

**Projectenwebsite**

**Arijn Borzo**

**Arno Stas**

**Andreas Lauwers**

**Lukas Petit**

|  |
| --- |
| **Projecten 1**  **Technische mentor: Pieter Van Peteghem**  **Taalmentor: Sabine Martens** |
| **Academiejaar 2018-2019** |
| **Professionele Bachelor ICT** |

Inhoud

[Codefragmentenlijst 3](#_Toc8774018)

[Figurenlijst 4](#_Toc8774019)

[Tabellenlijst 5](#_Toc8774020)

[Afkortingenlijst 6](#_Toc8774021)

[Begrippenlijst 7](#_Toc8774022)

[Inleiding 8](#_Toc8774023)

[1. Userexperience 9](#_Toc8774024)

[1.1 Verslag interviews 9](#_Toc8774025)

[1.2 Persona’s 10](#_Toc8774026)

[1.3 Storyboard 11](#_Toc8774027)

[1.4 Wireframe 11](#_Toc8774028)

[2 Laravel 12](#_Toc8774029)

[2.1 Framework 12](#_Toc8774030)

[2.2 Databanktype 12](#_Toc8774031)

[2.3 Databankmodel 12](#_Toc8774032)

[2.4 Migrations en seeders 14](#_Toc8774033)

[2.5 Eloquente modellen 16](#_Toc8774034)

[2.6 Controllers 17](#_Toc8774035)

[2.7 Blade 17](#_Toc8774036)

[2.8 Routes 18](#_Toc8774037)

[2.9 Authenticatie 19](#_Toc8774038)

[3 Frontend 20](#_Toc8774039)

[3.1 Vue 20](#_Toc8774040)

[3.2 CSS 20](#_Toc8774041)

[4 Functionaliteiten 22](#_Toc8774042)

[4.1 Gebruikers aanmaken 22](#_Toc8774043)

[4.2 Integratie van de Belbin test 22](#_Toc8774044)

[4.3 Project toevoegen 22](#_Toc8774045)

[4.4 Goedkeuren project 22](#_Toc8774046)

[4.5 Klassensysteem 22](#_Toc8774047)

[5 Planning 23](#_Toc8774048)

[Conclusie 24](#_Toc8774049)

[Literatuurlijst 25](#_Toc8774050)

[Bijlagenoverzicht 26](#_Toc8774051)

[Bijlage 1: E-mail meneer Sanders 27](#_Toc8774052)

[Bijlage 2: Interview mevrouw Martens 29](#_Toc8774053)

[Bijlage 3: Enquête van studenten 30](#_Toc8774054)

[Bijlage 4: Storyboard 34](#_Toc8774055)

[Bijlage 5: Persona’s 35](#_Toc8774056)

[Bijlage 6: Wireframes 37](#_Toc8774057)

# 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 Routes 18

Codefragment 9 Voorbeeld van Vue 19

Codefragment 10 Javascript in Vue 19

Codefragment 11 CSS in Vue 20

# Figurenlijst

Figuur 1 Screenshot oude projectenwebsite 9

Figuur 2 Databankmodel projectenwebsite 13

Figuur 3 Resultaten enquête: Categorie ondervraagde persoon 29

Figuur 4 Resultaten enquête: Hebt u al gebruik gemaakt van de projectenwebsite? 29

Figuur 5 Resultaten enquête: Hoe hebt u uw projectgroep gevonden? 29

Figuur 6 Resultaten enquête: Hoe was uw ervaring? 30

Figuur 7 Resultaten enquête: Wat vindt u van de grafische opmaak? 30

Figuur 8 Resultaten enquête: Moeten projecten gecategoriseerd kunnen worden? 30

Figuur 9 Resultaten enquête: Hoe moet de groepsvorming gebeuren? 31

Figuur 10 Resultaten enquête: Overige opmerkingen 31

Figuur 11 Resultaten enquête: Wat zijn uw grootste frustraties met de site? 31

Figuur 12 Resultaten enquête: Welke functionaliteiten ontbreken? 32

Figuur 13 Resultaten enquête: Waarom gebruikt u het platform? 32

Figuur 14 Storyboard: De eerste login van student? 33

Figuur 15 Persona docent 34

Figuur 16 Persona student 35

Figuur 17 Wireframe overzichtspagina 36

Figuur 18 Wireframe archiefpagina 37

Figuur 19 Wireframe nieuw projectvoorstel 38

Figuur 20 Wireframe detailpagina gewone gebruiker 39

Figuur 21 Wireframe detailpagina projectleider 40

Figuur 22 Wireframe Belbin test 40

# Tabellenlijst

# Afkortingenlijst

CSS Cascading Style Sheets

HTML HyperText Markup Language

PHP Hypertext Preprocessor

MySQL

# Begrippenlijst

Stickfigures Personen getekend met lijnen

Framework Een softwarepakket waarin verschillende componenten automatisch geladen worden. Hierdoor moet de ontwikkelaar niet zelf verschillende pakketten aanmaken.

Inserts Commando waarmee gegevens in een tabel van een databank worden  
 toegevoegd.

# 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 forum 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.

# 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.

Figuur 1: Screenshot oude projectenwebsite

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. De overige knelpunten waar beide docenten een uiteenlopende mening hebben, zijn inspiratie voor de vragenlijst van de studenten (Bijlage 3).

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.

## 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.

## 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.

## 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 frontenddessigners 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 alles 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.

# 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]

## 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).

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.

## 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 opslaggrootte 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.

## 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 meedere 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 meedere rijen uit een andere tabel;
* veel-op-veelrelaties: meedere 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

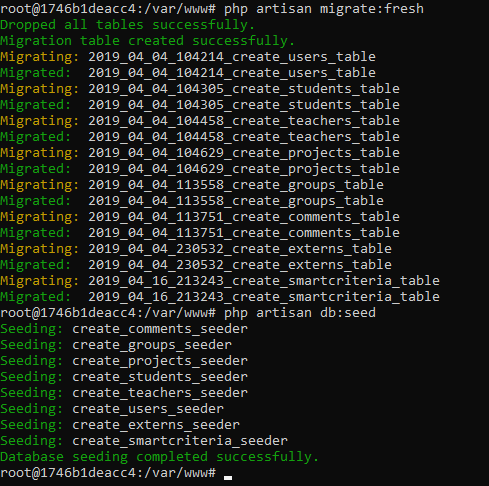
Het databankmodel van de projectenwebsite bestaat uit acht tabellen. Elk is minstens verbonden met één of meerdere tabellen:

* 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.

## 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.



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).

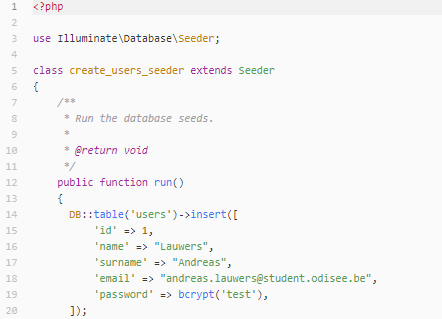
De upfunctie wordt opgeroepen bij het uitvoeren van het commando ‘php artisan migrate’. Hierin wordt beschreven hoe de tabel zal heten en welke kolommen er zijn. Bij iedere kolom wordt ook het type van de kolom vermeld en of het veld van de kolom verplicht moet ingevuld zijn.

De downfunctie wordt gebruikt bij het verwijderen van bestaande migrations. Hierbij zal de tabel automatisch verwijderd worden, indien ze nog zou bestaat. [2]



Codefragment 3: Migrationbestand gebruikers

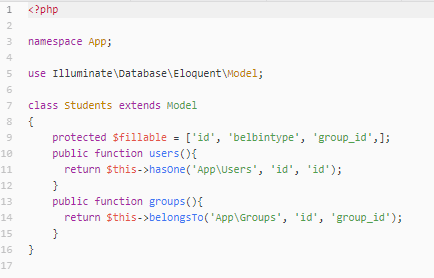
Door de seederbestanden worden de tabellen ingevuld (codefragment 4). Op deze manier kan de databank snel voorzien worden van bruikbare data en verliest de webdeveloper geen tijd. Elke seeder bestaat uit een functie ‘run’. Deze bestaat uit een reeks inserts[[1]](#footnote-1). Bij elke insert wordt zowel de tabelnaam vermeld, alsook de kolommen met bijhorende data. [3]



Codefragment 4: Seederbestand gebruikers

## Eloquente 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.

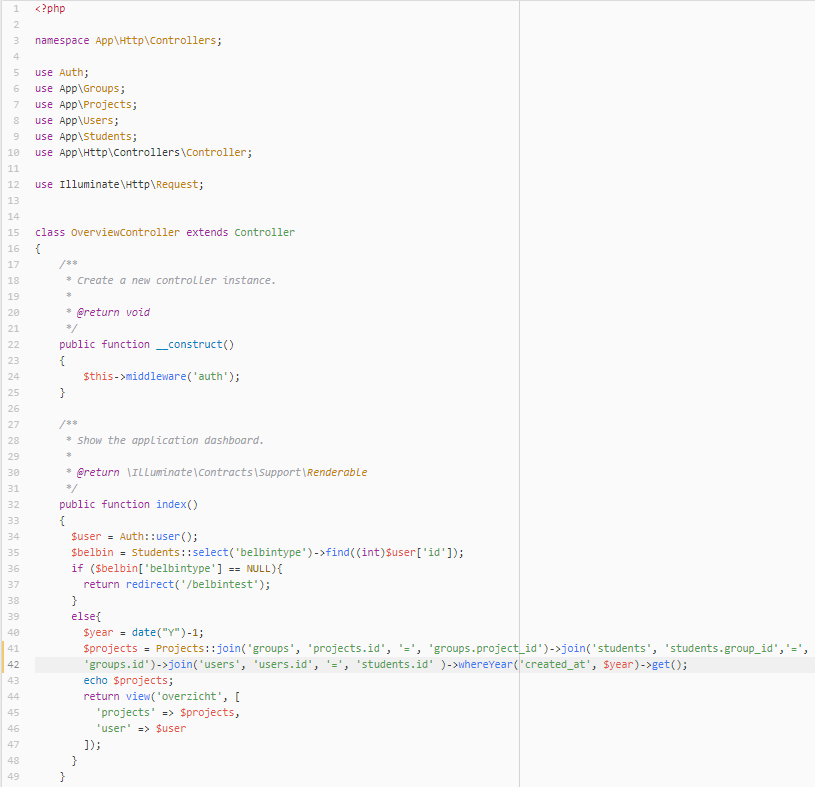
Eloquente modellen zijn telkens op eenzelfde manier opgebouwd. Bovenaan staan steeds verwijzingen naar geïncludeerde bestanden. Deze worden door Laravel automatisch gegenereerd. Vervolgens wordt een klasse gedefinieerd die overgeërfd wordt van een basis Laravelmodel. Bovenaan in deze klasse wordt een 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. [4]

Codefragment 5: Eloquent model student

## 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’[[2]](#footnote-2) of een ‘return view’[[3]](#footnote-3). [5]



Codefragment 6: Controller van overzichten

## Blade

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

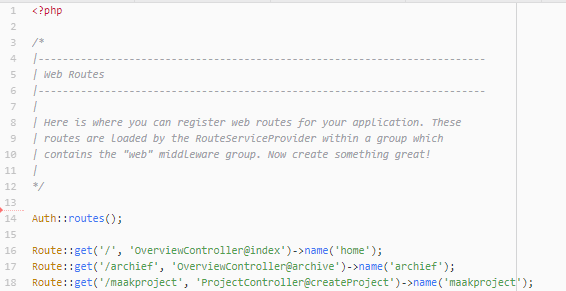
Blade is de template engine van Laravel. In deze templates wordt HTML-code gegenereerd op basis van meegegeven parameters. De engine zet deze parameters om in codefragmenten om zo de website te vervolledigen. Zo is het bijvoorbeeld mogelijk om verschillende functies in een Bladedocument te plaatsen. 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). [6]



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

## 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.



Codefragment 8: Routes

Eén van Laravels belangrijkste hulpmiddelen zijn de routebestanden (codefragment 8). Hierin wordt bepaald welke functie uit een controller wordt opgeroepen als er naar een bepaalde URL wordt gesurft. [7]

## 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. [8]

# BootstrapVue

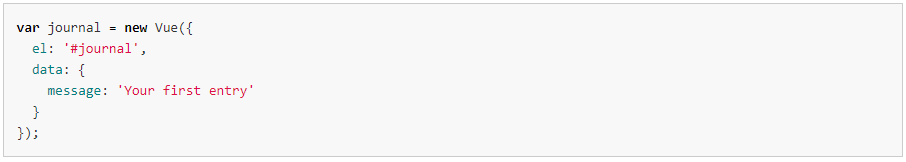
## Framework Vue

Vue is een javascriptframework voor het bouwen van gebruikersinterfaces. In tegenstelling tot andere monolithische frameworks, is Vue vanaf de basis ontworpen om incrementeel toepasbaar te zijn. De kern bibliotheek is alleen 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.

Wat het team vooral naar Vue.js trok is dat Javascript eenvoudig kan worden geschreven. Aan de slag gaan met Vue.js is zeer makkelijk. De broncode is zeer 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. [9]

Om een voorbeeld te geven van wat er bedoeld wordt:

Codefragment 9: Voorbeeld van Vue

Voor iemand die al een paar Javascript bibliotheken heeft gezien, is het niet moeilijk om alles in dit voorbeeld te begrijpen zonder enige documentatie. En meestal met andere frameworks is dit waar de eenvoud ophoudt.

Codefragment 10: Javascript in Vue

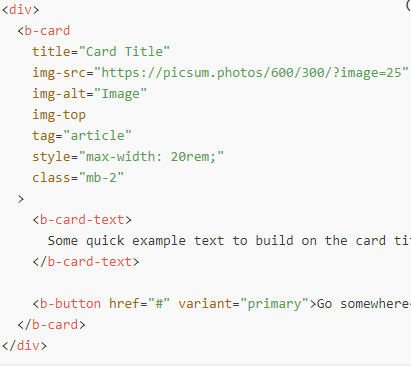
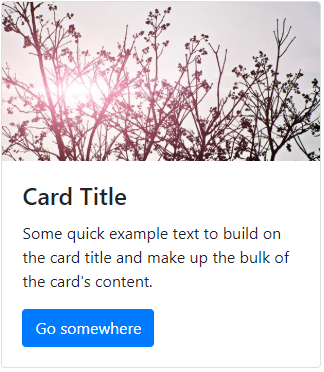
## Componenten

De meeste frontendframeworken zijn zo makkelijk omdat de applicatie bestaat uit componenten. Elk belangrijk stukje van de website dat niet kleiner genomen kan worden wordt beschreven in een component. Elk component heeft dus zijn eigen HTML, Javascript en CSS indien nodig, in één pagina.

De bestaat de website uit één groot component die op zichzelf bestaat uit meerdere componenten die weer uit andere componenten kunnen bestaan. Elk component kan dan een parent en child component bevatten. Parent A is het component waarin het huidige component B zich in bevind en B is dan het child component 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.

## BootstrapVue

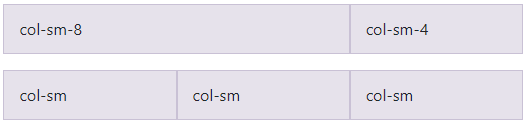
BootstrapVue is het framework waarmee te werk is gegaan, het is exact hetzelfde als Vue maar Bootstrap (toolkit om HTML, CSS, Javascript makkelijker te implementeren) wordt hier mee in geïntegreerd. Dit maakt het mogelijk zo weinig mogelijk code te moeten schrijven voor wat men wil bereiken. Bootstrap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

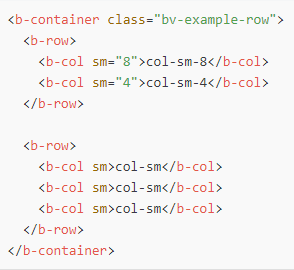
  
Codefragment 11: Card in Vue

Dit wordt allemaal duidelijk in het voorbeeld hierboven, waar een “card” voorgesteld wordt.  
Het begint met de tag ´b-card´. Dit is een voorgeschreven onderdeel van BootstrapVue. Men moet enkel nog in de tag, de opties zoals de border, afbeeldingen, stijl… die men wilt wijzigen, aanpassen of invoegen veranderen. De voorgemaakte klassen kan men ook terugvinden in de optie class.  
Normaal is een klasse, een stijl die zal toepassen op meerdere elementen. Bootstrap heeft voorgemaakte klassen, waardoor de meest voorkomende stijlen enkel met enkele letters die men toevoegt aan de klassen van het object. In dit voorbeeld is ‘mb-2’ gelijk aan ‘margin-bottom = 2rem;’

## HTML

Elk component heeft dus zijn eigen HTML-, Javascript- en CSS-gedeelte. HTML wordt geschreven in een <template> tag, Vue heeft een ingebouwd grid en layout systeem wat zeer overzichtelijk maakt om alles mooi te plaatsen waar men het wil. Alsook om de site responsive te maken.





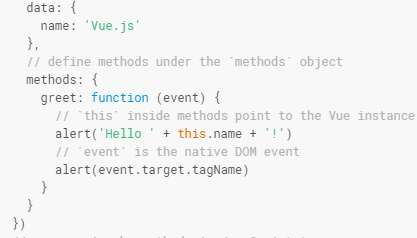
Codefragment 12: CSS in Vue

De pagina wordt bekeken als een tabel van rijen en kolommen en elke rij, kolom kan je aanpassen naar eigen wil. De wijdte, hoogte, responsive maken enzoverder.

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

## Javascript

In het <script> gedeelte wordt de Javascript code geschreven, de functionaliteit. Hier wordt de data beschreven en bewerkt.



Codefragment 12: CSS in Vue

In het script gedeelte kan alles meegegeven, gedeclareerd en bewerkt worden. De parent component kan “props” doorgeven, dit zijn de data die kan doorgegeven worden. Daarnaast zijn uiteraard nog de huidige data aanwezig. Deze kunnen bewerkt worden in het gedeelte van de methoden. Er zijn ook lifecycle methoden, deze worden altijd, stuk voor stuk, opgeroepen bij elk deel van het laadproces van een component.

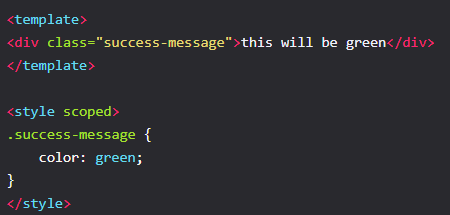
Simpel voorbeeld: er worden “props” doorgegeven van de parent component, zijn deze nog niet beschikbaar als de website ingeladen wordt. Enkel als de pagina met het huidige component geladen wordt. Dus enkel en alleen dan kunnen deze data bewerkt worden. Één van de vele lifecycle methoden houdt in dat deze pas opgeroepen wordt nadat het component ingeladen is, nl. mounted. In de methode kunnen dus de “props” bewerkt worden aangezien men daar ze al doorgegeven zijn en dus het child component bezit heeft er van.

## CSS

Voor traditionele websites zijn er veel manieren waarop CSS kan worden geïmplementeerd. De meest voorkomende manier is om de stijlen in een eigen CSS-bestand te zetten.

Er is meestal een globaal CSS-bestand en dan één CSS-bestand per pagina. 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.

In Vue kan elk onderdeel zijn eigen styling hebben die op het onderdeel is gescreend.



Codefragment 13: CSS in Vue

Let op het scoped attribuut. Zonder dit zou de CSS globaal worden toegepast. Iets dat liefst vermeden wordt. Dus als er hierna nog een component wordt gecreëerd met een div-element die gebruik maakt van het succes bericht van de klasse, zal dat element niet groen worden.

## Besluit

Wat er in Vue dan vaak wordt tegengekomen is een heleboel kleine componenten, met elk hun tekst en functionaliteit. Waarbij slechts weinig tot geen CSS bij hoort. Samen met een CSS framework is dit geëindigd met eenvoudige en kleine CSS!

Vue voorkomt dat je constant moet wisselen tussen JavaScript-, HTML- en CSS-bestanden omdat alles in één bestand zit. Wat het zeer overzichtelijk maakt en makkelijk om alles apart te behandelen.

# Functionaliteiten

## 4.1 Gebruikers aanmaken

Uitleggen hoe dit gebeurt.

## Integratie van de Belbin test

Zeggen hoe de belbintesten werken en welke invloed dat deze heeft.

## Project toevoegen

Zeggen wie een project kan toevoegen en wat er allemaal me kan gebeuren.

## Goedkeuren project

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.

## Klassensysteem

Uitleg over verschil tussen de functionaliteiten tussen leerkrachten, leerlingen en externen.

# Planning

In het project zijn er 4 grote stukken, 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 | **User Experience** | Groep |
| 14 maart 2019 | Vergadering #3 en Afspraak #2 begeleider Brainstorm design en database-ontwerp, feedback wireframe | Groep |
| 21 maart 2019 | **HTML** | Groep |
| 28 maart 2019 | **CSS** | Groep |
| 4 april 2019 | Afspraak#3 begeleider | Groep |
| 22 april 2019 | **Laravel, MySQL, Blade** | Andreas en Lukas |
| 25 april 2019 | Vergadering #4 en  Feedback Laravel, MySQL, Blade | Groep |
| 25 april 2019 | Afspraak #4 begeleider | Groep |
| 2 mei 2019 | **Javascript & Jquery** | Arijn en Arno |
| 8 mei 2019 | Vergadering #5 Feedback Javascript & JQuery | Groep |
| 9 mei 2019 | Afspraak #5 begeleider | Groep |
| 15 mei 2019 | **Extra’s** | Groep |
| 16 mei 2019 | Vergadering #6 Overlopen eindproduct, documentatie afwerken. | Groep |
| 22 mei 2019 | Eindproduct indienen | Groep |

# Conclusie

Wat we mogen concluderen na al het werk dat we verricht hebben over onszelf, het project, de samenwerking en de site in het algemeen.

# 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/authenticatie. [Accessed 5 mei 2019]. |
| [9] | Vue.js, “Introduction,” Vue Documentatie, [Online]. Available: https://vuejs.org/v2/guide. [Accessed 5 mei 2019]. |
| [10] | S. Sanders, *sven.sanders@odisee.be, Vragen omtrent project projectenwebsite,* 24 februari 2019. |

# Bijlagenoverzicht

Bijlage 1: E-mail meneer Sanders

Bijlage 2: Interview mevrouw Martens

Bijlage 3: Enquête van studenten

Bijlage 4: Storyboard

Bijlage 5: Persona’s

Bijlage 6: Wireframes

## 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 planningen, opvolgingsdocumenten...

Als ik daar zelf een paar minimumeisen 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 allerhande 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.



Figuur 3: Resultaten enquête: Categorie ondervraagde persoon



Figuur 4: Resultaten enquête: Hebt u al gebruik gemaakt van de projectenwebsite?



Figuur 5: Resultaten enquête: Hoe hebt u uw projectgroep gevonden?



Figuur 6: Resultaten enquête: Hoe was uw ervaring?



Figuur 7: Resultaten enquête: Wat vindt u van de grafische opmaak?



Figuur 8: Resultaten enquête: Moeten projecten gecategoriseerd kunnen worden?



Figuur 9: Resultaten enquête: Hoe moet groepsvorming gebeuren?



Figuur 10: Resultaten enquête: Overige opmerkingen



Figuur 11: Resultaten enquête: Wat zijn uw grootste frustraties met de site?



Figuur 12: Resultaten enquête: Welke functionaliteiten ontbreken?



Figuur 13: Resultaten enquête: Waarom gebruikt u het platform?

## Bijlage 4: Storyboard



Figuur 14: Storyboard: Eerste login van student?

## Bijlage 5: Persona’s



Figuur 15: Persona docent



Figuur 16: Persona student

## Bijlage 6: Wireframes



Figuur 17: Wireframe overzichtspagina



Figuur 18: Wireframe archiefpagina



Figuur 19: Wireframe nieuw projectvoorstel



Figuur 20: Wireframe detailpagina gewone gebruiker



Figuur 21: Wireframe detailpagina projectleider



Figuur 22: Wireframe pagina Belbin test

1. Commando waarmee gegevens in een tabel van een databank worden toegevoegd. [2] [↑](#footnote-ref-1)
2. Deze coderegel zorgt ervoor dat de browser wordt omgeleid naar een andere webpagina op een ander webadres. [5] [↑](#footnote-ref-2)
3. Dit codeblok voorziet dat er een webpagina wordt geladen met de correct meegegeven parameters. [5] [↑](#footnote-ref-3)