



Afstudeerverslag

Search Traffic Monitor

Bedrijf: OrangeValley Consultancy B.V.

Student: F. Perdaan, 08011249

Examinatoren:

Dhr. A.A. Nederend

Mevr. A.A.A.M. Jacobs

Datum

1 Juni 2012

Auteur

Ferdy Perdaan

Versie

1.0

DILBERT by Scott Adams





Search Traffic Monitor

Titel	: Search Traffic Monitor
Student	: F. Perdaan
Studentnummer	: 08011249
Onderwijsinstelling	: De Haagse Hogeschool
Opleiding	: Informatica
Afstudeerperiode	: 06-02-2012 / 01-06-2012
Examinatoren	: dhr. A.A. Nederend mevr. A.A.A.M. Jacobs
Bedrijf	: OrangeValley Consultancy B.V.
Afdeling	: 42functions - Webdevelopment
Bedrijfsbegeleider	: dhr. E. Blacquièrè
Versie	: 1.0



Referaat

Perdaan, F, Afstudeerverslag – 'Search Traffic Monitor', Breukelen, OrangeValley Consultancy B.V., Juni 2012.

Dit afstudeerverslag ter afronding van de opleiding Informatica aan De Haagse Hogeschool te Zoetermeer beschrijft het proces rondom de werkzaamheden die, gedurende de periode van februari tot en juni 2012, in het kader van de genoemde opdracht 'Search Traffic Monitor' zijn uitgevoerd. Middels dit afstudeerverslag wordt inzichtelijk gemaakt welke werkzaamheden hiervoor ondernomen zijn.

Gedurende de afstudeerperiode heeft de afstudeerder zich bezig gehouden met het verder ontwikkelen van Search Traffic Monitor. Search Traffic Monitor is een applicatie ontwikkeld door OrangeValley met het doel om web consultants te ondersteunen in het analyseren van websites. De werkzaamheden gedurende het afstuderen omvat het uitvoeren van een performanceanalyse en het uitbreiden van de applicatie met een sales module; welke nieuwe inzichten verschaft betreffende conversiewaardes.

Descriptoren

- Search Engine Marketing
- Performanceanalyse
- STRICA
- Scrum
- Bezoekersgedrag
- Zoekmachinegedrag



Versiebeheer

Versie	Datum	Auteur	Mutatie
0.8	15-03-2012	Ferdy Perdaan	+ Opzet initiële structuur
	26-03-2012	Ferdy Perdaan	+ § 3.1, § 3.2, § 3.3, § 3.5, § 3.7
	30-03-2012	Ferdy Perdaan	+ § 3.4
	04-04-2012	Ferdy Perdaan	+ § 4.1.1
	10-04-2012	Ferdy Perdaan	+ § 6.3.2, § 6.4
	16-04-2012	Ferdy Perdaan	+ § 4.2, § 4.3, Reviseren
	16-04-2012	Arno Nederend	Feedback
	02-05-2012	Ferdy Perdaan	+ § 4.1.2, § 4.1.3, Reviseren
	08-05-2012	Ferdy Perdaan	Reviseren
	11-05-2012	Ferdy Perdaan	+ § 7, Reviseren
	12-05-2012	Ferdy Perdaan	+ § 6.3, Voorblad, Referaat
	14-05-2012	Ferdy Perdaan	+ § 9.4, Reviseren
	15-05-2012	Arno Nederend, Nanny Jacobs	Feedback
0.9	15-05-2012	Ferdy Perdaan	Wijzigingen opbouw document
	15-05-2012 t/m 21-05-2012	Ferdy Perdaan	Reviseren kern
	22-05-2012	Arno Nederend	Feedback
	22-05-2012	Ferdy Perdaan	Verbeteren Scrum termen, Reviseren kern
	23-05-2012	Ferdy Perdaan	Reviseren kern
	25-05-2012	Ferdy Perdaan	Reviseren kern
	26-05-2012	Ferdy Perdaan	+ § 5, § 8.3, Reviseren
	27-05-2012	Ferdy Perdaan	Reviseren
	28-05-2012	Ferdy Perdaan	+ § 7.3, Inleiding, Reviseren § 9.4
	28-05-2012	Sander Benschop	Feedback
	29-05-2012	Ferdy Perdaan	+ § 7.1.4, Reviseren
	30-05-2012	Ferdy Perdaan	Reviseren
1.0	31-05-2012	Ferdy Perdaan	+ § 9.1, § 9.2, § 9.3, Voorwoord
	31-05-2012	Eduard Blacquièrre	Feedback



Voorwoord

Dit verslag beschrijft het proces rondom de werkzaamheden die ik tijdens het afstuderen heb verricht. Met het doel inzicht te verschaffen in de afwegingen en keuzes die ik gedurende het afstuderen heb gemaakt.

Voor de totstandkoming van dit verslag wil ik allereerst Arno Nederend bedanken voor zijn rol als begeleidend examiner. Met name het vertrouwen en de extra hulp bij het schrijven van het afstudeerverslag waardeer ik ten zeerste. Ten tweede wil ik Nanny Jacobs bedanken voor haar rol als tweede examiner en daarmee alle feedback die ze heeft gegeven.

Verder wil ik mijn studiegenoot Sander Benschop bedanken voor alle feedback die hij heeft gegeven ter verbetering van mijn afstudeerverslag.

Tenslotte wil ik Eduard Blacqui re bedanken voor zijn rol als product owner, zijn rol als bedrijfsbegeleider en zijn constante vertrouwen in de ontwikkeling van Search Traffic Monitor. Zonder hem had Search Traffic Monitor niet bestaan zoals we deze nu kennen.

Ferdy Perdaan

Zoetermeer, juni 2012



Inhoudsopgave

1	Inleiding	8
2	OrangeValley	10
2.1	OrangeValley	10
2.2	42functions	11
3	Opdracht	12
3.1	Search Traffic Monitor	12
3.2	Doelstelling	13
3.3	Werkwijze	14
3.4	Op te leveren producten	15
3.4.1	Adviesrapport performanceproblemen	15
3.4.2	Specifieke statistische functionaliteit	15
3.4.3	Sales module	16
3.5	Betrokkenen	17
3.6	Afbakening	18
3.7	Planning	18
4	Adviesrapport	19
4.1	Werkwijze	19
4.1.1	Reeds bekende bottlenecks	20
4.1.2	Systeemprofilering	20
4.1.3	Query analyse	22
4.2	Bevindingen	24
4.3	Conclusie	26
5	Requirements	27
5.1	Product backlog	27
5.2	Rapportage	28
5.3	Management	29
6	Architectuur	32
6.1	Wireframes	32
6.2	Mockup	33
6.3	Technisch ontwerp	34
6.3.1	Architectuur	34
6.3.2	HTML/JS: View	35
6.3.3	PHP: Controller	35
6.3.4	Python: Data interface	36
6.3.5	MongoDB: Webdatabase	37
6.3.6	MySQL: Management database	38
6.3.7	PHP: Dataverwerking	38
6.4	Migratie	39
6.5	Prototype	40
7	Realisatie	41
7.1	Testen	41
7.1.1	Productietest	41
7.1.2	Unittest	42
7.1.3	Integratie- en acceptatietest	50
7.1.4	Overdracht	50
7.2	Rapportage	51
7.3	Management	53



8	Operationeel werk	54
8.1	<i>Google Analytics SiteCheck</i>	55
8.2	<i>Google AdWords Dynamische Advertenties</i>	56
8.3	<i>Reparatie Rate Your Company</i>	57
9	Evaluatie	58
9.1	<i>Producten</i>	58
9.2	<i>Werkwijze</i>	60
9.3	<i>Eervolle vermelding</i>	61
9.4	<i>Beroepstaken</i>	62
10	Begrippenlijst	64
11	Bijlagen	67



1 Inleiding

Gedurende de periode van februari tot juni 2012 is de afstudeerder werkzaam geweest als developer bij 42functions, een dochteronderneming van OrangeValley. In deze periode heeft hij zich gericht op het verder ontwikkelen van één van OrangeValley's innovatieprojecten genaamd Search Traffic Monitor.

Het doel van de afstudeeropdracht is om een performanceanalyse uit te voeren waarin verbeterpunten voor Search Traffic Monitor's huidige performanceproblemen worden voorgedragen. Na het afronden van het adviesrapport houdt de afgestudeerde zich bezig met het ontwikkelen van een sales module als uitbreiding op Search Traffic Monitor.

Het afstudeerverslag is geschreven met als doel inzicht te geven in de werkzaamheden, keuzes en afwegingen van de afstudeerder gedurende het afstudeertraject. Aan de hand van dit afstudeerverslag kunnen de examinatoren beoordelen of de afstudeeropdracht met afdoende diepgang is afgerond en of hiermee de vooraf gedefinieerde beroepstaken zijn behaald.

Dit afstudeerverslag is opgebouwd uit de volgende hoofdstukken

- *OrangeValley*

Het eerste hoofdstuk wordt OrangeValley geïntroduceerd. In dit hoofdstuk worden de diensten van OrangeValley nader beschreven ter achtergrondinformatie voor de afstudeeropdracht.

- *Opdracht*

Het tweede hoofdstuk is een samenvatting van het plan van aanpak. In dit hoofdstuk worden Search Traffic Monitor, de doelstelling, de werkwijze, de afbakening en de risico's nader toegelicht.

- *Adviesrapport*

Het derde hoofdstuk beschrijft uitvoerig hoe het adviesrapport tot stand is gekomen, vanuit welke perspectieven het rapport is opgesteld en de conclusies waartoe het rapport heeft geleid.

- *Requirements, Architectuur en Realisatie*

Het vierde hoofdstuk omschrijft de requirements die gedurende het afstudeertraject worden gepraktiseerd. In hoofdstuk vijf, architectuur, worden deze requirements omgezet naar een architectuur welke in hoofdstuk zes, realisatie, wordt uitgewerkt.



- *Operationeel werk*

Het zevende hoofdstuk gaat in op de operationele werkzaamheden die onverwacht zijn uitgevoerd. Dit hoofdstuk specificeert zich tot de werkzaamheden welke een raakvlak hebben met de afstudeeropdracht.

- *Evaluatie*

In het achtste hoofdstuk wordt er een evaluatie gegeven over de opgeleverde producten en de werkwijze die gehanteerd is om deze producten te realiseren.

- *Begrippenlijst en Bijlagen*

Ten slotte volgt er een begrippenlijst die vaktermen toelicht in het afstudeerkader, deze begrippen worden doorgaans herkenbaar gemaakt middels een voetnoot. De begrippenlijst wordt gevolgd door een bijlage waarin alle documentatie relevant voor het afstudeerverslag is opgenomen.



2 OrangeValley

2.1 OrangeValley

OrangeValley is een consultancybureau gespecialiseerd in het verbeteren van het online rendement op websites. Websites worden altijd opgezet met een doel, het is de taak van OrangeValley om ervoor te zorgen dat zoveel mogelijk bezoekers bijdragen aan dit doel. Zo kan het doel van een webshop bijvoorbeeld zijn dat de bezoeker een item op de website aanschafft.

Voor het verhogen van de conversie op een website biedt OrangeValley verschillende diensten aan. Eén van de belangrijkste diensten hiervoor is Search Engine Marketing (SEM) waarbij men zich richt op het binnen halen van zo veel mogelijk bezoekers via zoekmachines.

Binnen Search Engine Marketing worden er op twee manieren bezoekers geworven. Waar Search Engine Advertising zich richt op bezoekers die binnen komen via advertenties (betaald zoekverkeer) richt Search Engine Optimisation zich op bezoekers die via de gewone zoekresultaten binnen komen (organisch zoekverkeer).

The screenshot shows a Google search for 'autoverzekering'. The search bar at the top contains the text 'autoverzekering' and a magnifying glass icon. Below the search bar, it says 'Zoeken' and 'Ongeveer 3.190.000 resultaten (0,24 seconden)'. The results are divided into two main sections: 'Advertenties met betrekking tot autoverzekering' (Paid) and 'Advertenties - Waarom deze advertenties?' (Organic). The 'Betaald' section includes ads from 'Goedkope autoverzekering? | Independer.nl', 'Autoverzekering Berekenen' from 'www.rabobank.nl/', 'Goedkope Autoverzekering' from 'www.ditzo.nl/Autoverzekering', 'Goede autoverzekering' from 'www.centraalbeheer.nl/Autoverzekering', and 'Goedkope Autoverzekering?' from 'www.klaverblad.nl/autoverzekering'. The 'Organisch' section includes results from 'Independer.nl voor de beste autoverzekering van 2012 (tip!)', 'Autoverzekering: vergelijk online de goedkoopste auto verzekering.', 'Autoverzekering ANWB - Auto verzekering als beste uit de test!', and 'Premie Autoverzekering ... - Verzekeringen - Autoverzekering Afsluiten'.

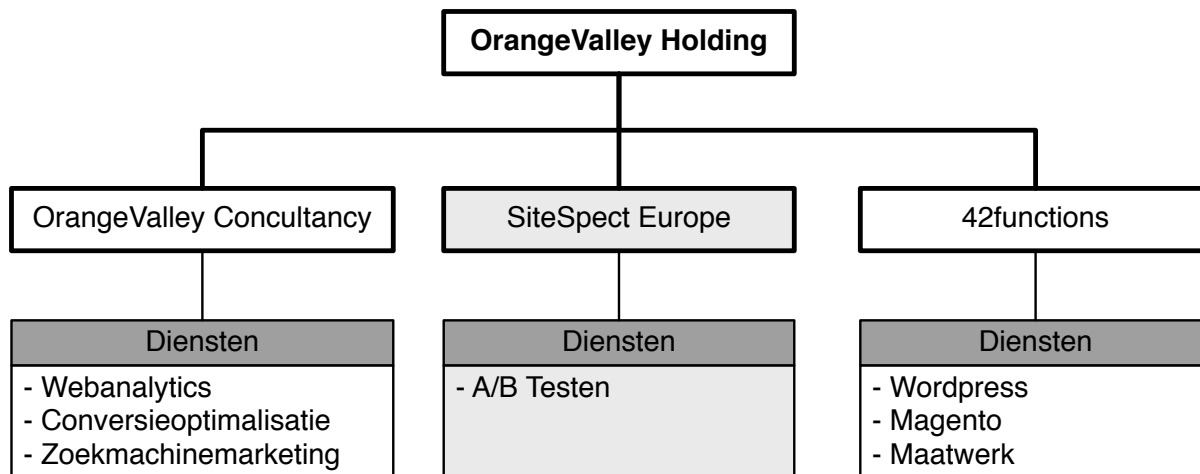
Bezoekers komen binnen via advertenties of via de gewone zoekresultaten.

Voor het genereren van een zoekresultaat maakt de zoekmachine gebruik van een serie complexe berekeningen om te zoeken welke websites relevant zijn voor de gegeven zoekterm, het is het doel van Search Engine Optimisation om een zo hoog mogelijke positie binnen deze resultaten te bemachtigen. Hiervoor worden vaak wijzigingen aan de website gemaakt om relevanter te zijn voor bepaalde zoektermen. Het is echter moeilijk te achterhalen wat exact de invloed van de wijziging voor de zoekmachine is geweest, terwijl dit wel iets is dat de klant vaak van tevoren wil weten. Een applicatie die in staat is om dergelijke inzichten te genereren zou voor de consultants dan ook erg handig zijn.



2.2 42functions

Om de focus van OrangeValley op conversieoptimalisatie te behouden is begin 2011 de dochteronderneming 42functions opgericht. 42function richt zich in tegenstelling tot OrangeValley volledig op webontwikkeling.



Organogram van OrangeValley Holding B.V.

Door als web consultancy bureau een eigen development dienstverlening te hebben is OrangeValley in staat om haar online marketing advies te vertalen naar concrete technische oplossingen. Hiermee onderscheid zij zich van haar concurrenten en is ze in staat om innovatieve projecten als Calltracking, Audience Targeting en Search Traffic Monitor te starten.

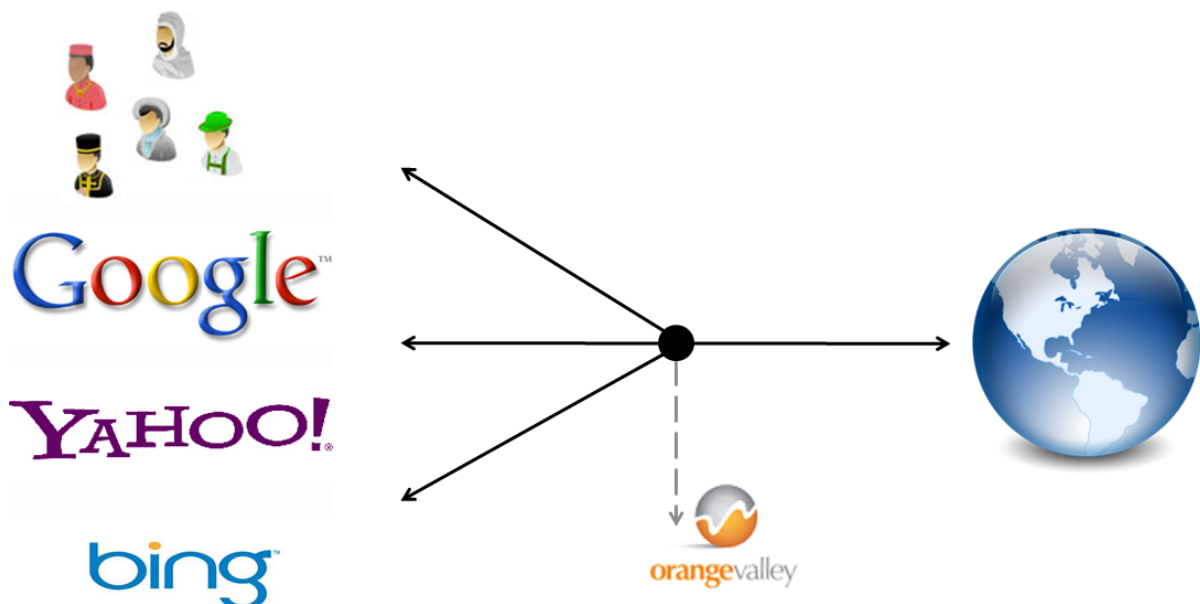


3 Opdracht

3.1 Search Traffic Monitor

Search Traffic Monitor is een applicatie ontwikkeld door OrangeValley die in staat is innovatieve rapportages te genereren op het gebied van Search Engine Marketing. Hetgene wat deze rapportage uniek maakt is dat er naast het bezoekersgedrag ook het gedrag van zoekmachines wordt geanalyseerd, een gegeven wat vaak nog mist binnen de Search Engine Marketing analyses.

Search Traffic Monitor is in 2009 begonnen als een kleine zelfstandige applicatie welke op de website van de klant geïnstalleerd werd om extra informatie over bezoekers en zoekmachines te registreren. Search Traffic Monitor kon vervolgens op basis van de verzamelde data vertellen met welke regelmaat de zoekmachines langs kwamen en via welke positie binnen de zoekresultaten het organisch zoekverkeer¹ binnen zijn gekomen.



Search Traffic Monitor monitort de communicatie tussen de server en de bezoekers.

Onderhand is Search Traffic Monitor uitgegroeid tot een multi-server oplossing welke haar data verzameld middels logfileanalyse². Tevens is ze in staat om haar data te combineren met externe bronnen als Google Analytics, Google Adwords, Pingdom en SiteSpect om zo nog meer inzichten te kunnen genereren. Zo kan het bijvoorbeeld interessant zijn om te zien hoe het crawlgedrag³ zich verhoudt tot down time van de server.

¹ Bezoekers die met behulp van een zoekmachine op een website terecht komen.

² In logfiles wordt voor elk serververzoek gebruikersinformatie weggeschreven

³ Crawler: AI welke in staat is geautomatiseerd webpagina's op te halen



Het uiteindelijke doel van Search Traffic Monitor is om genoeg data te verzamelen opdat het gehele STRICA¹ traject in kaart kan worden gebracht. Met behulp van het complete STRICA traject kan er exact aangetoond worden wat de exacte effecten zijn van Search Engine Marketing. Het volledig in kaart brengen van dit traject kan ook wel vergeleken worden met het vinden van de heilige graal.

Stap	Datum	Omschrijving
A	23-03-2012 14:15	Paginatitel gewijzigd
C	23-03-2012 17:52	Google komt langs
I	24-03-2012 09:32	Wijziging opgenomen in index
R	24-03-2012 09:32	Positie toegenomen met drie plekken
T	31-03-2012 12:00	Bezoek is met 25% toegenomen ten opzichte van vorige maand
S	31-03-2012 12:00	Een toename van 25% heeft geleid tot 6% extra conversie ter waarde van €42,-

Voorbeeld STRICA, groene onderdelen kunnen reeds worden geanalyseerd

3.2 Doelstelling

Een van de doelen voor Search Traffic Monitor is dat deze op korte termijn kan worden toegevoegd aan de standaard 'gereedschapskist' van OrangeValley's consultants. Search Traffic Monitor moet de consultants in haar werkzaamheden ondersteunen door bepaalde activiteiten te automatiseren en aanvullende rapporten te genereren betreffende het gedrag van zoekmachines en bezoekers.

Voor het genereren van dergelijke rapporten heeft Search Traffic Monitor momenteel vijf minuten tot enkele uren nodig. Om efficiënt met de applicatie te kunnen werken is door het management besloten dat deze wachttijd moet worden teruggedrongen naar een maximum van vijf seconden. Het is dan ook noodzakelijk dat de performance van de applicatie grondig geoptimaliseerd.

Naast het verbeteren van de performance is het tijd om de volgende stap richting STRICA te zetten, daarvoor wordt Search Traffic Monitor uitgebreid met een sales module. Door de applicatie uit te breiden met een sales module kan Search Traffic Monitor in het vervolg ook inzichten creëren op het gebied van SEA².

¹ Sales Traffic Ranking Indexing Crawling Adjustment

² Search Engine Advertising



3.3 Werkwijze

Alvorens er begonnen kan worden aan de ontwikkeling van een nieuwe sales module is het noodzakelijk dat het adviesrapport wordt opgesteld. De bouw en het ontwerp van de sales module is namelijk sterk afhankelijk van de bevindingen die voortvloeien uit het adviesrapport. Indien er bijvoorbeeld geadviseerd wordt om gebruik te gaan maken van een alternatief databasemanagementsysteem kan de product owner ervoor kiezen om de sales module in een apart traject te ontwikkelen waarin gelijk gebruik wordt gemaakt van het alternatieve databasemanagementsysteem.

Na het afronden van het adviesrapport wordt er met behulp van Scrum een iteratief traject opgezet waarmee de sales module wordt ontwikkeld. Hiervoor worden sprints van twee weken gehanteerd, dit in verband met het verzoek van de stakeholders om elke twee weken een voortgangsrapportage op te leveren.

Voor het indelen van de sprints wordt er eerst samen met de stakeholders, SEO en SEA consultants een product backlog opgesteld waarin alle gewenste functionaliteiten als backlog-items worden opgenomen. Op basis van een inspanning/opbrengst analyse worden de backlog-items door de product owner aan de sprints toebedeeld. In deze analyse wordt de ontwikkeltijd afgewogen tegen de bedrijfswaarde voor de stakeholders.

Activiteiten

- Opstellen adviesrapport performanceproblemen
- Realisatie specifieke specialistische functionaliteit
- Realisatie sales module
- Opstellen product backlog

Activiteiten [iteratief op basis van sprint]

- Opstellen technisch ontwerp
- Opzetten testtraject
- Realisatie functionaliteit
- Opstellen burndown chart
- Opstellen sprint backlog



3.4 Op te leveren producten

3.4.1 Adviesrapport performanceproblemen

Het adviesrapport bevat een grondige analyse van Search Traffic Monitor's performanceproblemen. In dit rapport wordt Search Traffic Monitor op vijf niveaus onderzocht om zo een compleet beeld van de performance te vormen. Voor elk niveau wordt de huidige situatie in kaart gebracht en wordt er gekeken naar mogelijke bottlenecks¹ die zich binnen de huidige implementatie voor doen. Voor elke bottleneck worden er vervolgens mogelijke oplossingen voorgedragen ter 'verbreding' van de bottleneck. Ter afsluiting worden de voorgedragen oplossingen tegen elkaar afgewogen op basis van een inspanning/opbrengst analyse.

Het is uiteindelijk aan de product owner om aan te geven voor welke oplossingen er gekozen gaat worden. Afhankelijk van de prioriteit die de product owner aan de performance toekent kan er besloten worden om de oplossingen gedurende de ontwikkelfase van het afstuderen al te realiseren.

3.4.2 Specifieke statistische functionaliteit

Search Traffic Monitor is een applicatie die gebruikt wordt voor het opstellen van adviesrapporten, het is daarom van uiterst belang dat de data die Search Traffic Monitor toont accuraat is. Om rekening te houden met uitschieters binnen de dataset wordt daarom vaak gebruik gemaakt van een mediaan². Zoekmachines personaliseren bijvoorbeeld de zoekresultaten steeds meer, waardoor gerapporteerde posities uitschieters kunnen bevatten. De mediaan geeft hierbij een betrouwbaarder beeld van de ranking positie dan bijvoorbeeld het gemiddelde.

Het berekenen van een mediaan is een functie die MySQL standaard niet ondersteund. Dit met het gevolg dat we de mediaan via omwegen moeten gaan berekenen, dit leidt vaak tot onnauwkeurigheid of een traag systeem. Om de mediaan vooralsnog te kunnen berekenen moeten we momenteel de data tijdelijk naar de applicatie transporteren. Hiervoor is naast het gebruik van meerdere query's extra geheugen nodig waarin de resultaten tijdelijk worden opgeslagen. In sommige gevallen wil een reeks honderden megabytes aan data beslaan wat afgewogen moet worden tegenover het maximum van 32 megabyte waarover we in PHP³ beschikken.

Om bovenstaande constructies te vermijden is het gewenst dat MySQL wordt uitgebreid met een mediaan functionaliteit middels een UDF⁴. Er is expliciet gekozen voor een uitbreiding op het databasemanagementsysteem om zo de optimale performance te behalen. Door de mediaan in de database te berekenen hoeven er geen alternatieve query's meer worden uitgevoerd, dit scheelt zowel in de dataoverdracht als de werklast op de database. In tegenstelling tot PHP heeft MySQL namelijk veel meer geheugen tot haar beschikking omdat zij ook gebruik mag maken van de harde schrijf om haar data tijdelijk weg te schrijven.

¹ Mogelijk knelpunt voor performance.

² Middelste waarde van een geordende datareeks.

³ Omdat PHP als sub proces per aanvraag gestart wordt kan het geheugen snel vol lopen wanneer er veel aanvragen plaats vinden, daarom wordt er per aanvraag maximaal 32MB geheugen gealloceerd.

⁴ User Defined Function, C++ code extensie voor MySQL distributie



3.4.3 Sales module

Een onderdeel van de afstudeeropdracht is het uitbreiden van Search Traffic Monitor met een geheel nieuwe sales module. Voor de bouw van deze module wordt er eerst met de stakeholders en een SEA consultant een product backlog met de gewenste functionaliteiten opgesteld. Deze functionaliteiten worden primair opgesteld aan de hand van grafische ontwerpen die vervolgens door de developers worden opgedeeld in technische functionaliteiten opdat deze als backlog-items aan de product backlog kunnen worden toegevoegd.

Voor het bouwen van de sales module wordt er gebruik gemaakt van test-driven development. Voor de nieuwe functionaliteiten worden eerst technische ontwerpen opgesteld om zo eventuele problemen binnen de architectuur prematuur te kunnen herkennen. Nadat de ontwerpen zijn opgesteld worden er unittests opgezet waarmee de kwaliteit van de code kan worden bewaakt. Pas nadat de testen zijn gebouwd worden de functionaliteiten pas werkelijk geïmplementeerd.

Na oplevering van de sales module moet Search Traffic Monitor in staat zijn om inzichten te verschaffen op het gebied van SEA. Hierbij wordt de focus gelegd op het tonen van de conversiewaardes. De conversiewaardes kunnen namelijk concreet worden gebruikt om aan te tonen wat de financiële resultaten zijn van zoekmachineoptimalisatie.



3.5 Betrokkenen

De volgende personen zijn betrokken bij de afstudeeropdracht

<i>Rol</i>	<i>Medewerker</i>	<i>Functie</i>
Stakeholder / Product owner	Eduard Blacquièrre	Consultant zoekmachine optimalisatie
Stakeholder	Ortwin Verreck	Managing partner
ScrumMaster / Developer	Ferdy Perdaan	Technical consultant
Developer	Mark Abspoel	Data architect
Doelgroep	SEM consultants	

Stakeholder

De stakeholders voeden de product backlog met nieuwe gewenste functionaliteiten. Binnen de scope van het afstuderen worden echter enkel functionaliteiten met betrekking tot sales module en performance in behandeling genomen.

Product owner

De product owner staat garant voor de belangen van alle stakeholders. Hij is daarmee verantwoordelijk voor het toewijzen van de backlog items aan de verschillende sprints. Hiervoor wordt aan elk backlog-item een prioriteit toegekend op gebaseerde op de bedrijfswaarde afgewogen tegen de tijd die ervoor nodig is.

ScrumMaster

De ScrumMaster wijst alle sprint items aan de projectleden. Het is aan de ScrumMaster om het overzicht te houden over de voortgang van de sprints en het algehele project. Tevens is het de taak van de ScrumMaster om alle Scrum documentatie op te stellen.

Developer

De developers zijn verantwoordelijk voor de realisatie van de gewenste functionaliteiten. Hun taken worden na elke sprint door de ScrumMaster toebedeeld.

Doelgroep

Omdat Search Traffic Monitor een applicatie is welke toegevoegd moet worden aan de standaard 'gereedschapskist' van de consultants worden deze ter zijner tijd gevraagd om mee te helpen met het testen van de applicatie. Tevens is de doelgroep in staat om backlog-items toe te voegen aan de product backlog, het is echter aan de product owner en de developers om aan te geven of deze backlog-items bijdragen aan het doel van Search Traffic Monitor.



3.6 Afbakening

- Focus op sales module

De sales module is slechts een subonderdeel van Search Traffic Monitor. Er wordt enkel aan andere onderdelen gewerkt wanneer dit onvermijdelijk is, bijvoorbeeld door wijzigingen die gemaakt moeten worden in de structuur van de database.

- Framework buiten beschouwing

De nieuwe module wordt geschreven als uitbreiding op een het reeds gebruikte framework. Hiervoor geldt dat er enkel ontwerpen en testcases worden opgesteld betreffende onze extensies op dit framework.

- Bestaande onderdelen buiten beschouwing

Reeds bestaande onderdelen van Search Traffic Monitor worden buiten beschouwing gelaten, hier worden geen ontwerpen dan wel unittests voor geschreven.

3.7 Planning

Binnen het afstuderen zijn de onderstaande mijlpalen ingepland, de sprints worden laten door de product owner ingedeeld.

Mijlpaal	Datum
Start afstuderen	6 februari
<i>Plan van aanpak compleet</i>	8 februari
<i>Adviesrapport opgesteld</i>	15 maart
<i>Product backlog samengesteld</i>	16 maart
<i>Afronding sprint 1</i>	30 maart
<i>Afronding sprint 2</i>	13 april
<i>Afronding sprint 3</i>	27 april
<i>Afronding sprint 4</i>	11 mei
<i>Afronding sprint 5</i>	1 juni
Eind afstuderen	1 juni



4 Adviesrapport

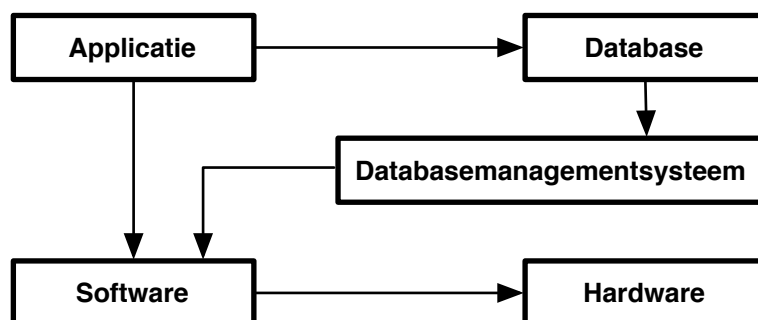
4.1 Werkwijze

Alvorens de performance van Search Traffic Monitor verbeterd kan worden moet eerst in kaart worden gebracht waarom de performance momenteel niet aan het gewenste niveau voldoet. Er is daarom besloten het adviesrapport op te stellen aan de hand van de vijf aspecten welke het meeste invloed hebben op de performance.

Binnen het adviesrapport worden de volgende aspecten onderzocht

- Applicatie
- Software
- Hardware
- Database
- Databasemanagementsysteem

Deze vijf aspecten zijn naar voren gekomen door te kijken waardoor de performance van de applicatie allemaal wordt beïnvloed. Hierbij is software een verzamelterm voor de resterende software op de server, denk hierbij aan Apache, PHP, en het operating system. Omdat Search Traffic Monitor op een geïsoleerde server draait hoeft er geen rekening gehouden te worden met processen afkomstig van andere applicaties, deze zijn namelijk afwezig.



Performance van de applicatie wordt beïnvloed door meerdere niveaus in het systeem

Een onderdeel welke we hier buiten beschouwing laten is het netwerk, hier is expliciet voor gekozen omdat wij geen invloed op het netwerk kunnen uitoefenen. Met name omdat de performance van het netwerk mede afhankelijk is van de internetaansluiting van de bezoeker kunnen we hier geen concrete uitspraken over geven.

Voor het detecteren en registreren van bottlenecks zijn er drie stappen gehanteerd. Voor elke gevonden bottleneck zijn er één of meerdere oplossingen voorgedragen opdat de product owner op basis van een inspanning/opbrengst analyse kan aangeven welke oplossingen het waard zijn om uiteindelijk te realiseren.



4.1.1 Reeds bekende bottlenecks

Aangezien ik de enige ontwikkelaar ben die de afgelopen anderhalf jaar verder heeft gewerkt aan Search Traffic Monitor ligt alle kennis van de applicatie bij mij. Op basis van mijn kennis zijn de reeds bekende bottlenecks geïdentificeerd als uitgangspunt voor het adviesrapport. Het betreffen hier voornamelijk bottlenecks die bekend zijn binnen de applicatie en de structuur van de database.

4.1.2 Systeempromfilering

Een andere methode voor het vinden van bottlenecks gebeurt aan de hand van systeempromfilering. Bij systeempromfilering wordt er een technisch profiel van een webpagina gevormd waarin er de laadtijd, de query's en het aantal requests¹ die benodigd zijn om de webpagina op te bouwen worden gemonitord. Hierbij is een expliciet onderscheid gemaakt tussen query's en requests omdat requests de werklust op de server verhogen terwijl query's de werklust op de database verhogen.

Voor het opbouwen van een systeemprofiel wordt er gebruik gemaakt van de systeempromfilering van het framework in combinatie met de console van Google Chrome. Wanneer er bepaalde elementen uitspringen binnen het profiel wordt er op de pagina ingezoomd om te analyseren hoe de pagina verbeterd kan worden. Vervolgens wordt er gekeken of dit probleem zich ook op andere pagina's voor doet. Indien het probleem zich voor doet op meerdere pagina's is het, met het oog op de gemiddelde performance, aantrekkelijker om het probleem te verhelpen.

¹ Individueel verzoek bij de server



Voorbeeld paginaprofiel

Pagina	/stm/diagnostic/dashboard
Laadtijd	
<i>Genereren</i>	110ms
<i>Structuur overgedragen</i>	240ms
<i>Query</i>	32ms
<i>Totaal</i>	440ms
Requests	41
<i>JavaScript</i>	12
<i>Styling</i>	3
Afbeeldingen	22
<i>API</i>	3
<i>Overig</i>	1
Query's	22
0.0008 DESCRIBE `mbr_member`	
0.00024 SELECT * FROM `mbr_member` WHERE member_id = 1	
ect.	

Het eerste wat hier opvalt is de hoeveelheid requests die benodigd zijn om de pagina te generen. Door de hoeveelheid requests te verminderen neemt de werklast op de server af en laad de pagina over het algemeen sneller. Wanneer we het aantal verzoeken kunnen terugdringen tot de helft betekent dit dat we met dezelfde server twee maal de hoeveelheid bezoekers aan kunnen ten opzichte van nu.

Om de hoeveelheid requests terug te dringen kan er gebruik worden gemaakt van een gestandaardiseerde oplossing die ook hier van toepassing is. Om het aantal JavaScript requests terug te dringen kunnen de bestanden samengevoegd worden tot één gecomprimeerd bestand. Voor de afbeeldingen geldt een soortgelijke oplossing, indien de onderlinge afbeeldingen niet met elkaar conflicteren qua bestandstype/doel op de webpagina kan er gekozen worden om deze te combineren tot een sprite sheet¹.

Een tweede punt wat opvalt in het bovenstaande verzoek is de hoeveelheid query's. Zoals reeds beschreven in 2.4 *Hoeveelheid query's* van het adviesrapport kan de hoeveelheid query's op basis van duplicatie worden terug gedrongen naar 15 query's, waarvan er 9 overbodig worden uitgevoerd door het framework zelf. Wanneer we de hoeveelheid query's terug kunnen dringen naar de 6 die we daadwerkelijk nodig hebben neemt de querytijd af naar een respectabele 5ms, ook hiervoor geldt dat we dan in het vervolg zes maal de hoeveelheid bezoekers aankunnen ten opzichte van nu.

¹ Afbeelding samengesteld uit meerdere afbeeldingen



4.1.3 Query analyse

Aan de hand van de systeemprofilering vormt er zich een complete lijst van alle query's die binnen de applicatie plaatsvinden. Binnen deze index specificeren we ons vervolgens tot de query's die gebruikt worden voor het genereren van de rapportages. De resterende query's gaan namelijk over een dergelijk kleine dataset dat de inspanning voor de optimalisatie niet opweegt tegen de performancewinst die het oplevert.

Voor elke resterende query wordt bij de database om een gedetailleerd profiel gevraagd. Aan de hand van dit profiel kan er vervolgens achterhaald worden waar de database haar tijd aan besteed bij het uitvoeren van de query, ofwel de bottleneck.

Voorbeeld queryprofiel

Totaal aantal rijen	39.043
Groote dataset	39Mb
Duratie	715.74 seconden

Duratie	Activiteit
0.000107	starting
0.000028	Opening tables
0.000003	System lock
0.000004	Table lock
0.000047	init
0.000013	optimizing
0.000385	statistics
0.000036	preparing
0.001565	Creating tmp table
1.377318	Sorting for group
0.000006	executing
714.347579	Copying to tmp table
0.008686	Sorting result

Nu is het bovenstaande query profiel geen uitzondering op de andere profielen, het databasemanagementsysteem geeft aan problemen te hebben met het kopiëren van data naar een tijdelijke tabel. De oplossing voor dit probleem kan op meerdere niveaus liggen, en voor elk niveau zijn weer meerdere oplossingen te bedenken.

1. Het probleem bevindt zich in de query

Nu we weten dat deze query problemen heeft met het aanmaken van een tijdelijke tabel, kan de query dan zo worden geoptimaliseerd dat het gebruik van de tijdelijke tabel wordt vermeden? Is het wellicht beter om deze query te splitsen in twee separate query's die in de code worden samengevoegd, of kan de query anders worden geformuleerd om vooralsnog eenzelfde resultaat te krijgen?



2. Het probleem bevind zich in de database

Een oplossing voor dit probleem kan zijn door gebruik te maken van aggregaties¹. We weten dat onze dataset groot is, waarom blijven we dan toch altijd de gehele dataset ondervragen als we ook een selectieve subset van de data kunnen ondervragen.

Aan de andere kant kan het zijn dat de gekozen tabelstructuur helemaal niet de juiste structuur is voor het gewenste doeleinde. Zo is er in dit geval gebruik gemaakt van een InnoDB structuur. Kenmerkend aan deze structuur is dat deze de primaire sleutel van de tabel, in ons geval een varchar van 52 karakters, aan elke index vast plakt. In het geval van de bovenstaande query hebben we te maken met zes geïndexeerde kolommen (domain, path, keywords, visit_id, date, redirect). Om de grote van de dataset te verkleinen kan er daarom gekozen worden om de primaire sleutel te verkleinen of om een andere tabelstructuur te gebruiken.

3. Het probleem bevind zich in het databasemanagementsysteem

De performance van het databasemanagementsysteem is afhankelijk van de resources die eraan worden vrijgegeven. Het feit dat het creëren van een tijdelijke tabel zo veel tijd nodig heeft kan duiden op een tekort in het toegewezen geheugen of de maximale toegestane grootte van tijdelijke tabellen.

4. Het probleem bevind zich in de server

Dat het kopiëren van data naar een tijdelijke tabel zo veel tijd nodig heeft kan wijzen op een tekortkoming in de I/O snelheid² van de harde schijf. Een ander probleem kan zijn dat de server te weinig RAM geheugen heeft waardoor deze in de eerste instantie geforceerd wordt gebruik te maken van de harde schijf.

Nu kan het probleem zich dus op vier plaatsen voor doen. Om erachter te komen welke oplossing(en) permanent het performanceprobleem oplossen moeten de gegeven oplossingen uitvoerig getest worden. Hiervoor is tijd en expertise nodig. Zo kan het foutief afstellen van je databasemanagementsysteem configuratie het crashen van de server tot gevolg hebben wanneer er meer geheugen gealloceerd wordt dan de server tot haar beschikking heeft.

¹ Een specifieke dataset, in termen van MySQL vergelijkbaar met een gecachde view

² Input/Output snelheid, snelheid waarmee data op de schijf kan worden gelezen en weggeschreven



4.2 Bevindingen

Data-integriteit

Gedurende het analyseren van de databasestructuur werd het duidelijk dat er een cruciale denkfout was gemaakt in het opzetten van de structuur. Deze fout had namelijk niet enkel betrekking op de grootte van de database maar ook de integriteit van de data. Omdat klanten en consultants naar deze data kijken was het dan ook noodzakelijk dat deze fout direct verholpen werd.

Het is redelijk gebruikelijk dat er een overlap ontstaat tussen twee logfiles, als oplossing hiervoor hadden we een primaire sleutel aan elke dataregel toegevoegd om te voorkomen dat deze data niet dubbel werd toegevoegd. Deze primaire sleutel wordt gevormd door een sleutel van de dataregel te combineren met het regelnummer. Door de toevoeging van het regelnummer zorgde de sleutel er voor dat een logfile niet tweemaal kon worden verwerkt maar dit loste nog niet het probleem van overlap op omdat hier de regelnummers niet overeen komen.

Er is daarom een nieuwe primaire sleutel ontworpen die volledig gebaseerd is op de dataregel. Hiermee hebben we de grootte van de primaire sleutel van 52 karakters weten te reduceren tot 14 karakters. Omdat in onze toenmalige datastructuur de primaire sleutel aan elke index wordt toegevoegd, twaalf in totaal, kromp de dataset van 5GB terug naar 3GB.

MongoDB

Het grootste performance probleem zit in de database, de datasets zijn te groot voor MySQL om te verwerken en uit te lezen. Na intensief testen met aggregaties, indexen en normalisatie¹ is er gekeken naar alternatieve databasemanagementsystemen. Omdat we in Search Traffic Monitor in verband met de snelheid zelf geen gebruik maken van vreemde sleutels, transacties en row locking zijn we binnen onze keuze niet beperkt tot relationele databases.

Eén van de alternatieven waarop we ons gefocust hebben is het non-relatieve databasemanagementsysteem MongoDB. Als test is de toenmalige grootste database gemigreerd waarna er soortgelijk query's op werden uitgevoerd. Waar MySQL minuten tot uren nodig had voor het uitvoeren van een dergelijke query was MongoDB binnen enkele seconden klaar. Op basis van de documentatie, de schaalbaarheid, de laagdrempeligheid en de bewezen performance is er daarom besloten om op MongoDB over te stappen.

Een bijkomend voordeel van MongoDB is dat deze op een veel andere manier query's uitvoert op een database. In plaats van een UDF te ontwikkelen voor het ophalen van een mediaan kunnen we nu een JavaScript functie ontwikkelen die de mediaan berekent. Hiermee verbeterd zowel de codecomplexiteit als de onderhoudbaarheid van de code. Zo bleek het uitbreiden van MySQL met een UDF geen gemakkelijke taak.

¹ De oorspronkelijke dataset is in verband met de performance gedenormaliseerd opgeslagen

Graphical User Interface

De interface van Search Traffic Monitor is momenteel geheel opgebouwd uit widgets. Dit zijn kleine rapporten die zich richten op één specifiek aspect uit de vergaarde data welke tezamen een compleet rapport vormen. Het voordeel van deze widgets is dat je ze binnen de applicatie overal kan hergebruiken waardoor je altijd toegang hebt tot je belangrijkste widgets.

Hoewel het gebruik van widgets in de eerste instantie een goed idee leek bleek in de praktijk dat de gebruiker er niet intuïtief mee om gaat. Dit omdat de widgets separaat van elkaar werken waardoor het soms onduidelijk is wat de widget exact toont. Daarnaast neemt de laadtijd van de pagina en de werkdruk op de server toe door de grote hoeveelheden widgets die per pagina ingeladen worden.



Naast de zichtbare widgets worden er ook nog widgets ingeladen onder de tab 'Websearch', verder worden er ook nog door de gebruiker opgeslagen widgets ingeladen.

De beoogde oplossing voor dit probleem is om de widgets samen te voegen tot zelfstandige rapporten. Hiermee neemt het aantal server requests af wat ten gunste komt voor de laadtijd van de pagina. Tevens kan er door het gebruik van rapporten gebruik worden gemaakt van aggregaties, waarmee de inhoud van de rapporten nog sneller geladen kan worden. In verband met de grote hoeveelheden widget varianten was dit voorheen niet mogelijk in verband met een gebrek aan serverruimte.



4.3 Conclusie

Alvorens het adviesrapport kon worden afgerond is op 15 maart 2012, op basis van bevindingen uit het adviesrapport, samen met de product owner besloten om een nieuwe versie van Search Traffic Monitor te ontwikkelen. De voorgestelde oplossingen vereisten dergelijk veel aanpassingen in de applicatie dat het verstandiger was om een nieuwe applicatie te bouwen. Hierbij worden de bevindingen uit het adviesrapport in acht genomen gedurende het opstellen van het ontwikkeltraject om herhaling van performanceproblemen te voorkomen.

Omdat er al concrete oplossingen waren gevormd voor het oplossen van de performanceproblemen van Search Traffic Monitor is er besloten om geen tijd meer te besteden aan het afronden van het adviesrapport. Het documenteren van deze bevindingen had voor de product owner in verhouding tot het ontwikkelen van de nieuwe applicatie geen toegevoegde waarde meer.



5 Requirements

5.1 Product backlog

Op 15 maart 2012 heeft de product owner, in het kader van de hoeveelheid aanpassingen benodigd voor het verbeteren van de performance, besloten dat het verstandiger is om een nieuwe applicatie te ontwikkelen. In plaats van nu een nieuwe sales module te ontwikkelen worden de resterende tien weken gebruikt voor het bouwen van een nieuwe applicatie. De product backlog wordt in de nieuwe situatie dan ook aangevuld op basis van een vernieuwde doelstelling.

Omdat Search Traffic Monitor te groot is om binnen tien weken volledig te herbouwen is er samen met de stakeholders een nieuwe doelstelling gedefinieerd. Hierbij is besloten om de belangrijkste rapporten 'na' te bouwen zodat de nieuwe versie vooralsnog inzichten vertoont betreffende crawling, ranking en traffic in het STRICA traject.

Rapportage

- Bezoekersrapport
- Zoektermrapport
- Zoekmachinerapport
- Filters

Management

- Beheren van accounts
- Beheren van verwerkingsproces
- Beheren van profielen
- Beheren van connectie met externe diensten
- Beheren van geautomatiseerde regels - *optioneel*

Uit het adviesrapport bleek dat de graphical user interface, ofwel layout, mede verantwoordelijk is voor de performanceproblemen binnen Search Traffic Monitor. Voordat de bovenstaande requirements kunnen worden uitgewerkt is het dan ook noodzakelijk dat er een nieuwe layout ontworpen wordt. Hiervoor worden meerdere stadia doorlopen om te testen of de layout en interactie met de applicatie conform de wensen en eisen van de product owner zijn voor er een permanente architectuur wordt opgezet.

Stadia

- Wireframes
- Mockup
- Technisch ontwerp
- Migratie
- Prototype



5.2 Rapportage

De rapporten zijn gebaseerd op de huidige Search Traffic Monitor, hiervan zijn de meest waardevolle rapportages meegenomen om zo veel mogelijk inzicht te verschaffen in het STRICA traject. Middels het gebruik van filters is de gebruiker vervolgens in staat om bepaalde varianten van de rapportages te genereren.

Bezoekersrapport

Het bezoekersrapport toont hoeveel bezoekers er per pad binnen zijn gekomen. Zodra de gebruiker op een pad klikt gaat deze naar een detailrapport waarop alle zoektermen waarmee de bezoekers op dat pad binnen zijn gekomen worden getoond.

Zoektermrapport

Het zoektermrapport toont de regelmaat waarmee bezoekers binnen komen via een bepaalde zoekterm. Indien de data bekend is wordt in dit rapport ook de gemiddelde positie getoond waarmee de bezoeker via de zoekterm binnen komt.

Zoekmachinerapport

Het zoekmachine rapport is het primaire rapport dat gebruikt wordt om inzicht te verschaffen op het gebied van zoekmachinegedrag. Dit rapport toont de hoeveelheid en de regelmaat waarmee de zoekmachines een bepaald pad crawlen.

Filters

Filters worden gebruikt om de inhoud van de rapportages te beïnvloeden. Met behulp van de filters kan een rapport vanuit meerdere perspectieven worden bekeken opdat deze een multifunctioneel doel krijgen. Zo is het bijvoorbeeld mogelijk om een bezoekersrapport rapport te filteren op een verkeersbron. Aan de hand van deze filter kan het zoekverkeer vanuit Google vergeleken worden met het zoekverkeer vanuit Yahoo, wat waardevolle inzichten kan geven over de types bezoekers die binnen komen via de verschillende zoekmachines.

Bij het bouwen van de filters beperken we ons voor nu tot de filters die ook in de huidige Search Traffic Monitor worden aangeboden. Daarnaast implementeren we ook een drempelwaarde filter die alle rijen onder de gedefinieerde drempel verbergt, dit om minder interessante data uit de rapporten weg te filteren.

Filters

- Datum
- Pad
- Statuscode
- Zoekterm
- Zoekmachine
- Verkeersbron
- Drempelwaarde



5.3 Management

Beheren van accounts

Search Traffic Monitor is een applicatie die veel klantgevoelige data bevat, het is dan ook van belang dat de applicatie goed beveiligd wordt en dat klanten enkel hun eigen rapportages kunnen inzien. Daarom wordt er gebruik gemaakt van een ledensysteem waarmee de data afgeschermd wordt.

Binnen het ledensysteem moet er rekening gehouden worden met verschillende rollen. Een klant mag enkel haar eigen data inzien in tegenstelling tot OrangeValley consultants die toegang hebben tot alle data.

Beheren van verwerkingsproces

Search Traffic Monitor verzamelt haar data primair middels logfileanalyses. Deze logfileanalyses vinden plaats op een ander onderdeel en een andere server(s) binnen de architectuur, om dit proces te overzien is een inzicht in het verwerkingsproces wenselijk.

In dit overzicht specificeren we ons tot twee onderdelen units en de taken. Units zijn subonderdelen in het verwerkingsproces, dit zijn stukken software. Taken zijn feitelijk de logregels over de stukken data die door de units worden verwerkt.

Taken schetsen een historisch profiel over de performance van een unit, omdat deze logregels representeren is het niet wenselijk dat taken aangepast/toegevoegd of verwijderd worden via het beheer. Het beheer specificeert zich dan ook enkel tot een lijst weergave waarin alle taken worden weergegeven.

De units moeten echter volledig beheerd kunnen worden via het management. Men moet in staat zijn om te zien hoeveel taken actief zijn op de specifieke unit, indien dit aantal te hoog is moet de gebruiker in staat zijn om nieuwe units toe te voegen via het beheer. Dit wil zeggen dat er een connectie naar een externe server moet worden opgezet waarop de installatie van een nieuwe unit plaats vindt. De aanschaf en configuratie van een nieuwe server wordt hierbij buiten beschouwing gelaten, dit is in verband met de gehanteerde connectiemethodiek niet mogelijk. Naast het installeren moet het ook mogelijk zijn om units te herinstalleren en deïnstalleren.

Beheren van profielen

Elk profiel representeert een website waarop Search Traffic Monitor actief is. Het wil voorkomen dat een klant meerdere websites beheert, daarom moet het mogelijk zijn om meerdere profielen aan een gebruiker te koppelen. Middels het profielenbeheer worden de profielen volledig geconfigureerd.

Aan de hand van het profiel beheert de gebruiker welke databaseconnectie er gebruikt moet worden, wat voor tracking type er wordt gebruikt, hoe de data moet worden opgeschoond en aan welke externe diensten het profiel gekoppeld is. In dit beheer moet echter een onderscheid worden gemaakt tussen de verschillende rollen, zo mag een gebruiker enkel haar externe diensten aanpassen. Het recht voor het aanmaken van nieuwe profielen is daarmee voorbehouden tot de OrangeValley consultants.



Beheren van connectie met externe diensten

Search Traffic Monitor is een applicatie die graag van externe diensten gebruik maakt om nog meer inzichten te kunnen genereren. Zo kon Search Traffic Monitor in het verleden een koppeling leggen met diensten als Google Analytics, Google Webmaster, Google AdWords, Pingdom en SiteSpect. In verband met het korte tijdsbestek beperken we dit tot slechts twee diensten benodigd voor het downloaden van logfiles bij klanten, zijnde SiteSpect en een nieuwe generieke FTP verbinding.

Diensten beperken zich tot de primaire inlog van een externe dienst, het profiel gaat hier dieper op in door een profiel/account binnen de dienst te selecteren. Zo wordt er in het profiel een Google Analytics dienst geselecteerd waarna de accounts en profielen van de geselecteerde dienst worden getoond. Het gemak hiervan is dat de gebruiker de inlog van al haar connecties vanuit één punt kan beheren en deze niet overal opnieuw in hoeft te vullen.

Waar mogelijk wordt er gebruik gemaakt van een OAuth connectie in verband met de veiligheid. Normaliter gebruikt men een login en wachtwoord om in te loggen bij een externe dienst, OAuth verschilt hierin door gebruik te maken van een gegeneerde token¹. Het voordeel van een token is dat ook werkt nadat de gebruiker haar login of wachtwoord veranderd, hiermee is de connectie aanzienlijk stabiel. Tevens geeft het gebruik van tokens het voordeel dat de gebruiker haar wachtwoord niet aan ons hoeft te verstrekken. Indien een token uitlekt heeft de hacker hier niets aan, een token is namelijk enkel geldig vanaf ons domein in tegenstelling tot een login/wachtwoord combinatie.

¹ Domein gebonden unieke sleutel die de login van een gebruiker representeert



Beheren van geautomatiseerde regels – *optioneel*

Search Traffic Monitor is een applicatie die uiteindelijk ook de gebruiker moet gaan sturen richting de adviezen en zelf ook acties kan ondernemen. Binnen Search Traffic Monitor doen we dat middels geautomatiseerde regels, de applicatie erkent enkele evenementen en de gebruiker heeft vervolgens de mogelijkheid om daar acties aan te koppelen. Een voorbeeld hiervan is de stopknop, deze regel pauzeert automatisch een AdWords campagne zodra een website downtime ondervindt, zodra de website weer bereikbaar is wordt de campagne automatisch weer geactiveerd.

Het maken van geautomatiseerde regels is vrij complex, de gebruiker moet namelijk op een intuïtieve en gebruikersvriendelijke manier in staat zijn om nieuwe regels te vormen. Hiervoor is een dynamisch formulier vereist waarbij nieuwe evenementen en acties naar gelang aan de regel worden toegevoegd.

Mocht er genoeg tijd overblijven in het afstudeertraject dan wordt er een opzet gemaakt voor het samenstellen van de regels, hierin beperken we ons tot acties en evenementen die bijdragen aan het geautomatiseerd binnen halen van de logfiles. Na het afstudeertraject kunnen de acties en evenementen door de product owner naar gelang worden aangevuld. Denk hierbij aan de stopknop, trendanalyses en het automatisch investeren in AdWords campagnes.

Evenementen

- Specifieke datum
- Globale datum

Acties

- SiteSpect logfile downloaden
- FTP logfile downloaden

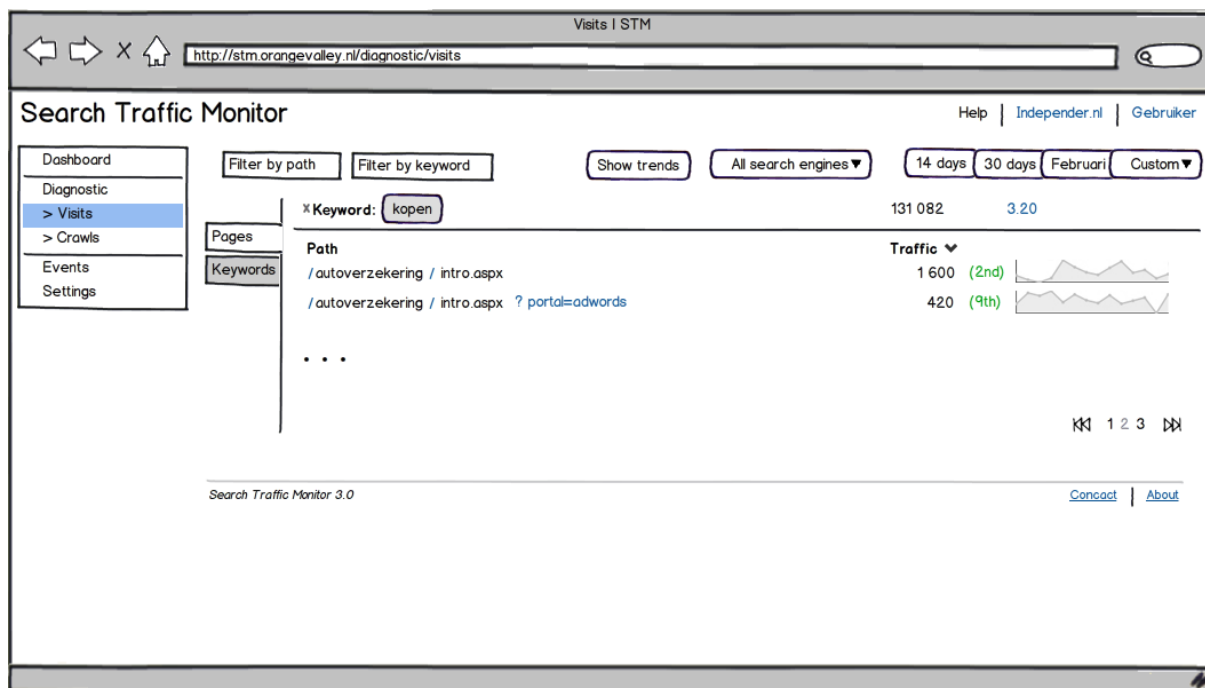


6 Architectuur

6.1 Wireframes

Wireframes zijn ruwe schetsen welke gebruikt worden als opzet voor de nieuwe interface. In dit stadium wordt de focus gelegd op de indeling van de website en de manier waarop de data wordt gerepresenteerd, de 'looks' van de applicatie worden in dit stadia nog achterwege gelaten. Om een compleet beeld van de uiteindelijke applicatie te vormen is het gebruikelijk dat van elk schermtype een wireframe wordt opgesteld.

De wireframes zijn ontworpen op basis van de vooraf gedefinieerde requirements, hierbij is rekening gehouden met de bevindingen uit het adviesrapport opdat fouten gemaakt in de originele Search Traffic Monitor hier niet in terug komen. Een groot verschil hierbij is dat de widgets plaats hebben gemaakt voor een rapport. Hierbij staan de filters altijd in beeld zodat het voor de gebruiker duidelijk is waar zij op dat moment naar kijkt.



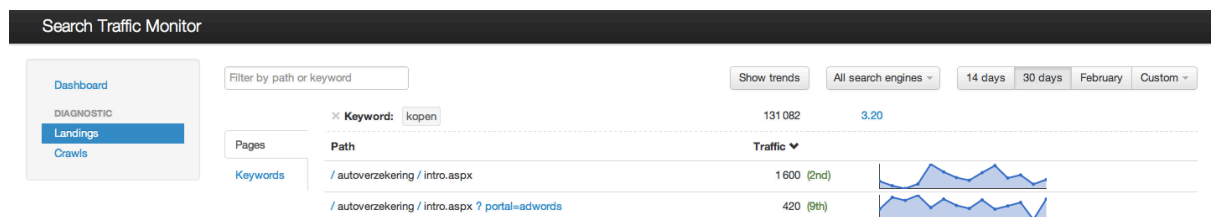
Wireframe voor een bezoekersrapport



6.2 Mockup

De wireframes zijn na afronding aan de product owner voorgedragen opdat de interface en interactie met de applicatie theoretisch¹ kon worden getest. Op basis van zijn akkoord zijn de wireframes vervolgens uitgewerkt tot een mockup. Waar wireframes zich beperken tot een grove schets is een mockup een web versie die ook een beeld geeft van de uiteindelijke layout.

Om de mockup snel te kunnen bouwen wordt de pagina gevuld met dummy data en wordt functionaliteit achterwege gelaten. De mockup is namelijk niet meer dan een opvolgende gebruikerstest om af te stemmen met de product owner of hij content is met de layout van de applicatie.



Mockup voor een bezoekersrapport

Na akkoord van de product owner kan er begonnen worden aan de bouw van een prototype waarmee een beeld wordt geschetst van de uiteindelijke applicatie. Dit wordt gedaan door de mockup uit te breiden met functionaliteiten en de dummy data vervangen wordt door echte data. Alvorens het prototype gebouwd kan worden is er eerst een technisch ontwerp en een migratie vereist.

¹ Er is verbaal uitgelegd hoe de interactie met de applicatie eruit zou komen te zien

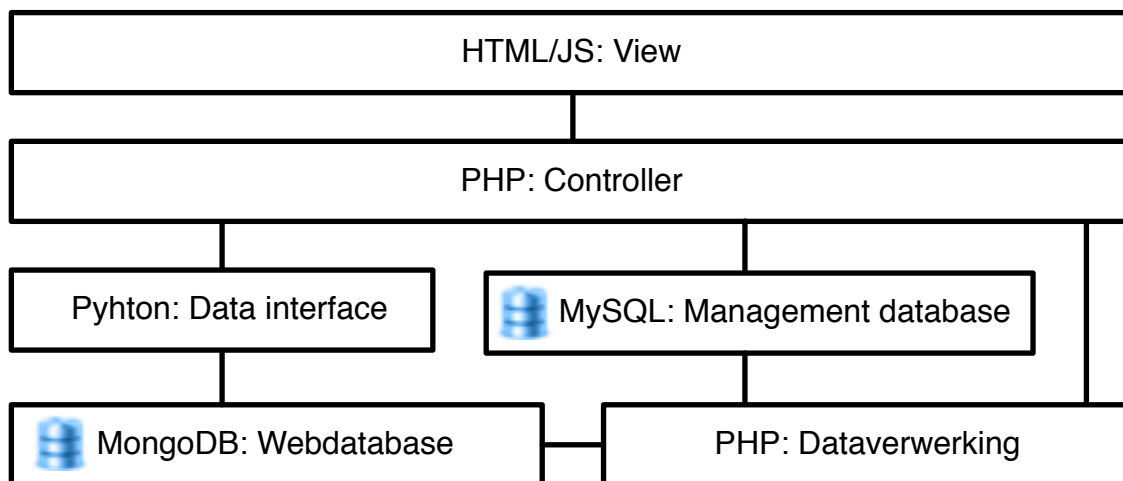


6.3 Technisch ontwerp

6.3.1 Architectuur

Search Traffic Monitor is een applicatie die wordt ontwikkeld en beheert door 42functions, het is dan ook een vereiste dat de architectuur wordt opgezet conform de bedrijfsstandaard. Uitzondering hierop is het gebruik van MongoDB waarvoor het management uitdrukkelijk toestemming heeft gegeven, voor de andere onderdelen geldt dat er gebruik wordt gemaakt van MySQL en PHP.

De architectuur van Search Traffic Monitor is primair opgezet met het oog op performance en schaalbaarheid. Zoals reeds vermeld in 3.1 *Search Traffic Monitor* wordt een groot deel van de data verzameld aan de hand van logfileanalyse. Een logfile beslaat vaak enkele gigabytes aan data welke eerst vanaf de server van de klant moet worden gedownload waarna deze geanalyseerd en verwerkt wordt. Dit is een intensief proces waar de rest van de server hinder aan ondervindt, zo is er in het verleden besloten om de logfileanalyse vanaf een aparte server te draaien.



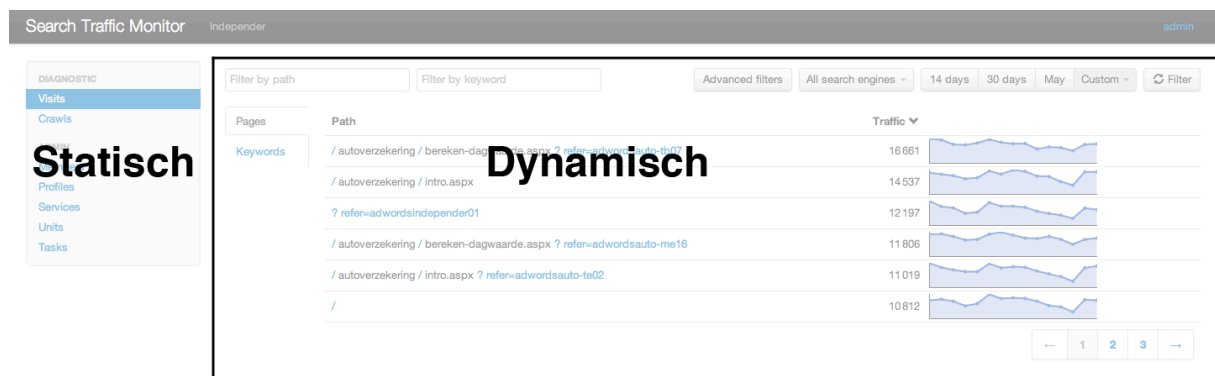
Architecturale opbouw van Search Traffic Monitor



6.3.2 HTML/JS: View

De architectuur binnen het framework dwingt ons gebruik te maken van een MVC¹ model, het gevolg hiervan is dat er een aparte laag binnen de architectuur wordt opgenomen die verantwoordelijk is voor de interactie en visuele weergave van de applicatie. Hierbij speelt vooral JavaScript een grote rol betreffende de interactie met rapportages en filters. Door de filters en rapportages volledig te beheren met behulp van JavaScript wordt de rekenkracht van client benut, iets waar normaliter één van onze servers voor gebruikt moest worden.

Een ander voordeel van een afgescheiden view is dat het een grote stap is richting het HTML5 pushstate principe. Het idee achter het HTML5 pushstate principe is dat de gebruiker eenmalig alle resources inlaad en vervolgens specifieke onderdelen binnen de website herlaad. Dit vermindert het aantal server requests aanzienlijk wat ten gunste is voor zowel de gebruiker als de server, denk hierbij aan dataoverdracht en laadtijd van de pagina.



HTML5 pushstate principe

6.3.3 PHP: Controller

De controller laag is verantwoordelijk voor het aansturen van de view en het applicatiebeheer. Tevens wordt de controller gebruikt voor het uitvoeren van taken op tijdbasis, denk hierbij aan geautomatiseerde regels die de gebruikers dagelijks informeren over afwijkingen binnen de dagelijkse trendanalyse.

Een andere taak van de controller is dat deze als doorgeefluik tussen de view en data interface fungeert waarin de filters en aanvraag voor de rapportage worden gevalideerd. Dit vindt plaats in de controller omdat deze de loginsessie van de gebruiker beheert, tevens is het onveilig om gebruikersdata zomaar door te sturen naar de data interface. Naarmate we ons dieper in de architectuur bevinden wordt er aangenomen dat de aangeleverde data valide is, dit omdat het inefficiënt is om de data constant te blijven hervalideren.

Omdat de controller laag wordt aangestuurd met behulp van een framework moet de controller conform de architectuur van het framework worden opgebouwd. Hierin wordt gebruik gemaakt van de klassen aangeboden vanuit het framework om de gedefinieerde requirements zo snel mogelijk te kunnen realiseren.

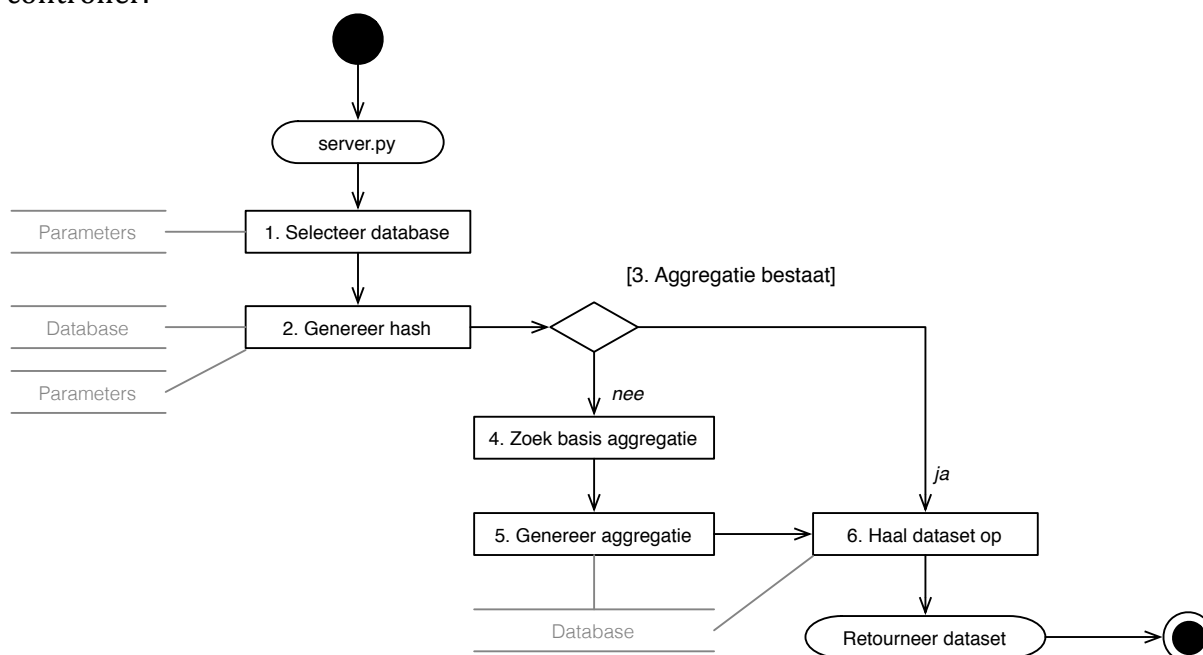
¹ Model View Controller



6.3.4 Python: Data interface

De data interface is verantwoordelijk voor het beheer over de aggregaties in de webdatabase. Om de rapporten snel te kunnen genereren wordt er in de nieuwe architectuur voor elk rapport een aggregatie in de webdatabase aangemaakt. Het is de taak van de data interface om de vraag vanuit de controller te converteren naar een valide aggregatie.

Allereerst wordt de aanvraag geconverteerd naar een sleutel die de naam van de aggregatie representeert. Indien er nog geen aggregatie bestaat wordt er op basis van de aanvraag een JavaScript functie gegenereerd de aggregatie in de MongoDB database aanmaakt. Uiteindelijk wordt de aggregatie uitgelezen en geretourneerd naar de controller.



Flowchart voor het aanmaken van een aggregatie, zie **Error! Reference source not found.** Technisch ontwerp voor beschrijvingen.

Opvallend aan deze laag is dat er gebruik wordt gemaakt van Python, een taal welke niet onder de bedrijfsstandaard van 42function valt. In verband met de druk achter het project is de bouw van deze laag uitbesteed aan een externe partij, hierbij heeft het management toegekend dat er gebruik mag worden gemaakt van Python indien dit het ontwikkeltraject versnelt.

Omdat er weinig tot geen Python kennis aanwezig is bij 42functions wordt deze laag op later termijn herschreven naar PHP. Dit ten behoeve van de overdracht van Search Traffic Monitor en gemak in het serveronderhoud, het is nu namelijk vereist om zowel Python als PHP op de server te onderhouden.

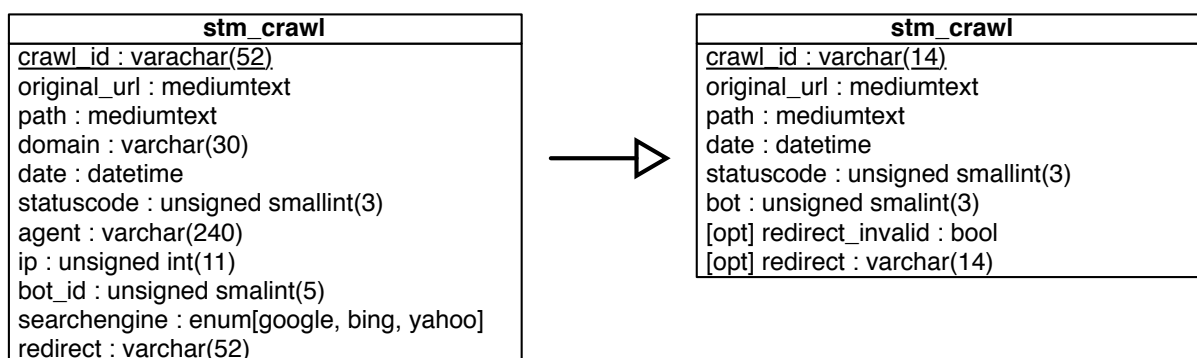


6.3.5 MongoDB: Webdatabase

De webdatabase wordt gebruikt om site specifieke data in te beheeren. Het betreft hier voornamelijk data rondom zoekmachinegedrag, paginaeigenschappen, annotaties¹ en bezoekersgedrag. Zoals reeds beschreven in 4.2 *Bevindingen* is er in verband met de performance en de grote van de datasets gekozen om hiervoor gebruik te maken van MongoDB.

Om het databasemanagementsysteem overzichtelijk te houden wordt er voor elke website gekoppeld aan Search Traffic Monitor een eigen database aangemaakt. Het voordeel hiervan is dat er geen identificatiekolom aan de tabellen hoeft worden toegevoegd waarmee we normaliter de websites van elkaar onderscheiden. Door het weghalen van deze extra kolom wordt de dataset kleiner en worden aggregaties sneller aangemaakt omdat er niet gefilterd hoeft te worden op deze extra kolom.

De structuur van de webdatabase is gebaseerd op de oude databasestructuur waarbij de bevindingen uit het adviesrapport in acht zijn genomen. Hierdoor is de structuur van de database efficiënter ingericht waardoor de dataset minder ruimte in beslag neemt.



Optimalisatie van zoekmachine tabel

In de bovenstaande afbeelding is als voorbeeld de optimalisatie van de zoekmachine tabel te zien. Door ongebruikte data te verwijderen, de primaire sleutel te verkorten en het bot_id met het searchengine kolom samen te voegen kunnen we de data veel efficiënter opslaan. Tevens maakt het gebruik van een non-relatieve database het mogelijk om 'kolommen' uit een de datarij weg te laten om de dataset zo klein mogelijk te houden een voorbeeld hiervan is de redirect² kolom, welke enkel opgeslagen wordt indien er sprake is van een redirect.

¹ Vakterm voor notities

² Automatisch doorverwijzen van de bezoeker naar een andere pagina



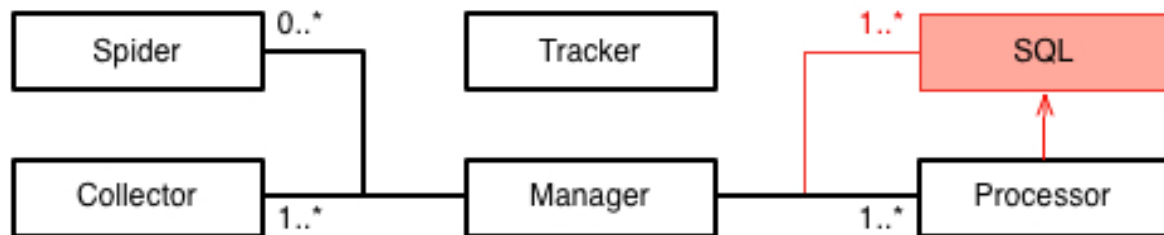
6.3.6 MySQL: Management database

De management database wordt gebruikt om de data die gedeeld wordt over de hele applicatie te beheren. Het betreft hier met name entiteiten als gebruikers, profielen, regels, taken, units, services e.d. die zelf geen data over de website bevatten.

Er is hier expliciet voor een MySQL oplossing gekozen om de applicatie laagdrempelig te houden, de hoeveelheid data in deze laag is dergelijk klein dat hier geen soortgelijke performanceproblemen optreden. Het is aannemelijker dat de server eerder met netwerkproblemen te maken heeft in verband met de hoeveelheid gebruikers, wat opgelost wordt door een tweede controller¹ aan te schaffen.

6.3.7 PHP: Dataverwerking

De dataverwerking is een reeds bestaand onderdeel binnen Search Traffic Monitor welke gebruikt wordt voor het vergaren en verwerken van de data. Omdat er gebruik gemaakt wordt van een nieuw databasemanagementsysteem is het noodzakelijk dat deze laag aangepast wordt om in het vervolg nog correct te werken, hierbij wordt tevens een performance optimalisatieslag uitgevoerd om het verwerken van de logfiles te versnellen.



Modulaire opbouw van dataverwerking

Gedurende de optimalisatieslag is er besloten om de SQL unit² te combineren met de processor-unit. Het oorspronkelijke doel van de SQL-unit was dat deze een de SQL-batches gegenereerd door de processor in één keer naar de database stuurde om de datastroom te kunnen beheren. In de praktijk bleek dit echter geen meerwaarde te bieden.

Door deze twee units te combineren hoeft er geen dataoverdracht meer plaats te vinden tussen de twee units, wat de totale verwerkingstijd aanzienlijk verkort. Tevens zorgt dit voor een afname in het code- en serveronderhoud omdat de SQL unit niet meer als aparte unit op een eigen server onderhouden hoeft te worden.

Deze laag is dergelijk opgezet dat de manager als loadbalancer fungeert welke de logfiles uit de collector onderverdeeld aan de processoren en spiders. Hierbij moet rekening worden gehouden met het feit dat elke unit op een eigen server kan staan. Middels taakmanagement worden nieuwe taken toegewezen aan de unit die op dat moment de minste werklast heeft.

¹ Conform onze serverarchitectuur krijgt deze haar eigen database en dataverwerking

² Module binnen de dataverwerking



6.4 Migratie

Alvorens er een prototype van Search Traffic Monitor gebouwd kan worden moet de data vanuit de oude applicatie naar de nieuwe applicatie worden gemigreerd. Zoals reeds beschreven zijn er twee type databases binnen Search Traffic Monitor. Omdat niet alle functionaliteit in de nieuwe Search Traffic Monitor aanwezig zijn en de management database vrijwel geen data bevat wordt enkel de webdatabase gemigreerd.

Na het uitvoeren van de migratie is de dataset in de nieuwe applicatie vergeleken met de dataset uit de oude applicatie. Hiervoor zijn enkele rapportages in de nieuwe applicatie nagebootst waarna deze vergeleken werden met de rapportages uit de oude applicatie.

Op basis van het trackingtype dat gebruikt is voor het verzamelen van de dataset wordt er een andere aanpak gehanteerd.

Logfiles

Bij het uitvoeren van de logfileanalyses wordt er van alle logfiles automatisch een backup gemaakt. Omdat aan een migratie van dergelijke profielen vaak een database van meerdere gigabytes gepaard gaat is het moeilijk om een export van de oude database te maken die we vervolgens zonder problemen kunnen importeren. Het is daarom eenvoudiger om alle logfiles opnieuw te laten verwerken. Dit garandeert dat alle data gelijk in het juiste formaat staat en het voorkomt dat er fouten gemaakt worden bij het opstellen van een migratieplan. Verder is dit gelijk een stres-test om te onderzoeken of het aangepaste verwerkingstraject, zie ook 6.3 *Technisch ontwerp*, naar behoren functioneert.

Trackingpixel

Voor het trackingpixel mechanisme worden er geen logfiles gegenereerd dan wel bijgehouden, de enige data die vanaf dit mechanisme bijgehouden wordt bevindt zich in de database. Er is dan ook geen andere oplossing dan gebruik te maken van een migratieplan.

Omdat de data gemigreerd wordt naar een ander type databasemanagementsysteem waarbij het tevens het datamodel gewijzigd moet er gebruik worden gemaakt van een migratiescript. Dit migratiescript haalt een rij uit de oude applicatie op, converteert de inhoud van de rij naar het nieuwe datamodel en voegt de rij tenslotte toe aan de nieuwe applicatie. Voor het in- en uitlezen van de data wordt er gebruik gemaakt van dezelfde modules die worden gebruikt in het verwerkingstraject en de controller module.

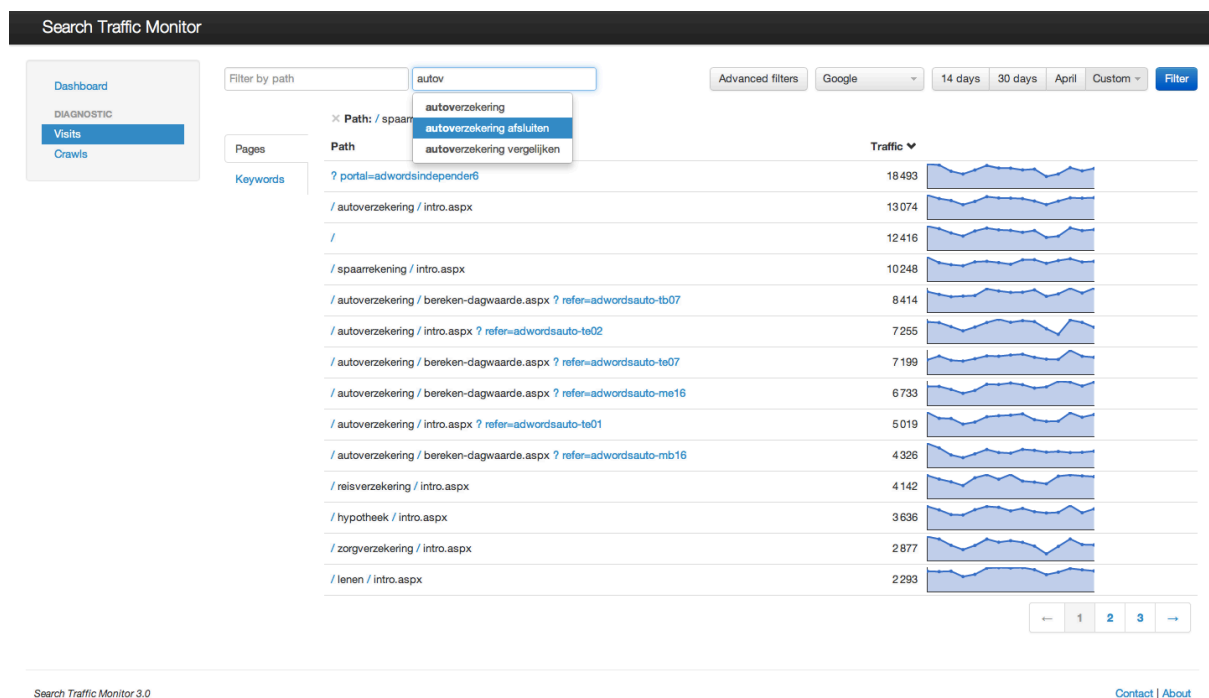


6.5 Prototype

De laatste stap die ondernomen wordt voor het bouwen van de nieuwe applicatie is het maken van een prototype. Dit prototype is een laatste gebruikerstest die gebruikt wordt om de werking van de uiteindelijke applicatie te ervaren. Bij het testen van het prototype wordt er gekeken naar de snelheid, de gebruikersinteractie en technische complicaties. Aan de hand van deze factoren kan het technisch ontwerp nog worden aangepast opdat er voor de nieuwe applicatie een solide basis wordt gevormd.

Het prototype is een tijdelijk tussenproduct, het is dan ook de bedoeling dat hier zo min mogelijk tijd aan wordt besteed. Daarom beperken we ons bij de bouw van het prototype enkel tot het belangrijkste onderdeel van de applicatie, namelijk de rapportage. Alle omliggende onderdelen die benodigd zijn voor het genereren van de rapportage worden op een statische manier in elkaar gezet.

Omdat de testresultaten van het prototype zoveel invloed hebben op het uiteindelijke product zijn zowel de product owner als de doelgroep gevraagd om het prototype door te testen. Gedurende deze test zijn er gelijktijdig meerdere gebruikers het prototype aan het testen waardoor er gelijk een concurrency-test plaats vond. Gedurende deze concurrency-test is er door de developers gemonitord hoe gelijktijdig gebruik van de applicatie de performance beïnvloed.



Prototype voor bezoekersrapport



7 Realisatie

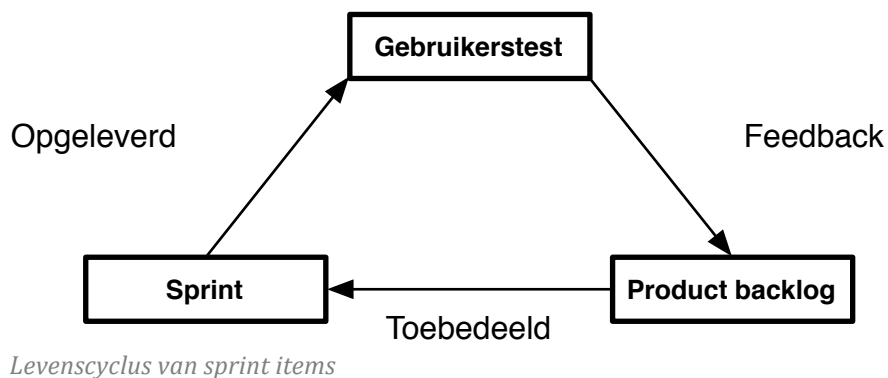
7.1 Testen

7.1.1 Productietest

Voordat we begonnen zijn aan de bouw van de applicatie zijn er meerdere stadia doorlopen om te testen of de product owner content is met de layout in interactie met de nieuwe applicatie, zie ook 5.1 *Product Backlog*. Na afronding van elk stadium is de product owner om feedback gevraagd opdat, indien nodig, het ontwerp tijdig kon worden aangepast.

Gedurende het doorlopen van de stadia is er tijdens de oplevering van het prototype een concurrency-test uitgevoerd gevonden waarbij meerdere gebruikers uit de doelgroep gelijktijdig de applicatie hebben gebruikt. Door gelijktijdig meerdere gebruikers met het prototype te laten werken kon vroegtijdig de performance van de uiteindelijke applicatie worden ervaren.

Door gebruik te maken van een iteratief ontwikkeltraject, Scrum, zijn oneffenheden in de applicatie, resulterend uit gebruikerstesten, aan de product backlog toegevoegd. Deze sprint items kunnen vervolgens aan een sprint worden toebedeeld waardoor ze net als andere functionele eisen het afgesproken ontwikkeltraject volgen. Het voordeel hierbij is dat niet elk sprint item genoeg prioriteit heeft dat het gelijk opgelost moet worden, dergelijke bugs kunnen hierdoor dus ook uitgesteld worden naar een latere sprint.





7.1.2 Unittest

7.1.2.1 Aanpak

Search Traffic Monitor is een applicatie die de OrangeValley consultants dagelijks moet gaan ondersteunen in haar werk. Het is dan ook uiterst belangrijk dat data die in de rapporten getoond wordt geen fouten bevat. Omdat het vanuit een tijd technisch perspectief niet reëel is om per release de hele applicatie door te testen is er daarom gebruik gemaakt van test-driven development.

Test-driven development wil zeggen dat alvorens er een requirement in de code wordt geïmplementeerd er eerst unittests worden aangemaakt. Deze unittests controleren of de code-implementatie functioneert zoals verwacht, belangrijk hierbij is dat bij het opzetten van de testcases rekening wordt gehouden met zowel valide als invalide scenario's.

Na het opzetten van de unittests worden de functionaliteiten vervolgens één voor één geïmplementeerd. Aan de hand van het aantal slagende unittests kan men de progressie van de bouwfase monitoren, wanneer alle testcases slagen is de beoogde functionaliteit succesvol geïmplementeerd.

Unit test

PHP: 2 failed, 280 passed

Test	install resource/utest/module/stm/models/Unit.class.php(143) reinstall resource/utest/module/stm/models/Unit.class.php(165)	Passed	Execution time
STM_Unit	2	34	0.363
STM_Profile	0	35	0.030
STM_ProfileForm	0	58	0.068
STM_ProfileTable	0	22	0.015
STM_Service	0	21	0.006
STM_ServiceForm	0	22	0.012

Door middel van unittests wordt in één keer de gehele applicatie getest.

De bovenstaande afbeelding toont ons de output van de unittests, hierin is te zien dat twee testcases binnen STM_Unit unittest falen. Dit rapport geef daarmee aan dat de corresponderende klasse, STM_Unit nog functionaliteiten mist of dat deze niet naar behoren zijn geïmplementeerd.

Om er zeker van te zijn dat de applicatie niet vervuild raakt door de unittests vindt het testen plaats op een aparte omgeving. Deze omgeving is waar mogelijk een exacte kopie van de productieomgeving om zo onverwachte fouten te voorkomen. De testomgeving bevat een eigen database waarnaar in de testcases kan worden gerefereerd.

Bij het opzetten van de unittests worden de functionaliteit, afhankelijk van het doeleinde, vanuit vier verschillende aspecten benaderd. Gezamenlijk vormen deze aspecten een compleet beeld of de applicatie naar behoren functioneert.

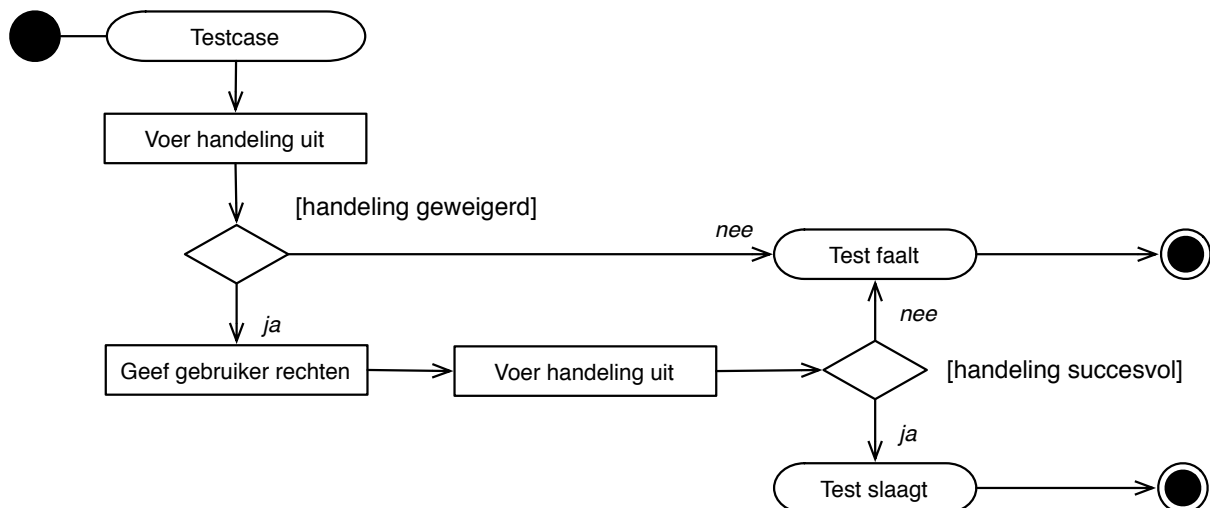


Veiligheid

Omdat Search Traffic Monitor veel klantgevoelige data bevat wordt er middels unittesten aandacht besteed aan het testen van de veiligheid van de applicatie. Deze testen richten zich primair op de veiligheid binnen de applicatie, netwerktesten worden hier bijvoorbeeld achterwege gelaten.

Niet elke gebruiker is gemachtigd tot het uitvoeren van bepaalde handelingen. Er wordt daarom getest of de gebruiker toegang heeft tot bepaalde punten in de applicatie, of zij gemachtigd is tot het uitvoeren van handelingen en met name of deze handelingen dan al niet in de interface voor haar verborgen worden.

Voor het testen van de handeling worden er minimaal twee test cases opgezet. In de eerste testcase wordt getest of de testgebruiker zonder rechten gemachtigd is tot het uitvoeren van de handeling, waarbij een foutmelding wordt verwacht. Vervolgens worden de benodigde rechten aan de gebruiker toegekend waarna de testcase wordt herhaald, waarbij de testgebruiker dit maal wel gemachtigd is tot het uitvoeren van de handeling.

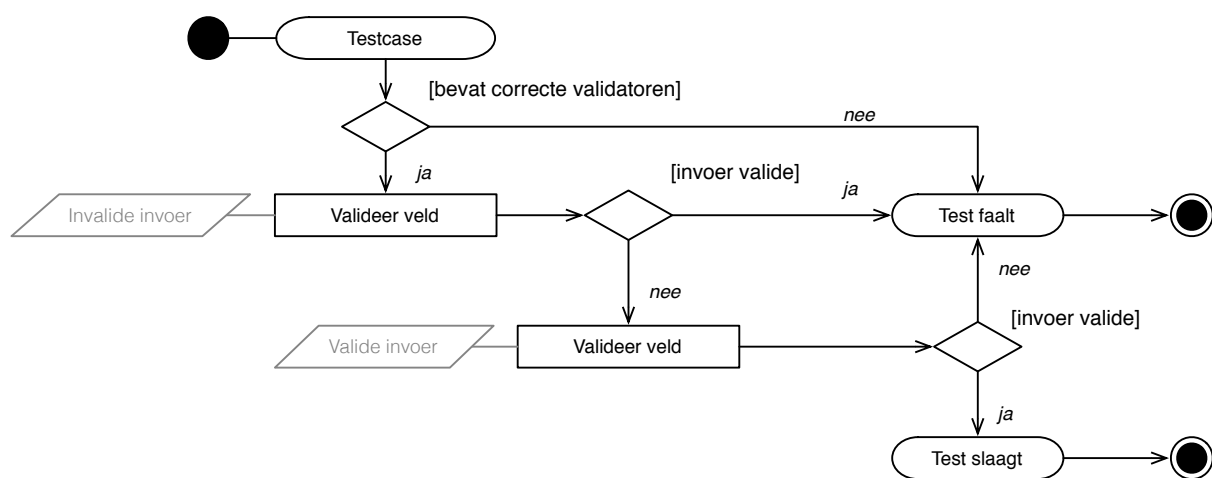


Testcase voor uitvoeren van handeling



Naast het testen van handelingen testen we ook op de gebruikersinvoer. Naarmate we ons dieper in de architectuur bevinden verwacht de applicatie dat de aangeleverde data valide is. Het is daarom belangrijk dat de gebruikersinvoer in een zo vroeg mogelijk stadium wordt gevalideerd, dit geldt met name voor formulieren.

Het valideren van de invoer gebeurt primair met behulp van het framework, omdat hier de aanname voor kan worden gemaakt dat deze altijd naar behoren functioneert worden er minimaal drie testcases per invoerveld opgezet. In de eerste testcase wordt er gevalideerd of de verwachte validatoren aan het invoerveld zijn gekoppeld, voorbeelden van validatoren zijn: verplichte invoer, emailvalidatie, numerieke waarde, minimale lengte, ect. In de tweede testcase wordt het veld getest op invalide invoer om te testen of de validatie niet door het formulier zelf overschreden wordt en de validatie faalt zoals verwacht. Ten slotte wordt het veld getest met valide invoer waarbij getest wordt op het slagen van de test.



Testcase voor het valideren van invoer



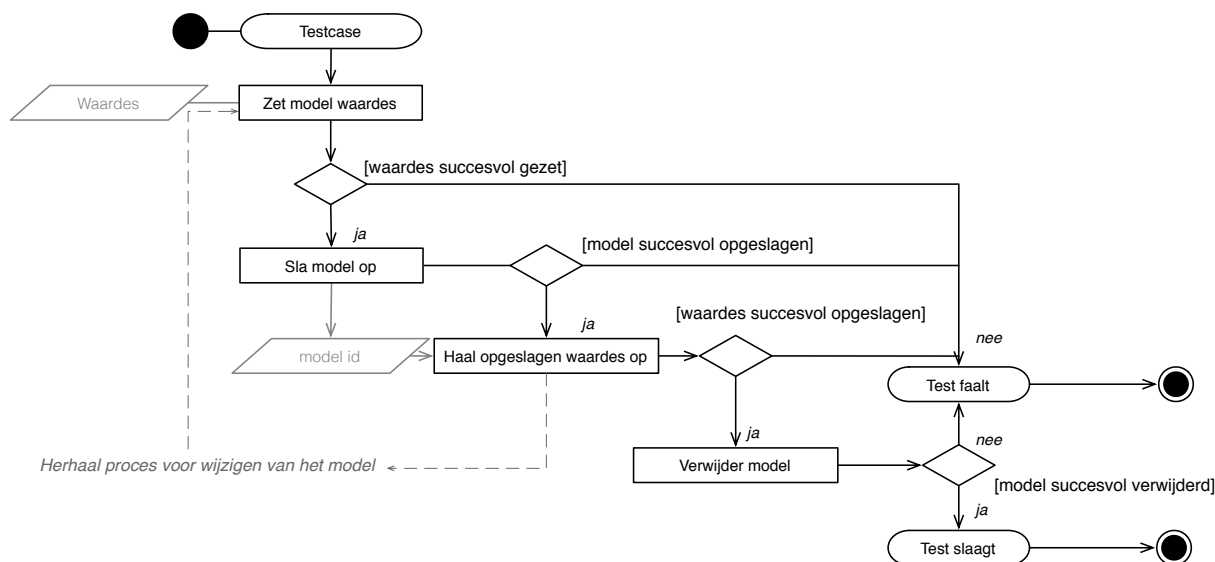
Functioneel

Functionele testen zijn verantwoordelijk voor het testen of de code-implementaties naar behoren functioneren. Het betreft hier voornamelijk specialistische functionaliteit waar een unieke testroute voor benodigd is.

Eén van de terugkerende kernonderdelen binnen de applicatie zijn modellen. Modellen representeren de code-implementatie van een database-entiteit. Met behulp van de modellen kunnen database-entiteiten worden toegevoegd, gewijzigd en verwijderd. Mede in verband met veiligheid van de applicatie is het dan ook van belang dat de modellen correct functioneren, denk hierbij aan de rechten van de gebruiker die verkeerd worden uitgelezen.

Voor het testen van de modellen wordt er eerst getest of de getters en setters van het model naar behoren werken. Dit wordt gedaan door een dataset in de test te definiëren welke één voor één in het model wordt gezet. Zodra de getters en setters correct werken wordt het model opgeslagen in de database. Eventueel worden hier extra testcases aan toegevoegd waarbij er gekeken wordt naar conflicten tussen unieke sleutels in de database, denk hierbij aan een reeds bestaande loginnaam.

Na het opslaan wordt er een nieuw model aangemaakt welke de dataset van het zojuist opgeslagen model inlaad, er wordt hier een nieuw model gemaakt om er zeker van te zijn dat er niet per ongeluk waardes blijven staan van het opgeslagen model. De waardes van het nieuwe model worden vervolgens vergeleken met de gedefinieerde dataset om te testen of de waardes correct zijn opgeslagen in de database. Het bovenstaande proces wordt vervolgens herhaald voor het aanpassen van het model. Ten slotte wordt het model weer uit de database verwijderd.



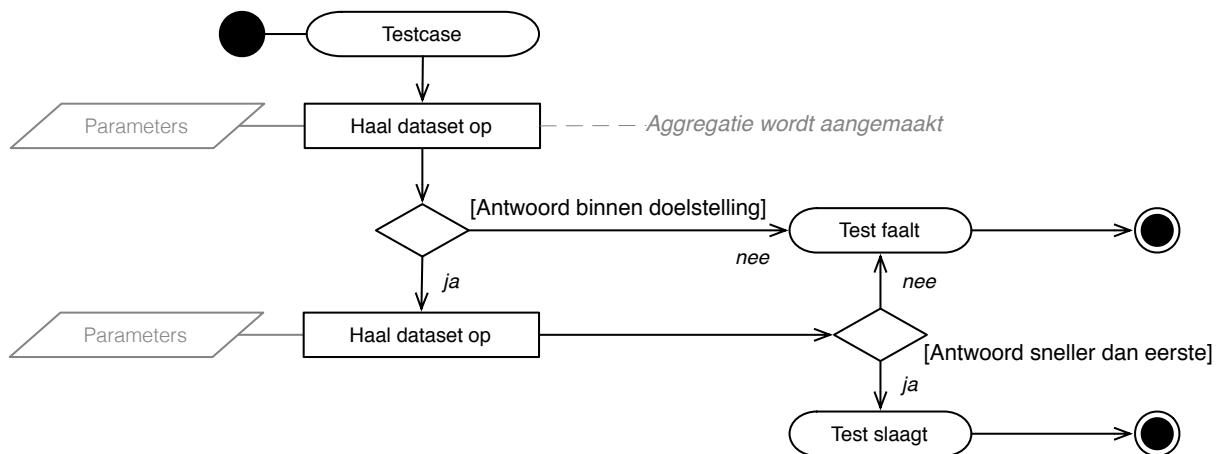
Testcase voor het valideren van een model



Performance

De beweegreden achter het bouwen van een nieuwe Search Traffic Monitor is de performance, het is dan ook niet meer dan logisch dat ook hier op getest wordt. Het testen van de performance gebeurt voornamelijk voor rapportage, hierin wordt getest of de rapportage geladen is binnen de afgesproken doelstelling van vijf seconden, zie 3.2 *Doelstelling*.

Bij het bouwen van de rapportage performance testen wordt er gekeken naar zowel het aanmaken van een nieuwe aggregatie als het ondervragen van een reeds bestaande aggregatie. Om de unittest te versnellen vindt deze test altijd plaats in combinatie met een data-integriteitstest.



Testcase voor het valideren van een model

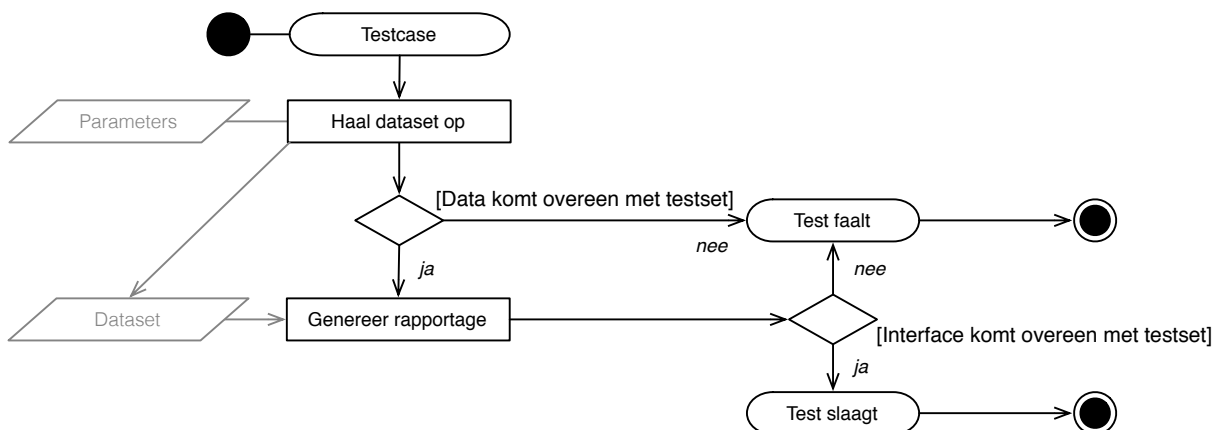


Data-integriteit

Omdat Search Traffic Monitor door de consultants wordt gebruikt voor het opstellen van SEM analyses kunnen we het niet veroorloven dat er foutieve data in de rapporten wordt getoond. Voor alle rapportages is daarom een unittest opgezet welke zowel de geretourneerde dataset als datarepresentatie test.

Keyword ▼	Traffic		Rank		
zorgverzekeringen vergelijken	79	51	1.00	1.00	
zorgverzekeringen	52	55	3.00	2.18	
zorgverzekering vergelijken 2012	26	22	2.00	2.31	
zorgverzekering vergelijken	145	116	1.87	2.65	
zorgverzekering studenten vergelijken	16	19	5.00	4.00	

In de bovenstaande afbeelding wordt een 'trending keywords' rapportage getoond, dit rapport vergelijkt de populariteit van een zoekterm met een voorgaande periode. In dergelijke rapportages is een fout makkelijk gemaakt wanneer de periodes in de verkeerde kolom getoond worden. Naast het testen van de geretourneerde dataset wordt daarom ook de datarepresentatie getest om te valideren of de data in de juiste kolom getoond wordt, hierbij wordt tevens rekening gehouden met eventuele dataconversie die plaats vindt.



Testcase voor het valideren van een data-integriteit



7.1.2.2 Uitvoering

In de nieuwe architectuur worden er verschillende lagen gedefinieerd welke allen een andere beleid volgen betreffende het opzetten van de unittests. Dit in verband met de taal die in de architecturale laag gehanteerd wordt.

PHP: Controller

Voor het testen van de controller wordt er gebruik gemaakt van de UTest plugin, een plugin gebouwd als uitbreiding op framework. Deze plugin is bekend met de codestructuur die binnen framework gehanteerd wordt en is aan de hand daarvan in staat om een globale unittests te genereren. Dit met het voordeel dat de developer niet per ongeluk testcases vergeet bij het bouwen van de unittest.

De controller richt zich primair op het testen van de functionaliteit en veiligheid van de applicatie.

```
/** EXECUTE -----
 * Returns whether the action is executed successfully.
 *
 * @param  enum $error = Pie_ErrorManager::ERROR_QUEUE The way a possible error should be handled.
 * @return bool
 */
public function execute($error = Pie_ErrorManager::ERROR_QUEUE)
{
}

/** TEST EXECUTE -----
 * Test case for execute method.
 * Returns whether the action is executed successfully.
 */
public function test_execute()
{
    $expected = null;
    $result    = false;

    // Method parameters
    // Type: enum
    // Description: The way a possible error should be handled.
    $error = Pie_ErrorManager::ERROR_QUEUE;

    // --- Start testing
    //$result = $this->object->execute($error);

    // --- End testing

    $this->assertEquals($expected, $result);
}
```

De UTest plugin genereert automatisch een testcase aan de hand van de codestructuur



HTML/JS: View

Voor het testen van de interface kan tevens gebruik worden gemaakt van UTest plugin. Het voordeel hiervan is dat dan alle testresultaten tezamen met de resultaten van de controller worden getoond, hiermee is in één overzicht de status van de gehele server zichtbaar, zie ook *6.3.1 Architectuur*.

Omdat het onmogelijk is om met unittests te valideren of de layout er correct uit ziet¹ blijft het testen van een correcte layout een menselijke taak. De view legt daarom de focus op het testen van de data-integriteit, performance en de functionaliteit van de rapportages en filters.

Python: Data interface

Voor het bouwen van de unittests in Python wordt er een zelfstandige testklasse geschreven die verantwoordelijk is voor het uitvoeren van de unittest. Er is hier expliciet gekozen om gebruik te maken van een eigen klasse in verband met de tijdelijke aard van de laag, zie ook *6.3.4 Python: Data interface*.

Voor de data interface ligt de focus van het testen op de functionaliteit, het testen op de data-integriteit is hier namelijk minder van belang aangezien deze al plaats vindt in de view laag. De beweegredenen om de data-integriteit in de view te testen is omdat in de view tevens de datarepresentatie wordt getest.

PHP: Dataverwerking

Conform de afbakening van de opdracht, zie ook *3.6 Afbakening*, wordt het testen van de dataverwerking achterwege gelaten. In verband met de korte tijdsperiode beschikbaar voor het afstuderen wordt het bouwen van de unittests voor deze laag uitgesteld tot na de afstudeerperiode.

¹ Zien is een zintuig dat nog niet aan (commerciële) computers is voorbehouden



7.1.3 Integratie- en acceptatietest

Normaliter wordt er aan het einde van het ontwikkeltraject een integratie- en acceptatietest uitgevoerd waarbij men de gehele applicatie nog één laatste maal test voor deze op de productieomgeving wordt gezet. Omdat Search Traffic Monitor momenteel een interne applicatie is waar relatief veel kosten aan gepaard zijn, wordt de productieomgeving tevens als acceptatieomgeving gebruikt. Het gevolg hiervan is dat applicatie gedurende de ontwikkeling al constant door onder andere de product owner wordt getest. Na elke sprint meeting is er een release op de productieomgeving gedaan welke de wijzigingen van de sprint bevatte.

Naast het gebruik van de productieomgeving voor het testen wordt de applicatie al door enkele consultants gebruikt voor onderdelen binnen hun SEM analyses voor websites van klanten. Door dit continue gebruik van de applicatie, de relatief korte sprints en de uitgebreide unittests heeft het uitvoeren van een integratie- dan wel acceptatietest geen toegevoegde waarde.

7.1.4 Overdracht

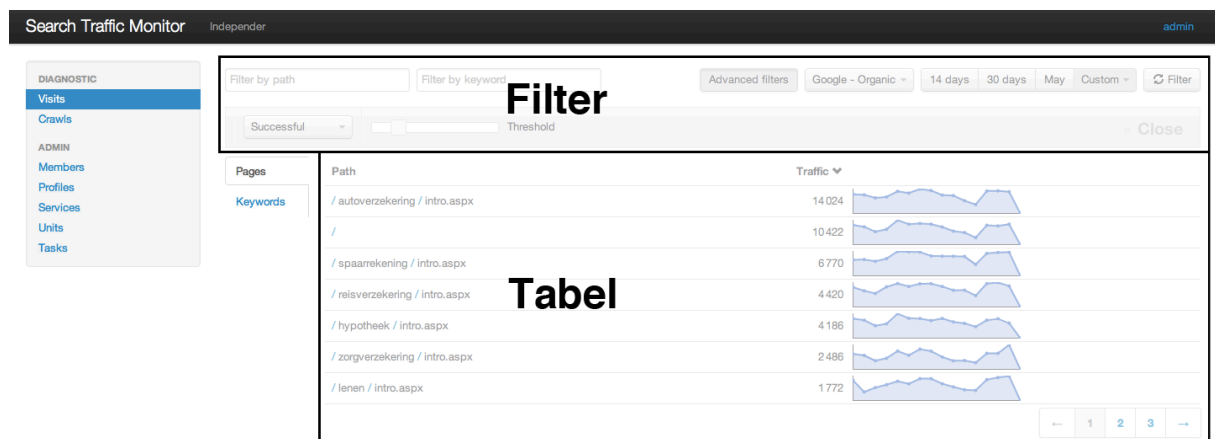
Om de complexiteit en overdraagbaarheid van de applicatie te testen heeft er een test overdracht plaats gevonden. Hierbij is het technisch ontwerp overgedragen aan een developer waarna hem gevraagd is of hij, met de kennis uit het technisch ontwerp, in staat is om Search Traffic Monitor verder te ontwikkelen. Dit bleek geen probleem, de voorbeeld cases in het technisch ontwerp beschreven de opbouw van de applicatie afdoende opdat deze door ontwikkeld kan worden.



7.2 Rapportage

Als we kijken naar de opbouw van de rapportages zien we dat deze consistent uit twee elementen bestaat, een tabel waarin de data getoond wordt en filters waarmee de inhoud van de tabel kan worden beïnvloed. Hierbij zijn de filters een terugkomend element welke tussen de verschillende rapporten hergebruikt worden.

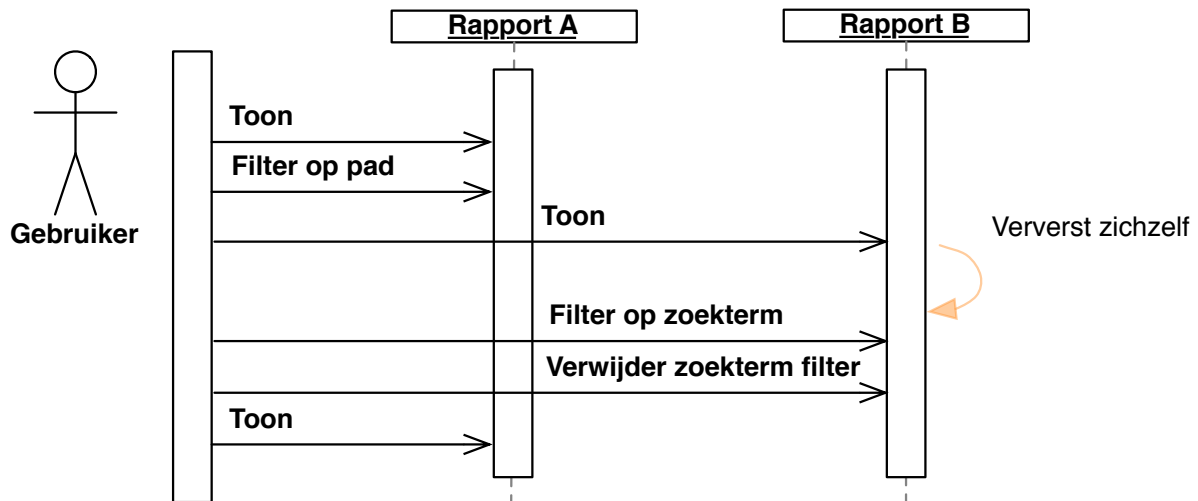
In de bouw en het ontwerp is er daarom dan ook gekozen om de filters globaal te definiëren waarna deze gekoppeld worden aan de tabellen. Op deze manier is het mogelijk om de filters tussen meerdere rapportages te delen waarbij de filterwaardes worden behouden, dit zodat de gebruiker niet constant de filters aan hoeft te passen bij het wisselen van rapportage.



Omdat niet elk rapport dezelfde filters ondersteund moet er bij het wisselen van de rapportages tevens rekening worden gehouden met het wisselen van de filters. Zo is het bijvoorbeeld niet mogelijk om een rapportage betreffende zoekmachines te filteren op een zoekterm, het is dan ook niet relevant om de zoektermfilter te tonen bij een dergelijk rapport. Bij het tonen van een rapport wordt er daarom over de filters heen geïtereerd om te garanderen dat enkel de gerelateerde filters getoond worden.



Ter preventie van data-misinterpretatie wordt er bij het tonen van een rapport gekeken of er tussentijds filters zijn gewijzigd. Dit wordt gedaan door de filterwaardes om te zetten in een sleutel welke wordt vergeleken met de laatst bekende sleutel, indien deze waardes niet overeen komen wordt de dataset in het rapport automatisch ververst.



De dataset wordt enkel ververst wanneer er sprake is van een tussentijdse wijziging. Wanneer rapport A voor het tweede maal wordt getoond is de tussentijdse wijziging alweer ongedaan gemaakt, hierdoor is het verversen van de dataset overbodig.

Om de rapportages snel te genereren wordt er gebruik gemaakt van aggregaties. Elk rapport kan bestaan uit één of meerdere aggregaties die gedefinieerd worden in een configuratiebestand. Door de aggregatietypes te beheren vanuit een configuratiebestand kan de applicatie gemakkelijk worden uitgebreid met nieuwe rapporten zonder dat er codewijzigingen voor benodigd zijn.



7.3 Management

Het management gedeelte van de applicatie is verantwoordelijk voor het beheer over alle entiteiten binnen Search Traffic Monitor. De opbouw van een dergelijk entiteitbeheer is in grootte lijnen hetzelfde daarom wordt er een relatief gestandaardiseerd traject gevolgd.

Allereerst wordt de applicatie uitgebreid met een tabel welke een lijst van de entiteiten in de database representeert. Middels deze tabel kunnen entiteiten worden verwijderd of kan men navigeren naar het toevoegen dan wel wijzigen van een entiteit. De inhoud en handelingen omtrent de tabel worden bepaald aan de hand van de rechten van de gebruiker. Zo geeft de tabel een foutmelding indien de gebruiker handelingen probeert uit te voeren waar zij niet toe gemachtigd is.

Profile list Add profile		
Action Add profile Items 0 / 3 Total 3		
Name	Domain	Database
Independer Edit Activate Delete	www.independer.nl	independer@localhost
PieFramework	pieframework.nl	stm@stmdev.orangevalley.nl
Springer.com	www.springer.com	springer@localhost

Beheren van de profielen middels een tabel

Vervolgens wordt er een modelklasse geschreven welke een code-implementatie van de database-entiteit representeert. Middels een formulier is de gebruiker vervolgens in staat om entiteiten te maken/wijzigen. Dit formulier is verantwoordelijk voor de validatie van de gebruikersinvoer opdat enkel valide entiteiten in de applicatie worden opgeslagen.

Profile list	Add profile	
Information	Data gathering	Keywords

Information

Name

Domain

Member

Database

Host

Name

Toevoegen van een nieuw profiel middels een formulier



8 Operationeel werk

Een reeds bekend risico voor het afstuderen is dat een verleden met het bedrijf de kans op het verrichten van operationele werkzaamheden vergroot. In het begin van het afstudeertraject is er daarom rekening gehouden met het afronden en overdragen van de lopende zaken.

In de tweede week van het afstudeertraject zijn door onvoorziene omstandigheden twee werknemers binnen 42 functions weggevallen. Vanuit het management kwam dan ook de vraag om gedurende deze hectische periode het team te ondersteunen in het operationele werk. Deze periode heeft uiteindelijk vijf weken aangehouden, hierin zijn echter drie specifieke werkzaamheden in terug gekomen welke op een niveau toch gerelateerd zijn aan Search Traffic Monitor.



8.1 Google Analytics SiteCheck

Eens per jaar organiseert Google een Google Analytics User Conference waarbij er dieper wordt ingegaan op de mogelijkheden van Google Analytics. Op deze conferentie krijgen bedrijven de gelegenheid om workshops en presentaties met betrekking tot Google Analytics te geven.

Als marketing stunt had OrangeValley bedacht om een tool te bouwen die controleert of de Google Analytics op een website op een correcte wijze is geïmplementeerd. Hiervoor maakt de tool gebruik van een crawler welke recursief de pagina's van de website ophaalt en analyseert. Na de analyse stuurt de tool een pdf via de mail op waarin vaak verbeterpunten staan.

Omdat Search Traffic Monitor van een zelfde principe gebruikt maakt voor het detecteren van wijzigingen op de website was het in verband met het korte tijdsbestek logisch dat de kennis van Search Traffic Monitor ook voor het ontwikkelen van deze tool werd ingezet.



8.2 Google AdWords Dynamische Advertenties

Vooruitlopend op de afstudeeropdracht kwam er via de AdWords consultants de vraag of wij advertenties dynamisch konden maken aan de hand van een XML feed. Eén van onze klanten in de reisbranche heeft namelijk te maken met voortdurend wisselende prijzen die zowel in de advertenties als de AdWords campagnes aangepast moet worden.

Omdat het hier een vraag vanuit een klant betrof kreeg het een hogere prioriteit dan het adviesrapport, zodoende werd er vooruit gewerkt aan de Google AdWords integratie in Search Traffic Monitor. Op basis van de Google sandbox¹ is er een AdWords integratie gebouwd in Search Traffic Monitor welke in staat was om dynamische advertenties te generen op basis van een XML feed.

Voordat deze integratie getest kon worden op een productieomgeving moet er sinds kort goedkeuring gevraagd worden aan Google. Google bepaalt aan de hand van het doel van de applicatie wie er toegang krijgt tot hun API. Op 17 februari is er samen met de SEA afdeling een aanvraag bij Google ingediend, hier is tot op heden nog geen reactie op geweest. Dit is een onvoorzien risico die per toeval vroegtijdig is getackeld, indien dit niet vroegtijdig was opgepakt waren we hier pas tijdens het ontwikkelen van de sales module tegenaan gelopen, in dat geval hadden we een hele module moeten ontwikkelen op basis van de sandbox. Hoewel het ontwikkelen op basis van de sandbox in de theorie geen probleem is blijkt uit verleden dat er in de praktijk nog wel eens kleine verschillen tussen de sandbox en productieomgeving voordoen.

¹ Veilige testomgeving van Google



8.3 Reparatie Rate Your Company

Rate Your Company is één website ontwikkeld door 42functions. Op deze website kunnen bezoekers zien wie de populairste werkgevers in Nederland zijn, waarbij ze eventueel een beoordeling achter kunnen laten over hun eigen werkgever.

Om het registreren van de bedrijven niet aan de bezoekers over te laten maakt de website gebruik van een LinkedIn integratie. Via LinkedIn wordt bedrijfsinformatie automatisch doorzocht, opgehaald en opgeslagen.



Rate Your Company in opdracht van FNV Bondgenoten

In de eerste instantie leek de LinkedIn integratie naar behoren te functioneren dus werd de website volgens planning in productie gezet. Na de eerste drieduizend beoordelingen bleek echter dat sommige beoordelingen onder het verkeerde bedrijf werden geplaatst. De eerste vragen vanuit de klant waren dan ook hoe komt dit, hoe vaak is dit voorgekomen en kunnen we dit oplossen?

Na enig onderzoek bleek dat er iets fout was gegaan nadat op het laatste moment één van de requirements is veranderd. Zo bleek dat bij het registreren van een nieuw bedrijf de beoordeling verkeerd geplaatst werd indien een andere bezoeker in de tussentijd van de zoekfunctie gebruik had gemaakt. Iets wat in de acceptatietest niet naar voren is gekomen door het beperkte aantal testgebruikers.

Om de beoordelingen te kunnen repareren moesten we achterhalen wanneer er meerdere bezoekers op de website actief waren; hierbij moesten we ons specificeren tot een bezoeker welke een beoordeling aan het plaatsen was terwijl een andere bezoeker in de tussentijd een zoekopdracht uitvoerde. Indien deze situatie zich voor deed moest er achterhaald worden welk bedrijf de bezoeker de bezoeker aan het beoordelen was en waar de beoordeling uiteindelijk geplaatst is.

Normaliter klinkt het terugvinden van zoiets als een onmogelijke taak, zij het niet dat we complete gebruikerssessies nabootsen in de logfileanalyses van Search Traffic Monitor. Door een variant van de logfileanalyse te ontwikkelen en deze te specificeren op de hierboven beschreven situatie hebben we uiteindelijk alle foutief geplaatste beoordelingen kunnen repareren.



9 Evaluatie

9.1 Producten

Search Traffic Monitor

Aangezien we nog steeds geen toestemming hebben ontvangen voor het gebruik van de Google AdWords API hadden we nooit een volwaardige sales module kunnen ontwikkelen. De module had voor het bedrijf geen waarde gehad en was voor de hogeschool niet afstudeerwaardig geweest.

Ik ben daarom erg content over Search Traffic Monitor zoals deze nu is opgeleverd. De keuze om Search Traffic Monitor opnieuw te ontwikkelen is uiteindelijk de juiste keuze geweest. In deze korte ontwikkelperiode is de applicatie in vele aspecten aanzienlijk verbeterd ten opzichte van haar voorganger.

Verbeteringen

- Ziet er prachtig uit
- Is intuïtief in gebruik
- Is veel gebruiksvriendelijker
- Is technisch beter opgebouwd
- De performance is aanzienlijk verbeterd
- Kwaliteit is stabiel door unittests

Nadat er besloten was een nieuwe applicatie te ontwikkelen is er samen met de product owner voor de resterende tien weken een nieuwe doelstelling gedefinieerd. In deze doelstelling is voorgedragen om vier rapportages te maken, een beheer te ontwikkelen en indien er genoeg tijd voor over was, een start te maken met de geautomatiseerde regels.

Uiteindelijk zijn er vijf rapportages ontwikkeld waaronder een 'trending keywords' rapport die een trendanalyse geeft over de populariteit van een zoekterm afgezet tegen een voorgaande periode. Alle rapportages zijn aangevuld met een CSV export knop zodat de consultants ook in staat zijn om de data te gebruiken in Excel. Het beheer is zoals afgesproken ontwikkeld en is er redelijke opzet gemaakt voor de geautomatiseerde regels. Naar mijn optiek is de doelstelling dan ook ruim behaald.

In tegenstelling tot de oude situatie bevind Search Traffic Monitor zich voor het eerst in een staat waarin deze ook door klanten kan worden gebruikt. Wat tot mijn spijt een gemiste doelstelling voor januari 2012 was. Een doel die ik persoonlijk dit jaar vooralsnog verwacht te halen; dat enkele consultants Search Traffic Monitor al gebruiken lijkt dit beeld te stimuleren.



Adviesrapport

Op 15 maart 2012 is er door de product owner besloten geen tijd meer te besteden aan het afronden van het adviesrapport. Hoewel het een compleet terechte keuze is doet dit geen eer aan de staat waarin deze nu is opgenomen in de bijlage. Het adviesrapport is overduidelijk een onafgerond product die zich hooguit in een conceptfase bevindt.

15 maart 2012 was tevens de datum waarop het adviesrapport volgens planning af had moeten zijn. Door onverwachte operationele werkzaamheden was deze planning echter niet meer haalbaar. Mocht dit zich niet voorgedaan hebben dan was er het adviesrapport conform de planning opgeleverd waarna we vooralsnog hetzelfde ontwikkeltraject in waren gegaan.

Hoewel ik erg tevreden ben over de voltooide onderdelen in het adviesrapport, zoals de het advies om over te stappen naar een alternatief framework, representeert het verslag niet goed waarom er besloten is over te stappen van het databasemanagementsysteem. Het probleem bij het schrijven van het adviesrapport bevond zich niet in het testen en het vergelijken van de softwarepakketen maar bevond zich in het documenteren van de werkwijze en bevindingen. Het gevolg hiervan is dat bijvoorbeeld dat het adviesrapport niet beschrijft waarom er gekozen is voor MongoDB.

Dataverwerking

Een bewust ondergeschoven product in het afstudeerverslag is de dataverwerking. Hoewel dit onderdeel buiten de afbakening van het afstudeerverslag valt moest door een wijziging van het databasemanagementsysteem en bevindingen uit het adviesrapport de dataverwerking vooralsnog aangepast worden. Zoals beschreven in *6.3.7 PHP: Dataverwerking* heeft bij het aanpassen gelijk een optimalisatieslag plaats gevonden.

Door de SQL-unit samen te voegen met de processor-unit, een reverse IP lookup te verwijderen en het uitlezen van de logfiles te verbeteren hebben we de gemiddelde verwerkingstijd voor een logfile van 45 minuten kunnen terugbrengen tot een gemiddelde van 3 minuten. Een aanzienlijke verbetering waarmee het tonen van real time data steeds dichterbij komt, nu wordt het downloaden van een logfile de beknellende factor is.



9.2 Werkwijze

Scrum

Hoewel, door de opbouw, onderbelicht in het afstudeerverslag is er voor het ontwikkelen van de nieuwe applicatie gebruik gemaakt van Scrum. Deze is conform volgens planning gestart op 15 maart 2012 waarna er vier tweewekelijkse sprints volgden. Hoewel er vijf gepland waren is in overleg met de product owner ervoor gekozen om de vijfde sprint volledig te gebruiken voor het afronden en verbeteren van het afstudeerverslag.

Het verloop van de sprints verliep relatief goed. Enkel in de laatste twee sprints zijn niet alle sprint-items behaald, de reden hiervoor was dat teamgenoot Mark Abspoel vanaf sprint twee beperkt inzetbaar werd.

Wireframes, Mockups en een Prototype

Alvorens er begonnen is aan de definitieve ontwikkeling van de applicatie zijn er meerdere stadia doorlopen waarmee het ontwerp is getest. In deze stadia zijn de producten zo snel mogelijk ontwikkeld waarbij het ontwerp minder nauw gevolgd wordt en unittesten achterwege worden gelaten. Het gevolg hiervan is dat er heel snel resultaten zichtbaar zijn, wat verkeerde verwachtingen wekt bij de opdrachtgever omdat het hier een tussenproduct betreft.

Zo is het prototype door de gecreëerde verwachtingen uiteindelijk in de applicatie geïmplementeerd. Pas in de laatste sprint nam de druk weer een beetje af waarna deze vervangen kon worden door een permanente versie conform ons ontwerp.

Uiteindelijk heeft het doorlopen van deze drie stadia ook nog goede gevolgen gehad, de stakeholders waren erg enthousiast over de voortgang en hadden vrijwel geen aanmerkingen. Bij de bouw van het prototype werd duidelijk dat het opbouwen van de view een andere aanpak vereiste dan we voor ogen hadden. Zo bleek het beter om de filters ook middels JavaScript te doen, hiermee worden de filters beheert vanuit één locatie en kunnen ze gemakkelijker gekoppeld worden aan de rapportages.

Unittest

Het gebruik van unittests is iets wat binnen 42functions nog niet gebruikelijk is. Zo is de voorgaande versie van Search Traffic Monitor ontwikkeld zonder unittest. Het gevolg hiervan was dat er soms onderdelen kapot gingen na een release. Omdat het testen van de gehele applicatie namelijk onredelijk was werden enkel de gewijzigde onderdelen getest. Het gehele testgebied was echter moeilijk te definiëren door de vele onderliggende afhankelijkheden in de applicatie.

Door nu gebruikt te maken van test-driven development is het uitvoeren van releases een stuk stabiel geworden en neemt de kwaliteit van de applicatie aanzienlijk toe. Ik hoop hiermee een voorbeeld neer te zetten voor de rest van het bedrijf opdat ook daar vaker gebruik wordt gemaakt van unittest.



9.3 Eervolle vermelding

Onze consultants zijn dagelijks bezig met het verbeteren van websites om hoger te ranken in de zoekindex. Het draait hierbij allemaal om hoe relevant jouw pagina is voor de bezoeker en of deze pagina überhaupt gevonden kan worden door de zoekmachine.

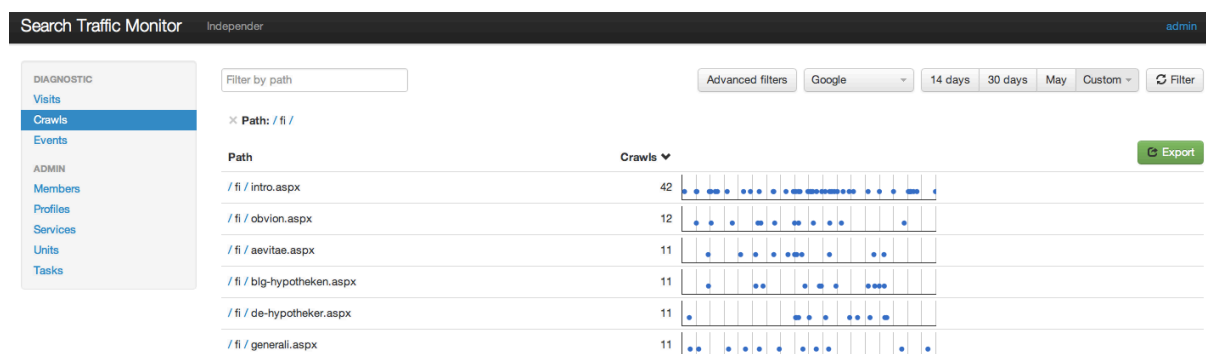
Nu zijn zoekmachines dergelijk geavanceerd dat ze vrij goed door de website heen navigeren en alle pagina's netjes indexeren. De zoekmachines hebben echter ook maar een beperkte hoeveelheid resources dus zij navigeren beperkt door de website heen, dit noemen we het crawl budget. Zodra jouw crawl budget bereikt is stopt de zoekmachine met het doorlopen van de site en komt hij later weer terug.

Nou is dit vaak geen probleem, tenzij je een webshop beheert en van de zoekmachines verwacht dat ze netjes door de productlijsten heen navigeren om ze te indexeren. Spijtig genoeg is dit een punt dat voorheen nog niet goed door de zoekmachines werd opgepakt wat ten koste ging van het crawl budget.

Pagina: **1** | 2 | 3 | ... | 23 | ▶ Volgende

Om dit op te lossen hebben zoekmachines vorig jaar aan gegeven in het vervolg te gaan kijken naar specifieke elementen op de pagina die aangeven dat er sprake is van paginatie. Middels deze elementen kunnen we aangeven wat de volgende pagina en wat de voorgaande pagina is binnen onze gedefinieerd reeks, hierdoor wordt het crawl budget netjes verdeeld over de hele lijst.

Nu hebben we Search Traffic Monitor al langer draaien op independer.nl, dergelijk lang dat independer data bevat van voor en na deze wijziging van de zoekmachine. Een prima case dan ook om over te publiceren.



Crawlgedrag na invoer van de technische oplossing

Product owner Eduard Blacqui re,   n van de grootste namen op het gebied van Search Engine Optimisation binnen Nederland, heeft daarom nu een artikel¹ gepubliceerd waarin hij de effecten van de paginatie-oplossing toont. Hierbij is Search Traffic Monitor als bron opgenomen. Dit toont aan dat het bedrijf in Search Traffic Monitor geloofd en vertrouwt op de kwaliteit van de applicatie.

¹ Voor het artikel zie: <http://orangevalley.nl/maximaal-seo-resultaat-paginerings/>



9.4 Beroepstaken

1.3 Selecteren van standaardsoftware

Adviseren over en/of selecteren van software, zoals een applicatie (commerciële software of open source), framework (al dan niet gebaseerd op open standaarden), databasemanagementsysteem of besturingssysteem.

Een deel van mijn afstuderen besteed ik aan het analyseren van de performanceproblemen binnen Search Traffic Monitor. Binnen deze analyse wordt er gekeken naar de structuur van de database, het gebruikte databasemanagementsysteem, de server waarop het systeem draait en ten slotte het systeem zelf. Aan de hand van deze analyse worden er vervolgens oplossingen en bijbehorende adviezen voorgesteld opdat het management vervolgens een gegronde keuze kan maken.

In het eerste stadium van het afstudeertraject is een adviesrapport opgesteld waarin adviezen zijn voorgedragen ter verbetering van de performance. In de totstandkoming van het adviesrapport is er gezocht naar bottlenecks in de applicatie middels systeem-profilering en query analyses. Ter verbetering van de bottlenecks zijn er verschillende adviezen voorgedragen aan de hand van een inspanning/opbrengst analyse.

Twee van de voorgedragen adviezen betroffen het overstappen van softwarepakket. Door middel van verschillende benchmarking testen bleek het gunstiger voor Search Traffic Monitor om zowel van framework als van databasemanagementsysteem over te stappen naar alternatieve pakketten. De product owner heeft uiteindelijk aan de hand van het adviesrapport besloten een vernieuwde Search Traffic Monitor te ontwikkelen.

3.2 Ontwerpen systeemdeel

Beschrijven van systeemdelen (subsystemen, componenten, modules), hun onderliggende structuur en het gedrag in detail, zodanig dat bouwen van het systeemdeel mogelijk is. Ontwerpen (grafische) user interface.

Omdat er gebruik wordt gemaakt van een framework wordt de architectuur van de SEA module al afgedwongen. Er wordt voor deze beroepstaak dan ook enkel klassendiagrammen, use-cases, sequentiediagrammen en grafische ontwerpen opgesteld. Deze worden terug gekoppeld aan de opdrachtgever om zo een duidelijk beeld van de gewenste functionaliteit te vormen.

Alvorens de vernieuwde versie van Search traffic Monitor ontwikkeld kon worden is er een technisch ontwerp opgesteld waarin de structuur van de applicatie is gedefinieerd. Hiervoor is gebruik gemaakt van klassendiagrammen, EER modellen, flowcharts, wireframes en sequentiediagrammen. Het gebruik van use-case-diagrammen is hier expliciet achterwege gelaten omdat deze geen toegevoegde waarde bleken te hebben.



3.3 Bouwen applicatie

Verfijnen en/of transformeren van het (detail)ontwerp van de systeemdelen (subsystemen, componenten, modules) van een applicatie. Bouwen en documenteren van de systeemdelen. Systeemdelen samenstellen tot een werkende applicatie.

De reeds ontworpen systeemdelen worden met behulp van test-driven development geïntegreerd in Search Traffic Monitor. Dit wordt op een professionele manier gedaan door gebruik te maken van SVN en een volledige OTAP ontwikkelstraat.

Na het opzetten van het technisch ontwerp is deze uitgewerkt tot een werkende applicatie. Hiervoor is gebruik gemaakt van de standaarden van 42functions, waarbij geldt dat er gebruik wordt gemaakt van een OTAP¹ ontwikkelstraat en SVN voor het gestructureerd uitvoeren van releases.

De complexiteit en overdraagbaarheid van de applicatie is getest door het technisch ontwerp aan een buitenstaande developer over te dragen waarbij gevraagd is of hij met behulp van het technisch ontwerp in staat van een nieuwe functionaliteit te integreren.

3.5 Uitvoeren van en rapporteren over het testproces

Opstellen testscript, waaronder het specificeren van de testgegevens, de benodigde startsituatie en de uitvoerverwachtingen per testgebied. Uitvoeren test plan en opstellen rapportage.

De nieuw ontworpen systeemdelen worden eerst omgezet naar unittests om zo de kwaliteit van de code gedurende het hele ontwikkeltraject te monitoren.

Voordat de applicatie gebouwd werd zijn er meerdere stadia doorlopen waarmee getest is of de product-owner content was met de layout en gebruikersapplicatie van de nieuwe applicatie. Zo zijn er eerst wireframes, mockups en een prototype gebouwd waarmee de product-owner een beeld kreeg van de uiteindelijke applicatie.

Voor het bouwen van de applicatie is er gebruik gemaakt van test-driven development, waarbij eerst unittests worden geschreven alvorens de functionaliteit wordt geïmplementeerd. Binnen deze unittests wordt er gekeken naar de functionaliteit, de veiligheid, data-integriteit en performance.

Omdat de applicatie al in gebruik is ondervindt deze continue gebruikerstesten waar feedback uit resulteert. Er heeft tevens een concurrency-test plaats gevonden na het afronden van het prototype waarbij de doelgroep gelijktijdig van de applicatie gebruik maakten.

¹ Ontwikkel Test Acceptatie Productie omgeving



10 Begrippenlijst

- *API*
Application Programming Interface, communicatie laag binnen de applicatie die het communiceren met het onderdeel mogelijk maakt. Dit kan om zowel interne als externe communicatie gaan.
- *Scrum - Backlog item*
Eén specifiek item van de product backlog.
- *Crawler*
Een AI die in staat is om geautomatiseerd webpagina's op te vragen. Doorgaans wordt dit gebruikt om de inhoud van de webpagina's te kunnen analyseren.
- *Cache*
De cache is tijdelijke opslag. In de webwereld wordt dit grotendeel toegepast op het inladen van de webpagina. Het doel hiervan is om de werkdruk op de server te minimaliseren en de bezoeker snel de webpagina te tonen. Om de bezoekers niet verouderde pagina's te tonen is de cache vaak gepaard aan een tijdslimiet, waarna deze automatisch vervalst.
- *Domeinmodel*
Een databasemodel waarin de tabellen en de relaties tussen de tabellen worden vastgelegd.
- *EER*
Extended Entity Relationship model, een databasemodel waarin de tabellen en de relaties tussen de tabellen worden vastgelegd. In tegenstelling tot het domeinmodel bevindt het EER diagram zich op een hoger abstract niveau, deze is niet gericht op een specifieke database waardoor hier vaak meer relaties zichtbaar zijn. Zie ook *Error! Reference source not found. Technisch ontwerp*.
- *Iteratie*
Vastgestelde herhalende periode waarin een circulatie van de gekozen software ontwikkelmethodiek plaats vindt.
- *Model View Controller*
Het MVC model is een veelgebruikt design pattern waarin de applicatie opgedeeld wordt in drie onderdelen. Het scheiden van deze lagen bevordert de leesbaarheid en herbruikbaarheid van de code.



- *Organisch zoekverkeer*
Organisch zoekverkeer definieert men als bezoekers die met behulp van een zoekmachine op de website binnen komen, hierbij wordt zoekverkeer via advertenties en betaalde zoekopdrachten uitgesloten.
- *OTAP*
Ontwikkel, Test, Acceptatie, Productie omgeving. Het werken via de OTAP methode wordt gebruikt om het aantal bugs te reduceren.
- *Product backlog*
Een featurelist samen gesteld door de stakeholders, ingericht aan de hand van de Scrum methodiek.
- *Request*
Een request is een verzoek richting de server voor een specifiek bestand. Een webpagina wordt tegenwoordig gevormd uit meerdere requests. De voornaamste requests omvatten: structuur, styling, plaatjes en gebruikersinteractie. Het aantal requests per pagina loopt over het algemeen vrij snel op, voor het opvragen van nu.nl worden bijvoorbeeld 78 requests uitgevoerd.
- *RRM*
Relationeel Representatie Model, tekstuele weergave van domeinmodel.
- *SEA*
Search Engine Advertisement, het investeren in zoekmachine marketing. Dit is eenvoudig gezegd reclame inkopen bij een zoekmachine op bepaalde zoektermen met het doel om meer bezoekers te halen.
- *SEM*
Search Engine Marketing, verzamelterm voor SEO en SEA.
- *SEO*
Search Engine Optimisation, het optimaliseren van de website gericht op zoekmachines. Het doel is om de websites zo hoog mogelijk op gerelateerde sleutelwoorden binnen de zoekmachine te laten positioneren.
- *Scrum*
Scrum, een software ontwikkelmethodiek werkende met korte iteraties.
- *Scrum - Sprint backlog*
Een samenvattend rapport voorgaande aan een nieuwe iteratie binnen de Scrum ontwikkelmethodiek. Hierin wordt beschreven welke backlog items er in de sprint opgeleverd worden.
- *Scrum - Sprint burndown*
Een samenvattend rapport over de geboekte resultaten van een iteratie. Met de resultaten van dit rapport kan de planning bijgestuurd worden.



- *STRICA*
Sales Traffic Ranking Indexing Crawling Adjustment. Dit traject staat garant voor het compleet in kaart brengen van de effecten van Search Engine Marketing adviezen.
- *VPS*
Virtual Private Server, een gehele server die wordt verhuurd aan één klant. In de context van dit verslag wordt deze voornamelijk gebruikt als webserver.



11 Bijlagen

A	Literatuurlijst	68
B	Plan van aanpak (1)	70
C	Plan van aanpak (2)	82
D	Technisch ontwerp	96
E	Scrum	130
F	Adviesrapport	143
G	Afstudeerplan	193
H	Tussentijdse beoordeling concept	199
I	Tussentijdse beoordeling assessment	201
J	Evaluatie bedrijfsmentor	205



A Literatuurlijst



Uitgever/locatie	Titel	Datum¹	Auteurs
1 O'Reilly Media	<i>High Performance MySQL</i>	juni 2008	Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy D. Zwaodny, Arjen Lentz & Derek J. Balling
2 http://mysql.com/	<i>MySQL</i>	20 februari 2012	Oracle Corporation
3 http://tinyurl.com/2fuaeyk	<i>SQL Query Order of Operations'</i>	13 maart 2012	Ben Nadel
4 Pearson Education Benelux	<i>Praktisch UML</i>	15 mei 2007	Jos Wamer & Anneke Kleppe
5 http://mongodb.org	<i>MongoDB</i>	7 maart 2012	10gen
6 http://couchdb.apache.org/	<i>CouchDB</i>	7 maart 2012	Apache Software Foundation
7 http://oracle.com	<i>Oracle</i>	7 maart 2012	Oracle Corporation
8 http://www.carrieretijger.nl	<i>SMART doelen stellen</i>	16 mei 2012	René Pijlman

¹ Voor webpagina's wordt de datum van raadplegen getoond



B Plan van aanpak (1)



Inhoudsopgave

1	Aanleiding	72
2	Doelstelling	72
3	Werkwijze	73
4	Afbakening	74
4.1	<i>Afbakening</i>	74
4.2	<i>Randvoorwaarden</i>	74
5	Op te leveren producten	75
5.1	<i>Adviesrapport performanceproblemen</i>	75
5.2	<i>Specifieke statistische functionaliteit</i>	75
5.3	<i>Sales module</i>	76
5.4	<i>Plan van Aanpak</i>	76
6	Rollen	77
7	Risicoanalyse	78
8	Mijlpalen en planning	79
9	Bijlage	80
9.1	<i>Voorbeeld: sprint backlog</i>	80
9.2	<i>Voorbeeld: sprint burndown</i>	81



1 Aanleiding

Search Traffic Monitor is softwarepakket welke inzicht geeft in het gedrag van zoekmachines. Door het gedrag van zoekmachines te analyseren kan OrangeValley haar diensten beter op de zoekmachines afstemmen om zo nog betere resultaten voor haar klanten te boeken. Eén van de doelen van OrangeValley is dan ook om Search Traffic Monitor aan de standaard toolkit van haar consultants toe te voegen.

Het enige wat OrangeValley hiervan weerhoud is dat Search Traffic Monitor momenteel te traag is voor operationeel gebruik. Search Traffic Monitor heeft momenteel vijf minuten aan laadtijd per rapport nodig, dit met slechts één actieve databaseconnectie en een kleine database van 12 gigabytes. Voordat Search Traffic Monitor aan de standaard toolkit van de consultants kan worden toegevoegd is het noodzakelijk om de huidige performanceproblemen te analyseren en verbeteren voordat de consultants met Search Traffic Monitor aan de slag kunnen.

Naast het oplossen van de performanceproblemen is het gewenst dat Search Traffic Monitor in staat is om het gehele STRICA¹ traject in kaart te brengen. Momenteel mist Search Traffic Monitor echter nog de functionaliteit om inzicht te geven in de onderdelen Indexing en Sales.

2 Doelstelling

Aan het einde van de afstudeerperiode moet er een adviesrapport liggen voor het verbeteren van de performance. Afhankelijk van de prioriteit die de opdrachtgever toekent aan de verschillende verbetervoorstellen worden deze voorstellen gedurende het afstuderen gerealiseerd.

Naast het verbeteren van de performance is het tijd om de volgende stap richting STRICA te zetten, daarvoor wordt Search Traffic Monitor uitgebreid met een sales module. Door het systeem uit te breiden met een sales module kan Search Traffic Monitor in het vervolg ook inzichten creëren op het gebied van SEA².

¹ Sales Traffic Ranking Indexing Crawling Adjustment

² Search Engine Advertising



3 Werkwijze

Alvorens er begonnen kan worden aan de bouw van een nieuwe sales module is het noodzakelijk dat het adviesrapport wordt opgesteld. De bouw en het ontwerp van de sales module is namelijk sterk afhankelijk van de bevindingen die voortvloeien uit het adviesrapport. Indien er bijvoorbeeld geadviseerd wordt om gebruik te gaan maken van een alternatief databasemanagementsysteem kan de product owner ervoor kiezen om de sales module in een apart traject te ontwikkelen waarin gelijk gebruik wordt gemaakt van het alternatieve databasemanagementsysteem.

Na het afronden van het adviesrapport wordt er met behulp van Scrum een iteratief ontwikkeltraject opgezet waarmee de sales module wordt ontwikkeld. Hiervoor worden sprints van twee weken gehanteerd, dit in verband met het verzoek van de stakeholders om elke twee weken een voortgangsrapportage op te leveren.

Voor het indelen van de sprints wordt er eerst samen met de stakeholders en SEA consultants een product backlog opgesteld waarin alle gewenste functionaliteiten als backlog-items worden opgenomen. Op basis van een inspanning/opbrengst analyse worden de sprint items door de product owner aan de sprints toebedeeld. In deze analyse wordt de ontwikkeltijd afgewogen tegen de bedrijfswaarde voor de stakeholders.

Activiteiten

- Opstellen adviesrapport performanceproblemen
- Realisatie specifieke specialistische functionaliteit
- Realisatie sales module
- Opstellen product backlog

Activiteiten [iteratief op basis van sprint]

- Opstellen technisch ontwerp
- Opzetten testtraject
- Realisatie functionaliteit
- Opstellen burndown chart
- Opstellen sprint backlog



4 Afbakening

4.1 Afbakening

- Focus op sales module

De sales module is slechts een subonderdeel van Search Traffic Monitor. Er wordt enkel aan andere onderdelen gewerkt wanneer dit onvermijdelijk is, bijvoorbeeld door wijzigingen die gemaakt moeten worden in de structuur van de database.

- Framework buiten beschouwing

De nieuwe module wordt geschreven als uitbreiding op het reeds bestaande Zend Framework. Hiervoor geldt dat er enkel ontwerpen en testcases worden opgesteld betreffende onze extensies op dit framework.

- Bestaande onderdelen buiten beschouwing

Reeds bestaande onderdelen van Search Traffic Monitor worden buiten beschouwing gelaten, hier worden geen ontwerpen dan wel unittests voor geschreven.

4.2 Randvoorwaarden

De afstudeerder

- De afstudeerder is vijf dagen per week op de bedrijfslocatie aanwezig.
- Gedurende de afstudeerperiode van 6 februari tot 1 juni beperkt de afstudeerder zijn werkzaamheden tot de werkzaamheden besproken binnen dit plan van aanpak.

OrangeValley

- OrangeValley geeft de afstudeerder de gelegenheid om aan het afstudeerdossier te werken, hier wordt wekelijks één dag voor ingeplanned. Indien nodig kan er bij de opdrachtgever een verzoek worden ingediend voor extra dagen.



5 Op te leveren producten

5.1 Adviesrapport performanceproblemen

Het adviesrapport bevat een grondige analyse van Search Traffic Monitor's performanceproblemen. In dit rapport wordt Search Traffic Monitor op vijf niveaus onderzocht om zo een compleet beeld van de performance te vormen. Voor elk niveau wordt de huidige situatie in kaart gebracht en wordt er gekeken naar mogelijke bottlenecks¹ die zich binnen de huidige implementatie voor doen. Voor elke bottleneck worden er vervolgens mogelijke oplossingen voorgedragen ter 'verbreding' van de bottleneck. Ter afsluiting worden de voorgedragen oplossingen tegen elkaar afgewogen op basis van een inspanning/opbrengst analyse.

Het is uiteindelijk aan de product owner om aan te geven voor welke oplossingen er gekozen gaat worden. Afhankelijk van de prioriteit die de product owner aan de performance toekent kan er besloten worden om de oplossingen gedurende de ontwikkelfase van het afstuderen al te realiseren.

5.2 Specifieke statistische functionaliteit

Search Traffic Monitor is een applicatie die gebruikt wordt voor het opstellen van adviesrapporten, het is daarom van uiterst belang dat de data die Search Traffic Monitor toont accuraat is. Om rekening te houden met uitschieters binnen de dataset wordt daarom vaak gebruik gemaakt van een mediaan².

Het berekenen van een mediaan is een functie die MySQL standaard niet ondersteunt. Dit met het gevolg dat we de mediaan via omwegen moeten gaan berekenen, dit leidt vaak tot onnauwkeurigheid of een traag systeem. Om de mediaan vooralsnog te kunnen berekenen moeten we momenteel de data tijdelijk naar de applicatie transporteren. Hiervoor is naast het gebruik van meerdere query's extra geheugen nodig waarin de resultaten tijdelijk worden opgeslagen. In sommige gevallen wil een reeks honderden megabytes aan data beslaan wat afgewogen moet worden tegenover het maximum van 32 megabyte waarover we in PHP³ beschikken.

Om bovenstaande constructies te vermijden is het gewenst dat MySQL wordt uitgebreid met een mediaan functionaliteit middels een UDF⁴. Er is expliciet gekozen voor een uitbreiding op het databasemanagementsysteem om zo de optimale performance te behalen. Door de mediaan in de database te berekenen hoeven er geen alternatieve query's meer worden uitgevoerd, dit scheelt zowel in de dataoverdracht als de werklast op de database. In tegenstelling tot PHP heeft MySQL namelijk veel meer geheugen tot haar beschikking omdat zij ook gebruik mag maken van de harde schrijf om haar data tijdelijk weg te schrijven.

¹ Mogelijk knelpunt voor performance.

² Middelste waarde van een geordende datareeks.

³ Omdat PHP als sub proces per aanvraag gestart wordt kan het geheugen snel vol lopen wanneer er veel aanvragen plaats vinden, daarom wordt er per aanvraag maximaal 32MB geheugen gealloceerd.

⁴ User Defined Function, C++ code extensie voor MySQL distributie



5.3 Sales module

Een onderdeel van de afstudeeropdracht is het uitbreiden van Search Traffic Monitor met een geheel nieuwe sales module. Voor de bouw van deze module wordt er eerst met de stakeholders en een SEA consultant een product backlog met de gewenste functionaliteiten opgesteld. Deze functionaliteiten worden primair opgesteld aan de hand van grafische ontwerpen die vervolgens door de developers worden opgedeeld in technische functionaliteiten opdat deze als backlog-items aan de product backlog kunnen worden toegevoegd.

Voor het bouwen van de sales module wordt er gebruik gemaakt van test-driven development. Voor de nieuwe functionaliteiten worden eerst technische ontwerpen opgesteld om zo eventuele problemen binnen de architectuur prematuur te kunnen herkennen. Nadat de ontwerpen zijn opgesteld worden er unittests opgezet waarmee de kwaliteit van de code kan worden bewaakt. Pas nadat de testen zijn gebouwd worden de functionaliteiten pas werkelijk geïmplementeerd.

Na oplevering van de sales module moet Search Traffic Monitor in staat zijn om inzichten te verschaffen op het gebied van SEA. Hierbij wordt de focus gelegd op het tonen van de conversiewaardes. De conversiewaardes kunnen namelijk concreet worden gebruikt om aan te tonen wat de financiële resultaten zijn van zoekmachineoptimalisatie.

5.4 Plan van Aanpak¹

In het plan van aanpak wordt de aanpak voor het afstuderen beschreven, hieronder valt een beschrijving van de activiteiten, de werkwijze, de methodieken, de risico's en de mijlpalen. Het doel van het plan van aanpak is om de opdrachtgever en de projectleden een duidelijk beeld te geven over hun betrokkenheid, werkzaamheden en verwachtingen van het afstudeerproject.

¹ Dit document is het plan van aanpak in kwestie



6 Rollen

De volgende personen zijn betrokken bij de afstudeeropdracht

<i>Rol</i>	<i>Medewerker</i>	<i>Functie</i>
Stakeholder / Product owner	Eduard Blacquièrre	Consultant zoekmachine optimalisatie
Stakeholder	Ortwin Verreck	Managing partner
ScrumMaster / Developer	Ferdy Perdaan	Technical consultant
Developer	Mark Abspoel	Data architect
Doelgroep	SEM consultants	

Stakeholder

De stakeholders voeden de product backlog met nieuwe gewenste functionaliteiten. Binnen de scope van het afstuderen worden echter enkel functionaliteiten met betrekking tot sales module en performance in behandeling genomen.

Product owner

De product owner staat garant voor de belangen van alle stakeholders. Hij is daarmee verantwoordelijk voor het toewijzen van de backlog items aan de verschillende sprints. Hiervoor wordt aan elk backlog item een prioriteit toegekend op gebaseerde op de bedrijfswaarde afgewogen tegen de tijd die ervoor nodig is.

ScrumMaster

De ScrumMaster wijst alle sprint-items aan de projectleden. Het is aan de ScrumMaster om het overzicht te houden over de voortgang van de sprints en het algehele project. Tevens is het de taak van de ScrumMaster om alle Scrum documentatie op te stellen.

Developer

De developers zijn verantwoordelijk voor de realisatie van de gewenste functionaliteiten. Hun taken worden na elke sprint door de ScrumMaster toebedeeld.

Doelgroep

Omdat Search Traffic Monitor een applicatie is welke toegevoegd moet worden aan de standaard 'gereedschapskist' van de consultants worden deze ter zijner tijd gevraagd om mee te helpen met het testen van de applicatie. Tevens is de doelgroep in staat om backlog-items toe te voegen aan de product backlog, het is echter aan de product owner en de developers om aan te geven of deze backlog-items bijdragen aan het doel van Search Traffic Monitor.



7 Risicoanalyse

Omdat ik al anderhalf jaar bij OrangeValley in dienst ben komen er enkele additionele risico's bij welke het afstuderen in gevaar brengen, het is daarom belangrijk dat er afspraken worden gemaakt om deze risico's te voorkomen.

- **Doelstelling wordt niet behaald**

In de doelstelling stellen we dat er naast een adviesrapport ook een SEA module wordt opgeleverd. Later geven we aan dat de oplossingen in het adviesrapport, afhankelijk van de prioriteit, eventueel in de tweede fase kunnen worden gerealiseerd. Wanneer de SEA module plaats maakt voor performance is er een kans dat de SEA module niet voltooid is aan het einde van het afstuderen, waarmee de doelstelling niet behaald is.

Voor OrangeValley is dit acceptabel omdat ze in dat geval bewust performance boven SEA kiezen, het is enkel de vraag of de competenties van het afstuderen nog behaald kunnen worden wanneer de SEA module (deels) weg valt. Het is daarom aan de afstudeerder om contact met de school coördinator op te nemen wanneer er van dit risico sprake is. In overleg met de school coördinator en het bedrijf kan er besloten worden om aan de performance te werken of juist niet.

- **Operationeel werk**

Naast het meedenken met collega's en het meehelpen aan oplossen van kleine problemen is het niet de bedoeling dat ik operationeel werk verricht, dat wil zeggen dat ik volledig ingepland moet staan voor mijn studie en voor Search Traffic Monitor. Om dit risico te voorkomen is de projectmanager ingelicht, hij let erop dat er genoeg tijd wordt besteed aan het afstuderen.

- **Minder 'zichtbare' resultaten**

In het normale ontwikkeltraject binnen OrangeValley is het ongebruikelijk dat er ook unit tests en uml diagrammen worden opgeleverd, deze bieden namelijk geen zichtbare toegevoegde waarde voor de klant. In het kader van mijn afstuderen wordt hier een uitzondering op gemaakt, dit betekent dat een deel van de normaal gesproken bouwtijd 'verloren' gaat aan het maken van ontwerpen en testen. Naast het bouwen, ontwerpen en testen moet er ook tijd vrij gemaakt worden voor het schrijven van het afstudeerdossier. Om dit

Uit ervaring met OrangeValley blijkt dat er niet gedaan wordt aan het bouwen van unit tests en het opstellen van systeemontwerpen, dit wordt simpelweg niet gedaan omdat dit geen zichtbare toegevoegde waarde biedt. In het kader van mijn afstuderen wordt hier dan ook een uitzondering op gemaakt, 'zichtbare' resultaten maken dus plaats voor het maken van ontwerpen en testen. Dit wil zeggen dat er tijdens deze periode minder gebouwd wordt dan normaliter wordt gedaan.

Het management is hierover ingelicht en toont zich hier bewust van te zijn. Het is de taak aan de afstudeerder en de bedrijfsmentor om tijdig in te zien wanneer de verwachtingen van de opdrachtgever(s) te hoog liggen opdat hier nog tijdig over gesproken kan worden.



8 Mijlpalen en planning

Mijlpaal	Datum
Start afstuderen	6 februari
<i>Plan van aanpak compleet</i>	8 februari
<i>Adviesrapport opgesteld</i>	15 maart
<i>Product backlog samengesteld</i>	16 maart
<i>Afronding sprint 1</i>	30 maart
<i>Afronding sprint 2</i>	13 april
<i>Afronding sprint 3</i>	27 april
<i>Afronding sprint 4</i>	11 mei
<i>Afronding sprint 5</i>	1 juni
Eind afstuderen	1 juni



9 Bijlage

9.1 Voorbeeld: sprint backlog

Periode	22 december 2010 - 5 januari 2011
Tijd per unit	30 Minuten
Units beschikbaar	87
Product owner	Eduard Blacquièrre
Projectleden	Ferdy Perdaan

Sprint items

#	Naam	Units
6.2.4 - 4	STM – Url Rapport - Crawlrate	2
6.2.4 - 6	STM – Url Rapport - Filter	2
8.2	Dashboard login / integratie ZF Authentication	8
8.3	Wisselen van klantenprofiel voor OV medewerkers	2
1.5	API - Module	20
1.2	API - Google Analytics API	10
8.4	Integratie SEO null meting	14
6.2.7	STM - Zoekmachine rapport	8
8.1	Opzetten dev omgeving voor het dashboard	8
Totaal		74

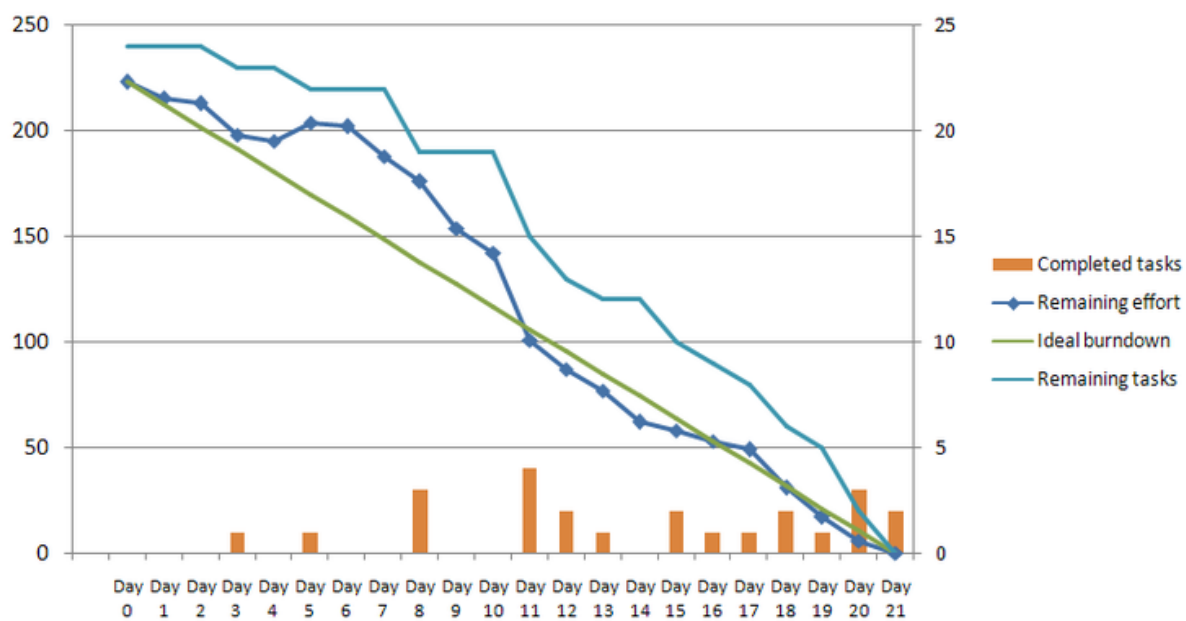
Sprint buffer

43,5 (50%)

#	Naam	Units
3.3.2	Geautomatiseerde regels - Module	14
1.1	API - Google Adwords API	10
3.3.3	Geautomatiseerde regels – Rapportage uitbreiding	6
Totaal		30



9.2 Voorbeeld: sprint burndown





C Plan van aanpak (2)



Inhoudsopgave

1	Aanleiding	84
2	Doelstelling	84
3	Rollen	85
4	Werkwijze	86
5	Afbakening	87
5.1	<i>Afbakening</i>	<i>87</i>
5.2	<i>Randvoorwaarden</i>	<i>87</i>
6	Op te leveren producten	88
6.1	<i>Technisch</i>	<i>88</i>
6.2	<i>Product backlog</i>	<i>88</i>
6.3	<i>Plan van Aanpak</i>	<i>88</i>
6.4	<i>Rapportage</i>	<i>89</i>
6.5	<i>Management</i>	<i>90</i>
7	Risicoanalyse	92
8	Mijlpalen en planning	93
9	Bijlage	94
9.1	<i>Voorbeeld: sprint backlog</i>	<i>94</i>
9.2	<i>Voorbeeld: sprint burndown</i>	<i>95</i>



1 Aanleiding

Search Traffic Monitor is een applicatie ontwikkeld door OrangeValley welke in staat is nieuwe inzichten te generen op het gebied van Search Engine Marketing. Hiervoor combineert zij data van externe bronnen met haar eigen data om zo waardevolle rapporten te genereren. Hetgene wat deze rapporten zo waardevol maakt is dat er naast het bezoekersgedrag ook het gedrag van zoekmachines wordt getoond, een gegeven wat vaak nog mist binnen de analyses voor Search Engine Marketing.

Voor het laden van dergelijke rapporten heeft Search Traffic Monitor momenteel vijf minuten tot enkele uren nodig. Om efficiënt met de applicatie te gaan werken moet deze wachttijd teruggedrongen worden naar een maximum van vijf seconden. Het is daarom noodzakelijk dat de performance van de applicatie wordt herzien en aanzienlijk wordt verbeterd.

Tijdens de brainstormsessie op dinsdag 13 maart 2012 is samen met de opdrachtgevers en de developers besloten dat dit het beste kan worden gedaan door Search Traffic Monitor opnieuw te ontwikkelen. Dit op basis van de voorgedragen oplossingen ten behoeve van de performance.

2 Doelstelling

Aan het eind van dit project moet de vernieuwde versie van Search Traffic Monitor in staat zijn om snel en efficiënt rapporten te genereren voor OrangeValley en haar klanten. Hiervoor beperken we ons tot de hieronder gespecificeerde functionaliteiten. De structuur van de applicatie moet zich op een punt bevinden dat het opzetten van nieuwe testen en rapportages in een 'handomdraai' gaat.

Rapportage

- Bezoekers rapport
- Crawl rapport
- Pagina specifiek rapport
- Zoekterm rapport

Management

- Beheren van leden
- Beheren van profielen
- Beheren van externe diensten
- Beheren van geautomatiseerde regels



3 Rollen

De volgende personen zijn betrokken bij de afstudeeropdracht

<i>Rol</i>	<i>Medewerker</i>	<i>Functie</i>
Stakeholder / Product owner	Eduard Blacquièrre	Consultant zoekmachine optimalisatie
Stakeholder	Ortwin Verreck	Managing partner
ScrumMaster / Developer	Ferdy Perdaan	Technical consultant
Developer	Mark Abspoel	Data architect
Doelgroep	SEM consultants	

Stakeholder

De stakeholders voeden de product backlog met nieuwe gewenste functionaliteiten. Tezamen bepalen zij aan de hand van prioritering de inhoud van de sprints. Binnen de scope van het afstuderen worden echter enkel functionaliteiten met betrekking tot sales module en performance in behandeling genomen.

Product owner

De product owner staat garant voor de belangen van alle stakeholders. Hij is daarmee verantwoordelijk voor het toewijzen van de backlog items aan de verschillende sprints. Hiervoor wordt aan elk backlog item een prioriteit toegekend op gebaseerde op de bedrijfswaarde afgewogen tegen de tijd die ervoor nodig is.

ScrumMaster

De ScrumMaster wijst alle sprint items aan de projectleden. Het is aan de ScrumMaster om het overzicht te houden over de voortgang van de sprints en het algehele project. Tevens is het de taak van de ScrumMaster om alle Scrum documentatie op te stellen.

Developer

De developers zijn verantwoordelijk voor de realisatie van de gewenste functionaliteiten die voor een sprint ingepland staan. Wanneer deze planning niet gehaald wordt is het aan de ScrumMaster om te verantwoorden wat de reden daarvan is.

Doelgroep

Omdat Search Traffic Monitor een applicatie is welke toegevoegd moet worden aan de standaard 'gereedschapskist' van de consultants worden deze ter zijner tijd gevraagd om mee te helpen met het testen van de nieuwe functionaliteiten. Tevens is de doelgroep in staat om sprint items toe te voegen aan de product backlog, het is echter aan de product owner en de developers om aan te geven of deze sprint items bijdragen aan het doel van Search Traffic Monitor.



4 Werkwijze

Op 15 maart 2012 hebben de stakeholders in het kader van de hoeveelheid aanpassingen resulterend uit het performance besloten dat het verstandiger is om een nieuwe applicatie te ontwikkelen. In plaats van nu een nieuwe salesmodule te ontwikkelen worden de resterende tien weken gebruikt voor het bouwen van een nieuwe applicatie. De product backlog wordt in de nieuwe situatie dan ook aangevuld op basis van de vernieuwde doelstelling.

Omdat Search Traffic Monitor te groot is om binnen tien weken volledig te herbouwen is er samen met de stakeholders een nieuwe doelstelling gedefinieerd. Hierbij is besloten om de belangrijkste rapporten 'na' te bouwen zodat de nieuwe versie vooralsnog inzichten vertoont betreffende crawling, ranking en traffic in het STRICA traject.

Rapportage

- Bezoekersrapport
- Zoektermrapport
- Zoekmachinerapport
- Filters

Management

- Beheren van accounts
- Beheren van verwerkingsproces
- Beheren van profielen
- Beheren van connectie met externe diensten
- Beheren van geautomatiseerde regels - *optioneel*

Uit het adviesrapport bleek dat de graphical user interface, ofwel layout, mede verantwoordelijk is voor de performanceproblemen binnen Search Traffic Monitor. Voordat de bovenstaande requirements kunnen worden uitgewerkt is het dan ook noodzakelijk dat er een nieuwe layout ontworpen wordt. Hiervoor worden meerdere stadia doorlopen om te testen of de layout en interactie met de applicatie conform de wensen en eisen van de product owner zijn voor we een permanente architectuur opzetten.

Stadia

- Wireframes
- Mockup
- Technisch ontwerp
- Migratie
- Prototype



5 Afbakening

5.1 Afbakening

- Focus op interface

De interface is slechts een subonderdeel van Search Traffic Monitor. De subonderdelen verantwoordelijk voor de verwerking van de data kunnen in een later stadium worden geüpdatet opdat ook deze samenwerken met de nieuwe database structuur. Dit om er zeker van te zijn dat de structuur van de database permanent is.

- Focus op vier rapporten

Gedurende de ontwikkeling van het systeem ligt de focus op de in hoofdstuk 2 gedefinieerde rapporten. Het is belangrijk dat deze rapporten gegenereerd worden aan het eind van het project, ongeacht de andere rapporten welke op de product backlog worden gedefinieerd.

- Framework buiten beschouwing

Het nieuwe systeem wordt geschreven als uitbreiding op het reeds bestaande Pie Framework. Hiervoor geldt dat er enkel ontwerpen en testcases worden betreffende onze extensies op dit framework.

- Bestaande onderdelen buiten beschouwing

Reeds bestaande onderdelen van Search Traffic Monitor welke worden hergebruikt voor het nieuwe systeem worden buiten beschouwing gelaten, hier worden geen ontwerpen dan wel unit tests voor geschreven.

5.2 Randvoorwaarden

De afstudeerder

- De afstudeerder is vijf dagen per week op de bedrijfslocatie aanwezig.
- Gedurende de afstudeerperiode van 6 februari tot 1 juni beperkt de afstudeerder zijn werkzaamheden tot de werkzaamheden besproken binnen dit plan van aanpak.

OrangeValley

- OrangeValley geeft de afstudeerder de gelegenheid om aan het afstudeerdossier te werken, hier wordt wekelijks één dag voor ingepland. Indien nodig kan er bij de opdrachtgever een verzoek worden ingediend voor extra dagen.



6 Op te leveren producten

6.1 Technisch

Er moet een technisch ontwerp beschikbaar zijn waarin zowel de architectuur van de database als de architectuur van de code zichtbaar is. Dit ontwerp wordt gemaakt in de publiekelijk geaccepteerde vorm UML. Indien nodig worden deze ontwerpen aangevuld met use-case diagrammen en sequentiediagrammen om zo de interactie met het systeem te verduidelijken.

6.2 Product backlog

De product backlog is een lijst waarop alle gewenste functionaliteiten staan vermeld. De backlog items zijn in kleine activiteiten opgedeeld en hiërarchisch ingedeeld opdat ze gemakkelijk in sprint kunnen worden ingedeeld.

6.3 Plan van Aanpak¹

In het plan van aanpak wordt de aanpak voor het bouwen van de vernieuwde Search Traffic Monitor beschreven, hieronder valt een beschrijving van de activiteiten, de werkwijze, de methodieken, de risico's en de mijlpalen. Het doel van het plan van aanpak is om de opdrachtgever en de projectleden een duidelijk beeld te geven over hun betrokkenheid, werkzaamheden en verwachtingen van het project.

¹ Dit document is het plan van aanpak in kwestie



6.4 Rapportage

De rapporten zijn gebaseerd op de huidige Search Traffic Monitor, hiervan zijn de meest waardevolle rapportages meegenomen om zo veel mogelijk inzicht te verschaffen in het STRICA traject. Middels het gebruik van filters is de gebruiker vervolgens in staat om bepaalde varianten van de rapportages te genereren.

Bezoekersrapport

Het bezoekersrapport toont hoeveel bezoekers er per pad binnen zijn gekomen. Indien de gebruiker op een pad klikt gaat deze naar een detailrapport welke alle zoektermen toont waarmee men op dat pad binnen is gekomen.

Zoektermrapport

Het zoektermrapport toont de regelmaat waarmee bezoekers binnen komen via een bepaalde zoekterm. Indien de data bekend is wordt in dit rapport ook de gemiddelde positie getoond waarmee de bezoeker via de zoekterm binnen komt.

Zoekmachinerapport

Het zoekmachine rapport is het primaire rapport dat gebruikt wordt om inzicht te verschaffen op het gebied van zoekmachinegedrag. Dit rapport toont de hoeveelheid en de regelmaat waarmee de zoekmachines een bepaald pad crawlen.

Filters

Filters worden gebruikt om de inhoud van de rapportages te beïnvloeden. Met behulp van de filters kan een rapport vanuit meerdere perspectieven worden bekeken opdat deze een multifunctioneel doel krijgen. Zo is het bijvoorbeeld mogelijk om een bezoekersrapport rapport te filteren op een verkeersbron. Aan de hand van deze filter kan het zoekverkeer vanuit Google vergeleken worden met het zoekverkeer vanuit Yahoo, wat waardevolle inzichten kan geven over het type bezoekers dat binnen komt via de verschillende zoekmachines.

Bij het bouwen van de filters beperken we ons voor nu tot de filters die ook in de huidige Search Traffic Monitor worden aangeboden. Daarnaast implementeren we ook een drempelwaarde filter die alle rijen onder de gedefinieerde drempel verbergt, dit om minder interessante data uit de rapporten weg te filteren.

Filters

- Datum
- Pad
- Statuscode
- Zoekterm
- Zoekmachine
- Verkeersbron
- Drempelwaarde



6.5 Management

Beheren van accounts

Search Traffic Monitor is een applicatie die veel klantgevoelige data bevat, het is dan ook van belang dat de applicatie goed beveiligd wordt en dat klanten enkel hun eigen rapportages in kunnen zien. Daarom wordt er gebruik gemaakt van een ledensysteem waarmee alle data afgeschermd wordt.

Binnen het ledensysteem moet er rekening gehouden worden met verschillende rollen. Een klant mag enkel haar eigen data inzien in tegenstelling tot OrangeValley consultants die toegang hebben tot alle data.

Beheren van verwerkingsproces

Search Traffic Monitor geraakt aan haar data middels logfileanalyse. Deze logfileanalyse vind op een ander onderdeel en een andere server(s) binnen de architectuur plaats, om dit proces te overzien is daarom een inzicht in het verwerkingsproces wenselijk.

In dit overzicht specificeren we ons tot twee onderdelen units en de taken. Units zijn subonderdelen in het verwerkingsproces, dit zijn stukken software. Taken zijn feitelijk de logregels over de stukjes data die door de units worden verwerkt.

Taken schetsen een historisch profiel over de performance van een unit, omdat deze logregels representeren is het daarom niet wenselijk dat taken aangepast/toegevoegd of verwijderd worden via het beheer. Het beheer specificeert zich dan ook enkel tot een lijst weergave waarin alle taken worden weergegeven.

De units moeten echter volledig beheerd kunnen worden via het management. Men moet in staat zijn om te zien hoeveel taken actief zijn op de specifieke unit, indien dit aantal te hoog is moet de gebruiker in staat zijn om nieuwe units toe te voegen via het beheer. Dit wil zeggen dat er een connectie naar een externe server moet worden opgezet waarop de installatie van een nieuwe unit plaats vind. De aanschaf en configuratie van een nieuwe server wordt hierbij buiten beschouwing gelaten, dit is in verband met de gehanteerde connectiemethodiek niet mogelijk. Naast het installeren moet het ook mogelijk zijn om units te deactiveren, herinstalleren en deïnstalleren.

Beheren van profielen

Elk profiel representeert een website waarop Search Traffic Monitor gebruikt wordt. Een klant moet namelijk in staat zijn om meerdere profielen te beheren. Middels het profielenbeheer worden de profielen volledig geconfigureerd.

Aan de hand van het profiel de gebruiker aan welke databaseconnectie er gebruikt moet worden, wat voor tracking type er wordt gebruikt, hoe de data moet worden opgeschoond en de koppelingen met externe diensten. In dit beheer moet echter een onderscheid worden gemaakt tussen de verschillende rollen, zo mag een klant enkel haar externe diensten aanpassen. Het is dan ook niet mogelijk voor klanten om een nieuw profiel aan te maken, dit recht is behouden tot de OrangeValley consultants.



Beheren van connectie met externe diensten

Search Traffic Monitor is een applicatie die graag van externe diensten gebruik maakt om nog meer inzichten te kunnen genereren. Zo kon Search Traffic Monitor in het verleden een koppeling leggen met diensten als Google Analytics, Google Webmaster, Google AdWords, Pingdom en SiteSpect. In verband met het korte tijdsbestek beperken we dit tot slechts drie diensten, zijnde Google Analytics, SiteSpect en een nieuwe generieke FTP verbinding welke nodig is voor het downloaden van logfiles bij klanten.

Diensten beperken zich tot de primaire inlog van een externe dienst, het profiel gaat hier vervolgens dieper op in door een profiel/account binnen de dienst te selecteren. Zo wordt er in het profiel een Google Analytics dienst geselecteerd waarna de accounts en profielen van de geselecteerde dienst selecteert. Het gemak hiervan is dat de gebruiker de inlog van al haar connecties vanuit één punt kan beheren en deze niet overal opnieuw in hoeft te vullen.

Waar mogelijk wordt er gebruik gemaakt van een OAuth connectie in verband met de veiligheid. Normaliter gebruikt men een login en wachtwoord om in te loggen bij een externe dienst, OAuth verschilt hierin door gebruik te maken van een gegeneerde token¹. Het voordeel van een token is dat ook werkt nadat de gebruiker haar login of wachtwoord veranderd, hiermee is de connectie aanzienlijk stabiel. Tevens geeft het gebruik van tokens het voordeel dat de gebruiker haar wachtwoord niet aan ons hoeft te verstrekken. Indien een token uitlekt heeft de hacker hier niets aan, een token is namelijk enkel geldig vanaf ons domein in tegenstelling tot een login/wachtwoord combinatie.

¹ Domein gebonden unieke sleutel die de login van een gebruiker representeert



7 Risicoanalyse

- **Behalen competenties niet aantoonbaar**

Omdat er twee developers op dit project actief zijn is het moeilijk aantoonbaar of de afstudeerder de vooraf gedefinieerde competenties heeft behaald. Dit probleem kan aangetoond worden door middel van SVN, hierin is zichtbaar wie welke wijziging heeft gemaakt tijdens het opslaan van bestanden.

- **Operationeel werk**

Naast het meedenken met collega's en het meehelpen aan oplossen van kleine problemen is het niet de bedoeling dat de afstudeerder operationeel werk verricht, dat wil zeggen dat de afstudeerder volledig ingepland staat voor zijn studie en voor Search Traffic Monitor. Om dit risico te voorkomen zijn er afspraken gemaakt met het bedrijf en is er aan de bedrijfsmentor en de projectmanager gevraagd om hierop te letten.

- **Minder 'zichtbare' resultaten**

In het normale ontwikkeltraject binnen OrangeValley is het ongebruikelijk dat er ook unit tests en uml diagrammen worden opgeleverd, deze bieden namelijk geen zichtbare toegevoegde waarde voor de klant. In het kader van het afstuderen wordt hier een uitzondering op gemaakt, dit betekent dat een deel van de normaal gesproken bouwtijd 'verloren' gaat aan het maken van ontwerpen en testen. Verder is er naast het werken aan de gedefinieerde opdracht ook tijd gereserveerd voor het schrijven van het afstudeerdossier.

Het management is hierover ingelicht en toont zich hier bewust van te zijn. Het is de taak aan de afstudeerder en de bedrijfsmentor om tijdig in te zien wanneer de verwachtingen van de opdrachtgever(s) te hoog liggen opdat hier nog tijdig over gesproken kan worden.

- **Prototype wordt permanent**

Het bouwen van een prototype is een prima principe om kinderfouten in het ontwerp te ontdekken, het schept echter bij de opdrachtgever een beeld dat het product al af is en klaar voor gebruik. Een reëel risico is dan ook dat het prototype langzaam aan de echte applicatie gaat worden in verband met de verwachtingen die ermee geschept worden. Dit kan vermeden worden door de opdrachtgever erg duidelijk te maken dat een prototype nog lang niet het uiteindelijke product is.



8 Mijlpalen en planning

Mijlpaal	Datum
Start afstuderen	6 februari
<i>Plan van aanpak compleet</i>	15 maart
<i>Wireframes opgesteld</i>	16 maart
<i>Prototype compleet</i>	21 maart
<i>Product backlog opgesteld</i>	22 maart
<i>Systeemontwerp opgesteld</i>	27 maart
<i>Migratiemodel opgesteld</i>	30 maart
<i>Afronding sprint 1</i>	1 april
<i>Afronding sprint 2</i>	13 april
<i>Afronding sprint 3</i>	27 april
<i>Afronding sprint 4</i>	11 mei
<i>Afronding sprint 5</i>	1 juni
Eind afstuderen	1 juni



9 Bijlage

9.1 Voorbeeld: sprint backlog

Periode	22 december 2010 - 5 januari 2011
Tijd per unit	30 Minuten
Units beschikbaar	87
Opdrachtgever	Eduard Blacquièrre
Projectleden	Ferdy Perdaan

Sprint items

#	Naam	Units
6.2.4 - 4	STM – Url Rapport - Crawlrate	2
6.2.4 - 6	STM – Url Rapport - Filter	2
8.2	Dashboard login / integratie ZF Authentication	8
8.3	Wisselen van klantenprofiel voor OV medewerkers	2
1.5	API - Module	20
1.2	API - Google Analytics API	10
8.4	Integratie SEO null meting	14
6.2.7	STM - Zoekmachine rapport	8
8.1	Opzetten dev omgeving voor het dashboard	8
Totaal		74

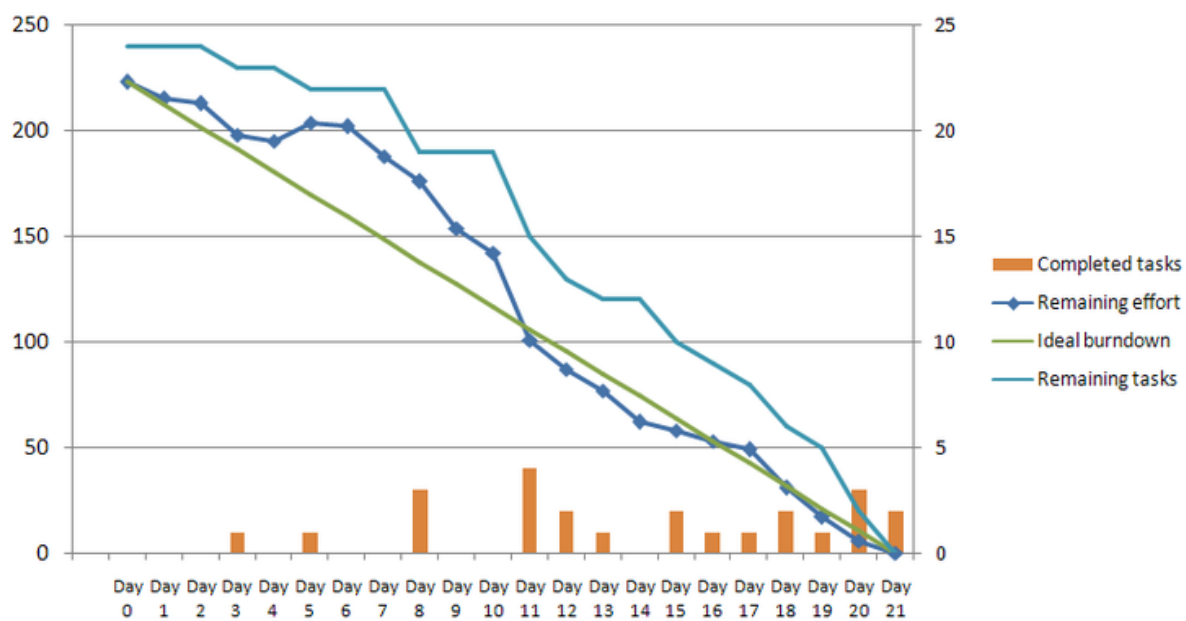
Sprint buffer

43,5 (50%)

#	Naam	Units
3.3.2	Geautomatiseerde regels - Module	14
1.1	API - Google Adwords API	10
3.3.3	Geautomatiseerde regels – Rapportage uitbreiding	6
Totaal		30



9.2 Voorbeeld: sprint burndown





D Technisch ontwerp

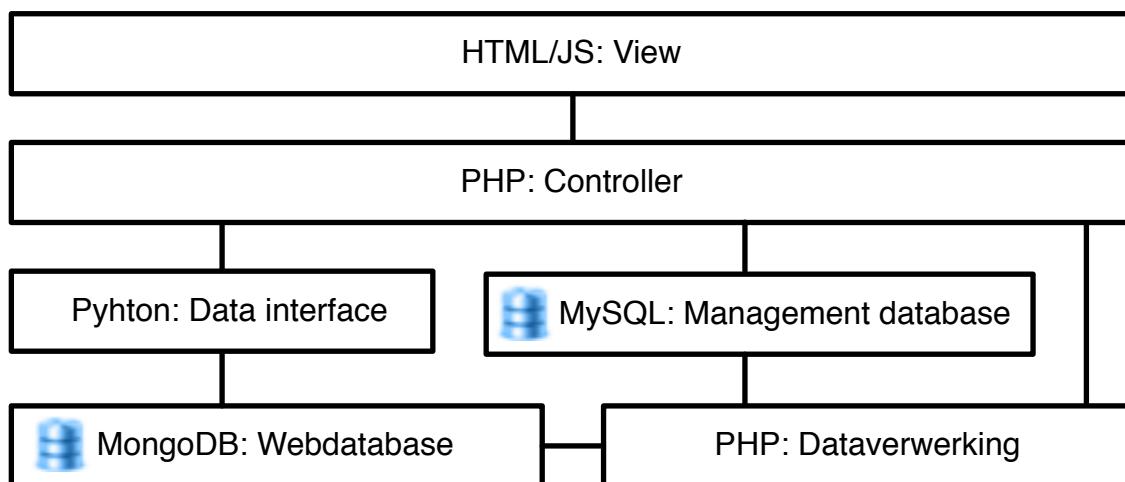


Inhoudsopgave

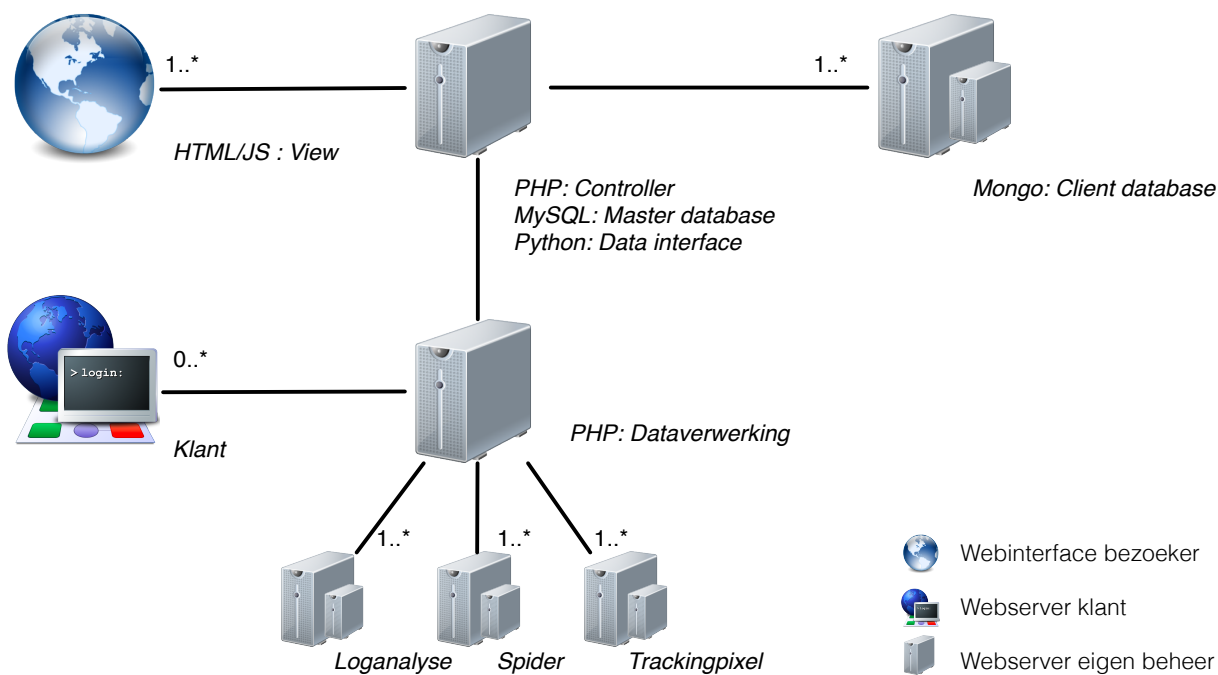
1	Architectuur	98
2	EER	99
2.1	<i>Web</i>	99
2.2	<i>Management</i>	100
3	Klassendiagram	101
3.1	<i>HTML/JS: View</i>	101
3.2	<i>PHP: Controller</i>	102
3.3	<i>Python: Data interface</i>	106
4	Sequentiediagram	107
4.1	<i>Wisselen rapportage</i>	107
4.2	<i>Ophalen dataset</i>	108
5	Flowchart	110
5.1	<i>View: ophalen dataset</i>	110
5.2	<i>Data interface: ophalen dataset</i>	112
5.3	<i>Globale testcases</i>	114
6	Wireframes	116
6.1	<i>Zoekmachine rapport</i>	116
6.2	<i>Bezoekers rapport</i>	117
6.3	<i>Zoekterm rapport</i>	118
6.4	<i>Management lijst</i>	119
6.5	<i>Management gedetailleerd</i>	120
6.6	<i>Geautomatiseerde regels</i>	121
7	Realisatie	122
7.1	<i>Toevoegen nieuwe entiteit</i>	122
7.2	<i>Toevoegen nieuwe rapportage</i>	129



1 Architectuur



Architecturale opbouw van Search Traffic Monitor

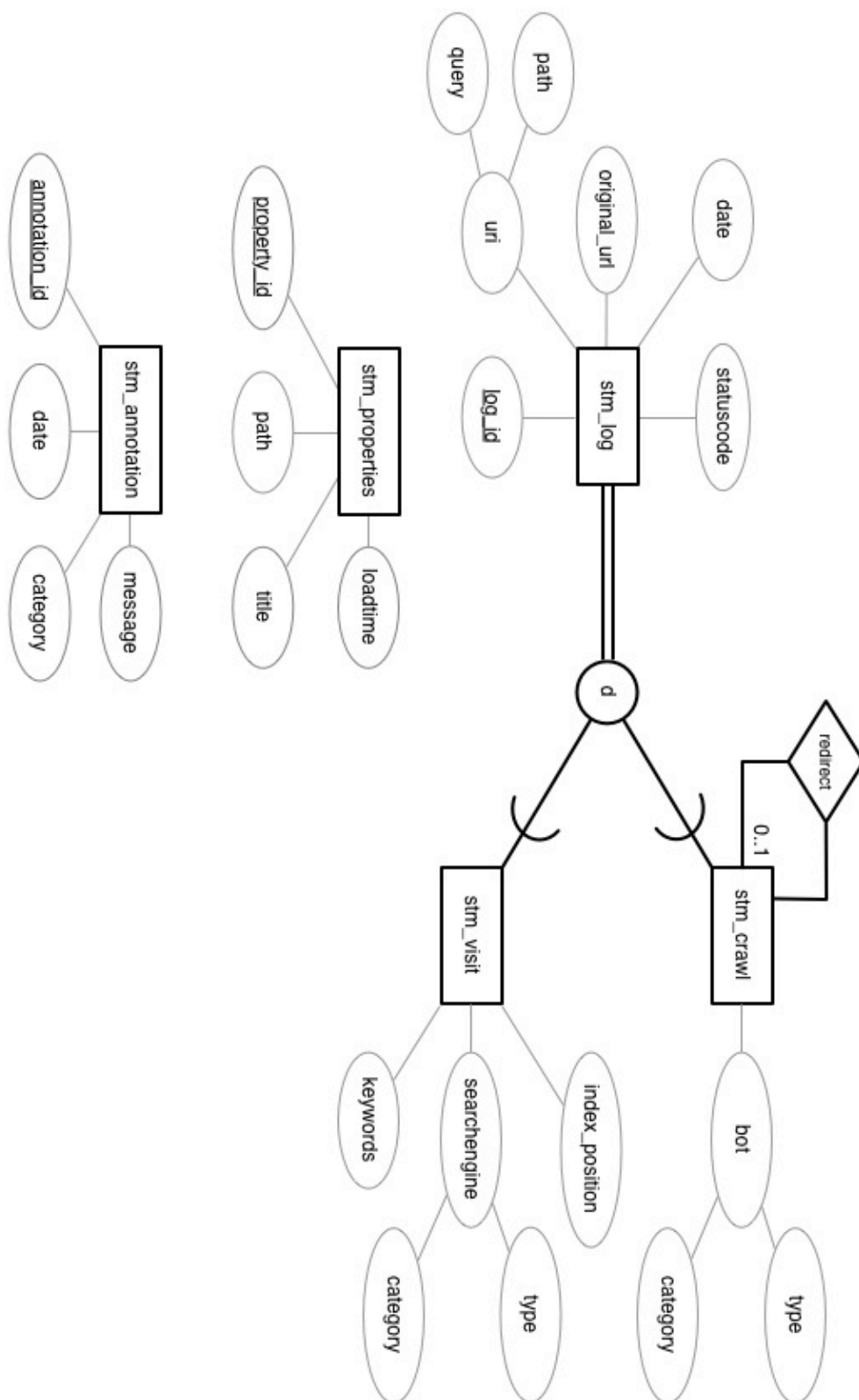


Architectuur uitgedrukt in servers



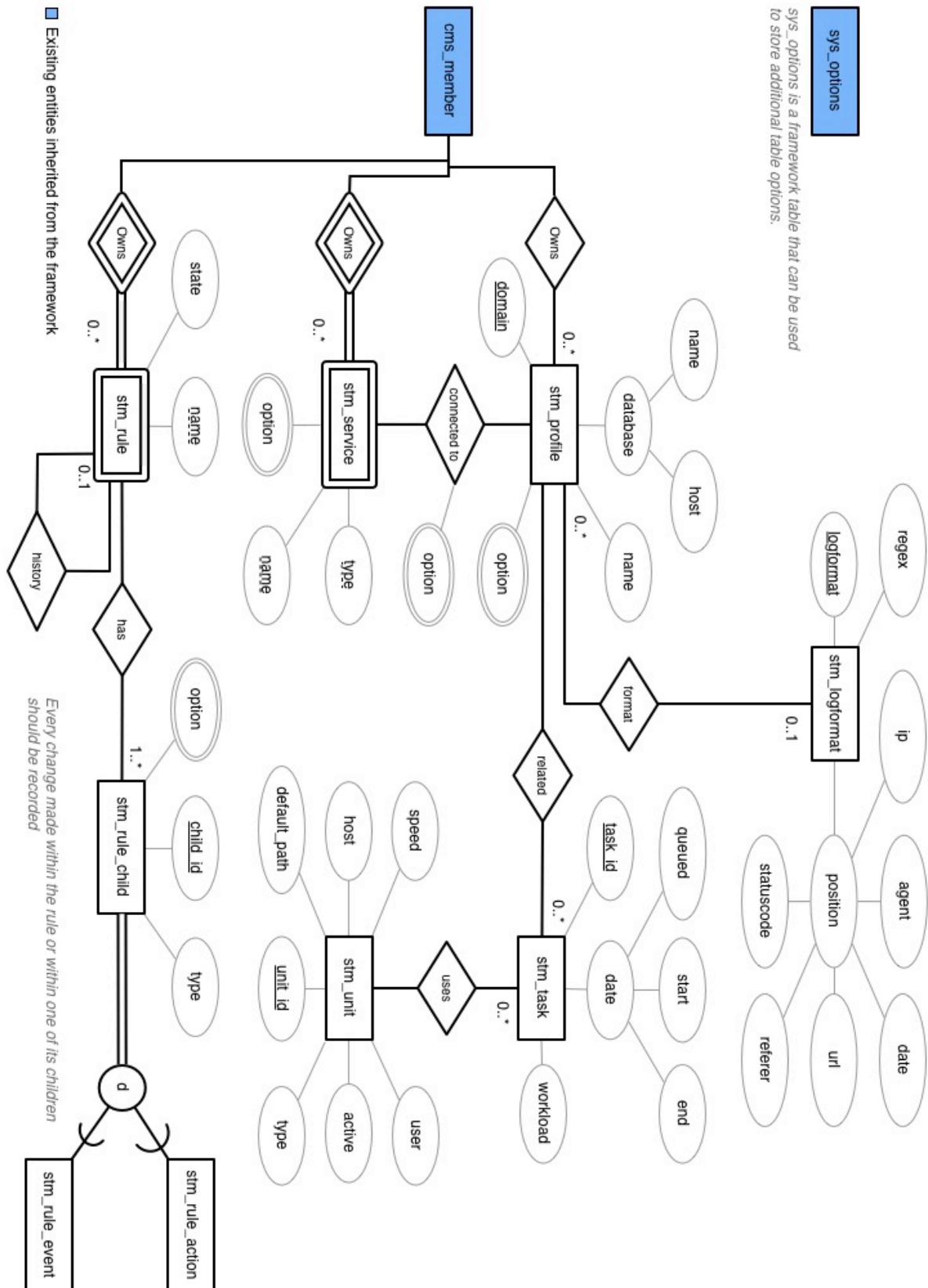
2 EER

2.1 Web





2.2 Management





3 Klassendiagram

3.1 HTML/JS: View

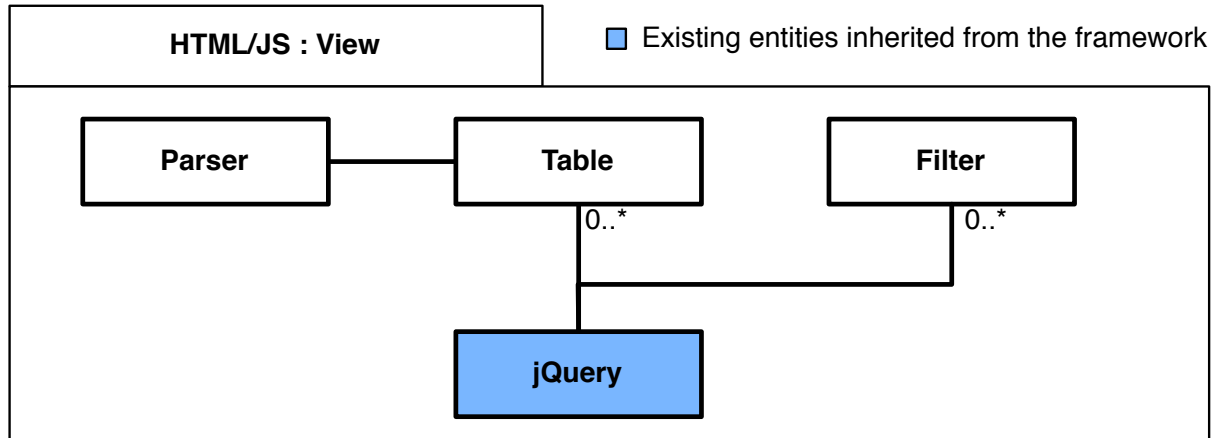


Table
+ \$element : json + option : json + colspan : int + rows : json[total: int=0, offset: int=0, limit: int=0] + loading : bool = false + hash : string + __construct(element : string json, options : json null) + fetch(force: bool = false, partial: bool = false) + _hash(params : json null) : string + hide() + page(number : int null, append : int = 0) + _params() : json + renderRows(rows : array) + _renderCols(cols : array) + _renderNavigation() + show()

Filter
+ list : json + \$element : json + option : json + __construct(element : string json, options : json null) + hide() + params(params : json) + show() + store() + value() : var

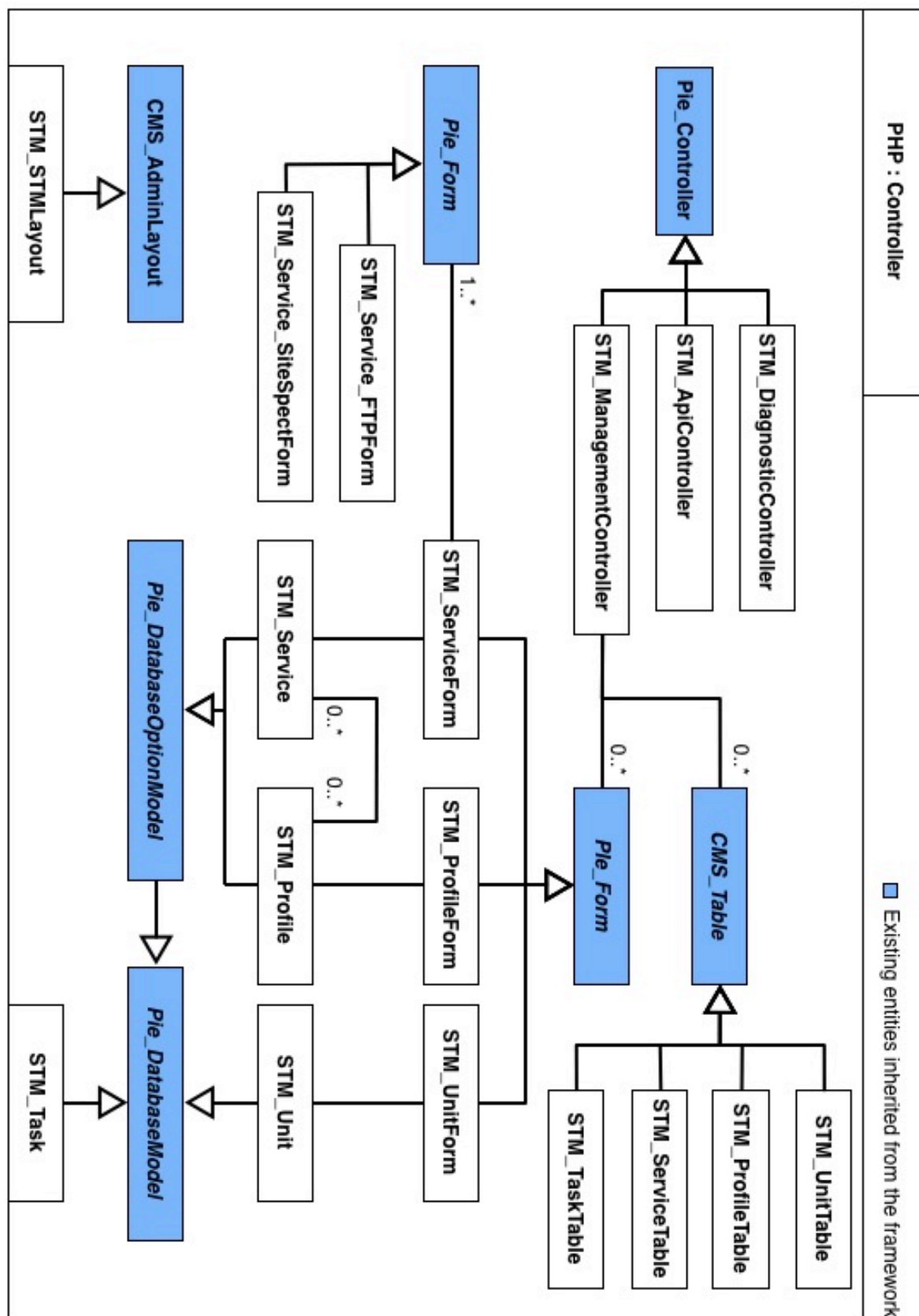
jQuery
<u>\$.eachFilter(filters : array, method : string, arg : array null) : array</u> <u>\$.showActiveFilter()</u> \$.fn.stmTable(option : string json, arg : array null) : json \$.fn.stmFilter(option : string json, arg : array null) : json

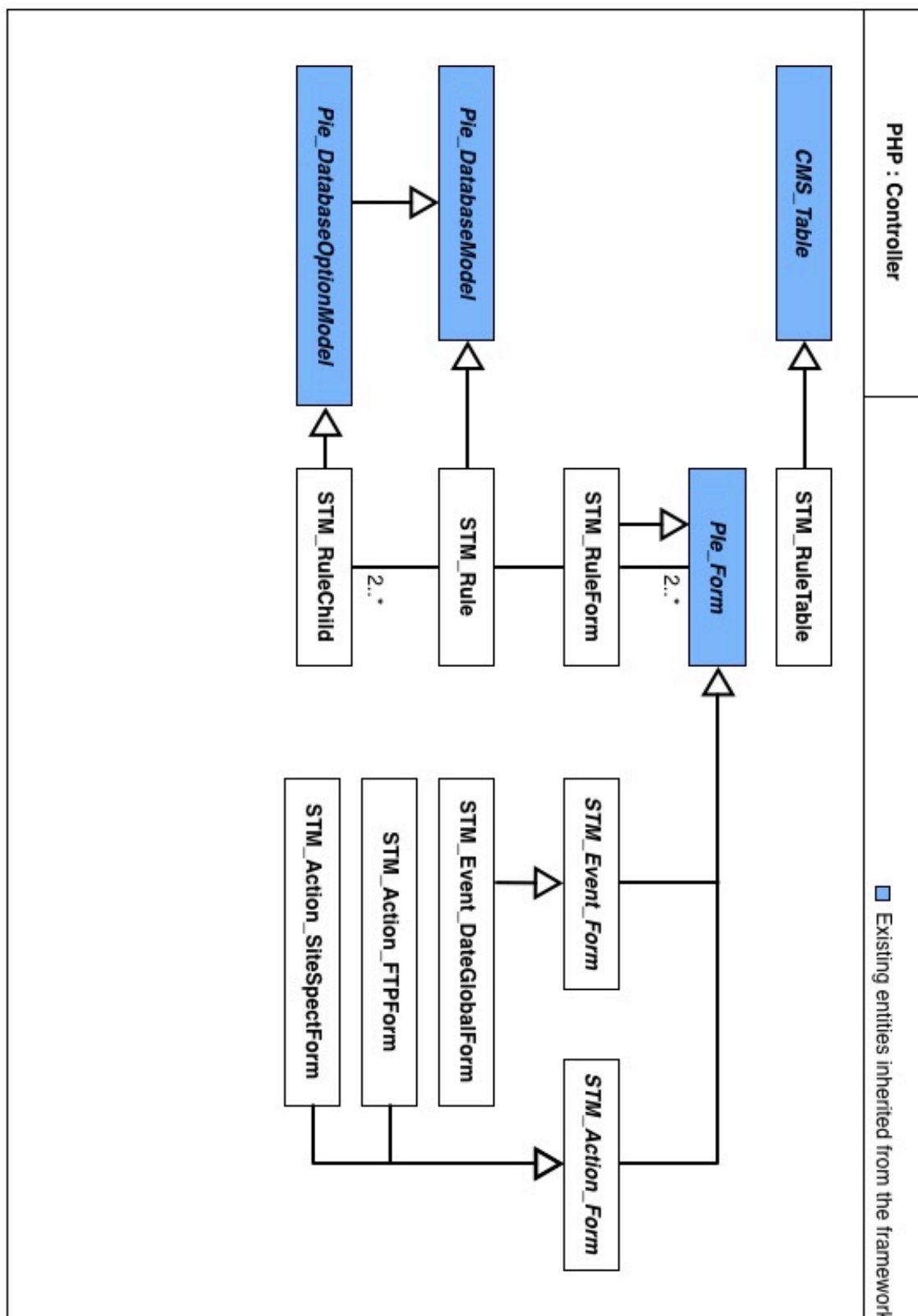
Parser
+ parseUrl(url : string, length : int null) : string + parseKeyword(keyword : string, length : int null) : string + roundNumber(number : float, decimals : int) : float + formatNumber(number : int) : string + compareClass(prev : int, cur : int) : string + substr(val : string, length : int null) : string

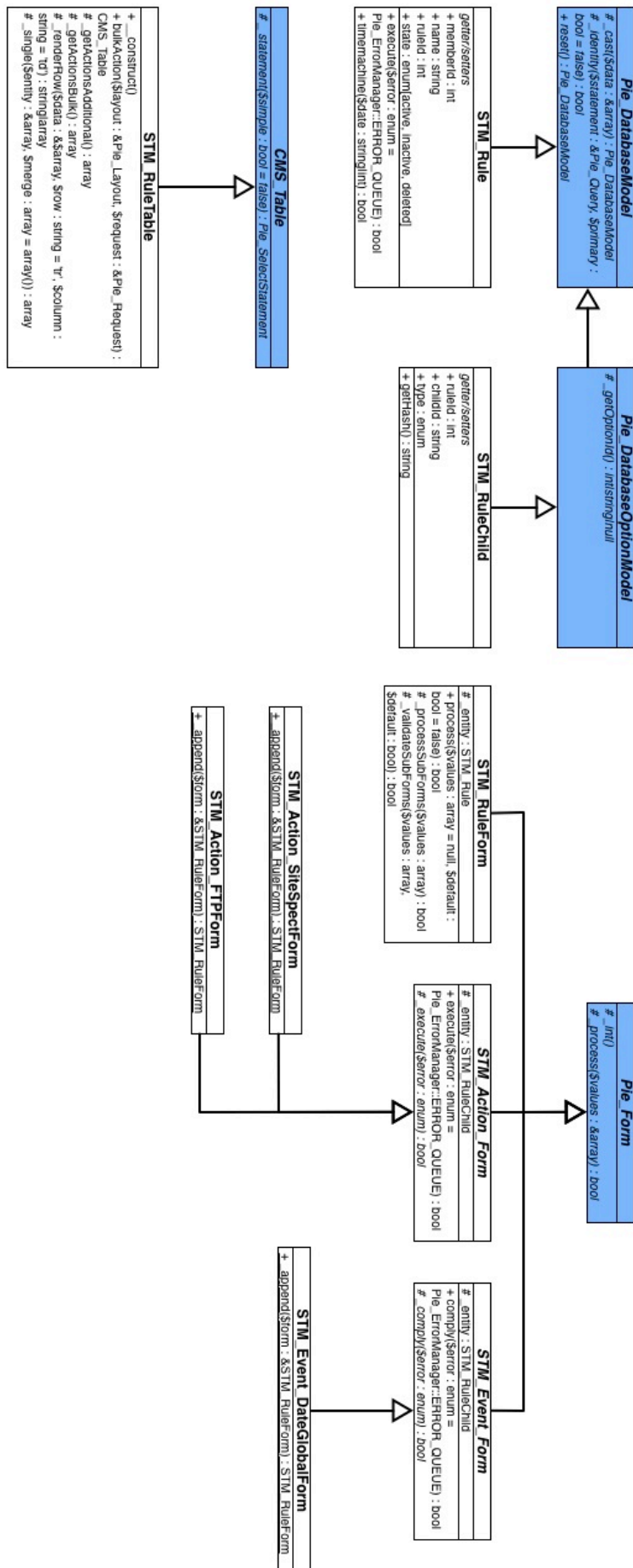
jQuery is extended to preform as access panel for our classes



3.2 PHP: Controller

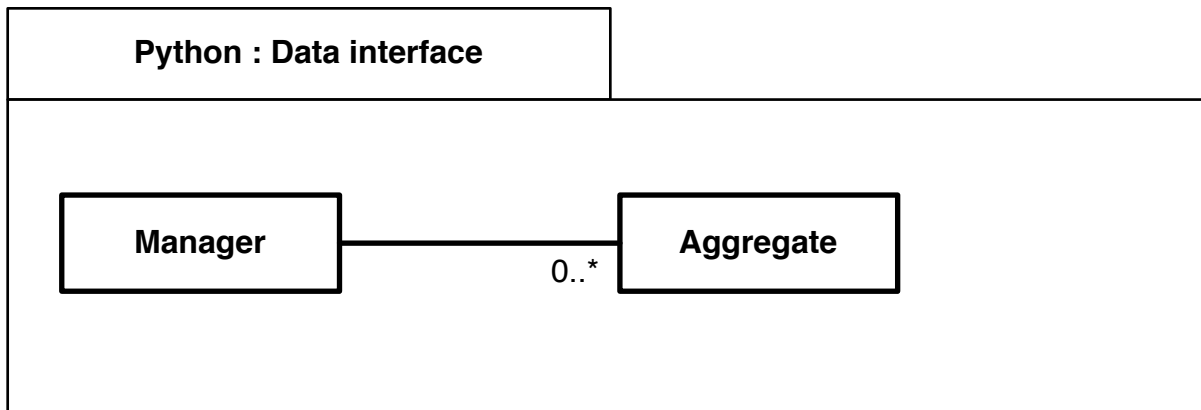








3.3 Python: Data interface



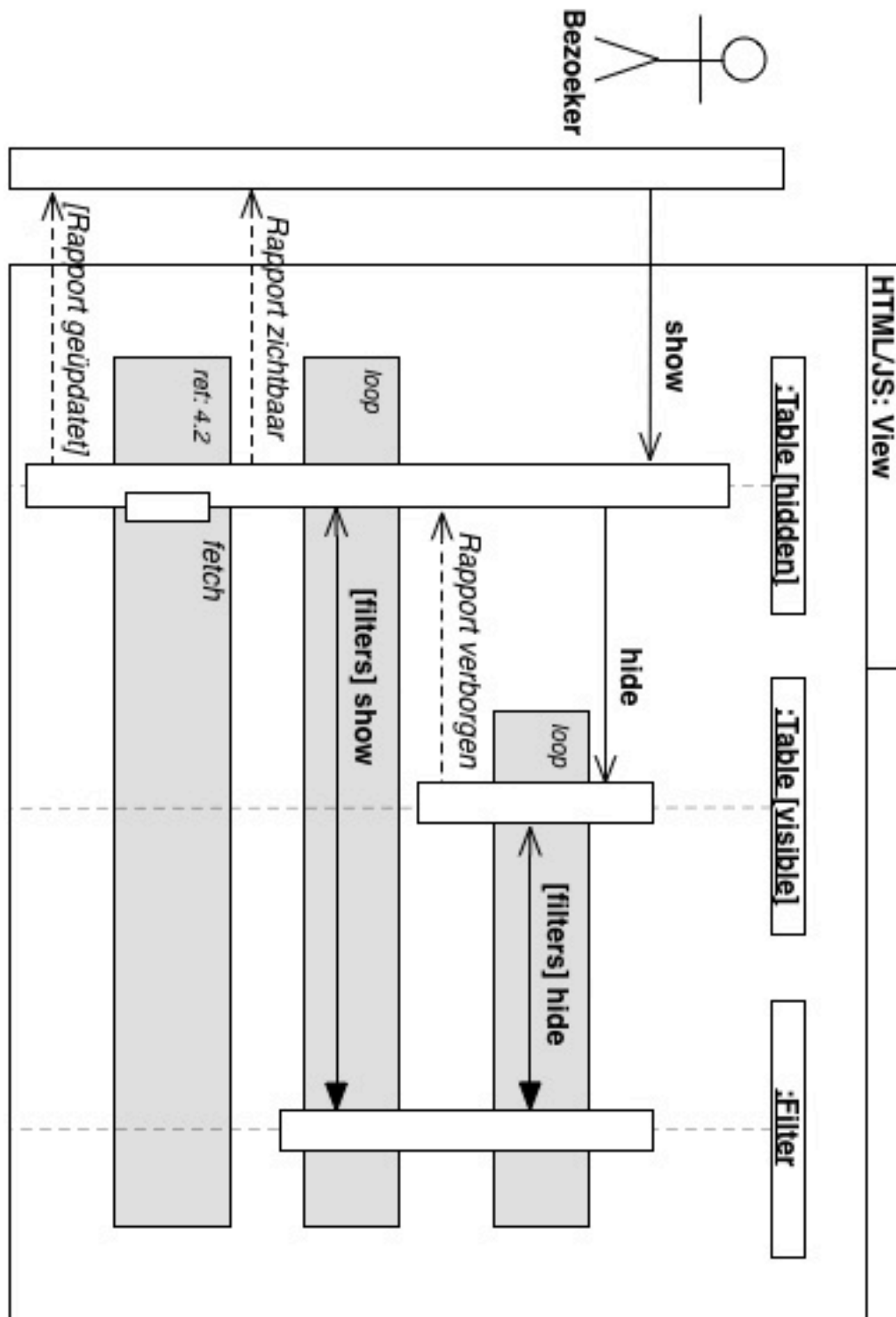
Manager
+ db : string + all_aggs : dict + fixed_aggs : set + __init__(db : string) + agg_info(agg : string, create : bool = True) : dict + agg_list(selector: enum) : dict + cleanup() + clear_aggregates() + col_name(agg : string, create : bool = True) : string + compute_aggregate(node : string, base : string = None, out : Function = print_func) + compute_aggs(update_aggs : bool, base_aggs : dict, force : bool = False, out : function = null_func, dry_run : bool = False, progress : bool = False, extra_fields : bool = False) + computed_col(agg : string, super_ok : bool = True, out : function = null_func) + count(agg : string, filters : array = None) : int + _get_all_aggs : array + import_agg(agg : set) + last_updated_agg(agg : string) : DatelNone + last_updated_data_sets(data_sets : string) : DatelNone + list_aggregates() + query(agg : string, filters : array = None, sort : string : None, limit : int = None, skip : int = None, two_tier : bool = False) : dict + total(agg : string, filters : array = None) : dict + update_info(info : dict)

Aggregate
+ data_sets : dict + entry : set + base_fields : set + filters : dict + name : string + extra_fields : set + static : bool + virtual : bool + __str__() : string + __repr__() : string + copy() : Aggregate + dict_representataion() : dict + distance(base : dict, debug : Function = null_func) : int + filter_object(base : dict) : dict + finalize_function(base : dict = None) : string + map_function(base : dict = None, extra_fields : bool = False) : string + post_filter(filter : dict = {}, sort : string) : dict + prepare(base : dict = None) + reduce_function(base : dict = None, extra-fields : bool = False) : string + scope_object() : dictlNone + sort_object(sort : string) : string + trending() : bool + update_values()



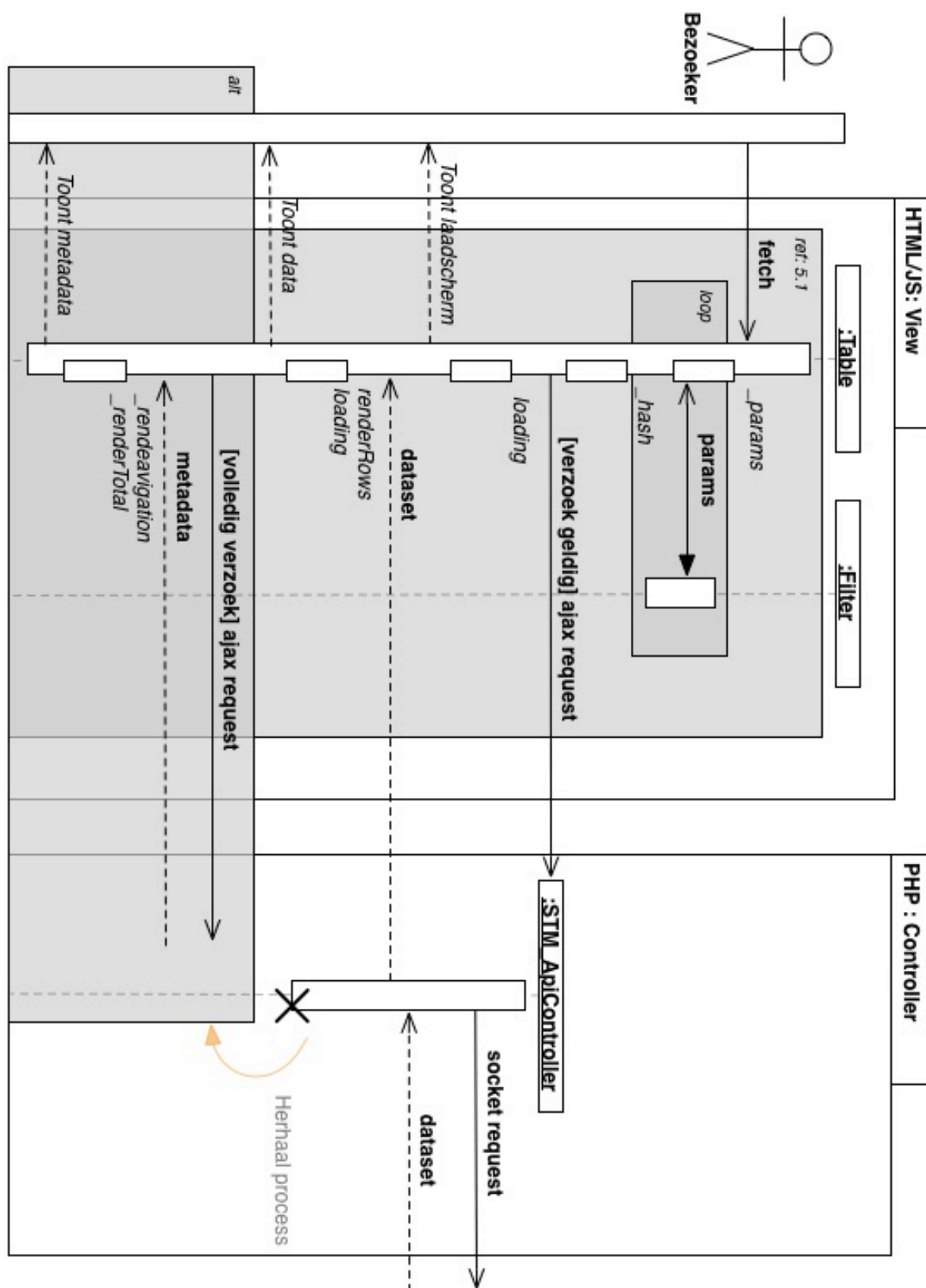
4 Sequentiedigram

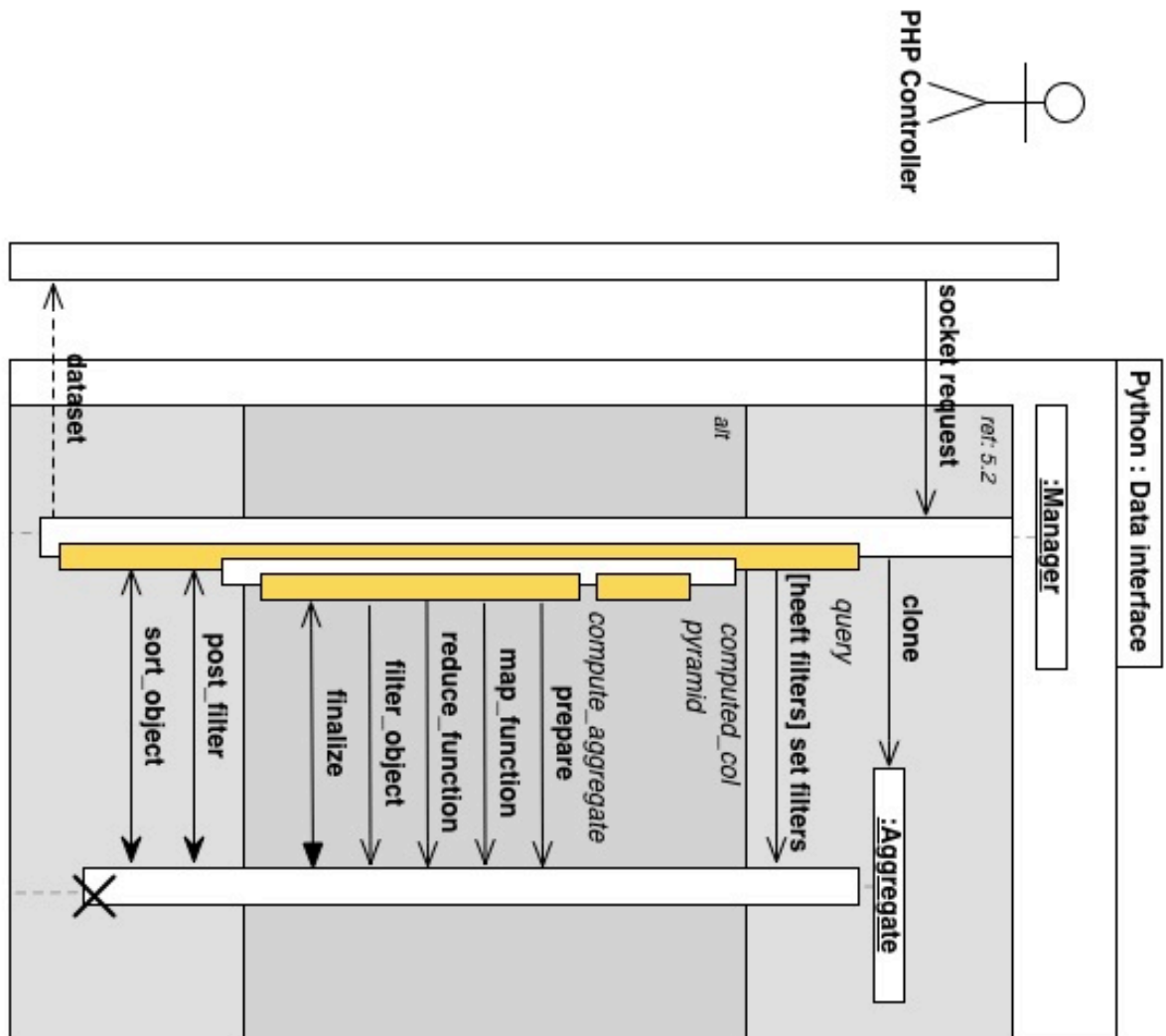
4.1 Wisselen rapportage





4.2 Ophalen dataset

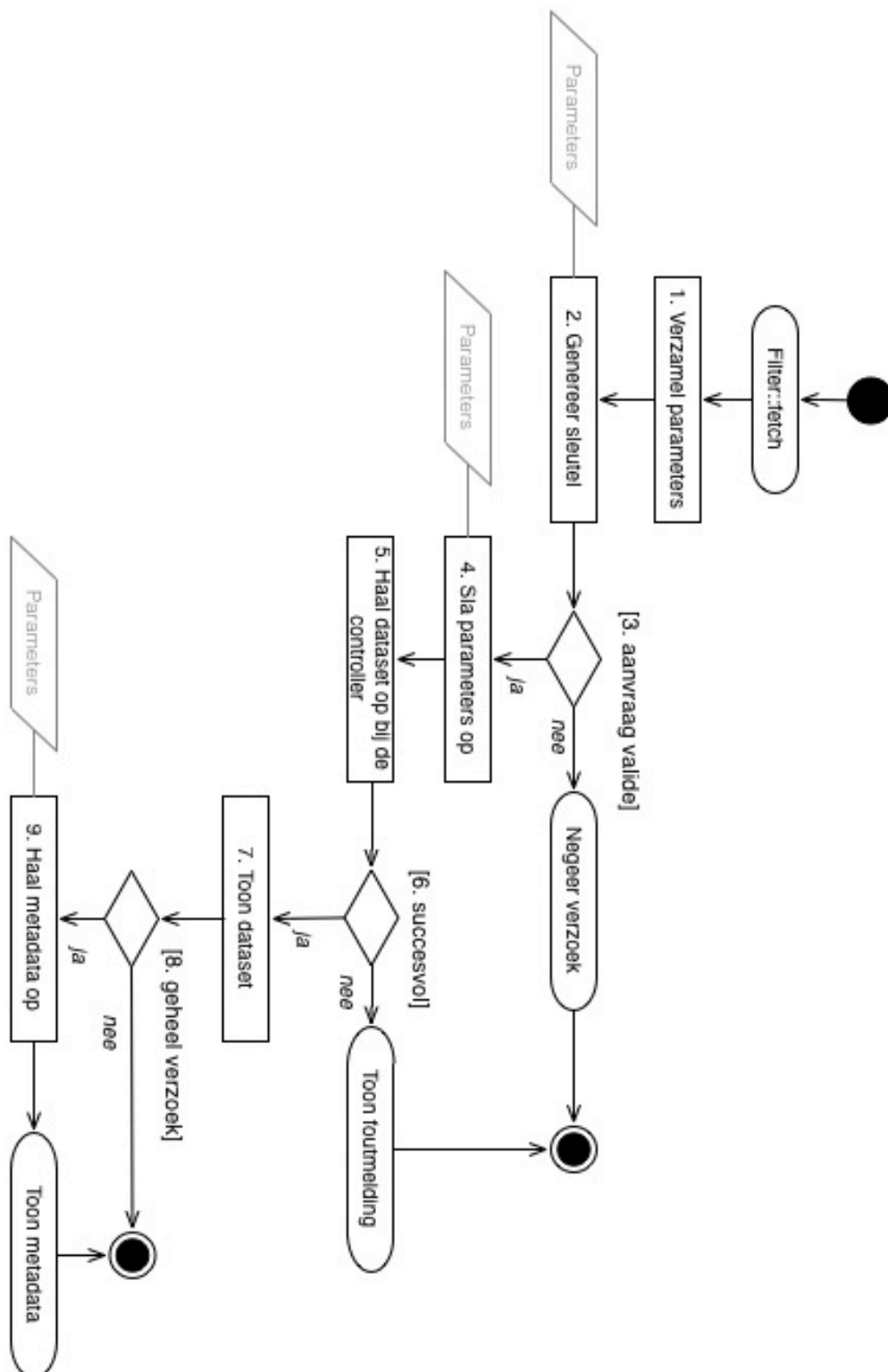






5 Flowchart

5.1 View: ophalen dataset





1. Verzamel parameters

In deze stap itereert de tabel over al haar filters heen om zo een lijst met parameters te genereren. Elke filter voegt haar eigen parameters aan deze lijst toe zodra deze als een valide waarde worden geacht.

2. Genereer sleutel

Aan de hand van de parameterlijst wordt een sleutel gegenereerd die aangeeft of de tabel gewijzigd is ten opzichte van de voorgaande aanvraag. Deze check wordt uitgevoerd om te voorkomen dat de data niet onnodig wordt opgehaald.

3. [Aanvraag valide]

Zoals in stap 2 is aangegeven wordt er sleutel gegenereerd om te kijken of de dataset ververst moet worden. Daarnaast zijn er nog twee manieren waarbij een aanvraag als valide wordt gemarkeerd. Pas zodra er de tabel in bezit is van een dataset wordt er gekeken naar de sleutel, de eerste aanvraag is hierdoor altijd geldig. Verder heeft de developer de mogelijkheid om de aanvraag te forceren indien gewenst.

4. Sla parameters op

Om het gebruikersgemak van de applicatie te verbeteren worden de parameters opgeslagen. Bij het herladen van een pagina worden de filters automatisch goed gezet aan de hand van de opgeslagen parameters. Omdat niet elk filter ondersteuning hoeft te bieden voor het herladen van data, wat bij onder andere een zoekterm-filter het geval is, wordt er door de filters geïtereerd om deze hun eigen waarde al dan niet op te onthouden.

5. Haal dataset op bij de controller

Nu wordt er met behulp van een API een connectie opgezet met de data interface van de controller, deze voert validatie en parameterconversie toe om op haar beurt de dataset op te vragen bij de data interface. Toon gedurende de aanvraag een laadscherm.

6. [succesvol]

De aanvraag van de dataset hoeft niet altijd valide te zijn of succesvol te verlopen, indien dit niet het geval is wordt er een foutmelding getoond.

7. Toon dataset

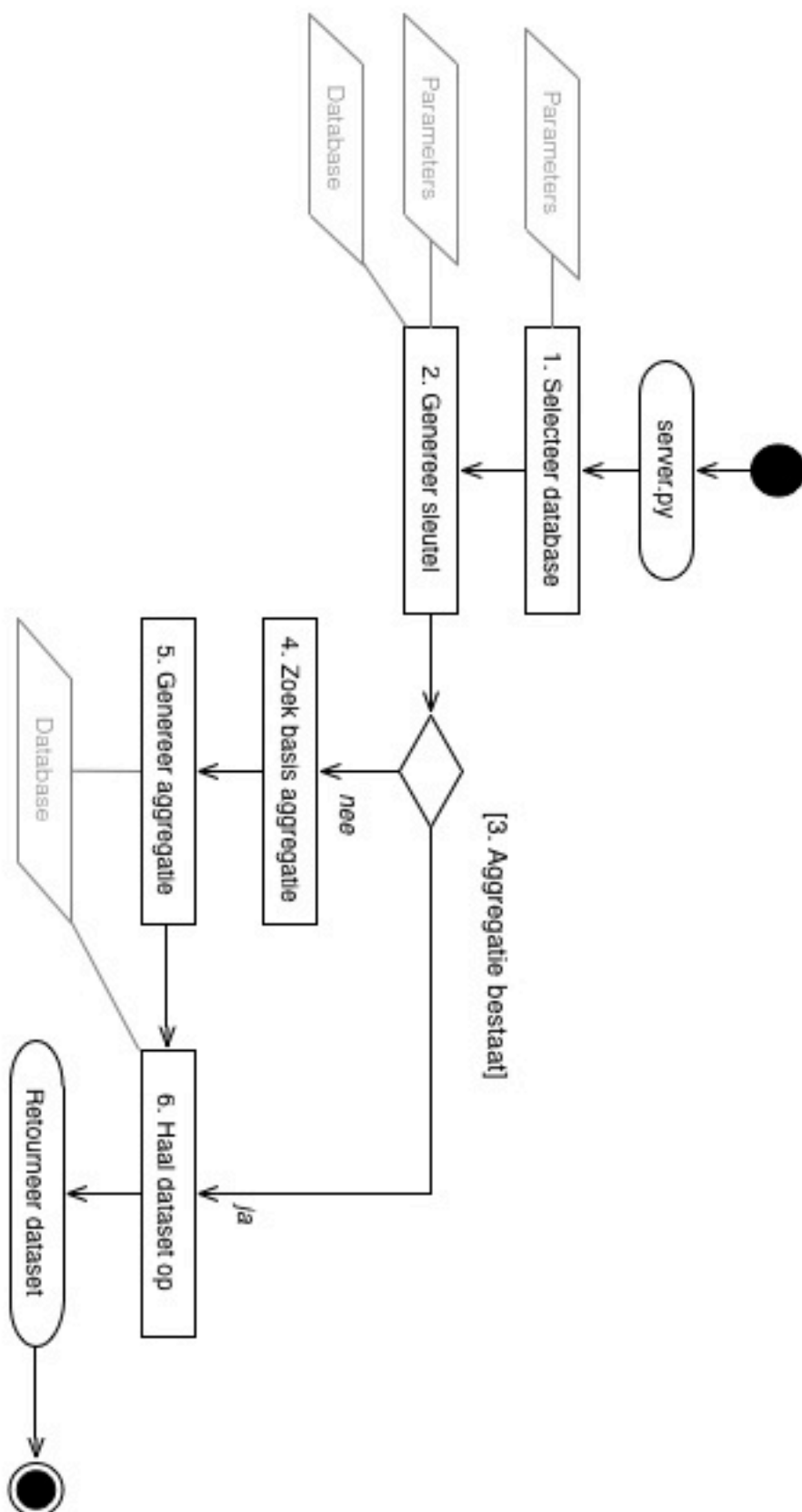
Indien de aanvraag succesvol is wordt de dataset in de tabel uitgetekend, hiervoor maakt ze gebruik van haar eigen tekenmethodieken.

8. [geheel verzoek] / 9. Haal metadata op

Wanneer de dataset niet leeg is en er is sprake van een volledig verzoek wordt er nog een extra verzoek verstuurd voor metadata. Dit betreft met name een aanvraag voor het totaal aantal rijen in de dataset en mogelijk een verzoek voor een gegroepeerde totaalrij. Dit gebeurt in de meeste gevallen als separate aanvragen in het vervolg kunnen de mogelijk samengevoegd worden met stap 5.



5.2 Data interface: ophalen dataset





1. Selecteer database

Allereerst wordt er aan de hand van de parameters de juiste database geselecteerd. Op basis van deze database worden vervolgens aggregaties geselecteerd en bevroegd.

2. Genereer sleutel

Op basis van de parameters wordt er een sleutel gegenereerd welke garant staat voor de naam van de beoogde aggregatie. Hierin vinden enkele opschoningsacties plaats om ervoor te zorgen dat deze niet te specifiek zijn, zo moet paginatie, limitering en sortering bij het genereren van deze sleutel achterwege gelaten worden.

3. [Aggregatie bestaat]

Er wordt gekeken in de database of er reeds een aggregatie met de gegeven sleutel voorkomt, indien dit niet het geval is moet de aggregatie gegenereerd worden. Indien deze wel voorkomt wordt deze gelijk uitgelezen.

4. Zoek basis aggregatie

Alvorens de aggregatie wordt gegenereerd wordt er gekeken waar deze aggregatie op gebaseerd moet worden. In sommige gevallen is het efficiënter om de aggregatie te baseren op een andere aggregatie. Wanneer er geen basis aggregatie kan worden gevonden wordt de hoofdtabel gebruikt.

5. Genereer aggregatie

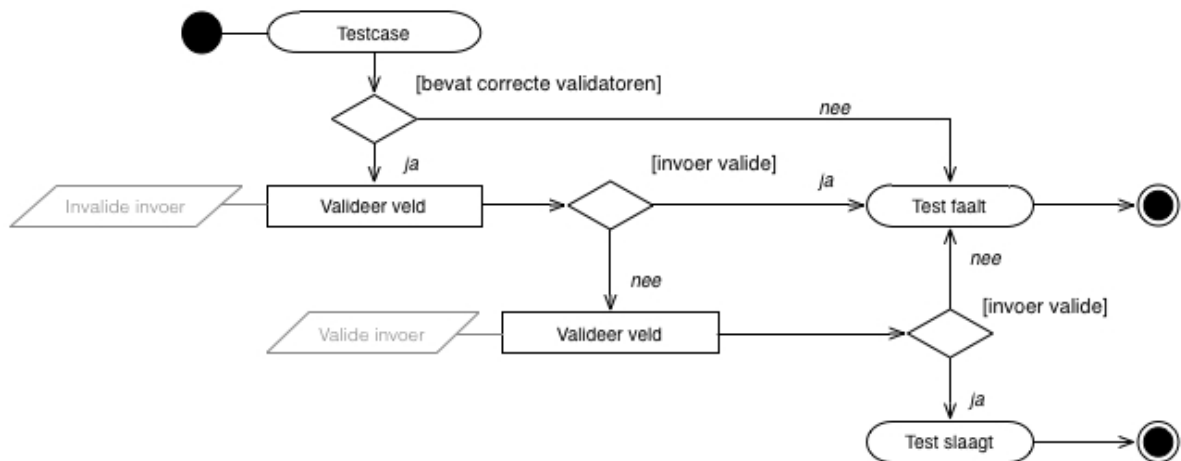
Aan de hand van de parameters wordt er een query samengesteld die verantwoordelijk is voor de inhoud van de aggregatie. De structuur en groepering van de aggregatie staan gedefinieerd in een configuratiebestand, hier heeft de gebruiker dan ook geen invloed op.

6. Haal dataset op

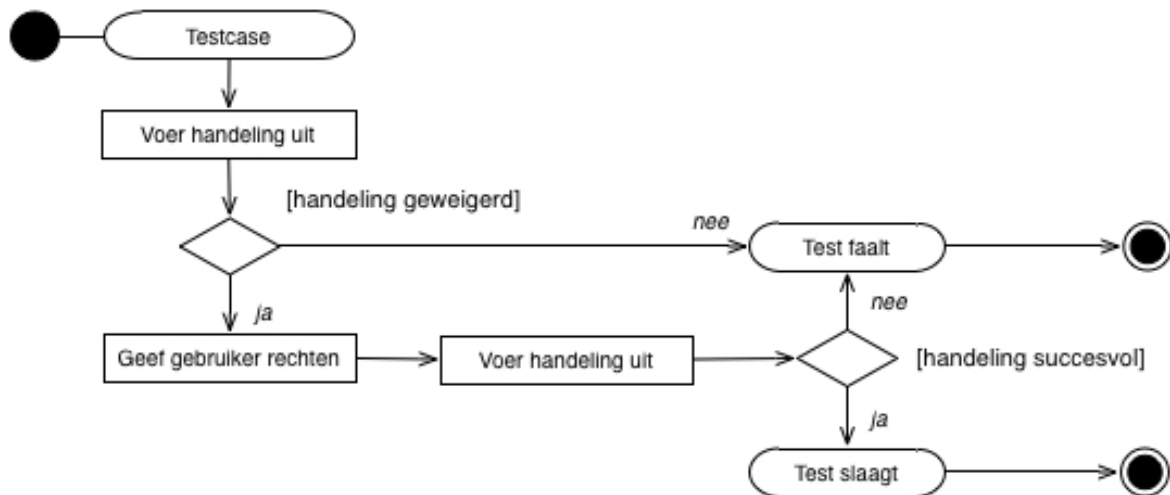
Uiteindelijk wordt er een query op de aggregatie uitgevoerd welke de gewenste dataset berekend en retourneert.



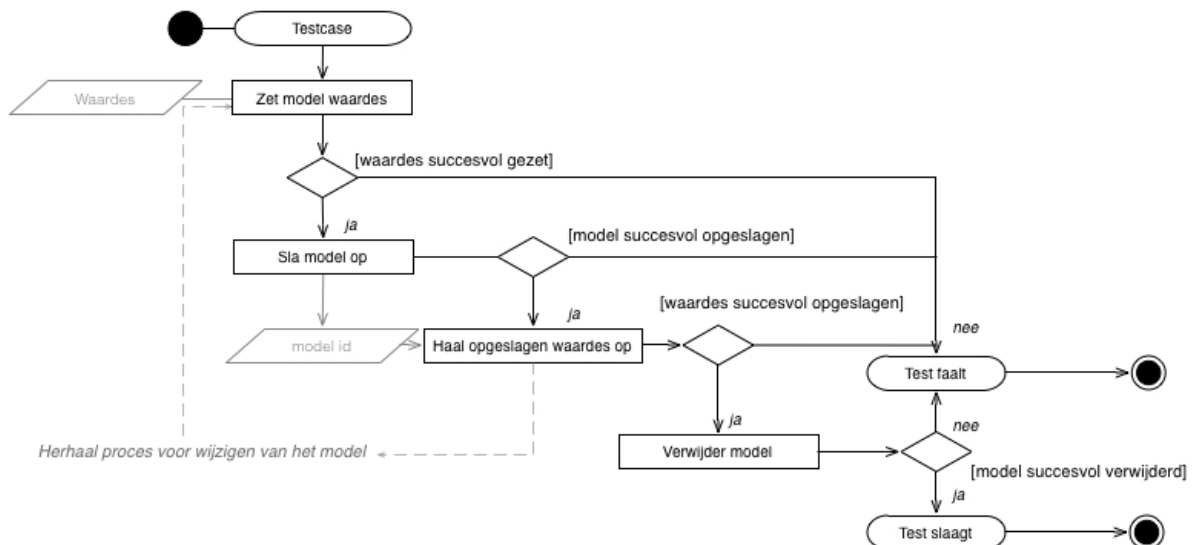
5.3 Globale testcases



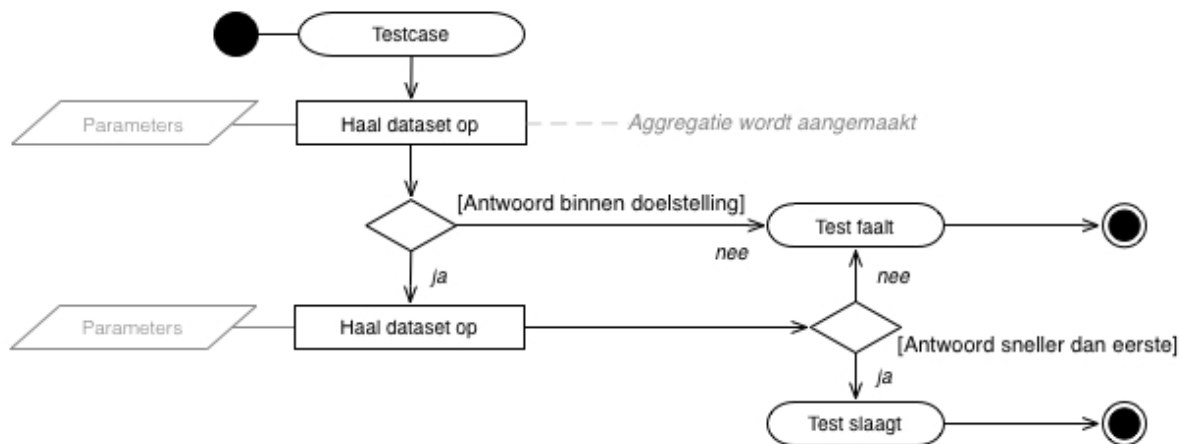
Testcase invoerveld



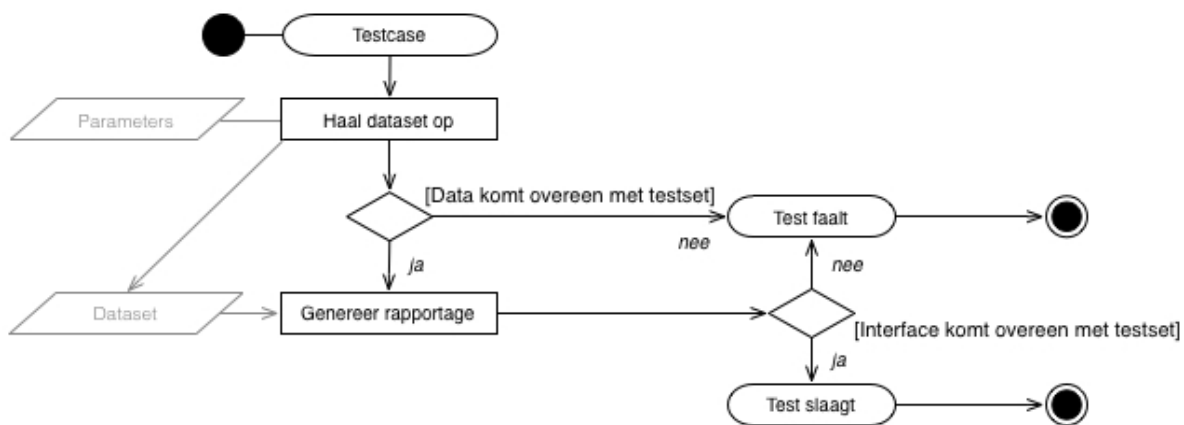
Testcase rechten



Testcase modellen



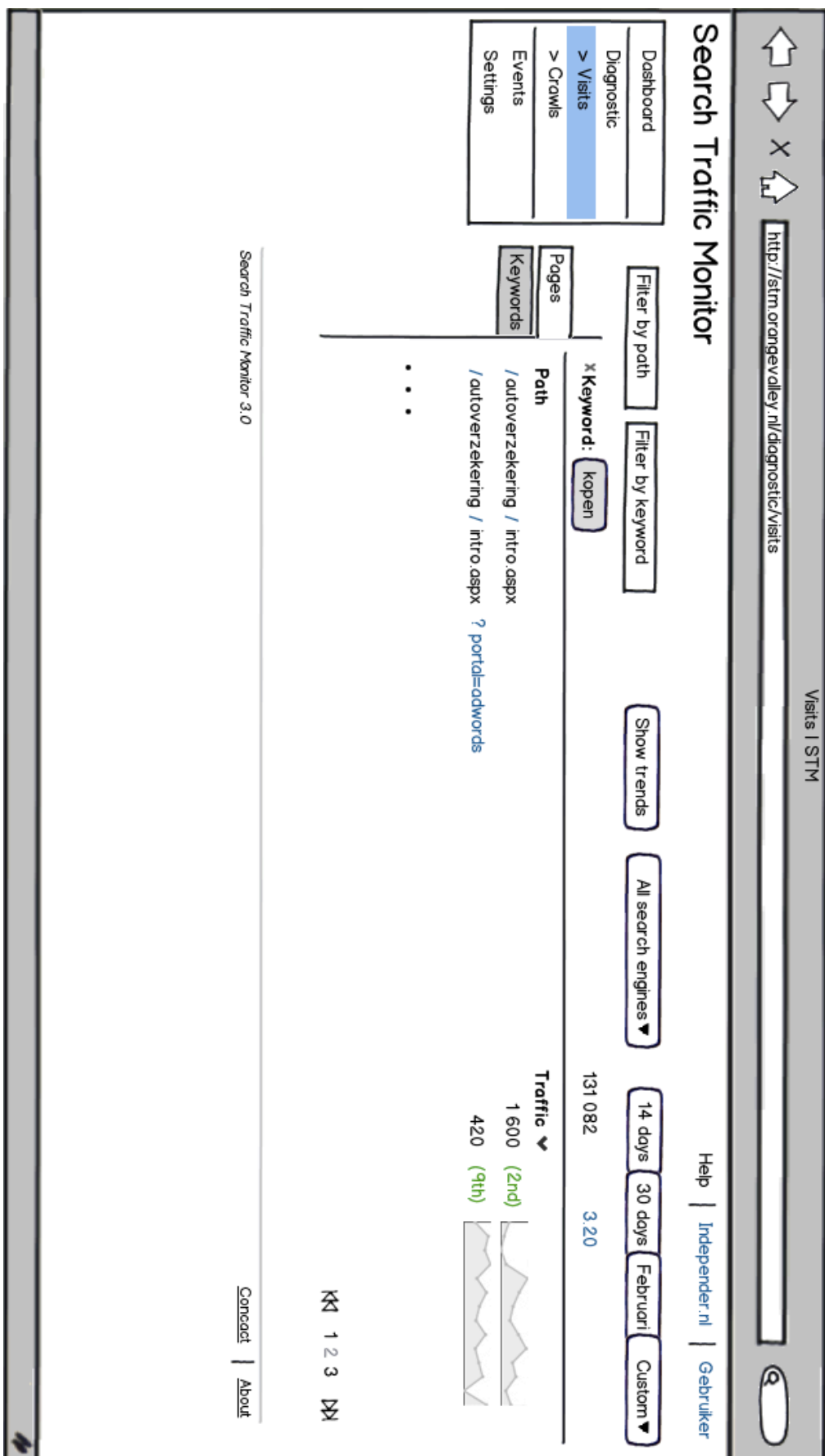
Testcase performance



Testcase data-integriteit

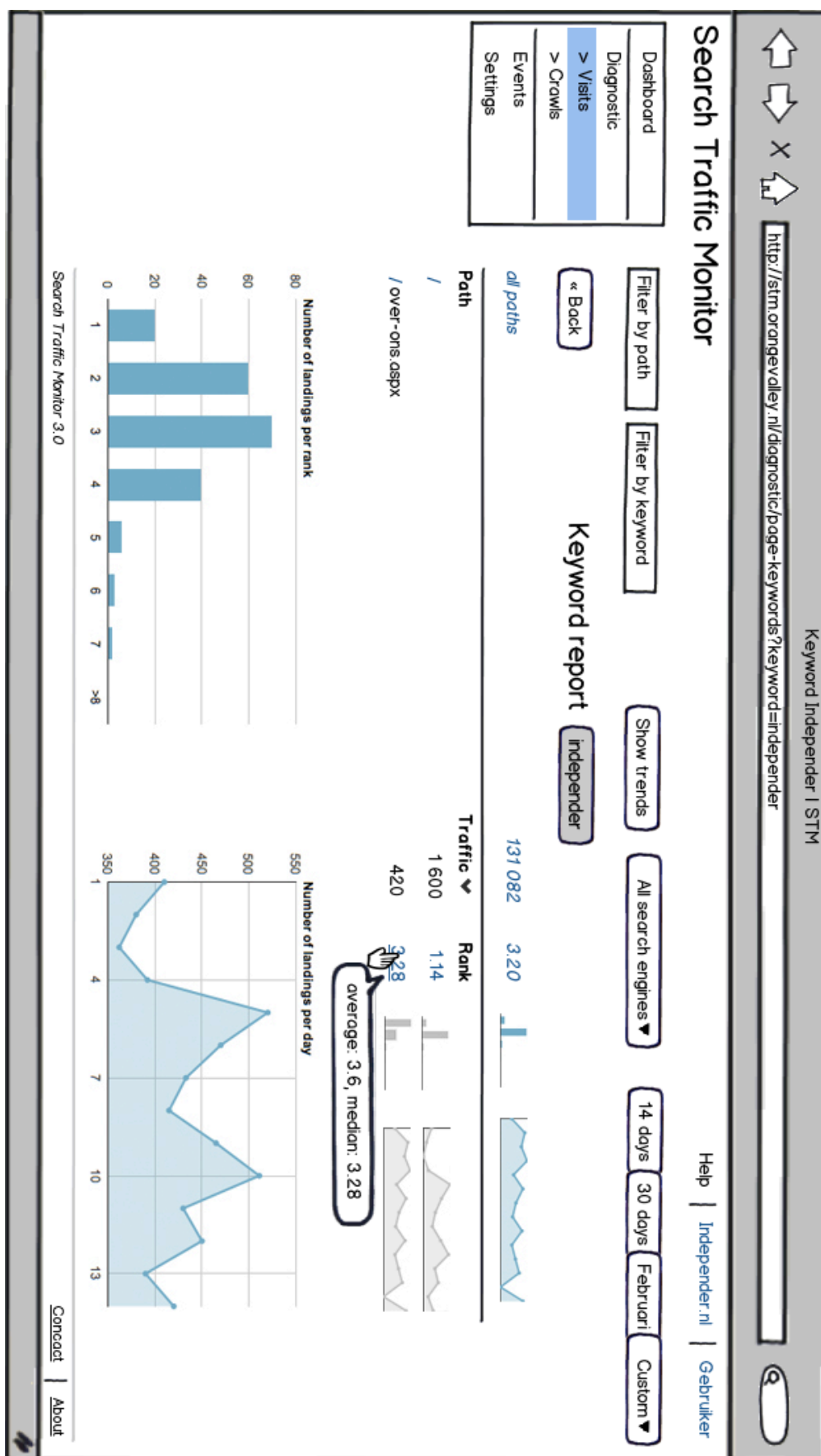


6.2 Bezoekers rapport





6.3 Zoekterm rapport





6.4 Management lijst

Profiles | STM

http://stm.orangevalley.nl/management/profile-list

Search Traffic Monitor

Help | [Independer.nl](#) | [Gebruiker](#)

Profile 0 / 20

Dashboard

Diagnostic

Events

Settings

> Profiles

> Services

> Tasks

> Databases

Action ▾

[Add profile](#)

Name ▾	Implementation	Last updated
Independer.nl	Logfile	Friday march 16 2012, 14:32
42 functions.nl	Trackingpixel	Live data
Edit Activate Trash		
...		

Search Traffic Monitor 3.0

[Contact](#) | [About](#)

KK 1 2 3 KK



6.5 Management gedetailleerd

← → × ↩

http://stm.orangevalley.nl/management/profile?profile=1

Edit Profile | STM

Dashboard
Diagnostic
Events

Settings
 > Profiles
 > Services
 > Tasks
 > Databases

Profile saved

Edit profile

Name
Independenr.nl

Tracking type
Logfiles

Save

Help |

Independenr.nl |

Gebruiker

Independenr.nl
42 functions.nl
orangevalley.nl

Search Traffic Monitor 3.0

Contact | About



6.6 Geautomatiseerde regels

Some text

Event Management | STM

http://stm.orangevalley.nl/management/rule

Search Traffic Monitor

Help | [Independer.nl](#) | [Gebruiker](#)

Dashboard

Diagnostic

> Visits

> Crawls

Events

Settings

Add rule

Name

AdWords Stopknop

Events

Actions

Campaign mutation

+

Add action

Notify

duplicate

delete

Email

sea@orangevalley.nl

Campaign mutation

duplicate

delete

Action

Pause

Account

OrangeValley - AdWords

Save

Search Traffic Monitor 3.0

[Contact](#) | [About](#)



7 Realisatie

7.1 Toevoegen nieuwe entiteit

I. Rechten toekennen

Rechten en rollen worden in het framework beheerd middels de `Pie_ACL` klasse. Het is gebruikelijk dat nieuwe rechten worden toegekend middels het configuratie bestand van de module; `/module/stm/build/module.build.php`.

```
// Extend ACL object
Pie_ACL::grant('member', array(
    'add_example',
    'read_example',
    'edit_example',
    'delete_example',

    // Read/edit/delete examples owned by other members
    'read_other_example',
    'edit_other_example',
    'delete_other_example'
));

// Test if the active member is allowed to read examples
Pie_Register::get('member')->hasRight('read_example')
```

Let op: De admin rol heeft standaard alle rechten, deze kunnen specifiek worden ingetrokken middels een revoke op de `Pie_ACL` klasse.



II. Controller uitbreiden

De applicatie moet worden uitgebreid met twee pagina's, een lijst om de entiteiten in te weergeven en een formulier waarmee entiteiten kunnen worden toegevoegd/gewijzigd. De eerste stap hiervoor is het uitbreiden van de controller met de twee methodes zodat de pagina's via het web te bereiken zijn; */module/stm/view/management/controller.class.php*.

```
// Formulier    => /management/example
public function exampleAction(Pie_Layout &$layout, Pie_Request &$request)
{
    $entity = new STM_Example();
    $entity->setExampleId($request->getParam('example'))
        ->load();

    // Test if the member is allowed to adjust the loaded entity
    $right = $entity->getExampleId() ? 'edit_example' : 'add_example';
    if(!Pie_Register::get('member')->hasRight($right, $entity->getMemberId()))
        Pie_ErrorManager::trigger(403);

    // Create and process form
    $layout->form = new STM_ExampleForm('example', 'example', array('entity' => $entity));
    if($layout->form->isSubmitted())
        $layout->form->process();

    // Editing example
    if($layout->form->entity->getExampleId())
    {
        $layout->form->setAction('?example='.$layout->form->entity->getExampleId());
        $layout->setContent('title', array(
            './example-list' => __('Example list'),
            './example' => array('content' => __('Add example'), 'right' => 'add_example'),
            __('Edit example')
        ));
    }
    // Adding new example
    else
    {
        $layout->setContent('title', array('./example-list' => __('Example list'), __('Add example')));
    }
}

// Tabel    => /management/example-list
public function exampleListAction(Pie_Layout &$layout, Pie_Request &$request)
{
    // Test if member is allowed to read examples
    if(!Pie_Register::get('member')->hasRight('read_example'))
        Pie_ErrorManager::trigger(403);

    $layout->list = new STM_ExampleTable();
    $layout->list->bulkAction($layout, $request);

    $layout->setContent('title', array(
        __('Example list'),
        './example' => array('content' => __('Add example'), 'right' => 'add_example')
    ));

    // Re-use existing view
    $this->setRenderTarget('virtual/view/public/pie-admin/list.phtml');
}
```



III. Lijst ontwikkelen

Voor het genereren van een lijstweergave van de entiteiten wordt er een variant geschreven op de CMS_Table klasse. Middels deze klasse wordt er een lijst gegenereerd welke gelijk ondersteuning biedt voor sortering en paginering. Indien gespecificeerd kan de lijst tevens uitgebreid worden met filters en bulk acties.

```
public function __construct()
{
    // Define table columns
    parent::__construct(array(
        'name' => array('content' => 'Name', 'field' => 'example_name', 'sort' => 1),
        'type' => array('content' => 'Type', 'field' => 'example_type', 'sort' => 1)
    ));
}

public function bulkAction(Pie_Layout &$layout, Pie_Request &$request)
{
    if($request->getParam('action'))
    {
        $entities = $request->getParam('example', array());
        $entities = is_array($entities) ? $entities : array($entities);

        $lang = count($entities) == 1 ? 'example' : 'examples';
        if(count($entities))
        {
            switch($request->getParam('action'))
            {
                case 'delete':
                    if(Pie_Register::get('member')->hasRight('delete_example'))
                    {
                        $statement = new Pie_MutateQuery(Pie_MutateQuery::TYPE_DELETE);
                        $statement->from('STM_EXAMPLE')
                            ->where('example_id IN ?', array($entities));

                        if(!Pie_Register::get('member')->hasRight('delete_other_example'))
                            $statement->where('member_id = ?', Pie_Register::get('member')->getMemberId());

                        if($count = $statement->execute())
                            $layout->addNotice('success', sprintf(__('%d %s deleted.'), $count, $lang));
                        else
                            $layout->addNotice('info', sprintf(__('No %s deleted.'), $lang));
                    }
                    else
                    {
                        $this->addPermissionNotice($layout);
                    }
                    break;
                default:
                    $layout->addNotice('error', __('Unknown action specified.));
            }
        }
        else
        {
            $layout->addNotice('info', __('No examples were specified to preform the action on.));
        }

        // Make sure the request isn't posted twice
        $request->removeParam('action')
            ->removeParam('example');
    }

    return $this;
}
```



```
protected function _single(array &$entity, array $merge = array())
{
    $seurl = stm_management_path('example', array('example' => $entity['example_id']));
    $sdurl = stm_management_path('example-list', array('example' => $entity['example_id'], 'action' => 'delete'));

    $actions = array(
        'edit' => '<a href = "'. $seurl. '">'. __('Edit'). '</a>',
        'delete' => '<a href = "'. $sdurl. '" class="danger" data-toggle = "confirm"
                    title = "'. __('Are you sure you want to delete this example?'). '"
                    data-header = "'. __('Delete'). ' \\''. Pie_Protection::output($entity['example_name']). '\\ ">'.
                    __('Delete').
        '</a>'
    );

    // Remove actions the member doesn't have the proper rights for
    if(!Pie_Register::get('member')->hasRight('edit_example'))
        unset($actions['edit']);
    if(!Pie_Register::get('member')->hasRight('delete_example'))
        unset($actions['delete']);

    $entity['example_name'] = '<strong>'. Pie_Protection::output($entity['example_name']). '</strong>
                             <div class = "list-actions">'. implode(' | ', $actions). '</div>';

    return parent::_single($entity, $merge);
}

protected function _statement($simple = false)
{
    $statement = new Pie_SelectQuery();
    $statement->from('STM_EXAMPLE', array('example_id', 'example_name', 'example_type'))
        ->orderBy('example_type ASC, example_name ASC');

    // Limit the examples connected to member
    if(!Pie_Register::get('member')->hasRight('read_other_example'))
        $statement->where('member_id = ?', Pie_Register::get('member')->getMemberId());

    return $statement;
}
```



IV. Model ontwikkelen

Modellen representeren de code-implementatie van een entiteit in de database. Hiervoor kan gebruik gemaakt worden van de klasse *Pie_DatabaseOptionModel* en *Pie_DatabaseModel* welke de koppeling met de database voor een groot gedeelte afhandelt.

```
// Cast is used to load an array into the model
protected function _cast(array &$data)
{
    parent::_cast($data);
    $this->setExampleId($data['example_id'])
        ->setName($data['example_domain']);

    return $this;
}

// -- Getters and setters
// ...

// Set the related table and load the identity
protected function _identity(Pie_Query &$statement, $primary = false)
{
    $statement->from('STM_EXAMPLE');

    if($this->getExampleId())
        $statement->where('example_id = ?', $this->getExampleId());
    else if(!$primary && $this->getName())
        $statement->where('example_name = ?', $this->getName());
    else
        return false;

    return true;
}

// Reset the model to its default values
public function reset()
{
    parent::reset();
    $this->_setData('example_id', null)
        ->_setData('example_name', null);

    return $this;
}
```



```
// The _store method is overloaded in case to remove the primary id from the changes
// and make sure the primary id is set when our model is saved (callback)
protected function _store($error, array &$changes, $method = null)
{
    try
    {
        unset($changes['example_id']);

        return parent::_store(Pie_ErrorManager::ERROR_EXCEPTION, $changes, 'setExampleId');
    }
    catch(Pie_Error $e)
    {
        if($e->getCode() == 23000)
        {
            $mapper = array('example_name' => 'name');
            $e->setMessage(sprintf(
                __('There already is an example registered with the given %s.'),
                __($mapper[$e->getData('field')])
            ));
            break;
        }

        $e->handle($error);
    }
    return false;
}
```



V. Formulier ontwikkelen

Tenslotte wordt er een formulier ontwikkelt waarmee de gebruiker in staat is het bovenstaande model aan te passen. Omdat het hier gebruikersinvoer betreft wordt er van het formulier verwacht dat deze de invoer valideert voor deze aan het model worden meegegeven.

```
// Initialize the form
protected function _init()
{
    if(!$this->entity)
        $this->entity = new STM_Example();

    // Add basic elements
    $this->addElement('text', 'example_name', array(
        'label' => 'Name',
        'required' => true,
        'value' => $this->entity->getName()
    ));

    $this->addElement('submit', 'submit', array('label' => 'Save', 'class' => 'btn-primary'));
}

// Process the form
protected function _process(array &$values)
{
    $entity = $this->entity->getExampleId() ? $this->entity : new STM_Example();
    $entity->setName($values['example_name']);

    try
    {
        if(!$entity->store(Pie_ErrorManager::ERROR_EXCEPTION))
            throw new Exception('An error occurred while saving the example.');

```

 $this->entity = $entity;
 $this->_setResponse('Example stored');
 return true;
 }
 catch(Exception $e)
 {
 $this->addError($e->getMessage());
 }
 return false;
}
```


```




7.2 Toevoegen nieuwe rapportage

I. Controller | API Uitbreiden

De wordt gebruikt om de data uit de data interface te converteren naar het juiste formaat. Hierin zijn twee formaten gedefinieerd, het API formaat en het export formaat. Dit met het verschil dat de export geen geneste output mag bevatten.

```
protected function _convertRow(array &$row, array &$args, array &$date)
    case 'keywords':
        $row = array(
            $row['_id'] ? $row['_id'] : '[not provided]',
            (int)$row['value']['nvisits'],
            isset($row['value']['rank_avg']) ? $row['value']['rank_avg'] : 0,
            $row['value']['nrank'],
            $this->_convertTrafficPerDay($row['value']['traffic_per_day'], $date['start'], $date['end']),
            $this->_convertRank($row['value']['rank']),
        );
        break;

protected function _exportRow(array &$row, array &$args)
    case 'keywords':
        $row = array(
            $row['_id'] ? $row['_id'] : '[not provided]',
            (int)$row['value']['nvisits'],
            isset($row['value']['rank_avg']) ? $row['value']['rank_avg'] : '-'
        );
        break;
```

II. JavaScript | Rapportage aanmaken

De laatste stap is het omzetten van de data uit de API naar een visueel rapport. Hiervoor wordt gebruik gemaakt van de stmTable jQuery plugin. Aan de hand van deze plugin wordt automatisch de data uit de API opgehaald en geconverteerd naar tabelweergave.

```
$('#keyword-table').stmTable({
    // Table columns
    cols : [{label:'Path', sortable: 'path'}, {label: 'Traffic', sortable: 'traffic', sorted: 1}, {}],
    // Static parameters passed with the request
    params : {type:'paths'},
    // Filters allowed on the table
    filters : ['date', 'path', 'statuscode', 'searchengine', 'threshold'],
    // Url tot the used API
    url : '/api/diagnostic',
    // Include total row
    total : true
    , single : function (row, key) {
        return '<tr>' +
            '<td>' + Parser.parseUri(row[0]) + '</td>' +
            '<td class = "number-col">' + Parser.formatNumber(row[1]) + '</td>' +
            '<td class = "chart-col">' +
                '<div class = "chart bar" style = "width:80px;" data = \'' + JSON.stringify(row[5]) + '\></div>' +
                '<div class = "chart line" style = "width:200px;" data = \'' + JSON.stringify(row[4]) + '\></div>' +
            '</td>' +
            '</tr>';
    }
    , afterRender : function (e, total) {
        if(typeof(google) == 'undefined')
            return;

        // Render charts
        // ...
    }
});
```



E Scrum



Inhoudsopgave

1	Product Backlog	132
2	Sprint 1	137
3	Sprint 2	139
4	Sprint 3	141
5	Sprint 4	142



1 Product Backlog

1.1 Architectuur

1.1.1 Wireframes

	Deeltaak	Units
1.	Zoekmachine rapport	3
2.	Bezoekers rapport	3
3.	Zoekterm rapport	1
4.	Management lijst	2
5.	Management gedetailleerd	2
6.	Geautomatiseerde regels	4
	Totaal	15

1.1.2 Mockups

	Deeltaak	Units
1.	View – Uitwerken layout	32
2.	View – Zoekmachine rapport	8
3.	View – Bezoekers rapport	8
4.	View – Zoekterm rapport	8
5.	View – Detail zoekterm rapport	8
	Totaal	64

1.1.3 EER

	Deeltaak	Units
1.	Web database	8
2.	Management database	8
	Totaal	16

1.1.4 Domeinmodel

	Deeltaak	Units
3.	HTML/JS: View	8
4.	PHP: Controller	16
5.	Python: Data interface	8
	Totaal	32



1.1.5 Processen in kaart brengen

	Deeltaak	Units
1.	Sequentiediagram – ophalen dataset	8
2.	Sequentiediagram – wisselen rapportage	8
3.	Flowchart – View: ophalen dataset	8
4.	Flowchart – Data interface: ophalen dataset	8
	Totaal	32

1.1.6 Prototype

	Deeltaak	Units
1.	View – Functionaliteit pad filter	4
2.	View – Functionaliteit keyword filter	4
3.	View – Functionaliteit searchengine filter	4
4.	View – Functionaliteit datum filter	4
5.	View – Functionaliteit threshold filter	4
6.	View – Functionaliteit sortering	4
7.	View – Functionaliteit paginatie	4
8.	View – Dynamisch opbouwen rapportage	16
9.	Controller – Rapportage API	2
10.	Data interface – Socket	4
11.	Data interface – Aggregatie opbouwen	16
12.	Data interface – Aggregaties beheren	32
	Totaal	98



1.2 Module

1.2.1 HTML/JS: View

	Deeltaak	Units
1.	View – Uitwerken filter	16
2.	View – Uitwerken rapport	16
	Totaal	32

1.2.2 PHP: Controller

	Deeltaak	Units
1.	Database – Uitwerken webdatabase	4
2.	Controller – Rapportage API	6
3.	Profiel – Model	8
4.	Profiel – Formulier	8
5.	Profiel – Beheerlijst	8
6.	Externe dienst – Model	8
7.	Externe dienst – Formulier	8
8.	Externe dienst – Formulier SiteSpect	8
9.	Externe dienst – Formulier FTP	8
10.	Externe dienst – Beheerlijst	8
11.	Unit – Model	8
12.	Unit – Formulier	8
13.	Unit – Beheerlijst	8
14.	Taak – Model	8
15.	Taak – Beheerlijst	8
16.	API - SiteSpect	8
17.	Geautomatiseerde regels – Beheerlijst	8
18.	Geautomatiseerde regels – Formulier	48
19.	Geautomatiseerde regels – Formulier globale datum	8
20.	Geautomatiseerde regels – Formulier FTP download	8
21.	Geautomatiseerde regels – Formulier SiteSpect download	8
22.	Geautomatiseerde regels – Model	8
23.	Geautomatiseerde regels – Model evenement/actie	8
	Totaal	218



1.2.3 PHP: Dataverwerking

	Deeltaak	Units
1.	Processor – MongoDB connectie integreren	12
2.	Processor – Wijzigen logfile uitlees methodiek	4
3.	Processor – Wijzigen opbouw zoekmachine/bots	6
4.	Processor – Toevoegen bot categorieën (mobile, image, media)	6
	Totaal	28

1.2.4 Python: Data interface

	Deeltaak	Units
1.	Data interface – Socket	4
2.	Data interface – Aggregatie opbouwen	64
3.	Data interface – Aggregaties beheren	16
4.	Data interface – Zoekmachine rapport	8
5.	Data interface – Bezoekers rapport	8
6.	Data interface – Zoekterm rapport	8
7.	Data interface – Detail zoekterm rapport	8
8.	Data interface – Trending zoekterm rapport	8
9.	Data interface – Ranking over tijd	4
	Totaal	128



1.3 Rapportage

1.3.1 Filters

	Deeltaak	Units
1.	View – Functionaliteit pad filter	6
2.	View – Functionaliteit keyword filter	6
3.	View – Functionaliteit searchengine filter	6
4.	View – Functionaliteit datum filter	6
5.	View – Functionaliteit threshold filter	6
	Totaal	30

1.3.2 Rapport

	Deeltaak	Units
1.	View – Zoekmachine rapport	6
2.	View – Bezoekers rapport	6
3.	View – Zoekterm rapport	6
4.	View – Detail zoekterm rapport	12
5.	View – Trending zoekterm rapport	16
6.	View – Ranking over tijd	6
7.	View – Non provided zoekterm	12
8.	Data pattern – Non provided zoekterm distributie	32
	Totaal	96

1.4 Overig

1.4.1 Exporteren

	Deeltaak	Units
1.	CSV/Excel	8
2.	PDF	?
	Totaal	8

1.4.2 Datamigratie

	Deeltaak	Units
1.	Migratieplan	4
2.	Migratie – trackingpixel database	4
3.	Migratie – logfile database	4
4.	Migratie – Testen data integriteit	6
	Totaal	18



2 Sprint 1

2.1 Sprint informatie

Periode 16 maart 2012 – 30 maart 2012
Tijd per unit 30 Minuten
Units beschikbaar 276

Product owner E. Blacqui re
Projectleden F. Perdaan, M. Abspoel

2.2 Sprint items

#	Naam	Units
1.1.1	Wireframes - Zoekmachine rapport	3
1.1.2	Wireframes - Bezoekers rapport	3
1.1.3	Wireframes - Zoekterm rapport	1
1.1.4	Wireframes - Management lijst	2
1.1.5	Wireframes - Management gedetailleerd	2
1.1.6	Wireframes - Geautomatiseerde regels	4
1.2.1	View – Uitwerken layout	32
1.2.2	View – Zoekmachine rapport	8
1.2.3	View – Bezoekers rapport	8
1.2.4	View – Zoekterm rapport	8
1.2.5	View – Detail zoekterm rapport	8
1.3.1	EER – Web database	8
1.3.2	EER – Management database	8
1.4.1	Domeinmodel – HTML/JS: View	8
1.4.2	Domeinmodel – PHP: Controller	16
1.4.3	Domeinmodel – Python: Data interface	8
1.5.1	Sequentiediagram – ophalen dataset	8
1.5.2	Sequentiediagram – wisselen rapportage	8
1.5.3	Flowchart – View: ophalen dataset	8
1.5.4	Flowchart – Data interface: ophalen dataset	8
4.2.1	Migratieplan	8
4.2.2	Migratie – trackingpixel database	4
4.2.3	Migratie – logfile database	4
4.2.4	Migratie – Testen data integriteit	6
1.6.1	View – Functionaliteit pad filter	4
1.6.2	View – Functionaliteit keyword filter	4
1.6.3	View – Functionaliteit searchengine filter	4
1.6.4	View – Functionaliteit datum filter	4
1.6.5	View – Functionaliteit threshold filter	4
1.6.6	View – Functionaliteit sortering	4
1.6.7	View – Functionaliteit paginatie	4
1.6.8	View – Dynamisch opbouwen rapportage	16



1.6.9	Controller – Rapportage API	2
1.6.10	Data interface – Socket	4
1.6.11	Data interface – Aggregatie opbouwen	16
1.6.12	Data interface – Aggregaties beheren	32
Totaal		279

2.3 Sprint buffer

138 (50%)

#	Naam	Units
	Feedback product owner	--
Totaal		0



3 Sprint 2

3.1 Sprint informatie

Periode 30 maart 2012 – 13 april 2012
Tijd per unit 30 Minuten
Units beschikbaar 276

Product owner E. Blacqui re
Projectleden F. Perdaan, M. Abspoel

3.2 Sprint items

#	Naam	Units
2.2.1	Database – Uitwerken webdatabase	4
2.3.1	Processor – MongoDB connectie integreren	12
2.3.2	Processor – Wijzigen logfile uitlees methodiek	4
2.3.3	Processor – Wijzigen opbouw zoekmachine/bots	6
2.3.4	Processor – Toevoegen bot categorie�n (mobile, image, media)	6
2.4.1	Data interface – Socket	4
2.4.2	Data interface – Aggregatie opbouwen	64
2.4.3	Data interface – Aggregaties beheren	16
2.4.4	Data interface – Zoekmachine rapport	8
2.4.5	Data interface – Bezoekers rapport	8
2.4.6	Data interface – Zoekterm rapport	8
2.4.7	Data interface – Detail zoekterm rapport	8
2.4.8	Data interface – Trending zoekterm rapport	8
2.4.9	Data interface – Ranking over tijd	4
2.2.3	Profiel – Model	8
2.2.4	Profiel – Formulier	8
2.2.5	Profiel – Beheerlijst	8
2.2.6	Externe dienst – Model	8
2.2.7	Externe dienst – Formulier	8
2.2.8	Externe dienst – Formulier SiteSpect	8
2.2.9	Externe dienst – Formulier FTP	8
2.2.10	Externe dienst – Beheerlijst	8
2.2.11	Unit – Model	8
2.2.12	Unit – Formulier	8
2.2.13	Unit – Beheerlijst	8
2.2.14	Taak – Model	8
2.2.15	Taak – Beheerlijst	8
2.2.16	API - SiteSpect	8
Totaal		272



3.3 Sprint buffer

136 (50%)

#	Naam	Units
4.1.1	Exporteren – CSV/Excel	8
2.2.1	View – Uitwerken filter	16
2.2.2	View – Uitwerken rapport	16
3.2.8	Data pattern – Non provided zoekterm distributie	32
Totaal		72



4 Sprint 3

4.1 Sprint informatie

Periode 13 april 2012 – 27 april 2012
 Tijd per unit 30 Minuten
 Units beschikbaar 136

Product owner E. Blacquièr
 Projectleden F. Perdaan, M. Abspoel

4.2 Sprint items

#	Naam	Units
2.1.1	View – Uitwerken rapport	16
2.1.2	View – Uitwerken filters	16
3.1.1	View – Functionaliteit pad filter	6
3.1.2	View – Functionaliteit keyword filter	6
3.1.3	View – Functionaliteit searchengine filter	6
3.1.4	View – Functionaliteit datum filter	6
3.1.5	View – Functionaliteit threshold filter	6
3.2.8	Data pattern – Non provided zoekterm distributie	32
2.4.9	Data interface – Ranking over tijd	4
3.2.1	View – Zoekmachine rapport	6
3.2.2	View – Bezoekers rapport	6
3.2.3	View – Zoekterm rapport	6
3.2.4	View – Detail zoekterm rapport	12
3.2.5	View – Trending zoekterm rapport	16
Totaal		144

4.3 Sprint buffer

68 (50%)

#	Naam	Units
4.1.1	Exporteren - CSV/Excel	8
2.2.22	Geautomatiseerde regels – Model	8
2.2.23	Geautomatiseerde regels – Model evenement/actie	8
2.2.17	Geautomatiseerde regels – Beheerlijst	8
2.2.19	Geautomatiseerde regels – Formulier globale datum	8
Totaal		40



5 Sprint 4

5.1 Sprint informatie

Periode 27 april 2012 – 11 mei 2012
Tijd per unit 30 Minuten
Units beschikbaar 136

Product owner E. Blacquièrre
Projectleden F. Perdaan, M. Abspoel

5.2 Sprint items

#	Naam	Units
3.2.7	View – Non provided zoekterm	12
3.2.8	Data pattern – Non provided zoekterm distributie	32
4.1.1	Exporteren – CSV/Excel	8
2.2.17	Geautomatiseerde regels – Beheerlijst	8
2.2.18	Geautomatiseerde regels – Formulier	48
2.2.19	Geautomatiseerde regels – Formulier globale datum	8
2.2.22	Geautomatiseerde regels – Model	8
2.2.23	Geautomatiseerde regels – Model evenement/actie	8
2.2.4	Controller – Rapportage API	6
Totaal		138

5.3 Sprint buffer

68 (50%)

#	Naam	Units
2.2.20	Geautomatiseerde regels – Formulier FTP download	8
2.2.21	Geautomatiseerde regels – Formulier SiteSpect download	8
3.2.6	View – Ranking over tijd	6
Totaal		22



F Adviesrapport



Inhoudsopgave

1	Inleiding	145
2	Applicatie	146
2.1	<i>Technische eisen</i>	146
2.2	<i>Aandachtspunten</i>	147
2.3	<i>Snelheid</i>	148
2.3.1	Query optimalisatie	148
2.3.2	API requests	148
2.4	<i>Hoeveelheid query's</i>	149
2.5	<i>Interface</i>	151
2.6	<i>Framework</i>	152
3	Hardware en OS	155
3.1	<i>Technische eisen</i>	155
3.2	<i>Aandachtspunten</i>	155
4	Database	156
4.1	<i>Technische eisen</i>	156
4.2	<i>Aandachtspunten</i>	156
4.3	<i>Inrichting</i>	158
4.3.1	Huidige situatie	158
4.3.2	Voorgestelde situatie	160
4.4	<i>Inhoud</i>	160
5	Database Management Systeem	161
5.1	<i>Technische eisen</i>	161
5.2	<i>Aandachtspunten</i>	162
5.3	<i>Executieplan</i>	163
5.3.1	Optimale situatie	163
5.3.2	MySQL	164
5.4	<i>Google BigQuery</i>	165
5.5	<i>MongoDB</i>	165
5.6	<i>CouchDB</i>	165
6	Bijlage	166
6.1	<i>Bronnen</i>	166
6.2	<i>Systeemconfiguratie</i>	167
6.2.1	Systeem	167
6.2.2	Database	167
6.2.3	PHP	174
6.3	<i>Query rapport</i>	182

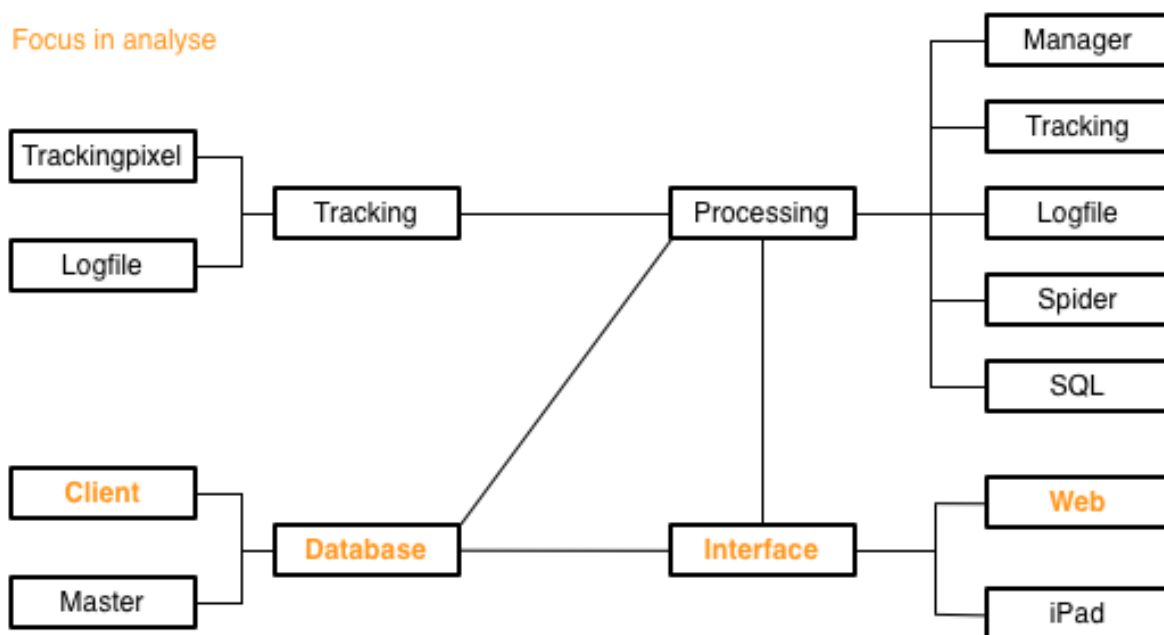


1 Inleiding

In dit adviesrapport wordt er kritisch gekeken naar de performanceproblemen waar Search Traffic Monitor momenteel mee te kampen heeft. Het streven is om de wachttijd per rapport, welke nu varieert tussen de vijf minuten en het crashen van de server, terug te dringen naar een gemiddelde van twee seconden. Dit rapport is dan ook bedoeld als eerste stap richting die twee seconden.

Dit adviesrapport belicht Search Traffic Monitors performance vanuit vier verschillende aspecten, zijnde: de applicatie, de hardware, de database en het databasemanagement-systeem. Met name deze vier aspecten hebben veel invloed op de performance van de applicatie. Per aspect worden de bottlenecks van de performance in kaart gebracht en worden er oplossingen aangedragen ter verbreding van de bottleneck. Het is uiteindelijk aan de opdrachtgever om aan de hand van de kosten/baten analyse te bepalen welke oplossingen er gepraktiseerd worden.

Binnen dit adviesrapport richten we ons enkel op het analyserend gedeelte van Search Traffic Monitor. Dit omdat de verwerkende onderdelen schaalbaar zijn opgezet waardoor de performance van deze onderdelen relatief eenvoudig te verbeteren is.



Simplistische representatie van de relaties tussen de primaire onderdelen binnen Search Traffic Monitor, hierin zijn relaties tussen de subonderdelen achterwege gelaten



2 Applicatie

2.1 Technische eisen

- De grens van 32Mb actief geheugen moet in acht gehouden worden ten behoeve van de kwaliteit, met betrekking tot performance, van het systeem.
- De applicatie maakt gebruik van een framework om zo snel nieuwe onderdelen te kunnen ontwikkelen
 - Het framework moet goed gedocumenteerd zijn
 - Opbouw van het framework
 - Configuratiemogelijkheden
 - Codebeschrijving
 - Functies
 - Methodes
 - Classes
 - Het framework moet actief door ontwikkeld worden en volledig ondersteund worden door haar ontwikkelaar
 - Het framework moet ondersteuning bieden voor het toevoegen van eigen plugins/modules
 - Het framework moet profiling ondersteunen
 - Uitgevoerde query's
 - Laadtijd
 - Geheugenverbruik



2.2 Aandachtspunten

Verwerkingssnelheid

De verwerkingssnelheid is de tijd waarin de server de pagina gegenereerd heeft, de overdrachtstijd van de server naar de browser wordt hiermee automatisch buiten beschouwing gelaten. Door de verwerkingssnelheid te optimaliseren kan de pagina sneller geladen worden, eventueel kan optimalisatie ook gebruikt worden om widgets uit de cache gelijk in te laden om zo een extra serververzoek te voorkomen.

Geheugen

Binnen PHP is het gebruikelijk om per serververzoek 32Mb geheugen te alloceren, wanneer het verzoek deze grens overschrijdt crashed het verzoek. Om deze grens niet te overschrijden beperken we ons in de acties die we uitvoeren; zo wordt bijvoorbeeld de opgevraagde dataset uit de database, soms bestaande uit enkele gigabytes, beperkt tot enkele megabytes. Naarmate we het geheugenverbruik te verbeteren kunnen we dus bijvoorbeeld grotere datasets tonen.

Om het geheugen te verbeteren is het noodzakelijk dat het framework zelf zo min mogelijk geheugen in beslag neemt en dat deze in staat is om alle objecten netjes op te ruimen om geheugen vrij te maken voor nieuwe acties.

Hoeveelheid query's

De laadtijd van de pagina wordt onder andere beïnvloed door de hoeveelheid query's op de pagina. Tevens wordt door extensief query gebruik onnodig de database belast. Zo kan het voorkomen dat een simpele query op een zware query moet wachten omdat de capaciteit van de database niet verder rekt, wat desastreus is voor de laadtijd van de pagina.

Interface

De interface beïnvloed de interne structuur van de applicatie. Door bepaalde mogelijkheden in de interface aan te bieden is het soms nodig om een andere architectuur in je applicatie te hanteren, dit hoeft in het optiek van performance echter niet altijd een gewenste architectuur te zijn.

Framework

Het gekozen framework heeft invloed op al de bovenstaande disciplines en is daarmee een belangrijk aandachtspunt. Tevens wordt de rest van de applicatie opgebouwd aan de hand van het framework, dit bepaald de architectuur van de applicatie en daarmee ook de complexiteit.

Er is tijdens de ontwikkeling gekozen om Search Traffic Monitor te schrijven met behulp van het Zend Framework. Hier is destijds voor gekozen omdat Zend Framework de netste code had en de meeste mogelijkheden bood ten opzichte van alternatieve frameworks. Verder telt mee dat Magento gebaseerd is op het Zend Framework, dit met het voordeel dat de aanwezige kennis over Magento gebruikt kon worden voor de ontwikkeling van Search Traffic Monitor.



2.3 Snelheid

2.3.1 Query optimalisatie

Een groot deel van Search Traffic Monitor bestaat uit rapporten. Deze rapporten zijn voor de gebruiker compleet aanpasbaar via een interface om zo te kunnen filteren op specifieke informatie. Omdat een rapport doorgaans uit complexer query's bestaan hebben deze doorgaans een zwaardere laadtijd, om dit te reduceren rapporten gecached.

Het cachen van rapporten wordt gedaan op basis van de instellingen van het rapport en de dataset welke zij toont. In het verleden is er de keus gemaakt om instellingen met betrekking tot de interface hierbij buiten beschouwing te laten. Dit met het doel om de cache tussen rapporten te kunnen delen, dit betekent dat rapporten altijd de volledige dataset ophalen ook wanneer bepaalde data niet in het rapport getoond wordt.

In de praktijk komt het echter niet tot weinig voor dat cachebestanden tussen meerdere rapporten gedeeld worden. Het is daarom efficiënter om enkel de data op te halen welke door het rapport gebruikt wordt. Door de dataset te reduceren verkleinen we de werkdruk op de database, voornamelijk in het aantal overbodige leesacties en de dataset welke zij behandelt en retourneert.

<i>Tijdsdruk</i>	Applicatie / drie tot vijf dagen
<i>Voordelen</i>	• Werkdruk op de database neemt af • De grootte van de gecacheerde bestanden neemt af
<i>Nadelen</i>	-

2.3.2 API requests

Om de laadtijd van de pagina te beperken is ervoor gekozen om rapporten met behulp van javascript via een API in te laden. Het voordeel van deze methode is dat meerdere rapporten simultaan geladen kunnen worden en dat de server minder snel tegen geheugenlimieten, resulterend uit slechte garbage collection, aan loopt. Het nadeel van deze methode is dat elk request in het huidige framework minimaal 28 query's¹ nodig heeft, dit is voor de API requests momenteel niet anders.

Om het aantal API requests te beperken kan ervoor gekozen worden om gecacheerde rapporten gelijk in te laden. Dit gaat echter wel ten koste van de laadtijd van de pagina, om een balans te maken tussen de laadtijd en de API requests moet er echter wel een restrictie worden gelegd op het actief geheugen en de huidige laadtijd.

<i>Tijdsdruk</i>	Single access point / één dag
<i>Voordelen</i>	• Werkdruk op de database neemt af • Hoeveelheid pagina requests neemt af
<i>Nadelen</i>	-

¹ Zie 0 Hoeveelheid query's



2.4 Hoeveelheid query's

Als we een standaard pagina binnen de ingelogde sectie opvragen zien we dat er 28 query's op de pagina worden uitgevoerd, dit ten ongunste van de werklust op de database en daarmee ook de laadtijd van de pagina. Het is belangrijk om juist deze 28 terug te dringen omdat deze 28 query's het minimum aantal query's op een pagina representeert.

Als we nader inzoomen op de lijst van uitgevoerde query's zien we dat veel query's indirect dubbel worden uitgevoerd. Een relatief eenvoudige performancewinst kan dan ook behaald worden door deze dubbele query's te vermijden, hiermee wordt het aantal query's al terug gedwongen tot 17 query's.

Verder kan de hoeveelheid query's worden beperkt door bepaalde data te cachen in de sessie van de gebruiker. Hierbij moet rekening gehouden worden met wijzigingen die binnen de sessie kunnen optreden, wanneer er sprake is van een relevante wijziging moet deze cache automatisch weer geleegd worden. Het automatisch legen van de cache kan gerealiseerd worden door de in te hooken op relevante save events binnen het framework.

<i>Tijdsdruk</i>	Single point / twee a drie dagen
<i>Voordelen</i>	• Werkdruk op de database neemt af • Opzet naar HTML5 pushstate
<i>Nadelen</i>	-

Zodra het maken van een API request op een efficiënte manier gebeurt bied dit een opening naar het HTML5 pushstate principe. Met behulp van het pushstate pincipe kunnen we het MVC¹ model beter gaan benutten door bij een secundaire request de wijzigende views op te halen in plaats van een volledige webpagina.

¹ Multi View Controller



Query
DESCRIBE mbr_member
SELECT * FROM mbr_member WHERE (member_id = 1)
DESCRIBE mbr_profile
SELECT * FROM mbr_profile WHERE (profile_id = 3)
DESCRIBE mbr_service
SELECT * FROM mbr_service WHERE (member_id = 1)
DESCRIBE mbr_profile
SELECT * FROM mbr_profile WHERE (member_id = 1) ORDER BY name ASC
DESCRIBE mbr_service_profile
SELECT * FROM mbr_service_profile WHERE (profile_id = 3)
SELECT * FROM mbr_service WHERE (service_id = 6)
SELECT * FROM mbr_service WHERE (service_id = 15)
SELECT * FROM mbr_service WHERE (service_id = 17)
SELECT * FROM mbr_service WHERE (service_id = 18)
SELECT * FROM mbr_service WHERE (service_id = 19)
SELECT * FROM mbr_service WHERE (service_id = 6)
SELECT * FROM mbr_service WHERE (service_id = 15)
SELECT * FROM mbr_service WHERE (service_id = 17)
SELECT * FROM mbr_service WHERE (service_id = 18)
SELECT * FROM mbr_service WHERE (service_id = 19)
DESCRIBE mbr_dockable
SELECT * FROM mbr_dockable WHERE (member_id = '1') AND (view = 'web')
DESCRIBE mbr_member_role
SELECT * FROM mbr_member_role WHERE (member_id = '1') ORDER BY role ASC
DESCRIBE mbr_custompage
SELECT * FROM mbr_custompage WHERE (member_id = '1') ORDER BY menu_title DESC
DESCRIBE mbr_service_profile
SELECT * FROM mbr_service_profile WHERE (profile_id = '3')

Soortgelijke query's zijn in dezelfde kleur weergegeven



2.5 Interface



2.6 Framework

Alternatief

Tijdens ontwikkelen van Search Traffic Monitor komt het relatief vaak voor dat we tegen het geheugenlimiet aan lopen. Dit komt omdat Zend Framework van de beschikbare 32Mb al 2Mb gebruikt om het framework draaiende te houden. Daarboven maakt Zend Framework extensief gebruik van objecten, wat ten koste gaat van het geheugen. Het is in de huidige PHP versie namelijk nog niet mogelijk om objecten compleet uit het geheugen op te ruimen.

Naast het extensieve geheugengebruik van Zend Framework lopen we nog tegen enkele ongewenste eigenschappen van het framework aan. Eén van deze eigenschappen is bijvoorbeeld dat Zend Framework voor elke tabel eerst een description ophaalt opdat deze haar joins kan valideren op valide vreemde sleutels. Dat wij geen gebruik maken van vreemde sleutels is juist een bewuste keuze geweest om de snelheid van de database op te krikken. Door al deze kleine eigenschappen hebben we Search Traffic Monitor met een eigen library welke de functionaliteiten van Zend uitbreid en sommige eigenschappen onderdrukt.

Er wordt dan ook gekeken of het beter is om Search Traffic Monitor over te zetten naar een ander framework. Sinds enkele maanden geleden wordt er een framework ontwikkeld welke een alternatief kan bieden voor Zend Framework, namelijk het Pie Framework. Dit framework is simplistisch, makkelijk in gebruik en afgestemd op onze wensen met betrekking tot code. Omdat het Pie Framework van een soortgelijk MVC model gebruik maakt als Zend Framework kan de code zonder veel gedoe omgezet worden naar het nieuwe framework.

Benchmark

Om de performance van beide frameworks tegen elkaar af te zetten maken we gebruik van een benchmark. Voor deze benchmark is een deel van Search Traffic Monitor nagebouwd om zo de resultaten met elkaar te kunnen vergelijken. Beide platformen maken voor de benchmark gebruik van dezelfde database en hetzelfde systeem¹.

Binnen de benchmark wordt er gekeken naar drie disciplines: verwerkingstijd, geheugen en de hoeveelheid query's; deze disciplines worden verkregen door middel van profiling binnen de frameworks. Voor de benchmark vragen we specifieke pagina's 500 keer op om zo een gemiddelde van de matrices te verkrijgen, hiervoor is elke vorm van caching op het systeem en in de software uit gezet.

¹ Configuratie is te vinden in bijlage 7.1 *Systeemconfiguratie*



Globale pagina / Niet ingelogd

	Pie	Zend	Optimalisatie
Verwerkingstijd			
<i>Gemiddeld</i>	0.02417 s	0.03776 s	156.23%
<i>Mediaan</i>	0.02361 s	0.03591 s	152.10%
<i>Hoogst</i>	0.03902 s	0.12098 s	310.05%
<i>Laagst</i>	0.02026 s	0.03247 s	160.27%
Geheugen			
<i>Actief</i>	629292 b	2018512 b	320.76%
<i>Piek</i>	693744 b	2087924 b	300.96%
<i>Opruimvermogen</i>	9.29%	3.32%	279.46%
Hoeveelheid query's	0	0	0%

Globale pagina / Ingelogd

	Pie	Zend	Optimalisatie
Verwerkingstijd			
<i>Gemiddeld</i>	0.03680 s	0.082044 s	222.95%
<i>Mediaan</i>	0.03664 s	0.080532 s	219.79%
<i>Hoogst</i>	0.04872 s	0.122790 s	252.03%
<i>Laagst</i>	0.0286 s	0.075726 s	264.78%
Geheugen			
<i>Actief</i>	797336 b	2387072 b	299.38%
<i>Piek</i>	860892 b	2459196 b	285.65%
<i>Opruimvermogen</i>	7.37%	2.93%	251.72%
Hoeveelheid query's	6	28	133%

Bezoekersrapport / Ingelogd

	Pie	Zend	Optimalisatie
Verwerkingstijd			
<i>Gemiddeld</i>	0.15686 s	0.28011 s	178.57%
<i>Mediaan</i>	0.14877 s	0.27129 s	182.36%
<i>Hoogst</i>	0.23451 s	0.42713 s	182.14%
<i>Laagst</i>	0.14092 s	0.26151 s	185.57%
Geheugen			
<i>Actief</i>	1184336 b	5003876 b	422.50%
<i>Piek</i>	1501848 b	5636420 b	375.30%
<i>Opruimvermogen</i>	21.14%	11.22%	188.39%
Hoeveelheid query's	10	34	340%



Zend Framework

Voordelen	Nadelen
Grote community	Documentatie wordt gevoed door community
Grote gebruikersgroep	Complexer dan noodzakelijk
Groot development team	Geheugenintensief
Veel bestaande modules en plugins beschikbaar	

Pie Framework

Voordelen	Nadelen
Performance is hoog	Geen documentatie
Simplistische opbouw	Geen community
Is ontwikkeld door één van OrangeValley's developers	Geen gebruikersgroep
Goede profiling opties	Bevind zich nog in een beta fase
Geen geforceerde bestandsindeling	
Ondersteunt hooks ¹ uit natura	

<i>Tijdsdruk</i>	Applicatie / Afhankelijk van w
<i>Voordelen</i>	• Geeft mogelijkheid om best practices mee te nemen in de nieuwe code • Creëert aanzienlijk veel overzicht in de code • De performance neemt gemiddeld met 246,26% toe • Geeft de mogelijkheid om gelijk unit tests te gaan bouwen
<i>Nadelen</i>	• Het systeem moet weer getest worden in haar geheel • Magento developers zijn niet meer in staat mee te helpen in de Search Traffic Monitor ontwikkeling

¹ Evenementen binnen de code waarop de developer kan inhaken



3 Software

3.1 Technische eisen

- Er moet een unix kernel aanwezig zijn.
- Het systeem moet volledig beheert kunnen worden door OrangeValley, dit wil zeggen dat wij in staat moeten zijn om software dan wel hardware aan het systeem toe te voegen

3.2 Aandachtspunten

Software

Apache

PHP

MySQL

4 Hardware

4.1 Technische eisen

4.2 Aandachtspunten

Geheugen

Processor



5 Database

5.1 Technische eisen

- De database moet in staat zijn om één gigabyte aan gefilterde data binnen twee seconden te kunnen verwerken en te retourneren.
 - Mede afhankelijk van het databasemanagementsysteem
 - Mede afhankelijk van het systeem

5.2 Aandachtspunten

Inrichting

Datatypes

Indexen

Toegevoegde waarde van kolommen

Type tabel

InnoDB vs MyISAM, winst in geheugen (resulteert in snelheidswinst door verkleinde interne tabellen)

Aggregaties

Inhoud

Overbodige data voorkomen



```
SELECT
    CONCAT("http://", t1.domain, t1.path) AS `url`,
    ROUND(AVG(t1.index_position)) AS `position_base`,
    COUNT(DISTINCT t1.visit_id) AS `traffic_base`,
    COUNT(DISTINCT t1.keywords) AS `keyword_base`,
    ROUND(IF(AVG(t2.index_position) IS NOT NULL,
        AVG(t2.index_position), 0) - AVG(t1.index_position)) AS
`position_diff`,
    COUNT(DISTINCT t1.visit_id) - COUNT(DISTINCT t2.visit_id)
AS `traffic_diff`,
    COUNT(DISTINCT t2.keywords) - COUNT(DISTINCT t1.keywords)
AS `keyword_diff`
FROM
    `stm_visits` AS `t1`
LEFT JOIN `stm_visits` AS `t2` ON
    (t2.path = t1.path AND t2.date >= "2012-01-31 00:00:00"
AND t2.date < "2012-02-14 00:00:00")
WHERE
    ((t1.redirect IS NULL OR t1.redirect != t1.visit_id) AND
    (t1.date >= '2012-02-14'))
GROUP BY t1.path
ORDER BY traffic_diff DESC, t1.path ASC
LIMIT 500
```

Hoeveelheid rijen	39.043
Groote dataset	39Mb
Duratie	715.74 seconden

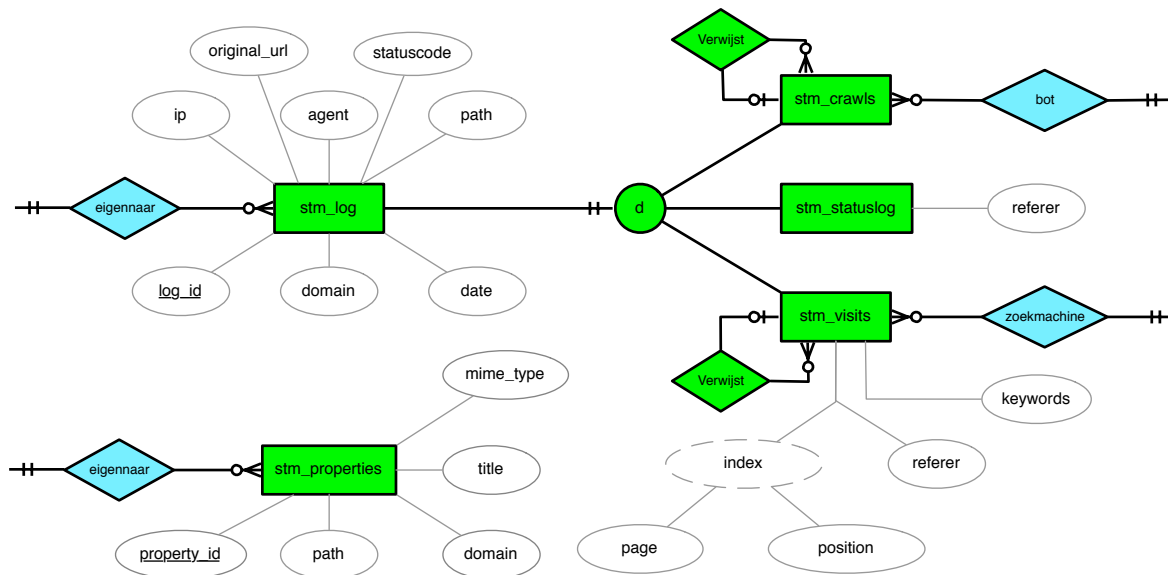
Duratie	Activiteit
0.000107	starting
0.000028	Opening tables
0.000003	System lock
0.000004	Table lock
0.000047	init
0.000013	optimizing
0.000385	statistics
0.000036	preparing
0.001565	Creating tmp table
1.377318	Sorting for group
0.000006	executing
714.347579	Copying to tmp table
0.008686	Sorting result



5.3 Inrichting

5.3.1 Huidige situatie

Master database
Client database



stm_crawls

Veld	Type	Lengte	Index
crawl_id	varchar	52	Primair
owner_id	mediumint	8	Key
original_url	mediumtext		
domain	varchar	30	Key
path	mediumtext		Key(100)
date	datetime		Key
statuscode	smallint	3	
agent	varchar	240	
ip	int	11	
redirect	varchar	52	Key
bot_id	smallint	5	Key

stm_properties

Veld	Type	Lengte	Index
property_id	int	11	Primair
owner_id	mediumint	8	Key
domain	varchar	50	Key
path	mediumtext	Date	Key(100)
title	varchar	240	
meme_type	varchar	50	



stm_statuslog

Veld	Type	Lengte	Index
log_id	varchar	52	Primair
owner_id	mediumint	8	Key
original_url	mediumtext		Key(150)
domain	varchar	30	Key
path	mediumtext		Key(100)
date	datetime		Key
statuscode	smallint	3	
agent	varchar	240	
ip	int	11	
referer	mediumtext		Key(150)

stm_visits

Veld	Type	Lengte	Index
visit_id	varchar	52	Primair
owner_id	mediumint	8	Key
original_url	mediumtext		
domain	varchar	30	Key
path	mediumtext		Key(100)
date	datetime		Key
statuscode	smallint	3	
agent	varchar	240	
ip	int	11	
category	enum		Key
referer	mediumtext		
redirect	varchar	52	Key
searchengine	varchar	30	Key
keywords	mediumtext		Key(100)
index_position	smallint	4	
index_page	smallint	3	



5.3.2 Voorgestelde situatie

Verkleinen primair id

Synchroon trekken van tabellen

Verwijderen owner_id, domain, index_page

5.4 Inhoud

Opschonen stm statuslog



6 Databasemanagementsysteem

6.1 Technische eisen

- Het DBMS¹ moet goed gedocumenteerd zijn
 - Beschrijving van de mogelijkheden van het DMBS
 - Syntax
 - Functies
 - Systeemconfiguratie
 - Code voorbeelden
- Het DMBS moet actief door ontwikkeld worden en volledig ondersteund worden door haar ontwikkelaar
- Het DBMS moet profiling ondersteunen
- Het DBMS moet bereikbaar zijn in php, bij voorkeur via PDO²
- Het DBMS moet in staat zijn om gelijktijdig meerdere query's uit te voeren
- Het DBMS moet volledig beheert kunnen worden door OrangeValley

¹ Databasemanagementsysteem

² PHP Data Objects, een klasse gebruikt voor het opzetten van een consistente databaseconnectie.



6.2 Aandachtspunten

Executieplan

Het executieplan bepaalt hoe efficiënt het databasemanagementsysteem haar data ophaalt. Door prematuur en correct te filteren kan het aantal onnodige leesacties en calculaties, gedaan door het databasemanagementsysteem, aanzienlijk worden beperkt.

Configuratie

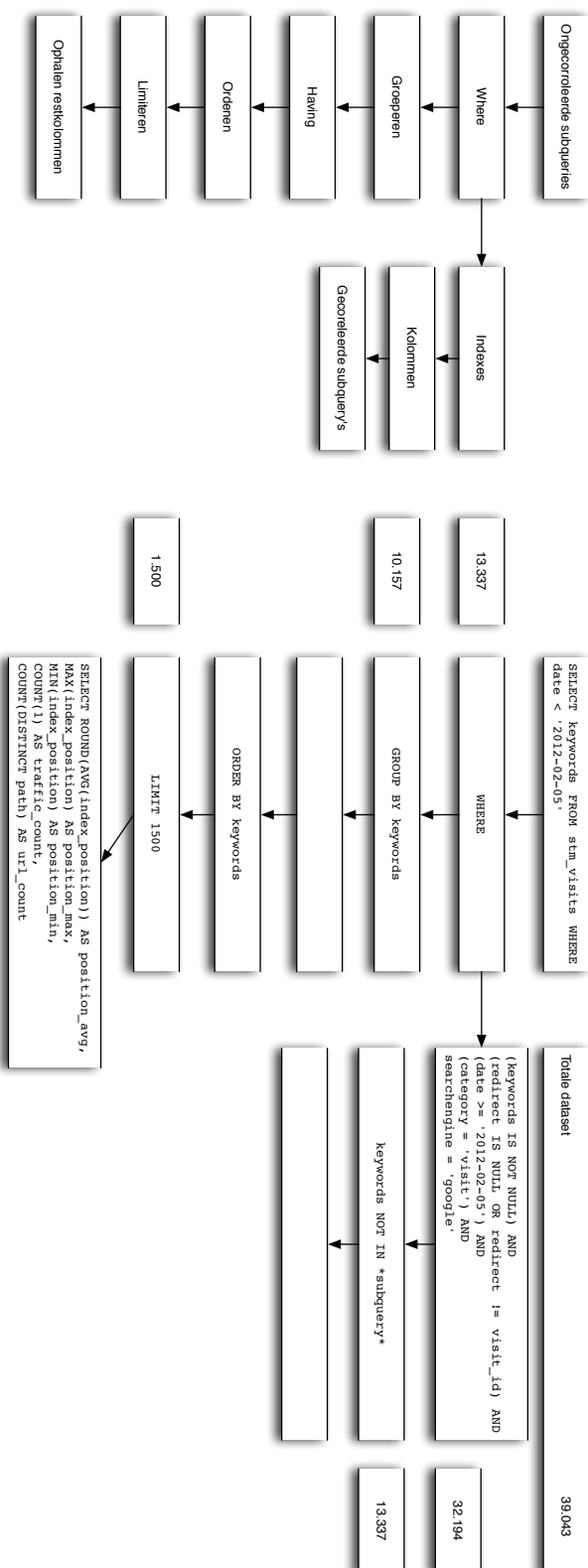
De configuratie bepaald hoeveel resources er aan databasemanagementsysteem worden toegewezen. Indien er teveel resources aan het databasemanagementsysteem worden toegewezen crashen er (cruciale) serverprocessen en wanneer er te weinig resources worden toegewezen is het databasemanagementsysteem traag.

De configuratie items die van belang zijn voor de snelheid van de database variëren aan de hand van de gekozen tabeltypes. Indien er in de database bijvoorbeeld veel gebruik wordt gemaakt van InnoDB tabellen heeft het geen zin om resources toe te wijzen aan MyISAM, dit heeft een averechts effect.



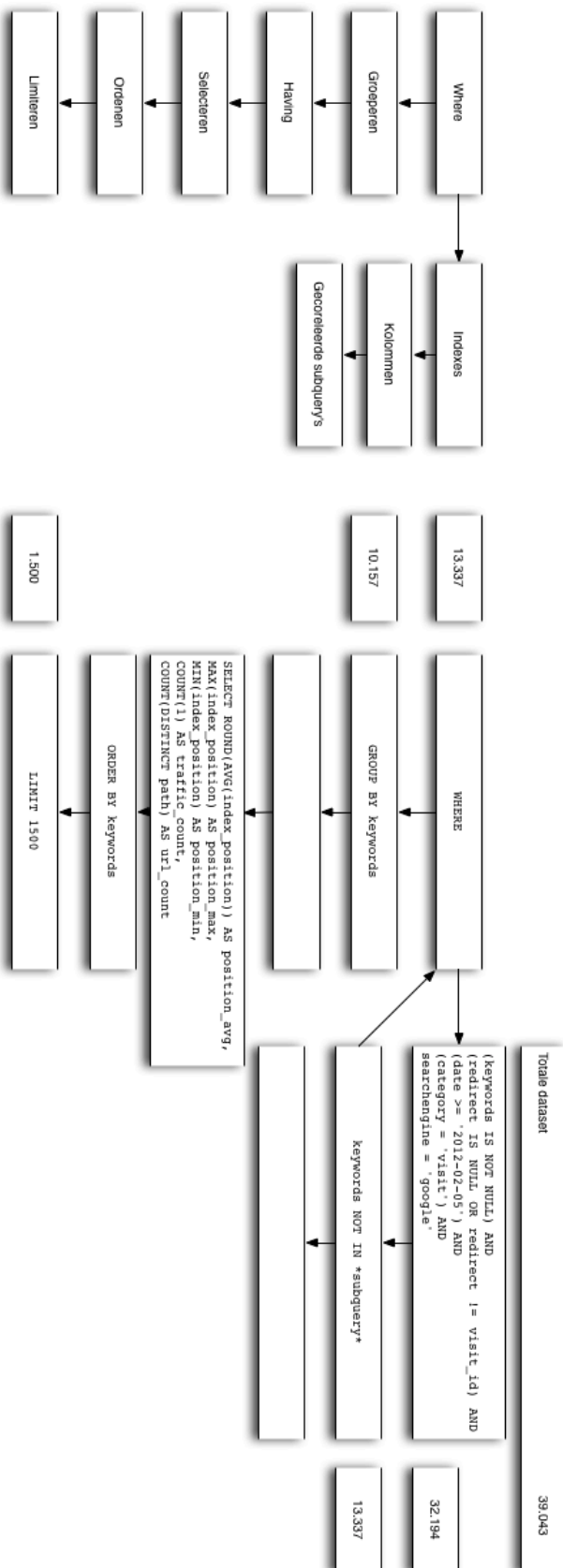
6.3 Executieplan

6.3.1 Optimale situatie





6.3.2 MySQL





6.4 Google BigQuery

Tijdens de Atmosphere 2011-conferentie heeft Google bekend gemaakt haar BigQuery-service ook beschikbaar te stellen voor bedrijven. Met de BigQuery-service kunnen bedrijven volgens Google snel grote hoeveelheden ruwe data analyseren. Zo stelt Google op haar website dat miljarden rijen aan data binnen enkele seconden kunnen worden geanalyseerd.

Een oplossing voor de performanceproblemen betreffende de database en het DBMS kan dan ook zijn door deze te vervangen met Google's BigQuery-service.

Nadelen, toch niet handig voor nu in haar huidige staat

Voordelen	Nadelen
Kan miljarden rijen binnen enkele seconden analyseren	Is niet in eigen beheer
	Kan niet uitgebreid worden met eigen functionaliteiten
	Is nog niet publiekelijk beschikbaar
	Maximum van 1000 query's per dag
	Elke query vraagt een timeout van 5 seconden
	Er zijn kosten aan verbonden ¹

6.5 MongoDB

6.6 CouchDB

¹ De specifieke kosten zijn nog niet gepubliceerd



7 Bijlage

7.1 Bronnen

Uitgever/locatie		Titel	Datum	Auteurs
1	O'Reilly Media	<i>High Performance MySQL</i>	juni 2008	Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy D. Zwaodny, Arjen Lentz & Derek J. Balling
2	http://tinyurl.com/2fuaeyk	<i>SQL Query Order of Operations'</i>	13 maart 2012	Ben Nadel



7.2 Systeemconfiguratie

7.2.1 Systeem

Variable	Waarde
Operating system	OSX 10.7.2
Processor	1.7 GHz Intel Core i5
Memory	4 GB 1333 MHz DDR3
MAMP	1.9
PHP	5.2.13
Apache	2.063
MySQL	5.1.44
Search Traffic Monitor repo	http://repo.orangevalley.nl/searchtrafficmonitor-com/trunk
Search Traffic Monitor revision	388
PieFramework repo	file:///Users/perdaan/Dropbox/SVN/framework-lan/trunk
PieFramework revision	118
PieFramework STM repo	http://repo.orangevalley.nl/searchtrafficmonitor-com/branches/piestm/module/stm
PieFramework STM revision	378

7.2.2 Database

Variable	Value
auto increment increment	1
auto increment offset	1
autocommit	ON
automatic sp privileges	ON
back log	50
basedir	/Applications/MAMP/Library/
big tables	OFF
binlog cache size	32,768
binlog direct non transactional updates	OFF
binlog format	STATEMENT
bulk insert buffer size	8,388,608
character set client	utf8
character set connection	utf8
character set database	latin1
character set filesystem	binary
character set results	utf8
character set server	latin1
character set system	utf8
character sets dir	/Applications/MAMP/Library/share/mysqlCharsets/
collation connection	utf8_general_ci
collation database	latin1_swedish_ci
collation server	latin1_swedish_ci
completion type	0
concurrent insert	1
connect timeout	10



datadir	/Applications/MAMP/db/mysql/
date format	%Y-%m-%d
datetime format	%Y-%m-%d %H:%i:%s
default week format	0
delay key write	ON
delayed insert limit	100
delayed insert timeout	300
delayed queue size	1,000
div precision increment	4
engine condition pushdown	ON
error count	0
event scheduler	OFF
expire logs days	0
flush	OFF
flush time	0
foreign key checks	ON
ft boolean syntax	+ -><()~*:""&
ft max word len	84
ft min word len	4
ft query expansion limit	20
ft stopword file	(built-in)
general log	OFF
general log file	/Applications/MAMP/tmp/mysql/mysql.log
group concat max len	1,024
have community features	YES
have compress	YES
have crypt	YES
have csv	YES
have dynamic loading	YES
have geometry	YES
have innodb	YES
have ndbcluster	DISABLED
have openssl	NO
have partitioning	YES
have query cache	YES
have rtree keys	YES
have ssl	NO
have symlink	YES
hostname	LaptopF.local
identity	0
ignore builtin innodb	OFF
init connect	
init file	
init slave	
innodb adaptive hash index	ON
innodb additional mem pool size	1,048,576
innodb autoextend increment	8
innodb autoinc lock mode	1



innodb buffer pool size	8,388,608
innodb checksums	ON
innodb commit concurrency	0
innodb concurrency tickets	500
innodb data file path	ibdata1:10M:autoextend
innodb data home dir	
innodb doublewrite	ON
innodb fast shutdown	1
innodb file io threads	4
innodb file per table	OFF
innodb flush log at trx commit	1
innodb flush method	
innodb force recovery	0
innodb lock wait timeout	50
innodb locks unsafe for binlog	OFF
innodb log buffer size	1,048,576
innodb log file size	5,242,880
innodb log files in group	2
innodb log group home dir	./
innodb max dirty pages pct	90
innodb max purge lag	0
innodb mirrored log groups	1
innodb open files	300
innodb rollback on timeout	OFF
innodb stats on metadata	ON
innodb support xa	ON
innodb sync spin loops	20
innodb table locks	ON
innodb thread concurrency	8
innodb thread sleep delay	10,000
innodb use legacy cardinality algorithm	ON
insert id	0
interactive timeout	28,800
join buffer size	131,072
keep files on create	OFF
key buffer size	8,384,512
key cache age threshold	300
key cache block size	1,024
key cache division limit	100
language	/Applications/MAMP/Library/share/mysql/english/
large files support	ON
large page size	0
large pages	OFF
last insert id	0
lc time names	en_US
license	GPL
local infile	ON



locked in memory	OFF
log	OFF
log bin	OFF
log bin trust function creators	OFF
log bin trust routine creators	OFF
log error	/Applications/MAMP/logs/mysql_error_log.err
log output	FILE
log queries not using indexes	OFF
log slave updates	OFF
log slow queries	OFF
log warnings	1
long query time	10
low priority updates	OFF
lower case file system	ON
lower case table names	0
max allowed packet	1,048,576
max binlog cache size	4,294,963,200
max binlog size	1,073,741,824
max connect errors	10
max connections	151
max delayed threads	20
max error count	64
max heap table size	16,777,216
max insert delayed threads	20
max join size	4,294,967,295
max length for sort data	1,024
max prepared stmt count	16,382
max relay log size	0
max seeks for key	4,294,967,295
max sort length	1,024
max sp recursion depth	0
max tmp tables	32
max user connections	0
max write lock count	4,294,967,295
min examined row limit	0
multi range count	256
myisam data pointer size	6
myisam max sort file size	2,146,435,072
myisam mmap size	4,294,967,295
myisam recover options	OFF
myisam repair threads	1
myisam sort buffer size	8,388,608
myisam stats method	nulls_unequal
myisam use mmap	OFF
ndb autoincrement prefetch sz	1
ndb cache check time	0
ndb connectstring	
ndb extra logging	0



ndb force send	ON
ndb index stat cache entries	32
ndb index stat enable	OFF
ndb index stat update freq	20
ndb report thresh binlog epoch slip	3
ndb report thresh binlog mem usage	10
ndb use copying alter table	OFF
ndb use exact count	ON
ndb use transactions	ON
net buffer length	16,384
net read timeout	30
net retry count	10
net write timeout	60
new	OFF
old	OFF
old alter table	OFF
old passwords	OFF
open files limit	2,560
optimizer prune level	1
optimizer search depth	62
optimizer switch	index_merge=on,index_merge_union=on,index_merge_sort_un ion=on,index_merge_intersection=on
pid file	/Applications/MAMP/tmp/mysql/mysql.pid
plugin dir	/Applications/MAMP/Library/lib/mysql/plugin
port	8,889
preload buffer size	32,768
profiling	OFF
profiling history size	15
protocol version	10
pseudo thread id	207
query alloc block size	8,192
query cache limit	1,048,576
query cache min res unit	4,096
query cache size	0
query cache type	ON
query cache wlock invalidate	OFF
query prealloc size	8,192
rand seed1	
rand seed2	
range alloc block size	4,096
read buffer size	131,072
read only	OFF
read rnd buffer size	262,144
relay log	
relay log index	
relay log info file	relay-log.info
relay log purge	ON



relay log space limit	0
report host	
report password	
report port	3,306
report user	
rpl recovery rank	0
secure auth	OFF
secure file priv	
server id	0
skip external locking	ON
skip networking	OFF
skip show database	OFF
slave compressed protocol	OFF
slave exec mode	STRICT
slave load tmpdir	/var/folders/hn/wkzskphs4ksdt28n04ft_mv00000gn/T/
slave net timeout	3,600
slave skip errors	OFF
slave transaction retries	10
slow launch time	2
slow query log	OFF
slow query log file	/Applications/MAMP/tmp/mysql/mysql-slow.log
socket	/Applications/MAMP/tmp/mysql/mysql.sock
sort buffer size	2,097,144
sql auto is null	ON
sql big selects	ON
sql big tables	OFF
sql buffer result	OFF
sql log bin	ON
sql log off	OFF
sql log update	ON
sql low priority updates	OFF
sql max join size	4,294,967,295
sql mode	
sql notes	ON
sql quote show create	ON
sql safe updates	OFF
sql select limit	4,294,967,295
sql slave skip counter	
sql warnings	OFF
ssl ca	
ssl capath	
ssl cert	
ssl cipher	
ssl key	
storage engine	MyISAM
sync binlog	0
sync frm	ON
system time zone	CET



table definition cache	256
table lock wait timeout	50
table open cache	64
table type	MyISAM
thread cache size	0
thread handling	one-thread-per-connection
thread stack	196,608
time format	%H:%i:%s
time zone	SYSTEM
timed mutexes	OFF
timestamp	1,329,383,603
tmp table size	16,777,216
tmpdir	/var/folders/hn/wkzskphs4ksdt28n04ft_mv00000gn/T/
transaction alloc block size	8,192
transaction prealloc size	4,096
tx isolation	REPEATABLE-READ
unique checks	ON
updatable views with limit	YES
version	5.1.44
version comment	Source distribution
version compile machine	i386
version compile os	apple-darwin8.11.1
wait timeout	28,800
warning count	0

Tabel	Rijen	Groote
stm_crawls	34.101	41.2 MB
stm_properties	359	0.13 MB
stm_statuslog	33.240	35.8 MB
stm_visits	39.043	76.4 MB
Totaal	465.384	153.53 MB

Database dump kan verstrekt worden op aanvraag



7.2.3 PHP

Variable	Waarde
Configure Command	<code>'./configure' '--with-mysql=/Applications/MAMP/Library' '--with-apxs2=/Applications/MAMP/Library/bin/apxs' '--with-gd' '--with-jpeg-dir=/Applications/MAMP/Library' '--with-png-dir=/Applications/MAMP/Library' '--with-zlib' '--with-freetype-dir=/Applications/MAMP/Library' '--prefix=/Applications/MAMP/bin/php5.2' '--exec-prefix=/Applications/MAMP/bin/php5.2' '--sysconfdir=/Applications/MAMP/conf/php5.2' '--with-soap' '--with-config-file-path=/Applications/MAMP/conf/php5.2' '--enable-track-vars' '--enable-bcmath' '--enable-ftp' '--enable-gd-native-ttf' '--with-bz2=/usr' '--with-ldap' '--with-mysqli=/Applications/MAMP/Library/bin/mysqli_config' '--with-sqlite' '--with-ttf' '--with-tlib=/Applications/MAMP/Library' '--enable-mbstring=all' '--with-curl=/Applications/MAMP/Library' '--enable-dbx' '--enable-sockets' '--enable-bcmath' '--with-imap=shared,/Applications/MAMP/Library/lib/imap-2007e' '--enable-soap' '--with-kerberos' '--enable-calendar' '--with-pgsql=shared,/Applications/MAMP/Library/pg' '--enable-dbase' '--enable-exif' '--with-libxml-dir=/Applications/MAMP/Library' '--with-gettext=shared,/Applications/MAMP/Library' '--with-xsl=/Applications/MAMP/Library' '--with-pdo-mysql=shared,/Applications/MAMP/Library' '--with-pdo-pgsql=shared,/Applications/MAMP/Library/pg' '--with-mcrypt=shared,/Applications/MAMP/Library' '--with-openssl'</code>
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MAMP/conf/php5.2
Loaded Configuration File	/Applications/MAMP/conf/php5.2/php.ini
Scan this dir for additional .ini files	(none)
additional .ini files parsed	(none)
PHP API	20041225
PHP Version	5.2.13
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, data, http, ftp
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
allow_call_time_pass_reference	On
allow_url_fopen	On
allow_url_include	Off
always_populate_raw_post_data	Off
arg_separator.input	&
arg_separator.output	&
asp_tags	Off
auto_append_file	no value



auto_globals_jit	On
auto_prepend_file	<i>no value</i>
browscap	<i>no value</i>
default_charset	<i>no value</i>
default_mimetype	text/html
define_syslog_variables	Off
disable_classes	<i>no value</i>
disable_functions	<i>no value</i>
display_errors	On
display_startup_errors	Off
doc_root	<i>no value</i>
docref_ext	<i>no value</i>
docref_root	<i>no value</i>
enable_dl	On
error_append_string	<i>no value</i>
error_log	/Applications/MAMP/logs/php_error.log
error_prepend_string	<i>no value</i>
error_reporting	8191
expose_php	On
extension_dir	/Applications/MAMP/bin/php5.2/lib/php/extensions/no-debug-non-zts-20060613/
file_uploads	On
highlight.bg	
highlight.comment	#FF8000
highlight.default	#0000BB
highlight.html	#000000
highlight.keyword	#007700
highlight.string	#DD0000
html_errors	On
ignore_repeated_errors	Off
ignore_repeated_source	Off
ignore_user_abort	Off
implicit_flush	Off
include_path	./:Applications/MAMP/bin/php5.2/lib/php
log_errors	On
log_errors_max_len	1024
magic_quotes_gpc	On
magic_quotes_runtime	Off
magic_quotes_sybase	Off
mail.force_extra_parameters	<i>no value</i>
max_execution_time	30
max_file_uploads	20
max_input_nesting_level	64
max_input_time	60
memory_limit	32M
open_basedir	<i>no value</i>
output_buffering	<i>no value</i>
output_handler	<i>no value</i>
post_max_size	32M
precision	12
realpath_cache_size	16K
realpath_cache_ttl	120
register_argc_argv	On
register_globals	Off
register_long_arrays	On
report_memleaks	On
report zend debug	On
safe_mode	Off



safe_mode_exec_dir	no value
safe_mode_gid	Off
safe_mode_include_dir	no value
sendmail_from	no value
sendmail_path	/usr/sbin/sendmail -t -i
serialize_precision	100
short_open_tag	Off
SMTP	localhost
smtp_port	25
sql.safe_mode	Off
track_errors	Off
unserialize_callback_func	no value
upload_max_filesize	32M
upload_tmp_dir	/Applications/MAMP/tmp/php
user_dir	no value
variables_order	EGPCS
xmlrpc_error_number	0
xmlrpc_errors	Off
y2k_compliance	On
zend.ze1_compatibility_mode	Off
Apache Version	Apache/2.0.63 (Unix) PHP/5.2.13 DAV/2
Apache API Version	20020903
Apache Server Administrator	you@example.com
Apache Hostname:Port	localhost:0
Apache User/Group	perdaan(501)/-1
Apache Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Apache Timeouts	Connection: 300 - Keep-Alive: 15
Apache Virtual Server	Yes
Apache Server Root	/Applications/MAMP/Library
Apache Loaded Modules	core prefork http_core mod_so mod_access mod_auth mod_auth_anon mod_auth_dbm mod_auth_digest mod_file_cache mod_echo mod_charset_lite mod_cache mod_disk_cache mod_mem_cache mod_example mod_case_filter mod_case_filter_in mod_ext_filter mod_include mod_deflate mod_log_config mod_env mod_mime_magic mod_cern_meta mod_expires mod_headers mod_usertrack mod_setenvif mod_proxy proxy_connect proxy_ftp proxy_http mod_bucketeer mod_mime mod_dav mod_status mod_autoindex mod_asis mod_info mod_cgi mod_cgid mod_dav_fs mod_vhost_alias mod_negotiation mod_dir mod_imap mod_actions mod_speling mod_userdir mod_alias mod_rewrite mod_php5
Apache engine	1
Apache last_modified	0
Apache xbithack	0
BCMath support	enabled
BZip2 Support	Enabled
BZip2 Stream Wrapper support	compress.bz2://
BZip2 Stream Filter support	bzip2.decompress, bzip2.compress
BZip2 Version	1.0.6, 6-Sept-2010
Calendar support	enabled
ctype functions	enabled
cURL support	enabled
cURL Information	libcurl/7.20.0 OpenSSL/0.9.7l zlib/1.2.5 libidn/1.15
Date date/time support	enabled
Date "Olson" Timezone	2010.2
Database Version	
Date Timezone Database	internal
date.default_latitude	31.7667



date.default_longitude	35.2333
date.sunrise_zenith	90.583333
date.sunset_zenith	90.583333
date.timezone	Europe/Amsterdam
DOM/XML	enabled
DOM/XML API Version	20031129
DOM libxml Version	2.7.6
DOM HTML Support	enabled
DOM XPath Support	enabled
DOM XPointer Support	enabled
DOM Schema Support	enabled
DOM RelaxNG Support	enabled
EXIF Support	enabled
EXIF Version	1.4 \$Id: exif.c 293036 2010-01-03 09:23:27Z sebastian \$
EXIF Supported EXIF Version	0220
EXIF Supported filetypes	JPEG,TIFF
Filter Input Validation and Filtering	enabled
Filter Revision	\$Revision: 293051 \$
Filter filter.default	unsafe_raw
Filter filter.default_flags	no_value
FTP support	enabled
GD Support	enabled
GD Version	bundled (2.0.34 compatible)
GD FreeType Support	enabled
GD FreeType Linkage	with freetype
GD FreeType Version	2.3.9
GD T1Lib Support	enabled
GD GIF Read Support	enabled
GD GIF Create Support	enabled
GD JPG Support	enabled
GD PNG Support	enabled
GD WBMP Support	enabled
GD XBM Support	enabled
GetText Support	enabled
hash support	enabled
Hashing Engines	md2 md4 md5 sha1 sha256 sha384 sha512 ripemd128 ripemd160 ripemd256 ripemd320 whirlpool tiger128,3 tiger160,3 tiger192,3 tiger128,4 tiger160,4 tiger192,4 snfru gost adler32 crc32 crc32b haval128,3 haval160,3 haval192,3 haval224,3 haval256,3 haval128,4 haval160,4 haval192,4 haval224,4 haval256,4 haval128,5 haval160,5 haval192,5 haval224,5 haval256,5
iconv support	enabled
iconv implementation	libiconv
iconv library version	1.13
iconv support	enabled
iconv implementation	libiconv
iconv library version	1.13
iconv.input_encoding	ISO-8859-1
iconv.internal_encoding	ISO-8859-1
iconv.output_encoding	ISO-8859-1
IMAP c-Client Version	2007e
IMAP Kerberos Support	enabled
json support	enabled
json version	1.2.1
LDAP Support	enabled
LDAP RCS Version	\$Id: ldap.c 293036 2010-01-03 09:23:27Z sebastian \$



LDAP Total Links	0/unlimited
LDAP API Version	3001
LDAP Vendor Name	OpenLDAP
LDAP Vendor Version	20219
libXML support	active
libXML Version	2.7.6
libXML streams	enabled
Multibyte Support	enabled
Multibyte string engine	libmbfl
Multibyte (japanese) regex support	enabled
Multibyte regex (oniguruma) version	4.4.4
Multibyte regex (oniguruma) backtrack check	On
mbstring.detect_order	no value
mbstring.encoding_translation	Off
mbstring.func_overload	0
mbstring.http_input	pass
mbstring.http_output	pass
mbstring.internal_encoding	no value
mbstring.language	neutral
mbstring.strict_detection	Off
mbstring.substitute_character	no value
Mycrypt Version	2.5.8
Mycrypt Api No	20021217
Mcrypt Supported ciphers	cast-128 gost rijndael-128 twofish arcfour cast-256 loki97 rijndael-192 saferplus wake blowfish-compat des rijndael-256 serpent xtea blowfish enigma rc2 tripledes
Mcrypt Supported modes	cbc cfb ctr ecb ncfb nofb ofb stream
mcrypt.algorithms_dir	no value
mcrypt.modes_dir	no value
Active Persistent Links	0
Active Links	0
Client API version	5.1.44
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/Applications/MAMP/tmp/mysql/mysql.sock
MYSQL_INCLUDE	-I/Applications/MAMP/Library/include/mysql
MYSQL_LIBS	-L/Applications/MAMP/Library/lib/mysql -lmysqlclient
mysql.allow_persistent	On
mysql.connect_timeout	60
mysql.default_host	no value
mysql.default_password	no value
mysql.default_port	no value
mysql.default_socket	/Applications/MAMP/tmp/mysql/mysql.sock
mysql.default_user	no value
mysql.max_links	Unlimited
mysql.max_persistent	Unlimited
mysql.trace_mode	Off
MYSQLI Client API library version	5.1.44
MYSQLI Client API header version	5.1.44
MYSQLI_SOCKET	/Applications/MAMP/tmp/mysql/mysql.sock
mysqli.default_host	no value
mysqli.default_port	3306
mysqli.default_pw	no value
mysqli.default_socket	no value



mysqli.default_user	no value
mysqli.max_links	Unlimited
mysqli.reconnect	Off
OpenSSL support	enabled
OpenSSL Version	OpenSSL 0.9.7i 14 Oct 2005
PCRE (Perl Compatible Regular Expressions) Support	enabled
PCRE Library Version	7.9 2009-04-11
pcre.backtrack_limit	100000
pcre.recursion_limit	100000
PDO drivers	sqlite, sqlite2, pgsql, mysql
PDO_PGSQL PostgreSQL(libpq) Version	8.4.2
PDO_PGSQL Module version	1.0.2
PDO_PGSQL Revision	\$Id: pdo_pgsql.c 293036 2010-01-03 09:23:27Z sebastian \$
PDO_SQLITE PECL Module version	(bundled) 1.0.1 \$Id: pdo_sqlite.c 293036 2010-01-03 09:23:27Z sebastian \$
PDO_SQLITE SQLite Library	3.3.7
PGSQL PostgreSQL(libpq) Version	8.4.2
PGSQL Multibyte character support	enabled
PGSQL SSL support	enabled
PGSQL Active Persistent Links	0
PGSQL Active Links	0
pgsql.allow_persistent	On
pgsql.auto_reset_persistent	Off
pgsql.ignore_notice	Off
pgsql.log_notice	Off
pgsql.max_links	Unlimited
pgsql.max_persistent	Unlimited
Posix Revision	\$Revision: 293036 \$
Reflection Version	\$Id: php_reflection.c 293850 2010-01-22 14:21:17Z jani \$
Session Support	enabled
Session Registered save handlers	files user sqlite
Session Registered serializer handlers	php php_binary
session.auto_start	Off
session.bug_compat_42	On
session.bug_compat_warn	On
session.cache_expire	180
session.cache_limiter	nocache
session.cookie_domain	no value
session.cookie_httponly	Off
session.cookie_lifetime	0
session.cookie_path	/
session.cookie_secure	Off
session.entropy_file	no value
session.entropy_length	0
session.gc_divisor	100
session.gc_maxlifetime	1440
session.gc_probability	1
session.hash_bits_per_character	4
session.hash_function	0
session.name	PHPSESSID



session.referer_check	no value
session.save_handler	files
session.save_path	/Applications/MAMP/tmp/php
session.serialize_handler	php
session.use_cookies	On
session.use_only_cookies	Off
session.use_trans_sid	0
SimpleXML Revision	\$Revision: 293036 \$
SimpleXML Schema support	enabled
Soap Client	enabled
Soap Server	enabled
soap.wsdl_cache	1
soap.wsdl_cache_dir	/tmp
soap.wsdl_cache_enabled	1
soap.wsdl_cache_limit	5
soap.wsdl_cache_ttl	86400
Sockets Support	enabled
SPL Interfaces	Countable, OuterIterator, RecursiveIterator, SeekableIterator, SplObserver, SplSubject
SPL Classes	AppendIterator, ArrayIterator, ArrayObject, BadFunctionCallException, BadMethodCallException, CachingIterator, DirectoryIterator, DomainException, EmptyIterator, FilterIterator, InfiniteIterator, InvalidArgumentException, IteratorIterator, LengthException, LimitIterator, LogicException, NoRewindIterator, OutOfBoundsException, OutOfRangeException, OverflowException, ParentIterator, RangeException, RecursiveArrayIterator, RecursiveCachingIterator, RecursiveDirectoryIterator, RecursiveFilterIterator, RecursiveIteratorIterator, RecursiveRegexIterator, RegexIterator, RuntimeException, SimpleXMLIterator, SplFileInfo, SplFileObject, SplObjectStorage, SplTempFileObject, UnderflowException, UnexpectedValueException
SQLite PECL Module version	2.0-dev \$Id: sqlite.c 293036 2010-01-03 09:23:27Z sebastian \$
SQLite Library	2.8.17
SQLite Encoding	iso8859
sqlite.assoc_case	0
Regex Library	Bundled library enabled
Dynamic Library Support	enabled
Path to sendmail	/usr/sbin/sendmail -t -i
assert.active	1
assert.bail	0
assert.callback	no value
assert.quiet_eval	0
assert.warning	1
auto_detect_line_endings	0
default_socket_timeout	60
safe_mode_allowed_env_vars	PHP_
safe_mode_protected_env_vars	LD_LIBRARY_PATH
url_rewriter.tags	a=href,area=href,frame=src,input=src,form=,fieldset=
user_agent	no value
Tokenizer Support	enabled
XCache Version	1.2.2
XCache Modules Built	catcher optimizer coverager assembler encoder decoder
XCache Readonly Protection	N/A
XCache Opcode Cache	enabled, 67,108,864 bytes, 1 split(s), with 8192 slots each



XCache Variable Cache	enabled, 67,108,864 bytes, 1 split(s), with 8192 slots each
XCache Shared Memory Schemes	mmap
XCache Coverage Auto Dumper	disabled
xcache.admin.enable_auth	On
xcache.cacher	On
xcache.coredump_directory	/Applications/MAMP/tmp/phpcore/
xcache.count	1
xcache.coveragedump_directory	no value
xcache.coverager	Off
xcache.gc_interval	0
xcache.mmap_path	/Applications/MAMP/tmp/xcache
xcache.optimizer	On
xcache.readonly_protection	no value
xcache.shm_scheme	mmap
xcache.size	64M
xcache.slots	8K
xcache.stat	On
xcache.test	no value
xcache.ttl	0
xcache.var_count	1
xcache.var_gc_interval	300
xcache.var_maxttl	0
xcache.var_size	64M
xcache.var_slots	8K
xcache.var_ttl	0
XML Support	active
XML Namespace Support	active
XML libxml2 Version	2.7.6
XMLReader	enabled
XMLWriter	enabled
XSL	enabled
XSL libxslt Version	1.1.26
XSL libxslt compiled against libxml Version	2.7.6
XSL EXSLT	enabled
XSL libexslt Version	1.1.26
YAZ Support	enabled
YAZ PHP/YAZ Version	1.0.14
YAZ Version	4.0.1
YAZ Compiled with YAZ version	4.0.1
ZLib Support	enabled
ZLib Stream Wrapper support	compress.zlib://
ZLib Stream Filter support	zlib.inflate, zlib.deflate
ZLib Compiled Version	1.2.3
ZLib Linked Version	1.2.5
zlib.output_compression	Off
zlib.output_compression_level	-1
zlib.output_handler	no value
Dbase	enabled



7.3 Query rapport

Inhoudsopgave

1	Notities	182
2	Visits	184
3	Crawls	185
4	Pagina's	186
5	Keywords	188
6	Clickdistributie	189
7	Sitemap	192



1 Notities

- Query's worden standaard gelimiteerd op 500 rijen in verband met het beperkte werkgeheugen beschikbaar in PHP.
- Er staat vaak een extra filter aan op basis van de zoekmachine, deze zijn in de nieuwe query sets echter nog niet geïmplementeerd.
- Url specifieke rapporten zijn doorgaans een samenstelling van rapporten met een onzichtbaar filter op het pad.
- Sommige query's zijn in staat om een nieuw variant te hebben, dit is een extra filter op de query op basis van een specifiek veld. De subquery erft doorgaans zijn filters van de primaire query.

```
SELECT
    CONCAT("http://", domain, path) AS `url`, GROUP_CONCAT(UNIX_TIMESTAMP(date) ORDER BY
    date) AS `visit_rate`,
    COUNT(1) AS `visit_count`, UNIX_TIMESTAMP(MIN(date)) AS `visit_first`,
    UNIX_TIMESTAMP(MAX(date)) AS `visit_last`, SUM(statuscode LIKE "3%") AS `statuscode_redirect`,
    SUM(statuscode LIKE "4%") + SUM(statuscode LIKE "5%") AS `statuscode_error`
FROM
    `stm_visits`
WHERE
    (redirect IS NULL OR redirect != visit_id) AND
    (date >= '2012-02-16') AND
    (path NOT IN (
        SELECT
            path
        FROM
            stm_visits
        WHERE
            date < '2012-02-16' AND
            (redirect IS NULL OR redirect != visit_id)
    ))
GROUP BY path ORDER BY visit_count DESC,visit_last DESC,path LIMIT 500
```



2 Visits

Type	Statuscode
Filters	Datum, Zoekmachine, Pad, Zoekterm
Opmerkingen	-
Query	<pre>SELECT `searchengine` AS `field`, COUNT(1) AS `amount` FROM `stm_visits` WHERE (redirect IS NULL OR redirect != visit_id) AND (date >= '2012-02-16') GROUP BY searchengine LIMIT 500</pre>
Type	Lijst
Filters	Datum, Zoekmachine, Pad, Zoekterm, Custom (statuscode, visit_count)
Opmerkingen	visit_rate moet plaats maken voor een custom functie in de database, nieuw variant mogelijk op basis van pad
Query	<pre>SELECT CONCAT("http://", domain, path) AS `url`, GROUP_CONCAT(UNIX_TIMESTAMP(date) ORDER BY date) AS `visit_rate`, COUNT(1) AS `visit_count`, UNIX_TIMESTAMP(MIN(date)) AS `visit_first`, UNIX_TIMESTAMP(MAX(date)) AS `visit_last`, SUM(statuscode LIKE "3%") AS `statuscode_redirect`, SUM(statuscode LIKE "4%") + SUM(statuscode LIKE "5%") AS `statuscode_error` FROM `stm_visits` WHERE (redirect IS NULL OR redirect != visit_id) AND (date >= '2012-02-16') GROUP BY path ORDER BY visit_count DESC,visit_last DESC,path LIMIT 500</pre>



3 Crawls

Type	Statuscode
Filters	Datum, Zoekmachine, Pad
Opmerkingen	-
Query	<pre>SELECT `bot_id` AS `field`, COUNT(1) AS `amount` FROM `stm_crawls` WHERE (redirect IS NULL OR redirect != crawl_id) AND (date >= '2012-02-16') GROUP BY bot_id LIMIT 500</pre>

Type	Lijst
Filters	Datum, Zoekmachine, Pad, Custom (statuscode, visit_count)
Opmerkingen	crawl_rate moet plaats maken voor een custom functie in de database, nieuw variant mogelijk op basis van pad
Query	<pre>SELECT CONCAT("http://", domain, path) AS `url`, GROUP_CONCAT(UNIX_TIMESTAMP(date) ORDER BY date) AS `crawl_rate`, COUNT(1) AS `crawl_count`, UNIX_TIMESTAMP(MIN(date)) AS `crawl_first`, UNIX_TIMESTAMP(MAX(date)) AS `crawl_last`, SUM(statuscode LIKE "3%") AS `statuscode_redirect`, SUM(statuscode LIKE "4%") + SUM(statuscode LIKE "5%") AS `statuscode_error` FROM `stm_crawls` WHERE (redirect IS NULL OR redirect != crawl_id) AND (date >= '2012-02-16') GROUP BY path ORDER BY crawl_count DESC, crawl_last DESC, path LIMIT 500</pre>



4 Pagina's

Type	Lijst
Filters	Datum, Zoekmachine, Pad
Opmerkingen	Een mediaan van de index positie is tevens gewenst, nieuw variant mogelijk op basis van pad
Query	<pre> SELECT CONCAT("http://", domain, path) AS `url`, ROUND(AVG(index_position)) AS `position_avg`, MAX(index_position) AS `position_max`, MIN(index_position) AS `position_min`, COUNT(1) AS `traffic_count`, COUNT(DISTINCT keywords) AS `keyword_count` FROM `stm_visits` WHERE (redirect IS NULL OR redirect != visit_id) AND (date >= '2012-02-16') GROUP BY path ORDER BY traffic_count DESC, path ASC LIMIT 500 </pre>

1

Type	Trending lijst
Filters	Datum, Zoekmachine, Pad
Opmerkingen	-
Query	<pre> SELECT CONCAT("http://", t1.domain, t1.path) AS `url`, ROUND(AVG(t1.index_position)) AS `position_base`, COUNT(DISTINCT t1.visit_id) AS `traffic_base`, COUNT(DISTINCT t1.keywords) AS `keyword_base`, ROUND(IF(AVG(t2.index_position) IS NOT NULL, AVG(t2.index_position), 0) - AVG(t1.index_position)) AS `position_diff`, COUNT(DISTINCT t1.visit_id) - COUNT(DISTINCT t2.visit_id) AS `traffic_diff`, COUNT(DISTINCT t2.keywords) - COUNT(DISTINCT t1.keywords) AS `keyword_diff` FROM `stm_visits` AS `t1` LEFT JOIN `stm_visits` AS `t2` ON t2.path = t1.path AND t2.date >= "2012-02-02 00:00:00" AND t2.date < "2012-02-16 00:00:00" WHERE (t1.redirect IS NULL OR t1.redirect != t1.visit_id) AND (t1.date >= '2012-02-16') GROUP BY t1.path ORDER BY traffic_diff DESC, t1.path ASC LIMIT 500 </pre>





5 Keywords

Type	Lijst
Filters	Datum, Zoekmachine, Pad, Zoekterm
Opmerkingen	Een mediaan van de index positie is tevens gewenst, nieuw variant mogelijk op basis van keywords
Query	<pre>SELECT `keywords`, ROUND(AVG(index_position)) AS `position_avg`, MAX(index_position) AS `position_max`, MIN(index_position) AS `position_min`, COUNT(1) AS `traffic_count`, COUNT(DISTINCT path) AS `url_count` FROM `stm_visits` WHERE (keywords IS NOT NULL) AND (redirect IS NULL OR redirect != visit_id) AND (date >= '2012-02-16')) GROUP BY keywords ORDER BY traffic_count DESC, keywords ASC LIMIT 500</pre>

Type	Trending lijst
Filters	Datum, Zoekmachine, Pad, Zoekterm
Opmerkingen	-
Query	<pre>SELECT `t1`.`keywords`, ROUND(AVG(t1.index_position)) AS `position_base`, COUNT(DISTINCT t1.visit_id) AS `traffic_base`, ROUND(IF(AVG(t2.index_position) IS NOT NULL, AVG(t2.index_position), 0) - AVG(t1.index_position)) AS `position_diff`, COUNT(DISTINCT t1.visit_id) - COUNT(DISTINCT t2.visit_id) AS `traffic_diff` FROM `stm_visits` AS `t1` LEFT JOIN `stm_visits` AS `t2` ON t2.keywords = t1.keywords AND t2.date >= "2012-02-02 00:00:00" AND t2.date < "2012-02-16 00:00:00" WHERE (t1.keywords IS NOT NULL) AND (t1.redirect IS NULL OR t1.redirect != t1.visit_id) AND (t1.date >= '2012-02-16') GROUP BY t1.keywords ORDER BY traffic_diff DESC, position_diff DESC, keywords ASC LIMIT 500</pre>





6 Clickdistributie

Type	Visits
Filters	Pad, Keyword, Source
Opmerkingen	Rapport wordt opgebouwd uit twee query's
Query	<pre>SELECT date, ROUND(SUM(position * amount) / SUM(amount)) AS position, MAX(amount) / SUM(amount) AS significance FROM (SELECT DATE(date) AS `date`, index_position AS `position`, COUNT(1) AS `amount` FROM `stm_visits` WHERE keywords IS NOT NULL AND index_position IS NOT NULL AND keywords LIKE '%keyword%' GROUP BY DATE(date), `index_position` ORDER BY `amount` ASC, `index_position` ASC, `date` ASC) t GROUP BY date ORDER BY date -- verwerkt output naar ander formaat SELECT DATE(date) AS `date`, COUNT(1) AS `amount` FROM `stm_visits` WHERE DATE(date) IN (-- datums uit bovenstaande query --) AND keywords IS NOT NULL AND (redirect IS NULL OR v.redirect != visit_id) AND keywords LIKE '%keyword%' GROUP BY DATE(date) ORDER BY `date` ASC</pre>



Type	Crawling
Filters	Pad, Datum, Source
Opmerkingen	Rapport wordt opgebouwd uit twee query's
Query	<pre>SELECT date, ROUND(SUM(position * amount) / SUM(amount)) AS position, MAX(amount) / SUM(amount) AS significance FROM (SELECT DATE(date) AS `date`, `index_position` AS `position`, COUNT(1) AS `amount` FROM `stm_visits` WHERE keywords IS NOT NULL AND index_position IS NOT NULL AND (redirect IS NULL OR redirect != visit_id) AND (path LIKE '%pad%') GROUP BY DATE(date), `index_position` ORDER BY `amount` ASC, `index_position` ASC, `date` ASC) t GROUP BY date ORDER BY date -- verwerkt output naar ander formaat SELECT DATE(date) AS `date`, COUNT(1) AS `amount` FROM `stm_crawls` WHERE DATE(date) IN (-- datums uit bovenstaande query --) AND (redirect IS NULL OR redirect != crawl_id) AND (date >= '2012-02-16 00:00:00') AND (path LIKE '%pad%') GROUP BY DATE(date) ORDER BY `date` ASC</pre>



7 Sitemap

Type	Lijst
Filters	Pad, Datum
Opmerkingen	-
Query	<pre>SELECT `p`.`domain`, `p`.`path`, MAX(p.date) AS `last_update`, COUNT(c.crawl_id) AS `crawled_amount`, MIN(c.date) AS `crawled_first` FROM `stm_properties` AS `p` LEFT JOIN `stm_crawls` AS `c` ON c.path = p.path AND c.date >= '2012-02-16 00:00:00' GROUP BY `p`.`path` ORDER BY `crawled_first` DESC LIMIT 500</pre>

Type	Status
Filters	Pad, Datum
Opmerkingen	Toont het aantal gecrawlde paden op basis van de sitemap
Query	<pre>SELECT COUNT(DISTINCT path) FROM `stm_properties` SELECT COUNT(DISTINCT path) FROM `stm_properties` WHERE (path IN (SELECT `path` FROM `stm_crawls` WHERE (date >= '2012-02-16 00:00:00')))</pre>



G Afstudeerplan



Afstudeerplan

Informatie afstudeerder en gastbedrijf *(structuur niet wijzigen)*

Afstudeerblok: 2012-1.1 (start uiterlijk 6 februari 2012)

Startdatum uitvoering afstudeeropdracht: 6 februari 2012

Inleverdatum afstudeerdossier volgens jaarrooster: 1 juni 2012

Studentnummer: 08011249

Achternaam: dhr Perdaan

() weghalen niet van toepassing*

Voorletters: F

Roepnaam: Ferdy

Adres: Zijlberg 47

Postcode: 2716NB

Woonplaats: Zoetermeer

Telefoonnummer: 079 3520 494

Mobiel nummer: 06 480 420 85

Privé emailadres: f.perdaan@hotmail.com

Opleiding: Informatica

Locatie: Zoetermeer

() weghalen niet van toepassing*

Variant: voltijd

Naam studieloopbaanbegeleider: A.A. Nederend

Naam begeleidend examiner: A.A. Nederend

Naam tweede examiner: A.A.A.M. Jacobs

Naam bedrijf: OrangeValley

Afdeling bedrijf: Software development

Bezoekadres bedrijf: De Corridor 27

Postcode bezoekadres: 3621ZA

Postbusnummer:

Postcode postbusnummer:

Plaats: Breukelen

Telefoon bedrijf: 31 308 200 200

Telefax bedrijf: 31 308 200 208

Internetsite bedrijf: <http://orangevalley.nl>

Achternaam opdrachtgever: dhr Verreck

() weghalen niet van toepassing*

Voorletters opdrachtgever: O

Titulatuur opdrachtgever: Doctorandus

Functie opdrachtgever: Managing Partner

Doorkiesnummer opdrachtgever:

Email opdrachtgever: ortwin@orangevalley.nl

Achternaam bedrijfsmentor: dhr Blacquièr

() weghalen niet van toepassing*

Voorletters bedrijfsmentor: E

Titulatuur bedrijfsmentor: Bachelor of Business Administration

Functie bedrijfsmentor: Senior SEO Consultant

Doorkiesnummer bedrijfsmentor:

Email bedrijfsmentor: eduard@orangevalley.nl

Doorkiesnummer afstudeerder:

Functie afstudeerder (deeltijd/duaal): Technical Consultant



Titel afstudeeropdracht:

Het verder door ontwikkelen van het reeds bestaande softwaresysteem Search Traffic Monitor.

Opdrachtomschrijving

1. Bedrijf

OrangeValley is een bedrijf dat zich specialiseert in het optimaliseren van online conversie. Hierbij worden de websites van klanten geanalyseerd om zo een advies te geven ter verbetering van hun online rendement. Indien gewenst kan de klant er voor kiezen om deze adviezen in de praktijk toe te passen door hun SEO¹ en/of SEA² campagnes door OrangeValley te laten beheren. Aansluitend op de conversie optimalisatie biedt OrangeValley haar klanten ook multi variatie testen en het bouwen van websites aan, dit wordt echter gedaan onder de holding SiteSpect (testen) en 42functions (bouwen).

OrangeValley bestaat nu ruim drie jaar met een team van circa vijftientig medewerkers. Het bedrijf focust zich op middelgrote tot grote bedrijven, omdat deze vaak een groter budget hebben voor het verbeteren van hun conversietraject. Kleine bedrijven zien vaak nog niet in dat de optimalisatie zich ruimschoots terug verdiend. Enkele klanten van OrangeValley zijn: Independer, VNU, ANWB, ING (SiteSpect), Slagharen (42functions) en Movares (42functions).

2. Probleemstelling

Gedurende mijn afstuderen hou ik mij bezig met het verder ontwikkelen van Search Traffic Monitor, één van OrangeValley's innovatie projecten gefinancierd door de WBSO subsidie. Search Traffic Monitor is een systeem welke inzicht geeft in het STRICA³ traject. Het STRICA traject toont de effecten in crawling gedrag, indexering, ranking, traffic en sales na een aanpassing op een website.

OrangeValley heeft het doel om Search Traffic Monitor aan de standaard toolkit van haar consultants toe te voegen. Voor het realiseren van dit doel is het noodzakelijk dat de performance problemen waar Search Traffic Monitor momenteel nog last van heeft worden opgelost. Na het oplossen van deze performanceproblemen is het gewenst dat Search Traffic Monitor in staat is om het gehele STRICA traject in kaart te brengen. Momenteel mist Search Traffic Monitor echter nog de functionaliteit om inzicht te geven in de onderdelen Indexing en Sales.

¹ Search Engine Optimization

² Search Engine Advertising

³ Sales Traffic Ranking Indexing Crawling Adjustment



3. Doelstelling van de afstudeeropdracht

Aan het einde van de afstudeerperiode moet er een adviesrapport liggen voor het verbeteren van de performance. Afhankelijk van de prioriteit die de opdrachtgever toekent aan de verschillende verbetervoorstellen worden deze voorstellen gedurende het afstuderen gerealiseerd.

Verder moet Search Traffic Monitor in staat zijn om ook het sales gedeelte in kaart te brengen. Dit wil zeggen dat er SEA module aan het reeds bestaande systeem gekoppeld moet worden. Deze module bevat onder andere een koppeling met Google AdWords. De data uit Google AdWords wordt vervolgens met de reeds verzamelde data gecombineerd om zo tot nieuwe inzichten te komen.

4. Resultaat

Search Traffic Monitor bevat een nieuwe SEA module welke in staat is om het effect op de sales na een aanpassing op de website in kaart te brengen. Verder moet er een adviesrapport betreffende performance aan het management voorgedragen zijn. Dit adviesrapport bevat onder andere oplossingen voor de performanceproblemen binnen Search Traffic Monitor.



5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

De afstudeerperiode bestaat uit twee fasen. Tijdens de eerste fase wordt het plan van aanpak, de product backlog en het adviesrapport ten behoeve van de performance opgesteld. Voor de tweede fase wordt er gebruik gemaakt van de SCRUM ontwikkelmethodiek, dit in verband met de voortdurend wijzigende requirements en de prioritering die aan deze requirements worden toegewezen. Omdat ik de enige actieve developer ben binnen het project worden de dagelijkse meetings ingepland met de bedrijfsmentor. In deze dagelijkse meetings wordt de voortgang van het project, afstuderen en de actieve sprint besproken. Aan het eind van elke tweewekelijkse sprint wordt er een sprint meeting gehouden met de opdrachtgever om zo de doelen van de volgende sprint vast te stellen.

De volgende activiteiten worden uitgevoerd gedurende afstudeeropdracht:

- Opstellen plan van aanpak
- Opstellen adviesrapport betreffende performanceproblemen
 - o Analyse naar bottlenecks
 - Database
 - DBMS
 - Server
 - Systeem
 - o Oplossingen voor gevonden bottlenecks
 - o Advies waarin de voor en nadelen van de oplossingen tegen elkaar worden afgewogen om zo de meest geschikte oplossing aan te bevelen
- Uitbreiden product backlog met gewenste SEA functionaliteiten
- Ontwerpen specifieke statistische functionaliteit
- Integreren specifieke statistische functionaliteit
- Testen statistische functionaliteit

Iteratief

- Ontwerpen nieuwe functionaliteit
- Integreren nieuwe functionaliteit
- Testen nieuwe functionaliteit
- Opstellen burndown chart

6. Op te leveren (tussen)producten

De tussenproducten staan reeds gespecificeerd in het bovenstaande paragraaf.



7. Te demonstreren competenties en wijze waarop

1.3 Selecteren van standaardsoftware

Adviseren over en/of selecteren van software, zoals een applicatie (commerciële software of open source), framework (al dan niet gebaseerd op open standaarden), databasemanagementsysteem of besturingssysteem.

Een deel van mijn afstuderen besteed ik aan het analyseren van de performanceproblemen binnen Search Traffic Monitor. Binnen deze analyse wordt er gekeken naar de structuur van de database, het gebruikte database management systeem, de server waarop het systeem draait en ten slotte het systeem zelf. Aan de hand van deze analyse worden er vervolgens oplossingen en bijbehorende adviezen voorgesteld opdat het management vervolgens een gegronde keuze kan maken.

3.2 Ontwerpen systeemdeel

Beschrijven van systeemdelen (subsystemen, componenten, modules), hun onderliggende structuur en het gedrag in detail, zodanig dat bouwen van het systeemdeel mogelijk is. Ontwerpen (grafische) user interface.

Omdat er gebruik wordt gemaakt van een framework wordt de architectuur van de SEA module al afgedwongen. Er worden voor deze competentie dan ook enkel klassendiagrammen, use-cases, sequentiediagrammen en grafische ontwerpen opgesteld. Deze worden terug gekoppeld aan de opdrachtgever om zo een duidelijk beeld van de gewenste functionaliteit te vormen.

3.3 Bouwen applicatie

Verfijnen en/of transformeren van het (detail)ontwerp van de systeemdelen (subsystemen, componenten, modules) van een applicatie. Bouwen en documenteren van de systeemdelen. Systeemdelen samenstellen tot een werkende applicatie.

De reeds ontworpen systeemdelen worden met behulp van test-driven development geïntregeerd in Search Traffic Monitor. Dit wordt op een professionele manier gedaan door gebruik te maken van SVN en een volledige OTAP ontwikkelstraat.

3.5 Uitvoeren van en rapporteren over het testproces

Opstellen testscript, waaronder het specificeren van de testgegevens, de benodigde startsituatie en de uitvoerverwachtingen per testgebied. Uitvoeren test plan en opstellen rapportage.

De nieuw ontworpen systeemdelen worden eerst omgezet naar unit tests om zo de kwaliteit van de code gedurende het hele ontwikkeltraject te monitoren.



H Tussentijdse beoordeling concept



Bespreking concept	Tussentijds assessment	Eerste beoordeling
---------------------------	-------------------------------	---------------------------

Formulier bespreking concept afstudeerdossier

Student: Ferdy Perdaan

Studentnummer: 08011249

Datum: 10 april 2012

Tijdens de bespreking is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	X	
b	Het afstudeerdossier is digitaal beschikbaar	X	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen		X
d	Het goedgekeurde afstudeerplan is aanwezig	X	
e	Het plan van aanpak is aanwezig	X	
f	Reeds geleverd commentaar is aanwezig	n.v.t.	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken		X
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	*	*

Verbeterpunten:

De huidige beschrijving van de werkzaamheden laat niet zien wat je allemaal gedaan hebt. Daarom is het voor mij nu niet mogelijk te bepalen of de afstudeeropdracht naar behoren wordt uitgevoerd. Neem tijd om het een en ander goed te beschrijven, ook al gaat dat nu even ten kosten van het product voor het bedrijf.

Opmerkingen:

Maak een afstudeerverslag van de werkzaamheden van de afgelopen 10 weken, zoals je dat ook zou doen bij het uiteindelijk in te leveren verslag. We hebben de afspraak gemaakt dat je dat op maandag 16 april naar mij zal opleveren en dat ik dat opgeleverde verslag digitaal van commentaar zal voorzien.

Naam begeleidend examiner: Arno Nederend

Datum: 10 april 2012



I Tussentijdse beoordeling assessment



Bespreking concept	Tussentijds assessment	Eerste beoordeling
--------------------	------------------------	--------------------

Formulier tussentijds assessment

Student: Ferdy Perdaan

Studentnummer: 08011249

Datum: 15 mei 2012

eerste / tweede TTA: eerste

Tijdens het tussentijds assessment is het volgende geconstateerd:		ja	nee
a	Het voortgangsverslag is ontvangen	x	
b	Het afstudeerdossier is digitaal beschikbaar	x	
c	Het afstudeerdossier is opgebouwd conform de richtlijnen	x	
d	Het goedgekeurde afstudeerplan is aanwezig	x	
e	Het plan van aanpak is aanwezig	x	
f	Reeds geleverd commentaar is aanwezig	x	
g	Het afstudeerdossier geeft voldoende inzicht in de stand van zaken	x	
h	De afstudeeropdracht is tot nu toe naar behoren uitgevoerd	.	.

Aanpak	O	T	V	G
Passend			x	
Theoretisch verantwoord			x	
Samenhang uitvoering beroepstaken			x	

Beroepstaken op afgesproken niveau uitgevoerd?		O	T	V	G
1	Selecteren van standaardsoftware			x	
2	Ontwerpen systeemdeel		x		
3	Bouwen applicatie			x	
4	Uitvoeren van en rapporteren over het testproces		x		



5					
6					
7					
8					

Producten	O	T	V	G
<i>Tussenproducten</i>		x		
<i>Eindproducten</i>		x		

Effectief communiceren	O	T	V	G
<i>Binnen afstudeerbedrijf</i>			x	
<i>Afstudeerdossier</i>		x		

Reflectie	O	T	V	G
<i>Inzicht in eigen functioneren</i>			x	
<i>Inzicht in eigen leerproces</i>		x		

Toelichting per beoordelingscriterium

Aanpak
Student heeft gegeven het project een aanpak beschreven. De verdeling in een fase inventarisatie van performance knelpunten en advisering gevolgd door een fase ontwikkeling lijkt adequaat. De beschrijving in het huidige afstudeerdossier is verwarrend en behoeft verbetering.

Beroepstaken op afgesproken niveau uitgevoerd?
Met name het ontwerp, middels de geleverde modellen en het testproces en het resultaat daarvan lijken voor verbetering vatbaar.

Producten



Het adviesrapport is niet afgemaakt, omdat de organisatie (mede op grond van het opgeleverde deel van het adviesrapport) de keuze heeft gemaakt voor nieuwbouw i.p.v. bijbouw. Het systeemontwerp is deels voldoende gedocumenteerd, maar lijkt nog onvoldoende beschreven om te komen tot een overdraagbare situatie binnen de organisatie. De rapportage over het testproces is nog niet van voldoende niveau.

Effectief communiceren

Student heeft examinatoren goed op de hoogte gehouden en lijkt binnen de organisatie voldoende draagvlak te creëren voor zijn oplossingen. De rol van de productowner in het sturingsproces van de uitvoering van het project is nog onvoldoende zichtbaar

Reflectie

De evaluatiehoofdstukken in het afstudeerdossier ontbreken nog. In het TTA gesprek lijkt de student de verbeterpunten te begrijpen en gaat daarmee aan de slag.

Advies

x	Positief (bindend advies)
	Verlengen (vrijblijvend advies)
	Negatief (vrijblijvend advies)

Besluit student

Aankruisen welke beslissing de student heeft genomen (alleen na vrijblijvend advies)

x	Afstudeerdossier wordt op afgesproken datum ingeleverd	Inleverdatum: 1 juni 2012
	Afstudeerperiode wordt verlengd	Inleverdatum:
	Student stopt met afstudeeropdracht	

Naam begeleidend examinerator: Arno Nederend

Naam tweede examinerator: Nanny Jacobs

Datum: 18 mei 2012



J Evaluatie bedrijfsmentor



Evaluatieformulier afstuderen

In te vullen door opdrachtgever c.q. bedrijfsmentor(en)

Student: 08011249

Periode: 2012-1.1, 06-02-2012 / 01-06-2012

Bedrijf c.q. instelling: Orange Valley Consultancy BV

Bedrijfsmentor: E. F. Blacquièrre

Plaats: Breukelen

Datum: 31-05-2012

1. Heeft de student zich zelf snel en goed ingewerkt in het bedrijf en de uit te voeren afstudeeropdracht?

Jazeker!

Het voordeel was natuurlijk de eerdere werkervaring bij OrangeValley. Maar Ferdy heeft zich snel en goed ingewerkt op de afstudeeropdracht, waarbij hij de dagelijkse gang van zaken los heeft gelaten.

2. Hoe beoordeelt u de communicatieve vaardigheden van de student (in de samenwerking met collega's, in contacten met de opdrachtgever, bij mondelinge presentaties, schriftelijke rapportages)?

De communicatieve vaardigheden van Ferdy zijn ruim voldoende. Hij heeft zijn plaats gevonden in het team en communiceert open en duidelijk met collega's. Hij staat achter zijn opvattingen en hij verwoordt en beargumenteert dit concreet.



3. Hoe heeft de student tijdens het uitvoeren van de opdracht gefunctioneerd?

• Qua verantwoordelijkheid	goed / voldoende / matig / onvoldoende
• Qua zelfstandigheid	goed / voldoende / matig / onvoldoende
• Qua planmatig werken	goed / voldoende / matig / onvoldoende
• Qua creativiteit	goed / voldoende / matig / onvoldoende
• Qua productiviteit	goed / voldoende / matig / onvoldoende
• Qua samenwerken met collega's	goed / voldoende / matig / onvoldoende
• Qua draagvlakontwikkeling	goed / voldoende / matig / onvoldoende
• Qua inspelen op bedrijfscultuur	goed / voldoende / matig / onvoldoende
• Qua rekening houden met de specifieke context van het bedrijf	goed / voldoende / matig / onvoldoende
• Qua het op gang brengen van de nodige veranderingen	goed / voldoende / matig / onvoldoende

4. Hoe beoordeelt u de kennis en kunde van de student in verhouding tot wat u verwacht van een bijna afgestudeerde?

De kennis en kunde van Ferdy is in lijn met wat OrangeValley van een bijna afgestudeerde verwacht.

5. Hoe beoordeelt u de kwaliteit van de opgeleverde (tussen)producten?

De kwaliteit van de opgeleverde producten en tussenproducten is goed. De producten voldeden in veel gevallen aan de gestelde eisen. In andere gevallen was er meestal slechts één aanpassing nodig om het product aan de eisen te schikken.



6. Bent u tevreden over het opgeleverde (eind)product?

Ja.

- **In hoeverre heeft u gekregen wat is afgesproken?**

We hebben gekregen wat er afgesproken is en zelfs meer dan dat. Naast concrete adviezen om de performance te verbeteren (doelstelling) is onder andere een deel van de adviezen al doorgevoerd.

- **In hoeverre voldoet het (eind)product aan uw verwachtingen?**

Het eindproduct voldoet aan de verwachtingen.

- **Wat is de bruikbaarheid en onderhoudbaarheid hiervan?**

De bruikbaarheid is ruim voldoende. Dit bleek al tijdens het afstudeertraject. De onderhoudbaarheid is nog een punt van aandacht, omdat veel kennis nog gecentraliseerd is bij Ferdy.

- **Wat gebeurt er met het opgeleverde (eind)product?**

Het advies van Ferdy wordt – voor zover nog niet gedaan – uitgevoerd. Al tijdens het afstudeertraject bewees het product (Search Traffic Monitor) al meer toegevoegde waarde te bieden in de werkzaamheden van OrangeValley dan daarvoor.

- **Kunt u direct met het opgeleverde product aan de slag?**

Ja.

7. Zijn er nog aspecten voor u van belang die nog niet aan de orde zijn geweest?

Ik wil graag nog een compliment uitspreken over de inzet en het enthousiasme van Ferdy. Hij heeft zich gecommitteerd aan het product Search Traffic Monitor, waarmee het duidelijk “zijn” project werd.

8. Bent u bereid een volgende keer weer uw medewerking te verlenen aan het beschikbaar stellen van een afstudeerplaats (graag met toelichting)?

Ja. Wanneer afstudeerders een vergelijkbare werkhouding, kennisniveau en inzet als Ferdy hebben, stellen we graag een afstudeerplaats beschikbaar.