



Professionele Bachelor Toegepaste Informatica



Ideashooter: beheren van ideeën

Jonas Kempeneers

Promotoren:

Pieter Quinart
Tim Dupont

Forwrd
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019



Professionele Bachelor Toegepaste Informatica



Ideashooter: beheren van ideeën

Jonas Kempeneers

Promotoren:

Pieter Quinart
Tim Dupont

Forwrd
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2018-2019

Dankwoord

Ik heb deze scriptie geschreven als afstuderende van de opleiding Toegepaste Informatica aan de Hogeschool PXL te Hasselt.

Als eerste wil ik graag PXL Smart-ICT en vooral Niek Vandael en Tim Dupont, mijn hogeschoolpromotor, bedanken voor de begeleiding tijdens deze stage. Ik kon steeds bij hen terecht voor vragen of constructieve feedback. Daarnaast hebben ze mij de nodige ruimte gegeven om de stageopdracht deels zelf in te vullen.

Forwrd en Jef Martens, de klant, mag ik zeker niet vergeten. Zonder hen had ik niet aan dit leuke project kunnen meewerken. De samenwerking is heel vlot en fijn verlopen wat het project alleen maar ten goede is gekomen.

Nogmaals wil ik iedereen die heeft meegewerkt aan dit project en stageopdracht bedanken, jullie hebben ervoor gezorgd dat is een zeer interessante en leerzame stage is geworden.

Abstract

Binnen een bedrijf is het moeilijk om een goede communicatie te realiseren tussen de werknemers en het management. Doordat ze elkaar weinig zien kan de drempel te hoog liggen om over de problemen op de werkvloer te praten. Ideashooter zorgt voor een vlottere communicatie tussen werknemers en management vlotter verloopt. Dat doet het door werknemers de mogelijkheid te geven om ideeën in te sturen. Deze ideeën gaan van eenvoudige vragen over de temperatuur in werkhallen tot fundamentele processen die misschien niet zo goed lopen als mogelijk. De werkgever heeft op zijn beurt de mogelijkheid om wedstrijden met onderliggende vragen op te stellen waaraan al dan niet een prijs wordt gekoppeld. Werknemers kunnen op wedstrijdvragen antwoorden en komen daardoor in aanmerking om de gekoppelde prijs te winnen.

De kern van deze applicatie is het evalueren oftewel ‘scopen’ van de ingestuurde antwoorden of ideeën. De antwoorden kunnen geëvalueerd worden aan de hand van vooropgestelde criteria. Wanneer de antwoorden geëvalueerd zijn, krijgen ze een score die laat zien of het gegeven antwoord al dan niet goed is. Voor een goede vraag of een goed idee kan een projectgroep opgestart worden waaraan werknemers gekoppeld zijn. In zo’n projectgroep wordt geprobeerd het idee te verwezenlijken.

De stageopdracht is uitgeschreven door Forwrd, maar de uitvoering vindt plaats op PXL Research. Forwrd specialiseert zich in het maken van webapplicaties, IoT en software voor interactieve ‘photoboosts’. Het maken van Ideashooter situeert zich in het eerste aspect, namelijk het maken van webapplicaties.

De webapplicatie is geschreven in Laravel, een PHP-framework, in samenwerking met Vue.js, een JavaScript framework voor *userinterfaces* en *single-page applications*. In Laravel zit de *templating* taal Blade die gebruikt wordt om de views dynamischer te maken met bijvoorbeeld *foreach* lussen. Ondanks Blade zijn de views niet altijd dynamisch genoeg, wat een goede reden is om Vue.js te gebruiken.

Ideashooter is een betaalde applicatie voor bedrijven. Hierdoor is het nuttig om automatische betalingen te voorzien zodat de klant zich niet moet bezighouden met de betalingen zelf. Om te bepalen welke betalingsservice het best past voor Ideashooter omvat dit onderzoek een exploratiefase en vergelijkingsfase van drie kandidaten die populair zijn in België en daarbuiten, namelijk Mollie, Stripe en Icpay. Deze drie kandidaten zijn gekozen aan de hand van een bevraging naar ervaringen binnen het team. De exploratiefase is een literatuurstudie van elke optie, de kosten van elke transactie en de betaalmogelijkheden. De vergelijkingsfase bevat een proof of concept van elke geschikte kandidaat ter illustratie van de implementatie.

Inhoudsopgave

| | |
|---|-----|
| Dankwoord | ii |
| Abstract | iii |
| Lijst van gebruikte figuren..... | vii |
| Lijst van gebruikte tabellen | ix |
| Lijst van gebruikte afkortingen..... | x |
| Inleiding..... | 1 |
| I. Stageverslag..... | 2 |
| 1 Bedrijfsvoorstelling | 2 |
| 2 Stagevoorstelling..... | 4 |
| 2.1 Probleemstelling..... | 4 |
| 2.2 Oplossing..... | 4 |
| 2.2.1 Doelstellingen..... | 4 |
| 2.3 Omgeving..... | 7 |
| 3 Uitwerking stageopdracht | 9 |
| 3.1 Analyse..... | 9 |
| 3.2 Ontwikkeling | 11 |
| 3.2.1 Setup | 11 |
| 3.2.1.1 Project genereren en hosten op webserver..... | 11 |
| 3.2.1.2 Database..... | 12 |
| 3.2.1.3 Importeren HTML en CSS in Laravel | 12 |
| 3.2.2 Authenticatie | 13 |
| 3.2.2.1 Rollen | 13 |
| 3.2.2.2 Autorisatie..... | 15 |
| 3.2.3 Wedstrijden..... | 15 |
| 3.2.4 Scoping..... | 17 |
| 3.2.4.1 Scoping criteria aanpassen..... | 17 |
| 3.2.4.2 Antwoorden evalueren | 20 |
| 3.2.5 Projectgroepen | 22 |
| 3.2.5.1 Business case | 26 |
| 4 Reflectie..... | 27 |
| II. Onderzoekstopic | 28 |
| 1 Onderzoeksvraag | 28 |
| 2 Onderzoeksmethode..... | 29 |

| | | |
|---------|--|----|
| 3 | Uitwerking onderzoek | 29 |
| 3.1 | Inleiding | 29 |
| 3.1.1 | Wat houdt een automatische betaling in?..... | 29 |
| 3.2 | Exploratie van de drie kandidaten..... | 29 |
| 3.2.1 | Mollie | 29 |
| 3.2.1.1 | Betalingsmogelijkheden | 30 |
| 3.2.1.2 | Kostprijs..... | 31 |
| 3.2.1.3 | Minimum- en maximumbedragen..... | 31 |
| 3.2.1.4 | Support..... | 32 |
| 3.2.1.5 | Betrouwbaarheid | 32 |
| 3.2.2 | Stripe | 32 |
| 3.2.2.1 | Betalingsmogelijkheden | 33 |
| 3.2.2.2 | Kostprijs..... | 34 |
| 3.2.2.3 | Minimum- en maximumbedragen..... | 34 |
| 3.2.2.4 | Support..... | 34 |
| 3.2.2.5 | Betrouwbaarheid | 34 |
| 3.2.3 | Icepay | 35 |
| 3.2.3.1 | Betaalmogelijkheden | 35 |
| 3.2.3.2 | Kostprijs..... | 35 |
| 3.2.3.3 | Minimum- en maximumbedragen..... | 36 |
| 3.2.3.4 | Support..... | 36 |
| 3.2.3.5 | Betrouwbaarheid | 36 |
| 3.3 | Vergelijkingfase..... | 36 |
| 3.3.1 | Prijsvergelijking..... | 36 |
| 3.3.1.1 | Conclusie prijsvergelijking..... | 37 |
| 3.3.2 | Vergelijking van support | 38 |
| 3.3.2.1 | Conclusie vergelijking support..... | 38 |
| 3.3.3 | Vergelijking betrouwbaarheid | 38 |
| 3.3.3.1 | Conclusie betrouwbaarheid | 38 |
| 3.3.4 | Proof of concept | 38 |
| 3.3.4.1 | Mollie | 38 |
| 3.3.4.2 | Stripe | 44 |
| 3.3.4.3 | Vergelijklingsmatrix PoC | 48 |
| 3.3.4.4 | Conclusie PoC | 48 |
| | Conclusie | 49 |
| | Bibliografie | 51 |

Lijst van gebruikte figuren

| | |
|--|----|
| Figuur 1 Logo Forwrd | 2 |
| Figuur 2 Calculus Gateway Control Unit | 3 |
| Figuur 3 MGX Interactieve videobooth..... | 3 |
| Figuur 4 Scoping: subcategorie met bijhorende criteria..... | 5 |
| Figuur 5 Bondige use-case diagram | 6 |
| Figuur 6 Vue.js component in HTML-bestand | 7 |
| Figuur 7 Staafdiagram van top vijf gebruikte databanken..... | 8 |
| Figuur 8 Architectuur van Ideashooter | 8 |
| Figuur 9 Eerste wireframes | 9 |
| Figuur 10 Scherm om wedstrijd te beantwoorden in storyboard | 10 |
| Figuur 11 Prototype van scherm 'wedstrijdvraag beantwoorden' | 10 |
| Figuur 12 Nodige extensies om Laravel te hosten..... | 11 |
| Figuur 13 Methode voor het aanmaken van tabel | 12 |
| Figuur 14 Toevoegen van component met behulp van Blade..... | 12 |
| Figuur 15 Foreach lus met Blade en Vue.js component | 12 |
| Figuur 16 Rollen op profielpagina..... | 13 |
| Figuur 17 Many to many relatie in model User..... | 13 |
| Figuur 18 Many to many relatie in model Role | 13 |
| Figuur 19 Errorpagina binnen Ideashooter | 14 |
| Figuur 20 Methode die controleert of een gebruiker de correcte rollen heeft | 14 |
| Figuur 21 Controller action method die een controle doet op de rollen die de gebruiker heeft | 14 |
| Figuur 22 Auth middleware op route | 15 |
| Figuur 23 Auth middleware in constructor van een controller | 15 |
| Figuur 24 Wedstrijdvragen toevoegen of verwijderen | 15 |
| Figuur 25 For-loop in Vue.js | 16 |
| Figuur 26 Notificatieblokje van onbeantwoorde vragen | 16 |
| Figuur 27 Onbeantwoorde vraag..... | 16 |
| Figuur 28 Beantwoorde vraag | 16 |
| Figuur 29 Via methode in Notification | 17 |
| Figuur 30 Wedstrijdvragen en progressiebalk..... | 17 |
| Figuur 31 Wedstrijd met vragen en antwoorden | 18 |
| Figuur 32 Categorie marktattractiviteit met subcategorie klantbehoefte en bijhorende criteria | 18 |
| Figuur 33 Overliggend scherm om subcategorie aan te passen..... | 19 |
| Figuur 34 Overliggend scherm om criteria aan te passen..... | 19 |
| Figuur 35 Object om data over criteria tussentijds op te slaan | 20 |
| Figuur 36 Manier om variabelen uit app.js bestand op te vragen | 20 |
| Figuur 37 Overzicht openstaande wedstrijden | 20 |
| Figuur 38 Overzicht ingestuurde antwoorden en de score..... | 20 |
| Figuur 39 Keuze tussen vijf criteria binnen elke subcategorie | 21 |
| Figuur 40 Formulier om scoping af te ronden | 21 |
| Figuur 41 Overzicht ingestuurde antwoorden | 22 |
| Figuur 42 Overzicht van beantwoorde wedstrijd vragen..... | 22 |
| Figuur 43 Overzicht van antwoorden bij het opmaken van een projectgroep | 23 |
| Figuur 44 Overzicht van alle gebruikers die een antwoord ingezonden hebben | 23 |
| Figuur 45 Overzicht van geselecteerde antwoorden en groepsleden op detailpagina van projectgroep | 24 |

| | | |
|-----------|---|----|
| Figuur 46 | Overzicht van de huidige status van de projectgroep | 25 |
| Figuur 47 | Overzicht van actieve projectgroepen..... | 25 |
| Figuur 48 | Tabel met verzoeken om deel te nemen aan projectgroep | 26 |
| Figuur 49 | Webpagina om business case op te stellen | 26 |
| Figuur 50 | Financieel overzicht binnen het scherm 'business case' | 27 |
| Figuur 51 | Logo Mollie | 30 |
| Figuur 52 | Logo Bancontact..... | 30 |
| Figuur 53 | Logo iDeal | 30 |
| Figuur 54 | Logo PayPal | 31 |
| Figuur 55 | Logo Stripe | 33 |
| Figuur 56 | Logo Apple Pay..... | 33 |
| Figuur 57 | Logo Google Pay..... | 33 |
| Figuur 58 | Status van alle services van Stripe..... | 35 |
| Figuur 59 | Overzicht bestellingen binnen Mollie..... | 39 |
| Figuur 60 | Overzicht betalingen binnen Mollie | 40 |
| Figuur 61 | Methode om betaling aan te maken voor Mollie | 41 |
| Figuur 62 | Activeren van betalingsmethoden in Mollie | 42 |
| Figuur 63 | Betaling met kredietkaart via Mollie | 43 |
| Figuur 64 | Betaling met kredietkaart via Mollie met testaccount..... | 44 |
| Figuur 65 | Code om informatie over betaling op te vragen bij Mollie..... | 44 |
| Figuur 66 | Overzicht van betalingen binnen Stripe | 45 |
| Figuur 67 | Methode om betaling aan te maken voor Stripe | 46 |
| Figuur 68 | Methode om in de frontend de gebruiker naar de checkout van Stripe te sturen..... | 46 |
| Figuur 69 | Inladen van script om functionaliteit van Stripe te kunnen gebruiken | 46 |
| Figuur 70 | Betalings scherm van Stripe..... | 47 |
| Figuur 71 | Methode om een plan aan te maken bij Stripe | 47 |
| Figuur 72 | Methode om terugkerende betaling op te zetten bij Stripe..... | 47 |
| Figuur 73 | Betalings scherm voor terugkerende betalingen bij Stripe | 48 |

Lijst van gebruikte tabellen

| | |
|---|----|
| Tabel 1 Overzicht minimum- en maximumbedragen bij Mollie..... | 32 |
| Tabel 2 Prijsvergelijking Mollie, Stripe en Icepay | 37 |
| Tabel 3 Vergelijking van support | 38 |
| Tabel 4 Vergelijking uptime..... | 38 |
| Tabel 5 Vergelijking PoC's | 48 |
| Tabel 6 Samenvatting van alle vergelijkingen | 49 |

Lijst van gebruikte afkortingen

| | |
|-------|---|
| FTP | File Transfer Protocol – Protocol dat uitwisselen van bestanden tussen computers mogelijk maakt [1] |
| HTTPS | HyperText Transfer Protocol Secure – Beveiligde versie van HTTP die ervoor zorgt dat data die tussen client en server geëncrypteerd verzonden kan worden. |
| JSON | JavaScript Object Notation – Methode om data te formatteren die makkelijk te lezen is voor mensen en gemakkelijk te ontleden en genereren voor computers [2]. |
| JWT | JSON Web Tokens – Gebruikt als access tokens voor authenticatie |
| NPM | Node Package Manager – Een tool om <i>node packages</i> te beheren |
| MVC | Model View Controller – Architecturaal patroon om applicaties te ontwikkelen |
| ORM | Object Relational Mapping - Techniek binnen objectgeoriënteerde programmeertalen om incompatibele data om te zetten naar data die wel compatibel is met de database [3] |
| PoC | Proof of Concept - Praktische demonstratie van het idee |
| URL | Uniform Resource Locator – Uniek adres van bijvoorbeeld een website |

Inleiding

Deze bachelorproef is opgedeeld in twee delen, het eerste deel gaat over de ontwikkeling van Ideashooter en het tweede deel is een onderzoek naar automatisch betalingen.

Forwrd heeft deze stageopdracht uitgeschreven in opdracht van Jef Martens, de klant. Hij wil graag een platform laten bouwen dat de communicatie tussen werknemers en het management vlotter doet verlopen. Op dat platform, dat de naam Ideashooter draagt, kunnen de leden van het management vragen stellen aan de werknemers in de vorm van wedstrijden waar ze al dan niet een prijs aan koppelen. Werknemers kunnen de vragen beantwoorden of op eigen initiatief een idee indienen. Vervolgens kan het management de vragen evalueren en feedback geven op het idee van de werknemers.

In het onderzoeksgedeelte worden drie providers van automatische betalingen onder de loep genomen. Welke passen het best voor een project zoals Ideashooter en kunnen geïmplementeerd worden in Laravel? Daarnaast wordt de prijs per transactie bekeken en de ondersteuning die de providers bieden.

I. Stageverslag

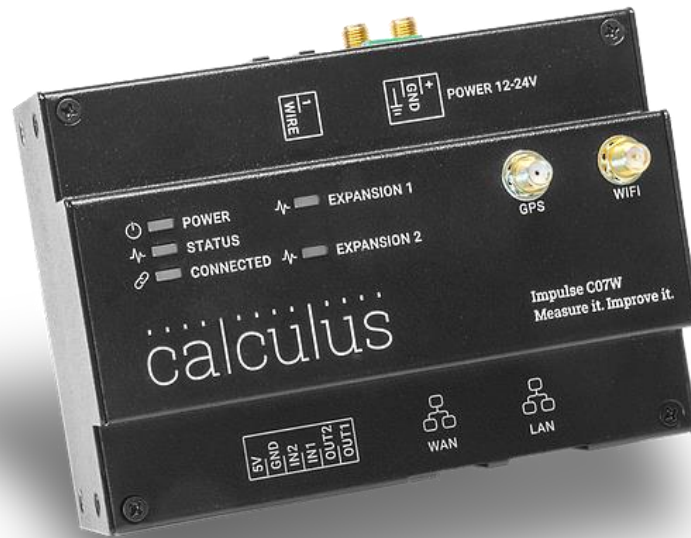
1 Bedrijfsvoorstelling

Forwrd (Figuur 1) is een bedrijf in Zutendaal dat actief is op drie vlakken namelijk webapplicaties, 'Internet of things' oftewel IoT en interactieve communicatie. Deze stageopdracht situeert zich binnen het deel van de webapplicaties.



Figuur 1 Logo Forwrd

Forwrd is ontstaan vanuit het maken van websites in bijberoep. Na een tijdje leerden ze meerdere mensen kennen, en zagen ze een job waarbij ze hun eigen baas zijn erg zitten. Doordat het netwerk van Forwrd steeds groter werd en er zich meer kansen voor deden is Forwrd vertakt in Calculus (IoT) en MGX, uitgesproken 'magix' (interactieve communicatie). Calculus heeft haar eigen module om big data te analyseren, namelijk de Calculus Gateway Control Unit (Figuur 2). De door de module verzamelde data kan daarna grafisch weergegeven worden in een webapplicatie. De klant kan op die manier snel ingrijpen wanneer dat nodig is. Forwrd probeert meer en meer anticiperend te werken zodat de klant eerder kan ingrijpen. Een voorbeeld hiervan is het installeren van temperatuursensoren bij koelcellen van een festival. De generatoren die de koelcellen van energie voorzien, kunnen maar een bepaalde last aan. Door de temperatuursensoren kunnen ze anticiperen op de energienood van de generatoren. Zo'n projecten doen ze in samenwerking met enkele grotere festivals uit België zoals Tomorrowland, Tomorrowland Winter en Graspop.



Figuur 2 Calculus Gateway Control Unit

De tweede vertakking van Forwrd, MGX, houdt zich bezig met interactieve *photo-* en *videobooths*. In zo'n *booth* kunnen gebruikers foto's of video's maken van zichzelf en de *booth* zorgt voor een interactieve achtergrond. Zo'n achtergrond kan bijvoorbeeld een grote dinosauriër zijn (Figuur 3). Voor deze projecten hebben ze samengewerkt met Studio City in Macau (casino), Zoo van Antwerpen en Coca-Cola. Samen met Coca-Cola gaan ze met meerdere GoPro's beelden maken van mensen die op een waterslide zitten. Daarna kunnen de verschillende videofragmenten softwarematig aan elkaar gemaakt worden zodat ze samen één groot beeld vormen.



Figuur 3 MGX Interactieve videobooth

2 Stagevoorstelling

2.1 Probleemstelling

Binnen een bedrijf is het niet altijd even gemakkelijk om een vlotte communicatie te bekomen tussen management en werknemers. De werknemers kunnen bijvoorbeeld steeds op verplaatsing zijn waardoor ze zo goed als nooit in contact komen met leden van het management. Daarnaast kan de drempel te hoog zijn voor de werknemers om iemand van het management aan te spreken. Wanneer ze dat wel doen brengen ze niet vlug de ideeën of problemen die ze hebben ter sprake.

Een mogelijke oplossing om deze stap te vergemakkelijken is een suggestiedoos. Een soort brievenbus waar werknemers een briefje in kunnen steken waarop zij hun idee schrijven. Het idee kan gaan van manieren om het werk te vereenvoudigen of minder belastend te maken tot leuke ideeën voor het personeelsfeest. Een goed idee kan beloond worden met een prijs, een waardebon of een tablet bijvoorbeeld. De mogelijkheid om iets te winnen, verhoogt de kans op participatie van de werknemers.

Een suggestiedoos brengt echter moeilijkheden mee die zwaar doorwegen. Zo is het na een tijd zeer moeilijk om alle ideeën bij te houden en op te volgen. In de praktijk is ook gebleken dat werknemers na verloop van tijd objecten zoals boterhammen in de suggestiedoos deponeerden. In de bedrijven waar de klant dit concept al opgezet had werd een Excelbestand gebruikt om de ideeën op te volgen, maar helaas wordt ook dit na een tijd onoverzichtelijk.

2.2 Oplossing

Deze opdracht is bedacht door Jef Martens, de klant. Bovendien is hij ook degene waarvoor de webapplicatie gemaakt wordt. De klant heeft eerst Forwrd aangesproken om de applicatie te maken, vervolgens zijn zij een samenwerking met PXL Research aangegaan om de applicatie verder uit te werken.

De opdracht omvat het bouwen van Ideashooter, een platform dat een oplossing tracht te bieden voor het probleem dat in de probleemstelling 2.1 besproken is. Het proces om Ideashooter te bouwen is opgesplitst in meerdere delen. Het eerste deel, de analyse, is afgerond voor de start van de stage. Voor het design en een groot deel van de frontend heeft een designer/frontend ontwikkelaar de meeste views al gemaakt. Het tweede deel, de stageopdracht, is het bouwen van het platform. Hieronder hoort het functioneel maken van de views en het uitwerken van de backend. Na de stage loopt de ontwikkeling van het project verder om meer features te implementeren.

2.2.1 Doelstellingen

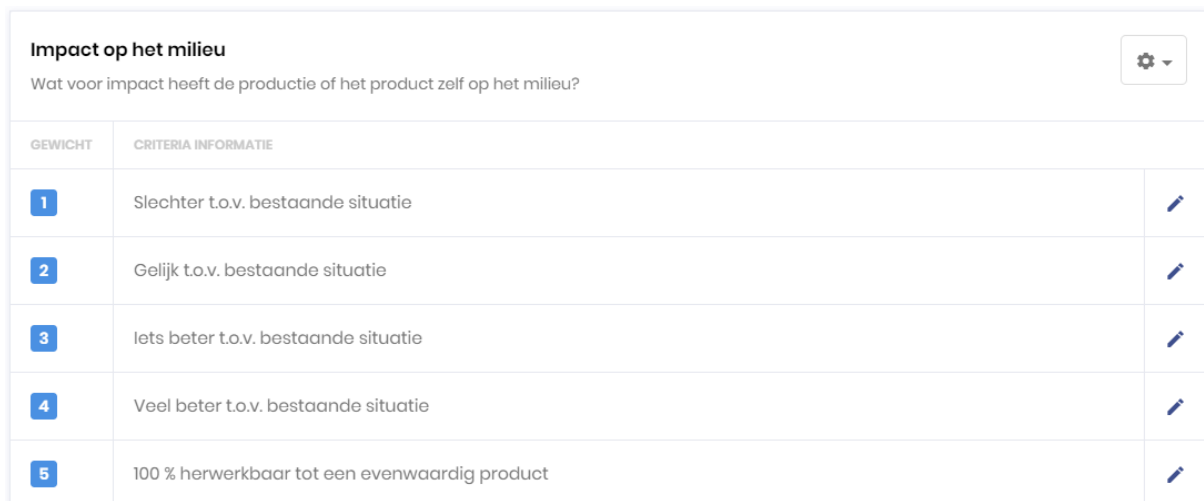
Ideashooter houdt als doel voor ogen om ideeën of problemen uit te wisselen tussen het management en de werknemers van het bedrijf. Elk bedrijf kan de applicatie gebruiken tegen betaling en deze zal beschikbaar zijn via een *subdomain* van ideashooter.com. Dit *subdomain* zal bestaan uit de naam van het bedrijf gevolgd door '.ideashooter.com'. Bijvoorbeeld: pxl.ideashooter.com.






Werknemers kunnen inloggen op het platform met een e-mailadres en wachtwoord. Daarnaast kunnen ze ook het profiel aanvullen met persoonlijke informatie zoals een telefoonnummer. Indien ze dit niet doen, maken ze ook geen kans op prijzen aangezien het management deze informatie nodig heeft om de winnaars te bereiken. Een andere mogelijkheid voor het management is om de werknemers te contacteren via hun e-mailadres, dat kan men standaard in de databanken van de

applicatie terugvinden. Maar omdat e-mails niet altijd goed ontvangen, denk aan een *spam*-folder, of gelezen worden, wordt er gevraagd om ook een telefoonnummer op te geven.

Binnen Ideashooter kan een select groepje gebruikers van het management wedstrijden opstellen, zij krijgen de naam '*innovationboard*'. Elke wedstrijd heeft een thema, een prijs, en omvat minimum één en maximum vijf wedstrijdsvragen. Daarnaast kan degene die de vraag opstelt ook kiezen welke doelgroep(en) deze wedstrijd te zien krijg(t)(en) op zijn dashboard. Werknemers kunnen vervolgens antwoorden geven op de vragen van een wedstrijd. Wie het beste voorstel heeft op basis van de evaluatiecriteria, zie hierna: 3.2.4, wint de prijs die bij de wedstrijd vooropgesteld is. De prijzen die de deelnemers kunnen winnen kan het management telkens zelf kiezen, bijvoorbeeld een cadeaubon of een broodjeslunch voor heel het team. Er kan ook gekozen worden om geen prijs te koppelen aan een wedstrijd.

Aan de hand van de vooropgestelde '*scoping criteria*' en een algoritme wordt de kwaliteit van een antwoord of idee bepaald. De *scoping* is het evalueren van vragen of ideeën die werknemers hebben ingestuurd. Om het evalueren in goede banen te leiden zijn er subcategorieën en bijhorende criteria vooropgesteld zoals in Figuur 4 weergegeven.



| Impact op het milieu | | |
|---|---|---|
| Wat voor impact heeft de productie of het product zelf op het milieu? | | |
| GEWICHT | CRITERIA INFORMATIE | |
| 1 | Slechter t.o.v. bestaande situatie |  |
| 2 | Gelijk t.o.v. bestaande situatie |  |
| 3 | Iets beter t.o.v. bestaande situatie |  |
| 4 | Veel beter t.o.v. bestaande situatie |  |
| 5 | 100 % herwerkbaar tot een evenwaardig product |  |

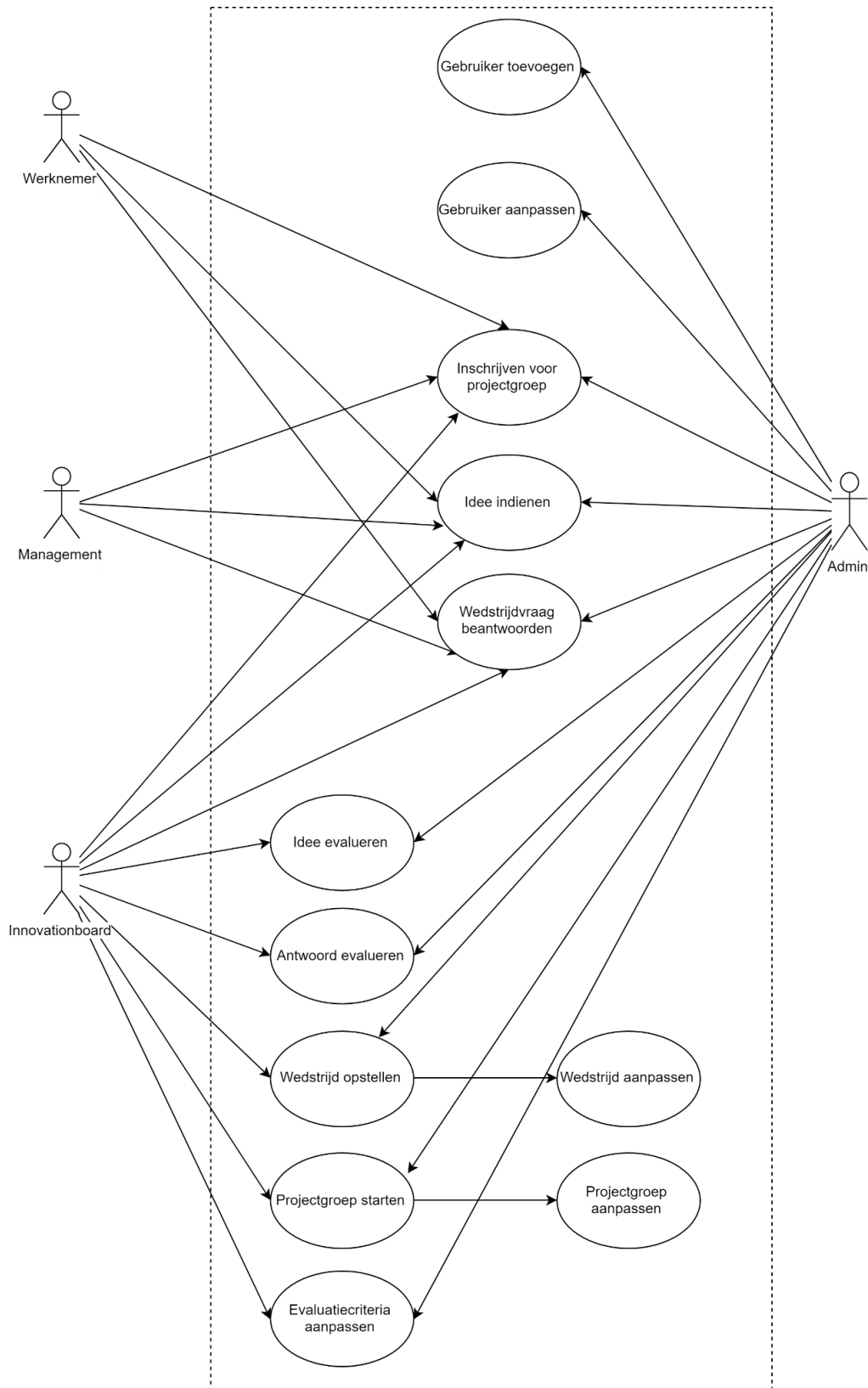
Figuur 4 Scoping: subcategorie met bijhorende criteria

De leden van het *innovationboard* kunnen de subcategorieën en criteria zelf aanmaken, aanpassen of verwijderen. Er zijn telkens vijf criteria die onder een subcategorie horen, en elke van de vijf criteria heeft een ander gewicht. Des te groter het gewicht des te zwaarder het zal doorwegen in het algoritme dat de score zal berekenen.

Wanneer er genoeg goede antwoorden of ideeën ingestuurd worden, kan er een projectgroep opgestart worden. Bovendien kunnen werknemers die zich willen engageren voor een bepaald idee lid worden van een projectgroep, of toegevoegd worden door iemand van het *innovationboard*. In een projectgroep gaan de leden effectief over tot actie om een bepaald idee te verwezenlijken.

Niet enkel het *innovationboard* kan vragen stellen door middel van wedstrijden, de werknemers kunnen op hun beurt ook vragen stellen aan het *innovationboard*. Bijvoorbeeld de vraag om de verlichting in een werkplaats te verbeteren of een idee om gezamenlijk de werkplaats ordelijk te houden. Zo'n idee kan op dezelfde manier geëvalueerd worden door leden van het *innovationboard*.

Ter illustratie van de functionaliteit binnen Ideashooter is er een *use case* diagram bijgevoegd (Figuur 5).



Figuur 5 Bondige use-case diagram

2.3 Omgeving

Voor de implementatie van de applicatie is er gekozen voor Laravel, een PHP MVC framework, en Vue.js, een JavaScript framework. Vue.js wordt standaard geleverd bij Laravel en wordt gebruikt voor het dynamisch maken van de webpagina's. Laravel zal steeds de pagina verversen wanneer er iets aangepast wordt, net zoals alle *serverside* frameworks. In sommige gevallen ziet het er vlotter uit wanneer Vue.js gebruikt wordt en de pagina niet verversen. Bijvoorbeeld wanneer er een extra rij in een tabel dient te verschijnen na het aanklikken van een bepaalde knop.

Een factor in de keuze van deze technologieën is dat bij Laravel weinig *boilerplatecode* geschreven dient te worden. Voor authenticatie is bijvoorbeeld maar één commando nodig om de nodige code te genereren en vanaf dan kan je registreren, inloggen en het wachtwoord herstellen via e-mail. Daarna kunnen de gegenereerde bestanden aangepast worden waar nodig. Zo kan het mogelijk zijn dat om het wachtwoord te veranderen een strengere validatie nodig is, denk bijvoorbeeld aan de lengte en complexiteit van het wachtwoord.

Om de webpagina's dynamisch te maken, heeft men gekozen voor Vue.js. Dit omdat het gebruik hiervan geadviseerd is door Laravel en de documentatie afgestemd is op deze combinatie. Andere keuzemogelijkheden zijn React.js of Angular. Als Angular gebruikt wordt, dient er eerst een API geschreven te worden die vanuit Angular aangesproken kan worden. Daardoor zou er een scheiding zijn tussen front- en backend. Het probleem hiervan is dat er bijvoorbeeld met JWT of iets dergelijks gewerkt dient te worden om ervoor te zorgen dat de gebruikers veilig kunnen inloggen. JWT zijn versleutelde tokens die ervoor zorgen dat data veilig tussen twee partijen uitgewisseld kan worden [4]. Het implementeren van JWT kan veel tijd in beslag nemen en dient correct te gebeuren om het nut ervan te verzekeren. Wanneer Laravel gebruikt wordt als front- en backend heeft Laravel controle over de hele applicatie. Doordat Laravel dergelijke beveiliging ingebouwd heeft dient de ontwikkelaar dit niet zelf te implementeren.

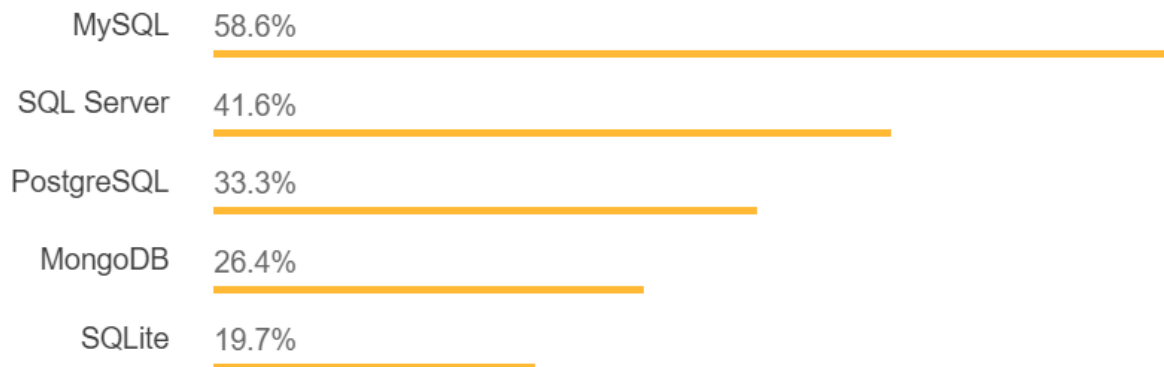
Vue.js is een *lightweight* framework om de frontend van webapplicaties te bouwen. Het werkt net zoals React.js en Angular met *componenten* die de ontwikkelaar zelf kan bouwen en hergebruiken. Zo'n *componenten* kunnen daarna in de stijl van HTML-tags in de HTML-bestanden geschreven worden zoals in Figuur 6 geïllustreerd. Een groot voordeel van Vue.js is dat deze overal snel geïmplementeerd kan worden, ook in combinatie met andere frameworks of *libraries*. Daarnaast is het mogelijk om een volledige applicatie te bouwen in Vue.js maar dat valt buiten de *scope* van deze opdracht. Vue.js zal binnen Ideashooter gecombineerd worden met de *templatingtaal* Blade, die standaard in het pakket van Laravel zit. Blade zorgt ervoor dat in de *templates* logica geschreven kan worden zoals een *foreach*-lus of een *if/else*-statement.

```
<scoping-section>
</scoping-section>
```

Figuur 6 Vue.js component in HTML-bestand

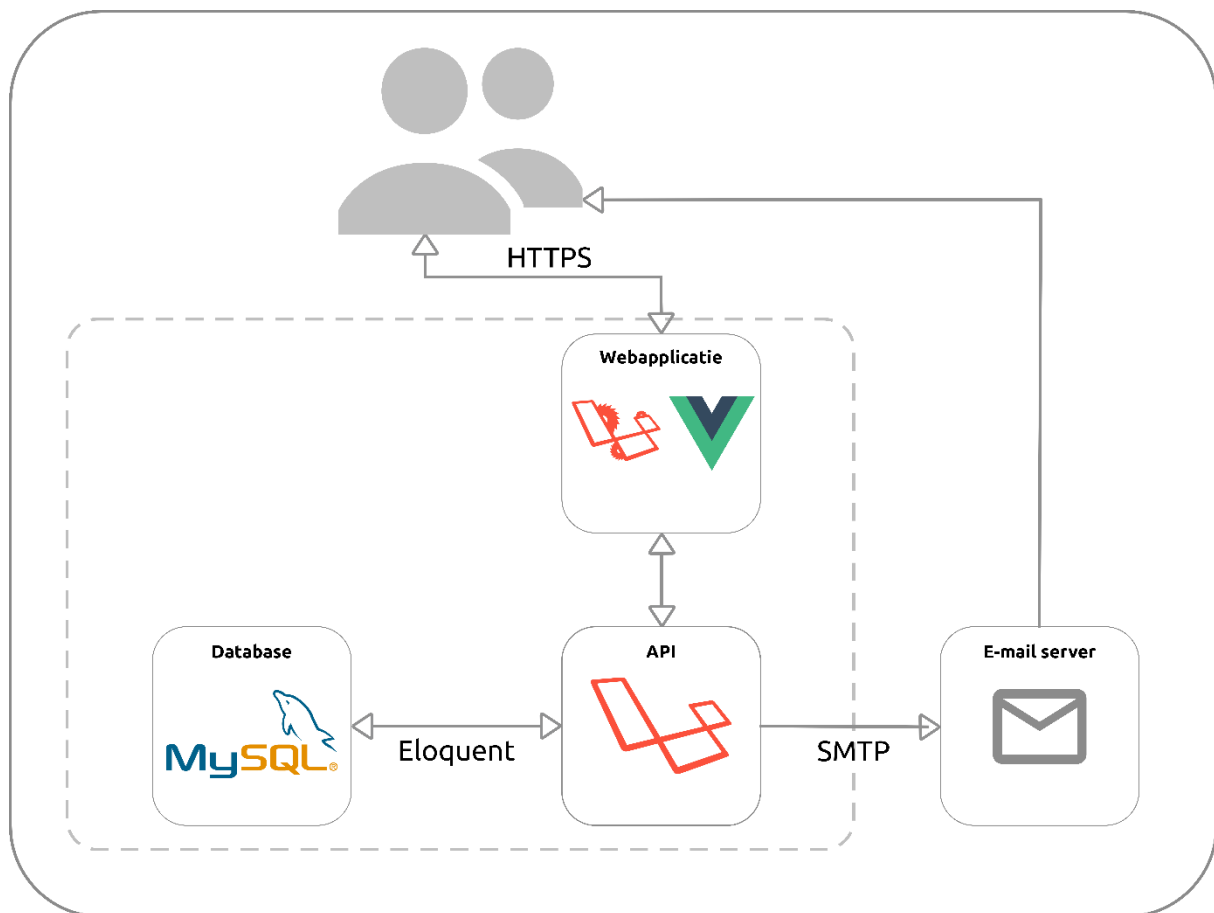
Laravel ondersteunt enkele databases actief, namelijk MySQL, PostgreSQL, SQLite en SQL Server. Om de kosten tijdens de ontwikkelperiode te drukken, is er gekozen om MySQL te gebruiken. De PXL host namelijk MySQL-databanken die gebruikt kunnen worden voor dit project. Daarnaast is MySQL volgens een enquête van StackOverflow [5] de meest gebruikte database van 2018. Figuur 7 laat zien hoeveel

procent, van de 100.000 ontwikkelaars die hebben deelgenomen aan de enquête in 2018, gewerkt hebben met MySQL.



Figuur 7 Staafdiagram van top vijf gebruikte databanken

In Figuur 8 wordt een korte samenvatting van de technologieën schematisch weergegeven.



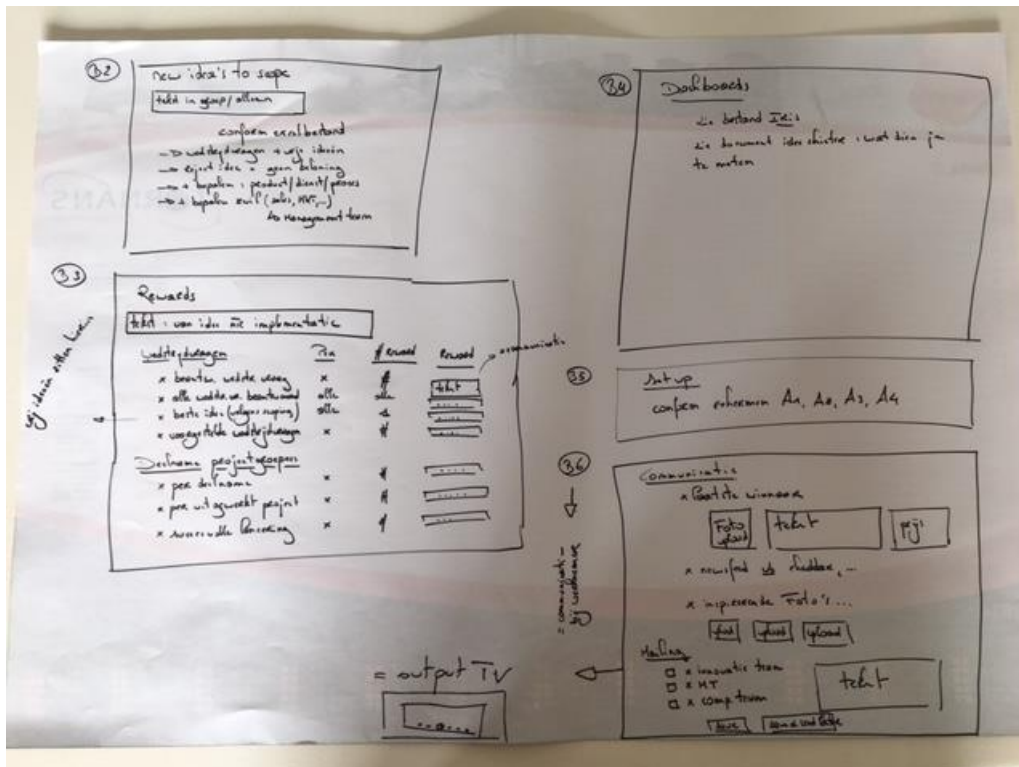
Figuur 8 Architectuur van Ideashooter

3 Uitwerking stageopdracht

3.1 Analyse

De analyse van het project is reeds afgerond wanneer de stage start. Niek Vandael heeft dit samen met de designer/frontend ontwikkelaar Gert Pellens en de klant Jef Martens gedaan.

De wireframes (Figuur 9) zijn in het begin van de analyse gemaakt op papier door de klant. Hij heeft in grote lijnen zijn idee omgezet in een aantal schermen om daarna de *requirements* te bespreken met Niek en Gert.



Figuur 9 Eerste wireframes

Tijdens het tweede deel van de analyse is Niek aan de slag gegaan met de wireframes en heeft deze omgezet in een *storyboard*. Door middel van een *storyboard* kan de ontwerper laten zien hoe de applicatie er in grote lijnen uit zal zien. In Figuur 10 wordt een scherm weergegeven van het *storyboard* om een wedstrijd te beantwoorden.



Figuur 10 Scherm om wedstrijd te beantwoorden in storyboard

Nadat men de *storyboards* besproken had, is Gert verder gegaan met het ontwerpen van de prototypes. Een prototype geeft gedetailleerder weer hoe het eindproduct er uit zal zien. Er is bijvoorbeeld meer aandacht voor kleuren, het lettertype en de algemene vormgeving. Het prototype in Figuur 11 toont detail weer dan het storyboard weergegeven in Figuur 10.

Hoe gaan we papierverspilling tegen?

Win een **Coolblue cadeaubon t.w.v. 100€**

Een woordje uitleg over de vraag. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus ac orci et purus tempor semper in non sem. Vestibulum accumsan orci in malesuada vestibulum. Sed nec dignissim dolor. Cras ante enim, aliquet sed sem non, porttitor tristique mi.

Mijn antwoord

Type je antwoord hier...

Attachment toevoegen

Selecteer bestand

Antwoord verzenden

Figuur 11 Prototype van scherm 'wedstrijdvraag beantwoorden'

3.2 Ontwikkeling

3.2.1 Setup

3.2.1.1 Project genereren en hosten op webserver

Om een Laravel-project te starten kan het commando `laravel new ideashooter` gebruikt worden. Dit zal de nodige code en bestandsstructuur genereren om verder op te bouwen.

Omdat de klant de webapplicatie van op afstand wil testen dient het project gehost te worden op een webserver die extern toegankelijk is. Zoals de database kan de webapplicatie tijdens de ontwikkelingsfase ook gehost worden op een server van de PXL. Hogeschool PXL gebruikt Apache als webserver en DirectAdmin om de webserver te beheren.

Er zijn verschillende opties om de code op de server te plaatsen maar aangezien er maar één persoon aan dit project werkt en de applicatie op het einde van de stage naar Azure of iets dergelijks verhuist, is er gekozen om de bestanden simpelweg te uploaden door middel van FTP. In PhpStorm, de IDE die tijdens deze stage gebruikt is, zit standaard een FTP-client. Die client kan ingesteld worden om telkens wanneer het bestand opgeslagen wordt, het bestand te uploaden. Als er meerdere mensen aan eenzelfde project werken, kan dit een probleem vormen. Bij elke upload zullen de bestanden die aangepast zijn overschreven worden op de server. De persoon die de laatste aanpassing doet zal dus ook de versie bepalen op de server. Om die reden is het niet aangewezen om deze methode te gebruiken wanneer er met meerdere personen aan de code gewerkt wordt. In zulke situaties is een *pipeline* die gekoppeld is aan een *gitrepository* een betere oplossing.

Om Laravel te kunnen hosten is enkel een webserver zoals Apache of Nginx nodig met PHP-versie 7.1.3 of hoger. Daarnaast dienen er enkele extensies geactiveerd te zijn, zie Figuur 12 [6].

- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- BCMath PHP Extension

Figuur 12 Nodige extensies om Laravel te hosten

3.2.1.2 Database

Het maken en onderhouden van tabellen in de database gaat via Eloquent, een ORM die Laravel gebruikt. Om een tabel aan te maken, dient er eerst een *migration* gemaakt te worden met het commando `php artisan make:migration`. In de gegenereerde *migration* kunnen de kolommen gedefinieerd worden volgens de regels van Eloquent. Buiten de naamgeving en datatypes kunnen ook additionele eigenschappen meegegeven worden, zoals de lengte van een tekst.

```
Schema::create('employee_questions', function (Blueprint $table) {
    $table->increments('id');
    $table->integer('user_id')->unsigned();
    $table->string('question');
    $table->string('additional_info', 500);
    $table->string('status');
    $table->binary('file')->nullable();
    $table->timestamps();
});
```

Figuur 13 Methode voor het aanmaken van tabel

In Figuur 13 worden de functies ‘unsigned()’ en ‘nullable()’ toegevoegd aan een kolom. Daardoor zal respectievelijk de kolom ‘user_id’ in de tabel geen negatieve getallen kunnen bevatten met als gevolg dat de maximum positieve waarde dubbel zo hoog zal zijn en de kolom ‘file’ een NULL-waarde kan bevatten.

De migrations worden telkens uitgevoerd wanneer het commando `php artisan migrate:fresh` gebruikt wordt. De bestaande tabellen en hun inhoud worden allemaal verwijderd en terug opnieuw aangemaakt. Dat is vooral handig wanneer de database gevuld is met test data die niet meer nodig is. Wanneer het argument ‘*fresh*’ weggelaten wordt, zullen de tabellen enkel geüpdatet worden.

3.2.1.3 Importeren HTML en CSS in Laravel

Om de HTML en CSS die de frontend ontwikkelaar gemaakt heeft te kunnen gebruiken dienen deze bestanden het eerst geïmporteerd te worden in Laravel. Laravel gebruikt in plaats van de standaard .html bestandsextensie de .blade.php extensie. Daardoor kan een ontwikkelaar gebruik maken van “Blade syntax” om de views dynamischer te maken. Met deze syntax kan een *foreach*lus geschreven worden om in de view te itereren over data die door de *controller* teruggeven wordt (Figuur 15). Daarnaast kunnen bijvoorbeeld ook componenten toegevoegd worden in de views (Figuur 14).

```
@foreach($contests as $contest)

    <contest-table></contest-table>

@endforeach
```

Figuur 15 Foreach lus met Blade en Vue.js component

De aangeleverde HTML-bestanden bevatte veel *duplicate* code waardoor ze best opgedeeld werden in *components* binnen Laravel. Zogenaamde *components* zijn afzonderlijke stukjes code die idealiter de *duplicate* code gaan verminderen doordat ze hergebruikt kunnen worden. De voornaamste stukken

code die in *components* gezet werden waren de navigatiebalk en header. Wanneer er nadien code aangepast wordt, dient dit maar op één plaats te gebeuren in plaats van in elke view waar die code gebruikt is.

3.2.2 Authenticatie

Laravel biedt de mogelijkheid om de *boilerplatecode* voor authenticatie te laten genereren door het commando `php artisan make:auth`. Enkele bestanden die gegenereerd worden, zijn de views om in te loggen en het wachtwoord te veranderen indien de gebruiker dit vergeten is. Ook de *controllers* en *migrations* worden aangemaakt, deze bevatten code om gebruikers te laten aanmelden.

3.2.2.1 Rollen

Elke gebruiker van Ideashooter krijgt één of meerdere rollen toegewezen. De rollen die toegewezen kunnen worden zijn werknemer, manager en *innovationboard*. Iedereen die Ideashooter gebruikt heeft de standaardrol van werknemer. Daarnaast kan een werknemer tot het management behoren en eventueel ook tot het *innovationboard*. Het *innovationboard* is een groep werknemers die ervoor instaan om wedstrijden op te stellen en de ingestuurde antwoorden te evalueren.

De rollen staan in de database gedefinieerd en hebben een *many to many* relatie met de *user*. De gebruiker kan op zijn profiel zien tot welke rollen hij behoort (Figuur 16).

Ideashooter – Lid Werknemer / Management / Innovation

Figuur 16 Rollen op profielpagina

Door de implementatie van rollen kan er op het platform een onderscheid gemaakt worden tussen verschillende groepen van gebruikers. Sommige webpagina's mogen bijvoorbeeld enkel toegankelijk zijn voor gebruikers die de rol *innovationboard* hebben, denk aan de pagina om wedstrijd vragen op te stellen of de pagina om gebruikers aan projectgroepen te koppelen.

In de code vertaalt zich dat naar een model *User* die een *many to many* relatie definieert met het model *Role* (Figuur 17).

```
public function roles()
{
    return $this->belongsToMany(Role::class);
}
```

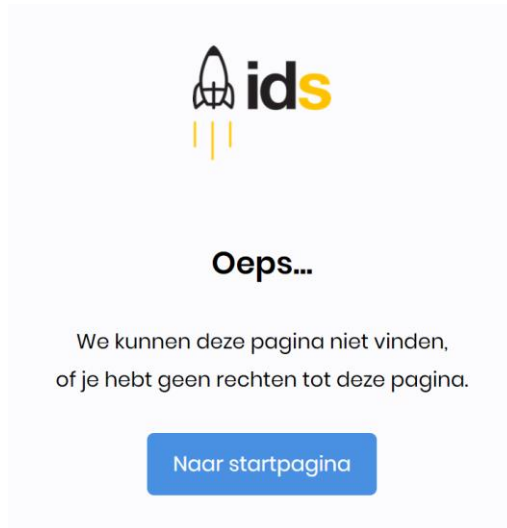
Figuur 17 Many to many relatie in model User

In het model *Role* gebeurt hetzelfde maar dan aan de andere kant van de relatie (Figuur 18).

```
public function users()
{
    return $this->belongsToMany(User::class);
}
```

Figuur 18 Many to many relatie in model Role

In de klasse *User* staan nog enkele methoden, zoals 'authorizeRoles()' weergegeven in Figuur 20, om te controleren of de ingelogde gebruiker de juiste rol(len) heeft om de pagina te bezoeken. Indien dit niet het geval is zal de gebruiker naar een 'not-authorized' pagina (Figuur 19) gestuurd worden die weergeeft wat er misgelopen is.



Figuur 19 Errorpagina binnen Ideashooter

```
public function authorizeRoles($roles)
{
    if (is_array($roles)) {
        return $this->hasAnyRole($roles) ||
            abort(401, 'Je hebt geen rechten om deze actie uit te voeren');
    }
    return $this->hasRole($roles) ||
        abort(401, 'Je hebt geen rechten om deze actie uit te voeren');
}
```

Figuur 20 Methode die controleert of een gebruiker de correcte rollen heeft

De *authorizeRoles* methode kan worden aangeroepen in een *action method* van de *controller* met als parameter de rollen die toegang krijgen, zoals in Figuur 21 beschreven.

```
function showInnovationQuestions(Request $request)
{
    $request->user()->authorizeRoles(['management', 'innovationboard']);

    return view('pages.question.innovation-questions', [
        'contests' => Contest::where('user_id', Auth::id())
            ->where('published', false)
            ->get();
    ]);
}
```

Figuur 21 Controller action method die een controle doet op de rollen die de gebruiker heeft

3.2.2.2 Autorisatie

Daarbuiten kan aan gebruikers ook de toegang ontzegd worden op basis van authenticatie. Zo hebben gebruikers die niet aangemeld zijn geen toegang tot bepaalde webpagina's. Evenzeer hebben de aangemelde gebruikers geen toegang tot alle pagina's. Een voorbeeld hiervan is dat het scherm om aan te melden niet meer toegankelijk mag zijn voor een gebruiker die reeds aangemeld is. In Laravel is dit gemakkelijk te implementeren door middleware. Middleware is een HTTP-filter die tussen de *request* en *response* zit. In Laravel zit bijvoorbeeld de *Authenticate* middleware die zal controleren of de gebruiker geautoriseerd is of niet. Die middleware kan aangeroepen worden op een route (Figuur 22) of in de *constructor* van een controller (Figuur 23).

```
Route::get('profile', 'UserController@showProfile')->middleware('auth');
```

Figuur 22 Auth middleware op route

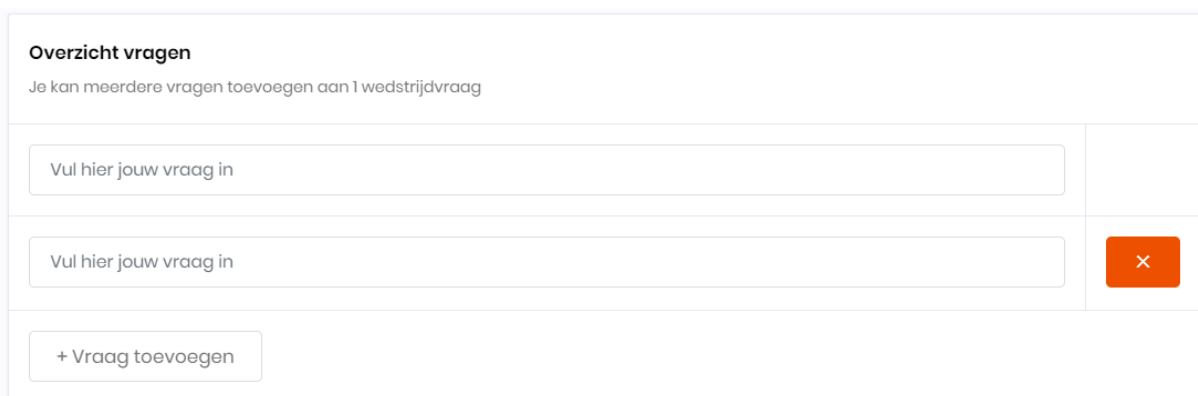
```
public function __construct()  
{  
    $this->middleware('auth');  
}
```

Figuur 23 Auth middleware in constructor van een controller

3.2.3 Wedstrijden

Het *innovationboard* kan wedstrijden opstellen die een onderwerp, prijs en maximum vijf vragen bevatten. Een onderwerp en één vraag zijn verplicht; de prijs kan leeg blijven wanneer er geen prijs is voorzien. Zo'n prijs kan bijvoorbeeld een lunch zijn voor het hele team.

Initieel staat op de pagina maar één invulveld voor een wedstrijdvraag. De gebruikers kunnen velden toevoegen als ze meer vragen willen koppelen aan de wedstrijd. Daarnaast kunnen ze de velden ook terug verwijderen wanneer er te veel werden toegevoegd (Figuur 24). Om de *user experience* te verbeteren kan de pagina best niet ververs worden voor deze functionaliteit, iets wat Laravel wel doet als het veranderingen op de pagina wil tonen. Dit is een ideale insteek om Vue.js te gebruiken.



The screenshot shows a web interface for managing questions. At the top, it says 'Overzicht vragen' and 'Je kan meerdere vragen toevoegen aan 1 wedstrijdvraag'. Below this, there are three rows of input fields, each containing the placeholder text 'Vul hier jouw vraag in'. The second row has a red square button with a white 'X' icon to its right. At the bottom left, there is a button labeled '+ Vraag toevoegen'.

Figuur 24 Wedstrijdvragen toevoegen of verwijderen

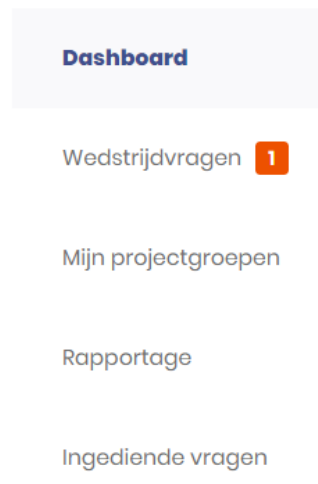
Wedstrijdvragen zitten in een *array*. Wanneer de wedstrijd opgeslagen wordt, worden deze één voor één in de database geplaatst. Om een rij toe te voegen wordt er een nieuw object aan de *array* van wedstrijd vragen toegevoegd. In het overzicht van de vragen itereert Vue.js over het aantal vragen dat

in de *array* zit (Figuur 25). Zo zal met een druk op de knop 'Vraag toevoegen' ook een nieuw invulveld om een vraag toe te voegen verschijnen.

```
<question-row v-for="(item, index) in questions">
```

Figuur 25 For-loop in Vue.js

De wedstrijden zijn minstens naar één doelgroep gericht, bijvoorbeeld de logistieke dienst van het bedrijf. Wanneer het *innovationboard* één of meerdere doelgroepen aanduidt, krijgen de gebruikers die tot de aangeduide groepen horen een e-mail toegestuurd. Daarnaast komt er ook een klein rood blokje (Figuur 26) in de navigatiebalk naast het item 'Wedstrijdvragen' met daarin het aantal onbeantwoorde vragen.



Figuur 26 Notificatieblokje van onbeantwoorde vragen

Wanneer de gebruikers via een link in de e-mail of via de webapplicatie naar het overzicht van de wedstrijdvragen navigeren, ziet men de wedstrijden en bijhorende vragen die gepubliceerd zijn. Naast de vragen staat een kleine *badge* die laat zien of de vraag reeds beantwoord is door de gebruiker (Figuur 27 en Figuur 28).

onbeantwoord

Hoe kunnen we minder papier gebruiken?

Figuur 27 Onbeantwoorde vraag

ingediend

Hoe kunnen we minder papier gebruiken?

Figuur 28 Beantwoorde vraag

Laravel heeft een mechanisme dat toe laat om notificaties uit te sturen. Dit kan door een nieuwe *notification* aan te maken met het commando `php artisan make:notification`. In zo'n *notification*, een klasse van Laravel, staat een methode 'via()'. Daarin staan de wijzen waarop de notificatie uitgestuurd wordt.

```

public function via($notifiable)
{
    return ['mail', 'database'];
}

```

Figuur 29 Via methode in Notification

De ‘via’ methode, die in Figuur 29 beschreven staat, zal notificaties uitsturen via e-mail en zal deze ook in de database plaatsen. De notificaties die in de database staan hebben een kolom *read_at* die standaard op ‘NULL’ staat. Wanneer de notificatie gelezen wordt zal daar de huidige tijd en datum geplaatst worden. Aan de hand van deze *feature* van Laravel is het mogelijk om de *badges* in Figuur 27 en Figuur 28 te implementeren.

3.2.4 Scoping

3.2.4.1 Scoping criteria aanpassen

Als werknemers wedstrijd vragen beantwoorden komen de antwoorden terecht op de pagina ‘toegewezen vragen’. Daar worden alle wedstrijden weergegeven die de leden van het *innovationboard* hebben gepubliceerd. Naast elke wedstrijd staat een progressiebalk die weergeeft hoeveel antwoorden al geëvalueerd zijn (Figuur 30).

| | |
|---------------------|--------------|
| Ecologischer werken | 0/2 _____ |
| De ideale werkplek | 0/1 _____ |

Figuur 30 Wedstrijdvragen en progressiebalk

Als de leden van het *innovationboard* op een vraag klikken krijgen ze een gedetailleerd overzicht van de wedstrijd en de bijhorende antwoorden die de werknemers ingestuurd hebben zoals Figuur 31 weergeeft.

Ecologischer werken

Projectgroep starten

win Waardebon Coolblue C50

0/2 vragen geëvalueerd







| Hoe kunnen we minder papier gebruiken? | | |
|--|---|-------|
| WERKNEMER | ANTWOORD | SCORE |
| Jonas Kempeneers | Minder documenten afdrukken en meer elektronisch beschikbaar stellen. | D |

| Zijn er producten die we kunnen vervangen door een meer ecologische variant? | | |
|--|---|-------|
| WERKNEMER | ANTWOORD | SCORE |
| Jonas Kempeneers | We kunnen gerecycleerd papier gebruiken om de documenten af te drukken die louter informatief zijn. | D |

Figuur 31 Wedstrijd met vragen en antwoorden

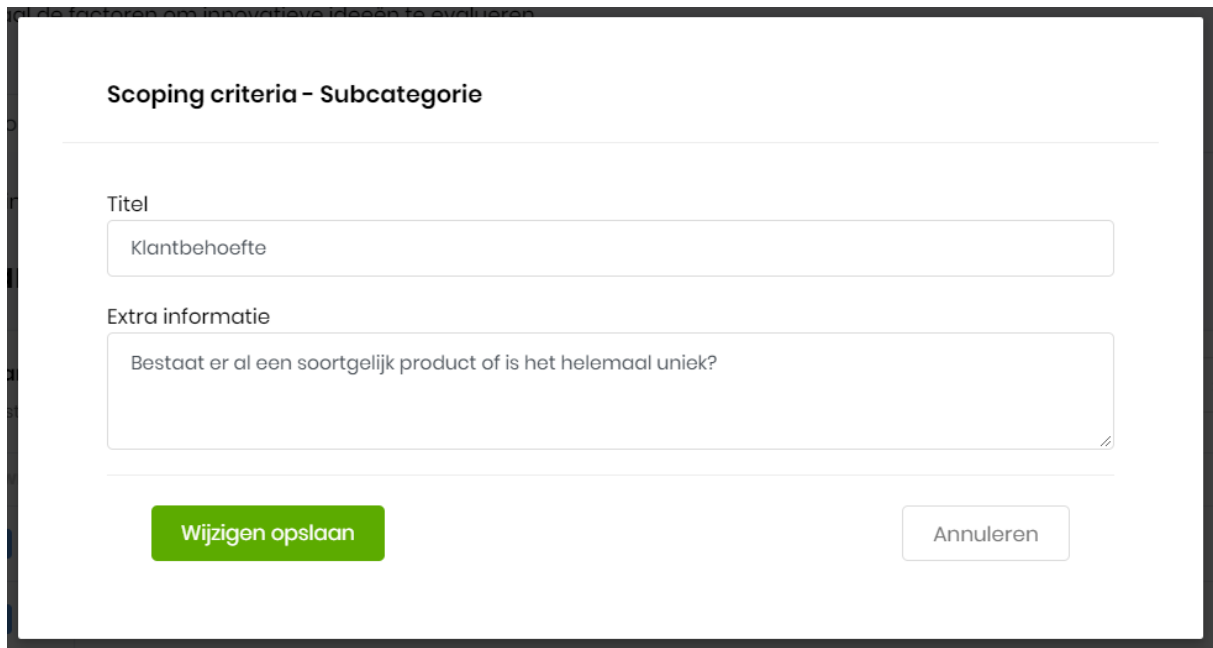
Wanneer leden van het *innovationboard* op een antwoord klikken komen ze terecht op een scherm waar een antwoord geëvalueerd kan worden aan de hand van verschillende criteria. De criteria zijn standaard meegeleverd bij Ideashooter, maar kunnen ook aangepast worden op het scherm 'scoping criteria'. Daar staan de categorieën en subcategorieën onderverdeeld onder de 3 tabbladen 'producten', 'diensten' en 'processen'. Onder elk tabblad staan twee categorieën met elk 5 subcategorieën en de bijbehorende *scoping* criteria (Figuur 32).

Marktattractiviteit

| Klantbehoefte | | |
|---|--|---|
| Bestaat er al een soortgelijk product of is het helemaal uniek? | |  |
| GEWICHT | CRITERIA INFORMATIE | |
| 1 | Vergelijkbaar met een bestaand product in een bestaande markt? |  |
| 2 | Iets beter dan een bestaand product in een bestaande markt? |  |
| 3 | Het product creëert een nieuw segment in de bestaande markt? |  |
| 4 | Beantwoord een onbeantwoorde latente klantenbehoefte |  |
| 5 | Beantwoord een onbeantwoorde waargenomen klantenbehoefte |  |

Figuur 32 Categorie marktattractiviteit met subcategorie klantbehoefte en bijhorende criteria

Een subcategorie heeft altijd 5 criteria waarvan elk een ander gewicht heeft, des te hoger het gewicht des te gunstiger. Elke subcategorie of criterium kan aangepast worden door de leden van het *innovationboard*.



Scoping criteria - Subcategorie

Titel

Klantbehoefte

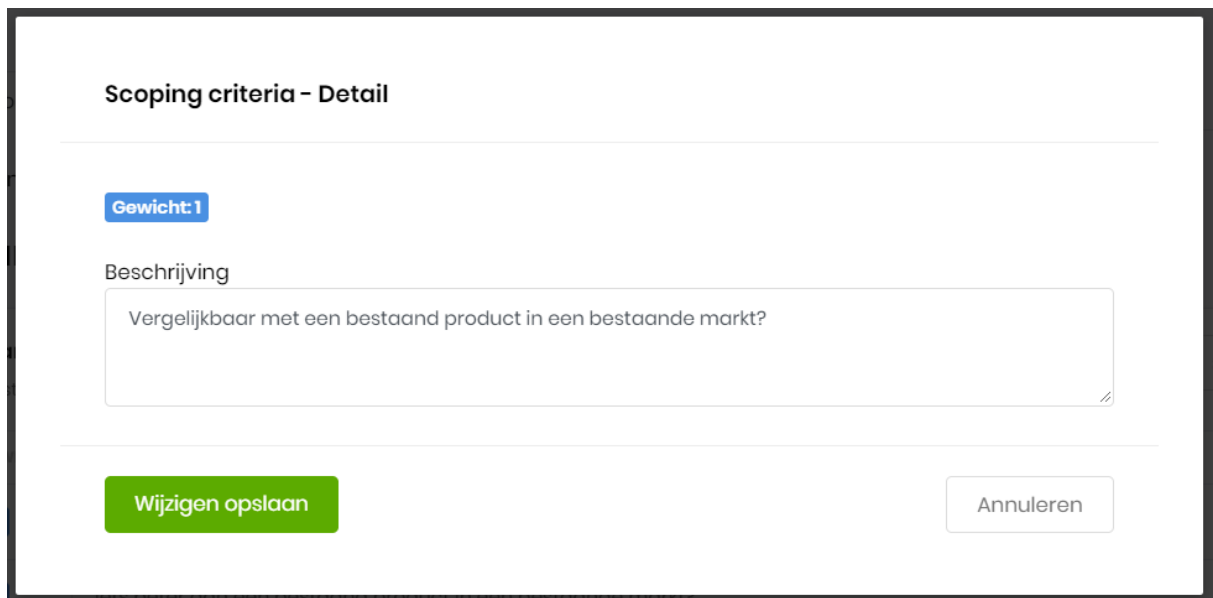
Extra informatie

Bestaat er al een soortgelijk product of is het helemaal uniek?

Wijzigen opslaan Annuleren

Figuur 33 Overliggend scherm om subcategorie aan te passen

Wanneer een gebruiker een subcategorie wil aanpassen, krijgt hij een overliggend scherm te zien waar de titel en extra informatie aangepast kan worden (Figuur 34). De twee velden zijn verplicht en dienen minstens 4 karakters te bevatten, uitgezonderd spaties. Hetzelfde geldt voor het aanpassen van de afzonderlijke criteria:



Scoping criteria - Detail

Gewicht: 1

Beschrijving

Vergelijkbaar met een bestaand product in een bestaande markt?

Wijzigen opslaan Annuleren

Figuur 34 Overliggend scherm om criteria aan te passen

Omdat het overliggend scherm een Vue-component is, kan de data van de beschrijving niet in Vue zelf opgehaald worden. Dit komt doordat Ideashooter geen API *endpoints* heeft die enkel data terugsturen.

Om die reden wordt de beschrijving tussentijds opgeslagen in het app.js bestand oftewel de *root instance*.

Allereerst dient er een variabele aanwezig te zijn om de tekst van de beschrijving op te slaan. Omdat het *id* en het gewicht ook getoond wordt, staan er 3 variabelen in app.js. Daarom is er gekozen om deze drie variabelen in een object te plaatsen (Figuur 35), met als gevolg dat het app.js bestand overzichtelijker is.

```
scopingCriterium: {  
  id: Number,  
  weight: Number,  
  content: String,  
}
```

Figuur 35 Object om data over criteria tussentijds op te slaan

Wanneer de informatie in het app.js bestand staat, kunnen waarden in de componenten of in de views van Laravel opgevraagd worden. In Figuur 36 wordt de code weergegeven om de variabelen op te vragen in de componenten.

```
id: this.$root.scopingCriterium.id,  
weight: this.$root.scopingCriterium.weight,  
content: this.$root.scopingCriterium.content,
```

Figuur 36 Manier om variabelen uit app.js bestand op te vragen

3.2.4.2 Antwoorden evalueren

Om te zien welke antwoorden ingestuurd zijn door de werknemers kunnen de leden van het *innovationboard* naar “Beantwoorde wedstrijdvragen” navigeren, zoals Figuur 37 weergeeft, en daarna klikken op de wedstrijd om de details te bekijken. (Figuur 38).

Mijn toegewezen vragen

1/1 antwoord geëvalueerd

| | |
|------------------------|-----|
| Groepssfeer verbeteren | 1/1 |
|------------------------|-----|

Figuur 37 Overzicht openstaande wedstrijden

| Hoe kunnen we er voor zorgen dat het werk minder rugbelastend is? | | |
|---|--|-------|
| WERKNEMER | ANTWOORD | SCORE |
| Jonas Kempeneers | Cursus tiltechnieken voor alle mensen die risico lopen ... | ? |

Figuur 38 Overzicht ingestuurde antwoorden en de score

Om antwoorden te evalueren kunnen de gebruikers op een antwoord klikken. Daar worden de subcategorieën en bijhorende criteria weergegeven die ingesteld zijn op de pagina van de ‘*scoping criteria*’. Binnen elke subcategorie bestaat de keuze om één criterium aan te duiden (Figuur 39) die de uiteindelijke score mee zal bepalen.

| Waarde binnen 3 jaar |
|--|
| <input checked="" type="radio"/> < € 100.000 |
| <input type="radio"/> € 100.000 - € 200.000 |
| <input type="radio"/> € 200.000 - € 300.000 |
| <input type="radio"/> € 300.000 - € 400.000 |
| <input type="radio"/> > € 400.000 |

Figuur 39 Keuze tussen vijf criteria binnen elke subcategorie

De score wordt berekend op basis van een algoritme dat de klant aangeleverd heeft. Telkens wanneer de gebruiker een criterium aanduidt zal de score herrekend worden en vervolgens te zien zijn in een formulier onder de subcategorieën (Figuur 40).

Scoping criteria score: **Score: D**

Score overschrijven?

Scoping score: C ▼

Tags

ecologie

Evaluatie uitleg

Je kan hier extra informatie toevoegen aan je evaluering.

Opslaan Overslaan

Figuur 40 Formulier om scoping af te ronden

Als degene die de evaluatie doet vindt dat de uiteindelijke score naar zijn of haar gevoel niet correct is, kan de score overschreven worden. Dit kan ook gebruikt worden indien een vraag of antwoord geen economisch voordeel oplevert, zoals een vraag rond een werkhal die te koud is. Dit kan een belangrijk probleem zijn voor de werknemers ondanks een potentiële slechte score die het idee krijgt.

Daarnaast kan de beoordelaar tags meegeven aan een evaluatie zodat achteraf gefilterd kan worden op bepaalde kernwoorden zoals ecologie. Indien het nodig is kunnen de gebruikers bijkomende informatie geven over de evaluatie in het daarvoor bestemde veld, maar dat is niet verplicht.

Nadat de evaluatie ingestuurd is zal de werknemer die het antwoord gepost heeft op de hoogte gehouden worden via een e-mail met daarin de feedback die gekoppeld is aan de score. De feedback zit standaard in de database en kan per bedrijf aangepast worden.

Werknemers kunnen ingestuurde antwoorden terug bekijken op het scherm ‘rapportage’. Daar worden de details van het antwoord weergegeven, zoals de gekregen feedback wanneer het antwoord geëvalueerd is (Figuur 41).

| TYPE | ONDERWERP | RESULTAAT |
|----------------|-----------|---|
| Wedstrijdvraag | Ergonomie | A Goed idee. Het heeft een code A gekregen wat wil ... |

Figuur 41 Overzicht ingestuurde antwoorden

3.2.5 Projectgroepen

Wanneer er vragen of ideeën ingestuurd zijn die goed genoeg zijn om tot uitvoering te brengen kunnen de leden van het *innovationboard* een projectgroep aanmaken. Dat kan door op de knop ‘projectgroep starten’ te klikken die op de pagina ‘beantwoorde wedstrijdvrage’ staat weergegeven (Figuur 42).

Ecologie

Projectgroep starten

win Waardebon Coolblue €50

2/2 antwoorden geëvalueerd

Op welke manier kunnen we ecologischer werken?

| WERKNEMER | ANTWOORD | SCORE |
|---------------|--|----------|
| Jan Janssens | In plaats van alle mededelingen overal op te hangen kunnen we de minder belangrijke ook gewoon doorsturen via mail. | A |
| Peter Peeters | In de materiaalruimte staat de verlichting dikwijls de hele dag te branden terwijl we daar enkel 's ochtends en 's avonds echt bezig zijn. We kunnen eventueel lichtknoppen met een timer installeren? | B |
| An Anders | Geen papier meer bestellen, dan kunnen we het ook niet verspillen. | D |

Figuur 42 Overzicht van beantwoorde wedstrijdvrage

Nadat de gebruiker op deze knop klikt, verschijnt er een scherm om meer informatie te geven over de projectgroep zoals een naam voor de groep. Daarnaast wordt er een overzicht van alle antwoorden weergegeven die de werknemers hebben ingediend voor de wedstrijd, zoals Figuur 43 illustreert. In dit overzicht kunnen de leden van het *innovationboard* de antwoorden aanduiden die betrekking hebben op de projectgroep, zo is er een referentie om achteraf te bepalen vanuit welke ideeën de projectgroep begonnen is.

| | | | |
|--------------------------|--|---------------|----------|
| <input type="checkbox"/> | In plaats van alle mededelingen overal op te hangen kunnen we de minder belangrijke ook gewoon doorsturen via mail. | Jan Janssens | A |
| <input type="checkbox"/> | In de materiaalruimte staat de verlichting dikwijls de hele dag te branden terwijl we daar enkel 's ochtends en 's avonds echt bezig zijn. We kunnen eventueel lichtknoppen met een timer installeren? | Peter Peeters | B |
| <input type="checkbox"/> | Geen papier meer bestellen, dan kunnen we het ook niet verspillen. | An Anders | D |

Figuur 43 Overzicht van antwoorden bij het opmaken van een projectgroep

Naast de antwoorden wordt een overzicht getoond (Figuur 44) van de gebruikers die een antwoord hebben ingestuurd. Net zoals bij de antwoorden kunnen ze de gebruikers selecteren die meewerken in de projectgroep. Daarnaast zijn ze verplicht om één projectleider aan te duiden. Er kan steeds een gebruiker toegevoegd worden die nog niet in het overzicht staat via een zoekbalk onder het overzicht.

Deze personen hebben een antwoord ingediend voor deze wedstrijdvrage

| Toevoegen aan projectgroep | Projectleider | Projectsponsor | Naam werknemer |
|-------------------------------------|----------------------------------|----------------------------------|------------------|
| <input checked="" type="checkbox"/> | <input type="radio"/> | <input type="radio"/> | Jonas Kempeneers |
| <input checked="" type="checkbox"/> | <input checked="" type="radio"/> | <input type="radio"/> | Jan Janssens |
| <input checked="" type="checkbox"/> | <input type="radio"/> | <input checked="" type="radio"/> | Jos Vermeulen |

Andere werknemers toevoegen

Figuur 44 Overzicht van alle gebruikers die een antwoord ingezonden hebben

Nadat de informatie ingegeven is, komt de gebruiker terecht op een detailpagina van de projectgroep die ze net gestart hebben. Daar krijgen ze weer een overzicht van de antwoorden en gebruikers die ze hebben toegevoegd aan de projectgroep (Figuur 45). Elk groepslid kan dit overzicht raadplegen om zijn rol en taak te vinden binnen deze groep.

| Antwoorden | |
|--|---|
| In plaats van alle mededelingen overal op te hangen kunnen we de minder belangrijke ook gewoon doorsturen via mail | A |
| In de materiaalruimte staat de verlichting dikwijls de hele dag te branden terwijl we daar enkel 's ochtends en 's avonds echt bezig zijn. We kunnen eventueel lichtknoppen met een timer installeren? | B |
| Geen papier meer bestellen, dan kunnen we het ook niet verpillen. | D |

| Projectgroep leden | | | |
|--------------------|------------|---------------|----------------------------------|
| NAAM | GROEP | PROJECTROL | E-MAIL ADRES |
| Jonas Kempeneers | Werknemers | Team | jonas.kempeneers2@student.pxl.be |
| Jan Janssens | Werknemers | Projectleider | jan.janssens@gmail.com |
| Jos Vermeulen | Management | Sponsor | jos.vermeulen@gmail.com |

Figuur 45 Overzicht van geselecteerde antwoorden en groepsleden op detailpagina van projectgroep

Bovendien is er ook een overzicht met informatie over de huidige status van de projectgroep. In dit overzicht (Figuur 46) staat bijvoorbeeld meer informatie over de fase waarin het project zich bevindt, elke projectgroep zit namelijk in één van vijf fasen. De eerste fase is altijd de 'bouw business case' waar de projectgroep een business case gaat opstellen om te bepalen of het project rendabel kan zijn. Vanuit die eerste fase kan de projectgroep overgaan naar de verdere fasen zoals 'ontwikkeling', 'testen & valideren', 'lanceren' en 'afgesloten'. Het is niet verplicht om alle fasen te doorlopen. Wanneer de business case geen positief resultaat laat zien, kan het bijvoorbeeld gebeuren dat de projectgroep afgesloten wordt zonder in een andere fase terecht te komen.

In datzelfde overzicht (Figuur 46) staat informatie over hetgeen dat van iedereen binnen de projectgroep verwacht wordt. Per fase heeft elk groepslid een andere taak, afhankelijk van zijn rol binnen het project. Er zijn drie rollen die een gebruiker kan krijgen binnen een project namelijk: projectleider, projectsponsor en teamlid.

| Status projectgroep | |
|--------------------------|---|
| HUIDIGE FASE | Bouw business case |
| ROL VAN DE SPONSOR | Ondersteun het team met het opmaken van de business case |
| ROL VAN DE PROJECTLEIDER | Je maakt deel uit van het projectteam en stuurt dit aan. Je zorgt voor een goede en aangename samenwerking. De communicatie met de projectsponsor loopt via jou. |
| ROL VAN HET TEAM | Ga aan de slag met de tabbladen 'Business case' en focus op de markt. Wat is de markt? Hoe groot is deze, welk marktaandeel ga je hebben? Welke omzetten ga je realiseren? Definieer het idee, baken het idee technisch af en werk de technische aspecten uit. Vertaal deze aspecten naar een ontwikkelplan onder de tab ontwikkeling. Maak een voorlopige inschatting van de kosten onder de tab 'testen en valideren' en 'lanceren'. Ga na of het idee haalbaar is onder de tab 'financieel overzicht' en bij voldoende positieve kasstroom schuif je het project een fase op. Indien dit niet lukt overleg je met je sponsor en annuleer je het project. |

Figuur 46 Overzicht van de huidige status van de projectgroep



Van zodra de projectgroepen opgericht zijn kunnen de werknemers een overzicht bekijken van alle actieve projectgroepen zoals in Figuur 47 weergegeven. Bovendien wordt hier ook weergegeven van welke projectgroepen men al lid is.

| 🌟 Mijn projectgroepen | | |
|-------------------------|--|--------------------|
| PROJECTOMSCHRIJVING | | PROJECTFASE |
| Dit is een nieuwe groep | | Bouw business case |
| Passende naam | | Bouw business case |

| Andere actieve projectgroepen | | |
|---|-------------|--------------------|
| PROJECTOMSCHRIJVING | INSCHRIJVEN | PROJECTFASE |
| In hoogte instelbare bureaus. | Inschrijven | Bouw business case |
| Projectgroep om ergonomie te verbeteren | Inschrijven | Ontwikkeling |

Figuur 47 Overzicht van actieve projectgroepen

Als een gebruiker zich wilt inschrijven voor een bepaalde projectgroep is dat mogelijk door op de knop 'Inschrijven' te klikken. Dit zorgt ervoor dat de leden van deze projectgroep een notificatie krijgen. Elk lid van de projectgroep kan een verzoek om te participeren van een andere gebruiker accepteren of afwijzen (Figuur 48).

| Accepteer nieuwe groepsleden | | | | |
|------------------------------|------------|----------------------------------|---|---|
| NAAM | GROEP | E-MAIL ADRES | ACCEPTEEER | WEIGER |
| Jonas Kempeneers | Werknemers | jonas.kempeneers2@student.pxl.be |  |  |

Figuur 48 Tabel met verzoeken om deel te nemen aan projectgroep

3.2.5.1 Business case

In de eerste fase dienen de leden van de projectgroep een business case op te stellen. Aan de hand van die case kunnen de leden bepalen in hoeverre het project haalbaar is. Zo is het mogelijk dat het project een te groot budget vereist om rendabel te zijn. Dit kan bijvoorbeeld een reden zijn om het project af te sluiten.

De business case kan opgemaakt worden in de applicatie zelf op het scherm 'business case'. In dit scherm zijn er enkele tabbladen zichtbaar met telkens enkele inputvelden waarin de leden van de projectgroep informatie kunnen ingeven zoals in Figuur 49 weergegeven. De data die de leden ingeven wordt opgeslagen in de database om later te kunnen raadplegen en om het financiële overzicht te kunnen opstellen.

De markt

Ontwikkeling

Testen en valideren

Lanceren

Financieel overzicht

Project evaluatie

Welk probleem lossen we op?

Definieer het idee om dit probleem op te lossen en baken het idee technisch af

Hoe uniek is de oplossing?

Bestaat deze oplossing al? Welke alternatieven zijn overwogen?

Wanneer ervaren we succes?

Hoe gaan we achteraf succes meten? Aan welke voorwaarden dient te zijn voldaan?

Figuur 49 Webpagina om business case op te stellen

Op het tabblad 'financieel overzicht' worden alle ingegeven kosten weergegeven (Figuur 50). Daarnaast berekent de applicatie de verwachte levensduur en terugverdientijd van het project. Op die manier krijgen de leden van de projectgroep een duidelijk overzicht over de status van het project.

| Kosten | |
|---------------------|-------------|
| CATEGORIE | BUDGET IN € |
| Ontwikkeling | 200 |
| Testen en valideren | 100 |
| Lanceren | 300 |
| Totaal | 600 |

| Verwachte levensduur | |
|------------------------------------|--------|
| Hoe lang zal het project mee gaan? | 3 jaar |

| Terugverdientijd | |
|--|--------|
| Hoe lang zal duren voor het project terug verdiend is? | 1 jaar |

Figuur 50 Financieel overzicht binnen het scherm 'business case'

4 Reflectie

Voor de stage begon merkte ik dat er redelijk wat stress aanwezig was bij mezelf. Doordat ik enkele meetings heb kunnen bijwonen voor de stage aanvatte, kreeg ik een beter beeld van de komende stageperiode en verdween het grootste deel van mijn stress.

Tijdens deze stage heb ik een platform ontwikkeld dat effectief gebruikt zal worden, iets wat ik tot nu toe op school nog niet gedaan heb. Wetende dat dit project door de klant gebruikt zal worden, zorgde voor een bijkomende druk die bij de eerdere projecten niet aanwezig was. Bovendien moest ik ook verantwoording afleggen ten opzichte van de persoon die investeert in dit project. Wekelijks zat ik samen met hem om de vooruitgang te overlopen.

Naast het verbeteren van mijn soft-skills ben ik ook technisch sterker geworden. Dit zowel in de gebruikte technologieën als in het afstemmen van de front- en backend in een applicatie. Het meest voorkomende probleem was dat ik niet genoeg had nagedacht over de verbinding tussen de twee of over de functionaliteit die geïmplementeerd diende te worden. Met als gevolg dat ik tijdens de eerste weken regelmatig stukken van de applicatie opnieuw geschreven heb, wat veel tijd kostte. Ik vergat bijvoorbeeld regelmatig dat iets dynamisch moest zijn, zoals een score berekenen op basis van aangeduide criteria, en gebruikte dan eerst de *templatingtaal* van Laravel in plaats van Vue.js. Na een tijdje werd mij duidelijk dat ik meer tijd moest investeren in het nadenken over wat ik ging implementeren en vooral de manier waarop alvorens dit ook effectief te doen. Achteraf gezien was dit zonder twijfel de betere manier van werken. Na verloop van tijd wist ik wat mogelijk was, en vooral wat niet mogelijk is, bij het gebruik van beide en maakte ik deze fout minder en minder.

Het project is vlot verlopen doordat er elke week een sprintplanning met een sprintdemo plaatsvond. Hierdoor hadden de klant en de bedrijfspromotor steeds de kans om feedback te geven op de

voortgang van het project. Bij aanvang van de stage hebben we de duurtijd van het project op tien weken geschat. Hoe meer de stage vorderde bleek dat deze tijdspanne niet voldoende was om alle *features* te verwerken. De twaalf weken van de stage zijn niet voldoende om dit project te voltooien. Daarom heb ik de kans gekregen om aan de stage nog twintig dagen vakantiewerk te koppelen. Zo kan ik een afgewerkt product afleveren aan de klant.

Ik heb tijdens deze stageperiode grotendeels zelfstandig gewerkt, ik ben van nature iemand die niet snel om hulp vraagt en eerder zelf naar een passende oplossing zoekt. Zeker binnen de context van applicatieontwikkeling vind ik dit een goede eigenschap omdat je zeer veel informatie kan vinden op het internet. Natuurlijk zijn er ook valkuilen verbonden aan die zelfstandigheid. Zo zou ik waarschijnlijk minder tijd verloren hebben als ik in het begin eerst even om raad vroeg over de manier van implementatie bij mijn bedrijfspromotor.

Naast het technische aspect van de stage vind ik dat ik ook een goede dosis zelfstandigheid toonde bij het plannen van de sprintplanning. Zo heb ik enkele keren de sprintplanning en demo zelfstandig gepland en geleid, zonder dat de bedrijfspromotor aanwezig was. Dat ik het vertrouwen kreeg om deze taken zelfstandig te plannen en uit te voeren heb ik zeker als een compliment ervaren.

Ik ben zeer tevreden dat ik deze opdracht gekozen heb omdat de opdracht zich perfect verleende om door mij voortgetrokken te worden. Er was bijvoorbeeld veel plaats voor mijn inbreng tijdens de wekelijkse meetings. Doordat ik dit mocht doen, heb ik het gevoel dat ik het project zeker positief beïnvloed heb, wat veel voldoening met zich meebrengt ten opzichte van een project dat al tot in de details gepland is.

II. Onderzoekstopic

1 Onderzoeksvraag

Is het mogelijk om automatische betalingen toe te voegen in een applicatie die met Laravel gemaakt is? Zo ja, welke providers bestaan er en welke is de beste keuze voor dit project?

Er zijn enkele criteria die in acht genomen worden om de beste keuze te maken. Een belangrijk criterium zijn de betalingsmogelijkheden die het populairst zijn in België en Nederland en de ondersteuning ervan, zoals Bancontact, kredietkaarten en overschrijvingen. Een ander criterium is de prijs, welke van de providers is het voordeligst per betaling als het over grotere bedragen gaat? Daarnaast is de moeilijkheidsgraad van de implementatie niet onbelangrijk. Een goede documentatie is een belangrijke maatstaf binnen dit criterium. Is er binnen de documentatie een tutorial die uitlegt hoe de service het best geïmplementeerd wordt?

Het implementeren van de automatische betalingen is grotendeels een voordeel voor de klant. Hij dient zich namelijk geen zorgen meer te maken over de continuïteit van het betalingssysteem. De bedrijven kunnen een licentie voor de applicatie aankopen via een webapplicatie in plaats van met manuele overschrijvingen te werken.

Om de service te kunnen gebruiken dient de provider van automatische betalingen compatibel te zijn met Laravel of PHP in het algemeen in de vorm van *packages* of *libraries* die ze aanbieden.

Om wat input te krijgen van onder andere de klant en frontend-developer stond de onderzoeksvraag op de agenda tijdens de eerste meeting van de stageperiode. Tijdens de meeting werden er enkele providers van automatische betalingservices voorgesteld zoals Selly.gg, MoneyBird, Stripe en Mollie.

Voor het begin van het onderzoek werd duidelijk dat Selly.gg en Moneybird geen goede kandidaten zijn voor Ideashooter. Selly.gg werkt vooral in samenwerking met Stripe en biedt enkel een extra betalingsmogelijkheid, namelijk het gebruik van *cryptocurrencies* zoals Bitcoin, Ethereum, Litecoin,... Het gebruik van *cryptocurrencies* is geen vereiste voor online betalingen binnen Ideashooter, waardoor het weinig zin heeft om Selly.gg verder te onderzoeken. Moneybird daarentegen is geen goede kandidaat omdat het een tool is om hulp te bieden bij boekhouding, en in mindere mate om daadwerkelijke online betalingen te verwerken.

Ter vervanging van Selly.gg en Moneybird gaat dit onderzoek Icpay behandelen. Icpay is ook een betaalplatform dat gebruikt wordt in België.

2 Onderzoeksmethode

Het onderzoek wordt uitgevoerd in twee verschillende fasen. Aan de ene kant is er een exploratie van de drie kandidaten: Mollie, Stripe en Icpay. Aan de andere kant is er een vergelijking van deze drie met in het achterhoofd de financiële instellingen in België. Om een vergelijking te maken omtrent de implementatie zijn er proof of concepts gemaakt aan de hand van de documentatie die de providers aanbieden.

3 Uitwerking onderzoek

3.1 Inleiding

3.1.1 Wat houdt een automatische betaling in?

In de context van dit onderzoek is een automatische betaling gedefinieerd als een betaling die de gebruiker via het internet verricht en die vervolgens op de rekening van de ontvanger komt. Tussen deze twee partijen zit een systeem dat de rest van de transactie afhandelt, zoals het controleren en aanvaarden van de betaling en het uitbetalen van het verkregen bedrag aan de verkoper.

Deze betalingen kunnen ook van terugkerende aard zijn wanneer het gaat om een bedrag dat bijvoorbeeld maandelijks betaald dient te worden. Zulke betalingen worden ook automatisch afgehandeld zonder dat de gebruiker of ontvanger tussentijds een handeling stelt.

3.2 Exploratie van de drie kandidaten

In dit deel gebeurt de exploratie van de drie gekozen providers van automatische betalingen voor dit onderzoek: Mollie, Stripe, en Icpay.

3.2.1 Mollie

De eerste provider die bekeken wordt is Mollie (Figuur 51), een bedrijf in Nederland met als doel ondernemingen efficiënter te helpen werken door een betaaloplossing aan te bieden [7].

The Mollie logo consists of the word "mollie" in a lowercase, blue, sans-serif font.

Figuur 51 Logo Mollie

3.2.1.1 Betalingsmogelijkheden

Mollie ondersteunt de meest voorkomende kredietkaarten zoals Mastercard, Visa en American Express. [7].



Figuur 52 Logo Bancontact

Mollie ondersteunt Bancontact (Figuur 52), een veelgebruikte betaalmethode [8] in België. Elke gebruiker die een bankkaart heeft van een actieve bank in België en aan online bankieren doet, kan gebruik maken van Bancontact bij Mollie.

Verschillende banken in België hebben zelf een mobiele applicatie waarmee een gebruiker kan betalen, en hebben een zogenaamde 'betaalknop'. Mollie biedt ook daarvoor ondersteuning, concreet ondersteunen ze de mobiele applicaties van KBC, Belfius en ING.



Figuur 53 Logo iDeal

In Nederland is iDeal (Figuur 53) de meest gebruikte betaalmethode waardoor Mollie ook dit aanbiedt als betaalmogelijkheid.

Naast Bancontact en iDeal kan de gebruiker ook via PayPal (Figuur 54) betalen. PayPal is een bedrijf uit de Verenigde Staten dat onder de noemer *wallet* valt. Een *wallet* is een virtuele portemonnee waar

zelf verschillende betalingsmogelijkheden aan gekoppeld kunnen worden. Zo kan bijvoorbeeld zowel een kredietkaart of bankkaart toegevoegd worden aan de *wallet*. [9] Hierdoor hebben gebruikers enkel een bankkaart of kredietkaart nodig om te kunnen betalen.



Figuur 54 Logo PayPal

Naast de reeds genoemde betalingsmogelijkheden biedt Mollie ondersteuning aan voor een waaier van minder bekende betalingsmogelijkheden zoals Klarna, SOFORT, EPS (Oostenrijk), GiroPay (Duitsland), Postepay (Italië) en paysafecard. Als de gebruiker geen van bovenstaande methoden ter beschikking heeft, biedt Mollie een SEPA-overschrijving aan. Dit zijn overschrijvingen die binnen Europa kunnen verricht worden. [10]

3.2.1.2 Kostprijs

Aan elke betaling die via een betalingssysteem gaat, hangt een prijskaartje. De kost van elke transactie hangt af van de methode die gebruikt wordt. Betaalt de gebruiker via Bancontact, dan kan dit duurder of goedkoper zijn dan bijvoorbeeld via Mastercard. Dit wordt onderzocht binnen dit onderdeel.

Bij Mollie geldt de regel dat elke betaling geld kost wanneer ze succesvol is, indien de betaling onsuccesvol is zal er ook niets aangerekend worden.

Veel prijzen zijn variabel bij Mollie, maar voor enkele methoden ligt de prijs volledig vast [11]. Zo kost Bancontact voor elke transactie €0,39 euro, iDEAL €0,29 en overschrijvingen of domiciliëring €0,25.

Andere methoden hebben bij Mollie altijd een vaste prijs gekoppeld met een variabele kost. Die variabele kost is uitgedrukt in procenten. Mastercard en Visa kosten bijvoorbeeld beide €0,25 plus een variabele kost van 1,8%. Op kleine bedragen kan dit voordelig zijn maar aangezien voor Ideashooter grotere bedragen betaald zullen worden dan in de gemiddelde webshop zal de variabele kost ook hoger oplopen. Voor een bedrag van €1000,00 zal de kostprijs van een transactie via Mastercard of Visa bij Mollie dus ongeveer €18,25 kosten. Deze kostprijs is berekend ervan uitgaande dat de klant een Europese, persoonlijke Mastercard of Visa gebruikt. Wanneer het een niet-Europese en/of zakelijke kredietkaart is bedraagt de variabele kost 2,8%. Daardoor bedraagt de transactiekost op hetzelfde bedrag €28,25.

3.2.1.3 Minimum- en maximumbedragen

Omdat Ideashooter met grotere bedragen zal werken is er ook een tabel toegevoegd met de minimum- en maximumbedragen die toegelaten zijn bij Mollie, zie tabel (Tabel 1) [12]. De bedragen in de tabel zijn vastgesteld door Mollie en kunnen verschillen per gebruiker, afhankelijk van zijn bank. Zo kan het voor komen dat de bank niet toe laat om betalingen te doen die groter zijn dan een bepaald bedrag. Indien nodig is kan er wel contact opgenomen worden met Mollie om te grens van de maximumbedragen te verhogen.

Tabel 1 Overzicht minimum- en maximumbedragen bij Mollie

| Betaalmethode | Minimumbedrag | Maximumbedrag |
|--------------------------------|---------------|---------------|
| Bancontact | €0,02 | €50.000 |
| Kredietkaart | €0,01 | €2.000 |
| iDEAL | €0,01 | €50.000 |
| Overschrijving | €0,01 | €1.000.000 |
| Paypal | €0,01 | €8.000 |
| Betaalknop Belfius, ING en KBC | €0,01 | €50.000 |

3.2.1.4 Support

Naast de betaalmogelijkheden en de kostprijs is de ondersteuning die Mollie biedt ook een belangrijke factor. Op de site van Mollie staat [13] dat er op twee manieren ondersteuning is voorzien namelijk via mail of telefonisch. Volgens de website wordt er binnen de twee werkdagen een antwoord gegeven, wat een vrij lange termijn is. Aan de andere kant kan via de telefoon sneller een antwoord bekomen worden. Het nadeel is dat Mollie enkel een betalend Nederlands nummer ter beschikking stelt. Dit wil zeggen dat de kostprijs van de ondersteuning oploopt naargelang de lengte van het telefoongesprek. Daarnaast is de helpdesk enkel beschikbaar tussen 09:00 uur en 18:00 uur van maandag tot vrijdag.

3.2.1.5 Betrouwbaarheid

Wanneer de service om te betalen niet werkt bestaat de kans dat potentiële klanten afhaken. Daarom is het belangrijk dat de service een *uptime* heeft van ongeveer 100%.

Op de website van Mollie is er een pagina die actuele en voorgaande storingen weergeeft [14]. Om een beeld te krijgen van de *uptime* is er gekeken naar storingen in de drie voorgaande maanden. Mollie heeft tijdens die drie maanden te maken gehad met één volledige storing van hun service die in totaal drie minuten duurde. Daarnaast zijn er enkele storingen geweest bij de betaalservices zoals Bancontact en iDeal, maar die worden buiten beschouwing gelaten omdat te maken hebben met de service van Mollie.

Een storing van drie minuten op een tijdsperiode van drie maanden betekent dat Mollie 99.997% *uptime* had.

3.2.2 Stripe

De volgende provider die onder de loep genomen wordt is Stripe (Figuur 55), een bedrijf dat gevestigd is in de Verenigde Staten.



Figuur 55 Logo Stripe

3.2.2.1 Betalingsmogelijkheden

In dit onderdeel gaan we dieper in op de betalingsopties die Stripe aanbiedt. Omdat de applicatie vooral in België en mogelijks in Nederland gebruikt zal worden is het belangrijk dat de betalingsopties die Stripe ter beschikking stelt ook bruikbaar zijn in deze landen.

Stripe is een bedrijf dat zich situeert in de Verenigde Staten en ondersteund dus Mastercard, Visa en American Express [15] betalingsmethoden. Naast de kredietkaarten biedt Stripe ook de mogelijkheid om via Bancontact te betalen.

Buiten de kredietkaarten en Bancontact biedt Stripe ook ondersteuning aan enkele mobiele *wallets*. Hiermee kan de gebruiker online of in fysieke winkels betalingen doen, wat vooral in de Verenigde Staten een groeiende markt is.



Figuur 56 Logo Apple Pay

Het probleem met zo'n *wallets* is dat ze dikwijls van bedrijven uit de Verenigde Staten komen waardoor er nog niet zo'n grote ondersteuning is in België. Stripe ondersteunt bijvoorbeeld Apple Pay (Figuur 56) en Google Pay (Figuur 57). Apple Pay is sinds kort beschikbaar in België maar kan enkel gekoppeld worden aan een bankkaart van BNP Paribas Fortis, Fintro en Hello Bank [16]. De rest van de betalingsmogelijkheden omvatten de bekende kredietkaarten zoals Mastercard en Visa. Google Pay krijgt ondersteuning van een extra bank, namelijk KBC.



Figuur 57 Logo Google Pay

Als de gebruiker zulke *wallets* wilt gebruiken is hij ook afhankelijk van het type besturingssysteem van zijn smartphone. Vervolgens is het niet zeker dat zijn bank ondersteund wordt door de *wallet*.

Stripe ondersteunt 135 verschillende valuta's. [17] Dit zorgt ervoor dat het grootste deel van de wereld hier gebruik van kan maken.

3.2.2.2 Kostprijs

Bij Stripe hangt ook aan elke transactie een prijskaartje. Er zijn enkele betaalmethoden die een vaste prijs hebben. Zo kost een transactie met iDEAL €0,45 en een SEPA-overschrijving €0,35. Andere betaalmethoden hebben een vaste prijs met een extra variabele kost in procenten. Zo kost een betaling met Bancontact, Mastercard of Visa €0,25 plus 1,4% extra kosten. Voor een bedrag van €1.000,00 euro zou de transactiekost €14,25 bedragen, aangenomen dat de transactie met een Europese kaart gebeurd is.

Indien het om Mastercard of Visa gaat, zijn steeds de Europese bedragen in rekening genomen.

3.2.2.3 Minimum- en maximumbedragen

Bij Stripe zijn er minder strenge restricties als het gaat om het minimum- en maximumbedrag van transacties. Het minimumbedrag dat per transactie nodig is hangt af van de gebruikte valuta. Wanneer de Euro als valuta gebruikt wordt is er een minimum van €0,50 en bij het gebruik van USD is dat \$0,50.

Het maximumbedrag bij Stripe is het getal 999.999,999, onafhankelijk van welke valuta gebruikt wordt voor de transactie [18].

3.2.2.4 Support

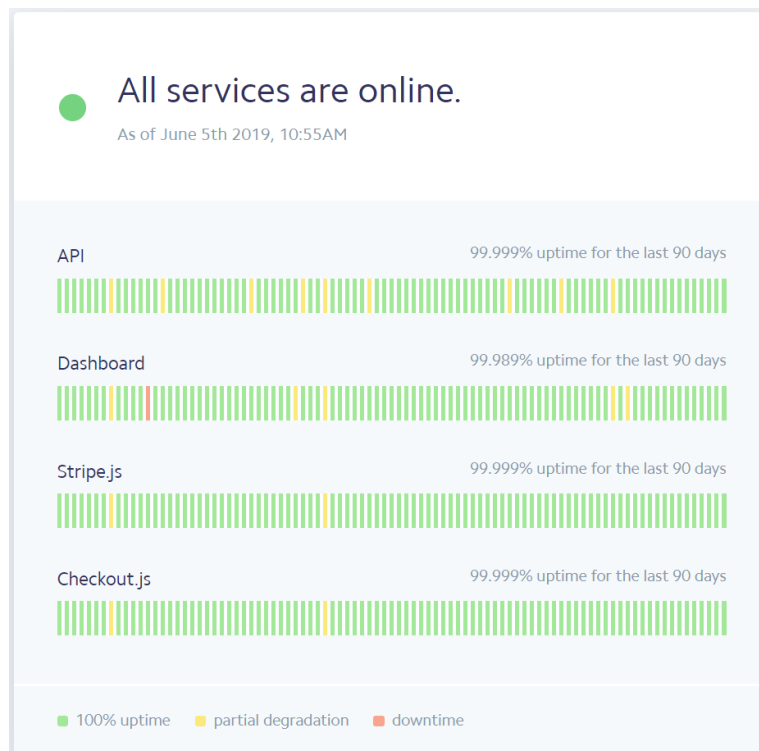
Voor ondersteuning bij problemen kan de ontwikkelaar via drie methoden terecht bij Stripe namelijk via e-mail, een individuele chatsessie met een medewerker en telefonisch [19]. Een antwoord op een e-mail kan volgens de site één dag duren. De tijd die een werknemer nodig heeft om telefonisch of via een chatsessie contact op te nemen wordt weergegeven op de website van Stripe. Op het moment van schrijven duurt het 3 minuten voor er iemand antwoord op een chatbericht en 5 minuten om een medewerker te laten bellen.

In een praktijktest duurt het 3 uur voor een medewerker een e-mail stuurt, minder dan 1 minuut om te antwoorden op een chatbericht en ongeveer 7 minuten om telefonisch contact op te nemen.

De ondersteuning van Stripe kan 24/7 gecontacteerd worden hoewel de wachttijden tijdens daluren, zoals 's nachts', langer kunnen zijn. Het is belangrijk om te vermelden dat alle ondersteuning van Stripe gratis is. De medewerkers van Stripe bellen een opgegeven nummer waardoor de ontwikkelaar zelf niet hoeft te bellen en dus ook geen kosten heeft.

3.2.2.5 Betrouwbaarheid

Net als Mollie is er bij Stripe een webpagina [20] waar actuele en voorafgaande storingen weergegeven worden. Tijdens de afgelopen drie weken was de service van Stripe een van 99,999% de tijd beschikbaar zoals weergegeven in Figuur 58. Het dashboard, dat onafhankelijk werkt van hun service, was één keer op drie maanden volledig onbereikbaar voor dertien minuten. Dit wil zeggen dat de gebruikers van Stripe de gegevens van betalingen en dergelijke niet konden raadplegen. De API heeft negen keer een storing meegemaakt die enkel zorgden voor gedeeltelijke degradatie. Een gedeeltelijke degradatie kan bijvoorbeeld een vertraging zijn tussen de client en de API.



Figuur 58 Status van alle services van Stripe

3.2.3 Icepay

Icepay is een Nederlands bedrijf dat ook actief is in België. Buiten software om online betalingen te verrichten bieden zij ook betaalterminals aan die zij de 'Point-of-Sale terminals' noemen, maar dat ligt buiten de *scope* van dit onderzoek.

3.2.3.1 Betaalmogelijkheden

Icepay biedt de meest voorkomende kredietkaarten aan zoals Mastercard, Visa en American Express.

Zoals de voorgaande bedrijven Mollie en Stripe biedt Icepay ook Bancontact aan om betalingen te verrichten.

Naast deze populaire betaalmogelijkheden zijn er ook enkele die specifiek voor bepaalde landen populair zijn, moest Ideashooter internationaal bekendheid winnen. Zo ondersteunt Icepay giro pay (Oostenrijk), JCB (Japanse kredietkaart), Nordea Bank (Noorse bank/kredietkaart), EPS (Oostenrijk), iDeal (Nederland) en Dankort (Denemarken).

3.2.3.2 Kostprijs

Zoals de voorgaande providers hanteert ook Icepay bij bepaalde betaalmogelijkheden een vaste kost en bij anderen een vaste plus een variabele kost [21].

Bij Icepay wordt er een onderscheid gemaakt tussen mensen met een 'Professional' of 'Enterprise' account. Gebruikers met een 'Professional' account hebben als richtlijn een transactievolume van minstens 500 transacties per maand, en een 'Enterprise' account 1250 transacties per maand. Naast de gradaties van de twee verschillende accounts rekent Icepay nog een extra kost aan als je binnen een bepaald bereik van het aantal transacties zit. Als het transactievolume tussen 0 en 1000 zit zal er

een extra kost van €0,10 per transactie aangerekend worden. Aangezien Ideashooter in het begin wellicht een lager transactievolume van 1000 transacties en zelfs 500 transacties per maand zal hebben, wordt er van uit gegaan dat er bij elke transactie een extra kost van €0,10 bijkomt. Daarnaast rekent Icpay maandelijks, ongeacht het transactievolume en aangenomen dat het om een 'professional' account gaat, 500 transacties met een kostprijs van €0,10 aan. Met andere woorden kost het gebruik van Icpay €50,00 euro per maand, ondanks dat de klant bijvoorbeeld maar 5 transacties in één maand te verwerken krijgt.

De kost van Bancontact ligt bij Icpay vrij hoog met €0,95 per transactie (bovenstaande €0,10 bijgeteld), maar is wel een vaste kost wat als gevolg heeft dat grotere bedragen van bijvoorbeeld €1.000,00 goedkoper zijn bij Icpay.

Voor kredietkaarten komt er buiten een vaste prijs die €0,25 bedraagt en een variabele kost van 1,9% ook een opstartkost bij als de maandelijkse omzet minder dan €25.000,00 euro bedraagt. Wanneer dat het geval is bedraagt de extra kost om kredietkaarten te betalen €100,00. Met bovengenoemde kosten zal de transactiekost €19,25 bedragen wanneer de gebruiker een transactie verricht met een bedrag van €1.000,00.

3.2.3.3 Minimum- en maximumbedragen

Bij Icpay is het maximumbedrag 10.000,00 dollar of euro voor kredietkaarten, iDEAL, Bancontact, Paypal, overschrijvingen, EPS en GiroPay. Het minimumbedrag is steeds 0,01 dollar of euro.

3.2.3.4 Support

Icpay biedt ondersteuning via e-mail en een chatsessie met een medewerker, telefonisch contact opnemen is niet mogelijk. De reactietijd via een chatsessie is minder dan twee minuten en via mail ongeveer 5 uur. Een chatsessie kan enkel gestart worden tussen 09:00 uur en 17:30 uur van maandag tot vrijdag. Tijdens de chatsessies kan er met een werknemer van "sales" gesproken worden die minder kennis hebben over de technische aspecten. Indien er een technische vraag of probleem is kan er via GitHub een issue aangemaakt worden.

De ondersteuning die Icpay biedt is gratis, ook wanneer de gebruiker nog geen account heeft.

3.2.3.5 Betrouwbaarheid

In tegenstelling tot Mollie en Stripe is er bij Icpay geen webpagina waar de actuele en voorgaande storings worden weergegeven. Tijdens een chatsessie met de medewerkers van Icpay werd aangegeven dat zij een *uptime* 100% hebben en dat er enkel storings plaatsvinden bij de betaalmethodes zelf. Een systeem dat 100% van de tijd naar behoren werkt is quasi onmogelijk en is daarom twijfelachtig.

3.3 Vergelijkingsfase

3.3.1 Prijsvergelijking

Om van start te gaan met de prijsvergelijking staat in Tabel 2 een overzicht van de transactiekosten per betaalmethode.

Tabel 2 Prijsvergelijking Mollie, Stripe en Icepay

| | Mollie | Stripe | Icepay* |
|--|----------------|---------------|-----------------|
| Bancontact | € 0,39 | € 0,25 + 1,4% | € 0,95 |
| Kredietkaart (Europa) | € 0,25 + 1,8 % | € 0,25 + 1,4% | € 0,25 + 1,9%** |
| PayPal | € 0,45 + 3,4% | / | € 0,20 + 3.4% |
| iDEAL (Nederland) | € 0,29 | € 0,45 | € 0,35 |
| SEPA-overschrijving/ domiciliëring (Europa) | € 0,25 | € 0,35 | € 5/maand |

*Icepay rekent minimum 500 transacties per maand aan ongeacht het transactievolume. Elke transactie bij Icepay kost €0,10 als het transactievolume per maand lager is dan 1000. Er wordt maandelijks dus minstens €50,00 aangerekend.

De kosten in bovenstaande tabel (onder Icepay) houden al rekening met de extra €0,10.

** Bij omzet lager dan €25.000,00 per maand kost het opzetten om te betalen met Mastercard en Visa €100,00.

Bovenstaande tabel geeft een overzicht van de kostprijs van elke transactie, afhankelijk van welke provider gebruikt wordt.

De prijs van Bancontact verschilt relatief veel per provider, een transactie via Bancontact kost bij Icepay meer dan het dubbele dan bij Mollie. Het zijn echter beide wel vaste prijzen, in tegenstelling tot Stripe. Doordat Stripe een meerprijs van 1.4% vraagt voor elke betaling met Bancontact, zal een transactie met grotere bedragen meer geld kosten.

Naast Bancontact verschilt de prijs bij het gebruik van kredietkaarten veel per transactie. Op een bedrag van €1.000,00 kost een transactie bij Mollie, Stripe en Icepay respectievelijk €18,25, €14,25 en €19,25.

Buiten deze twee betaalmethoden is er geen significant verschil tussen de transactiekosten.

3.3.1.1 Conclusie prijsvergelijking

Dat Icepay werkt op basis van een transactievolume per maand geeft weer dat de doelgroep klanten bevat die een relatief groot maandelijks transactievolume hebben. Als het quotum omtrent het transactievolume op het einde van de maand niet behaald is worden er extra kosten in rekening gebracht waar de klant weinig tot niets voor terug krijgt. Denk bijvoorbeeld aan een maand waar geen enkele transactie plaatsvindt, de klant betaalt voor de services van Icepay maar heeft er die maand niets aan. Daarnaast komt er een de maandelijks kost van €100,00 bij om het gebruik van kredietkaarten mogelijk te maken als de maandelijks omzet lager is dan € 25.000,00. Deze kosten zijn een groot nadeel bij het gebruik van Icepay.

Tussen Mollie en Stripe is het grootste verschil dat Mollie een vaste kost heeft voor Bancontact en Stripe een vaste plus variabele kost, wat in het voordeel van Mollie speelt. Aan de andere kant is de variabele kost voor een aankoop met een kredietkaart lager bij Stripe dan bij Mollie. Het is moeilijk om een objectieve en eenduidige keuze te maken tussen deze twee als de meest gebruikte betaalmethode

onbekend is. Als er veel Bancontact gebruikt wordt zal Mollie de betere keuze zijn, indien de kredietkaart het meest gebruikt wordt, neigt de keuze eerder naar Stripe.

3.3.2 Vergelijking van support

Tabel 3 Vergelijking van support

| | Media | Gratis | Contactmomenten |
|--------|---------------------------|---|-----------------------------|
| Stripe | Telefonisch, chat, e-mail | Ja | 24/7 |
| Mollie | Telefonisch, e-mail | Telefoongesprek betalend (internationaal) | 9:00 tot 18:00 op werkdagen |
| Icepay | Chat, e-mail | Ja | 9:00 tot 17:30 op werkdagen |

3.3.2.1 Conclusie vergelijking support

Als het over ondersteuning gaat is Stripe de duidelijke winnaar. Stripe biedt de meeste mogelijkheden om contact op te nemen en er zijn 24/7 medewerkers ter beschikking om vragen te beantwoorden. De reactietijd die ze bieden is zeer kort als het gaat om een telefonisch gesprek of een chatsessie. Daarnaast is alle ondersteuning gratis.

3.3.3 Vergelijking betrouwbaarheid

Tabel 4 Vergelijking uptime

| | Uptime |
|--------|---------|
| Stripe | 99,999% |
| Mollie | 99,997% |
| Icepay | 100% |

3.3.3.1 Conclusie betrouwbaarheid

De *uptime* van de drie providers ligt zeer dicht rond de gewenste 100% en bijgevolg verschillen de percentages tussen de drie zeer weinig. Omdat de *uptime* van Icepay quasi onmogelijk is en er gebrek is aan meer informatie bestaat er twijfel over de juistheid van deze cijfers. Daarom wordt de *uptime* van Icepay tijdens deze conclusie buiten beschouwing gehouden.

Theoretisch is Stripe de beste qua betrouwbaarheid, hun *uptime* is 0,002% hoger dan die van Mollie wat neerkomt op 2.18 minuten per drie maanden. In de praktijk zal dit verschil echter nauwelijks merkbaar zijn.

3.3.4 Proof of concept







Om een idee te krijgen van de moeilijkheidsgraad van de implementatie zijn er twee proof of concepts gemaakt. De eerste illustreert het gebruik van Mollie en de andere dat van Stripe.

3.3.4.1 Mollie

Om gebruik te kunnen maken van Mollie in een applicatie dient er eerst een account aangemaakt te worden. De enige informatie die een gebruiker daarvoor nodig heeft is een e-mailadres, voor- en

achternaam en een wachtwoord. Van zodra deze informatie opgegeven is, wordt er een account gemaakt waarmee de service getest kan worden; met dit soort account kunnen er geen echte betalingen uitgevoerd worden. Om het testaccount op te waarderen tot een volledig account dient de gebruiker meer informatie op te geven zoals de naam van het bedrijf, het adres, een btw-nummer,... Aangezien er nog geen vennootschap bestaat voor Ideashooter en er nog geen keuze is gemaakt tussen Mollie of Stripe, kan het testaccount niet opgewaardeerd worden voor deze PoC. Mollie biedt een dashboard aan dat een overzicht bevat van informatie die handig kan zijn voor de gebruiker. Zo is er een overzicht van bijvoorbeeld de orders die geplaatst zijn en hun status (Figuur 59) en de betalingen die reeds gebeurd zijn of nog open staan (Figuur 60).

Example

| Method | Amount | Status |
|---|---------|---------|
|  | 34,50 € | PAID |
|  | 29,99 € | EXPIRED |
|  | 59,75 € | OPEN |
|  | 99,50 € | PAID |
|  | 59,75 € | PAID |
|  | 99,50 € | OPEN |

Figuur 59 Overzicht bestellingen binnen Mollie

Example

| | | |
|------------------------|---------------|----------|
| 2× Example product | COMPLETED | 100,00 € |
| Hide product details ^ | | |
| Summary | | |
| Status | COMPLETED | |
| Unit price | 50,00 € | |
| VAT amount | 21,00 € (21%) | |
| Total amount | 100,00 € | |
| Example product | PAID | 50,00 € |
| Show product details v | | |

Figuur 60 Overzicht betalingen binnen Mollie

Mollie heeft een API-client voor PHP ervoor zorgt dat het implementeren van de betaling eenvoudig verloopt. De documentatie verwijst naar een *repository* op GitHub [12] hier kan men meer uitleg terugvinden over het gebruik van de client. Er zijn enkele vereisten voor het gebruik van de API: de gebruiker moet een account hebben, PHP 5.6 of hoger gebruiken en het verkeer tussen de API en client moet over een beveiligde verbinding (HTTPS) gebeuren. Nadat aan deze vereisten voldaan is, kan de gebruiker het pakket installeren met behulp van Composer. Dit is pakketmanager waarmee PHP-pakketten toegevoegd kunnen worden aan een project. Het commando dat daarvoor zorgt is `composer require mollie/mollie-apiphp:^2.0`.

Nadat het pakket ingeladen is, kan het gebruikt worden door aan het begin van de klasse een *use* statement te plaatsen met een verwijzing naar `Mollie\Api\MollieApiClient`. Daarna kan een betaling aangemaakt worden in de code door een nieuwe methode te schrijven zoals in Figuur 61 beschreven. Voor elke client waar een betaling aan gekoppeld wordt, dient de API-key ingesteld te worden.

```
function payWithCreditCard()
{
    $mollie = new MollieApiClient();
    $mollie->setApiKey(ApiKeys::API_KEY);

    $payment = $mollie->payments->create([
        'amount' => [
            'currency' => 'EUR',
            'value' => '2000.00'
        ],
        'description' => "Ideashooter - One year",
        "redirectUrl" => "https://03ba350f.ngrok.io/result",
        'method' => \Mollie\Api\Types\PaymentMethod::CREDITCARD,
    ]);

    return Redirect::to($payment->getCheckoutUrl(), 303);
}
```

Figuur 61 Methode om betaling aan te maken voor Mollie

Zo'n key is anders voor elk account en kan men terugvinden op het dashboard van Mollie. Daarna kunnen de details zoals het bedrag, de valuta en de betaalmethode meegegeven worden aan een nieuwe betaling. De gekozen betaalmethode dient geactiveerd te zijn op het dashboard van Mollie. Om dit te doen dient er enkel een vinkje aangeduid te worden en kan de eigenaar extra informatie invullen zoals de producten waarvoor geld ontvangen zal worden het maximum toegelaten bedrag (Figuur 62).

Bancontact

0,39 € per transaction

[Pricing & info](#)

Credit Card

[Pricing & info](#)

Please ensure your website contains all legally required policy information, shipping and billing information, and relevant pricing. Your customers need to be able to fill out their contact details during the order process. Be aware that your consumer can initiate a chargeback.

I've read and accept the text above

What will be the maximum amount per Credit Card transaction?

€

10000

Please briefly describe for what kind of products or services you want to receive Credit Card payments

Webapplication that enables employees to share ideas with the management

Figuur 62 Activeren van betalingsmethoden in Mollie

Zoals reeds vermeld in de vereisten, is het verplicht om de *requests* over een beveiligde verbinding te versturen. Omdat het om een *proof of concept* gaat, is er gekozen om Ngrok [22] te gebruiken. Ngrok is een tool om snel een tunnel op te zetten zodat applicaties die lokaal gehost worden van buitenaf bereikbaar zijn. Een bijkomend voordeel is dat de applicatie ook via HTTPS te bereiken is, anders zou het pakket van Mollie een error weergeven.

Indien de betaling succesvol is zal de gebruiker op de pagina van Mollie terechtkomen die instaat voor het voltooien van de betaling. Aangezien er met een testaccount gewerkt wordt, is het niet mogelijk om het scherm te zien waar de gebruiker terecht zou komen als het om een echte betaling gaat. Om toch een idee te geven van de vormgeving van zo'n scherm is Figuur 63 toegevoegd.

Berlin Bicycle € 864,00
Veloretti

Credit card payment

Credit card number: 0000 0000 0000 0000 Expiration date: 01 / 20

Cardholder name: Giel Cobben CVV number: 123

Continue payment

Figuur 63 Betaling met kredietkaart via Mollie

Figuur 64 geeft weer hoe het scherm er uit ziet met het testaccount. Buiten de algemene informatie over de betaling is er ook een sectie waar de ontwikkelaar kan kiezen welke status de betaling krijgt.

Ideashooter - One year 2.000,00 €
Ideashooter

Credit card

Op deze pagina kunt u de definitieve betalingsstatus van de testbetaling selecteren.

| | |
|-------------------------|---|
| BESCHRIJVING: | Ideashooter - One year |
| BEGUNSTIGDE: | Ideashooter |
| KAARTNUMMER: | 0000 0000 0000 6787 |
| KAARTHOUDEER: | J. Kempeneers |
| CVC: | *** |
| VERLOOPDATUM: | 05 / 2019 |
| STATUS VAN UW BETALING: | <input type="radio"/> Open <input type="radio"/> Betaald <input type="radio"/> Mislukt <input type="radio"/> Geannuleerd <input type="radio"/> Verlopen |

Ga verder >

Figuur 64 Betaling met kredietkaart via Mollie met testaccount

Als de gebruiker verder gaat zal hij verwezen worden naar de pagina die de ontwikkelaar heeft opgegeven (Figuur 61). De status en andere informatie van de betaling kan altijd opgevraagd worden (Figuur 65) wanneer de ontwikkelaar dit wil. Zo kan hij afhankelijk van de status die de betaling heeft vervolgtacties programmeren zoals een succespagina weergeven.

```
$payment = $mollie->payments->get($paymentId);
```

Figuur 65 Code om informatie over betaling op te vragen bij Mollie

3.3.4.2 Stripe

Net zoals voor Mollie moet er ook eerst een account aangemaakt worden alvorens Stripe gebruikt kan worden. De registratie is even eenvoudig als die van Mollie, de enige informatie die Stripe voorlopig vraagt is een e-mailadres, naam, achternaam en een wachtwoord. Ook bij Stripe is het zo dat het account net na de registratie nog een testaccount is, als de gebruiker echte betalingen wil uitvoeren moet hij extra informatie opgeven zoals de naam van de vennootschap, een btw-nummer, een rekeningnummer,...

Net als Mollie heeft Stripe ook een dashboard waar informatie wordt weergegeven over de betalingsstatus (Figuur 66) en andere gegevens zoals de bestellingen, het saldo, de klanten,...

| AMOUNT | DESCRIPTION | CUSTOMER | DATE |
|-------------|-----------------------------|-----------------------|-----------------|
| \$10.99 USD | pi_1EPqnDEOUkReD1hKhwtFSPL2 | | Apr 16, 2:53 PM |
| \$10.99 USD | pi_1EPqnCEOUkReD1hK174D6fMz | | Apr 16, 2:53 PM |
| \$5.00 USD | pi_1EPqXOEOUkReD1hKlgje1Ea3 | fdfqdsfdqsf@gmail.com | Apr 16, 2:42 PM |

Figuur 66 Overzicht van betalingen binnen Stripe

In de documentatie [23] van Stripe kan men een duidelijke tutorial terugvinden. Deze tutorial definieert twee stappen die de ontwikkelaar kan ondernemen om gebruik te maken van het betalingssysteem. Telkens wanneer men een stuk voorbeeldcode toont in de documentatie kan de lezer kiezen in welke programmeertaal hij wenst verder te gaan. Stripe geeft de keuze tussen Ruby, Python, PHP, Java, Node, GO en .NET.

De eerste stap omvat het aanmaken van een sessie. Voor zo'n sessie te creëren dient eerst het pakket van Stripe in het project geladen te worden, dit wordt helaas niet vermeld in de documentatie. Het correcte pakket is echter eenvoudig te vinden, er is namelijk net zoals bij Mollie een repository op GitHub [24] waar extra informatie te vinden is over het installeren en implementeren van het pakket. Het pakket inladen kan in dit geval via Composer met het commando `composer require stripe/stripe-php`.

Daarna kan de documentatie van stripe verder geraadpleegd worden om een sessie aan te maken. In Figuur 67 staat de methode die nodig is om een betalingssessie aan te maken. Net zoals bij Mollie kan hier het bedrag en andere nuttige informatie over de transactie opgegeven worden. Het is belangrijk om te weten dat de *amount* in eurocent is, het getal 1000 zou dus weergegeven worden als €10,00 op het betalingsscherm dat de gebruiker ziet. Aan het begin van de URL's wordt het protocol 'HTTP' weergegeven in plaats van 'HTTPS', wat duidt op verbinding die niet beveiligd is. Stripe zal dit toelaten wanneer de applicatie een *test-key* gebruikt, in de andere gevallen zal de applicatie een error weergeven.

```

function pay() {
    Stripe::setApiKey("sk_test_btQ0S8DTxusx0IcPJWi2mnu4007LDXV82f");

    $session = Session::create([
        'success_url' => 'http://localhost:8000',
        'cancel_url' => 'http://localhost:8000/cancel',
        'payment_method_types' => ['card'],
        'line_items' => [[
            'amount' => 200000,
            'currency' => 'eur',
            'name' => 'Ideashooter - One Year',
            'description' => 'The easiest way to manage ideas',
            'quantity' => 1,
        ]]
    ]);

    return view('pay', ['session' => $session->id]);
}

```

Figuur 67 Methode om betaling aan te maken voor Stripe

Nadat de betaling aangemaakt is, zal de methode een view met daarin het *id* van de sessie doorgeven. In deze view is de tweede stap geïmplementeerd.

In de tweede stap wordt de kant van de client geïmplementeerd in Javascript. Omdat in Laravel het gebruik van Vue.js aangemoedigd wordt, is er gekozen om daarvan gebruik te maken om de frontend af te handelen. Ook dit keer schiet de tutorial te kort in het vermelden van de nodige pakketten voor het gebruik van Stripe. Desondanks is het eenvoudig om de nodige stap te vinden op de site van NPM [25], het commando is namelijk `npm install stripe`.

In een nieuw Vue-component is de code (Figuur 68) van de tutorial geïmporteerd. Buiten de API-key en het *id* van de sessie is er geen verschil tussen de code in Figuur 68 en de code van de tutorial.

```

props: {
    sessionId: String,
},
stripe: Stripe('pk_test_t1K7fy8Yl48D4AiiUyTcPKSn00EknC1Kgi'),
error: String,
pay() {
    this.stripe.redirectToCheckout({
        sessionId: this.sessionId
    }).then(function (result) {
        this.error = result.error.message;
    });
},

```

Figuur 68 Methode om in de frontend de gebruiker naar de checkout van Stripe te sturen

Het enige wat men nu nog moet doen, is het script inladen dat Stripe ter beschikking stelt zoals in Figuur 69 beschreven staat.

```
<script src="https://js.stripe.com/v3/"></script>
```

Figuur 69 Inladen van script om functionaliteit van Stripe te kunnen gebruiken

De methode `'pay()'`, weergegeven in Figuur 68, kan daarna bijvoorbeeld gelinkt worden aan een knop om zo de betaling te initiëren. De gebruiker zal naar een betalings scherm van Stripe gestuurd worden waar hij zijn gegevens kan ingeven en de betaling kan afronden (Figuur 70).

The screenshot shows a Stripe payment interface. On the left, the product is 'Ideashooter - One Year' for € 2.000,00, described as 'The easiest way to manage ideas'. The user is identified as 'Jonas Kempeneers' with a 'TEST' tag. On the right, the 'Met kaart betalen' (Pay with card) form is visible. It includes an email field with 'jan.janssens@gmail.com', a card information field with card number '5555 5555 5555 4444', expiration date '09 / 20', and CVV '453'. The name on the card is 'Jan Janssens' and the region is 'België'. A blue button at the bottom right says 'Betaal € 2.000,00' with a lock icon. At the bottom left, it says 'Powered by stripe' with links for 'Voorwaarden' and 'Privacy'.

Figuur 70 Betalingscherm van Stripe

Het is ook mogelijk om bij Stripe een terugkerende betaling op te zetten. Hiervoor dient men eerst een 'plan' aan te maken, dit kan met de code beschreven in Figuur 71.

```
Plan::create([
  "amount" => 20000,
  "interval" => "month",
  "product" => [
    "name" => "Ideashooter"
  ],
  "currency" => "eur",
  "id" => "ideashooter"
]);
```

Figuur 71 Methode om een plan aan te maken bij Stripe

Vervolgens kan men het aangemaakte plan gebruiken in de methode om een betaling op te zetten, zoals Figuur 72 weergeeft.

```
$session = Session::create([
  'payment_method_types' => ['card'],
  'subscription_data' => [
    'items' => [[
      'plan' => 'ideashooter',
    ]],
  ],
  'success_url' => 'http://localhost:8000',
  'cancel_url' => 'http://localhost:8000/cancel',
]);
```

Figuur 72 Methode om terugkerende betaling op te zetten bij Stripe

Het betalingsscherm waar de gebruiker voor de betaling terechtkomt ziet er dan ook iets anders uit ten opzichte van het scherm bij een enkele betaling (Figuur 73).

Figuur 73 Betalingsscherm voor terugkerende betalingen bij Stripe

3.3.4.3 Vergelijkingsmatrix PoC

Tabel 5 Vergelijking PoC's

| | PHP-package beschikbaar | Handleiding | Testaccount | Dashboard |
|---------------|----------------------------|-------------|-------------|-----------|
| Stripe | Ja | Ja | Ja | Ja |
| Mollie | Ja | Ja | Ja | Ja |

3.3.4.4 Conclusie PoC

Zowel Stripe als Mollie stellen een *package* ter beschikking om hun service te implementeren in PHP. Beide providers hebben een handleiding die duidelijk weergeeft wat een ontwikkelaar kan doen om de service te implementeren. Daarnaast bieden ze beide een testaccount ter beschikking zodat de ontwikkelaar de implementatie eerst kan testen voordat hij een keuze maakt.

Doordat ze beide voldoen aan de criteria die nodig zijn binnen dit project is het moeilijk om een winnaar aan te duiden als het gaat over de PoC's. Er is geen significant verschil tussen beiden.

Conclusie

A. Samenvatting vergelijkingsmatrices

Tabel 6 Samenvatting van alle vergelijkingen

| | Prijzen | Support | Uptime | PoC |
|--------|---|--|-----------|--|
| Stripe | Goedkoper bij gebruik kredietkaart | 24/7 gratis ondersteuning via e-mail, chatsessies en telefonisch | 99,999% | Heeft PHP-pakket en handleiding voor implementatie |
| Mollie | Goedkoper bij gebruik Bancontact | Tussen 9:00u en 18:00u op werkdagen, via e-mail en telefonisch. Telefoongesprek (internationaal) betalend | 99,997% | Heeft PHP-pakket en handleiding voor implementatie |
| Icepay | Meerdere extra kosten en duurste bij gebruik kredietkaarten en Bancontact | Tussen 9:00u en 17:30u op werkdagen, gratis, via e-mail en chatsessies | 100%(?) / | |

B. Conclusie

Tijdens de vergelijking van de prijzen per transactie valt op dat Icpay meerdere extra kosten aanreken naast een enkele transactiekost. Voor het gebruik van kredietkaarten rekenen ze €100,00 aan wanneer de maandelijkse omzet lager is dan €25.000,00. Daarnaast rekenen ze elke maand 500 transacties van €0,10 aan ongeacht het transactievolume. Stripe en Mollie daarentegen rekenen geen maandelijkse of andere kosten aan buiten de kost per transactie. Stripe en Mollie liggen dicht bij elkaar wat de prijs betreft. Stripe is vooral duur als het om Bancontact gaat maar wel goedkoper bij transacties met een kredietkaart. Bancontact wordt meer door particulieren dan bedrijven gebruikt en zal hier dus minder relevant zijn. Aangenomen dat de meeste transacties met een kredietkaart of overschrijving gebeuren geniet Stripe de voorkeur.

Ook bij de ondersteuning heeft Stripe de bovenhand, zij bieden gratis ondersteuning die 24 uur per dag, zeven dagen op zeven, beschikbaar is. Daarnaast is er een ruime keuze tussen de kanalen waarmee contact opgenomen wordt. Mollie daarentegen biedt de mogelijkheid om door middel van een e-mail of telefonisch contact op te nemen via een betalend telefoonnummer. Hun ondersteuning is enkel beschikbaar tussen 09:00u en 18:00u op werkdagen. Bij Icpay kunnen de klanten contact opnemen via e-mail en chat die beschikbaar is tussen 09:00u en 17:30u op werkdagen.

Theoretisch wint Stripe ook op het aspect van *uptime*, hoewel het verschil met Mollie zeer miniem is. Icpay claimt dat hun service 100% van de tijd beschikbaar is. Omdat 100% *uptime* zo goed als onmogelijk is wordt Icpay op dit vlak buiten beschouwing gehouden.

Door de reeds genoemde is Icpay geen goede kandidaat voor dit project, vervolgens is hun service niet getest op het vlak van implementatie. Tussen Mollie en Stripe is er naar implementatie geen significant verschil, ze stellen beiden een PHP-pakket en handleiding ter beschikking om hun services te kunnen gebruiken.

Doordat Stripe op verschillende vlakken de bovenhand heeft, ervan uitgaande dat de klanten vooral met kredietkaart en overschrijvingen betalen, is de conclusie dat binnen dit project Stripe de betere keuze is.

Bibliografie

- [1] „Wikipedia,” Wikipedia, [Online]. Available: https://nl.wikipedia.org/wiki/File_transfer_protocol. [Geopend 08 03 2019].
- [2] „JSON,” [Online]. Available: <https://www.json.org/>. [Geopend 05 06 2019].
- [3] „Object-relational mapping - Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Object-relational_mapping. [Geopend 05 06 2019].
- [4] JWT.io, „JWT,” [Online]. Available: <https://jwt.io/>. [Geopend 29 03 2019].
- [5] Stackover, „Stack Overflow Developer Survey 2018,” StackOverflow, [Online]. Available: <https://insights.stackoverflow.com/survey/2018/>. [Geopend 29 03 2019].
- [6] „Installation - Laravel,” Laravel, [Online]. Available: <https://laravel.com/docs/5.8/installation>. [Geopend 29 03 2019].
- [7] „Mollie.be,” Mollie, [Online]. Available: <https://www.mollie.com/be/payments>. [Geopend 10 03 2019].
- [8] Mollie, „Mollie - Snel en betrouwbaar Bancontact betalingen accepteren,” Mollie, [Online]. Available: <https://www.mollie.com/be/payments/bancontact>. [Geopend 10 06 2019].
- [9] „PayPal,” PayPal, [Online]. Available: www.paypal.com. [Geopend 10 03 2019].
- [10] „Sepa Belgium,” [Online]. Available: <http://www.sepabelgium.be/nl/wat-sepa>. [Geopend 10 03 2019].
- [11] Mollie, „Mollie - Payments - Eén Integratie, Alle Grote Betaalmethoden,” Mollie, [Online]. Available: <https://www.mollie.com/be/payments>. [Geopend 10 06 2019].
- [12] Mollie, „Wat is het minimum- en maximumbedrag per betaalmethode? – Mollie Support,” Mollie, [Online]. Available: <https://help.mollie.com/hc/nl/articles/115000667365-Wat-is-het-minimum-en-maximumbedrag-per-betaalmethode->. [Geopend 14 05 2019].
- [13] Mollie, „Mollie – Contact,” Mollie, [Online]. Available: <https://www.mollie.com/nl/contact>. [Geopend 10 06 2019].
- [14] Mollie, „Mollie Status,” Mollie, [Online]. Available: <https://status.mollie.com/>. [Geopend 05 06 2019].
- [15] „Stripe,” Stripe, [Online]. Available: <https://stripe.com/be/payments/features#payment-options>. [Geopend 10 03 2019].
- [16] „Apple,” Apple, [Online]. Available: <https://www.apple.com/benl/apple-pay/>. [Geopend 10 03 2019].
- [17] „Stripe,” Stripe, [Online]. Available: <https://stripe.com/docs/currencies>. [Geopend 10 03 2019].

- [18] Stripe, „Supported Currencies | Stripe,” Stripe, [Online]. Available: <https://stripe.com/docs/currencies#minimum-and-maximum-charge-amounts>. [Geopend 14 05 2019].
- [19] Stripe, „Contact Us | Stripe: Help & Support,” Stripe, [Online]. Available: <https://support.stripe.com/contact>. [Geopend 10 06 2019].
- [20] Stripe, „Stripe System Status,” Stripe, [Online]. Available: <https://status.stripe.com/>. [Geopend 05 06 2019].
- [21] Stripe, „Stripe: Pricing & fees,” Stripe, [Online]. Available: <https://stripe.com/pricing>. [Geopend 10 06 2019].
- [22] Ngrok, „Ngrok - secure introspectable tunnels to localhost,” Ngrok, [Online]. Available: <https://ngrok.com>. [Geopend 01 05 2019].
- [23] Stripe, „Stripe Checkout Quickstart,” Stripe, [Online]. Available: <https://stripe.com/docs/payments/checkout/server>. [Geopend 02 05 2019].
- [24] Stripe, „stripe/stripe-php: PHP library for the Stripe API.,” Github, [Online]. Available: <https://github.com/stripe/stripe-php>. [Geopend 02 05 2019].
- [25] I. npm, „npm,” npm, Inc, [Online]. Available: <https://www.npmjs.com/package/stripe>. [Geopend 02 05 2019].
- [26] „Digitale versnelling,” Telenet, [Online]. Available: <https://digitaleversnelling.be/tipstricks/e-commerce/4-onmisbare-online-betaaloplossingen-laten-klanten-betalen-glimlach/>. [Geopend 16 03 2019].
- [27] „Icepay,” Icepay, [Online]. Available: <https://icepay.be/betaalmethoden/maestro-creditcard/>. [Geopend 16 03 2019].