**Projectplan**

**Betaalfunctie Digendo**

Datum: 15-04-2019

Versie: 1

Door: Remco van der Linden

Klas: IC16AO.B

Begeleidend docent: Rico van Dooren

|  |  |  |
| --- | --- | --- |
| Naam: Rico van Dooren | Naam: Frans van Duijnhoven | Naam: Remco van der Linden |
| Functie: Docent | Functie: Bedrijfsleider | Functie: Student |
| Datum: | Datum: | Datum: |
| Goedgekeurd (handtekening) | Goedgekeurd (handtekening) | Goedgekeurd (handtekening) |

**Goedkeuring**

Inhoud

[01 Inleiding 3](#_Toc6386018)

[02 Opdracht 3](#_Toc6386019)

[03 Doelstelling 3](#_Toc6386020)

[04 Afspraken 4](#_Toc6386021)

[05 Grenzen 4](#_Toc6386022)

[06 Plan van Aanpak 4](#_Toc6386023)

[Haalbaarheidstappen 6](#_Toc6386024)

[07 Eisen en planning (WorkBreakDown) 6](#_Toc6386025)

[08 Over dit document 7](#_Toc6386026)

[Afkortingen 7](#_Toc6386027)

[Afkorting 7](#_Toc6386028)

[Omschrijving 7](#_Toc6386029)

[Referenties 7](#_Toc6386030)

[Definities 7](#_Toc6386031)

[Gebruikte materialen 7](#_Toc6386032)

[09 Bijlagen 7](#_Toc6386033)

# 01 Inleiding

Mijn naam is Remco van der Linden en ik ben een student aan het ROC Ter Aa ter Helmond. Dit projectplan is een onderdeel van mijn Proeve van bekwaamheid. Ik loop stage bij het bedrijf Digendo en ben daar stagiair web development. Voor die reden heb ik besloten om een functie te ontwikkelen voor het vak web development.

Dit document is het projectplan dat deel uitmaakt van de proeve van bekwaamheid en maakt deel uit van P1-K1 werkproces 1, in het projectplan kun je de opdracht , doelstellingen, afspraken en het plan van aanpak terugvinden.

# 02 Opdracht

De opdracht die gemaakt zal worden heeft te maken met het onderdeel betalingen verrichten online, maar gedeeltelijk ook om data op te halen en dit te verwerken naar een product waar de data goed gebruikt kan worden door bedrijven en klanten. Voor elke betaalmethoden is er een of meerdere API’s. Het aanleggen van elke API apart kost veel tijd en is voor het grootste deel zinloos om te doen. Voor deze redenen zal er gebruik gemaakt worden door een beheerde API. Deze API staat ook wel bekend als Mollie.

Mollie is een partij die zich gespecialiseerd heeft in het creëren van een duidelijke en eenvoudige API, die toch goed werk verricht en het gebruik van meerdere API’s eruit sloopt en zorgt dat je met een API alle bekende betaalmethodes kunt aanroepen via de instellingen in je dashboard( als je bepaalde betaalmethodes wilt uitschakelen bijvoorbeeld).

# 03 Doelstelling

Mijn doelstellingen van dit project zijn als volgt:

Voor mij is het belangrijk dat ik kan laten zien dat ik in de praktijk ook werkzaamheden kan verrichten die te maken hebben met dit vak en een product maak dat functioneert en in de praktijk ook gebruikt kan worden. Dataverkeer kunnen uitlezen en het opslaan is een belangrijk deel van het maken van een website die gebruik maakt van een database en voor die reden is het ook belangrijk dat ik kan leren hoe ik het gebruik van inner/outer/left/right-joins in het gebruik van Laravel DB query’s begrijp en dit in de praktijk kan gebruiken.

Het gebruik van data models en het schrijven van functies daar in, is iets wat ik graag zou willen leren tijden het ontwikkelen van het project. Hiermee kun je makkelijk query’s bouwen en snel invoegen tot de database. Vragen en wensen van de bedrijven of klanten kunnen verwerken naar een werkend product is ook iets wat ik graag wil leren bij de ontwikkeling van dit project.

Daarnaast vind ik het zelf belangrijk dat ik leer werken met de documentatie van een framework dat live is gezet door de makers, deze informatie is te vinden online en is duidelijk wanneer je het begrijpt. Daarnaast wil ik leren hoe ik problemen kan oplossen door informatie op te zoeken via een zoekopdracht of een andere methode

Project georiënteerde doelstellingen:

Deze doelstellingen zullen onderdeel uitmaken van het product dat ik zal maken en de doelstellingen die ik daarmee wil behalen.

De eerste doelstelling die ik graag wil halen is het ontwikkelen van een werkende betaalfunctie die ook werkt naar eisen en wensen van de projectleider en andere personen die invloed kunnen uitdrukken, het schrijven van code die leesbaar is en gemaakt voor het uitvoeren aanpassingen is een onderdeel die terug moet komen in het product en ook heel belangrijk is voor de toekomende tijd.

Het opslaan van data is ook een belangrijk onderdeel voor het product, hiermee kan er veel informatie beschikbaar gesteld worden voor de klanten en bedrijven die daarbij behoren en zullen voor die reden ook opgeslagen worden in de database en beveiligd dat alleen de personen waar die informatie voor bedoelt is beschikbaar wordt gesteld. Voor het gebruiken van mollie zullen er tokens gemaakt worden om een connectie te mogen maken met mollie, dat maakt het gebruik makkelijker voor betalingen en het betalen van de bedrijven makkelijker te maken.

# 04 Afspraken

Digendo is een website die geschreven is via het Laravel-framework, dit is de reden waarom Mollie via Laravel controllers, models en methodes die bekend zijn binnen dat framework werken. Natuurlijk is het verplicht dat er niemand de functie kan betreden door een fout te maken bij het betalen of annuleren van een reservering, hiervoor zal de website getest moeten worden op het onderdeel fool-proof. Als dit niet gebeurt zullen de andere onderdelen niet getest moeten worden en wordt het de prioriteit om dit probleem op te lossen of herbouwen.

De data die de website teruggestuurd krijg, zal verwerkt en opgeslagen worden in de toebehorende databases. Hierin zal de informatie opgeslagen worden en zal op deze manier gelinkt worden aan de toebehorende reservering, ook wordt de data doorgestuurd naar de toebehorende gast via de email om te laten weten dat de betaling gelukt is en ze verwacht worden op de datum die gezet is.

# 05 Grenzen

Zoals al eerder geschreven was, is dat de data alleen met toebehorende personen gedeeld mag worden. Zo is het optioneel voor de gebruiker om de data te delen met medewerkers van Digendo als er problemen ontstaan tijdens een betaling.

Op het moment zijn er geen showstopper, en dit blijft waarschijnlijk zo voor de rest van het project.

Wanneer er vragen zijn vanuit mijn kant of iets teruggekoppeld wil krijgen, zal ik dit overleggen met de projectleider. Wanneer dit gedaan is zullen er besproken onderdelen neergeschreven worden in een aparte bijlagen.

# 06 Plan van Aanpak

De volgende stappen zullen gedaan worden om het project volledig te creëren:

1. **Het installeren van de Mollie plug-in via packagist**

Mollie heeft over de jaren heen veel aanpassingen gedaan aan hun API, dit lijdt nu tot een van de verbeterde API’s. Door het gebruik van één centrale API zorgt dit ervoor dat er snel een betaling aangemaakt kan worden, omdat de betalingsmethode in Mollie zelf gekozen kan worden. Hierdoor worden betalingen asynchroon uitgevoerd, voor een snelle betaling en doorsturen van data.

1. **Het aanmaken van nieuwe controllers en het uitbreiden van de reservatie blade.php**

De controllers zullen gebruikt worden om de PHP aan te roepen en uit te voeren terwijl de pagina nog moet laden, via de controller zal dan de data met een return view meegestuurd worden om de data te kunnen gebruiken, die data zal gebruikt worden om te kijken bij welk bedrijf er gereserveerd gaat worden en welke modules er aanwezig zijn.

De reserverings blade is de pagina die de gebruiker te zien gaat krijgen, op die pagina staat de GUI waar de gebruiker makkelijk een reservatie kan aanmaken. De pagina kan PHP variabele aanroepen met de blade manier **“*{{ $variabele }}”*** dit scheelt meer code voor en is makkelijker in gebruik.

1. **Het creëren van een betaalfunctie en de betaling afschrijven, data opslaan in de database**

De betaalfunctie is het onderdeel wat het langste zal duren in ontwikkelingstijd, er gaat veel tijd zitten in het opzetten van de betaling. De betaling wordt dan doorgevoerd naar de website van mollie waar de betaling uitgeoefend wordt. Wanneer de betaling is voltooid wordt er geld afgeschreven, de gebruiker zal dan terug naar de website gestuurd worden en krijgt dan de bedankt pagina te zien.

Natuurlijk moet er dan wel data in de database staan dat de betaling gelukt is, dit zal natuurlijk voor de betaling gedaan is gebeuren en in de database gezet worden met de status “forPayment”. Wanneer de betaling succesvol is uitgevoerd zal deze status veranderen naar “paid”. Het reserverings-id zal ook in de payment tabel geplaatst worden om een connectie te kunnen maken met de reservering die bij de betaling hoort.

1. **Een overzicht krijgen van betalingen van alle betalingen en klanten**

Een overzicht krijgen met de betalingen gemaakt bij een bedrijf en ook zien welke reservering er betaald heeft. Daarom is het ook belangrijk om de dashboard blade aan te passen om alleen betalingen te kunnen zien waarbij een betaling gelukt is of handmatig/import toegevoegd is. Hiervoor zal ook een aparte controller gemaakt worden waarbij er query’s uitgevoerd worden, die te maken hebben met dat bedrijf.

1. **Een mail sturen naar de klanten en laten merken dat de betaling succesvol is uitgevoerd.**

Een mail sturen naar de klant is een belangrijk deel, uiteindelijk moet deze mail ook informatie gaan vasthouden om te laten weten dat de betaling succesvol is uitgevoerd, dit kan door een simpele zin of door het te laten zien van alle bekende informatie van de betaling,

Aan het eind van de betaling is het ook handig om de klant de laten weten dat hun betaling is gelukt, daarom zal er een bedankt pagina te zien zijn als de betaling gelukt is.

## Haalbaarheidstappen

|  |  |
| --- | --- |
| Predecessor | Step |
| - | Het installeren van de Mollie plug-in via packagist |
| 1 | Het aanmaken van nieuwe controllers en het uitbreiden van de reservatie blade.php |
| 1,2 | Het creëren van een betaalfunctie en de betaling afschrijven, data opslaan in de database |
| 1,2,3 | Een overzicht krijgen van betalingen van alle betalingen en klanten |
| 1,2,3,4 | Een mail sturen naar de klanten en laten merken dat de betaling succesvol is uitgevoerd. |

# 

# 07 Eisen en planning (WorkBreakDown)

# 08 Over dit document

## Afkortingen

|  |  |
| --- | --- |
| Afkorting | Omschrijving |
| API | Application programming interface |
| DB | database |

## Referenties

<https://laravel.com/docs/5.5>

<https://nl.wikipedia.org/wiki/Application_programming_interface>

<https://docs.mollie.com/payments/overview>

<https://packagist.org/packages/mollie/mollie-api-php>

## Definities

|  |  |
| --- | --- |
| Definitie | Omschrijving |
| Showstopper | Technisch onderdeel van een ontwerp dat er voor kan zorgen dat de totale haalbaarheid van het project in gevaar komt. Het is belangrijk dat de eerst de showstoppers geïdentificeerd worden en getest worden.  Voorbeeld: als een apparaat bestuurd moet worden vanuit een PC maar de communicatie werkt niet tussen PC en apparaat, zal het dus nooit vanuit een PC bestuurd kunnen worden. Oplossing: eerst dit testen, dan pas naar de volgende stap. |

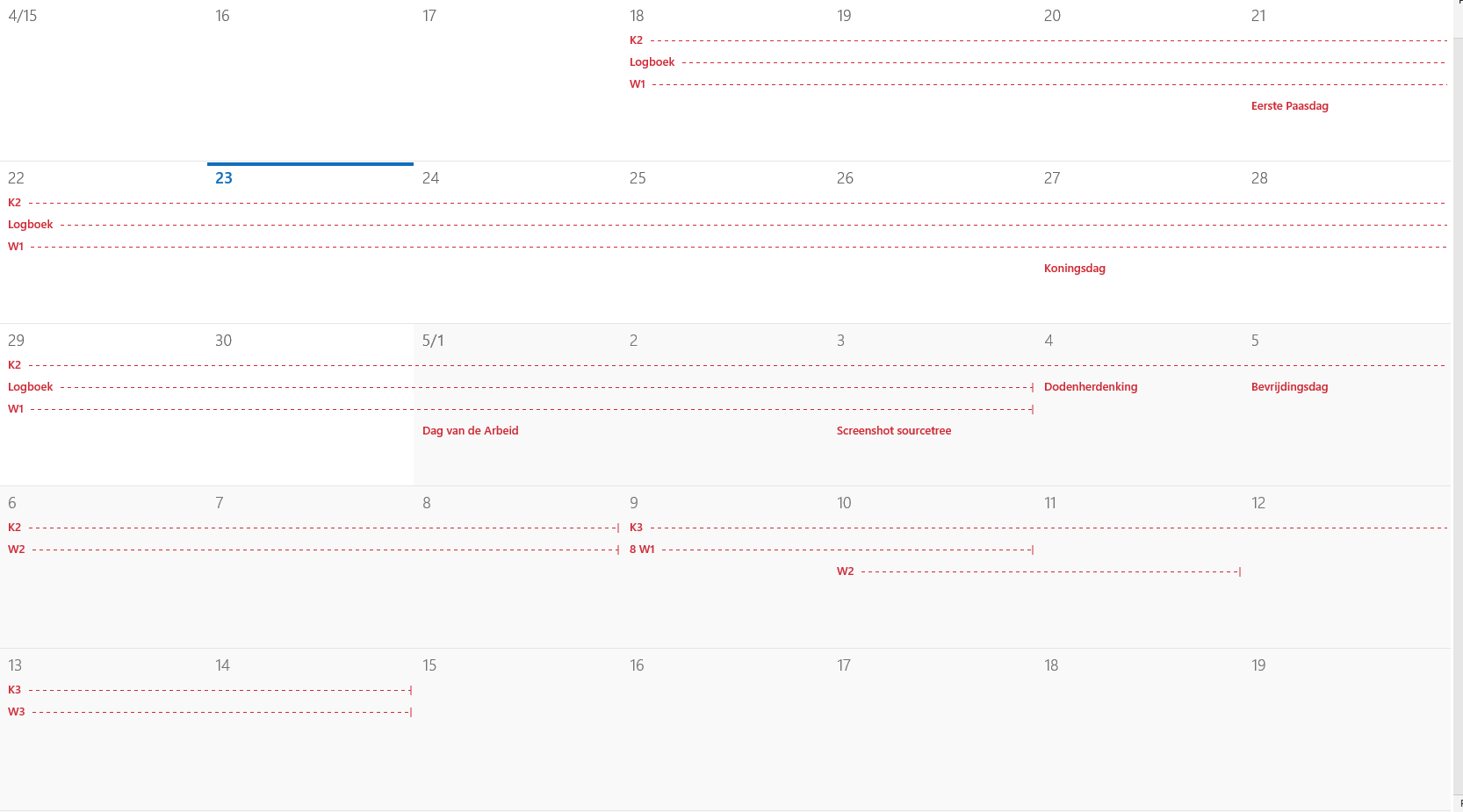
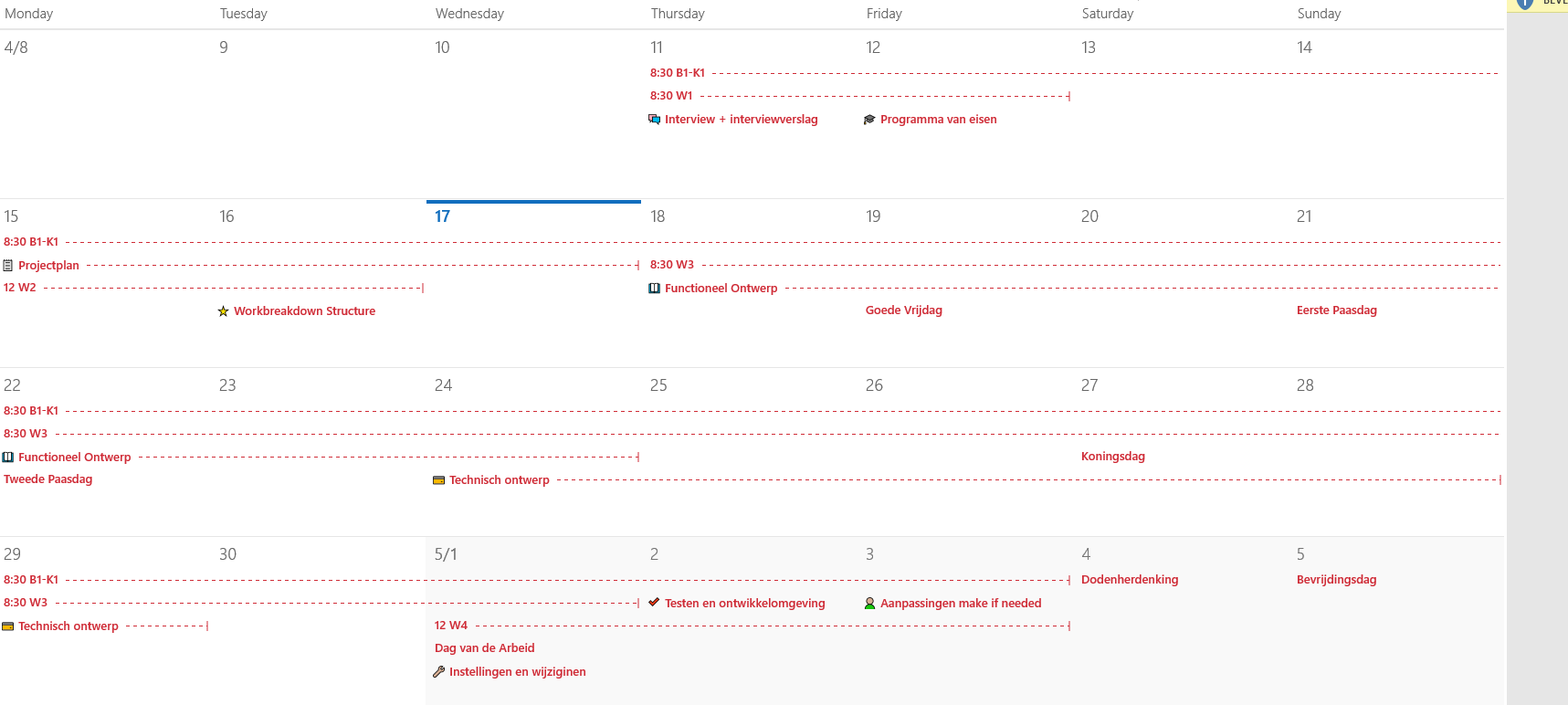
## 

## Gebruikte materialen

|  |  |
| --- | --- |
| Onderdeel | Omschrijving |
| Html | HyperText Markup Language (afgekort HTML) is een op SGML gebaseerde opmaaktaal voor de specificatie van documenten, voornamelijk bedoeld voor het wereldwijde web. Het is de standaard opmaaktaal voor webpagina's. |
| PHP | PHP (PHP: Hypertext Preprocessor) is een scripttaal, die bedoeld is om op webservers dynamische webpagina's te creëren |
| Database | Een plek waar je data op slaat voornamelijk met **SQL**, hier wordt belangrijke data opgeslagen totdat dit weer nodig is. |
| SQL | SQL (Structured Query Language) is een ANSI/ISO-standaardtaal voor een relationeel databasemanagementsysteem (DBMS). Het is een gestandaardiseerde taal die gebruikt kan worden voor taken zoals het bevragen en het aanpassen van gegevens in een relationele database. |
| Laravel | Laravel is een vrij, opensource-PHP-webframework. Kenmerken van Laravel zijn onder meer een modulair pakketsysteem met een dedicated dependency manager, diverse manieren om relationele databases te manipuleren, hulpprogramma's die helpen bij het uitvoeren en onderhouden van applicaties. |

# 09 planning

Planning B1-k1



# 09 Bijlagen

2.B: WBS

2.C: Terugkoppelingen