



# Projectenwebsite

Arijn Borzo

Andreas Lauwers

Lukas Petit

Arno Stas

**Projecten 1**

**Technische mentor: Pieter Van Peteghem**

**Taalmentor: Sabine Martens**

**Academiejaar 2018-2019**

**Professionele Bachelor ICT**

## Abstract

Develop an application which facilitates creation of projects. Upon account creation users must fill in the Belbin test to obtain their team role. Great teams can be put together, existing teams can be understood and improved, and everyone can feel that they are making a difference in the workplace. At this point in time a similar application exists and is up and running but is very outdated. One important advantage of the system over the previous version is that each student can log into his account to do various tasks. Students, supervisors and project managers have different privileges. Managers can create a project and if it is accepted by a supervisor, users can be added to the group according to their Belbin role.

Keywords: application, Belbin, projects, teams

## Inhoud

Codefragmentenlijst .....	4
Figurenlijst .....	5
Afkortingenlijst .....	6
Begrippenlijst.....	7
Inleiding.....	8
1. Userexperience.....	9
1.1 Verslag interviews .....	9
1.2 Persona's .....	10
1.3 Storyboard.....	11
1.4 Wireframe .....	11
2 Laravel .....	12
2.1 Framework .....	12
2.2 Databanktype .....	12
2.3 Databankmodel .....	12
2.4 Migrations en seeders .....	14
2.5 Eloquent modellen .....	16
2.6 Controllers .....	17
2.7 Blade.....	18
2.8 Vue in Laravel .....	18
2.9 Routes.....	19
2.10 Authenticatie.....	19
2.11 Conclusie .....	19
3. BootstrapVue.....	20
3.1 Framework Vue .....	20
3.2 Componenten.....	20
3.3 BootstrapVue.....	21
3.4 HTML .....	21
3.5 JavaScript.....	22
3.6 CSS .....	23
3.7 Besluit.....	23
4. Webtoepassing hosten.....	24
4.1 Azure portal.....	24
4.2 Azure App service .....	24

---

4.3	Web-app .....	24
4.4	Implementatiesites.....	25
4.5	Ondersteuning continue integratie en implementatie .....	26
4.6	Samenvatting.....	26
5.	Functionaliteiten .....	27
5.2.	Project toevoegen .....	27
5.3.	Goedkeuren project .....	27
5.4.	Klassensysteem .....	27
6.	Planning .....	28
	Conclusie .....	29
	Overzicht bijeenkomsten .....	30
	Literatuurlijst .....	31
	Bijlagenoverzicht .....	32
	Bijlage 1: E-mail meneer Sanders.....	33
	Bijlage 2: Interview mevrouw Martens.....	35
	Bijlage 3: Enquête van studenten.....	36
	Bijlage 4: Storyboard .....	40
	Bijlage 5: Persona's.....	41
	Bijlage 6: Wireframes .....	42

## Codefragmentenlijst

Codefragment 1	PHP artisan make commando .....	12
Codefragment 2	Uitvoeren van migrating en seeding .....	14
Codefragment 3	Migrationbestand gebruikers .....	15
Codefragment 4	Seederbestand gebruikers.....	16
Codefragment 5	Eloquent model student.....	16
Codefragment 6	Controller van overzichten .....	17
Codefragment 7	Nazien of een gebruiker ingelogd is m.b.v. Bladeparameters .....	18
Codefragment 8	Gebruik van Vue in Laravel.....	18
Codefragment 9	Routes.....	18
Codefragment 10	Voorbeeld van Vue .....	20
Codefragment 11	JavaScript in Vue.....	20
Codefragment 12	Card in Vue .....	21
Codefragment 13	Grid- en lay-out systeem in Vue .....	22
Codefragment 14	JavaScript methoden in Vue.....	22
Codefragment 15	CSS in Vue .....	23

## Figurenlijst

Figuur 1	Screenshot oude projectenwebsite .....	9
Figuur 2	Databankmodel projectenwebsite.....	13
Figuur 3	Card in Vue .....	21
Figuur 4	Grid- en lay-outsysteem .....	22
Figuur 5	Instellingen in Azure .....	25
Figuur 6	Voorbeeld van Implementatiesite. ....	25
Figuur 7	Resultaten enquête: Categorie ondervraagde persoon.....	37
Figuur 8	Resultaten enquête: Hebt u al gebruik gemaakt van de projectenwebsite?.....	37
Figuur 9	Resultaten enquête: Hoe hebt u uw projectgroep gevonden? .....	37
Figuur 10	Resultaten enquête: Hoe was uw ervaring? .....	38
Figuur 11	Resultaten enquête: Wat vindt u van de grafische opmaak? .....	38
Figuur 12	Resultaten enquête: Moeten projecten gecategoriseerd kunnen worden? .....	38
Figuur 13	Resultaten enquête: Hoe moet de groepsvorming gebeuren? .....	39
Figuur 14	Resultaten enquête: Overige opmerkingen.....	39
Figuur 15	Resultaten enquête: Wat zijn uw grootste frustraties met de site?.....	39
Figuur 16	Resultaten enquête: Welke functionaliteiten ontbreken? .....	40
Figuur 17	Resultaten enquête: Waarom gebruikt u het platform? .....	40
Figuur 18	Storyboard: De eerste login van student? .....	41
Figuur 19	Persona docent .....	42
Figuur 20	Persona student .....	43
Figuur 21	Wireframe overzichtspagina .....	44
Figuur 22	Wireframe archiefpagina .....	45
Figuur 23	Wireframe nieuw projectvoorstel.....	46
Figuur 24	Wireframe detailpagina gewone gebruiker .....	47
Figuur 25	Wireframe detailpagina projectleider.....	47
Figuur 26	Wireframe Belbin test .....	48

## Afkortingenlijst

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
MySQL	Structured Query Language
IDE	Integrated Development Environment
CLI	Command Line Interface
PaaS	Platform as a Service

## Begrippenlijst

Backend	Het achterliggende gedeelte achter de website (databank, functionaliteit,...).
Inserts	Commando waarmee gegevens in een tabel van een databank worden toegevoegd.
Framework	Een softwarepakket waarin verschillende componenten automatisch geladen worden. Hierdoor moet de ontwikkelaar niet zelf verschillende pakketten aanmaken.
Frontend	Het gedeelte dat zichtbaar is voor de gebruiker (layout, design).
Responsive	De website aanpassen zodat het voor elk scherm (groot scherm, laptop, tablet, smartphone) aangenaam is om te bekijken en te gebruiken.
Stickfigures	Personen getekend met lijnen.
Xtensio	Online gratis opensource platform waar men persona's kan maken.
Userexperience	Design van de website richten naar de specifieke doelgroep na onderzoek



## Inleiding

In dit project wordt een website ontwikkeld dat als doel heeft een nieuw platform op te zetten voor het vak *Project* dat gegeven wordt in het tweede jaar Elektronica-ICT aan de hogeschool Odisee. Het platform is noodzakelijk zodat de groepen makkelijker gevormd kunnen worden en projecten zonder problemen toegevoegd, bewerkt, verwerkt en geaccepteerd worden. Hierbij wordt de huidige procedure in vraag gesteld.

Momenteel is de website werkende, maar enorm gedateerd. Het is tot op heden mogelijk om als student en docent in te loggen. Qua functionaliteit voegen docenten projecten aan de database toe en keuren voorstellen goed. Studenten kunnen enkel een projectvoorstel indienen en zich hiervoor inschrijven. Bovendien hebben ze geen toegang tot afgelopen projecten om inspiratie op te doen. De efficiëntie is dus heel beperkt en de gebruikte resources zijn minimaal. Dit is de reden waarom alles grondig herwerkt wordt.

De website wordt dus volledig omgebouwd tot een platform waar de volledige groepsvorming kan plaatsvinden mede door bij nieuwe gebruikers een Belbin test uit te voeren. Op deze manier kan er een waarschuwing gegeven worden als er studenten zijn die niet bij elkaar passen. De tweede hoofdfunctionaliteit is het opstellen van nieuwe projecten samen met het goedkeuren van de (geselecteerde) docent.

Om deze website te maken zonder code van de bestaande website te gebruiken, wordt het volledige bouwproces van een nieuwe website doorlopen. Dit wil zeggen dat userexperience gebruikt wordt, waarna het verkregen design ontworpen zal worden. Dit gebeurt op basis van een specifiek bedacht databasemodel dat geïntegreerd wordt in het Laravel framework, het belangrijkste gedeelte. Daarna wordt het grafische gedeelte toegevoegd op basis van CSS en JavaScript. Dit wordt praktisch gerealiseerd door Bootstrapklassen en Vue-componenten.

Indien de nieuwe versie goedkeuring krijgt, kan deze worden gehost op hetzelfde adres van de huidige website. Op deze manier blijven de resources dezelfde.

In dit rapport worden de verschillende stappen bij het maken van een nieuwe website onderzocht. Zo worden de fases van userexperience uitgelegd in het eerste hoofdstuk. Vervolgens wordt er bekeken of er best gewerkt wordt met een framework (backend). Ook worden belangrijke delen van de HTML en CSS uitvergroot en uitgelegd. Ten slotte worden de technische uitwerking en functionaliteiten van de website overlopen.

## 1. Userexperience

Bij het verwerken van de huidige projectenwebsite is er terug vanaf nul begonnen. Dit wil zeggen dat er eerst onderzocht wordt wat de verlangens zijn van alle potentiële gebruikers. Vervolgens worden de twee belangrijkste persona's opgesteld. Dit zijn de stereotype gebruikers van de toekomstige website. Deze visualiseren een meer algemeen beeld van zowel student als docent.

Het storyboard toont op een zeer simpele manier de primaire werking van de website.

Als laatste worden de wireframes gemaakt. Dit zijn de schetsen die de frontenddesigners krijgen zodat ze weten hoe de lay-out eruit moet zien. Tegelijkertijd geeft het de uitvoerende programmeur nog vrijheid om kleine zaken te wijzigen en het grafisch nog mooier te maken.

### 1.1 Verslag interviews

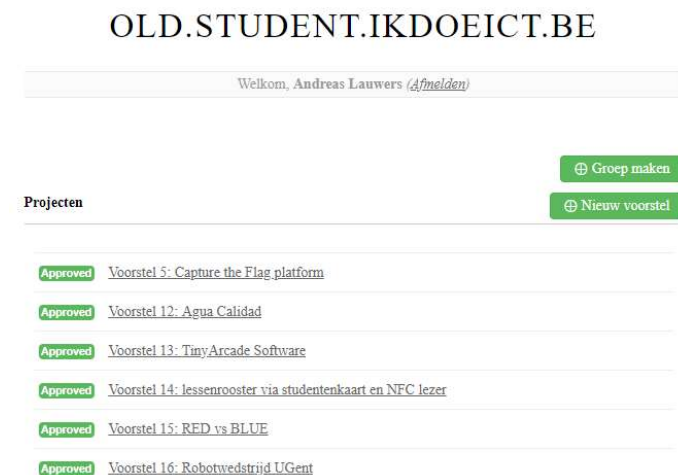
Bij het opstellen van de vragenlijst wordt onderscheid gemaakt tussen student en docent. Dit komt omdat de werking voor beide groepen verschillend is. Bij de docenten wordt er meer specifiek gevraagd naar de toekomst van de website, terwijl bij studenten meer wordt gekeken naar de problemen bij de huidige site.

Bij de docenten worden mevrouw Martens en meneer Sander geïnterviewd. Beide docenten zijn verantwoordelijk voor het vak Project + Wetenschappelijk rapporteren. Hierdoor zijn ze cruciaal voor de ontwikkeling van de site omdat zij er de komende jaren het meest gebruik van maken.

Omdat meneer Sanders niet onmiddellijk een afspraak kon maken, haalt hij de belangrijkste werkpunten aan via mail (Bijlage 1). Op deze manier loopt het maken van de site geen vertraging op. Uit deze mail kunnen al direct een aantal kernideeën worden gehaald. Deze ideeën vormen dan de basis voor een recentere en meer specifieke vragenlijst gemaakt voor mevrouw Martens (Bijlage 2). Hierdoor worden bepaalde zaken bevestigd en andere tegengesproken. Zo stelt meneer Sanders

voor om het ook mogelijk te maken om mededelingen te kunnen plaatsen op de site (net zoals bij Toledo het geval is). Daarnaast komen er ook nieuwe visies van beide docenten die zorgen voor vernieuwende ideeën die nog niet aanwezig zijn in de huidige website. Zo wordt er door mevrouw Martens nagedacht over een rechtstreekse implementatie van de Belbin test. Meneer Sanders hecht vooral belang aan de communicatie tussen docent en student. Zo komt hij met een idee om het de student mogelijk te maken een docent te selecteren waarmee hij zijn project reeds besproken heeft.

De aandachtspunten waar beide docenten het mee eens zijn, zijn zeker de zaken die het belangrijkste zijn. Zo valt bij beide docenten op dat de website te traag is en te ouderwets oogt voor hedendaagse normen (figuur 1). De overige knelpunten waar beide docenten een uiteenlopende mening hebben, zijn inspiratie voor de vragenlijst van de studenten (Bijlage 3).



Figuur 1: Screenshot oude projectenwebsite

Voor de vragenlijst van de studenten wordt met behulp van Google Docs een eenvoudig formulier opgesteld. Dit zorgt ervoor dat via sociale media en e-mail de enquête snel naar een groter publiek kan worden verstuurd. Door de interface van Google Docs, is het eenvoudig om een procentuele grafische voorstelling te krijgen. Deze resultaten zijn zeer handig voor het maken van de persona van de student.

## 1.2 Persona's

### 1.2.1 Docent

Uit de antwoorden van de vragen kan een globaal karakter van zowel student als docent opgesteld worden. Dit gebeurt door telkens te kijken welke mening de meeste personen van dezelfde categorie delen. Met behulp van Xtensio is het mogelijk om een voorstelling te maken van de stereotype student en docent. Door te kiezen van een template kan vlot te werk gegaan worden en kunnen de persona's op professionele manier ontworpen worden.

Na de interviews van meneer Sanders en mevrouw Martens is er een persona gemaakt van een fictieve docent (Bijlage 3A). Deze docent werd voorgesteld als een combinatie van de mening van de twee ondervraagde personen. De grootste prioriteiten waar de website volgens hem aan moet voldoen zijn de volgende:

- de groepsvorming moet eenvoudig en gestructureerd kunnen gebeuren;
- er moet een overzicht zijn van welke projecten goedgekeurd, afgekeurd of in afwachting zijn;
- de site moet een propere en overzichtelijke look hebben;
- het moet mogelijk zijn een project te bewerken indien dit nodig is. Dit zorgt ervoor dat een concept kan opgeslagen worden om er later aan verder te werken. Zo kan een student zijn eigen project later terug ophalen om extra gegevens hieraan toe te voegen. Indien een project afgekeurd wordt door een docent, kan het project worden bijgewerkt om toch goedgekeurd te worden.

### 1.2.2 Student

Net als bij de docent wordt ook voor de student een persona ontworpen. Omdat beide persona's voor dezelfde toepassing dienen, is er op gelet dat ze vergelijkbaar met elkaar zijn.

Met de antwoorden van de enquête van de studenten is de persona gemaakt van Arne. Hij stelt een student voor uit de richting Elektronica-ICT. Zijn focus ligt vooral op een productieve werking van de website. Hij wil op een efficiënte manier een afgewerkt product kunnen hebben tegen het eind van het semester. Bij de huidige website heeft hij een aantal frustraties die hij hoopt bij de nieuwe website verbeterd te zien, namelijk:

- er is geen makkelijke manier om te communiceren via de website;
- de website is traag, het duurt lang voor hij aangemeld is;
- er kan niet gefilterd worden op projecten, waardoor er ongeïnteresseerde studenten bij sommige projecten terecht komen.

### 1.3 Storyboard

Nadat de persona's opgesteld worden, kan er nagedacht worden over de acties die de site moet kunnen uitvoeren. De handelingen die kunnen uitgevoerd op de website worden later besproken. Deze worden omgevormd tot knoppen, links of andere doeleinden. Op de landingspagina zijn dit de acties die voor het bedrijf het belangrijkste zijn.

Ondertussen is het duidelijk wat de hoofddoelen zijn en één hiervan wordt geschetst in een storyboard. Dit is een kort en eenvoudig getekend stripverhaal dat het primaire doel weergeeft. Er is gekozen voor het aanmaken van nieuwe gebruiker die zich meteen ook aan een project wil toevoegen. De persoon moet zich aanmelden en voert onmiddellijk de Belbin test uit. Op het platform ziet hij dan verschillende onvolledige projectgroepen waar hij in kan behoren met hun al dan niet goedgekeurd projectvoorstel. De persoon kijkt wat rond en beslist voor een project dat hem het meeste aanstaat, communiceert met de groep en spreekt later af met zijn nieuwe groepsleden.

### 1.4 Wireframe

Als laatste stap van het userexperience-onderzoek, worden de wireframes opgemaakt. Dit zijn de eerste ruwe schetsen hoe de website eruit zal komen te zien. Bij deze primaire schetsen wordt nog geen rekening gehouden noch met kleur, noch afbeeldingen, noch design. Het is vooral belangrijk dat de lay-out van de website bepaald wordt. Foto's worden vervangen in een rechthoek met een kruis erdoor. Daarnaast worden alle pagina's in grijswaarden opgemaakt.

D.m.v. de persona's is de hoofdfunctionaliteiten van de website duidelijk. Samen met het theoretisch inzicht over gebruiksvriendelijkheid kan er nagedacht worden welke knop of actie waar geplaatst moet worden. Welke details meer in het oog moeten springen of welke absoluut niet mogen ontbreken om de gebruiker het zo gemakkelijk mogelijk te maken.

Het doel van de wireframes is het werk van de frontenddesigners te verlichten. Er is nu een sjabloon waarmee aan de slag kan gegaan worden. Tegelijkertijd bieden deze wireframes nog mogelijkheid tot kleine aanpassingen. Zo kunnen kleuren aangepast worden en kunnen bepaalde items licht van plaats veranderen om de algemene functionaliteit van de website te bevorderen.

Bij alle wireframes is alvast nagedacht over de onderdelen die overal gelijk blijven. Zo is de header en footer op alle pagina's gelijk. Daarnaast zijn op de overzichtspagina en archiefpagina al nagedacht over mogelijke filters en weergave mogelijkheden (lijst of rasterweergave). Dit zijn zaken die momenteel nog niet in de huidige site aanwezig zijn.

In de detailpagina van een project is vooral gefocust op het onderlinge verschil is studenten. Zo zullen studenten die nog niet in een groep zitten de keuze krijgen om zich kandidaat te stellen voor een welbepaald project. Personen die al in een project zitten kunnen anderzijds ook verzoeken van projectzoekende studenten accepteren of weigeren. Bovendien zal er een waarschuwing komen als meerdere mensen van hetzelfde Belbin type een groep proberen te vormen.

Bij de pagina om een nieuw project toe te voegen is vooral nagedacht over uitbreidingsmogelijkheden in vergelijking met de huidige handeling. Bij de vernieuwde versie zal het mogelijk zijn om direct de SMART-criteria in te vullen, alsook de hoofd- en nevenvragen. Zo kan de student al beginnen nadenken over deze gegevens als hij dit nog niet eerder gedaan had. Ook kan de student al een leerkracht aanduiden waarmee hij het project mogelijks al besproken heeft. Zo kan de

docent zelf projecten goedkeuren die hij met studenten besproken heeft. Dit zorgt voor een vlottere werking tussen de begeleider en de uitvoerende groep.

## 2 Laravel

Na onderzoek van de minimale vereisten van de te ontwerpen website, kan worden nagedacht over de structuur. Voor de huidige toepassing wordt gekozen om gebruik te maken van een framework.

[1] [2] [3] [4] [5] [6] [7] [8]

### 2.1 Framework

De projectenwebsite is gemaakt m.b.v. Laravel. Dit is een open source PHP-framework dat toelaat om eenvoudig met een databank te communiceren. Daarnaast biedt Laravel een hele hoop shortcuts aan om relatief snel een structurele website af te leveren. Zo is het mogelijk om volledige bestanden te genereren door één regel code in een terminal te typen (codefragment 1).

```
root@1746b1deacc4:/var/www# php artisan make:migration test_migration
Created Migration: 2019_05_02_160715_test_migration
```

*Codefragment 1: Aanmaken migratiebestand m.b.v. Artisan.*

Laravel is een van de populairste frameworks om mee te werken. Er is veel documentatie op het internet te vinden, wat het aanleren ervan vereenvoudigt.

### 2.2 Databanktype

Laravel biedt de mogelijkheid om met een databank te werken. Hierin zullen alle gegevens van studenten, docenten, projecten, ... opgenomen worden. Als een webpagina opgevraagd wordt, zullen de nuttige gegevens uit de databank gehaald worden om te implementeren op deze pagina. Laravel ondersteunt vier verschillende databanktypes:

- MySQL
- PostgreSQL
- SQLite
- SQL Server

Voor de projectenwebsite wordt gekozen om MySQL te gebruiken. Dit is het default databanktype van Laravel. MySQL wordt voornamelijk aangewend voor webtoepassingen. De opslag grootte van een volledig databankmodel is vrij klein. In het geval van de projectenwebsite past deze makkelijk op eenzelfde server. Indien dit niet het geval zou geweest zijn, moeten andere databanksystemen overwogen worden.

### 2.3 Databankmodel

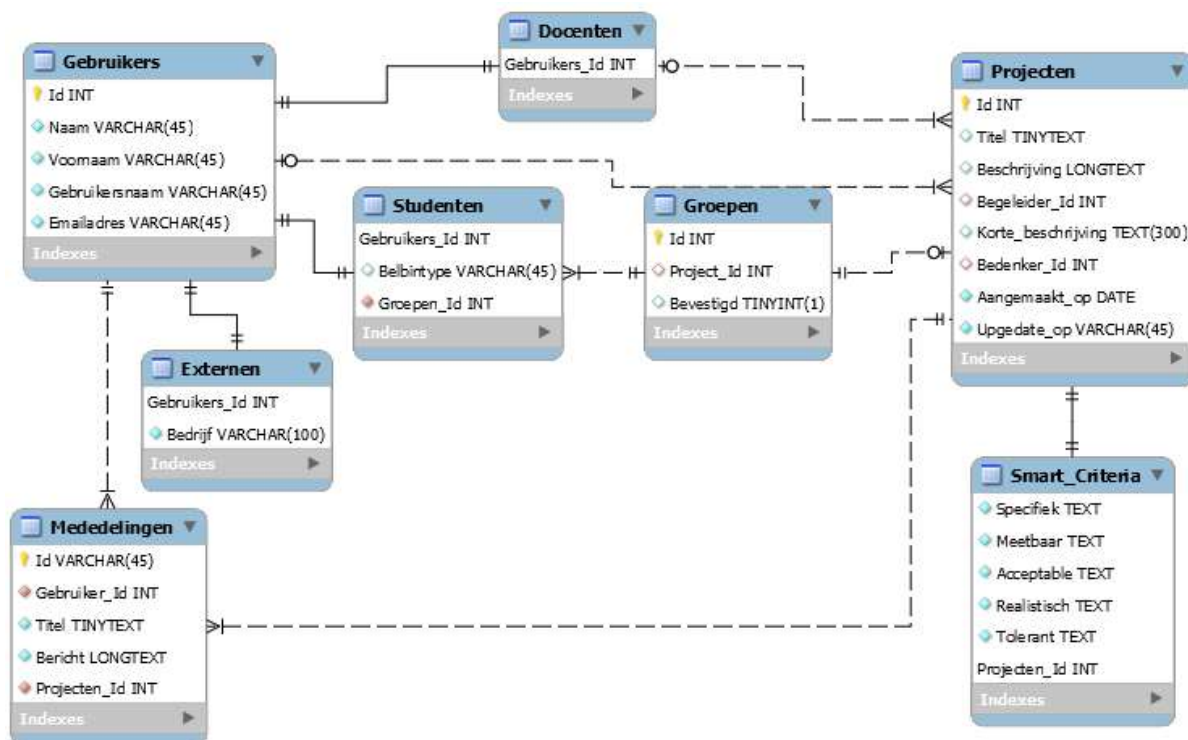
MySQL maakt het mogelijk om een databankmodel te creëren bestaande uit verschillende tabellen. Alle tabellen zijn onderling aan elkaar gelinkt. Zo verwijst de ene kolom van een tabel naar een kolom van een andere tabel. Het doel van deze tabellen is om ervoor te zorgen dat het globale databankmodel overzichtelijk blijft. Daarnaast bieden ze ook veiligheid. Als een webpagina geladen wordt, is het belangrijk om overbodige data te vermijden.

Tussen meerdere tabellen bestaan er verschillende soorten connecties. De belangrijkste en meest gebruikte relaties zijn:

- één-op-éénrelaties: één rij uit een tabel verwijst naar één rij uit een andere tabel;
- één-op-veelrelaties: één rij uit een tabel verwijst naar meerdere rijen uit een andere tabel;
- veel-op-veelrelaties: meerdere rijen uit een tabel verwijzen naar meerdere rijen uit een andere tabel. Voor deze relatie is nog een derde tabel nodig.

De relaties worden later uitgelegd aan de hand van het databankmodel van de projectenwebsite.

### Databankmodel projectenwebsite



Figuur 2: Databankmodel projectenwebsite

Het databankmodel (figuur 2) van de projectenwebsite bestaat uit acht tabellen. Elk is minstens verbonden met één of meerdere tabellen:

- gebruikerstabel: verzameling van de algemene gegevens van alle gebruikers;
- docententabel: verzameling van de gebruikersnummers van alle docenten;
- externentabel: verzameling van alle niet-studenten en niet-docenten. Hierbij wordt ook een bedrijf opgeslagen;
- studententabel: verzameling van het Belbintype en het groepsnummer van iedere student;
- groepentabel: verzameling van alle groepen en de projecten waarbij ze horen;
- projectentabel: verzameling van de gegevens van alle projecten;
- SMART-criteriatabel: verzameling van de SMART-criteria per project;
- mededelingentabel: verzameling van alle mededelingen bij projecten.



De docent-, externen- en studententabel hebben elk een één-op-éénrelatie met de gebruikerstabel. Elke gebruiker is dus ofwel een student, docent of externe. Projecten kunnen door gelijk welke gebruiker aangemaakt worden. Elk project heeft eigen SMART-criteria die ook ingevuld dienen te worden. Groepen kunnen gevormd worden door maximum 4 studenten. Als laatste kunnen gebruikers nog mededelingen plaatsen bij projecten. Dit is een veel-op-veelrelatie m.a.w. elke gebruiker kan meerdere mededelingen plaatsen bij verschillende projecten en ieder project kan meerdere mededelingen bevatten van verschillende gebruikers.

## 2.4 Migrations en seeders

Migrations en seeders zijn bestanden die in Laravel gebruikt worden om snel tabellen uit het databankmodel te maken. Dit is vooral handig als er met meerdere personen aan eenzelfde website wordt gewerkt. Door de migrations en seeders uit te voeren (codefragment 2), is iedere developer zeker dat hij exact dezelfde databank gebruikt. Zowel migration- als seederbestanden kunnen automatisch aangemaakt worden door opnieuw het commando 'PHP artisan make: migration <Naam>' uit te voeren.

```
root@1746b1deacc4:/var/www# php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2019_04_04_104214_create_users_table
Migrated: 2019_04_04_104214_create_users_table
Migrating: 2019_04_04_104305_create_students_table
Migrated: 2019_04_04_104305_create_students_table
Migrating: 2019_04_04_104458_create_teachers_table
Migrated: 2019_04_04_104458_create_teachers_table
Migrating: 2019_04_04_104629_create_projects_table
Migrated: 2019_04_04_104629_create_projects_table
Migrating: 2019_04_04_113558_create_groups_table
Migrated: 2019_04_04_113558_create_groups_table
Migrating: 2019_04_04_113751_create_comments_table
Migrated: 2019_04_04_113751_create_comments_table
Migrating: 2019_04_04_230532_create_externs_table
Migrated: 2019_04_04_230532_create_externs_table
Migrating: 2019_04_16_213243_create_smartcriteria_table
Migrated: 2019_04_16_213243_create_smartcriteria_table
root@1746b1deacc4:/var/www# php artisan db:seed
Seeding: create_comments_seeder
Seeding: create_groups_seeder
Seeding: create_projects_seeder
Seeding: create_students_seeder
Seeding: create_teachers_seeder
Seeding: create_users_seeder
Seeding: create_externs_seeder
Seeding: create_smartcriteria_seeder
Database seeding completed successfully.
root@1746b1deacc4:/var/www#
```

Codefragment 2: Uitvoeren van migrating en seeding

Migrations zijn bestanden waarin de databanktabellen zelf omschreven worden. Ieder migrationbestand bestaat uit 2 functies: de up- en downfunctie (codefragment 3).

De upfunctie wordt opgeroepen bij het uitvoeren van het commando 'php artisan migrate'. Hierin wordt beschreven hoe de tabel zal heten en welke kolommen er zijn. Bij iedere kolom wordt ook het type van de kolom vermeld en of het veld van de kolom verplicht moet ingevuld zijn.

De downfunctie wordt gebruikt bij het verwijderen van bestaande migrations. Hierbij zal de tabel automatisch verwijderd worden, indien ze nog zou bestaan.

```
1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('surname');
20             $table->string('email');
21             $table->string('password');
22             $table->rememberToken();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33         Schema::dropIfExists('users');
34     }
35 }
```

Codefragment 3: Migrationbestand gebruikers

Door de seederbestanden worden de tabellen ingevuld (codefragment 4). Op deze manier kan de databank snel voorzien worden van bruikbare data en verliest de webdeveloper geen tijd. Elke seeder bestaat uit een functie 'run'. Deze bestaat uit een reeks inserts<sup>1</sup>. Bij elke insert wordt zowel de tabelnaam vermeld, alsook de kolommen met bijhorende data.

<sup>1</sup> Commando waarmee gegevens in een tabel van een databank worden toegevoegd. [3]



```

1  <?php
2
3  use Illuminate\Database\Seeder;
4
5  class create_users_seeder extends Seeder
6  {
7      /**
8       * Run the database seeds.
9       *
10      * @return void
11      */
12      public function run()
13      {
14          DB::table('users')->insert([
15              'id' => 1,
16              'name' => "Lauwers",
17              'surname' => "Andreas",
18              'email' => "andreas.lauwers@student.odisee.be",
19              'password' => bcrypt('test'),
20          ]);

```

Codefragment 4: Seederbestand gebruikers

## 2.5 Eloquent modellen

Laravel maakt gebruik van eloquente modellen (codefragment 5). Dit zijn klassen die elk aan een eigen overeenkomstige tabel van het databankmodel kunnen gelinkt worden. Deze methode zorgt voor een extra laag tussen databank en website en vergroot hiermee de beveiliging naar de achterliggende data. Daarnaast worden het toevoegen en aanpassen van data vergemakkelijkt.

Eloquent modellen zijn telkens op eenzelfde manier opgebouwd. Bovenaan staan steeds verwijzingen naar geïncludeerde bestanden. Deze worden door Laravel automatisch gegenereerd. Vervolgens wordt een klasse gedefinieerd die overgeërfd wordt van een basis Laravelmodel. Bovenaan in deze klasse wordt een array aangemaakt met daarin alle kolommen die de overeenkomstige tabel uit de databank bezit. Dit zorgt ervoor dat het mogelijk is om items op te roepen aan de hand van deze kolomnamen.

Onder de array worden functies aangemaakt. Deze beschrijven de relaties tussen naburige tabellen. Hierdoor is het niet meer nodig om inner joins te gebruiken om data uit meerdere tabellen in 1 keer weer te geven.

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Students extends Model
8  {
9      protected $fillable = ['id', 'belbintype', 'group_id'];
10     public function users(){
11         return $this->hasOne('App\Users', 'id', 'id');
12     }
13     public function groups(){
14         return $this->belongsTo('App\Groups', 'id', 'group_id');
15     }
16 }
17

```

Codefragment 5: Eloquent model student

## 2.6 Controllers

Eens de modellen af zijn, kan er over controllers worden nagedacht. Deze worden net als eloquente modellen automatisch aangemaakt door Laravel. In deze bestanden komen allerlei functies terecht die ervoor zorgen dat de juiste databankgegevens worden gekoppeld aan de juiste webpagina (codefragment 6). Hierdoor eindigen de functies meestal met ofwel een 'redirect'<sup>2</sup> of een 'return view'<sup>3</sup>.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Auth;
6  use App\Groups;
7  use App\Projects;
8  use App\Users;
9  use App\Students;
10 use App\Http\Controllers\Controller;
11
12 use Illuminate\Http\Request;
13
14
15 class OverviewController extends Controller
16 {
17     /**
18      * Create a new controller instance.
19      *
20      * @return void
21      */
22     public function __construct()
23     {
24         $this->middleware('auth');
25     }
26
27     /**
28      * Show the application dashboard.
29      *
30      * @return \Illuminate\Contracts\Support\Renderable
31      */
32     public function index()
33     {
34         $user = Auth::user();
35         $belbin = Students::select('belbintype')->find((int)$user['id']);
36         if ($belbin['belbintype'] == NULL){
37             return redirect('/belbintest');
38         }
39         else{
40             $year = date("Y")-1;
41             $projects = Projects::join('groups', 'projects.id', '=', 'groups.project_id')->join('students', 'students.group_id', '=',
42             'groups.id')->join('users', 'users.id', '=', 'students.id')->whereYear('created_at', $year)->get();
43             echo $projects;
44             return view('overzicht', [
45                 'projects' => $projects,
46                 'user' => $user
47             ]);
48         }
49     }

```

Codefragment 6: Controller van overzichten

<sup>2</sup> Deze coderegel zorgt ervoor dat de browser wordt omgeleid naar een andere webpagina op een ander webadres. [5]

<sup>3</sup> Dit codeblok voorziet dat er een webpagina wordt geladen met de correct meegegeven parameters. [5]

## 2.7 Blade

Zoals eerder gezegd kan een controllerfunctie resulteren in een 'return view'. Hierbij wordt een Bladepagina opgeroepen en worden de bijhorende parameters meegegeven.

Blade is de template engine van Laravel. In deze templates wordt HTML-code gegenereerd op basis van meegegeven parameters. De engine zet deze parameters om in codefragmenten om zo de website te vervolledigen. Zo is het bijvoorbeeld mogelijk om verschillende functies in een Bladedocument te plaatsen. Naast if-statements, whileloops, forloops, ... kan ook zuivere PHP-code geïmplementeerd worden in deze pagina's. De parameters die toegevoegd zijn vanuit de functie van een controller kunnen weergegeven worden als tekst d.m.v. dubbele accolades (codefragment 7).

```

66 <body>
67 <div class="flex-center position-ref full-height">
68     @if (Route::has('login'))
69         <div class="top-right links">
70             @auth
71                 <a href="{{ url('/home') }}">Home</a>
72             @else
73                 <a href="{{ route('login') }}">Login</a>
74             @if (Route::has('register'))
75                 <a href="{{ route('register') }}">Register</a>
76             @endif
77             @endauth
78         </div>
79     @endif
80 
```

*Codefragment 7: Nazien of een gebruiker ingelogd is m.b.v. Bladeparameters.*

## 2.8 Vue in Laravel

In het hoofdstuk hieronder wordt Vue uitgebreid beschreven. Hier zal er gekeken worden naar de implementatie van Vue in Blade. Aangezien er in de frontend gebruik gemaakt wordt van Vue componentens moet er hiernaar verwezen worden in Blade.

Een groot voordeel van het gebruik van Vue is dat er meer in de frontend kan gebeuren. Dit heeft als gevolg dat de pagina niet bij elk request herladen moet worden. Hierdoor zal het platform sneller werken omdat het slechts 1 maal data moet ophalen.

In codefragment 8 wordt een Vue component opgeroepen in Blade. De parameter "\$projects" en "\$archieff" worden mee gegeven uit de controller. Via de Blade worden die parameters verder doorgegeven aan de props in de Vue files.

```

1  {{-- extend the parent tpl--}}
2  @extends('layouts.app')
3  {{-- set the pagetitle in the parent tpl--}}
4  @section('name', 'Overzicht')
5  @section('content')
6      @if(!$archieff)
7          <overzicht_vue :projects="{{ $projects }}"></overzicht_vue>
8      @else
9          <archieff_vue :projects="{{ $projects }}"></archieff_vue>
10     @endif
11 @endsection

```

*Codefragment 8: Gebruik van Vue in Laravel*

## 2.9 Routes

Laravel zorgt voor cleane URL's. Dit wil zeggen dat de links naar webpagina's aangepast kunnen worden zonder dat de opgeroepen bestanden hiervoor van naam moeten veranderen. Daarnaast worden er dan ook geen bestandsextensies weergegeven bij het surfen naar de pagina. Dit bevordert de ervaring van de gebruiker. Door de routes logisch op te stellen, zal een gebruiker in staat zijn om de juiste pagina te vinden. Zo verliest hij geen tijd bij het zoeken in menu's. Zeker bij grote (onoverzichtelijke) websites kan dit wel degelijk nuttig zijn.

```
1  <?php
2
3  /*
4  |-----
5  | Web Routes
6  |-----
7  |
8  | Here is where you can register web routes for your application. These
9  | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Auth::routes();
15
16 Route::get('/', 'OverviewController@index')->name('home');
17 Route::get('/archief', 'OverviewController@archive')->name('archief');
18 Route::get('/maakproject', 'ProjectController@createProject')->name('maakproject');
```

Codefragment 9:Routes

Eén van Laravels belangrijkste hulpmiddelen zijn de routebestanden (codefragment 9). Hierin wordt bepaald welke functie uit een controller wordt opgeroepen als er naar een bepaalde URL wordt gesurft.

## 2.10 Authenticatie

Laravel biedt een ingebouwde authenticatie aan. Door het commando 'php artisan make:auth' uit te voeren in een terminal, worden alle nodige bestanden aangemaakt. Hierdoor zal er een eenvoudige login- en registreerpagina gevormd worden, alsook de achterliggende controllers. Standaard gebeurt authenticatie op basis van de paswoord- en e-mailkolom uit de gebruikerstabel. Dit is ook van toepassing bij het maken van de projectenwebsite.

Doordat de authenticatie automatisch gebeurt, kunnen de andere controllers makkelijk controleren indien een gebruiker daadwerkelijk ingelogd is. Met behulp van de coderegel 'Auth::user()' worden alle gegevens van de ingelogde persoon geladen. Indien de waarde 'NULL' is, betekent dit dat de gebruiker nog niet is ingelogd.

## 2.11 Conclusie

Laravel is een gratis, opensource maar vooral eenvoudig PHP web framework. De documentatie is zeer uitgebreid. Door de grote community is er veel overzichtelijke documentatie beschikbaar. In het framework zitten veel handige ingebouwde functionaliteiten. De structuur is overzichtelijk waardoor de code goed leesbaar is. Door Eloquent ORM is de relatie tussen de database veel eenvoudiger. Via een speciaal routebestand wordt er gezorgd dat de juiste request met de juiste methodes verbonden worden. Dit maakt het voor de developer een stuk overzichtelijker.

## 3. BootstrapVue

### 3.1 Framework Vue

Vue is een JavaScriptframework voor het bouwen van gebruikersinterfaces. In tegenstelling tot andere monolithische frameworks, is Vue vanaf de basis ontworpen om zeer eenvoudig toepasbaar te zijn. De kernbibliotheek is hoofdzakelijk gericht op de presentatie laag en is eenvoudig te integreren met andere bibliotheken of bestaande projecten. Aan de andere kant is Vue ook perfect in staat om geavanceerde Single-Page Applicaties aan te sturen in combinatie met ondersteunende bibliotheken.

Eén van de redenen waarom voor Vue.js wordt gekozen, is dat JavaScript eenvoudig kan worden toegevoegd. De broncode is zeer goed leesbaar en de documentatie is de enige tutorial die er nodig is. Er zijn geen externe bibliotheken nodig. Het is ook niet vereist om plug-ins te installeren, hoewel er veel plug-ins beschikbaar zijn. Het aansluiten van Vue.js op bestaande code is ook heel eenvoudig (codefragment 10) [9].

Om een voorbeeld te geven van wat er bedoeld wordt:

```
<div id="journal">
  <input type="text" v-model="message">
  <div>{{message}}</div>
</div>
```

Codefragment 10: Voorbeeld van Vue [9]

Voor iemand die al ervaring heeft met JavaScript-bibliotheken, is codefragment 11 vrij eenvoudig te verstaan. De “new Vue” zorgt ervoor dat er een nieuwe component kan worden aangemaakt met daarin de nodige data. Deze relatief korte en simpele code is een van de sterktepunten van Vue.js in tegenstelling tot andere frameworks.

```
var journal = new Vue({
  el: '#journal',
  data: {
    message: 'Your first entry'
  }
});
```

Codefragment 11: JavaScript in Vue

### 3.2 Componenten

De meeste frontendframeworks zorgen ervoor dat een website kan worden opgebouwd uit componenten. Elk belangrijk afzonderlijk onderdeel van een site kan worden omgezet in zo een component. Elke component heeft dus zijn eigen HTML-, JavaScript- en CSS-code indien nodig.

Elke webpagina bestaat uit één grote component die opgebouwd is uit één of meerdere kleinere componenten. Elke component kan een parent- en childcomponent bevatten. Parent A is de component waarin de huidige component B zich bevindt en B is dan de childcomponent van component A. Zo is alles heel overzichtelijk en kan je elk deel perfectioneren naar eigen belang en zin, zonder dat je met de rest rekening moet houden. Alles is mooi geordend en als er een fout is, kan je ze makkelijk opsporen en verhelpen.

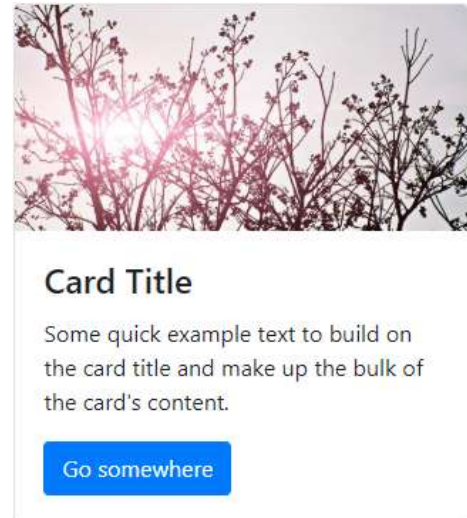
### 3.3 BootstrapVue

BootstrapVue is het framework waarmee te werk is gegaan, exact hetzelfde als Vue maar Bootstrap (toolkit om HTML, CSS, JavaScript makkelijker te implementeren) wordt hier mee in geïntegreerd. Dit maakt het mogelijk zo weinig mogelijk code te moeten schrijven voor wat men wil bereiken. Bootstrap werkt met voorgemaakte klassen, onderdelen, afkortingen, binnenwegen, ... om zo efficiënter te kunnen programmeren.

```
<div>
  <b-card
    title="Card Title"
    img-src="https://picsum.photos/600/300/?image=25"
    img-alt="Image"
    img-top
    tag="article"
    style="max-width: 20rem;"
    class="mb-2"
  >
    <b-card-text>
      Some quick example text to build on the card ti
    </b-card-text>

    <b-button href="#" variant="primary">Go somewhere</b-button>
  </b-card>
</div>
```

Codefragment 12: Card in Vue



Figuur 3: Card in Vue

In codefragment 12 wordt dit alles duidelijk, waar een “card” voorgesteld wordt. De tag “b-card” is een bestaand onderdeel in BootstrapVue. In deze tag kunnen extra opties zoals border, afbeeldingen, stijl... gewijzigd, ingevoegd worden. De voorgemaakte klassen kan men ook terugvinden in de optie class.

Een klasse is een groep van stijlen die toegepast zullen worden op meerdere elementen. Maar Bootstrap heeft echter ook voorgemaakte klassen, die de meest voorkomende stijlen bevatten. Met enkele letters worden deze zo toegevoegd aan het object. Zo verwijst ‘mb-2’ naar “margin-bottom = 2rem;” (codefragment 12) [10].

### 3.4 HTML

Elke component heeft dus zijn eigen HTML-, JavaScript- en CSS-gedeelte. HTML wordt geschreven in een “<template>”-tag, Vue heeft een ingebouwd grid- en lay-outsysteem wat alles zeer overzichtelijk en gebruiksvriendelijk maakt (figuur 4). Daarnaast zorgt dit systeem ervoor dat alle pagina’s ook responsive<sup>4</sup> gemaakt worden.

<sup>4</sup> Ervoor zorgen dat een website zich automatisch aanpast naargelang de grootte van het scherm van de gebruiker. [12]



col-sm-8		col-sm-4
col-sm	col-sm	col-sm

*Figuur 4: Grid- en lay-outsysteem*

```

<b-container class="bv-example-row">
  <b-row>
    <b-col sm="8">col-sm-8</b-col>
    <b-col sm="4">col-sm-4</b-col>
  </b-row>

  <b-row>
    <b-col sm>col-sm</b-col>
    <b-col sm>col-sm</b-col>
    <b-col sm>col-sm</b-col>
  </b-row>
</b-container>

```

*Codefragment 13: Grid- en lay-outsysteem in Vue*

De pagina wordt bekeken als een tabel van rijen en kolommen. Elke rij en/of kolom kan worden aangepast naar eigen wil: de breedte, hoogte, responsive, enz.

In elke rij en kolom die nodig is, wordt dan de nodige html geïmplementeerd met zijn eigen stijl die onderaan het bestand wordt toegevoegd in het CSS-gedeelte.

### 3.5 JavaScript

In het <script> gedeelte wordt de JavaScriptcode geschreven: de functionaliteit. Hier wordt de data beschreven en bewerkt.

```

data: {
  name: 'Vue.js'
},
// define methods under the 'methods' object
methods: {
  greet: function (event) {
    // 'this' inside methods point to the Vue instance
    alert('Hello ' + this.name + '!')
    // 'event' is the native DOM event
    alert(event.target.tagName)
  }
}
})

```

*Codefragment 14: JavaScript methoden in Vue*

In het scriptgedeelte kan alles meegegeven, gedeclareerd en bewerkt worden. De parentcomponent kan “props” doorgeven. Dit zijn data die kunnen doorgegeven worden. Daarnaast blijven uiteraard nog de huidige data aanwezig. Deze kunnen bewerkt worden in het gedeelte van de methoden. Er zijn ook lifecyclemethoden. Deze worden altijd, één voor één, opgeroepen bij elk deel van het laadproces van een component.

Voorbeeld: De “props” worden doorgegeven van de parentcomponent. Deze zijn nog niet beschikbaar als de website ingeladen wordt. Dit gebeurt pas als de pagina met de huidige component ingeladen wordt. Eén van de vele lifecyclemethoden houdt in dat deze pas opgeroepen wordt nadat de component ingeladen is, nl. mounted. In deze methode zijn dus de “props” al beschikbaar en kunnen ze bewerkt worden.

### 3.6 CSS

Voor traditionele websites zijn er veel manieren waarop CSS kan worden geïmplementeerd. De meest voorkomende manier is om de opmaak van elke webpagina in een eigen CSS-bestand te zetten. Voor kleinere pagina's is er vaak ook CSS binnen de HTML-pagina (meestal in het <head> gedeelte). Dit resulteert in grote CSS-bestanden, die moeilijk te beheren zijn.

```
1 <template>
2   <div class="success-message">this will be green</div>
3 </template>
4
5 <style scoped>
6   .success-message{
7     color: green;
8   }
9 </style>
```

Codefragment 15: CSS in Vue

In Vue kan elk onderdeel zijn eigen styling hebben (codefragment 15). Het scoped attribuut zorgt ervoor dat deze CSS-code enkel lokaal toegepast wordt. Anders bestaat het risico dat andere delen van de site die dezelfde klassenaam hebben ook aangepast worden.

### 3.7 Besluit

Vue is zeer handig voor het opsplitsen van lange codeteksten. Door alles op te bouwen in kleine componenten wordt overzicht bewaard op zowel HTML, CSS als JavaScript. Per component is elk codeblok dus vrij beperkt in lengte. Voor de projectenwebsite die relatief complex is zorgt dit ervoor dat er efficiënt kan worden gewerkt.



## 4. Webtoepassing hosten

Met Microsoft Azure kunnen er snel en efficiënt cloudoplossingen worden ontwikkeld, geïmplementeerd en beveiligd. Het implementatiemodel van Azure is flexibel. Zo kan er in Visual Studio code worden ontwikkeld. Deze kan worden doorgevoerd naar GitHub, gebouwd met Azure DevOps en vervolgens in Azure geïmplementeerd. Er kunnen ook een andere IDE (Integrated Development Environment) gebruikt worden. Ook is het mogelijk om rechtstreeks vanaf de ontwikkelcomputer code in de cloud te implementeren.

Er is een groot gamma aan programmeertalen, van C# tot Java en Node.js. De gebruiker kiest zelf het ontwikkelbesturingssysteem. Microsoft biedt en ondersteunt een breed scala aan ontwikkelprogramma's voor Windows, Linux en macOS.[11]

### 4.1 Azure portal

De eerste stap bij het hosten van de webtoepassing is het maken van een application service binnen het Azure-abonnement.

Er zijn verschillende manieren waarop dit kan worden gemaakt. Er kan Azure Portal, de Azure-CLI, en script of een IDE worden gebruikt.

De grafische interface van de Azure-portal is zeer gebruiksvriendelijk. In de portal kunnen er extra resources toegevoegd worden en bestaande resources kunnen worden aangepast.

### 4.2 Azure App service

Azure App Service is een volledig beheerd computerplatform in de Azure-omgeving, geoptimaliseerd voor het hosten van web-apps, REST-API's en back-ends.

Met dit PaaS (Platform as a Service) aangeboden door Microsoft Azure kan men zich concentreren op het ontwikkelen, terwijl Azure zorgt voor de infrastructuur die nodig is om de toepassingen uit te voeren en uit te breiden.

### 4.3 Web-app

Wanneer er een app moet worden gehost, kan er in de portal een nieuwe web-app worden aangevraagd. Door het maken van een web-app in Azure Portal, worden er een aantal hostingresources in App Service gemaakt. Deze kunnen gebruikt worden voor het hosten van een applicatie, ondersteund door Azure, ongeacht of deze is geschreven in ASP.NET Core, Node.js, PHP, enzovoort. In figuur 5 wordt een configuratie van een framework of programmeertaal van een app weergegeven.

Home > Alle resources > BestBike1234 - Toepassingsinstellingen wijzigen

**BestBike1234 - Toepassingsinstellingen wijzigen**  
App Service

Zoeken (Ctrl+/,)

Opslaan Negeren

**Instellingen**

- Toepassingsinstellingen wijzigen...
- Authentication / Authorizati
- Application Insights
- Beheerde service-identiteit
- Back-ups
- Aangepaste domeinen
- SSL-instellingen
- Networking
- Opschalen (App Service-plan)
- Uitschalen (App Service-plan)
- WebJobs
- Push
- MySQL In-app
- Eigenschappen

**Algemene instellingen**

.NET Framework version v4.7

PHP version 5.6

App Service ondersteunt de installatie van nieuwere Python-versies. Klik hier voor me...

Python version Uit

Java version Uit

Java minor version

Java web container

Platform 32-bit 64-bits

Web sockets Uit Aan

Always On Uit Aan

Figuur 5: Instellingen in Azure

Dankzij de vele beschikbare hulpprogramma's in Azure Portal kan er volledig bepaald worden hoe de applicatie wordt beheerd en beveiligd.

#### 4.4 Implementatiesites

Met behulp van Azure Portal kunnen er eenvoudig implementatiesites aan een App Service-web-app worden toegevoegd. Er kan bijvoorbeeld een implementatiesite voor fasering worden gemaakt. Hier kan code naar gepusht worden om deze te testen in Azure. Als alles in orde is, kan de staging-implementatiesite eenvoudig worden gewisseld met de productiesite. In Azure Portal gebeurt dit allemaal met een paar muisklikken (figuur 6).

Start > Alle resources > BestBike1234 - Implementatiesites

**BestBike1234 - Implementatiesites**  
App Service

Zoeken (Ctrl+/,)

+ Site toevoegen Wisselen

**Implementatie**

- Snelstart
- Implementatiereferenties
- Implementatiesites
- Implementatie-opties
- Implementatiecenter (preview)

NAAM	STATUS	APP SERVICE-PAN
bestbike1234-staging	Wordt uitgevoerd	BestBike1234-app-service-plan

Figuur 6: Voorbeeld van Implementatiesite.

## 4.5 Ondersteuning continue integratie en implementatie

Azure Portal biedt onmiddellijk na aanschaf continue integratie en implementatie met Azure DevOps, GitHub, Bitbucket, Dropbox, OneDrive en een lokale Git-opslagplaats voor de ontwikkelcomputer. Wanneer de applicatie verbonden wordt met een bron, doet de App Service de rest. De code en toekomstige wijzigingen hiervan zullen automatisch synchroniseren met de applicatie. Bovendien kan er met Azure DevOps een eigen build- en releaseproces worden gedefinieerd voor het compileren van de broncode, het uitvoeren van de tests, het bouwen van een release en ten slotte het implementeren van de release naar de applicatie, telkens wanneer de code wordt doorgevoerd. Dit gebeurt allemaal zonder tussenkomst van de gebruiker.

## 4.6 Samenvatting

Azure App Service vereenvoudigt het beheren en controleren van de applicatie in vergelijking met traditionele hostingopties. Met App Service kost het minder tijd en moeite om de applicatie uit te voeren en te beheren. Het biedt bovendien geavanceerde Cloudfuncties, zoals automatisch schalen en Git-integratie.

## 5. Functionaliteiten

### 5.1. Integratie van de Belbin test

Na het aanmaken van een account moeten de gebruikers een Belbin test invullen. Via deze test, die is gebaseerd op dr. Belbins ideeën en is ontwikkeld voor de Nederlandse markt, kan er bepaald worden in welke mate, vanuit de persoonlijkheid, de verschillende teamrollen er worden aangesproken.

Een team bestaande uit creatievelingen loopt het risico te verzanden in ideeën. Een team bestaande uit uitvoerders loopt het risico veel werkzaamheden te verzetten zonder bevredigend resultaat. Een team dat zich van dit alles bewust van is, kan hierdoor de verdeling van taken en functies organiseren met het oog op een betere werking van het team met gepast resultaat.

### 5.2. Project toevoegen

Projecten of ideeën kunnen door iedere gebruiker, zowel docent als student worden toegevoegd. Bij het aanmaken van een project moeten de SMART-criteria worden opgegeven. Deze worden gebruikt om meetbare en realistische doelstellingen te formuleren. Het vaststellen van meetbare doelstellingen levert betere resultaten op en maakt het achteraf eenvoudiger om de realisatie ervan te evalueren.

### 5.3. Goedkeuren project

In de oude versie worden projecten door docenten aan de database toegevoegd en voorstellen goedgekeurd. Studenten kunnen enkel een projectvoorstel indienen en zich hiervoor inschrijven. Bovendien hebben ze geen toegang tot afgelopen projecten om inspiratie op te doen. In de nieuwe versie kunnen projecten op een eenvoudige manier worden goed- of afgekeurd. Gebruikers kunnen ook projecten van de voorbije jaren bekijken om hen te helpen bij het ontwikkelen van een idee.

### 5.4. Klassensysteem

Op de nieuwe versie van de projectenwebsite is er een duidelijk verschil tussen docent en student. Studenten kunnen meerdere voorstellen indienen die op hun beurt worden goed- of afgekeurd. Indien een voorstel wordt afgekeurd kunnen studenten dit alsnog aanpassen naar de gevraagde criteria.

## 6. Planning

In het project zijn er 4 grote onderdelen, namelijk:

- Wireframe, usability, design
- HTML & CSS
- Web scripting
- Extra's (Responsive, Belbin test, chat, ...)

Einddatum	Mijlpaal	Wie?
20 februari 2019	Vergadering #1 Eerste brainstorm website	Groep
21 februari 2019	Afspraak #1 begeleider	Groep
Krokusvakantie	Vergadering #2 Tweede brainstorm website	Groep
11 maart 2019	<b>User Experience</b>	Groep
14 maart 2019	Vergadering #3 en Afspraak #2 begeleider Brainstorm design en database- ontwerp, feedback wireframes	Groep
21 maart 2019	<b>HTML</b>	Groep
28 maart 2019	<b>CSS</b>	Groep
4 april 2019	Afspraak#3 begeleider	Groep
22 april 2019	<b>Laravel, MySQL, Blade</b>	Andreas en Lukas
25 april 2019	Vergadering #4 en Feedback Laravel, MySQL, Blade	Groep
25 april 2019	Afspraak #4 begeleider	Groep
2 mei 2019	<b>Vue</b>	Arijn en Arno
8 mei 2019	Vergadering #5 Feedback Vue	Groep
9 mei 2019	Afspraak #5 begeleider	Groep
15 mei 2019	<b>Extra's</b>	Groep
16 mei 2019	Vergadering #6 Overlopen eindproduct, documentatie afwerken.	Groep
22 mei 2019	Eindproduct indienen	Groep

## Conclusie

De nieuwe website heeft een inlogstelsel voor studenten en docenten. Studenten moeten verplicht de Belbin test afleggen. Aan de hand daarvan krijgen ze voorgestelde groepen waarin ze terecht zouden passen. Ze beslissen wel nog zelf of ze bij de groep willen horen of niet, daarnaast kunnen ze een projectvoorstel toevoegen die dan door een docent goed- of afgekeurd kan worden. Verder kunnen de docenten opmerkingen meegeven.

Verder bestaat de website uit de nodige pagina's zoals het archief om inspiratie op te doen uit projecten van vorige jaren. De studentenpagina om informatie op te zoeken over een bepaalde student, groep, projectvoorstel of om te sorteren op aantal groepsleden, op achternaam, ...

Het is een klein maar krachtig platform, voor één bepaald doeleinde, nl. de groepsvorming van de projecten niet meer ten laste te leggen van de docenten. Maar eerder een taak van de studenten om zo snel mogelijk groepen te vormen en hun voorstel indienen zodat de docenten niet meer face to face of via e-mailberichten de voorstellen kunnen keuren en de groepen kunnen vaststellen in hun bestand. Dit platform bespaart dus een hoop tijd en typ- en administratiewerk voor de docenten.

## Overzicht bijeenkomsten

### Donderdag 14 februari:

Tijdens de eerste bijeenkomst wordt er beslist hoe het project wordt uitgewerkt: technieken, programmeertalen en databankmodel. Meerdere ideeën komen naar boven. Uiteindelijk wordt er gekozen om het frontendgedeelte met Vue te maken en het backend met Laravel, aangezien de deze technieken zo aangeleerd krijgen.

### Donderdag 21 februari:

De initiële ideeën worden besproken met de docent. Het databankmodel en -type komen hier ook aan bod. Uiteindelijk wordt er gekozen voor MySQL.

De taken hebben we verdeeld naargelang ieders voorkeur: twee frontend, twee backend. Het userexperienceonderzoek wordt gezamenlijk uitgevoerd. Daar wordt vandaag al aan begonnen. Er zijn 2 mails gestuurd naar docenten om hun visie van de nieuwe projectenwebsite na te vragen. Ook wordt er nagedacht over hoe er kan onderzocht worden wat studenten van de site verwachten.

### Donderdag 28 februari:

De groep heeft zijn research en werk over het userexperiencegedeelte gedaan. Nu worden de wireframes afgewerkt samen met de persona's. De planning van de komende weken wordt overlopen. Ieder werkt aan zijn deel. Het is de bedoeling om voor de paasvakantie de frontend klaar te hebben en de backend opgesteld te hebben zodat we in de paasvakantie dit kunnen omzetten.

### Vrijdag 19 april:

Na onze afspraak met meneer Van Peteghem zien we de achterstand van het project in. We bespreken meteen de stand van zaken om de backend te kunnen laten samensmelten met de frontend.

### Vrijdag 17 mei:

In deze bijeenkomst zijn 2 van de 5 hoofdpagina's omgezet en de filters werkende gemaakt.

### Dinsdag 21 mei:

De frontend- en backenddevelopers hebben opnieuw samen gezeten om de code op elkaar af te stemmen. De website is nu bijna volledig aangepast. De stijl is zoals we het willen. Het rapport is klaar op door gestuurd te worden.

## Literatuurlijst

### Online handleidingen

- [1] Laravel, "Database: Getting Started," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/database>. [Accessed 5 mei 2019].
- [2] Laravel, "Database: Migrations," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/migrations>. [Accessed 5 mei 2019].
- [3] Laravel, "Database: Seeding," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/seeding>. [Accessed 5 mei 2019].
- [4] Laravel, "Eloquent: Relations," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/eloquent-relationships>. [Accessed 5 mei 2019].
- [5] Laravel, "Controllers," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/controllers>. [Accessed 5 mei 2019].
- [6] Laravel, "Blade Templates," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/blade>. [Accessed 5 mei 2019].
- [7] Laravel, "Routing," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/routing>. [Accessed 5 mei 2019].
- [8] Laravel, "Authentication," Laravel Documentatie, [Online]. Available: <https://laravel.com/docs/5.8/authentication>. [Accessed 5 mei 2019].
- [9] Vue.js, "Introduction," Vue Documentatie, [Online]. Available: <https://vuejs.org/v2/guide>. [Accessed 5 mei 2019].
- [10] Bootstrap-vue.js, "Getting started," [Online]. Available: <https://bootstrap-vue.js.org/>. [Accessed 5 mei 2019].
- [11] S. Sanders, *sven.sanders@odisee.be*, *Vragen omtrent project projectenwebsite*, 24 februari 2019.
- [12] w3schools, "HTML Responsive Web Design," [Online]. Available: [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp). [Accessed 5 mei 2019].



## Bijlagenoverzicht

Bijlage 1:	E-mail meneer Sanders.....	33
Bijlage 2:	Interview mevrouw Martens.....	35
Bijlage 3:	Enquête van studenten .....	36
Bijlage 4:	Storyboard .....	40
Bijlage 5:	Persona's .....	41
Bijlage 6:	Wireframes .....	43

## Bijlage 1: E-mail meneer Sanders

Ik wil zeker eens met jullie afspreken, maar volgende week is dat erg moeilijk, omdat ik door stagebezoeken een beperkt aantal ogenblikken op school ben en door andere afspraken minder beschikbaar ben op het ogenblik dat ik op school ben.

Bekijk eens wat er voor jullie op dinsdag na 12u kan en welke mogelijkheden je hebt op donderdag, misschien kunnen we toch nog iets regelen.

Omdat het belangrijk is voor jullie om snel te kunnen starten, zet ik hieronder al een paar zaken op een rijtje.

Qua pure functionaliteit is het belangrijk dat er studenten en docenten kunnen worden toegevoegd, dat er voorstellen kunnen worden ingediend, goedgekeurd en bekeken, dat er groepen gevormd kunnen worden en dat er output geproduceerd kan worden die bruikbaar is voor het maken van plannings, opvolgingsdocumenten...

Als ik daar zelf een paar minimeisen tegenaan mag gooien, dan denk ik in de eerste plaats aan het volgende: Nu is er een koppeling met het Odisee domein voor logins. Dit is niet noodzakelijk, maar wel **cruciaal is dat het eenvoudig is om verschillende soorten gebruikers toe te kunnen voegen aan de hand van een e-mailadres**. Voor de hand liggende rollen zijn docenten en studenten, maar misschien moet er ook iets voorzien worden voor **externe indieners** en moet er misschien ook iets apart voorzien worden voor voorwaardelijk ingeschreven studenten voorstellen van studenten moeten goedkeuring krijgen van een docent. Vergeleken met de huidige procedure, zou het handig zijn als studenten **bij het indienen van het project, de docent kunnen selecteren waarmee ze overlegd hebben**, zodat die het zelf kan goedkeuren uit het systeem moet een groepsverdeling kunnen volgen. Die wordt gebruikt als basis voor allerlei andere documenten en dat gebeurt nu in **csv formaat**.

Het bovenstaande is echter een beschrijving van wat er nu is en waar je met wat kleine aanpassingen het gebruik toch al beter kan maken. Ik denk dan aan de mogelijkheid om voorstellen achteraf te bewerken, opmerkingen van een docent op een voorstel op een andere manier doorgeven aan studenten **zodat voorstellen sneller herwerkt worden...**

Wat volgens mij het voorstel veel interessanter maakt, is dat je **van dit beperkte hulpmiddel voor mij een omgeving probeert te maken die gans het proces van groepsvorming ondersteunt voor alle betrokkenen en in de eerste plaats de studenten** (met bijzondere aandacht voor diegenen die niet meteen een groep van vier kunnen vormen).

Ik denk dan bijvoorbeeld aan een linkedin-achtige uitbreiding waar studenten hun **talenten/interesses kunnen aangeven** en ook gevonden kunnen worden. Of aan de mogelijkheid om gelijkgestemde zielen te vinden, af te spreken en te proberen om samen tot een voorstel te komen. Misschien moet het ook **mogelijk zijn om erg losse ideeën rond te strooien** (b.v. ik wil iets maken dat bruikbaar is als rode draad activiteit voor een jeugdkamp) en zo studenten te vinden die niet noodzakelijk dezelfde technologische interesses hebben en daardoor dan uit te komen bij een heel ander project dan wat je op eigen houtje zou verzinnen.

Ik wil hierin zeker niet te veel beperkingen opleggen door één of andere richting aan te geven, net omdat ik het ook interessant vind dat jullie op basis van je eigen ervaring (jullie hebben de ganse procedure moeten doorlopen) eens bekijkt wat er voor jou anders en beter moet en hoe een online platform daarbij kan helpen.

In het bijzonder zou ik graag hebben dat je eens **nadenkt over hoe je platform studenten kan stimuleren om sneller in actie te komen om voorstellen in te dienen of groepen te vormen.** Daarbij mag je zelfs ingrijpen op de bestaande procedure. Misschien moet er al tussentijds eens een geforceerde samenkomst zijn van studenten die nog geen groep hebben en kan het platform daar suggesties voor doen. Misschien is het een goed idee als vanuit het platform studenten getriggerd worden om nieuwe voorstellen te bekijken of te weten dat er weer nieuwe groepen gevormd zijn of dat er nog maar een beperkt aantal studenten over blijft.

**Afhankelijk van welke richting jullie uitgaan met het project, kan het ook zijn dat je ruimte hebt om het project uit te breiden in het vervolg na de groepsverdeling.** Daar is een stukje voor het proces van de toekenning van de begeleiders, maar vooral ook voor de opvolging. **Misschien kun je de zaken die nu in het opvolgingsdocument van mevrouw Martens zitten, onderbrengen op de website en/of kan je hier een platform creëren waar de opvolging van de groep geregeld wordt.**

Wat mij betreft lag de focus bij het proces van de groepsvorming (dat is ook het meest problematische), maar als jullie na het samenrapen van opmerkingen en in overleg met jullie begeleider bij een ander doel uitkomen, kan dat voor mij ook.

Zoals hopelijk uit de oppervlakkigheid van mijn suggesties blijkt, zie ik het project eerder als een kans om jullie creativiteit los te laten op de vraag hoe de projectenwebsite jou had kunnen stimuleren om sneller een groep te vormen of een eigen voorstel in te dienen.

Ik hoop wel dat je met die suggesties al een beeld krijgt van waar je naartoe kan en dat we volgende week op de vermelde tijdstippen misschien wel een kleiner gat kunnen vinden om nog wat verduidelijking te geven.

Vriendelijke groeten

Sven Sanders [1]

## Bijlage 2: Interview mevrouw Martens

### **Wat zijn uw frustraties van de huidige website?**

-Het duurt zeer lang tegen dat je ingelogd bent

-Grafisch niet zo uitnodigend

### **Wat moet er zeker veranderd worden?**

-Melding Krijgen bij nieuw voorstel

-Beschrijving veranderen naar sjabloon met onderzoeksvraag, extra deelvragen, SMART-criteria overige beschrijving.

-Projecten sorteren op goed, onvoldoende, slecht, pending

-Site toegankelijk maken voor externen.

-Projecten kunnen goedgekeurd worden door alle leerkrachten.

-Schrijver projectvoorstel, projectvoorstel laten aanpassen indien onvoldoende of slecht.

### **Wat is uw mening over het implementeren van Belbin in onze site?**

Super idee!

Misschien belbin test meerdere keren laten doen, bijvoorbeeld in het eerste jaar al tijdens rebug.

Waarschuwing geven als meerdere gelijkaardige types in dezelfde groep zitten

### **Is het de bedoeling dat alles die nu op Toledo staat bij het vak Project + Wetenschappelijk rapporteren ook op de site komt te staan?**

Zeker niet, vooral de vorming van projecten moet de kern van de site zijn.

### **Overige opmerkingen?**

Hoe moet de site onderhouden worden?

Mogelijk maken om als leerkracht projecten te archiveren. Leerlingen kunnen volgende jaren dan zien welke projecten de afgelopen jaren zijn geweest ter inspiratie.

Toestaan om elkaars belbin testen in te kijken.

Mogelijkheid automatische groepsvorming a.d.h.v. algoritmes.

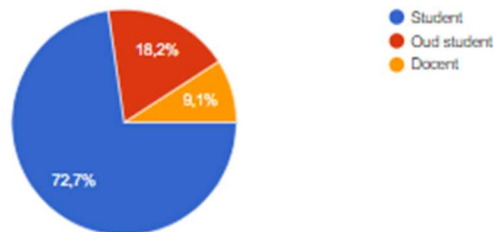
Berekenen hoeveel €€€ we zouden vragen als we een gelijkaardige site maken voor een externe.

### Bijlage 3: Enquête van studenten

De antwoorden die verkregen hebben uit onze enquête.

Ik ben...

11 antwoorden



Figuur 7: Resultaten enquête: Categorie ondervraagde persoon

Heeft u al gebruik gemaakt van de projectenwebsite?

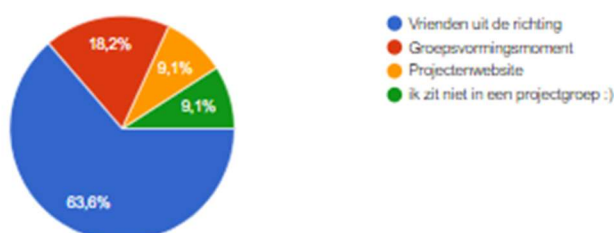
11 antwoorden



Figuur 8: Resultaten enquête: Hebt u al gebruik gemaakt van de projectenwebsite?

Hoe heeft u uw projectgroep gevonden?

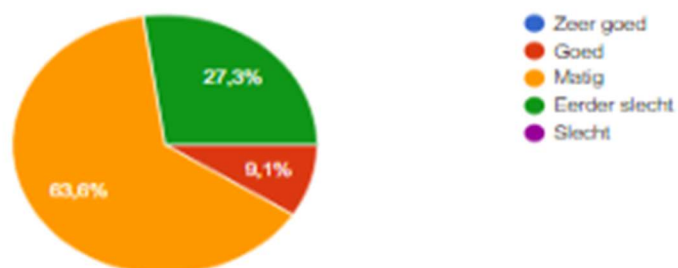
11 antwoorden



Figuur 9: Resultaten enquête: Hoe hebt u uw projectgroep gevonden?

### Hoe was uw ervaring met deze website?

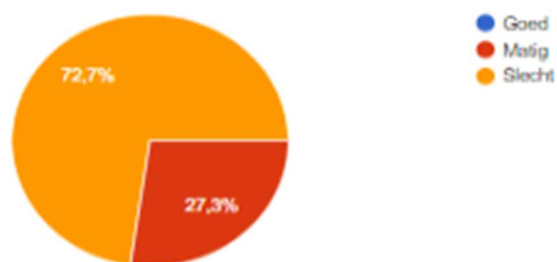
11 antwoorden



Figuur 10: Resultaten enquête: Hoe was uw ervaring?

### Wat vindt u van de grafische opmaak van de site?

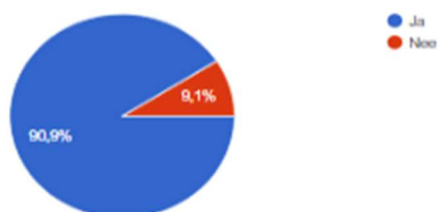
11 antwoorden



Figuur 11: Resultaten enquête: Wat vindt u van de grafische opmaak?

### Vindt u dat de projectvoorstellen gecategoriseerd moeten worden?

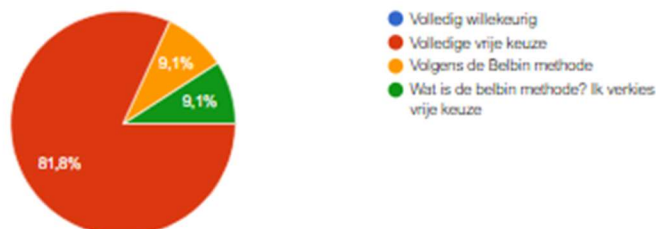
11 antwoorden



Figuur 12: Resultaten enquête: Moeten projecten gecategoriseerd kunnen worden?

## Hoe moet de groepsvorming volgens u gebeuren?

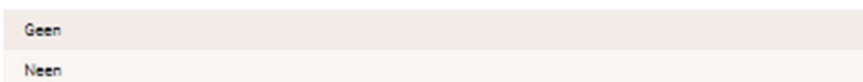
11 antwoorden



Figuur 13: Resultaten enquête: Hoe moet groepsvorming gebeuren?

## Nog andere opmerkingen betreffende de projectenwebsite?

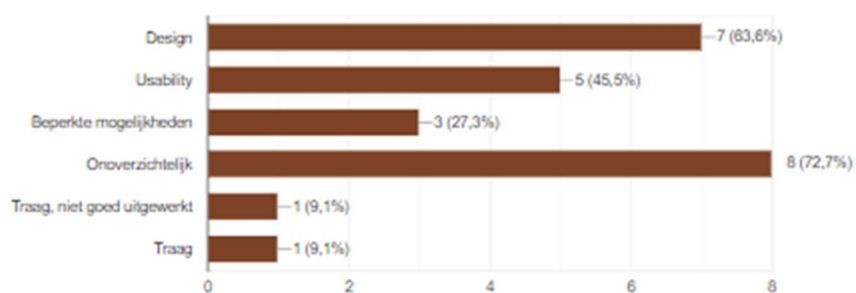
2 antwoorden



Figuur 14: Resultaten enquête: Overige opmerkingen

## Wat zijn uw grootste frustraties met de website?

11 antwoorden



Figuur 15: Resultaten enquête: Wat zijn uw grootste frustraties met de site?

### Welke functionaliteit(en) mist de huidige website?

14 antwoorden

Dat bepaalde projecten gegroepeerd worden en dat je kan filteren tussen deze groepen
Overzicht
Manier om te communiceren met andere studenten die een groepje zoeken
Preview van voorstellen
beter overzicht, historiek van wijzigingen aan voorstellen, zoekmogelijkheden
Een manier om te communiceren met groepsleden/projectbegeleiders
Projectvoorstel traject. Buiten ingediend en goedgekeurd de stappen er tussen en erna weergegeven.
Projecten vergelijken in 1 view
overzichtelijkheid, filters
Een makkelijke manier om zelf een voorstel in te dienen.
Een like verf
Archief van oude of afgekeurde projecten

Figuur 16: Resultaten enquête: Welke functionaliteiten ontbreken?

### Waarom maak je gebruik van het platform? Voor welke toepassingen?

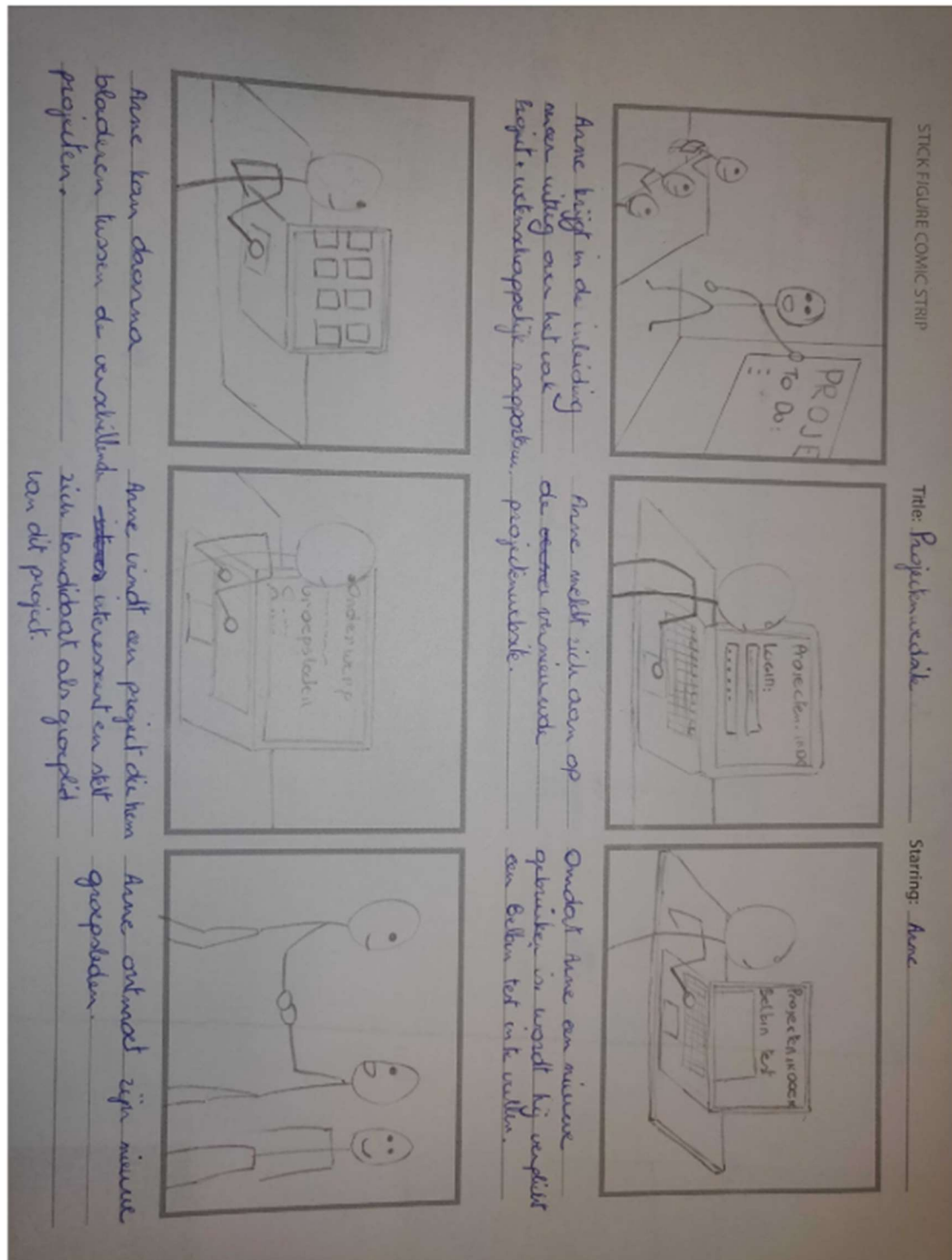
13 antwoorden

omruk voor een projectvoorstel
Fileserver
Projecten
Voor het school
Inschrijven project
Het opgeven van een projectvoorstel, bekijken van voorgestelde projecten door docenten
raadplegen projectvoorstellen
Om een project te kiezen/voorstellen 😊
Een voorstel voor project 1
Niet meer (en het is 'platform')
Projecten 1 & 2
Om een project te kiezen voor de opleiding
Om te kijken welke projecten andere studenten hebben

Figuur 17: Resultaten enquête: Waarom gebruikt u het platform?



## Bijlage 4: Storyboard



*Figuur 18: Storyboard: Eerste login van student?*

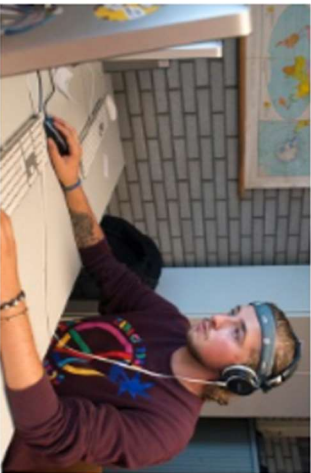
Bijlage 5: Persona's



Figuur 19: Persona docent

Bijlage 6: Wireframes

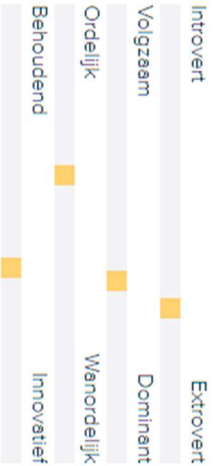
Arne De Ceuninck



"Strive for progress, not perfection."

Leeftijd: 20  
Beroep: Student ICT  
Woonplaats: Gent, België  
Hobby's: Frisbee, kajak, KSA

Persoonlijkheid



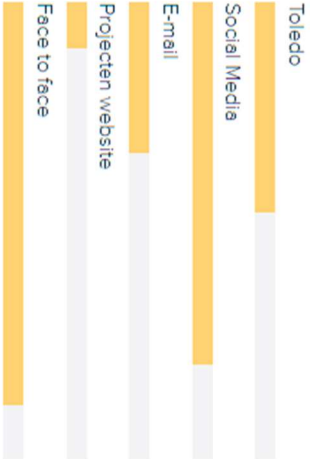
Doelstellingen

- Alle deadlines halen
- Productief in team werken
- Een afgewerkt project neerzetten
- Bijleren over de vakken heen

Frustraties

- Mensen die niet duidelijk communiceren.
- Ongeïnteresseerde mensen
- Mensen die de afspraken niet na komen.
- Trage websites

Communicatie kanalen



Beschrijving

Arne De Ceuninck is een student ICT aan Odisee hogeschool. Hij zit op kot in Gent en woont thuis, bij zijn ouders in Aalter. Hij heeft 2 jongere zussen en een oudere broer.

In het weekend is Arne zeer actief in het verenigingsleven in Aalter. Zo is hij hoofdleder bij de plaatselijke KSA en organiseert hij regelmatig evenementjes.

Hij heeft als afstudeer richting web & mobile gekozen. Hij zit in zijn 2de jaar en moet dit jaar een project doen met drie andere studenten.

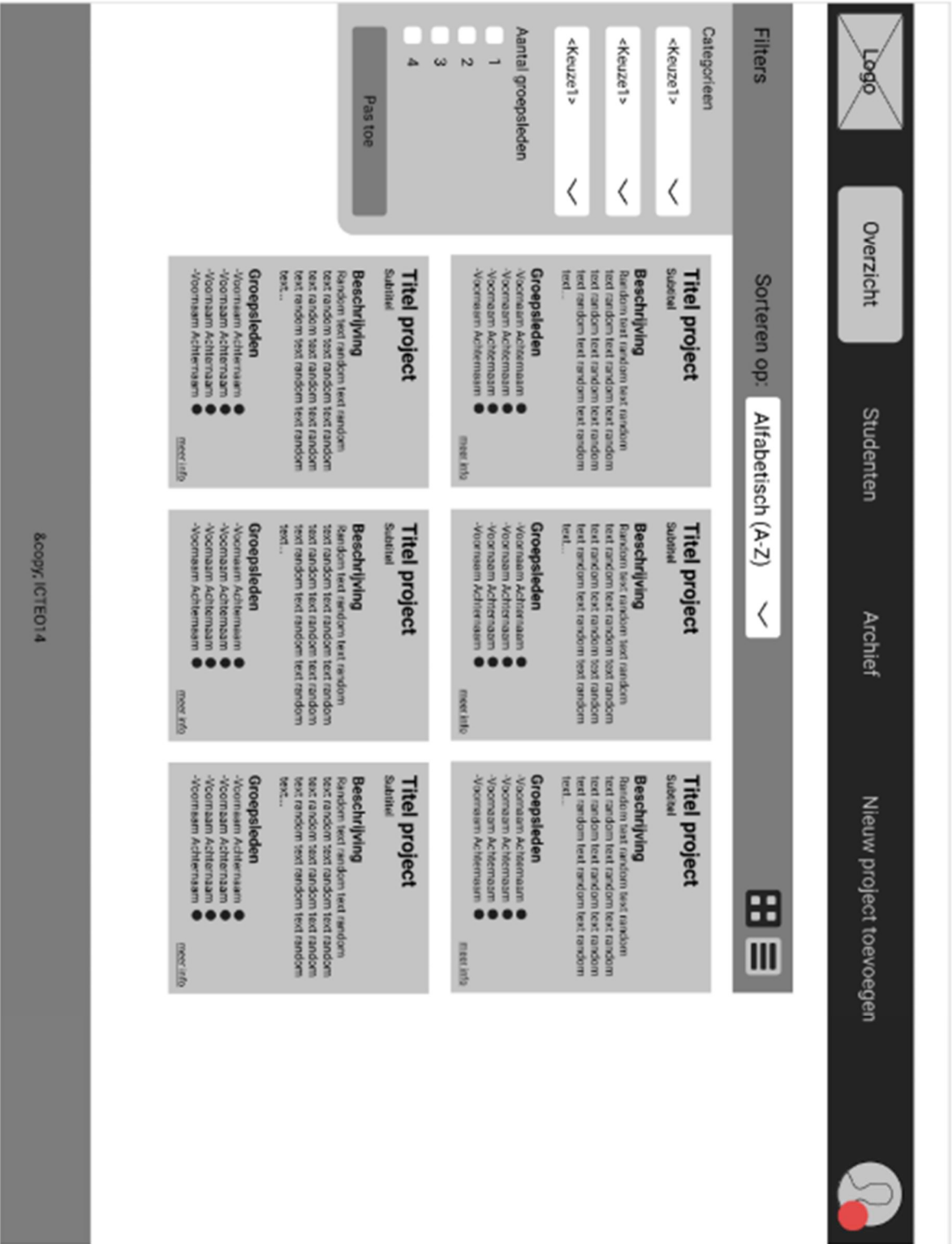
Hij heeft een groepje gevormd met twee bevriende studiegenoten, maar is nog op zoek naar een 4de. De zoektocht verloopt niet zo vlot.

Bij het zoeken naar het onderwerp zou Arne het interessant vinden om eens in het archief van oude projecten te kunnen snuisteren.

Ook zou hij graag zijn project openbaar stellen. Op die manier zouden geïnteresseerde studenten zich kunnen aanbieden om de project groep te vervollledigen.

Arne is zeer geïngageerd en heeft zin om een puik project neer te zetten

Figuur 20: Persona student



Figuur 21: Wireframe overzichtspagina



 [Overzicht](#) [Studenten](#) [Archief](#) [Nieuw project toevoegen](#) [Uitloggen](#)

Titel

Beschrijving

Hoofdvraag

Nevenvragen

SMART-criteria

☐ Specifiek

☐ Meetbaar

☐ Acceptabel

☐ Realiseerbaar

☐ Tijdsgebonden

Korte opsomming

Besproken met docent

☒ Ja

Kies een docent... ▾

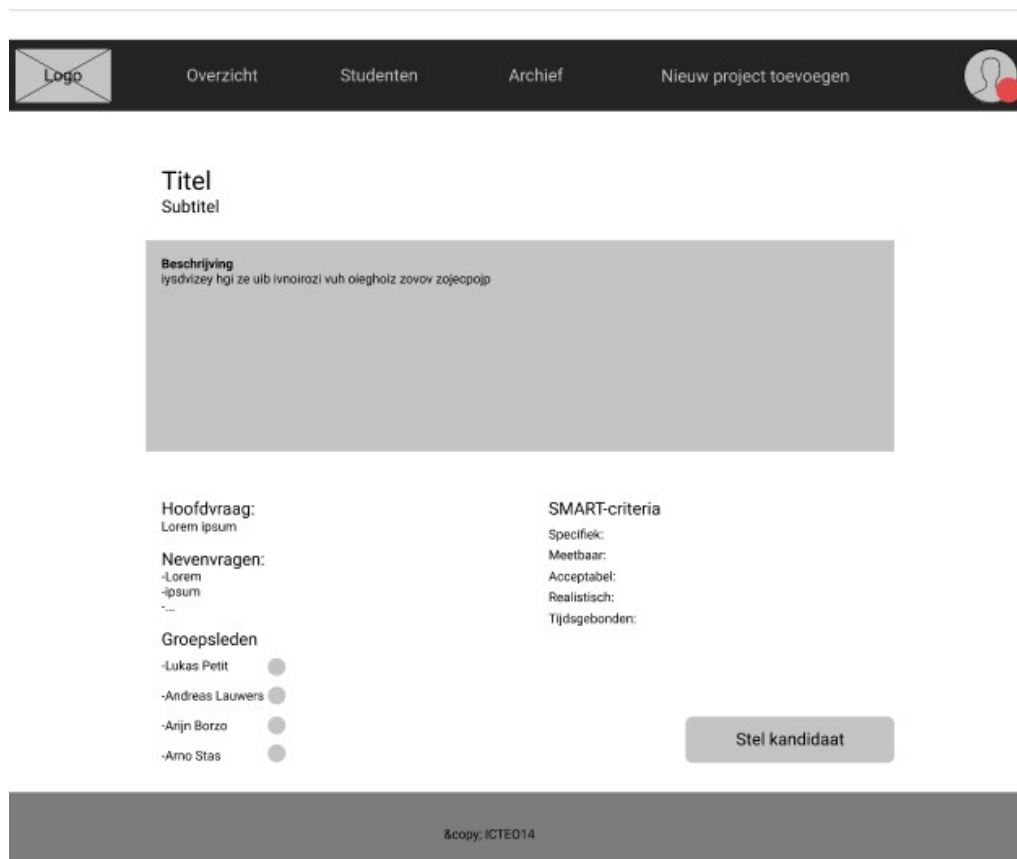
☐ Nee

Voeg project toe

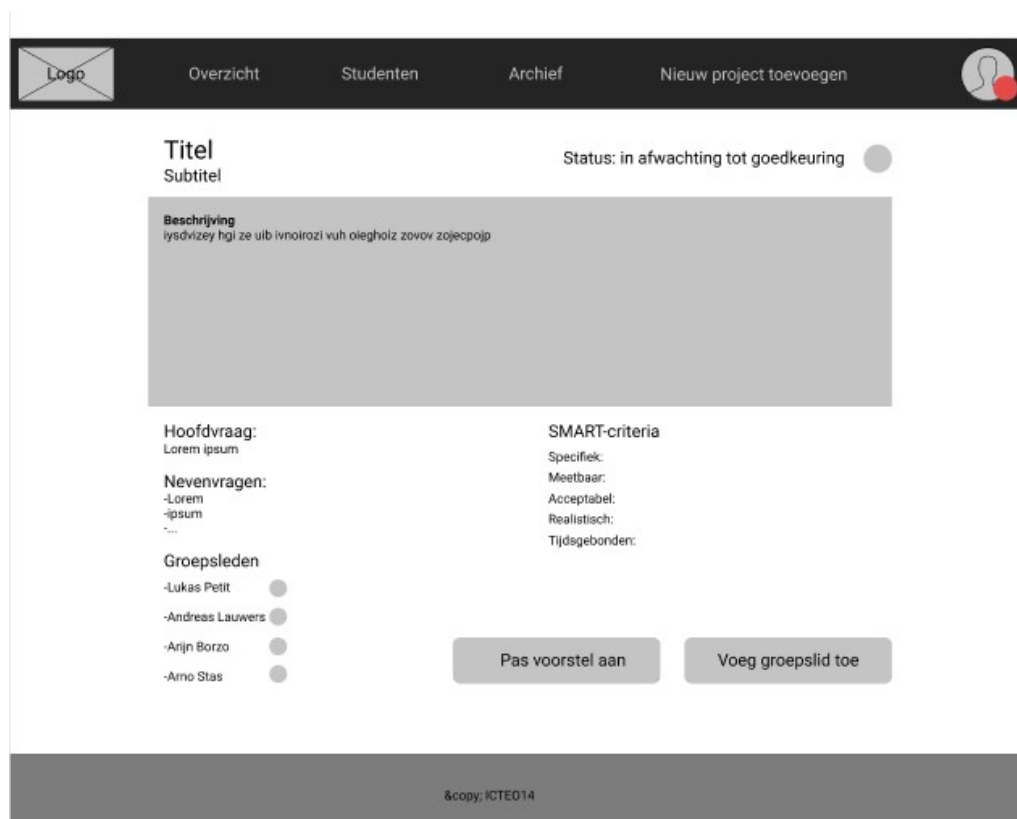
Bewaar als concept

ICTE014

Figuur 23: Wireframe nieuw projectvoorstel



Figuur 24: Wireframe detailpagina gewone gebruiker



Figuur 25: Wireframe detailpagina projectleider



*Figuur 26: Wireframe pagina Belbin test*