

2014•2015
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN
master in de industriële wetenschappen: elektronica-ICT

Masterproef

Ontwikkelen van een leveranciersportaal

Promotor :
Prof. dr. ir. Luc CLAESEN

Promotor :
Dhr. KOEN BECKERS

Ruben Claes
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

2014•2015
Faculteit Industriële
ingenieurswetenschappen
master in de industriële wetenschappen: elektronica-ICT

Masterproef

Ontwikkelen van een leveranciersportaal

Promotor :
Prof. dr. ir. Luc CLAESEN

Promotor :
Dhr. KOEN BECKERS

Ruben Claes
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Woord vooraf

Gedurende 4 jaar heb ik mijn opleiding tot Industrieel Ingenieur Elektronica-ICT gevolgd aan U-Hasselt. Ik heb deze opleiding afgesloten met een masterproject bij het bedrijf Punch Powertrain N.V. in Sint-Truiden.

Buiten de technische en praktische kennis die ik tijdens deze periode heb opgedaan was het ook vooral een zeer leerrijke en interessante ervaring.

Graag zou ik de volgende personen willen bedanken voor hun hulp en goede begeleiding om dit eindwerk tot een goed einde te brengen:

- Mijn bedrijfspromotoren de dhr. Koen Beckers & mevr. Martine Dewit, voor het opvolgen van mijn project en het beantwoorden van al mijn vragen;
- Mijn interne promotor Prof. dr. ir. Luc Claesen, dankzij zijn verdiepende kennis en inzicht kreeg ik voldoende feedback om deze masterproef te realiseren
- Alle werknemers van Punch Powertrain, voor de aangename sfeer in het bedrijf;
- Het bedrijf Punch Powertrain zelf, voor de leerrijke masterthesis die i in het bedrijf mocht doen;
- Mijn ouders en vrienden voor hun onvoorwaardelijke steun.

Ruben Claes

Heusden-Zolder, juni 2015

Inhoudstabel

Woord vooraf	1
Lijst van figuren.....	11
Inleiding.....	13
Situering.....	13
Probleemstelling.....	14
Doelstellingen	15
Methode en materiaal	17
Abstract	9
1. Literatuurstudie.....	21
1.1 Web portaal	21
1.1.1 Geschiedenis	21
1.1.2 Definitie	21
1.1.3 Type portalen	22
1.1.4 Voornaamste functies van een portaal	23
1.1.5 Wat is een portaal niet?	24
1.1.6 Toekomstige trends	24
1.2 CMF versus WAF	25
1.2.1 WAF.....	25
1.2.2 CMF	26
1.2.3 Misverstanden.....	26
1.2.4 Wat kiezen?.....	26
1.3 PHP versus de rest	27
1.3.1 Wat is PHP?	27
1.3.2 Statische en dynamische webpagina's?.....	27
1.3.3 Clientside scriptingtaal en serverside scriptingtaal?	27
1.3.4 Hoe PHP werkt	28
1.3.5 Waarom PHP?	29
1.3.6 Syntax PHP.....	29
1.4 MySQL	30
1.5 Laravel en AngularJS.....	30
2. Database.....	31
2.1 Specificaties	31
2.2 Database design	31
2.2.1 Tabellen	32

2.3	API	34
2.3.1	Aanpak	34
2.3.2	PDO	35
2.3.3	REST	38
3.	Framework	39
3.1	Database abstractie laag	39
3.2	Directory structuur.....	39
3.3	Configuratie	39
3.4	Composer.....	39
3.5	SEO	39
4.	Veiligheidsprincipes.....	41
4.1	Hash gebruikerswachtwoorden met Bcrypt.....	41
4.2	Vermijden van Cross-Site Scripting (XSS).....	43
4.3	Bestanden	43
4.4	Sessie fixatie	44
4.5	Sessie Hijacking	45
4.6	Cross-site request vervalsing	46
4.7	Groepen en Permissies	47
4.8	Htdocs.....	47
4.9	SQL-injectie.....	47
5.	Features	48
5.1	Web tracking system	48
5.2	Control management system.....	48
5.3	Bootstrap	48
5.4	PIWIK.....	Fout! Bladwijzer niet gedefinieerd.
6.	Documenten & Rapporten leveranciersportaal.....	49
6.1	Documenten van Punch Powertrain	49
6.1.1	Logistics	49
6.1.2	Quality	49
6.1.3	Purchasing	50
6.1.4	Others	50
6.2	Documenten van Leverancier	51
6.2.1	Leverancier Communicatie Matrix	51
6.2.2	Certificaten	52

6.3	Rapporten	52
6.3.1	Leveranciersbeoordeling	53
6.3.2	PPM	53
6.3.3	CLIP	53
6.3.4	Openstaande orders	53
6.3.5	Back-log	54
6.4	Specifieke Producten	54
7.	Besluit	56
	Literatuurlijst	57
	Bijlagen	59

Abstract

Punch Powertrain is een zelfstandige ontwerper, fabrikant en leverancier van CVT's (Continuously Variable Transmissions) & hybride aandrijflijnen in auto's. Deze innovatieve producten voorzien voertuigen van excellente rijdynamiek, verbeterde besturing, laag brandstofverbruik en verbeteren het rijplezier. De verschillende vestigingen zijn gelegen in Sint-Truiden en Nanjing (China).

Het doel van deze masterproef is het ontwikkelen van een leveranciersportaal waarbij de communicatie tussen de leverancier en Punch Powertrain efficiënter kan verlopen. Zo kan de leverancier specifieke bestanden zoals bijvoorbeeld de leveranciersbeoordeling, verpakkingsspecificaties en eventuele verkoopcontracten online terugvinden.

Bijkomend is er een link voorzien om leveranciers meer te integreren in software oplossingen en interne processen van Punch Powertrain. Een andere belangrijke taak van het leveranciersportaal is het web opvolgingssysteem. Om het web portaal volledig intern te beheren is er ook een CMS-systeem voorzien.

De webapplicatie is geschreven in de serverscripting-taal PHP en maakt gebruik van een MySQL-gegevensbase. De software is modulair opgebouwd en gebaseerd op het PHP-framework Laravel zodat er makkelijk andere componenten of toekomstige uitbreidingen kunnen worden toegevoegd.

Dankzij deze software verloopt de communicatie tussen leverancier en Punch Powertrain efficiënter en kunnen leveranciers op een snelle manier specifieke bestanden raadplegen. Ze kunnen ook meteen bepaalde gegevens uitlezen en via grafieken op het scherm hiervan een rapport afhalen. Door het web opvolgingssysteem kan men analytische informatie en waardevolle inzichten op een overzichtelijke manier raadplegen.

Lijst van figuren

Figuur 1 XAMPP Control Panel	18
Figuur 2 Virtuele Host	18
Figuur 3 Netbeans IDE	19
Figuur 4 Hierarchy of portaal types	23
Figuur 5 Hype Cycle	25
Figuur 6 Client-Servermodel php-site	28
Figuur 7 Hello World!	29
Figuur 8 JSON permissies	32
Figuur 9 Icoon kruimelpad	33
Figuur 10 Oude mysql_ connectie	35
Figuur 11 Nieuwe PDO connectie	36
Figuur 12 Prepared statement	37
Figuur 13 Aanmeldingsscherm	41
Figuur 14 Encryptie wachtwoord	42
Figuur 15 Bcrypt Encryptie wachtwoord.....	43
Figuur 16 Encryptie van oudere wachtwoorden	43
Figuur 17 Data zuiveren.....	43
Figuur 18 Mappenstructuur.....	44
Figuur 19 heeft de gebruiker rechten	44
Figuur 20 Session Fixation.....	45
Figuur 21 SESSION Hijacking	46
Figuur 22 Voorkomen van CSRF vervalsing	47
Figuur 23 Documenten Punch Powertrain.....	49
Figuur 24 Documenten Leverancier	51
Figuur 25 Leverancier Communicatie Matrix	52
Figuur 26 Rapporten Leverancier	52
Figuur 27 CLIP Rapport	53
Figuur 28 Open Orders Rapport	54
Figuur 29 Producten Leverancier	54
Figuur 30 Bestanden voor elk product.....	55

Inleiding

Situering

Deze thesis handelt over een project in het bedrijf Punch Powertrain N.V. in Sint-Truiden. Punch Powertrain is een zelfstandige ontwerper, fabrikant en leverancier van CVT's (Continuously Variable Transmissions) & hybride aandrijflijnen. Deze innovatieve producten voorzien voertuigen van excellente rijdynamiek, verbeterde besturing, zéér laag brandstofverbruik en verbeterde rijplezier. Punch Powertrain startte de productie van CVT's in de beginjaren 90. Sindsdien is de CVT uitermate veranderd: verbeterd brandstofverbruik, betrouwbaarheid, efficiëntie en rijcomfort. Om de uiteenlopende eisen van de bestuurder te dekken, hebben ze een breed gamma aan transmissies.

Met het hoofdbedrijf, dat gelegen is in het hart van Europa nl. Sint-Truiden, blijven ze in nauw contact met de rest van de internationale automobielsector. Met de focus om tot een wereldwijde speler uit te groeien, bouwden ze een 2^{de} productiebedrijf in Nanjing (China).

Punch Powertrain heeft een duidelijk doel: "Het uitgroeien tot één van de top 3 spelers van de wereld in de CVT markt". Deze strategie zorgt ervoor dat ze steeds complexere en hoog technologische componenten ontwikkelen.

In het onderzoek- en ontwikkelingscenter in België hebben ze speciaal gebouwde duurzaamheidstestbanken. Hiermee zijn ze in staat levenschte wegsituaties na te bootsen. Zo simuleren ze de volledige levenscyclus van een voertuig onder diverse rijomstandigheden. Het R&D team is steevast bezig met vernieuwing en ontwikkeling van de nieuwe generatie van CVT's en hybride aandrijflijnen. De activiteiten omvatten concept, navigatie, ontwerp en testen waarbij de focus duidelijk ligt om een zo hoog mogelijke performance te realiseren en zo laag mogelijk brandstofverbruik.

Punch Powertrain past een zeer strenge keuring en kwaliteitscontrole toe op al hun producten en werkzaamheden. Ze zijn volledig ISO/TS 16949 gecertificeerd.

CVT

Het concept achter de CVT (Continuously Variable Transmissions) is dat het anticipeert op de behoefte van de bestuurder en zo zijn bijna oneindige verhoudingen perfect aanpast.

Toekomst

Punch Powertrains visie op de toekomst is hernieuwbare mobiliteit. Dit houdt in dat ze op zoek zijn naar milieuvriendelijke alternatieven voor hun producten. Ze willen daarom een hybride voertuig creëren dat aan de moderne technologie en de economische en ecologische verwachtingen voldoet.

Probleemstelling

Vanuit Punch Powertrain kwam de vraag om een leveranciersportaal te ontwikkelen waarbij de communicatie tussen de leverancier en Punch Powertrain efficiënter kan verlopen. Zo kan de leverancier specifieke bestanden zoals bijvoorbeeld de leveranciersbeoordeling, verpakkingsspecificaties en eventuele verkoopcontracten online terugvinden. Deze gegevens zijn bestanden die uniek zijn voor elke leverancier. Ze kunnen ook meteen bepaalde gegevens uitlezen en via grafieken op het scherm hiervan een rapport afhalen.

Bijkomend is er een link voorzien in het web portaal om leveranciers meer te integreren in software oplossingen en interne processen van Punch Powertrain.

Een andere belangrijke taak van het leveranciersportaal is het web opvolgingssysteem. Dankzij een web opvolgingssysteem kan de administrator van het web portaal analytische informatie en waardevolle inzichten op een overzichtelijke manier opvragen. Deze informatie kan gaan over het inloggen van een gebruiker, het afhalen van bestanden, de tijd dat een leverancier op een webpagina verblijft enzovoort.

Om het web portaal volledig intern te beheren is er ook een CMS-systeem voorzien. Dit CMS-systeem zorgt ervoor dat de website volledig intern kan worden aangepast. Zo kunnen pagina's en gegevens toegevoegd worden aan het portaal indien nodig.

Doelstellingen

Om dit project te volbrengen waren er een aantal doelstellingen opgesteld die moesten worden behaald. Er zijn enkele hoofddoelstellingen die zeker moesten worden volbracht om tot een volwaardige masterproef te komen.

Algemeen gedeelte toegankelijk voor iedereen

Het leveranciersportaal moest voorzien zijn van een meer algemeen deel, dat voor iedereen toegankelijk is, en een deel dat beveiligd is. Het algemeen deel is toegankelijk is voor iedereen die het leveranciersportaal bezoekt.

In dit algemeen deel vinden bezoekers algemene informatie zoals: informatie over het leveranciersportaal, onderdelen catalogus, contactgegevens en een referentie naar de website van Punch Powertrain.

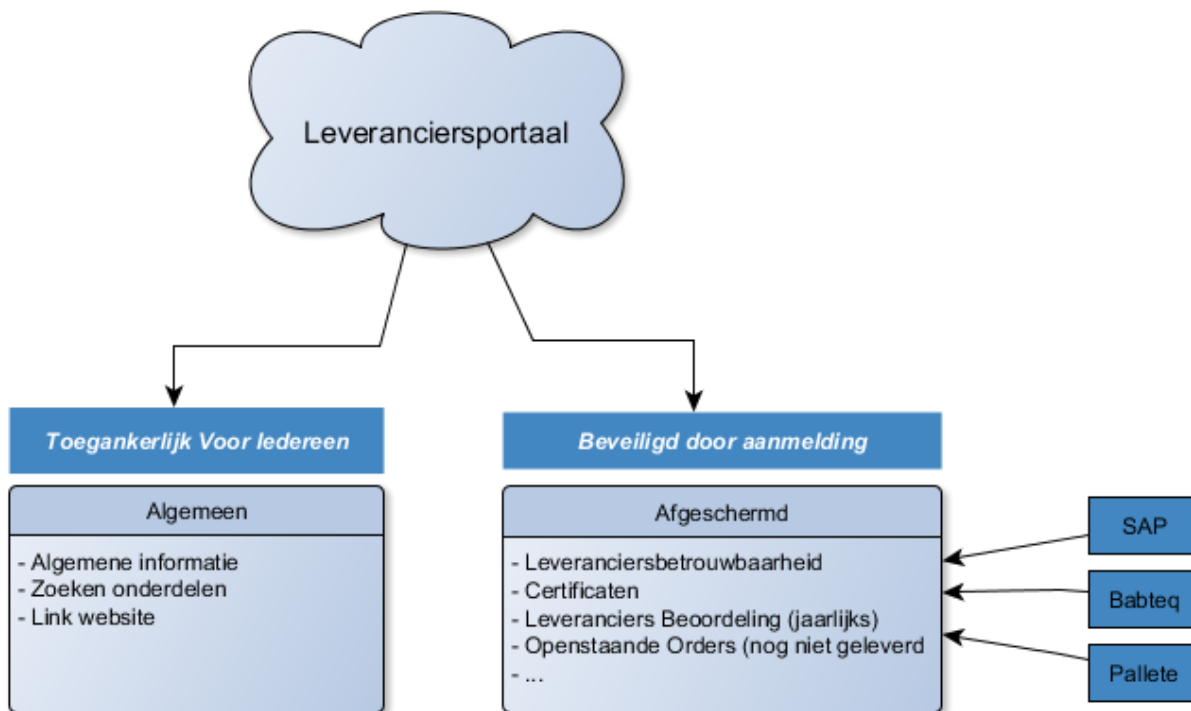
Beveiligd login-gedeelte toegankelijk voor leveranciers

In het beveiligde deel kunnen bestanden en informatie enkel én alleen door geverifieerde leveranciers worden bekeken. Na een succesvolle aanmelding kan een geverifieerde leverancier een aantal bestanden bekijken of afhalen. Deze bestanden zijn unieke gegevens die voor elke leverancier anders zijn, zoals bijvoorbeeld de leveranciersbeoordeling. Deze verschillende bestanden worden in een ander hoofdstuk dieper besproken.

Er zijn ook een aantal documenten van Punch Powertrain die voor elke leverancier dezelfde zijn. Deze algemene documenten zijn ook verzameld en beschikbaar gemaakt in het beveiligde deel.

De verschillende bestanden en gegevens worden geëxporteerd vanuit verschillende interne software toepassingen die men gebruikt binnen Punch Powertrain. Zo komen de leveranciersbetrouwbaarheidsgegevens uit een SAP-module terwijl de leveranciersbeoordeling uit het interne softwareplatform Babtec komt en de orders en openstaande orders uit Pallete.

Onderstaande figuur geeft een schematische voorstelling van het beveiligd en algemeen deel duidelijk weer.



FIGUUR 1 SCHEMATISCH OVERZICHT PORTAAL

Doordat alle informatie verzameld is op één plaats is het zeer gemakkelijk en handig voor leveranciers deze informatie meteen op te vragen en eventueel te downloaden.

Gegevens uitlezen uit software toepassingen

Eerst en vooral is het belangrijk om alle gegevens te verzamelen die uit de verschillende software toepassingen van Punch Powertrain komen. Hierbij moet er een manier worden bedacht om gegevens te verzamelen en te filteren per leverancier zodat ze daarna getoond kunnen worden op de website.

Een CMS-systeem ontwikkelen om de website te onderhouden

Een andere belangrijke taak is ervoor te zorgen dat de website moet kunnen onderhouden worden door werknemers van Punch Powertrain. Er moeten mogelijkheden worden ingebouwd zodat een werknemer kan inloggen en zo de structuur van de website gemakkelijk kan aanpassen. Niet enkel de structuur van de website kan worden aangepast maar ook de daadwerkelijke inhoud. Zo kunnen ze bijvoorbeeld nieuwe leveranciers toegang geven tot de website.

Mobiel schalen van de website

Omdat Punch Powertrain een internationaal bedrijf is en smartphones tegenwoordig een belangrijke rol spelen in het ontvangen en verzenden van webverkeer, moet ervoor gezorgd worden

dat de website gemakkelijk te raadplegen is vanuit een mobiele telefoon of tablet. Zo kan de leverancier gemakkelijk zijn gegevens raadplegen op een zeer overzichtelijke manier.

Methode en materiaal

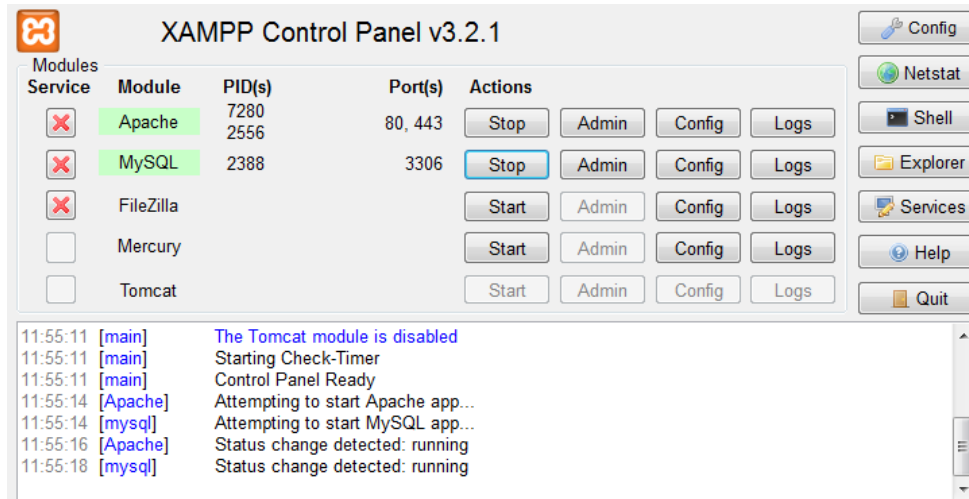
Om alle hoofddoelstellingen te realiseren is het noodzakelijk om de website eerst lokaal te draaien. Hiervoor zijn er drie softwarepakketten nodig. Deze worden in dit onderdeel kort beschreven.

XAMPP

In theorie kan alle code die gemaakt wordt direct op de goede plaats gezet worden, namelijk op de website <http://supplierportal.punchpowertrain.be>. In praktijk is dit niet zo een goede oplossing. Een betere oplossing is ervoor te zorgen dat het programmeerwerk dat verricht wordt de website <http://supplierportal.punchpowertrain.be> niet schaadt. Daarom is er gebruik gemaakt van het softwarepakket XAMPP.

XAMPP is een opensourceproject van ApacheFriends en bestaat onder andere uit een Apache-webserver, PHP, MySQL en Perl. XAMPP is zeer gemakkelijk te installeren en bevat een uitgebreide documentatie. Dankzij dit softwarepakket vormt het de ideale testomgeving voor deze masterproef.

Het zorgt er namelijk voor dat we een website kunnen laten draaien op een lokale webserver en één host, de localhost. Dit wil zeggen dat we dankzij XAMPP een eigen webserver kunnen opstarten op onze eigen machine (laptop). Hierdoor kunnen we zelf een website hosten zonder dat er een verbinding met het internet nodig is. Dit is zeer handig omdat we zo de geschreven PHP-code en MySQL query's gemakkelijk kunnen testen en uitvoeren voordat we ze later op de echte website <http://supplierportal.punchpowertrain.be> plaatsen. Hiermee kunnen we ook voorkomen dat als er fouten optreden in de code de website onbruikbaar wordt en dat gebruikers het platform dan niet meer kunnen gebruiken. Figuur 2 XAMPP Control Panel toont XAMPP waarbij er een lokale Apache-server en een MySQL database gestart is.



FIGUUR 2 XAMPP CONTROL PANEL

Virtuele host

Er is één probleem in verband met het draaien van een server op een eigen computer. Standaard, heb je alleen een single “domein” bv.: `http://localhost`. Dus als je werkt aan meer dan één website, zet je die onder “localhost” in verschillende mappen. Zo werkt het niet in de echte wereld en kan problemen veroorzaken bij het gebruik van server-side includes en root links.

Om dit op te lossen werd er gebruik gemaakt van een virtuele host. Virtuele Hosts geven de mogelijkheid om meer dan één web site en domein te maken op een computer. Een virtuele host laat niet alleen meerdere websites draaien op een computer, maar het laat ook bestanden opslaan voor sites overal op de computer en niet alleen in de standaard `C:\XAMPP\htdocs` map.

```

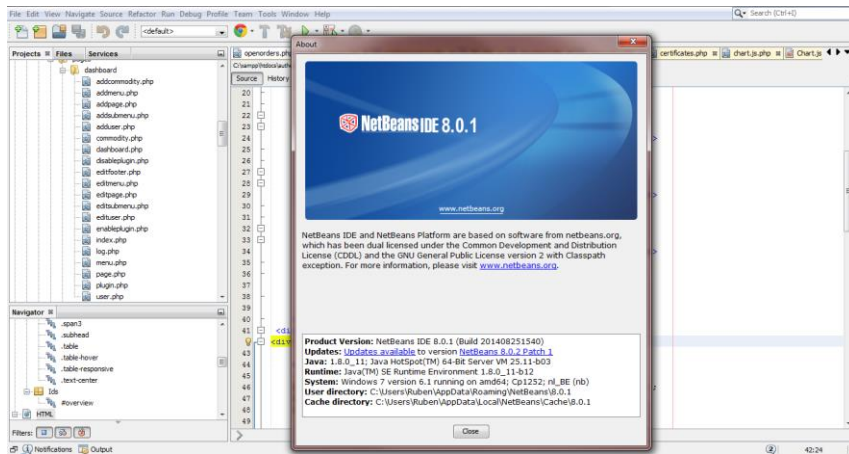
127.0.0.1      localhost
127.0.0.1      www.supplierportal.punchpowertrain.com
127.0.0.1      supplierportal.punchpowertrain.com

```

FIGUUR 3 VIRTUELE HOST

Netbeans IDE

Voor het programmeren van de PHP-code is er gebruikt gemaakt van de gratis software: Netbeans IDE. Deze software doet suggesties voor code die u kunt gebruiken in HTML en voor functies in PHP. Het doet syntax benadrukking en nog veel meer. Na de installatie van Netbeans is de professionele ontwikkelomgeving, de Integrated Development Environment (IDE) volledig gebruiksklaar.



FIGUUR 4 NETBEANS IDE

Git

Git is een vrij gedistribueerd versiebeheersysteem. Dankzij dit systeem wordt er voldoende informatie bewaard om oudere versies te kunnen terugvinden. Ook kan er voor elke wijziging in de code een omschrijving worden toegevoegd. Wijzigingen die later dan problemen blijken te veroorzaken kunnen er gemakkelijk mee ongedaan worden gemaakt.

phpMyAdmin

Een van de aantrekkelijkste oplossingen voor het ontwikkelen en beheren van MySQL-gegevensbanken is phpMyAdmin. PhpMyAdmin bestaat uit een verzameling PHP-applicaties waarmee je met webpagina's via een webserver en een webbrowser query's kunt uitvoeren voor het maken en bewerken van databases, tabellen en zoekopdrachten voor MySQL. PHPMyAdmin is bovendien een van de weinige front-ends met een Nederlandstalige gebruikersinterface.

PhpMyAdmin is opgenomen in distributies zoals XAMPP.

1. Literatuurstudie

1.1 Web portaal

De laatste jaren is de term “portaal” uitgegroeid tot een steeds populairdere term in de IT-sector en vele organisaties. Omdat deze masterthesis gaat over het ontwikkelen van een web portaal wordt in dit deel besproken wat een web portaal is en waarvoor het dient.

1.1.1 Geschiedenis

Oorspronkelijk waren deze zogenaamde portalen zoekmachines op het web. Het waren websites die dienden als een startpunt voor surfers om zo toegang en informatie te verkrijgen van het wereld wijde web.

Internet navigatie sites, zoals Yahoo!, MSN en Alta Vista vertegenwoordigden de volgende fase in de evolutie van de portaal ontwikkeling. De term “internet portaal” of “web portaal” werd bedacht voor deze megawebsites, omdat een groot aantal gebruikers ze gebruikten als een toegangspunt voor het surfen op het web. Deze sites waren voorzien van een aantal “key features” nl.: nieuws, e-mail, weer, beursnieuws en een zoekmachine. De term zoekmachine volstond al gauw niet meer, hoewel zoeken en navigeren nog steeds cruciaal was voor de meeste mensen in hun online ervaring.

Het duurde niet lang totdat bedrijven van alle groottes behoefte hadden aan een soortgelijk startpunt voor hun verschillende interne documenten en applicaties. Vele documenten en applicaties migreerden daarom naar deze web gebaseerde technologieën.

1.1.2 Definitie

Een web portaal is meestal een speciaal ontworpen webpagina die informatie verzamelt uit diverse bronnen op een uniforme manier. Deze informatie wordt gerangschikt op een manier die het meest geschikt is voor de gebruiker om toegang te krijgen.

Het zijn vaak browser-gebaseerde applicaties die activiteiten, waaronder bedrijfsprocessen binnen het bedrijf, verenigen door de toegang tot gestructureerde en ongestructureerde gegevens mogelijk te maken. Ze ondersteunen ook de integratie van applicaties naar de bedrijfsprocessen en verschaffen toegang tot actuele en consistente informatie. Populaire portalen zijn Yahoo, MSN, ...

1.1.3 Type portalen

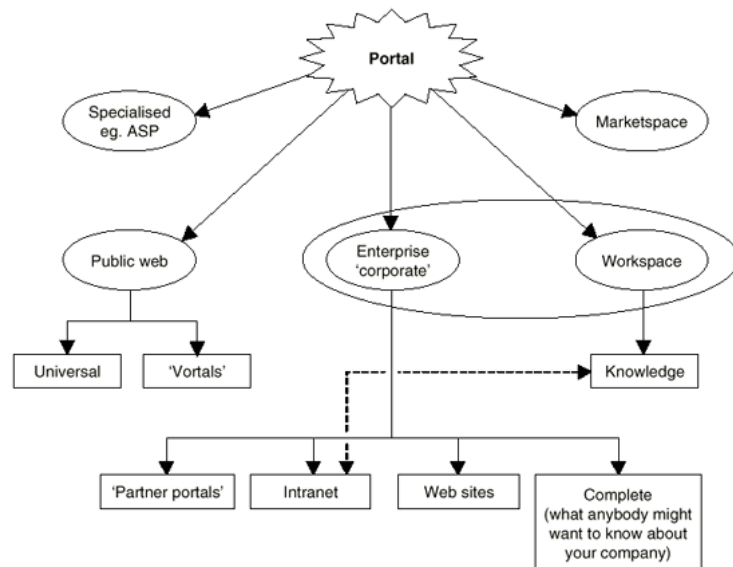
Er zijn verschillende soorten portals. De meeste kunnen worden onderscheiden op basis van de inhoud en hun gebruikers. Het is belangrijk om te weten welk soort portaal er voor deze masterthesis is ontwikkeld. Daarom worden er in dit gedeelte de voornaamste portalen kort besproken.

Verticale Portalen Dit zijn web portalen die enkel focussen op een specifieke industrie of domein. Ze bieden informatie en diensten aan op maat voor een niche publiek over een bepaald gebied of interesse. Verticale industrie portalen, ook wel bekend als “vortals”, zijn sites die een toegangspoort tot informatie leveren met betrekking tot een bepaalde sector, zoals verzekeringen of zoals in dit geval de auto-industrie.

Enterprise Information Portal Een enterprise information portal (EIP) of bedrijfsportaal, is een framework voor het integreren van informatie, mensen (leveranciers of partners) en processen over de bedrijfsgrenzen heen op soortgelijke wijze als de meer algemene web portalen. Aanvankelijk werd de term intranet portaal ook gebruikt. Het verschil met verticale portalen is dat bedrijfsportalen een beveiligd toegangspunt bieden, vaak in de vorm van een web gebaseerde user interface. Hierdoor hebben leidinggevenden, medewerkers, leveranciers en klanten toegang tot alle informatie, zakelijke toepassingen en diensten die nodig zijn om hun werk uit te voeren. Een publieke website van een bedrijf is niet automatisch een bedrijfsportaal. Het kan een portaal worden als de website gepersonaliseerde informatie en navigatie biedt. Meer geavanceerde enterprise portal-oplossingen bieden toegang via mobiele apparaten, zoals mobiele telefoons, Pda's, handheld pc's enz...

Overige Portalen In de wereld van portalen kent men nog andere belangrijke types. Deze portalen worden in Figuur 5 Hierarchy of portaal types schematisch weergegeven. Enkele van deze portalen, zoals gespecialiseerde portalen, marketpace portalen en openbare webportalen, zijn ontworpen voor een specifiek doel. Zo voorziet een gespecialiseerd portaal een ASP (application service provider) aan haar klanten voor toegang tot applicaties via een intranet of internet. Marketplace portalen bestaan om business-to-business of business-consumer e-commerce te ondersteunen. Terwijl openbare web portalen bestaan om makkelijk aantrekkelijke diensten aan te bieden aan het publiek.

A hierarchy of portal types



Source: Ovum (Enterprise Portals: New Strategies for Information Delivery/Chapter C)

FIGUUR 5 HIERANCHY OF PORTAAL TYPES

1.1.4 Voornaamste functies van een portaal

Een portaal moet de volgende functies en kenmerken ondersteunen:

Toegang Toegang geven tot informatie die nodig is (maar niet meer). Zo kan elke gebruiker de gewenste context krijgen. Bijvoorbeeld, een leverancier kan geen toegang krijgen tot bestanden van een administrator.

Indeling Een EIP categoriseert alle informatie, zodat het wordt geleverd aan de gebruiker binnen de gewenste context. Zo worden alle documenten per type gecategoriseerd en getoond.

Samenwerking Een EIP stelt individuen in staat om samen te werken, ongeacht de geografische locatie.

Personalisatie De informatie wordt voorzien door middel van een EIP en wordt gepersonaliseerd per persoon, voorkeuren en gewoontes.

Expertise en profilering Expertise en profilering zijn essentieel voor het samenwerkingsselement van een EIP. Individuen binnen een onderneming zijn geprofileerd op basis van hun ervaring en competenties. Als een iemand behoefte heeft om samen te werken met anderen, kan hij de juiste personen selecteren met de juiste kwalificaties voor het project.

Applicatie integratie Dit laat individuen toe om toegang te geven aan bestanden ongeacht de gebruikte applicatie. Zo kunnen ze verschillende externe applicaties gebruiken om hun informatie te delen met gebruikers.

Beveiliging De juiste veiligheidsmaatregelen zijn belangrijk in het ontwikkelen van een portaal. De gebruiker die zich aanmeldt krijgt alleen toegang tot informatie waarvoor hij gemachtigd is. Dit wordt verder besproken in 414 Veiligheidsprincipes.

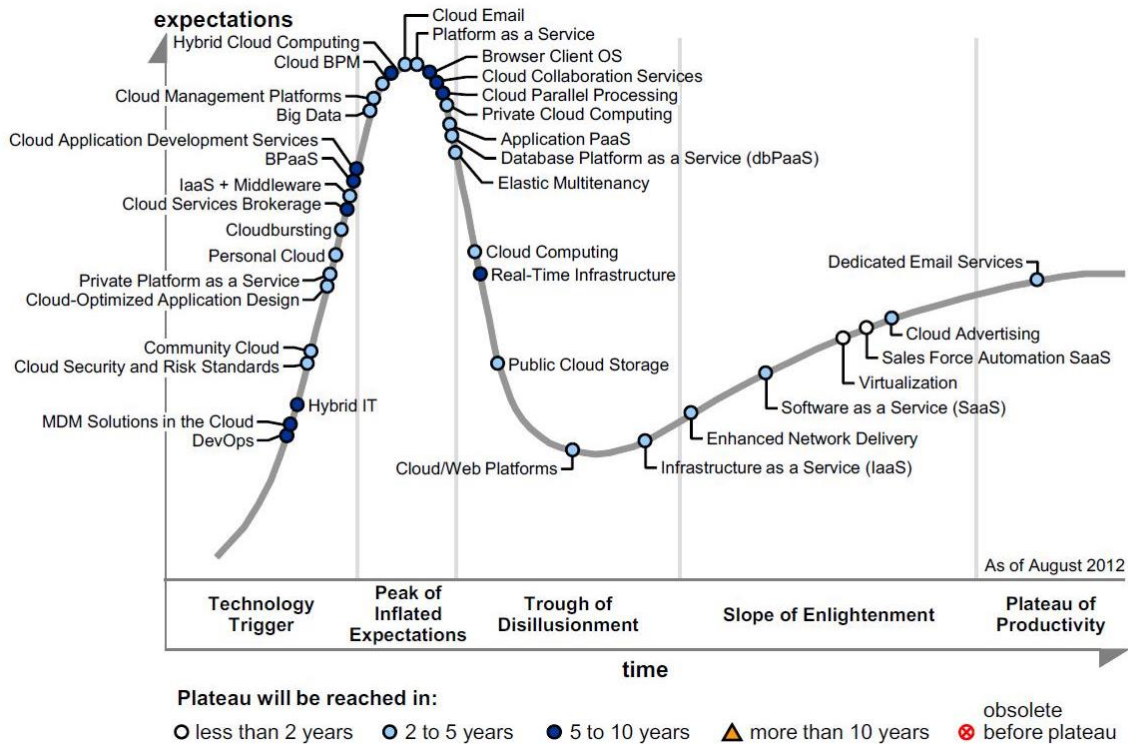
1.1.5 Wat is een portaal niet?

- Het is niet uitsluitend alleen een gewone website (welke statische data bevat)
- Het is niet uitsluitend een gepersonaliseerd intranet
- Het is niet uitsluitend een gepersonaliseerd extranet
- Het is niet enkel een persoonlijke front-end voor zakelijke toepassingen
- Het is geen persoonlijke management oplossing
- Het is niet uitsluitend een geavanceerde zoekmachine

1.1.6 Toekomstige trends

In een groot aantal analytische rapporten wordt de toekomstige ontwikkeling van portalen beschreven. Eén van de meest populaire verslagen is de “hype cycle of emerging technologies” vrijgegeven door Gartner Group.

In Figuur 6 Hype Cycle verwacht de Gartner Group dat de portaal hype zal kalmeren binnen 2 jaar. Allen dan zal een bredere markt precies weten welke specifieke functionaliteiten relevant en onmisbaar zijn.



FIGUUR 6 HYPE CYCLE

1.2 CMF versus WAF

Om de ontwikkeltijd te verkorten, bestaan er in de IT-wereld een tal van mogelijkheden. Zo bestaan er een aantal grote content management framework zoals: “Drupal, Joomla, Wordpress, ...” Wanneer een CMF-systeem gebruikt wordt kan het de ontwikkeltijd aanzienlijk versnellen, maar er zijn ook nadelen. Er zijn ook de zogenaamde PHP-frameworks: “Laravel, codeigniter, Yii, Symfony, ...” Deze frameworks bieden enorm veel mogelijkheden voor de programmeur. Voor deze masterproef tot een goed einde te brengen, werden de voor- en nadelen van PHP-frameworks en CMF-systemen met elkaar vergeleken. Uiteindelijk werd er gekozen om een eigen PHP-framework op te bouwen.

1.2.1 WAF

Wanneer de term “framework” wordt gebruikt, heeft men het over een *Web Application Framework (WAF)*. Een web applicatie framework is gefocust op coderen van code.

Programmeurs die houden van programmeren, zullen eerder kiezen voor een framework voor hun projecten.

1.2.2 CMF

Een CMF of content management framework is een *Application Programming Interface* voor het creëren van een op maat gemaakt content managementsysteem (CMS). Het is gebaseerd op modules in plaats van code. Natuurlijk kunnen ook programmeurs kiezen voor een CMS.

1.2.3 Misverstanden

Er zijn veel misverstanden over wat eigenlijk een framework en CMS kunnen doen. Veel liefhebbers van een CMS zijn van mening dat ze alles kunnen doen met een CMS terwijl veel fans van frameworks het tegengestelde beweren. De waarheid ligt in het midden, beiden kunnen bijna alle taken verrichten. Zelfs de mythe dat een CMS geen goede basisstructuur van code heeft, klopt niet. Drupal 8 is namelijk gebaseerd op het Symfony framework, dat een zeer goed object georiënteerd web application framework is.

1.2.4 Wat kiezen?

Tegenwoordig kennen PHP-frameworks steeds meer aanhang bij web ontwikkelaars. Enerzijds omdat ze een goede basis vormen voor hun project anderzijds zorgen ze voor vrijheid. Zo zijn programmeurs niet gebonden aan voorgeprogrammeerde modules. Deze modules van CMS-systemen zijn vaak complex wanneer ze aangepast moeten worden. Als men bijvoorbeeld een “functie Y” nodig heeft die niet bestaat in de module, kost dit vaak tijd en geld om deze functie te programmeren.

Het opensource karakter van PHP-frameworks zoals: “Laravel” helpt ook dat code wordt onderhouden en *bugs* worden verholpen. Dit alles zorgt ervoor dat een programmeur snel een veilige en toch wel complexe applicatie kan schrijven.

Het voordeel van CMS-systemen daarentegen is dat ze veel functionaliteit bieden “out-of-the-box”. Soms volstaat de software voor een bepaald project zonder enige aanpassing.

Voor deze masterproef werd het PHP-framework *Laravel* als basis genomen voor het leveranciersportaal.

1.3 PHP versus de rest

Als basis programmeertaal voor het leveranciersportaal werd gekozen voor de scripttaal PHP.

1.3.1 Wat is PHP?

Voor deze masterproef werd alle code in PHP geschreven. Daarom is het ook belangrijk een korte duiding te geven waar de naam PHP vandaan komt en wat PHP eigenlijk is.

PHP is in 1994 ontworpen door Rasmus Lerdorf, een senior softwareontwikkelaar bij IBM.

De naam PHP staat voor PHP Hypertext Preprocessor. Deze afkorting is een zogenaamd recursief acroniem, oftewel een afkorting die in zijn eigen betekenis voorkomt. PHP heeft zich met der tijd ontwikkeld tot een simpele taal die u in HTML-documenten kunt invoegen, tot een volwaardige, object georiënteerde programmeertaal, met een op C en Java gelijkende syntaxis. Omdat we moesten zorgen voor een objectgeoriënteerde oplossing met klassen is er gebruik gemaakt van PHP versie 5 (PHP5). Deze versie heeft een groot aantal verbeteringen, vooral wat betreft de functionaliteit voor objectoriëntatie en ook voor XML.

1.3.2 Statische en dynamische webpagina's?

PHP is een scripttaal die voornamelijk bedoeld is om op webserver dynamisch webpagina's te creëren.

Het verschil met statische websites is dat deze dynamische sites reageren op acties van individuele gebruikers.

Bij een statische webpagina werkt een webserver uitsluitend als een soort file server of bestandserver. De server doet eigenlijk weinig meer dan een opgevraagd bestand lezen van een harde schijf en dat verzenden naar de client.

1.3.3 Clientside scriptingtaal en serverside scriptingtaal?

Er zijn twee mogelijkheden om dynamische webpagina's te creëren. Enerzijds via clientside scriptingtalen zoals bv. Javascript en anderzijds via serverside scriptingtalen zoals PHP.

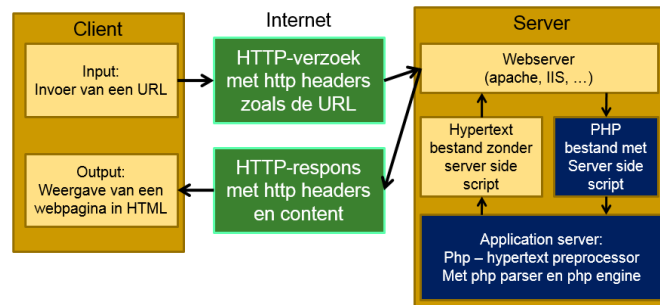
De manier waarop deze twee scriptingtalen werken zijn verschillend. Bij clientside scriptingtalen wordt programmacode ingebed in HTML-code en uitgevoerd op de client. Bij serverside scriptingtalen bevindt de programmacode op de server en daar wordt ze ook uitgevoerd.

Er is ook een verschil in gebruik. Clientside scriptingtalen worden meer gebruikt voor realtime dynamische acties bv. aanwijseffecten zoals roll-overs, controle van ingevoerde gegevens,...

Serverside scriptingtalen reageren daarentegen op acties van gebruikers en kunnen gebruikersgegevens verwerken zodat ze worden bijgehouden in een database.

Om een voorbeeld te geven in ons geval: een overlegcoördinator of zorgverlener drukt op een knop en er wordt een bepaald overleg gemaakt voor een specifieke patiënt. Een ander voorbeeld: Er kan bijvoorbeeld een patiënt met al zijn individuele gegevens worden opgeslagen in een database door te drukken op een knop van de website.

1.3.4 Hoe PHP werkt



FIGUUR 7 CLIENT-SERVERMODEL PHP-SITE

PHP wordt het meest gebruikt voor dynamische websites. In de meeste PHP-applicaties is dan ook een hoofdrol weggelegd voor het client-servermodel. Bovenstaand schema is een vereenvoudigd client-servermodel van een PHP-site.

Als er een PHP-bestand via de clientbrowser wordt opgevraagd van de server, verzend de client een http-verzoek om een webpagina. Bij het ontvangen van dit http-verzoek moet de webserver beslissen hoe er op het verzoek wordt gereageerd.

Dynamische webpagina's bevatten server-side script of serverscript. Deze zullen eerst moeten worden vertaald door een parser en daarna worden uitgevoerd door een engine.

De parser vertaalt deze scripttekst naar interne code en de engine voert deze code vervolgens uit. Tenslotte wordt er een http-respons teruggestuurd naar de client en wordt er een HTML-pagina in de browser getoond zonder serverscript.

1.3.5 Waarom PHP?

PHP is een van de makkelijkste webtalen (door het lage instapniveau) en voorziet tegelijk in grote doorgroeimogelijkheden. Zo kan men met PHP ook Object-Georiënteerd programmeren. PHP is ook vrij verkrijgbaar (open source) en draait op verschillende platformen (Windows, Linux, Mac OS X,...). Het meest populair is Linux. Voor deze bachelorproef wordt er Windows als besturingssysteem gebruikt en dus ook als ontwikkelplatform. Een ander voordeel van PHP is dat het compatibel is met bijna alle servers die vandaag gebruikt worden (Apache, Windows IIS,...). Zoals eerder vermeld, wordt er voor ons project een lokale Apache server gebruikt waarop PHP code zal worden uitgevoerd. Dit is enkel voor testdoeleinden. Nog een belangrijk voordeel van PHP is dat het veel populaire databases ondersteunt nl.: MySQL, PostgreSQL, DB2, Oracle, SQLite, Firebird/InterBase, Fontbase, Ingres en Sybase.

1.3.6 Syntax PHP

Qua syntax lijkt PHP het meest op de programmeertaal C. In tegenstelling tot C is PHP dynamisch getypeerd terwijl C statisch getypeerd is. Dit wil zeggen dat het datatype (int, string, double,...) van een variabele of functie pas gecontroleerd wordt op het moment dat deze wordt gebruikt. Net als vele andere scripttalen moeten alle variabelen worden voorafgegaan worden door een dollarteken ('\$'). Deze variabelen Dit is overgenomen uit de scripttaal Perl, waarvan PHP is afgeleid. PHP-code begint steeds met een <?php-tag en eindigt met ?>. In oudere, of slordige, PHP-script wordt er ook <? en ?> gebruikt. Een instructie in PHP eindigt steeds met een ; (puntkomma). Er mogen zoveel spaties en nieuwe regels gebruikt worden om de code leesbaar te maken. De PHP-interpreter zal deze witte ruimte toch negeren. Onderstaand voorbeeld toont een voorbeeld van PHP-code.

```
<?php
    echo 'Hello World!';
?>
```

FIGUUR 8 HELLO WORLD!

Deze code toont de zin "Hallo Wereld" op een websitepagina.

1.4 MySQL

De meeste server-side applicaties of websites maken gebruik van een database om gegevens in op te slaan. PHP wordt meestal in combinatie gebruikt met MySQL. MySQL is één van de meest populaire open source databasemanagementsystemen die tot op de dag van vandaag gebruikt wordt. MySQL is gratis om te gebruiken als de software die er mee gemaakt wordt ook open source is. Waar de naam MySQL vandaan komt, is niet duidelijk. SQL komt van Structured Query Language, en My is de naam van de dochter van een van de oprichters van het bedrijf MySQL, maar of dat ook de reden van My in de naam is, is onbekend.

MySQL is voortgekomen uit mSQL, een andere open- sourcedatabase, die door de oorspronkelijke ontwikkelaars van MySQL als te traag werd beschouwt. De functionaliteit van MySQL is gelijkwaardig aan die van de meeste 'grote' databases en nog steeds in ontwikkeling.

1.5 Laravel en AngularJS

Wanneer je begint met een nieuw web-project met behulp van PHP en MySQL, is het handig om de juiste tools voor de job te kiezen. Voor de basis van dit masterproject werden de volgende tools gebruikt: Laravel en AngularJS.

Laravel is een krachtig opensource PHP-framework waarmee code kan ontwikkeld worden die zeer kort en eenvoudig te lezen is. Dankzij Laravel kunnen geavanceerde projecten worden bekomen met weinig code. Volgens een enquête die aan ontwikkelaars werd gesteld in december 2013, die over de populariteit tussen verschillende PHP frameworks ging, wordt Laravel vermeld als het meest populaire PHP framework in 2013.

AngularJS is een JavaScript-framework dat ontwikkeld is om te voldoen aan de eisen van grote en complexe web applicaties. AngularJS werd oorspronkelijk ontwikkelt door een team van Google, maar inmiddels is er ook een grote open source-community actief op het net.

Met AngularJs ontwikkel je niet één grote applicatie waarin alles met alles verbonden is. In plaats daarvan maak je kleinere, gespecialiseerde modules. Deze zijn gemakkelijk los van elkaar te testen en te ontwikkelen

2. Database

2.1 Specificaties

Dit gedeelte gaat over de nauwkeurige specificatie van het leveranciersportaal. De lijst hieronder toont de details van de specificatie met bijhorende veronderstellingen.

Gebruikers De gebruikers van het web portaal zijn verdeeld in drie groepen: administrators, leveranciers en gast. Een administrator of beheerder heeft toegang tot alle delen van de website. Een geverifieerde leverancier kan enkel zijn unieke gegevens en grafieken zien. De gast account wordt gebruikt om een voorbeeldsite te tonen met fictieve data. Om zich aan te melden moet de gebruiker inloggen met zijn wachtwoord en unieke leverancierscode. De administrators van de website hebben geen leverancierscode en moeten zich enkel aanmelden met een wachtwoord. Het wachtwoord wordt opgeslagen in een gerelateerde database van het web portaal.

Communicatie Matrix Elke leverancier heeft een communicatie matrix die ingevuld moet worden. Dit is een verzameling van contactgegevens van de leverancier. Hierin staat bijvoorbeeld: de leveranciersnaam, email, accountmanager, verkoopdirecteur,...

Leverancierscode Alle leveranciers hebben een unieke leverancierscode bestaande uit 5 cijfers. Deze code is een extra veiligheidsmaatregel om de leverancier te onderscheiden van een hacker.

Groepen Zoals eerder vermeldt zijn er drie soorten gebruikers. Deze gebruikers worden onderverdeeld in groepen met permissies. Deze permissies zijn als het ware regels waaraan de gebruiker moet voldoen. Zo kan een leverancier niet de website aanpassen omdat hij daar geen permissie voor heeft.

Onderdelen Op het leveranciersportaal kunnen bezoekers de onderdelen of commodities bekijken waar Punch Powertrain naar opzoek is. Enkel leveranciers kunnen hierop reageren en een mail versturen voor onderdelen die ze kunnen leveren.

2.2 Database design

De ruggengraat van de applicatie is een MySQL-database. Bij het bedenken van een database-ontwerp is het belangrijk om dit correct te doen bij de eerste keer. Het is bijna onmogelijk om de database volledig opnieuw te ontwerpen als het werk zich in een ver gevorderd stadium bevindt. Het was één van de eerste stappen die werden uitgevoerd voor deze masterthesis. Met alle specificaties is er een eerste versie van het database-ontwerp gemaakt (**zie bijlage B**). Tijdens de ontwikkelingsperiode zijn er nog kleine aanpassingen gemaakt. Er zijn meer velden bijgemaakt voor extra variabelen.

2.2.1 Tabellen

Deze sectie geeft een korte toelichting van elke tabel van de database. De meeste velden hoeven niet expliciet worden beschreven. Alle ID-velden (ID, _id) zijn de primaire sleutels voor de bijhorende tabel en hebben een auto increment functie. Deze unieke identificatie zorgt voor redundantie. Wanneer bijvoorbeeld projectgegevens samen met leverancier gegevens in één record worden opgeslagen, kan de tabel vele malen het adres bevatten van dezelfde leverancier. Dit kan leiden tot een update en/of insert anomalie. Dankzij het normaliseren van de database wordt dit voorkomen.

users Deze tabel bevat de gebruikers die toegang hebben tot het leveranciersportaal. Het veld username is uniek en van het type varchar zoals ook de velden name, email en password. Het password veld heeft een varchar met lengte 256 om in de toekomst langere gehashte wachtwoorden op te slaan. Het veld group is van het type int en geeft aan of de gebruiker tot een leverancier (waarde 1), administrator (waarde 2) of gast (waarde 3) behoort.

communication_matrix Deze tabel bevat alle bijkomende informatie omtrent de leverancier. Het veld user_id en supplier_code_id bevat een waarde van het type int en verwijzen respectievelijk naar het id veld van de tabel users en de tabel supplier_codes. Daarnaast zijn er nog de velden supplier_name en supplier_email. Deze velden zijn beide van het type varchar.

supplier_codes In deze tabel kan men de leverancierscodes vinden. Het veld user_id bevat een waarde van het type int en verwijst naar het id veld van de tabel users. Het veld supplier_code bevat een unieke int waarde van 5 cijfers .

groups Deze tabel bevat alle groepen en bijhorende rechten of permissies. Het veld name bevat een waarde van het type varchar. Dit zijn de verschillende groepsnamen. Het veld permissions bevat een json string waarin de verschillende rechten per groep staan vermeld.

In Figuur 9 JSON permissies worden de permissies van een leverancier getoond.

```
{"admin": 0, "supplier":1,"moderator":0,  
"standardpage":1, "reaction":1}
```

FIGUUR 9 JSON PERMISSIES

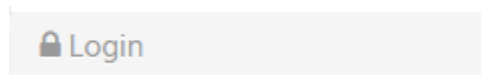
users_session In deze tabel worden alle ingelogde gebruikers getoond. Het veld hash is de PHP-sessie en bevat een gehashte waarde van het type varchar.

commodities In deze tabel worden alle onderdelen waar Punch Powertrain naar opzoek is opgeslagen. Het veld `user_id` bevat een waarde van het type `int` en verwijst naar het `id` veld van de tabel `users`. Het veld `date` is van het type `datetime`. De velden `title`, `tekst` en `path` van het type `tekst`.

reactions In deze tabel worden alle reacties van `commodities` in opgeslagen. De veldnaam `date` is van het type `datetime` en toont wanneer deze reactie werd toegevoegd. De velden `user_id` en `commodity_id` zijn waardes die verwijzen naar de `id`'s van de tabel `users` en `commodities`. Het veld `text` is van het type `text`.

Pages In deze tabel wordt alle informatie bijgehouden over de webpagina's die kunnen worden aangepast door het CMS-systeem. De velden `user_id`, `menu_id` en `submenu_id` bevatten een waarde van het type `int` en verwijzen naar het `id` veld van de tabellen `users`, `menu` en `submenu`. De velden `title`, `slug` zijn van het type `varchar`. De velden `heading`, `info` en `body` zijn van het type `text`. Het veld `visible` heeft een waarde van het type `boolean` (`true` or `false`). Door deze waarde kan men pagina's verbergen of tonen voor de gebruikers. Het veld `access` is een waarde van het type `int`. De verschillende waardes zijn een extra veiligheidsmaatregel voor de toegankelijkheid van de pagina. De waarde 0 zorgt ervoor dat de pagina voor iedereen zichtbaar is. De waarde 2 en 4 zijn respectievelijk voor behouden voor pagina's die toegankelijk zijn voor de leverancier en administrator.

menu Deze tabel bevat alle menu-items van de website. De velden `link`, `icon`, `title` en `page_name` bevatten een waarde van het type `varchar`. `link` is de url achter de menuknop. `icon` is het icoon van het kruimel pad dat op elke webpagina wordt weergegeven. Het veld `position` is een waarde van het type `int`. Hiermee wordt de positie van alle menuitems weergegeven. Het veld `visible` heeft een waarde van het type `boolean` (`true` or `false`). Door deze waarde kan men pagina's verbergen of tonen voor de gebruikers. Het veld `access` is een waarde van het type `int`. De verschillende waardes zijn een extra veiligheidsmaatregel voor de toegankelijkheid van de pagina. De waarde 0 zorgt ervoor dat de pagina voor iedereen zichtbaar is. De waarde 2 en 4 zijn respectievelijk voor behouden voor pagina's die toegankelijk zijn voor de leverancier en administrator.



FIGUUR 10 ICOON KRUIMELPAD

submenu Deze tabel bevat alle submenu-items van de website. De velden zijn grotendeels hetzelfde als de menu tabel. Er is één extra veld `nl.`: `parent_id`. Dit veld heeft een waarde van het type `int` en verwijst naar een `id` van de tabel `menu`.

footer In deze tabel wordt alle informatie over de footer van de webpagina opgeslagen.

notifications Deze tabel bevat alle notificaties van een gebruiker.

plugins Voor toekomstig gebruik. Hier kunnen modules of plug-ins voor het leveranciersportaal worden opgeslagen.

ISO certificaten Er zijn twee tabellen die informatie bevatten over respectievelijk de ISO bestanden en ISO TS bestanden nl.: iso_certificates en iso_ts_certificates. Het veld valid_until is van het type datetime. Dit is de datum tot wanneer het iso-certificaat geldig is. Het veld issue_by en path zijn beiden van het type text.

overige tabellen De tabellen account_manager, finance_administration, quality_administration, sales_director en sales_orders hebben ongeveer dezelfde veldnamen daarom worden deze samen besproken. De tabellen bevatten informatie van bijvoorbeeld de accountmanager, verkoopdirecteur, financemanager,... Het veld communication_id bevat een waarde van het type int en verwijst naar het id veld van de tabel communication_matrix. De velden name, title, email, phone en mobile zijn van het type varchar.

2.3 API

Zoals besproken in de doelstellingen is het noodzakelijk om het leveranciersportaal modulair op te bouwen zodat de flexibiliteit vergroot. Hiervoor is er een databaseabstractie laag voorzien en is er een API gecreëerd op niveau van dataopslag en –toegang.

Als eerste wordt de aanpak van procedureel naar object-georiënteerd programmeren toegelicht. In 2.3.2 wordt de database extensie PDO besproken. In 2.3.3 wordt er dieper ingegaan op krachtige functies, zodat de API wordt geprogrammeerd met een minimum aan functies. Omdat er toekomstgericht moet worden gedacht is een REST-laag toegevoegd aan de webapplicatie.

2.3.1 Aanpak

PHP is in de allereerste plaats een procedurele scripttaal, waaraan later mogelijkheden voor object georiënteerd programmeren werden toegevoegd. Bij een procedurele scripttaal zoals PHP is elke operatie die je uitvoert, één procedure, expressie, instructie, commando, enzovoort.

Deze procedures worden stap voor stap uitgevoerd in de volgorde waarin je ze in het tekstbestand met een PHP-script zet.

De manier die in het leveranciersportaal gebruikt wordt object georiënteerd PHP. Object georiënteerd PHP heeft een andere benadering. Bij object georiënteerd programmeren gebruik je objecten veel meer als losse bouwstenen.

OOP biedt enkele voordelen ten opzichte van procedureel programmeren. Het is namelijk zéér overzichtelijk mits je het goed implementeert. De code is ook voor veel toepassingen herbruikbaar zonder aanpassingen te moeten doen in elke applicatie. Men kan ook in veel grotere snelheden projecten ontwikkelen. Dit komt omdat men klassen die door andere zijn ontwikkeld kan hergebruiken in een eigen toepassing

2.3.2 PDO

De webapplicatie maakt gebruik van de PDO_MySQL extensie voor het openen van databaseverbindingen. De oude extensie `mysql_` is sinds PHP versie 5.5.0 verouderd en zal in de toekomst worden verwijderd.

De reden waarom deze extensie moet worden verwijderd is omdat het geen ondersteuning meer biedt voor moderne SQL-database concepten zoals prepared statements, transacties enz... Het andere probleem met `mysql_` is dat het de laatste tijd nog maar weinig aandacht krijgt van ontwikkelaars. Het wordt niet meer onderhouden. Dit veroorzaakt beveiligingsproblemen of het kan helemaal stoppen met werken met nieuwere versies van MySQL.

In plaats hiervan moet de MySQLi of PDO_MySQL extensie worden gebruikt. Onderstaand voorbeeld toont de oude manier om met een database te verbinden.

```
<?php
$link = mysql_connect('localhost', 'user', 'pass');
mysql_select_db('testdb', $link);
mysql_set_charset('UTF-8', $link);
```

FIGUUR 11 OUDE MYSQL_ CONNECTIE

In deze masterproef wordt gebruik gemaakt van de PDO extensie. PDO oftewel PHP Data Objects bestaat al sinds PHP versie 5.1 en is een zéér handige extensie voor het benaderen van verschillende databases. Het biedt ondersteuning voor 12 database servers maar in ons geval hebben we enkel diegene nodig die verbinding maakt met een MySQL database. Het ondersteunt tevens het principe van *named parameters*, waardoor een *prepared statement* gemakkelijker te lezen is dan bij *unnamed parameters* (mysqli).

Het verbinden met een database met behulp van PDO is vrij gemakkelijk. Het komt er eigenlijk op neer dat je een nieuwe instantie moet aanmaken van de PDO klasse.

Natuurlijk moeten bij het instantiëren van deze klasse wel de correcte parameters worden meegegeven. Onderstaand voorbeeld toont de nieuwe manier om met een database te verbinden.

```
$this->_pdo = new PDO(
    'mysql:host=' . Config::get('mysql/host') . ';'
    . 'dbname=' . Config::get('mysql/db'),
    Config::get('mysql/username'), Config::get('mysql/password'));
```

FIGUUR 12 NIEUWE PDO CONNECTIE

De parameters die gebruikt worden zijn vanzelfsprekend. Omdat het leveranciersportaal opgebouwd is als een framework, worden deze parameters uit een aparte *Config* klasse gehaald.

Het handmatig sluiten van een databaseverbinding is in principe niet echt nodig. Aangezien het bestaan van de verbinding direct afhangt van het bestaan van de instantie van de PDO klasse, zal de verbinding verbroken worden zodra het script wordt beëindigd. Het veelvoudig tussendoor sluiten en openen van verbindingen kan tot een zeker verlies van performantie leiden. Maar langs de andere kant staat hier tegenover dat elke openstaande verbinding onnodige resources in kan nemen.

Eén van de belangrijkste voordelen van PDO is dat je gebruik kan maken van *prepared statements*. Het grote voordeel van zo een *prepared statement* is dat een bepaalde query op een efficiënte manier vele malen achter elkaar uitgevoerd kan worden. Bovendien bieden deze *prepared statements* de programmeur de mogelijkheid tot het creëren van query's die veiliger zijn, waarvan de performance beter is en die eenvoudiger te coderen zijn. De veiligheidsvoordelen hebben betrekking tot SQL-injecties. Dit wordt in 4.9 besproken.

Bij het gebruik van deze *prepared statements* wordt er allereerst een template van de uit te voeren query naar de server gestuurd. Deze wordt door de server gecontroleerd op syntax fouten en uitgevoerd om te controleren of de query wel klopt. Daarna wordt deze opgeslagen in een speciale buffer.

Op het ogenblik dat de query daadwerkelijk uitgevoerd moet worden, stuurt de server de in de query in te vullen gegevens naar de database. Ten slotte wordt er een complete query gevormd en uitgevoerd.

Een ander pluspunt van PDO is dat het over een betere foutafhandeling beschikt. Hierbij is het mogelijk om in een *try/catch* blok de uitzondering op te vangen en te verwerken. Bij de oude *_mysql* extensie is er geen foutafhandeling mogelijk. De fout kan enkel gerapporteerd worden via de *mysql_error()* methode.

```
try {  
    // Prepare query  
    $this->_query = $this->_pdo->prepare($sql);  
    $x = 1;  
    if (count($params)) {  
        foreach ($params as $param) {  
            $this->_query->bindValue($x, $param);  
            $x++;  
        }  
    }  
  
    // Execute SQL  
    $this->_query->execute();  
    $this->_results =  
        $this->_query->fetchAll(PDO::FETCH_OBJ);  
    $this->_count = $this->_query->rowCount();  
  
    return $this;  
} catch (PDOException $e) {  
    // Write into log and display Exception
```

FIGUUR 13 PREPARED STATEMENT

Figuur 13 Prepared statement toont hoe we een insert query uitvoeren met behulp van deze *prepared statements*.

Deze functie voegt data toe aan de tabel. Als eerste wordt de *prepare()* methode uitgevoerd om de query naar de server te sturen waar deze wordt gecontroleerd.

Voor elke parameter in de query template roepen we de *bindValue()* methode aan om deze met een variabele te koppelen. Op die manier weet PHP straks welke variabele hij onder welke naam naar de server moet sturen zodat deze het begrijpt. Als er tenslotte aan de variabele een waarde wordt toegekend, kan de query worden uitgevoerd door de *execute()* methode.

Als alles goed is gegaan, krijgen we geen foutmeldingen en zal er een nieuwe record worden ingevoerd.

2.3.3 REST

3. Framework

3.1 Database abstractie laag

3.2 Directory structuur

3.3 Configuratie

3.4 Composer

3.5 SEO

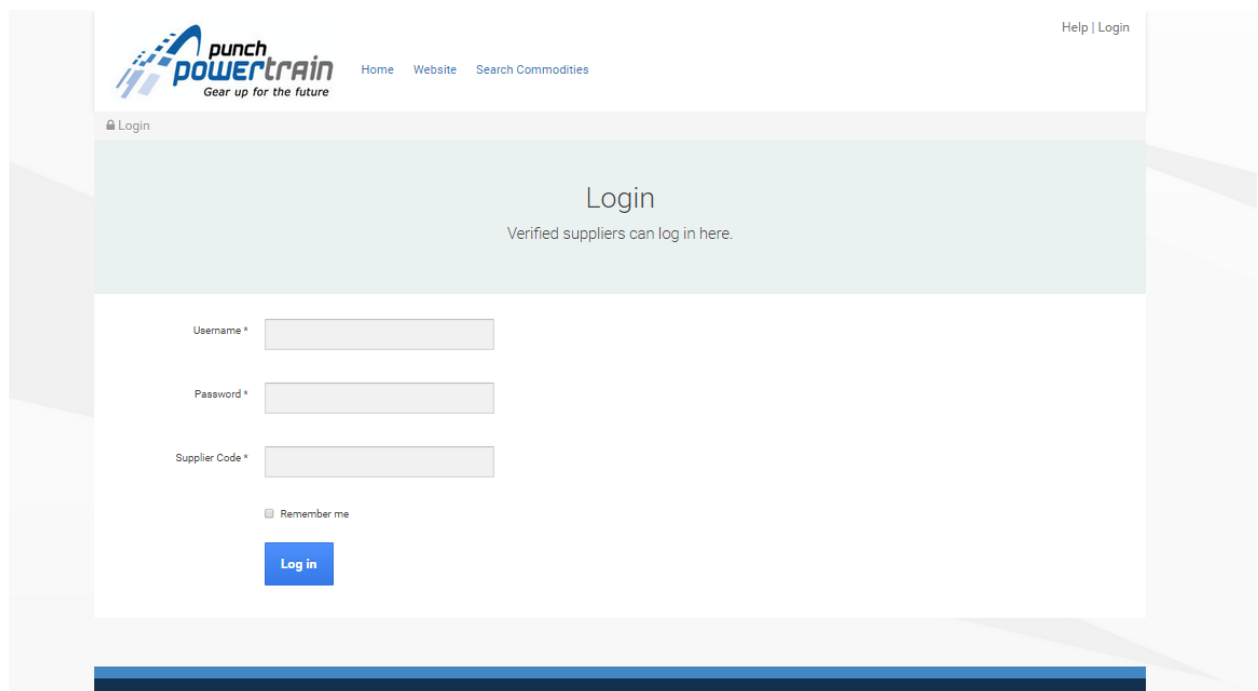
4. Veiligheidsprincipes

Sinds het ontstaan van Web 2.0 is het aantal directe aanvallen op websites sterk vergroot. Dit komt omdat er steeds meer informatie via sociale netwerken wordt gedeeld. De manier van zakendoen is ook veranderd. Steeds meer bedrijven zien het nut van het leveren van online diensten en informatie.

Hierdoor is de aandacht voor de veiligheid van web toepassingen sterk vergroot en is het zeker zo belangrijk als de beveiliging van het onderliggend computernetwerk en het besturingssysteem. Vandaar zijn er verschillende veiligheidsprincipes geïmplementeerd om het leveranciersportaal te beveiligen. Er wordt een korte beschrijving gegeven van de verschillende veiligheidsmaatregelen.

4.1 Hash gebruikerswachtwoorden met Bcrypt

Om toegang te krijgen tot het leveranciersportaal moet de leverancier of de administrator zich eerst aanmelden. Dit gebeurt via de loginpagina van de website. Op de loginpagina moet men een gebruikersnaam, wachtwoord en een unieke leverancierscode ingeven. Dit zorgt ervoor dat er een extra veiligheidsmaatregel wordt gecreëerd.



The screenshot shows the login interface for the Punch Powertrain portal. At the top left is the logo with the tagline 'Gear up for the future'. To the right are navigation links: 'Home', 'Website', and 'Search Commodities'. In the top right corner, there are links for 'Help' and 'Login'. The main heading is 'Login' with the subtext 'Verified suppliers can log in here.'. The form contains three required input fields: 'Username *', 'Password *', and 'Supplier Code *'. Below the password field is a 'Remember me' checkbox. A blue 'Log in' button is at the bottom of the form.

FIGUUR 14 AANMELDINGSSCHERM

De gebruikersnaam en wachtwoord zijn opgeslagen in een MySQL-tabel door een administrator. Het systeem zal, nadat er op de “Log in” knop wordt geklikt, een controle uitvoeren van de ingegeven waardes. De controle vergelijkt de ingeven waardes met de waardes die opgeslagen zijn in de database. Wanneer de controle is goedgekeurd zal het systeem de toegang verlenen voor de gebruiker.

Omdat wachtwoorden gevoelige informatie bevatten, worden deze gehashed opgeslagen in de database.

id	username	name	email	password	group	imagepath	joined
6	ruben	ruben claes		\$2y\$12\$0vrjgvcX2i079hfrQP6AjuFLTKHq2N5pawnwTTHli8...	2		2015-02-26 18:24:14

FIGUUR 15 ENCRYPTIE WACHTWOORD

Wanneer er een wachtwoord wordt opgeslagen in de database, zal dus eerst het wachtwoord gehashed worden en daarna zal deze hash worden opgeslagen. Dankzij hashing kan data niet teruggezet worden naar zijn oorspronkelijke vorm en identieke informatie zal altijd dezelfde hash-waardes produceren. Als hackers de database kraken, zien ze enkel zinloze tekens. Hierdoor is het vrijwel onmogelijk om het daadwerkelijke wachtwoord te achterhalen. Dit heeft ook als grote voordeel om brute force aanvallen te voorkomen.

Het wachtwoord is gehashed door gebruik te maken van BCrypt. Bcrypt is een “one-way” hash-algoritme waarbij een “salt” wordt gebruikt. Op dit moment is dit één van de veiligste methodes om wachtwoorden op te slaan in databases. In tegenstelling tot andere hash-algoritmes (SHA1, MD5,...) is bcrypt ontworpen om zeer traag te zijn.

Deze traagheid zorgt ervoor dat een aanvaller extreem veel resources en hardware moet implementeren om het wachtwoord te kunnen kraken. Doordat elk wachtwoord een unieke salt-string krijgt, kan je er zeker van zijn dat een aanvaller vrijwel niet in staat is het wachtwoord te kraken. Het bcrypt algoritme is ook toekomstbestendig, want je kan haar werkfactor gewoon verhogen naarmate de computers sneller worden. De werkfactor is het aantal hash herhalingen die zullen worden uitgevoerd.

In PHP 5.2 en later, is er een ingebouwde functie `password_hash()`.

```
public static function make($string)
{
    return password_hash($string, PASSWORD_BCRYPT, array(
        'cost' => 12
    ));
}
```

Figuur 16 Bcrypt Encryptie wachtwoord

Wanneer het hash-algoritme is verouderd zal het portaal automatisch het wachtwoord opnieuw encrypteren met een nieuwer hash-algoritme.

```
if (password_needs_rehash($this->data->password, PASSWORD_BCRYPT, $options)) {  
    $new_hash = password_hash($password, PASSWORD_BCRYPT);  
    $passwordUpdate = $this->db->update('users', $this->data()->id, array(  
        'password' => $new_hash,  
    ));  
    echo "Your logged in and with a changed password hash.";
```

FIGUUR 17 ENCRYPTIE VAN OUDERE WACHTWOORDEN

4.2 Vermijden van Cross-Site Scripting (XSS)

Een cross-site scripting aanval, ook bekend als XSS, is één van de top 5 security aanvallen die op dagelijkse basis over het internet worden uitgevoerd.

De aanval is eigenlijk een soort van code-injectie die mogelijk wordt gemaakt door het verkeerd valideren van gebruikersgegevens. Deze gebruikersgegevens worden meestal in de pagina ingegeven via een webformulier of met behulp van een veranderde hyperlink.

De ingegeven data kan schadelijke client-side code bevatten, zoals JavaScript, VBScript, HTML, CSS, Flash, enzovoort. Deze code wordt gebruikt om schadelijke data op te slaan op de server of het uitvoeren van een schadelijke actie binnen de browser van de gebruiker.

Het leveranciersportaal zorgt ervoor dat alle data die wordt ingeven door een gebruiker zal worden gezuiverd van Javascript code of iets dergelijks. Ook zal alle data afkomstig uit de database worden gefilterd.

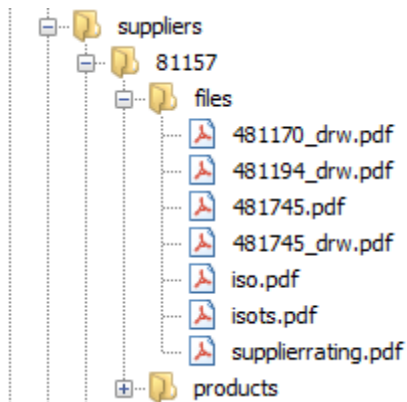
```
function escape($string) {  
    return htmlentities($string, ENT_QUOTES, 'UTF-8');  
}
```

FIGUUR 18 DATA ZUIVEREN

4.3 Bestanden

De bestanden zijn opgeslagen op de website per leverancierscode. Elke leveranciers heeft een directory met zijn code en hierin zullen de respectievelijke bestanden in staan.

Op het leveranciersportaal worden deze verschillende bestanden beveiligd door gebruik te maken van een .htdocs bestand.



FIGUUR 19 MAPPENSTRUCTUUR

Wanneer een bestand wordt opgevraagd om te downloaden, wordt er eerst gecontroleerd of deze gebruiker wel is ingelogd en of hij de juiste rechten heeft.

```

if ($user->isLoggedIn()) {
    if (!empty($_GET['name'])) {
        $filename = escape($_GET['name']);
        if (!empty($_GET['directory'])) {
            $directory = "upload/files/" . escape($_GET['directory']);
        } else {
            $directory = "upload/files";
        }
        Filesystem::get($filename, $directory);
    }
} else {
    Redirect::to('404');
}

```

FIGUUR 20 HEEFT DE GEBRUIKER RECHTEN

4.4 Sessie fixatie

Het inlog-systeem van het leveranciersportaal maakt gebruik van PHP sessies. Deze sessies zorgen ervoor dat er op een bepaalde manier informatie wordt bewaard over de gebruiker tussen opeenvolgende http-verzoeken.

Deze standaard-extensie op PHP is perfect voor het bijhouden van allerlei tijdelijke data zolang het op een veilige manier gebeurt. Het gebruik van een sessie begint met de functie “session_start()” waardoor een sessie voor een bepaalde gebruiker aangemaakt wordt. Een identifier wordt gegenereerd om de gebruiker te identificeren bij zijn verdere bezoek van de site. Deze identifier draagt meestal “PHPSESSID” als naam, maar dit kan aangepast worden in de php.ini. Deze unieke identifier wordt bij een volgend serververzoek in de vorm van een cookie of GET-argument meegegeven en identificeert op die manier de gebruiker.

Hierin schuilt het gevaar, want dit opent de mogelijkheid om de identificatie-string manueel te kiezen.

De bescherming hiertegen is vrij eenvoudig. De functie “session_regenerate_id()” zal om het halfuur de identifier voor de sessie wijzigen.

```
if (!isset($_SESSION['generated']) || $_SESSION['generated'] < (time() - 30)) {  
    session_regenerate_id();  
    $_SESSION['generated'] = time();  
}
```

FIGUUR 21 SESSION FIXATION

4.5 Sessie Hijacking

Buiten het zelf opleggen van een sessie-ID, kan een hacker de sessie identificatie-string ook op andere manieren te weten komen. Bijvoorbeeld door het analyseren of onderscheppen van het netwerkverkeer. In de beveiliging voorkomen we het onderscheppen van de identificatie-string niet, maar voeren we extra controles in om ervoor te zorgen dat het http-verzoek van de bezoeker kan onderscheiden worden van dat van de hacker.

Een http-verzoek is bedacht om staatloos te zijn. Dat wil zeggen dat een verbinding, tussen twee entiteiten, bestaat gedurende de tijd die nodig is om een verzoek naar een server te sturen en een resulterend antwoord van een client te ontvangen. Zodra deze overdracht is afgerond, zijn de twee entiteiten niet meer bewust van elkaar.

Het probleem met dit systeem is dat het gebruikt wordt voor veel geavanceerde methodes zoals transacties of aankopen op het web. Dit zorgt ervoor dat http-communicatie gebruik maakt van veel verschillende TCP-verbindingen. Daarom heeft de webserver behoefte aan een methode om de verbindingen van elke transactie of gebruiker te controleren.

De meest bruikbare methode is afhankelijk van een sessie pakket dat de webserver stuurt naar de browser na bijvoorbeeld een succesvolle client-authenticatie. Dit pakket wordt meestal opgeslagen op de server als een tijdelijk bestand voorzien van een ID, meestal bestaande uit een willekeurig getal plus de tijd en datum van wanneer de sessie gestart is. Men noemt dit pakket ook wel een sessie token en kan gebruikt worden op verschillende manieren, zoals in de URL of in de cookie-header van http.

Wanneer een hacker toegang krijgt tot deze gegevens kan het bijvoorbeeld de informatie van een legitieme gebruiker achterhalen.

Om dit te voorkomen maakt het leveranciersportaal gebruik van de `session.use_only_cookies` instelling om ervoor te zorgen dat PHP de sessie-ID slechts met een cookie kan beheren. Dit overschrijft automatisch het gebruik van transparante sessie-id's. `Session.use_trans_sid` wordt ook uitgeschakeld om te voorkomen dat de sessie-ID wordt gelekt in alle URI's.

```
// Prevent Session Hijacking
ini_set('session.use_only_cookies', true);
ini_set('session.use_trans_sid', false);

session_start();
```

FIGUUR 22 SESSION HIJACKING

4.6 Cross-site request vervalsing

Cross-site request vervalsing of CSRF is een vorm van session hijacking en staat ook wel bekend als een one-click attack of session riding. Het is een soort van kwaadaardig exploiteren van een website waarbij ongeautoriseerde commando's van een gebruiker die de website vertrouwt worden verzonden. In tegenstelling tot cross-site scripting (XSS), die het vertrouwen van een gebruiker voor een bepaalde site exploiteert, exploiteert CSRF het vertrouwen dat een site heeft in de browser van een gebruiker.

De aanval werkt door het toevoegen van een link of een script op een pagina van een site waarop de gebruiker bekend is of verondersteld geautoriseerd te zijn. Een gebruiker zou bijvoorbeeld kunnen surfen op een chat-forum en reacties plaatsen waartoe hij eigenlijk geen recht tot heeft.

Om dit tegen te gaan, zal het leveranciersportaal aan elk webformulier een geheime “token” toevoegen. Deze geheime token of code wordt bij het invoegen van gegevens vergeleken met de token van een sessie. Een sessie is een pakketje dat informatie bevat over een bepaalde handeling. Wanneer de token van het webformulier en van de sessie gelijk zijn, wordt verondersteld dat het formulier correct werd verstuurd.


```
public static function check($token)
{
    $tokenName = Config::get('session/token_name');

    if (Session::exists($tokenName) && $token === Session::get($tokenName)) {
        Session::delete($tokenName);
        return true;
    }

    return false;
}
```

FIGUUR 23 VOORKOMEN VAN CSRF VERVALSING

4.7 Groepen en Permissies

4.8 Htdocs

4.9 SQL-injectie

5. Features

5.1 Web tracking system

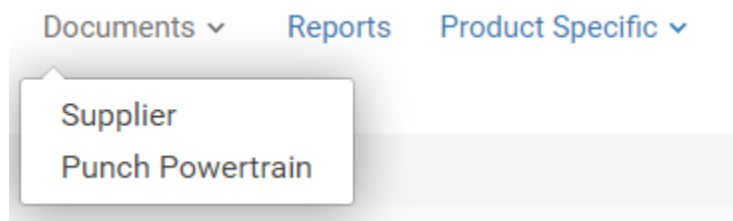
5.1.1 PIWIK

5.2 Control management system

5.3 Bootstrap

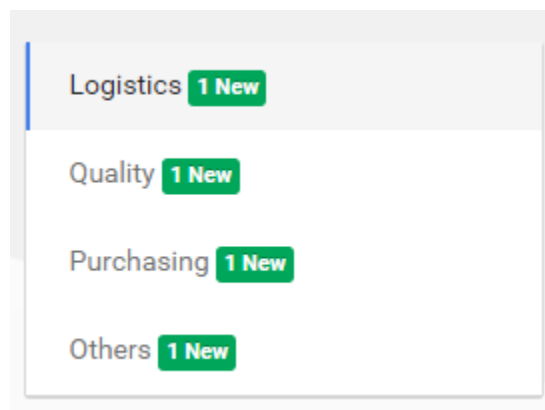
6. Documenten & Rapporten leveranciersportaal

Documenten en rapporten kunnen op vier verschillende plaatsen in het portaal worden bekeken en gedownload nl. Punch Powertrain documents, Supplier documents, Reports en Product specific. Naargelang hun inhoud worden ze onder deze vier categorieën verdeeld.



6.1 Documenten van Punch Powertrain

Dit zijn algemene documenten en richtlijnen die voor elke leverancier grotendeels hetzelfde zijn.



FIGUUR 24 DOCUMENTEN PUNCH POWERTRAIN

6.1.1 Logistics

Hier kan men de logistieke documenten die van toepassing zijn downloaden.

General preservation guidelines:

G:\Documentenbeheer\werkinstructies\SCM_HB\W_SCM_001_02_preservation guidelines.pdf

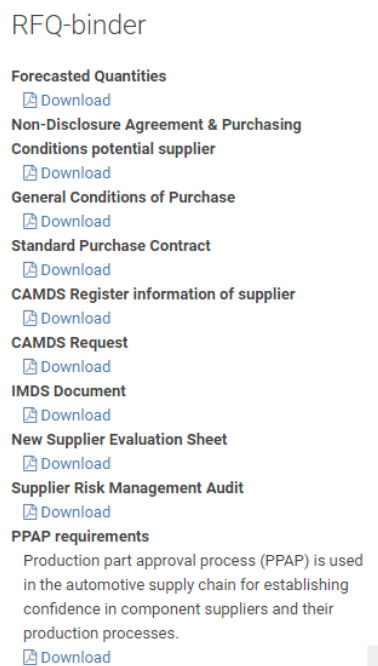
6.1.2 Quality

Hier kan men de quality documenten die van toepassing zijn downloaden.

Cleanliness Specifications:

6.1.3 Purchasing

Hier kan de leverancier een groot deel van de RFQ-binder (Request for quotation) downloaden. Onderstaande afbeelding toont de documenten die daartoe horen.



FIGUUR 25 VOORBEELD RFQ-BINDER

RFQ-binder:

6.1.4 Others

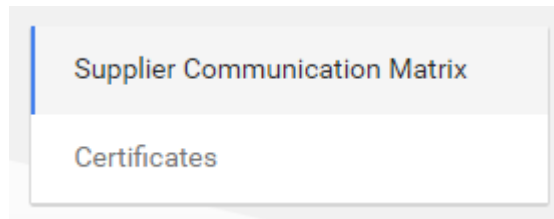
Hier kan men de overige bestanden downloaden.

Vendor performance & announcement:

G:\

6.2 Documenten van Leverancier

Dit zijn twee formulieren van de leverancier nl. de communicatie matrix en zijn certificaten. Deze formulieren kunnen volledig worden aangepast en de certificaten kunnen worden gedownload en geüpload.



FIGUUR 26 DOCUMENTEN LEVERANCIER

6.2.1 Leverancier Communicatie Matrix

Dit is een bestand met daarin de gegevens van de leverancier, verkoopafdeling, kwaliteit afdeling, logistieke afdeling en financiële verantwoordelijke. Deze gegevens kunnen door de leverancier worden aangepast indien nodig.

SUPPLIER INFO	
Supplier name	[REDACTED]
Supplier code	[REDACTED]
Supplier email	[REDACTED]
SALES	
Sales Orders	Name [REDACTED]
	E-mail blaiba@email.com
	Phone number [REDACTED]
	Mobile Phone [REDACTED]
	Fax Number 047
(Sales) Director	Name director
	Title director
	E-mail [REDACTED]
	Phone number 8484848
	Mobile Phone 8885
Account Manager	Name something
	Title something
	E-mail [REDACTED]
	Phone number something
	Mobile Phone something
QUALITY	
Quality	Name quality
	Title title
	E-mail [REDACTED]
	Phone number 2555
	Mobile Phone

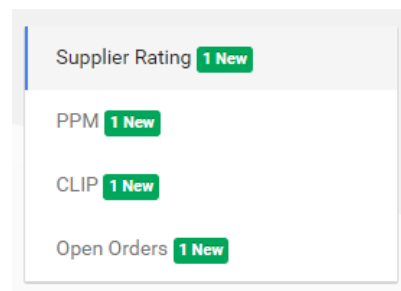
FIGUUR 27 LEVERANCIER COMMUNICATIE MATRIX

6.2.2 Certificaten

Dit zijn twee bestanden namelijk de ISO 1400 1 en ISO-TS certificaten. Ze kunnen gedownload en geüpload worden door de leverancier.

6.3 Rapporten

Er zijn vijf soorten rapporten die uniek zijn voor elke leverancier. Ze komen ook uit verschillende backend systemen van Punch Powertrain.



FIGUUR 28 RAPPORTEN LEVERANCIER

6.3.1 Leveranciersbeoordeling

De leveranciersbeoordeling is een bestand waarin de totale beoordeling van de leverancier in vermeld staat. Deze totale beoordeling bekomt men door een som te maken van de scores van onder meer: leverbetrouwbaarheid, commerciële en corporate performance en totale kwaliteitsperformance.

6.3.2 PPM

Process Performance Measure per jaar.

Latest 12 month PPM:

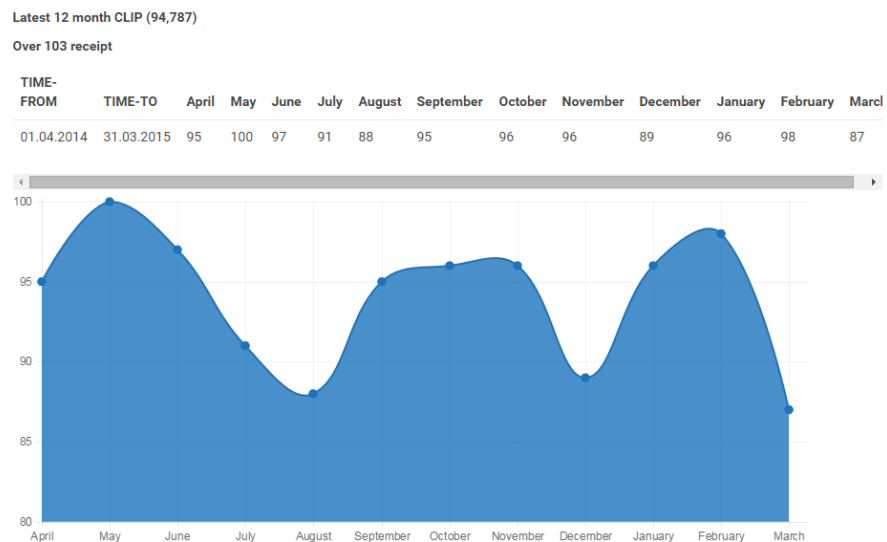
Babtec → .csv bestand

6.3.3 CLIP

CLIP score per jaar. Dit is een grafiek met gegevens van elke maand voor één jaar. Deze grafiek wordt gegenereerd door data te filteren uit he .CSV bestand waarin alle clip-scores in vermeld staan.

Latest 12 month CLIP:

SAP → .csv bestand



FIGUUR 29 CLIP RAPPORT

6.3.4 Openstaande orders

Dit is een lijst met nog niet geleverde orders van de leverancier.

Open Orders

Document Date	Short Text	Order Quantity	Order Unit	Still to be delivered (qty)	Still to be delivered (value)	Still to be invoiced (qty)	Still to be invoiced (val.)	Purchasing Document
7/01/2014	vouwdoos primaire pulley bedrukt	29.300	PC	3.950	6.952,00	4.130	7.268,80	101075
7/01/2014	omdoos secundaire pulley	1.290	PC	164	1.033,20	164	1.033,20	101075

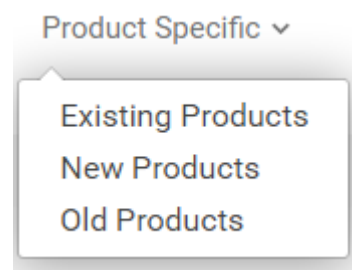
FIGUUR 30 OPEN ORDERS RAPPORT

6.3.5 Back-log

Zafr lijst:

6.4 Specifieke Producten

Hier kan de leverancier zijn bestaande, nieuwe of oude producten bekijken. De nieuwe producten kan hij zelf ingeven.



FIGUUR 31 PRODUCTEN LEVERANCIER

Het principe voor de bestaande, nieuwe en oude producten is hetzelfde. Voor elke product bestaat een unieke materiaalcode. Wanneer de leverancier naar zo'n materiaalcode van een product gaat, vindt hij verschillende bestanden voor dit product.

Deze bestanden zijn onder meer: “PPAP-document, 8D-report, Packing Instruction,...”.

PPAP-DOCUMENT	IMDS-Confirmation DOCUMENT
8D-report	Credit Note (pcs)
Chargeback	Drawing
Packing Instruction	

FIGUUR 32 BESTANDEN VOOR ELK PRODUCT

7. Besluit

7.1 Aanbevelingen

7.2 Persoonlijke ervaring

Door enkele maanden aan dit project te werken in een bedrijf leer je snel hoe het er in het bedrijfsleven aan toe gaat. Je leert de werknemers en stilaan ook het bedrijf kennen. Met deze ervaring achter de rug kan je voorbereid naar het echte bedrijfsleven stappen.

Een van de belangrijkste dingen die ik heb bijgeleerd tijdens deze periode is dat een software programma nooit volledig af is. Er zijn altijd nog dingen die verbeterd en aangepast kunnen worden.

De grote vrijheid in het zoeken naar de beste werkmethode gecombineerd met de hulp die ik kreeg zorgde ervoor dat de project zich vlot ontwikkelde en snel vorm kreeg

Literatuurlijst

[1] **PDO Verbinden met verschillende databases - PHP**

tutorials Opgevraagd Mei 04, 2015, van

<http://www.phphulp.nl/php/tutorial/overig/pdo-verbinden-met-verschillende-databases/534/inleiding/1364/>

[2] **MySQL** Opgevraagd Mei 04, 2015, van

<http://nl.wikipedia.org/wiki/Mysql>

[3] **NetBeans** Opgevraagd Mei 04, 2015, van

<http://nl.wikipedia.org/wiki/Netbeans>

[4] **Object georiënteerd** Opgevraagd Mei 04, 2015, van

<http://nl.wikipedia.org/wiki/Objectgeori%C3%ABnteerd>

[5] **The PHP Group. PHP: Traits - Manual. PHP: Hypertext Preprocessor.** [Online]

The PHP Group. Opgevraagd Mei 04, 2015, van

<http://php.net/manual/en/language.oop5.traits.php>.

[6] **PDO Tutorial for MySQL Developers** - Hashphp.org.

Opgevraagd Mei 04, 2015, van

http://wiki.hashphp.org/PDO_Tutorial_for_MySQL_Developers

- [7] **PHP** Opgevraagd Mei 04, 2015, van
<http://nl.wikipedia.org/wiki/Php>
- [8] **PHP: mysql_connect - Manual**. Opgevraagd Mei 04, 2015, van
<http://be2.php.net/function.mysql-connect>
- [9] **Van der Put, W. (2009). Het beste van PHP en MySQL** (p. 223).
- [10] **Van Emde Boas, G. (2005). PHP 5 en MySQL** (p. 261).
- [11] **waarom en hoe OOP - Programmeren Opgevraagd** Mei 04, 2015, van
<http://www.wetenschapsforum.nl/index.php/topic/183077-waarom-en-hoe-oop/>
- [12] **Why you Should be using PHP's PDO for Database Access - Tuts+ Code Tutorial**. Opgevraagd Mei 04, 2015, van
<http://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059>

Bijlagen

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Ontwikkelen van een leveranciersportaal

Richting: **master in de industriële wetenschappen: elektronica-ICT**

Jaar: **2015**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Claes, Ruben

Datum: **5/06/2015**