```

/*
update the likes from a post or project by the id
*/
updateLikesById (type, id, el) {
  const currentUser = this.auth.currentUser;
  if (currentUser) {
    const uid = currentUser.uid;
    this.database.ref(`${type}/${id}`).transaction(post => {
      if (post) {
        if (post.likes && post.likes[uid]) {
          post.likesCount--;
          post.likes[uid] = null;
          el.innerText = '';
        } else {
          post.likesCount++;
          if (!post.likes) {
            post.likes = {};
          }
          post.likes[uid] = true;
          el.innerText = 'Je hebt dit geliked';
        }
      }
    });
    return post;
  } else {
    console.log('Log in to like posts');
  }
}
}

```