



# DEVELOPING A SERVER ADMINISTRATION TOOL

---

Afstudeerscriptie Colin Duivis



## Titelpagina

### Auteur

Geschreven door: Duivis C.  
Studentnummer: 2192461  
Afstudeerrichting: ICT Software Engineering (Voltijd)  
Afstudeer Periode: februari 2014 t/m juni 2014

### Bedrijf

Naam: NetMatch B.V.  
Afdeling: Development  
Plaats: Tilburg  
Begeleider: Havermans J. (Project Manager)

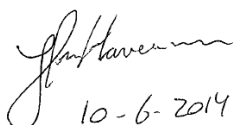
### School

School: Fontys Hogenscholen  
Assessor 1: Mols H.P.F.A.M.  
Assessor 2: Krielen M.J.A.

### Document

Versie: 1.0  
Status: Oplevering

Getekend voor gezien door Projectbegeleider:

  
10-6-2014

## Document Historie

Datum	Versie	Auteur	Omschrijving
<b>2014-02-10</b>	0.1	Colin Duivis	Opzet van het document
<b>2014-04-03</b>	0.2	Colin Duivis	Opnemen onderzoek in scriptie
<b>2014-05-14</b>	0.3	Colin Duivis	Opnemen ontwerpfase + begin implementatiefase
<b>2014-05-14</b>	0.4	Colin Duivis	Correcties van Jeroen Havermans en Erik Mols verwerken.
<b>2014-06-03</b>	0.5	Colin Duivis	Opnemen eind implementatiefase
<b>2014-06-07</b>	0.6	Colin Duivis	Laatste Correcties van Jeroen Havermans verwerken
<b>2014-06-09</b>	0.7	Colin Duivis	Correcties van Familie / Vrienden opnemen.
<b>2014-06-09</b>	1.0	Colin Duivis	Laatste stijl aanpassingen

## Goedkeuring Versie 1.0

*Projectbegeleider*

Naam:

Datum:

Paraaf:

*Docentbegeleider*

Naam:

Datum:

Paraaf:

## Voorwoord

De afgelopen maanden heb ik stage gelopen bij NetMatch te Tilburg, hier ben ik aan de slag gegaan met het ontwikkelen van een server administration tool.

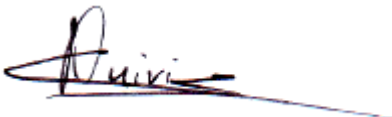
In deze afstudeerstage heb ik de basis van het systeem ontwikkeld waarmee systeembeheer en accounting van NetMatch alle servers, licenties en maandelijkse kosten bij kunnen houden en in kunnen zien.

Ik zou graag de volgende mensen willen bedanken:

- Jeroen Havermans: voor het begeleiden van de hele stage.
- Systeembeheer: voor het altijd vragen mogen stellen over hun systemen, het vertrouwen aan mij geven om in hun systemen te kijken en dat ze de backup terug hebben gezet van me database toen hij per ongeluk helemaal verwijderd was.
- Leo Hexspoor: voor het inzicht geven in de werkwijzen en standaarden binnen projecten van NetMatch en meehelpen met een paar moeilijke vraagstukken binnen project.
- Chris Campfens: voor het altijd mee willen helpen met vragen die ik had en voor het uitleggen van nieuwe technologieën en meedenken met welke .NET technologieën die binnen mijn project handig zouden kunnen zijn.
- Ron Otten: voor het tijd vrij maken om mij nieuwe front-end technologieën te leren en hier ook altijd even tijd kon vrij maken om mee te kijken bij mijn project als ik even een fout tegen kwam.

Verder wens ik u als lezer veel plezier bij het lezen van mij scriptie.

Colin Duivis



06-06-2014 Tilburg

## Samenvatting

NetMatch is een bedrijf dat e-commerce software levert aan tour operators, veel van deze software pakketten moeten op serversystemen (server) draaien. Deze servers zijn in beheer van NetMatch. Er zijn veel verschillende servers en al deze servers hebben hun eigen licenties die gebruikt moeten worden. En ook aangeschaft moeten worden.

Echter was er nog geen systeem aanwezig die alle licenties van de servers goed vastlegt. Zonder een goed systeem was er geen inzicht in welke maanden, welke software gebruikt werd. Zonder dat inzicht kon er ook niet eenvoudig herleid worden welke nieuwe licenties er weer aangeschaft moeten worden.

Om er achter te komen wat de beste methode voor NetMatch is om het bovengenoemde systeem (Server Administration Tool) te ontwikkelen, is er onderzoek gedaan naar wat de requirements van NetMatch zijn en is er gekeken of er al applicaties waren die aan deze requirements voldeden.

Uit het onderzoek bleek dat er geen applicatie was die de requirements van NetMatch ondersteunde en is er gekozen om zelf een applicatie te ontwikkelen.

Voordat de applicatie ontwikkeld kon worden moest er eerst vormgegeven worden aan de te ontwikkelen applicatie. Hiervoor heb ik eerst een database ontwerp gemaakt aan de hand van de requirements en een grafische ontwerp om de globale look and feel vast te leggen van de applicatie.

Na goedkeuring van de ontwerpen ben ik begonnen aan het opzetten van de applicatie. Voor het ontwikkelen van de applicatie zijn er eerst de ontwikkelomgeving, testomgeving en productie omgeving klaar gezet. Hierna heb ik de database opgezet en ben ik begonnen met het opzetten van de basis van het programma.

De basis van het programma bevat de structuur van het programma opzetten, koppeling maken met de database, koppeling maken met Active Directory (AD), de globale look and feel implementeren en het inloggen van de gebruikers.

Uiteindelijk kan er begonnen worden met het iteratief opleveren van de functionaliteiten. Dit gebeurt in overeenstemming met de eindgebruikers, welke besluiten wat de volgende stap binnen de applicatie wordt. De volgende schermen zijn opgeleverd: Bulk upload van licentie lijsten, overzicht van licenties, toevoegen van klanten, overzicht van klanten, toevoegen van servers en overzicht van servers.

De applicatie is op dit moment nog niet zover dat hij in productie genomen kan worden, maar het heeft wel een sterke basis waarin het merendeel van de logica al aanwezig is om de applicatie goed verder op te bouwen.

De aanbeveling is om het systeem verder te ontwikkelen, zoals deze nu ontwikkeld is ( dus iteratief samen met de eindgebruikers bepalen en kijken welke richting de applicatie op moet gaan ).

Verder zouden de volgende functionaliteiten als eerst ingebouwd moeten worden (in overleg met de eindgebruikers natuurlijk): koppeling servers klanten en licenties, licentie filteren op het gebruik van licenties, licentie filteren op maanden waarin ze gebruikt worden, licenties filteren op klant, server filteren op klant.

Met deze functionaliteiten kunnen licenties opgezocht worden bij klant en kan er terug in de geschiedenis gekeken worden wanneer licenties gebruikt zijn en wat de prijzen waren in die maanden.

## Abstract

NetMatch is a company who delivers software to tour operators, much of this software needs to be run on a server systems (server). NetMatch is in control of these servers. There are many different servers and each of these servers has its own licenses. These licenses need to be purchased.

The only problem is that there isn't any system which registers all licenses of the servers in a good manner. Without a good system, there is no way to see what licenses are you used on what servers in which month. Without this system there is no easy way to view which new licenses should be purchased for the next month.

To find out what the best method for NetMatch is, to develop the missing system (Server Administration Tool). To find this method, there has to be a research to find out what the requirements are and if there is any existing piece of software which will fulfill these requirements.

The research showed there are no existing applications which would fulfill the requirements, so we chose to develop our own application.

Before the Server Administration Tool can be developed, the application has to be designed. To do this I started with developing a database design and after that a graphical design for the basic look and feel of the application.

After approval of the designs I started with setting up the application. Before I developed the application at developer environment, test environment and a production environment were set up. After this I setup the database and started developing the basis of the application.

The basis of the application consists of: setting up the structure, making connection with the database, making connection with Active Directory (AD), implementing the global look and feel and logging in of the users.

Finally I could start with iteratively delivering functionalities. The functionalities to deliver were always done in accordance with the end users. The following screens were delivered: bulk upload of license lists, a summary of licenses, adding clients, a summary of clients, adding servers, a summary of servers.

The application at this moment, isn't ready for production yet, but it has a strong basis in which a great part of the applications logic already resides to easily continue developing. '

The recommendation is to keep developing the application, as it is currently being developed (iteratively in accordance with the end users).

Furthermore I recommend that the following screens should be built in first: connecting servers, clients and licenses, filtering of licenses on their usage, filtering licenses on months were they are used in, filtering licenses on clients and filtering servers on clients.

When these functionalities are added, all licenses can be filtered by their owning clients and the history of these clients and their monthly costs can be looked up.

## Inhoudsopgave

Verklarende Woordenlijst .....	8
1. Inleiding .....	10
2. Het Bedrijf.....	11
2.1. Mijn Positie binnen het Bedrijf.....	11
3. De Opdracht .....	12
3.1. Aanleiding.....	12
3.2. Situatie voor aanvang van het Project .....	12
3.3. Projectdoelstellingen.....	12
3.4. Project Aanpak .....	13
3.5. Project Scope.....	13
4. Het Onderzoek.....	14
4.1. Doel .....	14
4.2. Hoofdvraag .....	14
4.3. Deelvragen.....	14
4.4. Conclusie .....	14
4.5. Aanbeveling.....	14
5. De Onderzoeksfase.....	15
6. De Ontwerpfase.....	16
6.1. Database Ontwerp .....	16
6.2. Grafisch Ontwerp .....	17
6.3. Structuur.....	17
6.4. Technieken .....	18
6.5 Gebruikers Interactie.....	19
7. De Implementatiefase .....	22
7.1. Opzet Project.....	22
7.2. Omgevingen .....	23
7.3. Versiebeheer .....	23
7.4. Applicatie Ontwikkelen .....	24
8. Conclusie & Aanbevelingen .....	31
8.1. Conclusie .....	31
8.2. Aanbevelingen.....	31
Evaluatie / Reflectie.....	32
PID .....	32
Onderzoek .....	32
Opzetten Applicatie.....	32

Ontwikkelen Applicatie .....	32
Globaal.....	33
Literatuur Lijst .....	34
<b>Bijlage I: PID</b>	
<b>Bijlage II: Onderzoeksverslag</b>	
<b>Bijlage III: Requirements</b>	
<b>Bijlage IV: Requirement Gesprekken</b>	



## Verklarende Woordenlijst

<b>Active Directory (AD)</b>	Active Directory is een verzamelplaats voor alle rechten gegevens van groepen en gebruikers binnen het netwerk van een organisatie.
<b>ASP.NET</b>	ASP.NET is een code bibliotheek van Microsoft die veel standaard implementaties rondom het gebied van websites bevat.
<b>Branching</b>	Branching is een term die gebruikt wordt binnen versiebeheer. Deze term houdt in dat je een aftakking maakt vanaf een bepaalde versie van je software en vanaf daar weer verder kan ontwikkelen op een aparte tak die los komt te staan van de huidige ontwikkelingen.
<b>C#</b>	Een object georiënteerde programmeertaal van Microsoft.
<b>Client-side</b>	Client-side is een term die aangeeft dat een bepaalde actie aan de kant van de gebruiker van de applicatie plaats vindt.
<b>Cookie</b>	Cookie is een bestandje dat op geslagen wordt op de pc van de bezoeker van een website om bepaalde gegevens in bij te houden.
<b>Entiteit Relatie Diagram (ERD)</b>	ERD is een diagram die de verschillende de relaties van object met elkaar laat zien. Deze diagrammen worden vaak gebruikt voor databases.
<b>Entity Framework</b>	Entity Framework is een database koppeling die gebruikt wordt binnen .NET om connectie te maken met SQL databases.
<b>Entity Framework: Model First</b>	Deze aanpak van Entity Framework zet de aanwezige model klassen om naar een tabellen in een database.
<b>Entity Framework: Database First</b>	Deze aanpak van Entity Framework genereert model klassen vanuit de database.
<b>Front-end</b>	Front-end is het gedeelte van de software dat de interface aanbiedt aan de gebruiker van de software.
<b>HTTPS</b>	HTTPS is de beveiligde versie van HTTP die ervoor zorgt dat de communicatie geencrypt is.
<b>IIS</b>	IIS is een server van Microsoft waarop voornamelijk Microsoft websites gehost kunnen worden.
<b>Integrated development environment (IDE)</b>	IDE is een afkorting die staat voor een applicatie waarin software projecten gemaakt kunnen worden.
<b>Library</b>	Een collectie van code die gebruikt kan worden binnen een software applicatie.
<b>MVC 4</b>	MVC 4 is een project structuur van Microsoft die geoptimaliseerd is voor het ontwikkelen van websites. (MVC staat voor Model View Controller)
<b>Razor</b>	Razor is een script taal die helpt eenvoudig ASP.NET code aan te roepen in html pagina's.
<b>Requirements</b>	Requirements is een document waarin de eisen van de applicatie zijn opgenomen.
<b>Server-side</b>	Server-side is een term die aangeeft dat een bepaalde actie aan de kant van de server van de applicatie plaats vindt.
<b>Service Level Agreements (SLA)</b>	SLA is een overeenkomst die een werk contract tussen twee partijen vastlegt.
<b>Services Provider License Agreement (SPLA)</b>	Een licentie model van Microsoft die gebruikt wordt voor het maandelijks aanschaffen van licenties, dit gebeurt via een gecertificeerde reseller van Microsoft.
<b>Sessie Data / Session Data</b>	Session Data is een naam voor data die wordt bij gehouden voor de bezoeker van de website.

<b>TFS</b>	Team Foundation Server is de methode van Microsoft voor versiebeheer.
<b>Use Case Diagrammen (UCD)</b>	Use Case Diagrammen zijn diagrammen die bedoelt zijn om inzicht te geven in de functionaliteiten van het programma.
<b>Versiebeheer</b>	Versiebeheer is een methodiek van je software opslaan waarbij oude versies van je bestanden worden bij gehouden. Met als doel dat als er iets mis gaat je altijd nog terug kan gaan naar een eerdere versie waar het nog goed was.
<b>Visual Studio 2012 (VS 2012)</b>	VS 2012 is een ontwikkelomgeving van Microsoft die je instaat stelt .NET applicaties mee te ontwikkelen.

## 1. Inleiding

NetMatch is een bedrijf dat e-commerce software levert aan tour operators, veel van deze software pakketten moeten op serversystemen (server) draaien. Deze servers zijn in beheer van NetMatch zelf. Er zijn veel verschillende servers en al deze servers hebben hun eigen licenties die gebruikt moeten worden. En ook aangeschaft moeten worden.

Echter was er nog geen systeem aanwezig die alle licenties van de servers goed vastlegt. Zonder een goed systeem was er geen inzicht in welke maanden, welke software gebruikt werd. Zonder dat inzicht kon er ook niet eenvoudig herleid worden welke nieuwe licenties er weer aangeschaft moesten worden.

Dit is dan ook het project wat ontwikkeld moet worden. Een systeem dat alle servers kan bijhouden welke licenties die servers hebben en bij welke klanten van NetMatch de servers horen.

Deze scriptie legt het gehele traject vast dat doorlopen is om dit project op te zetten.

Hieronder staan de verdere hoofdstukken van dit verslag beschreven:

- **Het Bedrijf:** In dit hoofdstuk leer je NetMatch als een bedrijf beter kennen en er wordt uitgelegd wat mijn positie binnen NetMatch is.
- **De Opdracht:** Hier wordt de opdracht goed beschreven, dit hoofdstuk is een uitgebreide samenvatting van de PID.
- **Het Onderzoek:** In dit hoofdstuk wordt het onderzoek en de resultaten beschreven van mijn project.
- **De Onderzoeksfase:** In dit hoofdstuk wordt het proces van het onderzoek beschreven, hoe deze verlopen is en waar de knelpunten zaten.
- **De Ontwerpfase:** Hierin staat het proces van het ontwerpen van de applicatie, hierin staat het grafische ontwerp, database ontwerp, en de basiswerking van de applicatie in Use Case Diagrammen (UCD).
- **De Ontwikkelfase:** Hierin is het proces van het opzetten en ontwikkelen van de applicatie beschreven.
- **Conclusie & Aanbevelingen:** Hierin wordt een conclusie getrokken over het gehele project en aanbevelingen gedaan wat er verder nog moet gebeuren binnen het project.

## 2. Het Bedrijf

NetMatch is een Nederlands software bedrijf die opgericht is in 1998. NetMatch specialiseert zich in het ontwikkelen van e-commerce software voor tour operators. Hierbij wordt gebruik gemaakt van de meest recente Microsoft technologieën, zoals: Visual Studio 2013, .NET 4.5, ASP.NET MVC4 en SQL Server 2012.

De klanten van NetMatch variëren van traditionele aanbieders van pakketreizen, cruises, self-drive reizen, avonturenreizen, stedentrips en korte vakanties tot verhuurbedrijven van vakantiewoningen en portaalsites voor reizen.

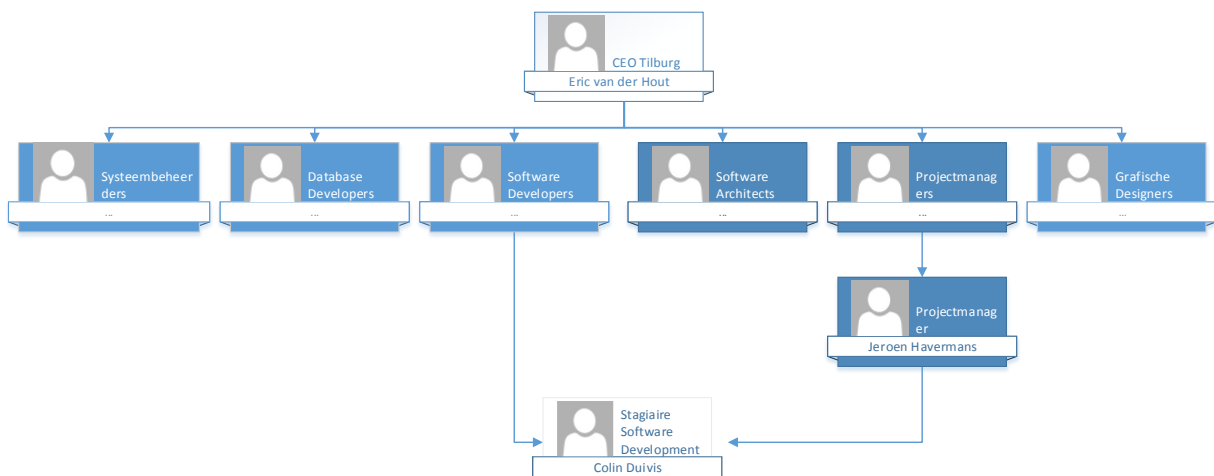
NetMatch is gevestigd in Amsterdam, Tilburg en in Cluj-Napoca (Roemenië). Verspreid over deze locaties heeft NetMatch ongeveer 70 werknemers.

NetMatch heeft 2 producten/services waar ze zich groot in maken:

- **TourWeb:** TourWeb is een totaal modulair systeem die gericht is op de markt van tour operators en online reisbureaus. Dit product geeft de mogelijkheid je huidige reissystemen uit te breiden, maar TourWeb kan ook zo ingezet worden dat je een vanuit hier je eigen reissysteem maakt. TourWeb is een van de systemen die NetMatch goed onderhoud en blijft door ontwikkelen.
- **Hosting:** Door de vele jaren ervaring die NetMatch heeft opgedaan in hosting van eigen servers en servers voor klanten, zijn ze expert geworden in het onderhouden en inrichten van servers en hierop allerlei services te hosten voor klanten. Deze ervaring gebruikt NetMatch om maatwerk hosting oplossingen voor bedrijven te bieden.

### 2.1. Mijn Positie binnen het Bedrijf

Ik werk in Tilburg op de Software Development afdeling. Het bedrijf heeft een platte bedrijfsstructuur in feite valt iedereen direct onder de CEO. Echter als stagiaire wordt ik beoordeeld en aangestuurd door Jeroen Havermans (Project Manager).



Figuur 1 Organigram NetMatch Tilburg met extra uitlichting van mijn positie binnen het bedrijf.

### 3. De Opdracht

Dit hoofdstuk is gebaseerd op het PID verslag (bijlage I).

#### 3.1. Aanleiding

NetMatch levert hosting services op Windows & Linux platformen aan klanten. NetMatch heeft meerdere servers, verspreid over Nederlandse datacenters: webserver, SQL-server, applicatie server, load balancers, caching servers, etc.

Bij het beheren van het platform van NetMatch komen ook veel randzaken kijken, denk hierbij aan het administreren en rapporteren van software licenties, bijhouden van lopende support contracten, een change log bijhouden, garanties bijhouden en het bijhouden van Service Level Agreements (SLA).

Deze administratie was aan het begin van de afstudeerstage verdeeld over meerdere documenten en applicaties. Sommige specifieke data zat vaak alleen in de hoofden van de systeembeheerders. Het doel van het project is deze informatie globaal toegankelijk en inzichtelijk te maken voor alle personen die deze informatie nodig hebben.

#### 3.2. Situatie voor aanvang van het Project

De server administratie en de licenties behorend bij de servers waren verdeeld over 2 plekken:

1. De administratie van de klanten, kosten en de licenties zijn in 1 groot Excel bestand opgenomen. Waarbij handmatig maandelijks de prijzen up-to-date gehouden moeten worden.
2. De administratie van de locatie van de servers en de indeling van de serverkasten zit in een programma opgenomen die RackTables ([Andriyanov, Dummer, & Ovsienko, 2013](#)) heet.

Maandelijks wordt er vanuit SEQUINT een prijslijst doorgegeven van nieuwe licenties en vanuit de nieuwe licentie prijzen op jaarbasis komen er updates in de prijzen, er vervallen oude licenties en komen nieuwe bij. Eén keer in de drie jaar wordt het contract met SEQUINT vernieuwd. De gegevens uit de maandelijks prijslijst moesten vervolgens doorgevoerd worden naar de eerder genoemde Excel sheet om zo voor alle klanten de prijzen weer up-to-date te hebben.

Deze manier van documenteren werd op dat moment erg omslachtig bevonden en buiten dat is deze manier ook zeer foutgevoelig. Zo zijn er eind 2013 drie maanden lang de verkeerde aantallen gerapporteerd, resulterend in een foutieve inkoopfactuur. Ook is er bij een audit van Microsoft al ondervonden dat het ophalen van alle gegevens over de servers waarop Microsoft licenties draaien een hele opgave is. Bij dit ophalen kwamen er ook weer enkele fouten binnen de administratie naar boven, dit keer in NetMatch' voordeel met een besparing tot gevolg. Hieruit blijkt nogmaals dat het voeren van een goede administratie zeer belangrijk is.

#### 3.3. Projectdoelstellingen

##### 3.3.1. Doel

Het doel van het project is om de server administratie samen te voegen in één applicatie. Deze applicatie moet centraal benaderbaar zijn, zodat eindgebruikers (zie hoofdstuk 3.5.) toegang hebben tot dezelfde informatie. Verder moet de applicatie ervoor zorgen dat de marge voor menselijke fouten geminimaliseerd wordt en dat de gebruiksvriendelijkheid toe neemt ten opzichte van de huidige situatie. Het streven is dat er binnen een jaar minder fouten zijn in de administratie ten opzichte van de huidige situatie.

### 3.3.2. Gewenste Situatie

Eén applicatie waarin alle servers met de huidige gebruikte data bijgehouden worden, zodat de informatie snel gevonden kan worden en het up-to-date houden van de informatie weinig moeite kost.

### 3.4. Project Aanpak

Voor de begin fase van het project is er gekozen voor een stapsgewijze projectstructuur waar met interviews met de projectbegeleider, de eindgebruikers en observatie van het huidige systeem vastgelegd kan worden wat de huidige eisen zijn van het systeem.

Op het moment dat de requirements van het project vast staan, word het onderzoek gestart.

In het geval van een eigen te bouwen applicatie wordt het volgende o.a. gemaakt:

- Database Structuur, Entity Relationship Diagram (ERD)
- Software Architectuur (Klassen Diagrammen)
- Grafische Designs

Als de uiteindelijke software geïmplementeerd gaat worden, wordt er een iteratieve methodiek toegepast om zo te kunnen toetsen of het project de richting op blijft gaan die de projectbegeleider en de eindgebruikers willen. Op deze manier is er nauwe samenwerking tussen de partijen en blijft iedereen betrokken bij het project waardoor het eindresultaat ook echt iets is waar de gebruikers en de projectbegeleider profijt bij hebben.

### 3.5. Project Scope

De scope van dit project blijft binnen NetMatch en betreft de volgende afdelingen en personen:

- NetMatch Systeembeheer (eindgebruiker)
- NetMatch Project Manager: Jeroen Havermans (projectbegeleider)
- NetMatch Boekhouding: Ronald Wisse (eindgebruiker)
- NetMatch Development: Colin Duivis (ontwikkelaar)

## 4. Het Onderzoek

Hieronder staan in het kort de belangrijkste hoofdstukken van het onderzoeksverslag (bijlage II) uitgelicht.

### 4.1. Doel

De doelstelling van dit onderzoek is het realiseren van een methode die voldoet aan de eisen van NetMatch voor een “Server Administration Tool”.

### 4.2. Hoofdvraag

Welke oplossing voldoet het best aan de eisen en criteria van NetMatch voor één “Server Administration Tool”?

### 4.3. Deelvragen

1. Welke processen en tools worden momenteel gebruikt en wat is de gewenste situatie?
2. Wat zijn de eisen van NetMatch aan de “Server Administration Tool”?
3. Wat zijn voor NetMatch de criteria waarmee de verschillende oplossingen vergeleken moeten worden?
  - a. Hoe wordt een tijdsriteria verrekend over de verschillende oplossingen?
4. Welke oplossingen zijn er die aan de eisen en criteria voldoen?
  - a. Welke bestaande oplossingen zijn er?
  - b. Op welke manier kunnen de huidige tools ingezet worden?

### 4.4. Conclusie

Na het filteren op welke applicaties er onder het budget blijven van € 5000,- per jaar, blijven alleen SpiceWorks en Racktables over. Van die applicaties heeft SpiceWorks het meeste aantal taken. (Taken zijn componenten die in het systeem gebouwd moeten worden, afgeleiden van de opgezette requirements).

Het probleem is echter dat de beste applicatie (SpiceWorks) die naar voren komt geen opensource applicatie is en maar 40,4% van alle taken en 59,5% van alle Must Have taken bevat. Om deze applicatie zover te krijgen dat het de uiteindelijke “Server Administration Tool” is die NetMatch wilt zal er heel wat aangepast moeten worden en dan is de vraag hoeveel aan te passen is met het modulaire plug-in systeem van SpiceWorks.

De enige opensource applicatie (RackTables) bevat slechts 25% van het totaal aantal taken en maar 28,6% van de Must Have taken. Deze applicatie zal dus aanzienlijk aangepast moeten worden om aan alle eisen voor de “Server Administration Tool” te voldoen.

De conclusie is dan ook dat geen van gevonden en geanalyseerde applicaties voldoet aan de eisen en criteria om deze ook daadwerkelijk te gebruiken als basis voor de te maken: “Server Administration Tool”.

### 4.5. Aanbeveling

Er wordt aanbevolen dat geen van de gevonden applicaties gebruikt worden als basis van de “Server Administration Tool”. Dit aangezien er niet voldoende voldaan wordt aan de taken en door de twijfel over de aanpasbaarheid van de applicaties.

Er wordt dan ook aanbevolen dat er een eigen applicatie ontwikkeld wordt.

## 5. De Onderzoeksfase

Ik ben het onderzoek begonnen met het vaststellen wat de huidige werkwijze is en wat de gewenste situatie is voor de verschillende partijen (Accounting, Systeembeheer en Management). Dit deed ik zowel om er achter te komen welke applicaties en werkwijzen er gehanteerd worden, als om te weten te komen wat NetMatch graag in hun nieuwe applicatie wilt zien. De resultaten van deze gesprekken heb ik gebruikt om de requirements op te stellen en als basis voor de latere deelvragen.

Vervolgens heb ik deze antwoorden omgezet in eisen waaraan de applicatie moet voldoen. Deze zijn opgenomen in het requirements document (bijlage III). Deze requirements heb ik ter goedkeuring naar alle partijen gestuurd. Buiten dat deze requirements nodig zijn voor het maken van het project, zijn ze ook nodig voor mijn onderzoek. Voornamelijk voor de laatste deelvraag waar de oplossingen vergeleken worden op basis van eisen en criteria.

Terwijl ik nog aan het wachten was op de goedkeuring van de requirements, ben ik de gesprekken gestart om achter de criteria voor de applicatie te komen. Ook deze criteria waren nodig voor de vergelijkingen in de laatste deelvraag.

Bij het zoeken naar een goede oplossing voor het project ben ik gaan kijken welke applicaties er gebruikt werden door NetMatch en welke applicaties er al op internet bestonden. Alle applicaties die de eerste selecties door kwamen zijn vervolgens tegenover de criteria en de eisen van NetMatch gehouden om te kijken of er een applicatie bestaat die al aan een grote meerderheid van alle eisen en criteria voldoet.

Voor een gedetailleerdere uitleg over hoe alle deelvragen onderzocht zijn, zie het hoofdstuk Onderzoeksmethode in bijlage II.

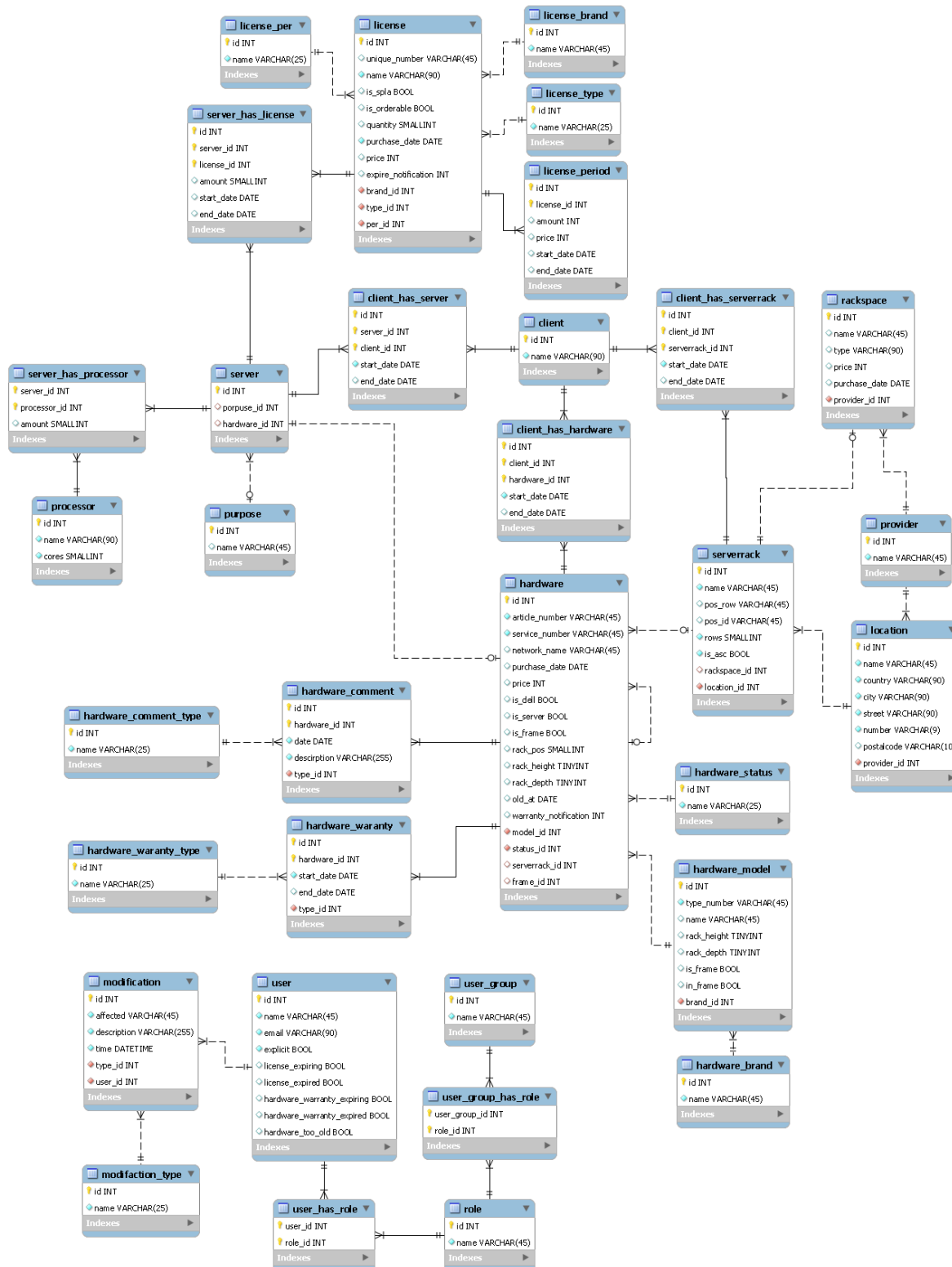
Voor het afronden van het onderzoek heb ik het verslag opgestuurd naar mijn projectbegeleider Jeroen Havermans. Op het moment dat deze goedgekeurd werd heb ik hem ook doorgestuurd naar mijn schoolbegeleider Erik Mols.



## 6. De Ontwerpfase

### 6.1. Database Ontwerp

In de ontwerpfase is er eerst een database ontwerp (figuur 2) gemaakt. Deze database is gebaseerd op de informatie verkregen uit de interviews van het onderzoek (bijlage IV) en de requirements die daaruit zijn opgesteld (bijlage III).



Figuur 2 Database Ontwerp

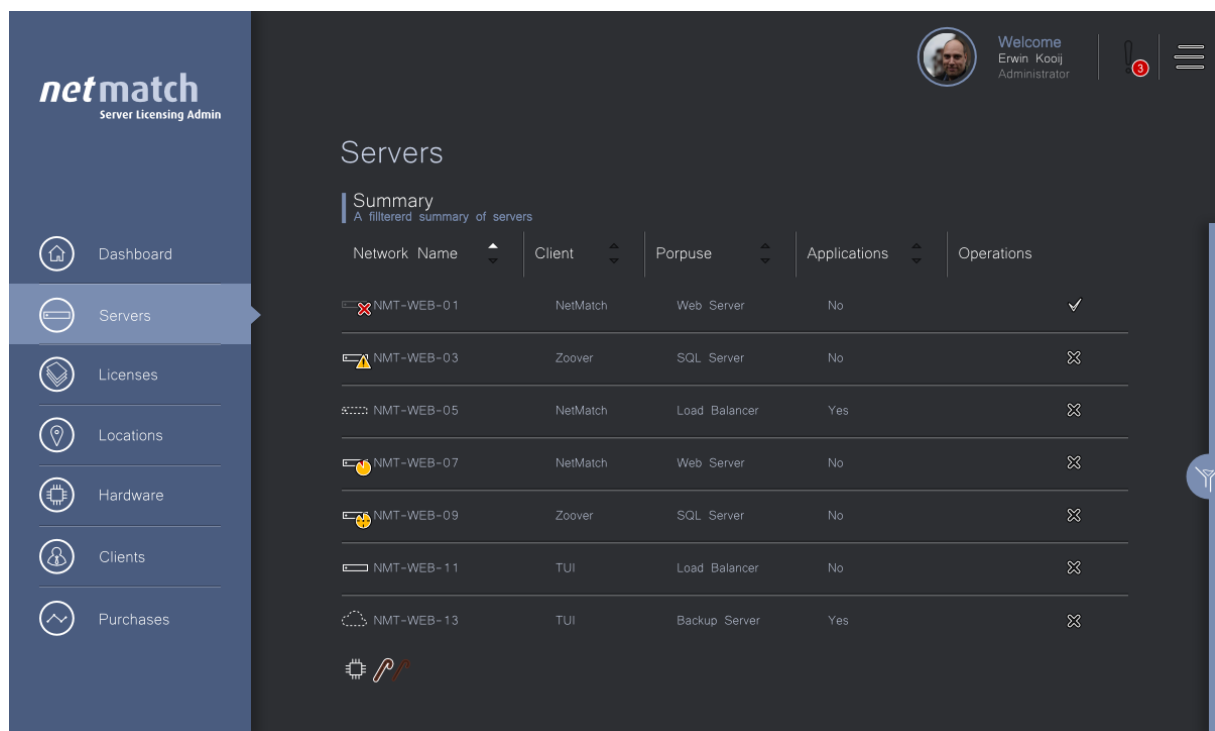
## 6.2. Grafisch Ontwerp

Na het database ontwerp moest er een globale “look and feel” van het programma opgezet worden die als template gebruikt kon worden voor de uiteindelijke applicatie.

Voordat er begonnen werd aan het ontwerpen van de applicatie is er eerst een lijst opgesteld van alle schermen die ongeveer gemaakt moeten worden.

Server Summary	Server Detail	Server Add/Edit
License Summary	License Detail	License Add/Edit
Client Summary	Client Detail	Client Add/Edit
Hardware Summary	Hardware Detail	Hardware Add/Edit
DataCenter Summary	DataCenter Detail/ Serrack Summary	DataCenter Add/Edit
Serrack Detail	Serrack Add/Edit	Settings
User Control	Purchase Summary	Purchase Month Compare
Changelog Summary		

Aangezien de applicatie gefaseerd ontwikkeld wordt, is er maar 1 ontwerp gemaakt die de basis en vaak terugkomende onderdelen visueel weergeeft (figuur 3).



Figuur 3 Grafisch Ontwerp

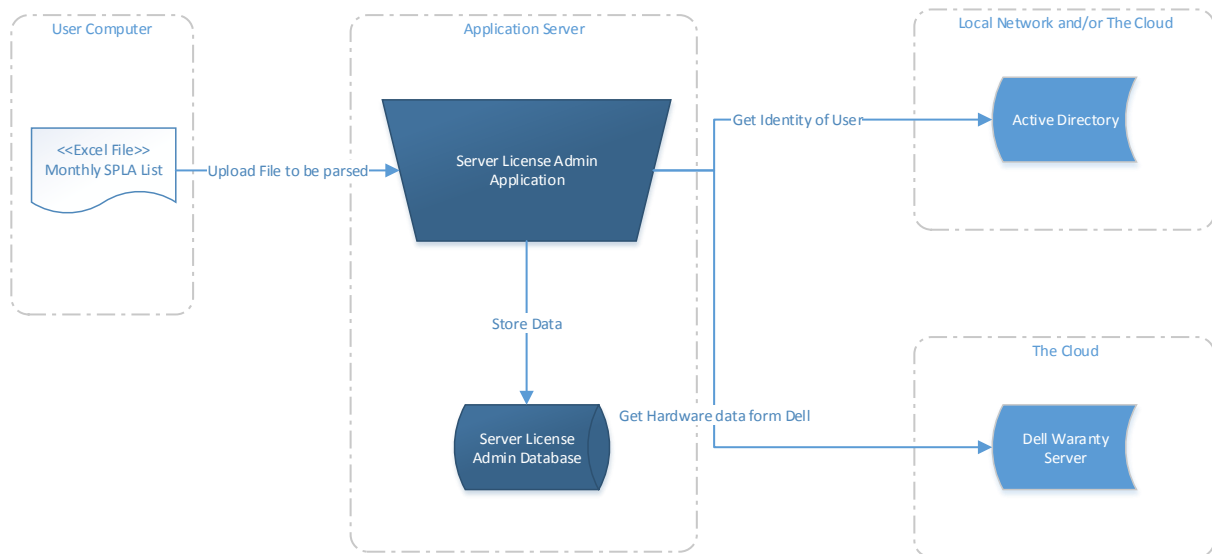
Het ontwerp is voorgelegd aan de eindgebruikers, die positief reageerde en er geen commentaar op hadden.

## 6.3. Structuur

De basis van de applicatie wordt opgebouwd vanuit het ontworpen database model. Aangezien de applicatie stukje bij beetje verder ontwikkeld gaat worden zijn er op het gebied van de structuur van de applicatie alleen maar high level designs gemaakt.

### 6.3.1. Communicatie

In figuur 4 is weergegeven waar de applicatie zijn gegevens ophaalt en opslaat.



Figuur 4 Data-access Diagram

### 6.3.2. Applicatie Structuur

De applicatie wordt verdeeld in 3 delen:

- **Web:** Hier zullen alle views opgebouwd worden en deze bevat de configuratie voor de website.
- **Bussiness:** Hierin worden alle controllers gedefinieerd waarmee communicatie mogelijk is. Deze laag communiceert tussen Web en Core.
- **Core:** Hierin zit alle logica van het programma, zoals het communiceren met de database, verwerken van de Excel bestanden, connectie maken met de Dell Warranty Server en connectie maken met de Active Directories.

### 6.4. Technieken

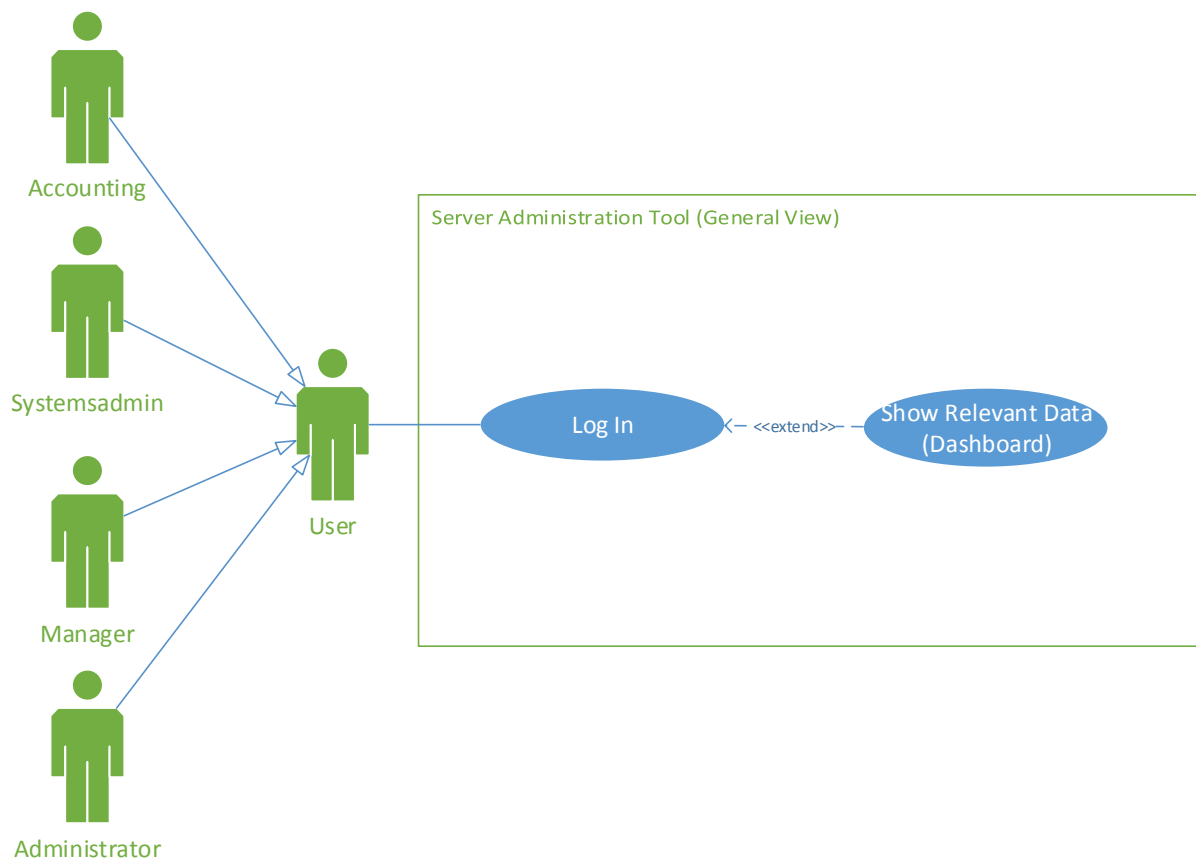
Aangezien de applicatie voor NetMatch intern gebruikt en onderhouden gaat worden. Is er gekozen voor de technieken die vaak binnen NetMatch gebruikt worden (NetMatch, 2013). Hierdoor is de drempel om in het project te stappen voor een ontwikkelaar van NetMatch lager. Het bijkomende voordeel is dat er nu meerdere ontwikkelaars zijn waarop terug gevallen kan worden als een moeilijk probleem zich voordoet. Het enige nadeel is dat de leercurve voor het starten van het project hoger ligt.

De onderstaande technieken zijn vastgesteld door te praten met verschillende ontwikkelaars. De backend technieken zijn besproken met Leo Hexspoor en Chris Campfens (Campfens, 2014) en de front-end technieken zijn besproken met Ron Otten (Otten, Front-end Technologiën, 2014).

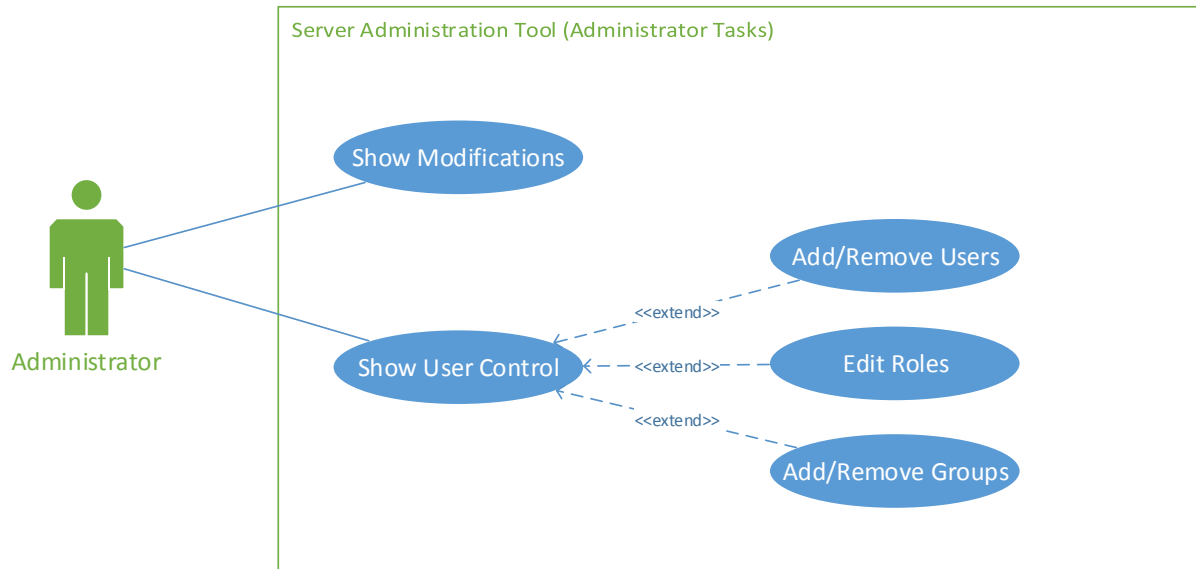
<b>Database</b>	MS SQL
<b>Programmeer Taal</b>	C#
<b>Framework</b>	ASP.NET 4.5 – MVC 4
<b>Database Connectie</b>	Entity Framework
<b>Front-end Scripting</b>	Razor   HTML 5   CSS 3   Javascript   jQuery   RequireJS   CanJS

## 6.5 Gebruikers Interactie

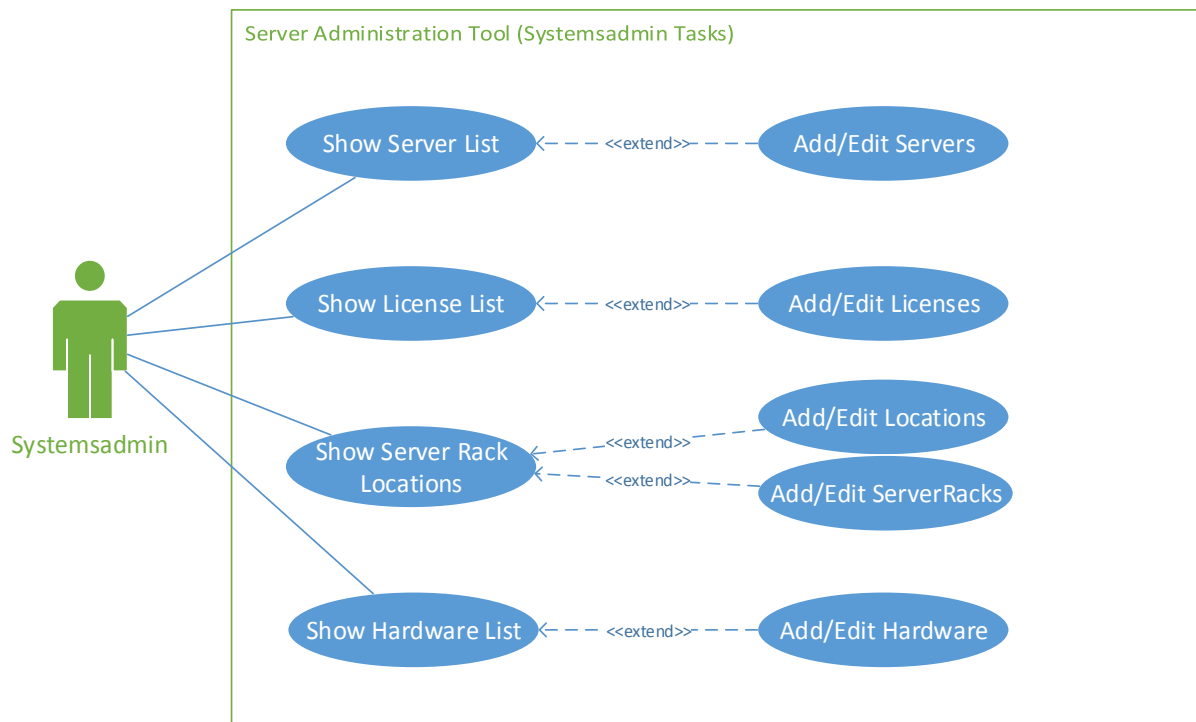
De globale interactie van de gebruikers met de applicatie staat hieronder in verschillende Use Case Diagrammen (UCD) uitgebeeld.



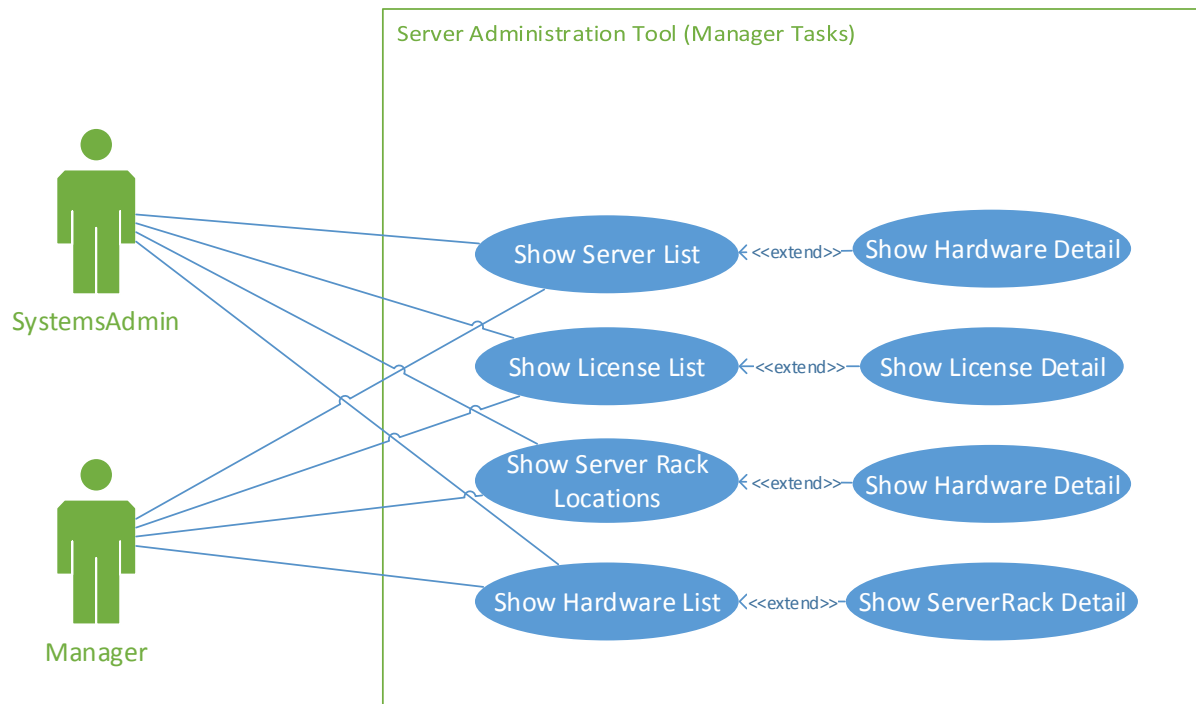
Figuur 5 UCD Login (Alle Gebruikers)



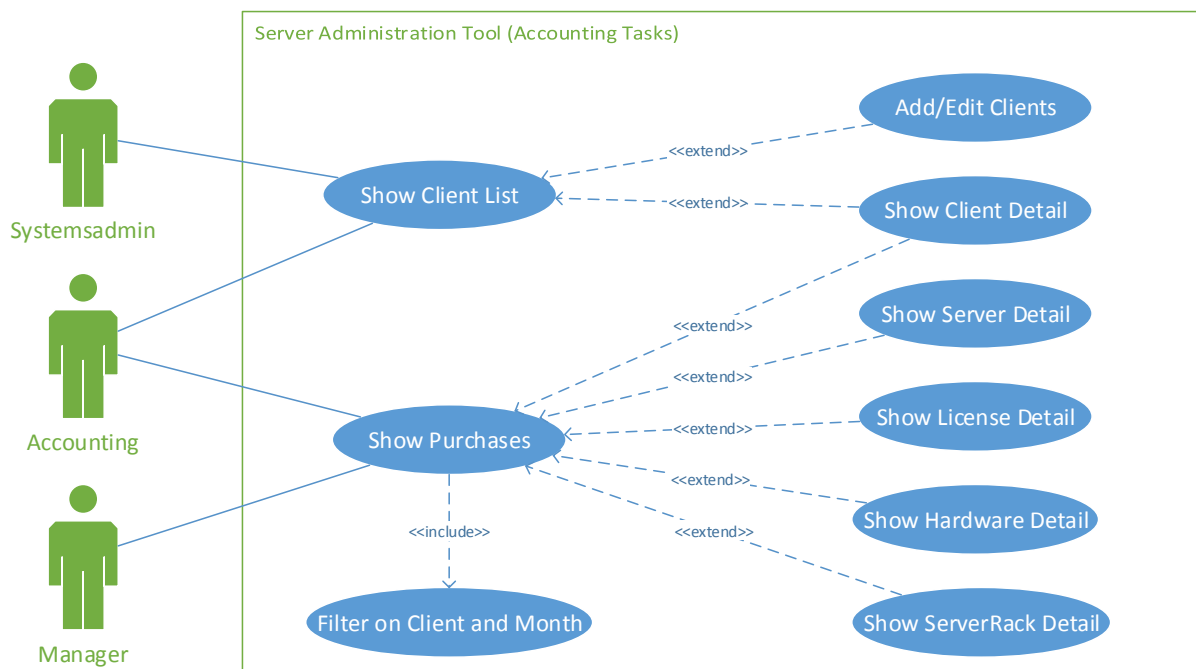
Figuur 6 UCD Administrator Acties



Figuur 7 UCD Systeembeheer Acties



Figuur 8 UCD Manager Acties

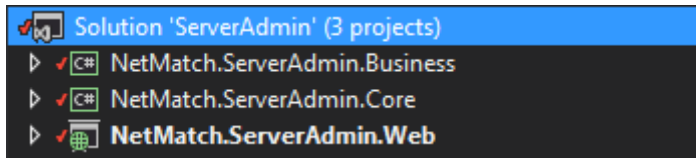


Figuur 9 UCD Accounting Acties

## 7. De Implementatiefase

### 7.1. Opzet Project

In het begin van de implementatiefase ben ik eerst begonnen met het opzetten van de structuur van de applicatie. Per project wordt in hoofdstuk 7.1.1. t/m 7.1.3. nog toegelicht wat de belangrijke mappen en bestanden zijn voor de projecten.



Figuur 10 Screenshot uit VS 2012 met de projectstructuur van de applicatie

#### 7.1.1. Web Project

*Mappen:*

<b>App_Start</b>	Bevat standaard config klassen vanuit ASP.NET MVC 4.
<b>Cache</b>	Hierin worden externe resources voor de layout van de website opgeslagen (zoals een profielfoto van een gebruiker).
<b>Content</b>	Hierin staan de afbeeldingen, stylesheets en fonts die voor de layout van de website gebruikt worden.
<b>Scripts</b>	Hierin staan alle Javascript bestanden die door de site gebruikt worden.
<b>Views</b>	Hierin staan alle template bestanden voor ASP.NET (Razor) om de view van de website op te bouwen.

*Bestanden:*

<b>Global.asax</b>	De eerste file die gerunt wordt door de web applicatie. Deze klasse voert alle acties uit die aan het begin van de applicatie moeten gebeuren, zoals het initialiseren van alle klassen in App_Start
<b>Web.config</b>	Hierin staan alle ASP.NET settings van de website. Deze settings bepalen het gedrag van de website van security tot connectie strings voor bijvoorbeeld een database.

#### 7.1.2. Core Project

*Mappen:*

<b>Models</b>	Hierin komen alle klassen te staan die de basis informatie van de applicatie bevatten.
<b>DataAccess</b>	Hierin worden alle klassen geplaatst die data moeten benaderen van externe bronnen.

#### 7.1.3. Bussiness Project

*Mappen:*

<b>Controllers</b>	Hierin staan klassen die er voor zorgen dat de juiste view met de juiste data worden voorzien.
--------------------	--

## 7.2. Omgevingen

Voordat er begonnen kan worden met het ontwikkelen en testen van de applicatie moet er eerst omgevingen opgezet worden waarop dit kan gebeuren.

Systeembeheer is verantwoordelijk voor het opzetten en beheren van alle website omgevingen, er is dan ook gevraagd of zij de omgevingen op willen zetten. Hieronder staan de opgezette omgevingen verder omschreven.

### 7.2.1. Ontwikkelomgeving

De ontwikkelomgeving wordt gebruikt door de ontwikkelaar, hier wordt de omgeving opgezet, ontwikkeld en getest of de applicatie werkt.

<b>Locatie:</b>	De ontwikkelaars lokale pc.
<b>Database:</b>	NMT-Web-02\Dev-ServersAdmin
<b>Integrated development environment (IDE):</b>	Visual Studio 2012

### 7.2.2. Testomgeving

Zodra een product/scherm af is in de ontwikkelomgeving zal deze live gezet worden in de testomgeving. Hier gaan de eindgebruikers testen of de schermen en het programma aan de gewenste functionaliteiten voldoen en wat er aangepast dient te worden.

<b>Locatie:</b>	NMT-Web-02
<b>Url:</b>	<a href="http://dev-serversadmin.netmatch.nl">http://dev-serversadmin.netmatch.nl</a>
<b>Database:</b>	NMT-Web-02\Dev-ServersAdmin

### 7.2.3. Productieomgeving

In deze omgeving komt het uiteindelijke product te draaien, welke de eindgebruikers in productie gaan gebruiken.

<b>Locatie:</b>	NMT-Web-02
<b>Url:</b>	<a href="http://serversadmin.netmatch.nl">http://serversadmin.netmatch.nl</a>
<b>Database:</b>	NMT-Web-02\ServersAdmin

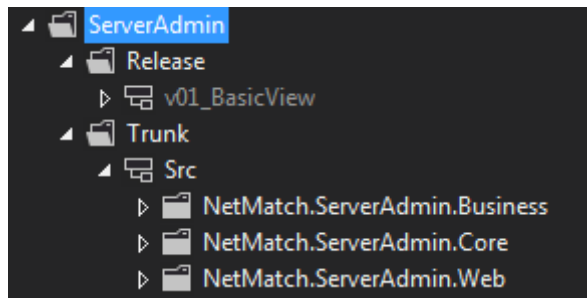
## 7.3. Versiebeheer

Alle projecten bij NetMatch worden bijgehouden op een Team Foundation Server (TFS) ([Hexspoor, 2011](#)). Dit project is daar geen uitzondering op. Dit project is op de NetMatch TFS server aangemaakt onder SystemsAdministration/ServerAdmin.

### 7.3.1. Mappen Structuur

In figuur 11 zie je een screenshot van de mappen structuur van het project binnen TFS.





Figuur 11 Screenshot VS 2012 TFS Explorer

### 7.3.2. Branching

Elke keer als een nieuw onderdeel van de applicatie getest wordt in de testomgeving wordt er een branch gemaakt van de ServerAdmin/Trunk/Src met een bijpassende naam en geplaatst onder de ServerAdmin/Release folder (zie figuur 11).

## 7.4. Applicatie Ontwikkelen

### 7.4.1. Basis opzetten

Na het opzetten van de omgevingen en de versiebeheer kon er eindelijk begonnen worden aan het bouwen van de applicatie. Als eerst is de Database opgezet zoals deze in figuur 2 te zien is.

Ik heb gekozen om Entity Framework Database First te gebruiken voor het genereren van alle model klassen binnen de applicatie, deze methode zorgt ervoor dat ik niet alle modelklassen uit hoeft te schrijven terwijl de logica en structuur al vast liggen in de database.

Bij het implementeren van de modellen vanuit de database waren er echter wat problemen met VS 2012 en de Entity Framework plugin, deze ontbrak namelijk. Na in VS2012 meerder plugins te de-installeren en installeren, VS 2012 te herinstalleren en de Windows registry op te schonen werkte VS 2012 weer naar behoren.

Toen Entity Framework met de Database first connectie er eindelijk goed inzat, constateerde ik dat de benaming van de tabellen volgens de MySQL standaard niet handig waren in de MS SQL database. De gegenereerde model klassen, namen de tabel als klasse naam over. Hierdoor heb ik de tabellen en hun properties hernoemt, zodat de applicatie begrijpelijke en standaard klasse namen heeft en de database voldoet aan de standaard naamconventies die NetMatch gebruikt binnen zijn projecten.



Figuur 12 Uiteindelijke Entity Framework Diagram na het opzetten van de basis.

Na de koppeling met de database heb ik eerst de globale layout van de applicatie ingevoegd, zodat de andere schermen die later ontwikkeld moeten worden vanaf een basis aangeroepen kunnen worden. Hierbij moest ik wennen aan werken met ASP.NET in combinatie met de Razor view engine. Om de basis van ASP.NET, MVC 4 en Razor te leren heb ik het volgende boek geraadpleegd (Galloway, Haack, Wilson, & Allen, 2012).

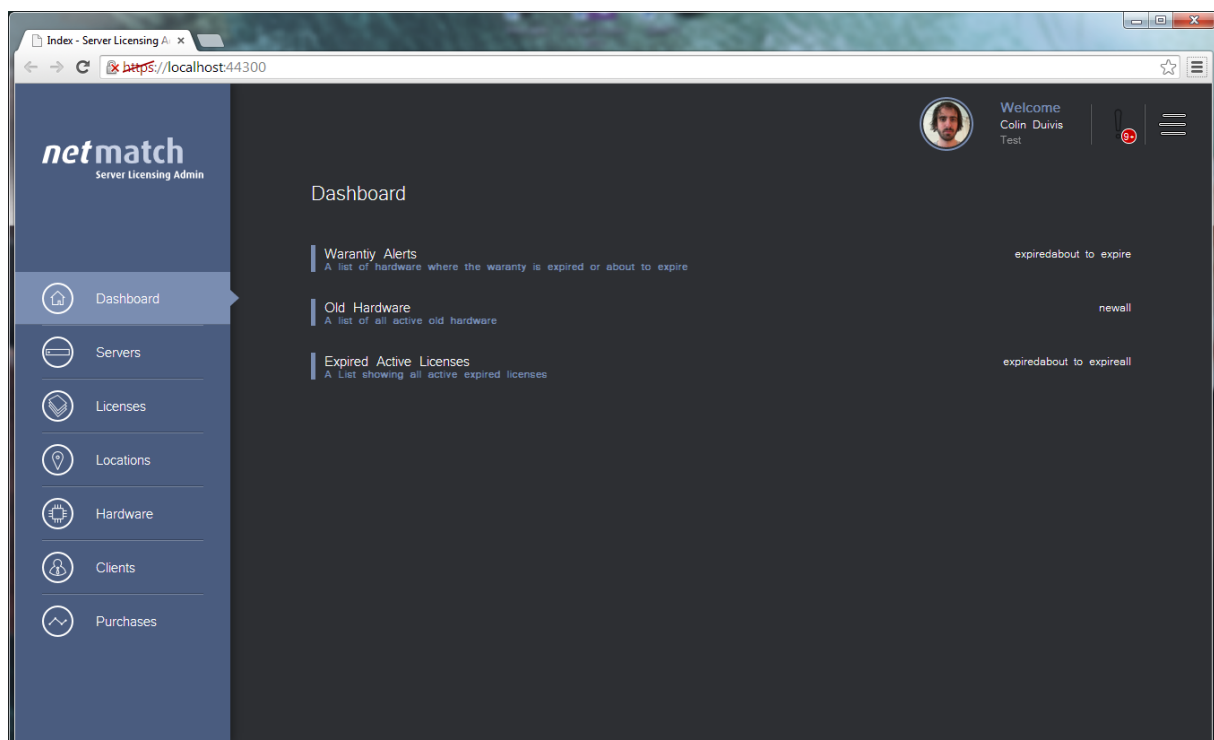
Met het design verwezenlijkt in de website en de koppeling met de database heb ik de eerste versie van de website op de testomgeving online gezet, zodat iedereen bekend kon raken met de website. Thuis constateerde ik dat de test omgeving publiek benaderbaar was vanaf het internet. Aangezien het niet de bedoeling is dat iedereen in de test omgeving kan, is er een ticket geplaatst bij

systeembeheer met de vraag of de website alleen voor intern netwerk van NetMatch benaderbaar gemaakt kon worden. Dit is gedaan door via IIS een Windows login af te dwingen. Hierdoor hebben allen mensen die rechten worden gegeven op de website via IIS toegang om in te loggen.

In eerste instantie wou ik het security en authenticatie gedeelte bewaren tot op het einde aangezien deze niet noodzakelijk was voor de basis functionaliteit van het systeem, aangezien deze al geconfigureerd was op de server heb ik gekozen het ook gelijk goed te implementeren. De website is zo ingesteld dat hij communicatie over HTTPS afdwingt. Verder moest de authenticatie uitgebreid worden zodat er een check wordt gedaan tegenover de database en Active Directory (AD).

Ik heb ervoor gekozen zoveel mogelijk de functionaliteiten van ASP.NET en MVC te gebruiken als deze aanwezig zijn, aangezien het niet rendabel is qua tijd om deze codes en structuren zelf te bedenken, plus dat de kansen op fouten bij eigen geschreven code groter zijn. Zo heb ik ook voor de autorisatie en authenticatie van gebruikers en rollen voor de MVC implementatie gekozen. Deze implementatie zorgt ervoor dat je pagina's makkelijk toegankelijk kan maken voor alleen specifieke gebruikers, groepen of rollen. MVC authenticatie en autorisatie: (Wasson, 2012), (Galloway, Haack, Wilson, & Allen, 2012).

Echter bleek het koppelen van de authenticatie en autorisatie gegevens van MVC aan de AD gegevens van NetMatch en de gebruikers gegevens van de applicatie niet makkelijk te zijn. Uiteindelijk is er een methode gevonden die inhaakt op de authenticatie van MVC, waardoor de geauthentiseerde gebruiker, met eigen gegevens uit de applicatie aangevuld kunnen worden. Deze oplossing is gebaseerd op de volgende bron: (John, 2013).



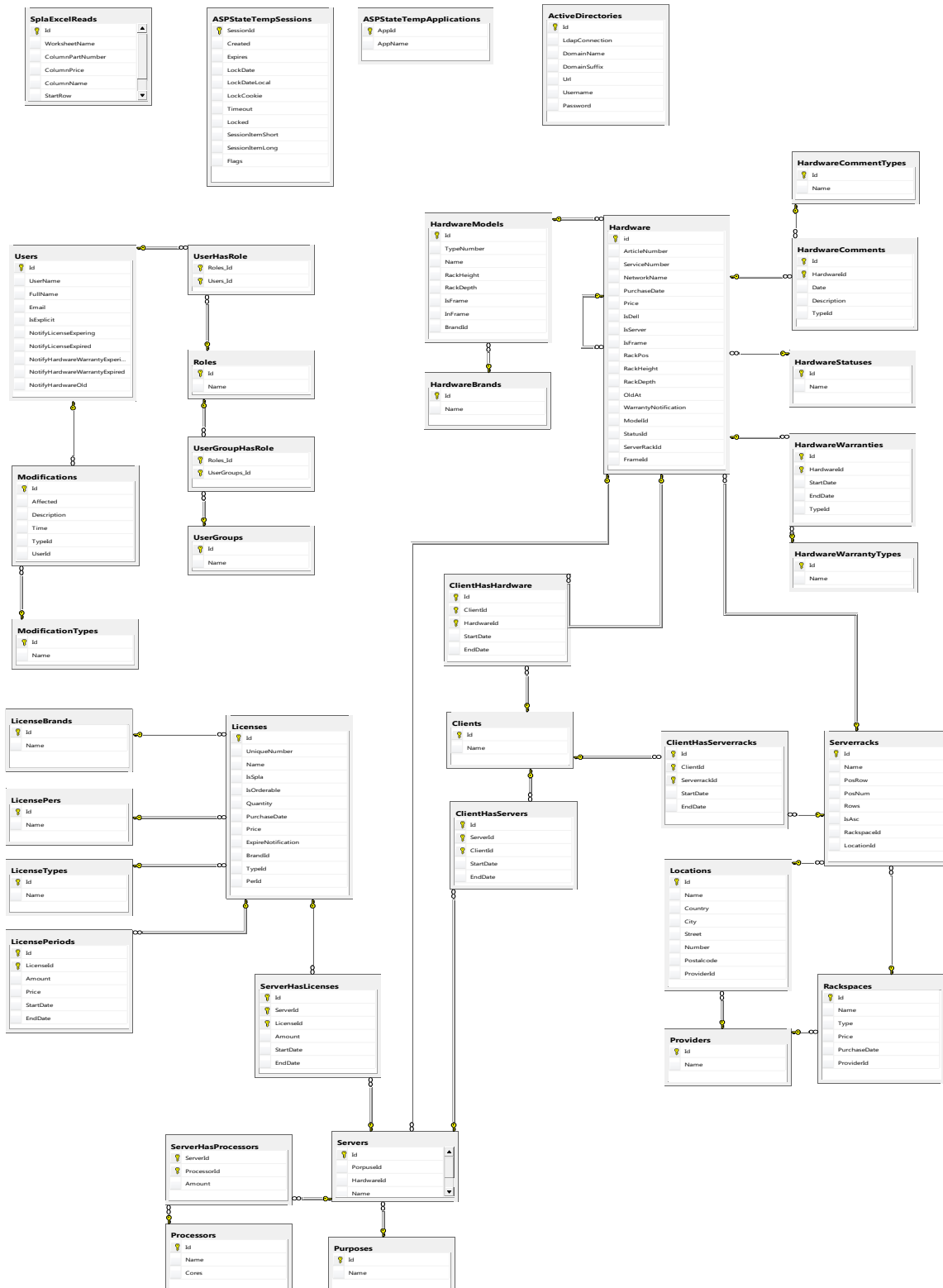
Figuur 13 Screenshot van de website na opzetten van de basis.

Verder moest er voor de applicatie nog een goede manier komen om de sessie data op te slaan voor de user. Ik heb besloten dit server-side te doen zodat er geen gevoelige data client-side opgeslagen wordt. Hiervoor was vanuit MVC al een standaard implementatie. Om het werkend te krijgen moest er alleen een script uitgevoerd worden die 2 nieuwe tabellen aan de database toe voegde en een kleine code toegevoegd worden aan de web-config. Aan de server-side wordt nu alles opgeslagen in

de database en aan de client-side wordt er een cookie opgeslagen. In deze cookie wordt alleen een encrypted uniek nummer van de sessie opgeslagen, waarmee de gebruiker zijn waardes kan ophalen.

Web.config code:

```
<sessionState cookieless="UseCookies" timeout="240" mode="SQLServer"
allowCustomSqlDatabase="true" sqlConnectionString=" " />
```



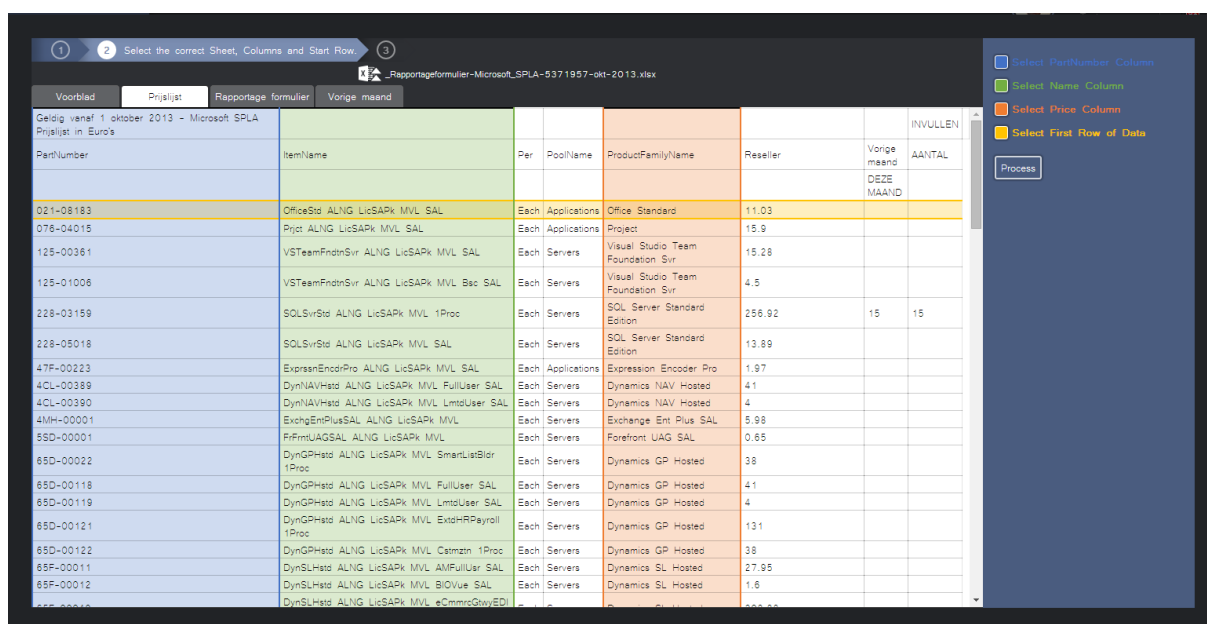
Figuur 14 Database na opzet van de basis

Voordat ik de eerste schermen ben gaan opzetten heb ik eerst afgesproken met Ron Otten (Otten, Front-end Technologien, 2014) om te kijken wat de gebruikelijke front-end technologieën zijn bij NetMatch. Na aanleiding van dit gesprek heb ik de standaard mappenstructuur voor Javascript overgenomen van NetMatch aangezien deze duidelijk inzicht geeft in welke javascript bestanden bij welke front-end pages horen. Verder heb ik de RequireJS library geïmplementeerd, zodat Javascript bestanden asynchroon ingeladen worden op het moment dat ze nodig zijn.

## 7.4.2. Iteratief functionaliteiten opleveren

### Iteratie 1: Bulk SPLA Upload

Nu de basis in het systeem zit kan er begonnen worden aan het ontwikkelen van de individuele schermen. Ik ben begonnen met het bulk upload scherm voor de SPLA lijsten. Aangezien het bijhouden van de maandelijkse SPLA licenties één van de belangrijkste taken is van het systeem.



Figuur 15 Screenshot van het selecteren van de kolommen en rijen om het spla-excel bestand goed uit te kunnen lezen.

### Iteratie 2: Licentie, Server en Klant Overzichten

Na het opleveren van de bulk SPLA upload ben ik in overeenstemming met Jeroen Havermans begonnen aan het opzetten van de overzichten van licenties, klanten en servers. Aangezien deze applicatie veel overzichten moet tonen van verschillende model klassen heb ik besloten om een generiek systeem te ontwikkelen die het mogelijk maakt een paar attributen toe te voegen aan een Entity Framework model klasse waardoor er “automatisch” een overzicht gegenereerd kan worden met bij behorende filters. De filters zullen toegevoegd worden wanneer deze nodig zijn.

```
class LicenseMetadata
{
    [Id]
    public int Id { get; set; }

    [Column("Unique Number", Order = 1)]
    public string UniqueNumber { get; set; }

    [Column("Name", Order = 2, Orderable = true)]
    public string Name { get; set; }

    public System.DateTime PurchaseDate { get; set; }

    [SlideDecimalFilter]
    public Nullable<decimal> Price { get; set; }

    [Column("", ColumnType = ColumnAttribute.Type.Container)]
    public virtual LicenseBrand LicenseBrand { get; set; }

    [Column("", ColumnType = ColumnAttribute.Type.Container)]
    public virtual LicenseType LicenseType { get; set; }
}
```

Figuur 16 Screenshot van model klasse met attributen voor overzicht generatie

Unique Number	Name	License Type	License Brand	Operations
021-08183	OfficeStd ALNG LicSAPk MVL SAL	SPLA	Microsoft	
076-04015	Pjdt ALNG LicSAPk MVL SAL	SPLA	Microsoft	
125-00381	VSTeamFndtrSvr ALNG LicSAPk MVL SAL	SPLA	Microsoft	
125-01006	VSTeamFndtrSvr ALNG LicSAPk MVL Bsc SAL	SPLA	Microsoft	
228-03159	SOLSvrStd ALNG LicSAPk MVL 1Proc	SPLA	Microsoft	
228-05018	SOLSvrStd ALNG LicSAPk MVL SAL	SPLA	Microsoft	
47F-00223	ExpressEnclrPro ALNG LicSAPk MVL SAL	SPLA	Microsoft	
4CL-00389	DynNAVHstd ALNG LicSAPk MVL FullUser SAL	SPLA	Microsoft	
4CL-00390	DynNAVHstd ALNG LicSAPk MVL LmtUser SAL	SPLA	Microsoft	
4MH-00001	ExchgEntPlusSAL ALNG LicSAPk MVL	SPLA	Microsoft	
55D-00001	FirmUAGSAL ALNG LicSAPk MVL	SPLA	Microsoft	
65D-00022	DynGPHstd ALNG LicSAPk MVL SmerListBldr 1Proc	SPLA	Microsoft	
65D-00118	DynGPHstd ALNG LicSAPk MVL FullUser SAL	SPLA	Microsoft	
65D-00119	DynGPHstd ALNG LicSAPk MVL	SPLA	Microsoft	

Figuur 17 Screenshot van licentie overzicht

### Iteratie 3: Toevoegen Klanten en Servers

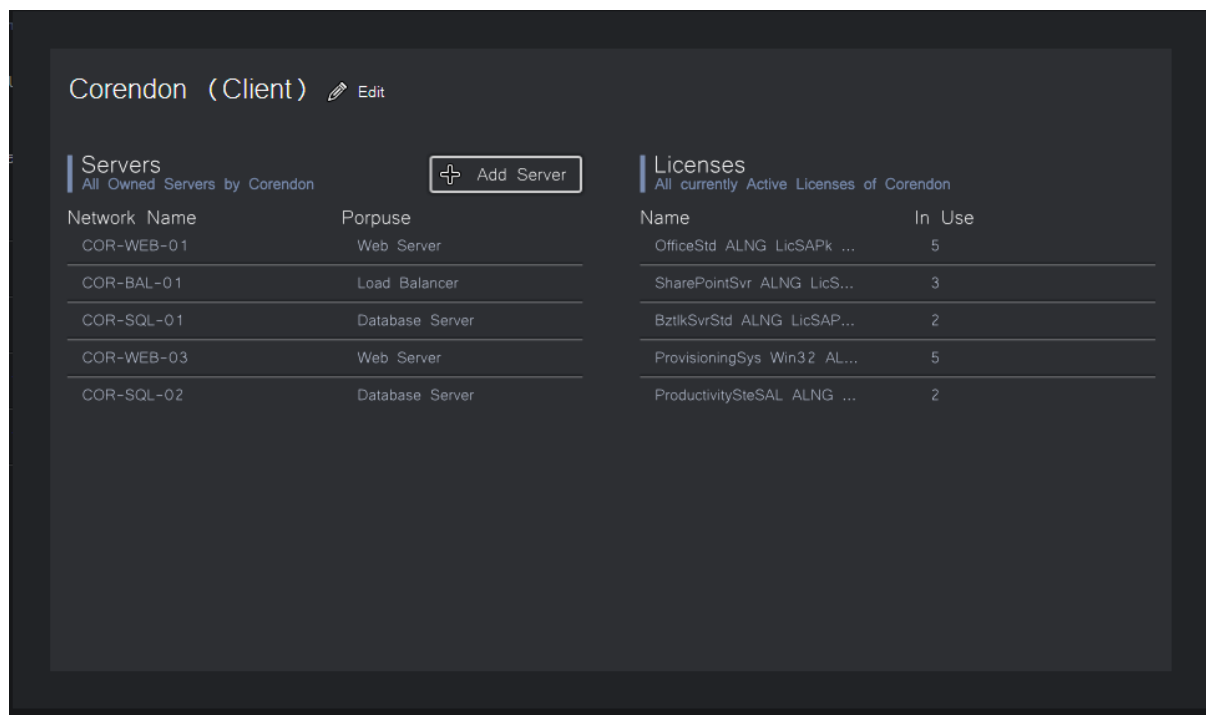
De volgende stap is het maken van simpele toevoeg mogelijkheden voor servers en klanten. Hier was het vooral belangrijk dat de servers en klanten toegevoegd konden worden, zodat deze vervolgens aan elkaar gekoppeld konden worden. Hier heb ik gebruik gemaakt van de CanJS library waarvan ik een workshop heb gehad van Ron Otten ([Otten, Workshop: Installing and using CanJS in your project, 2014](#)). Deze library zorgt ervoor dat je class objecten kan koppelen aan de view en deze individueel eenvoudig kan updaten en herladen zonder de hele pagina te moeten herladen.

Figuur 18 Screenshot van het toevoeg scherm voor een klant

#### Iteratie 4: Koppelen Servers, Klanten en Licenties

Nu alle basis ingrediënten aanwezig zijn voor de applicatie kunnen de interfaces uitgebreid worden voor het koppelen van de verschillende klassen.

Deze iteratie vindt op de tijd plaats dat de scriptie afgeschreven en opgeleverd wordt. De applicatie zal na het opleveren van de scriptie nog verder ontwikkeld worden. Na verwachting zal dit onderdeel ook af zijn voordat de afstudeerzitting plaats vindt.



Figuur 19 Mock-up klant detail pagina na koppeling van licenties, servers en klanten

## 8. Conclusie & Aanbevelingen

### 8.1. Conclusie

De basiscomponenten zoals licenties, servers en klanten kunnen nu bijgehouden worden. Dit zijn de kerncomponenten voor de applicatie. Op dit moment ontbreken er nog een aantal functionaliteiten, zoals het koppelen van de licenties, servers en klanten en het vergelijken van prijzen uit verschillende maanden.

De applicatie is op dit moment nog niet zover dat hij in productie genomen kan worden, maar het heeft wel een sterke basis waarin het grotendeel van de logica al aanwezig is om de applicatie goed verder op te bouwen. De sterke basis van het programma uit zich vooral in de structuren en technieken die nu al in de website zitten. Enkele voorbeelden van deze structuren en technieken zijn:

- Eenvoudig kunnen generen van overzicht pagina's.
- Koppeling met Database en Active Directory.
- NetMatch project structuur gebruikt voor een herkenbare opbouw voor doorontwikkeling.

Achteraf gezien was de gehele applicatie bouwen binnen een afstudeerperiode door 1 persoon, iets te optimistisch, vooral aangezien je niet alleen de applicatie ontwikkeld binnen die tijd, maar ook een onderzoek moet uitvoeren. Ik blijf echter nog steeds achter de keuze van het onderzoek staan om een eigen applicatie te bouwen, aangezien de meeste applicaties niet in de buurt kwamen bij de eisen die aan de applicatie gesteld werden en het de vraag zou zijn in hoeverre de applicaties aan te passen zijn.

### 8.2. Aanbevelingen

Mijn aanbeveling is om het systeem verder te ontwikkelen, zoals deze nu ontwikkeld is ( dus iteratief samen met de eindgebruikers bepalen en kijken welke richting de applicatie op moet gaan ).

Verder beveel ik aan de volgende functionaliteiten als eerst volgende in te bouwen (in overleg met de eindgebruikers natuurlijk):

1. Koppeling servers klanten en licenties.
2. Licentie filteren op het gebruik van licenties.
3. Licentie filteren op maanden waarin ze gebruikt worden.
4. Licenties filteren op klant.
5. Server filteren op klant.

Na deze functionaliteiten heb je namelijk de mogelijkheid licenties op te zoeken bij klant en terug in de geschiedenis te kijken wanneer licenties gebruikt zijn en wat de prijzen waren in die maanden.

Om tot een rudimentaire versie te komen die in productie genomen kan worden, zouden eerst de bovengenoemde functionaliteiten geïmplementeerd moeten worden. Vervolgens zouden de volgende acties nog uitgevoerd moeten worden:

1. Toevoegen van de mogelijkheid om uit te loggen.
2. Een scherm toevoegen om gebruikers te kunnen beheren.
3. Het toevoegen van de functionaliteit een lijst te kunnen genereren de te bestellen licenties.
4. En als laatste het verwijderen van view elementen die op dit moment niet gebruikt worden, zoals de warning-box en enkele pagina's die nog geen functionaliteiten bevatten.



## Evaluatie / Reflectie

### PID

Het PID was vrij snel opgezet (binnen een week). Bij het beoordelen kreeg ik alleen een paar opmerkingen over taal gebruik en het anders formuleren van sommige zinnen, dus ik neem aan dat het inhoudelijk goed in elkaar zit.

### Onderzoek

Het onderzoek duurde aanzienlijk langer dan ik ingepland had in de PID i.p.v. 3 weken heeft het onderzoek me 6 weken gekost. Dit lag vooral aan de verschillende gesprekken die ingepland moesten worden met de verschillende experts en mensen die ik nodig had voor informatie, plus dat verschillende afdelingen de requirements moesten goedkeuren dat ook nog enige tijd heeft gekost. Met deze tijden had ik in de planning van de PID geen rekening gehouden.

Doordat ik een paar dagen heb moeten wachten op goedkeuring van de requirements, voordat ik verder kon met mijn onderzoek, heb ik deze tijd gespendeerd aan het maken van Proof of Concepts (PoC) voor het uitlezen van Excel bestanden, zowel als voor connectie maken met de Active Directory. Dit heb ik gedaan om zowel mijn tijd nuttig te besteden, als de nieuwe technieken beter te leren gebruiken en te kijken of het haalbaar is dit zelf te bouwen.

Ook bij dit document had niet meer als één beoordeling nodig om een paar kleine spelfoutjes en een stukken tekst die net iets beter opgesteld konden worden, dus bij dit document ga ik er ook vanuit dat deze inhoudelijk goed is.

### Opzetten Applicatie

Het ontwerpen van de applicatie heeft me niet heel veel tijd gekost ik was hier binnen een week mee klaar zoals op de planning stond. Echter waren er onverwachte problemen met mijn VS 2012 die me een halve week a week hebben gekost om op te lossen, waardoor je weer meer achterstand oploopt dan in eerste instantie gepland.

De communicatie met systeembeheer was goed, ik heb voor het opzetten van de verschillende omgevingen een mail gestuurd. Dit was binnen een dag helemaal opgezet en klaar voor gebruik.

Verder heb ik voor dit project veel nieuwe technologieën en werkwijzen moeten leren die specifiek waren voor een Microsoft web project en me goed verdiept in wat de standaard werkwijze voor dit type project is bij NetMatch.

### Ontwikkelen Applicatie

Bij het ontwikkelen van de applicatie heb ik mezelf betrap op het vaak nadenken over hoe ik bepaalde onderdelen van de applicatie wil toevoegen en wat hier de beste locaties en structuur voor zijn.

Ik ben veel later begonnen aan het opleveren van de schermen dan ik verwacht had, hierdoor had ik uiteindelijk wel een worsteling met mezelf of ik het nu “quick and dirty” zou aanpakken, zodat ik veel schermen in de applicatie kon tonen, of dat ik het gestructureerd aanpak en een goede basis voor de applicatie neerleg. Ik heb het laatste gekozen, omdat ik wil dat op deze applicatie verder ontwikkeld kan worden zonder dat al de code herschreven moeten worden, omdat het onoverzichtelijk zou zijn, of omdat er veel fouten in zaten.

## Globaal

Ik heb veel nieuwe technologieën geleerd, wat altijd positief is. Echter het implementeren van nieuwe technologieën kost in de ontwikkeling altijd meer tijd, omdat je gaandeweg pas leert deze technologieën goed te gebruiken.

Verder vind ik dat ik goed nagedacht heb over de structuur en opzet van het project. Vooruitdenkend dat deze applicatie verder ontwikkeld gaat worden door NetMatch.

Het plannen van een project kan nog wel beter, ik zou hier op elke fase beter over na mogen denken of er onvoorziene fouten kunnen zijn en hiervoor buffers inplannen.

Ik heb geleerd dat het opzetten van een applicatie vanaf het begin, veel tijd kost als je hem gelijk future-proof wilt bouwen, dus zorgen dat er nog verder aan ontwikkeld kan worden door bijvoorbeeld opzetten van generieke libraries die het ontwikkelen van het product uiteindelijk versnellen.

Ik vind dat de applicatie qua basis goed opgezet is, alleen ik had op dit tijdstip in het project graag wat meer schermen af gehad en uitgebreider willen hebben.

## Literatuur Lijst

- Andriyanov, A., Dummer, A., & Ovsienko, D. (2013). *About*. Opgehaald van RackTables: <http://racktables.org/about.php>
- Campfens, C. (2014, 04 09). NetMatch Project Structuren. (C. Duivis, Interviewer)
- Galloway, J., Haack, P., Wilson, B., & Allen, K. S. (2012). *Professional ASP.NET MVC 4*. John Wiley & Sons, Inc.
- Hexspoor, L. (2011, 05 16). Versioning and branching the NetMatch Shared Libraries.
- John. (2013, 06 06). *Simple claims transformation and caching w/ windows authentication*. Opgehaald van StackOverflow: <http://stackoverflow.com/questions/16904639/simple-claims-transformation-and-caching-w-windows-authentication>
- NetMatch. (2013, 06 26). *NetMatch C# Coding Conventions*. Opgehaald van portal.netmatch.nl: <https://portal.netmatch.nl/expertise/dev/Guidelines/Code%20style%20guidelines/CSharp/NetMatchCSharpCodingConventions.html>
- Otten, R. (2014, 04 16). Front-end Technologiën. (C. Duivis, Interviewer)
- Otten, R. (2014, 05 20). Workshop: Installing and using CanJS in your project. Tilburg, Noord Brabant, Netherlands: NetMatch.
- Wasson, M. (2012, 11 27). *Authentication and Authorization in ASP.NET Web API*. Opgehaald van ASP.NET: <http://www.asp.net/web-api/overview/security/authentication-and-authorization-in-aspnet-web-api>

# Bijlage I: Project Initiation Document

Server Administration Tool

*Versie 1.1*



***net*match**

## Document Informatie

Datum voltooid: 25-2-2014

Status: Afgerond

## Auteur

Geschreven door: Colin Duivis

Studentnummer: 2192461

## Bedrijf

Naam: NetMatch B.V.

Plaats: Tilburg

## Document Historie

Date	Version	Author	Description
<b>2014-02-11</b>	0.1	Colin Duivis	Opzet van het document
<b>2014-02-17</b>	0.2	Colin Duivis	Verwerken feedback PID workshop
<b>2014-02-18</b>	1.0	Colin Duivis	Feedback van Jeroen Havermans verwerken ( NetMatch )
<b>2014-02-21</b>	1.1	Colin Duivis	Feedback van Erik Mols verwerken ( docentbegeleider )

## Goedkeuring Versie 1.1

*Projectbegeleider*

Naam:

Datum:

Paraaf:

*Docentbegeleider*

Naam:

Datum:

Paraaf:

## Managementsamenvatting

### Doel Document

Dit document is gemaakt om inzicht te geven in het te maken project. En om Fontys Hogescholen ( school ), Colin Duivis ( ontwikkelaar ) en NetMatch ( opdrachtgever ) op een lijn te krijgen betreffende wat de aanpak van het project gaat worden en wat de op te leveren producten zijn.

### Aanleiding

NetMatch levert hosting services op het Windows & Linux platformen aan hun eindklanten. NetMatch heeft meerdere serverracks, verspreid over meerdere Nederlandse datacenters: webserver, SQL-server, applicatie servers, load balancers, caching servers, etc.

Bij het beheren van het platform van NetMatch komen ook veel randzaken kijken, denk hierbij aan het administreren en rapporteren van software licenties, bijhouden van lopende support contracten, een change log bijhouden, garanties bijhouden en het bij houden van Service Level Agreements ( SLA ).

Deze administratie is op dit moment verdeeld over meerdere tools en Excel administraties. Het streven is om één web applicatie te implementeren met één gecentraliseerde administratie.

### Globale Aanpak

Voor de begin fase van het project is er gekozen voor een stapsgewijze projectstructuur waar met interviews met de projectbegeleider, interviews met de eindgebruikers en observatie van het huidige systeem vastgelegd kan worden wat de huidige eisen zijn van het systeem.

Op het moment dat de requirements van het project vast staan, word het onderzoek gestart.

Als de uiteindelijke software geïmplementeerd gaat worden wordt er een iteratieve methodiek toegepast om zo te kunnen toetsen of het project de richting op blijft gaan die de projectbegeleider en de eindgebruikers willen. Op deze manier is er nauwe samenwerking tussen de partijen en blijft iedereen betrokken bij het project waardoor het eindresultaat ook echt iets is waar de gebruikers en de projectbegeleider profijt bij hebben.

### Globale Kosten en Doorlooptijd

Voor dit project zijn 85 werkdagen ( 8 uur per dag ) te besteden om het te maken met een uitloop van maximaal 10 werkdagen. Mocht het blijken dat er eventuele investeringen nodig zijn voor de "Server Administration Tool" of software aangeschaft dient te worden ter ontwikkeling, dan kan dit op dat moment overlegd worden met NetMatch.

## Inhoudsopgave

Document Historie .....	1
Managementsamenvatting .....	2
Doel Document.....	2
Aanleiding.....	2
Globale Aanpak .....	2
Globale Kosten en Doorlooptijd .....	2
1. Inleiding .....	5
1.1. Opbouw Document .....	5
2. Achtergrond.....	6
2.1. Aanleiding.....	6
2.2. Context Opdracht .....	6
2.3. Context Onderzoek.....	7
2.4. Huidige situatie.....	7
3. Projectdefinitie .....	8
3.1. Projectdoelstellingen.....	8
3.1.1. Doel .....	8
3.1.2. Gewenste Situatie .....	8
3.1.3. Voordelen en Mogelijkheden .....	8
3.2 Project Aanpak .....	8
3.3. Project Scope .....	8
3.4. Productdecompositiestructuur ( PDS ) op project niveau.....	10
3.5. Productstroomschema ( PSS ) op project niveau .....	11
3.6. Productbeschrijving op project niveau.....	11
3.7. Projectbudget .....	12
3.8. Wijzigingsbudget .....	12
3.9. Benodigde resources .....	12
3.10. Uitsluitingen .....	12
3.11. Afhankelijkheden.....	12
3.12. Randvoorwaarden .....	12
3.13. Aannames.....	12
4. Projectorganisatiestructuur .....	13
4.1. Projectbegeleider .....	14
4.2. Ontwikkelaar .....	14
4.3. Eindgebruiker Accounting .....	14
4.4. Eindgebruiker Systeembeheer .....	14

4.4.1. Senior.....	14
4.5. Docentbegeleider .....	14
4.6. Assessor .....	14
5. Projectbeheersing .....	15
5.1. Rapportage .....	15
5.1.1. Documenten .....	15
5.1.2. Afspraken.....	15
5.2. Voortgangsbewaking .....	15
5.2.1. Planning Project.....	15
5.3. Toleranties.....	15
5.3.1. Projecttolerantie .....	15
5.4. Risicomanagement .....	15
5.5. Kwaliteitsbewaking.....	16
5.6. Wijzigingsprocedure.....	16



## 1. Inleiding

Dit document is gemaakt om inzicht te geven in het te maken project. En om Fontys Hogescholen ( school ), Colin Duivis ( ontwikkelaar ) en NetMatch ( opdrachtgever ) op een lijn te krijgen betreffende wat de aanpak van het project gaat worden en wat de op te leveren producten zijn.

### 1.1. Opbouw Document

Hieronder volgt een opsomming van de opbouw van het document en welke zaken in de verschillende hoofdstukken besproken zullen worden.

- **Achtergrond:** Hierin wordt de achtergrond van het bedrijf en de opdracht besproken.
- **Projectdefinitie:** Maakt duidelijk wat er precies gemaakt dient te worden en wat hiervoor de aanpak is.
- **Projectorganisatiestructuur:** Legt vast wie er allemaal betrokken is binnen het project en wat de verantwoordelijkheden zijn van deze betrokkenen.
- **Projectbeheersing:** Legt vast welke afspraken gemaakt zijn. Verder geeft dit hoofdstuk inzicht in wat de risico's binnen het project zijn en hoe met deze risico's wordt omgegaan.

## 2. Achtergrond

NetMatch is een bedrijf dat zich specialiseert in het ontwikkelen van e-commerce software voor tour operators. Hierbij wordt gebruik gemaakt van de meest recente Microsoft technologieën, zoals: Visual Studio 2013, .NET 4.5, ASP .NET MVC4 en SQL Server 2012.

NetMatch heeft zich gevestigd in Amsterdam, Tilburg en in Cluj-Napoca ( Roemenië ). Verspreid over deze locaties heeft NetMatch ongeveer 70 werknemers.

NetMatch heeft 2 producten/services waarin ze zich groot in maken:

1. **TourWeb:** Tourweb is een totaal modulair systeem die gericht is op de markt van tour operators en online reisbureaus. Dit product geeft de mogelijkheid je huidige reissystemen uit te breiden, maar TourWeb kan ook zo ingezet worden dat je een vanuit hier je eigen reissysteem maakt. TourWeb is een van de systemen die NetMatch goed onderhoud en tot de dag van vandaag blijft door ontwikkelen.
2. **Hosting:** Door de vele jaren ervaring die NetMatch heeft opgedaan in hosting van eigen servers en servers voor klanten, zijn ze expert geworden in het onderhouden en inrichten van servers en hierop allerlei services te hosten voor klanten. Deze ervaring gebruikt NetMatch dan ook om maatwerk Hosting oplossingen voor bedrijven te bieden.

### 2.1. Aanleiding

NetMatch levert hosting services op het Windows & Linux platformen aan hun eindklanten. NetMatch heeft meerdere serverracks, verspreid over meerdere Nederlandse datacenters: webserver, SQL-server, applicatie servers, load balancers, caching servers, etc.

Bij het beheren van het platform van NetMatch komen ook veel randzaken kijken, denk hierbij aan het administreren en rapporteren van software licenties, bijhouden van lopende support contracten, een change log bijhouden, garanties bijhouden en het bij houden van Service Level Agreements ( SLA ).

Deze administratie is op dit moment verdeeld over meerdere tools en Excel administraties. Het streven is om één web applicatie te implementeren met één gecentraliseerde administratie.

### 2.2. Context Opdracht

Er dient een applicatie geïmplementeerd te worden waarmee NetMatch hun verschillende servers administratief kan inzien en beheren.

De opdracht bestaat uit de volgende onderdelen:

1. **Analyseren:** Middels een analyse van de huidige tools en interviews met de interne gebruikers, dient er vastgelegd te worden welke eisen / requirements aan de "Server Administration Tool" toegeschreven kunnen worden.
2. **Ontwerpen:** Na de analyses dienen deze in verschillende ontwerpen vast te worden gelegd hoe deze uiteindelijk in de applicatie komen, denk hierbij aan de volgende ontwerpen:
  - a. Grafisch user interface ontwerp
  - b. Database ontwerp
  - c. Applicatie architectuur
3. **Implementeren:** Op basis van de uitkomst van het onderzoek wordt er een eigen ontwikkelde applicatie geïmplementeerd of een nog aan te schaffen bestaand stuk software.

### 2.3. Context Onderzoek

Er moet onderzocht worden wat de beste aanpak is voor het implementeren van een “Server Administration Tool” binnen NetMatch.

Het onderzoek bestaat uit de volgende stappen:

1. Onderzoek: Na het vastleggen van de requirements die in stap 1 en 2 van de opdracht word onderzocht, wordt er gekeken of er al huidige applicaties zijn die voldoen aan de gestelde requirements.
2. Analyseren: Op basis van verschillende criteria kijken of het meer passend is om deze applicaties aan te schaffen / aan te passen of dat het meer passend is om een eigen applicatie te implementeren.
3. Afronden: Conclusies trekken op basis van de uitkomst in stap 2 en een aanbeveling doen voor een aanpak.

### 2.4. Huidige situatie

Momenteel is de server administratie en de licenties behorend bij de servers verdeeld over 2 plekken:

1. De administratie van de klanten, kosten en de licenties zitten in 1 groot excel bestand opgenomen. Waarbij handmatig maandelijks de prijzen up-to-date gehouden moeten worden.
2. De administratie van de locatie van de servers en de indeling van de server kasten zit in een programma opgenomen die RackTables<sup>[1]</sup> heet.

[1] Opensource software pakket: <http://racktables.org/about.php>

Eén keer in het jaar wordt er vanuit SEQUINT een nieuwe prijslijst doorgegeven van nieuwe licenties en de nieuwe licentie prijzen ( of in geval van een nieuw contract één keer per 3 jaar). Deze moeten vervolgens doorgevoerd worden naar de eerder genoemde Excel sheet om zo voor alle klanten de prijzen weer up-to-date te hebben.

Deze manier van documenteren word op dit moment erg omslachtig bevonden en buiten dat is de huidige manier ook nog eens hoogst foutgevoelig bevonden. Zo zijn er eind 2013 drie maanden lang de verkeerde aantallen gerapporteerd. Ook is er bij een audit van Microsoft al ondervonden dat het ophalen van alle gegevens over de servers waarop Microsoft licenties draaien een hele opgave is. Bij dit ophalen kwamen er ook weer enkele fouten binnen de administratie naar boven.

### 3. Projectdefinitie

#### 3.1. Projectdoelstellingen

##### 3.1.1. Doel

Het doel van het project is om de bijgehouden server administratie uit de huidige situatie (zie hoofdstuk 2.4.) samen te voegen in één applicatie. Deze applicatie moet centraal benaderbaar zijn, zodat de gespecificeerde eindgebruikers (zie hoofdstuk 3.3.) toegang hebben tot dezelfde informatie. Verder moet de applicatie ervoor zorgen dat de marge voor menselijke fouten geminimaliseerd wordt en dat de gebruiksvriendelijkheid toe neemt ten opzichte van de huidige situatie. Het streven is dat er binnen een jaar minder fouten zijn in de administratie ten opzichte van de huidige situatie.

##### 3.1.2. Gewenste Situatie

Eén applicatie waarin alle servers met de huidige gebruikte data bijgehouden kunnen worden, zodat de informatie snel gevonden kan worden en het updaten van de informatie weinig moeite kost.

##### 3.1.3. Voordelen en Mogelijkheden

- Het onderzoek wat gemaakt wordt geeft duidelijk aan welke richting NetMatch op moet voor de optimale implementatie voor een "Server Administration Tool".
- Het uiteindelijke product zorgt ervoor dat de licentie prijzen eenvoudig up-to-date gehouden kunnen worden.
- Het uiteindelijke product zorgt ervoor dat alle administratie van de servers overzichtelijk ingezien kan worden.

#### 3.2 Project Aanpak

Voor de begin fase van het project is er gekozen voor een stapsgewijze projectstructuur waar met interviews met de projectbegeleider, interviews met de eindgebruikers en observatie van het huidige systeem vastgelegd kan worden wat de huidige eisen zijn van het systeem.

Op het moment dat de requirements van het project vast staan, wordt het onderzoek gestart.

In het geval van een eigen te bouwen applicatie wordt het volgende o.a. gemaakt:

- Database Structuur (ERD)
- Software Architectuur (Klassen Diagrammen)
- Grafische Designs

In het geval van een uit te breiden applicatie worden de volgende delen in kaart gebracht:

- Wat de huidige Database Structuur is (ERD).
- Hoe de huidige software in elkaar zit (Klassendiagram).
- Op welke punten binnen de software er uitgebreid gaat worden (ERD + Klassendiagram).
- Welke nieuwe schermen er ontworpen moeten worden.

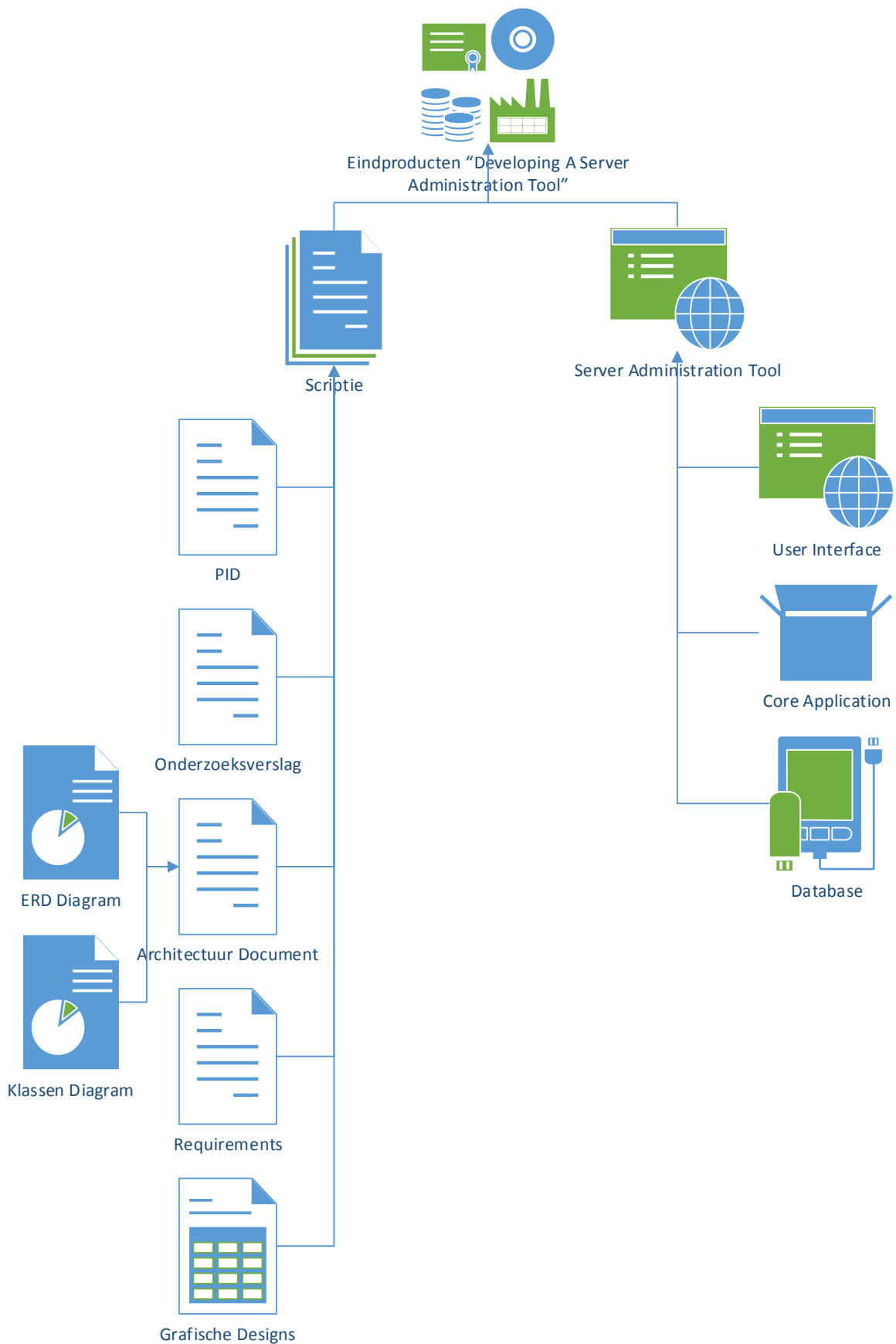
Als de uiteindelijke software geïmplementeerd gaat worden, wordt er een iteratieve methodiek toegepast om zo te kunnen toetsen of het project de richting op blijft gaan die de projectbegeleider en de eindgebruikers willen. Op deze manier is er nauwe samenwerking tussen de partijen en blijft iedereen betrokken bij het project waardoor het eindresultaat ook echt iets is waar de gebruikers en de projectbegeleider profijt bij hebben.

#### 3.3. Project Scope

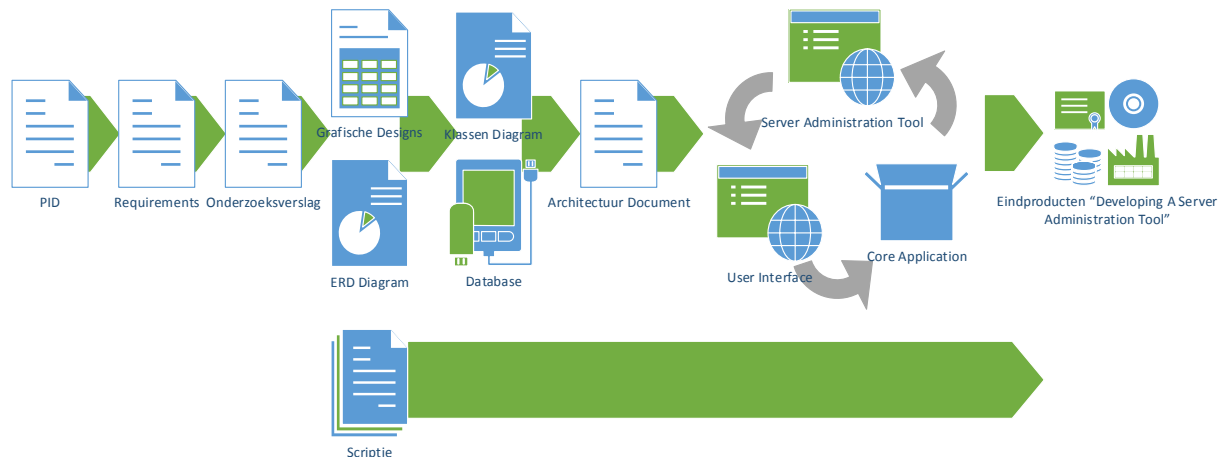
De scope van dit project blijft binnen NetMatch en betreft de volgende afdelingen en personen:

- NetMatch Systeembeheer (eindgebruiker)
- NetMatch Project Manager: Jeroen Havermans (projectbegeleider)
- NetMatch Boekhouding: Ronald Wisse (eindgebruiker)
- NetMatch Development: Colin Duivis (ontwikkelaar)
- Fontys Begeleiding: Erik Mols en/of Marcus Krielen (docentbegeleider)

## 3.4. Productdecompositiestructuur ( PDS ) op project niveau



### 3.5. Productstroomschema ( PSS ) op project niveau



### 3.6. Productbeschrijving op project niveau

De 2 belangrijkste producten voor NetMatch zijn het onderzoeksverslag en de “Server Administration Tool”. Voor school is het belangrijkste product de scriptie.

#### 3.6.1. Het onderzoeksverslag

Dit verslag is afhankelijk van wat er in de requirements vastgelegd gaat worden. Hier worden verder nog veel interviews gevoerd en deskresearch uitgevoerd om tot een goed en volledig verslag te komen, die kan aanbevelen wat de beste aanpak is.

Dit product wordt gemaakt door de ontwikkelaar van het project. De projectbegeleider en docentbegeleider zullen de uiteindelijke versie beoordelen ter goedkeuring. Tussentijds wordt er tussen de ontwikkelaar en de projectbegeleider de resultaten en de voortgang van het onderzoek doorgenomen.

#### 3.6.2. De Server Administration Tool

De “Server Administration Tool” is sterk afhankelijk van het onderzoek, aangezien hier uit zal blijken op welke manier de software samengesteld, geïmplementeerd, of gemaakt gaat worden.

Om te zorgen dat het product aan de gewenste eisen voldoet wordt er tweewekelijks getest met de uiteindgebruikers. Dit product wordt na de start van de implementatie wekelijks doorgesproken en beoordeeld door de projectbegeleider, om te kunnen waarborgen dat de kwaliteit goed blijft. Op het eind van het project vindt er een definitieve beoordeling plaats door de projectbegeleider.

#### 3.6.3. De Scriptie

In de scriptie wordt verslag gelegd van alle belangrijke momenten en beslissingen uit het project zowel als een uitgebreide beschrijving van het gehele project. Hierdoor is de scriptie afhankelijk van alle voorgaande producten. Dit document zal gedurende de loop van het hele project uitgewerkt en zo nodig bijgewerkt worden.

Om de kwaliteit van de scriptie te kunnen waarborgen zal er voor het einde van de uiteindelijk oplevering deze tweemaal naar de docentbegeleider op gestuurd worden om eventuele correctie punten aan te wijzen. Voor 10-06-2014 wordt de uiteindelijke versie van de scriptie opgestuurd voor een definitieve beoordeling.

### 3.7. Projectbudget

Voor dit project zijn 85 werkdagen ( 8 uur per dag ) te besteden om het te maken. Mocht het blijken dat er eventuele investeringen nodig zijn voor de “Server Administration Tool” of software aangeschaft dient te worden ter ontwikkeling, dan kan dit op dat moment overlegd worden met NetMatch.

### 3.8. Wijzigingsbudget

Voor dit project is geen wijzigingsbudget in valuta aanwezig. Mocht het zo zijn dat het project onvoldoende wordt bevonden door de projectbegeleider dan is er de mogelijkheid om uit te lopen op de originele 85 werkdagen met 10 extra werkdagen.

### 3.9. Benodigde resources

Voor dit project is een werkplek benodigd waar zowel documentatie op geschreven kan worden als software op ontwikkeld kan worden.

### 3.10. Uitsluitingen

Dit project beperkt zich tot het onderzoeken naar de best mogelijke “Server Administration Tool” voor NetMatch en het te maken/implementeren “Server Administration Tool”.

Er zijn geen andere partijen betrokken die invloed op het project kunnen hebben buiten de gestelde project scope.

### 3.11. Afhankelijkheden

Er moet een server aanwezig zijn die het gemaakte project kan hosten binnen het bedrijf of op het internet, zonder dit is er geen mogelijkheid dat de applicatie door alle medewerkers van NetMatch gebruikt kan worden.

Verder is er voor het ontwikkelen van de applicatie een test omgeving nodig, zodat er tijdens de ontwikkeling al getest kan worden.

### 3.12. Randvoorwaarden

Er moet medewerking zijn vanuit zowel de eindgebruikers als de projectbegeleider om tot een goed product te komen waarin de administratie van de servers en licenties bijgehouden en ingezien kan worden.

### 3.13. Aannames

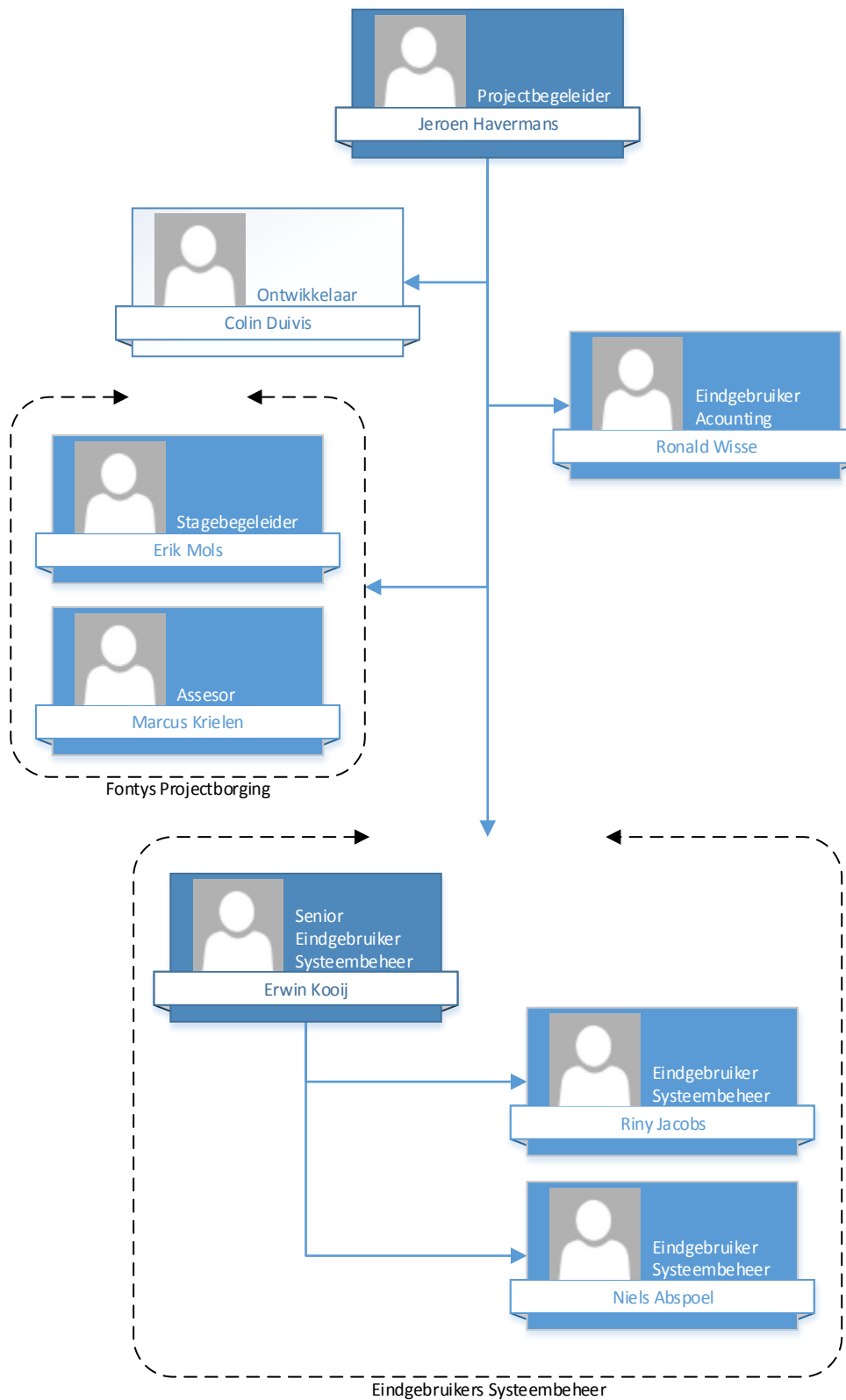
De docentbegeleider heeft tijd om de scriptie minimaal 2 maal voor de uiteindelijke opleverdatum na te kijken.

De projectbegeleider heeft wekelijks de tijd om de voortgang van het project door te nemen en feedback te geven op de gemaakte voortgang.

De eindgebruikers zijn bereid om tweewekelijks na de start van de implementatie de “Server Administration Tool” mee te testen en hierop constructieve feedback te geven.



## 4. Projectorganisatiestructuur



#### 4.1. Projectbegeleider

De projectbegeleider heeft de volgende taken binnen dit project

- Begeleiden ontwikkelaar waar nodig.
- Het tussentijds controleren van de status van het project.
- Het beoordelen van de uiteindelijke “Server Administration Tool”.
- Het beoordelen van de op te leveren documenten: PID, onderzoeksverslag en scriptie.
- Neemt het testen voor de eindgebruiker van accounting over als deze niet in staat is zelf te testen.

#### 4.2. Ontwikkelaar

Verantwoordelijk voor het uitvoeren van het gehele project.

Verantwoordelijk voor het inplannen en regelen van afspraken met alle eindgebruikers zowel voor het testen als voor het bespreken en interviewen van de eisen en huidige werkwijzen.

#### 4.3. Eindgebruiker Accounting

Verantwoordelijk voor het inzicht geven in welke eisen nodig zijn vanuit accountancy aan het project. Later in het project verantwoordelijk voor het testen van de User Interface van accountancy.

#### 4.4. Eindgebruiker Systeembeheer

Verantwoordelijk voor het inzicht geven in welke eisen nodig en gewenst zijn vanuit systeembeheer. Later in het project verantwoordelijk voor het testen van de User Interface van systeembeheer.

##### 4.4.1. Senior

De senior systeembeheerder is de eerste communicatielijn voor het verduidelijken en bespreken van eisen voor de “Server Administration Tool” vanuit systeembeheer.

#### 4.5. Docentbegeleider

De docentbegeleider is verantwoordelijk voor de volgende taken binnen het project:

- Beoordelen van de volgende documenten: PID, Onderzoeksverslag en de Scriptie
- Tussentijds corrigeren en nakijken van de volgende documenten: PID, Onderzoeksverslag en de Scriptie ( Mits de ontwikkelaar deze tijdig aanlevert voor de inleverdata van de individuele documenten ).

#### 4.6. Assessor

Verantwoordelijk voor het doornemen van de definitieve versies van de volgende documenten: PID, Onderzoeksverslag en Scriptie voordat de ontwikkelaar naar de verdediging van zijn scriptie gaat.

## 5. Projectbeheersing

### 5.1. Rapportage

#### 5.1.1. Documenten

Voordat de documenten naar school gestuurd worden ter definitieve beoordeling, zullen deze altijd eerst langs de projectbegeleider gaan, pas na accordering van de projectbegeleider wordt deze doorgestuurd naar school. Dit is van toepassing op de volgende documenten:

- PID
- Onderzoeksverslag
- Scriptie

#### 5.1.2. Afspraken

In eerste instantie is de ontwikkelaar verantwoordelijk voor het maken van de afspraken op het moment dat deze nodig worden bevonden. Als blijkt dat de ontwikkelaar om welke reden dan ook er niet in geslaagd de benodigde mensen bereid te vinden om af te spreken, schakelt de ontwikkelaar de hulp van de stagebegeleider in.

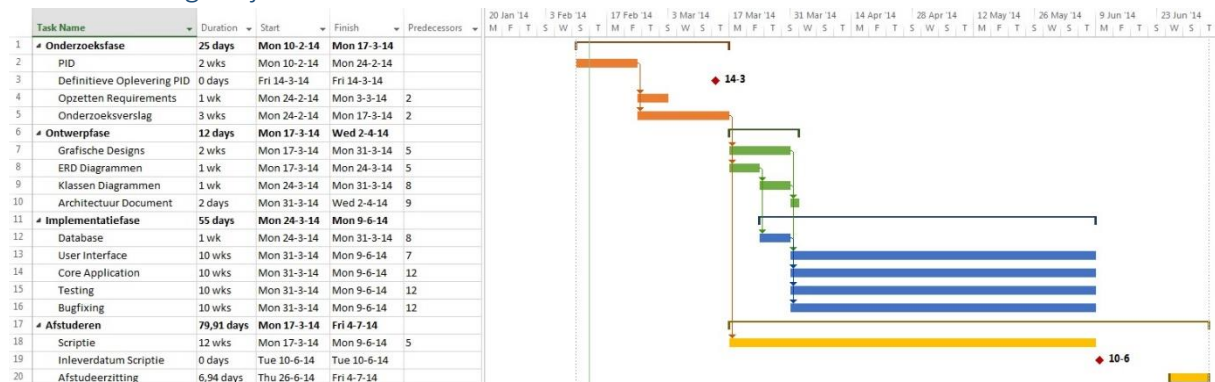
Afspraken zullen gemaakt worden via de mail, de stagebegeleider wordt altijd op de hoogte gesteld als er een officiële meeting gepland wordt.

Verder zal de docentbegeleider 1 a 2 maal langs komen op het bedrijf ter beoordeling van de voortgang van de ontwikkelaar. De eerste keer zal in de 4<sup>de</sup> week van het project gebeuren. Het tweede bezoek is optioneel.

### 5.2. Voortgangsbewaking

Er wordt wekelijks met de projectbegeleider gesproken over wat de huidige voortgang is. Hierbij zal de projectbegeleider aansporen om op de planning te blijven.

#### 5.2.1. Planning Project



### 5.3. Toleranties

#### 5.3.1. Projecttolerantie

Er is geen uitloop op het gestelde budget in de vorm van geld. En zoals vermeld in hoofdstuk 3.8. is er als uitlooptijd 10 extra werkdagen.

### 5.4. Risicomanagement

Het grootst aanwezige en enige noemenswaardige risico binnen het project is dat er te weinig tijd zou zijn om alle gewenste requirements door te voeren in de uiteindelijke applicatie, om dit risico te verkleinen is er gekozen om de Implementatiefase op een iteratieve methode uit te voeren. Dit geeft

de mogelijkheid om bij te sturen naar de gewenste functionaliteiten door de projectbegeleider en de eindgebruikers. Ook wordt er verwacht dat door deze nauwere samenwerking de implementatie gemotiveerder en dus productiever uitgevoerd zal worden.

### 5.5. Kwaliteitsbewaking

Om de kwaliteit te waarborgen word er door de ontwikkelaar getest met de eindgebruikers om te kijken of er geen bugs optreden en om te kijken of de kwaliteit is zoals de eindgebruiker verwacht.

Om de kwaliteit van de op te leveren documenten te waarborgen wordt er gestreefd deze zo spoedig mogelijk voor de deadline af te hebben, zodat eventuele fouten in de documenten er nog uitgehaald kunnen worden voordat de definitieve versie opgestuurd wordt. Verder zijn er vanuit school nog workshops ingepland die de mogelijkheid geven om je huidige documenten tussentijds alvast te kunnen laten beoordelen, deze momenten zullen ook gebruikt worden om de kwaliteit te kunnen waarborgen (mits het document niet al goedgekeurd is voor deze workshop).

### 5.6. Wijzigingsprocedure

In de implementatiefase wordt er gewerkt volgens een iteratieve methode, waarbij is vastgesteld dat er tweewekelijks een test is met de eindgebruikers en wekelijks met de projectbegeleider, dit zorgt ervoor dat op deze momenten nieuwe wijzigingen voorgesteld kunnen worden. Deze wijzigingen worden vervolgens opgenomen in de nieuwe taken en zullen dan uitgevoerd worden.

# Bijlage II: Onderzoeksverslag

Server Administration Tool

*Versie 1.0*



*net*match

## Document Informatie

Datum voltooid:

3-4-2014

Status:

Afgerond

## Auteur

Geschreven door:

Colin Duivis

Studentnummer:

2192461

## Bedrijf

Naam:

NetMatch B.V.

Plaats:

Tilburg

## Document Historie

Datum	Versie	Auteur	Omschrijving
<b>2014-02-10</b>	0.1	Colin Duivis	Opzet van het document
<b>2014-02-20</b>	0.2	Colin Duivis	Verbeteringen na (Fontys)workshop onderzoeksverslag
<b>2014-02-21</b>	0.3	Colin Duivis	Verbetering na commentaar van Jeroen Havermans
<b>2014-03-06</b>	0.4	Colin Duivis	Opnemen van resultaten na aanleiding van de genomen interviews. Omschrijven onderzoeksmethode volgens universitaire regels. Schrijven Conclusie en Aanbeveling.
<b>2014-04-01</b>	1.0	Colin Duivis	Verbeteringen na commentaar van Jeroen Havermans

## Goedkeuring Versie 1.0

*Projectbegeleider*

Naam:

Datum:

Paraaf:

*Docentbegeleider*

Naam:

Datum:

Paraaf:

## Inhoudsopgave

Document Historie .....	1
Begrippenlijst.....	4
Probleemstelling.....	5
Huidige Situatie .....	5
1. Doelstelling van het onderzoek.....	6
2. Hoofdvragen en deelvragen .....	6
2.1. Hoofdvraag .....	6
2.2. Deelvragen.....	6
3. Methode gegevensverzameling .....	7
3.1. Deelvraag: Processen, Tools en gewenste Situatie .....	7
3.2. Deelvraag: Eisen .....	8
3.3. Deelvraag: Criteria.....	8
3.3.1. Tijdsriteria over verschillende Oplossingen.....	8
3.4. Deelvraag: Oplossingen.....	8
3.4.1. Bestaande oplossingen.....	9
3.4.2. Huidige tools.....	10
4. Onderzoeksplanning.....	11
5. Resultaten.....	12
5.1. Processen, Tools en gewenste situatie.....	12
5.1.1. Systeembeheer.....	12
5.1.2. Accounting.....	13
5.1.3. Management .....	13
5.2. Eisen .....	14
5.3. Criteria.....	14
5.4. Tijdsriteria over verschillende Oplossingen.....	14
5.5. Bestaande Oplossingen .....	15
5.6. Huidige Tools .....	15
5.6.1. Welke data is te gebruiken binnen de tools? .....	16
5.6.2. Hoe is deze data te importeren / uittewisselen? .....	17
5.6.3. Wat is de hoeveelheid data en hoe vaak moet die ingevoerd worden?.....	17
5.6.4. Wenselijk om de applicatie om te bouwen naar “Server Administration Tool”? .....	18
5.7. Oplossingen .....	18
6. Conclusie .....	19
7. Aanbevelingen.....	19
Bronnen .....	20

<b>Bijlage I:</b>	<b>Requirement Gesprekken</b>
<b>Bijlage II:</b>	<b>Criteria Gesprekken</b>
<b>Bijlage III:</b>	<b>Requirements</b>
<b>Bijlage IV:</b>	<b>Bestaande Oplossingen</b>
<b>Bijlage V:</b>	<b>Huidige Tools</b>
<b>Bijlage VI:</b>	<b>Oplossingen</b>



## Begrippenlijst

<b>.NET</b>	Is een software framework geschreven door Microsoft. Dit software framework bevat veel library's in verschillende programmeertalen die door al deze programmeertalen gebruikt kunnen worden.
<b>Application Programming Interface (API)</b>	een interface die je in een programmeertaal kan gebruiken, waarbij je in een vaste structuur informatie kan benaderen, versturen en/of acties kan uitvoeren.
<b>C#</b>	Een programmeertaal die ontwikkeld is door Microsoft.
<b>Closedsource</b>	Een softwarepakket waarvan de broncode niet benaderbaar is.
<b>Command Line Interface (CLI)</b>	Een interface waarbij je puur op basis van het intypen van commando's (regels tekst) acties kan uitvoeren in het systeem en tekstueel respons kan krijgen van het systeem.
<b>Graphical User Interface (GUI)</b>	De grafische schil van een applicatie waar de gebruiker de betreffende applicatie mee kan bedienen.
<b>MoSCoW Methode</b>	Een methode om de eisen en de prioriteiten van deze eisen vast te kunnen leggen.
<b>Opensource</b>	Een opensource softwarepakket is een pakket waar je bij de broncode van de software kan. Dit geeft de programmeur de mogelijkheid zelf aanpassingen te maken en het systeem uit te breiden.
<b>Reseller</b>	Iemand die producten of diensten van een hoofdleverancier doorverkoopt.
<b>Service Level Agreement (SLA)</b>	Een afspraak tussen een opdrachtnemer en de klant waarin vast staat gelegd aan welke afspraken beide partijen zich te houden hebben.
<b>Services Provider License Agreement (SPLA)</b>	Een licentie model van Microsoft die gebruikt wordt voor het maandelijks aanschaffen van licenties, dit gebeurt via een gecertificeerde reseller van Microsoft.
<b>Software framework</b>	Een verzameling van verschillende programmeertalen.
<b>Software library</b>	Een collectie van code die gebruikt kan worden binnen een software applicatie.
<b>Software Licentie</b>	Een software licentie is het bewijs dat je eigenaar bent van een of meerdere softwarepakketten. En geeft je het recht deze te gebruiken binnen de voorwaarden gekoppeld aan deze licentie.
<b>User Interface (UI)</b>	Een term die gebruikt wordt voor het aanduiden van de interface waarmee gebruikers de applicatie kunnen bedienen. Van Command Line Interface (CLI) tot Graphical User Interface (GUI).

## Probleemstelling

NetMatch levert hosting services op Windows & Linux platformen aan klanten. NetMatch heeft meerdere serverracks, verspreid over meerdere Nederlandse datacenters: webserver, SQL-server, applicatie server, load balancers, caching server, etc.

Bij het beheren van het platform van NetMatch komen ook veel randzaken kijken, denk hierbij aan het administreren en rapporteren van software licenties, bijhouden van lopende support contracten, een change log bijhouden, garanties bijhouden en het bijhouden van Service Level Agreements (SLA) ([Service level agreement, 2013](#)).

Deze administratie is op dit moment verdeeld over meerdere tools en Excel administraties. Buiten deze tools zijn er nu ook nog specifieke gegevens die alleen aanwezig zijn in de hoofden van de gebruikers en dus niet/slecht toegankelijk zijn voor anderen.

## Huidige Situatie

Momenteel is de server administratie en de licenties behorend bij de servers verdeeld over 2 plekken:

1. De administratie van de klanten, kosten en de licenties zitten in 1 groot Excel bestand opgenomen. Waarbij handmatig maandelijks de prijzen up-to-date gehouden moeten worden.
2. De administratie van de locatie van de servers en de indeling van de server kasten zit in een programma opgenomen die RackTables ([Andriyanov, Dummer, & Ovsienko, 2013](#)) heet.

Maandelijks wordt er vanuit SEQUINT een prijslijst doorgegeven van nieuwe licenties en de nieuwe licentie prijzen op jaarlijkse basis komen er updates in de prijzen, er vervallen er oude licenties en komen er weer nieuwe bij. Eén keer in de drie jaar wordt het contract met SEQUINT vernieuwd. De gegevens uit de maandelijkse prijslijst moeten vervolgens doorgevoerd worden naar de eerder genoemde Excel sheet om zo voor alle klanten de prijzen weer up-to-date te hebben.

Deze manier van documenteren wordt op dit moment omslachtig bevonden en buiten dat is de huidige manier erg gevoelig voor fouten. Zo zijn er eind 2013 drie maanden lang de verkeerde aantallen gerapporteerd. Ook is er bij een audit van Microsoft ondervonden dat het ophalen van alle gegevens over de servers waarop Microsoft licenties draaien een hele opgave is. Bij dit ophalen kwamen er ook enkele fouten binnen de administratie naar boven.

## 1. Doelstelling van het onderzoek

De doelstelling van dit onderzoek is het realiseren van een methode die voldoet aan de eisen van NetMatch voor een "Server Administration Tool".

## 2. Hoofdvragen en deelvragen

### 2.1. Hoofdvraag

Welke oplossing voldoet het best aan de eisen en criteria van NetMatch voor één "Server Administration Tool"?

### 2.2. Deelvragen

1. Welke processen en tools worden momenteel gebruikt en wat is de gewenste situatie?(Processen, Tools en gewenste Situatie)
2. Wat zijn de eisen van NetMatch aan de "Server Administration Tool"? (Eisen)
3. Wat zijn voor NetMatch de criteria waarmee de verschillende oplossingen vergeleken moeten worden?(Criteria)
  - a. Hoe wordt een tijdsriteria verrekend over de verschillende oplossingen? (Tijdsriteria over verschillende oplossingen)
4. Welke oplossingen zijn er die aan de eisen en criteria voldoen?(Oplossingen)
  - a. Welke bestaande oplossingen zijn er?(Bestaande oplossingen)
  - b. Op welke manier kunnen de huidige tools ingezet worden? (Huidige tools)

### 3. Methode gegevensverzameling

#### 3.1. Deelvraag: Processen, Tools en gewenste Situatie

Welke processen en tools worden momenteel gebruikt en wat is de gewenste situatie?

Er is met de verschillende groepen(systeembeheer, accounting en management) binnen dit project een brainstormsessie / diepte interview gehouden.

- **Systeembeheer** is vertegenwoordigd door: Erwin Kooij, Riny Jacobs en Niels Abspoel.
- **Accounting** is vertegenwoordigd door: Ronald Wisse.
- **Management** is vertegenwoordigd door: Jeroen Havermans.

De volgende vragen vormde een leidraad voor de gesprekken:

Systeembeheer:

1. Welke processen doorlopen jullie nu / Wat is jullie werkwijze?
2. Welke tools gebruiken jullie voor het vastleggen van de huidige server administratie?
3. Is er informatie die jullie op dit moment missen?
4. Wat is voor jullie het belangrijkste om vast te leggen binnen de "Server Administration Tool"?
5. Welke informatie leggen jullie met de meeste regelmaat vast?
6. Welke informatie moeten jullie vaak opvragen?
7. Met welke partijen hebben jullie in de context van server administratie te maken?
8. Hoe zouden jullie willen dat de werkwijze zou zijn?
9. Hebben jullie nog extra eisen of ideeën?

Accounting:

1. Welke processen doorloop je nu / Wat is je werkwijze?
2. Welke tools gebruik je momenteel om de informatie van de server administratie in te zien of te verwerken?
3. Is er informatie die je op dit moment mist?
4. Welke gegevens moet je vast kunnen leggen in het systeem?
5. Welke informatie is belangrijk om in te kunnen zien vanuit accounting?
6. Met welke partijen heb je te maken in de context van server administratie?
7. Hoe zou je willen dat de werkwijze zou zijn?
8. Heb je nog extra eisen of ideeën voor de applicatie?

Management:

1. Welke processen doorloop je binnen het project?
2. Heb je zelf nog taken binnen de applicatie?
3. Welke personen / groepen moeten toegang krijgen tot de applicatie?
4. Is er informatie / overzichten die je momenteel mist die je graag zou willen zien in de applicatie?
5. Welke taal moet de user interface hebben?
6. Maakt het uit onder welke browsers de applicatie ondersteund wordt?
7. Zijn er eisen op het gebied van performance?
8. Heb je nog suggesties voor de huidige werkwijze van accounting of systeembeheer?
9. Heb je nog extra eisen of ideeën voor de applicatie?
10. Wat zijn de kern functionaliteiten?

Het doel van deze vragen is om de huidige situatie en tools, plus de gewenste situatie vast te leggen en op deze manier een beter inzicht te krijgen in het project en de eisen die aan de uiteindelijke applicatie gesteld moeten worden.

Alle uitkomsten van deze gesprekken zijn vastgelegd in bijlage I ( Requirement Gesprekken ).

### 3.2. Deelvraag: Eisen

Wat zijn de eisen van NetMatch aan de “Server Administration Tool”?

De eisen zijn opgesteld in overeenstemming met de drie verschillende partijen ( Accounting, Systeembeheer en Management ). Deze eisen vloeien voort uit de gesprekken uit de “Processen, Tools en gewenste Situatie” deelvraag. De overeenstemming is bereikt door feedback en goedkeuring te vragen aan alle drie de partijen. De prioriteitsbepaling is in het document gedaan volgens de MoSCoW methode ([International Institute of Business Analysis, 2009](#)).

De definitieve vorm van de eisen zijn opgenomen in bijlage III ( Requirements ).

### 3.3. Deelvraag: Criteria

Wat zijn NetMatch zijn criteria waarmee de verschillende aanpakken vergeleken moeten worden?

Deze informatie is opgehaald door een diepte interview met management te houden over wat de criteria zijn waarop verschillende oplossingen getoetst zouden mogen worden. De volgende vragen zijn als leidraad gebruikt om tot antwoorden te komen:

1. In geval van verschillende oplossingen die mogelijk zijn op welke criteria moet deze dan met elkaar vergeleken worden?
2. Aan hoeveel procent van de eisen moet de gekozen oplossing voldoen?
  - a. Welke eisen zijn hierbij het meest van belang?

Alle resultaten van de gesprekken zijn opgenomen in bijlage II ( Criteria Gesprekken ).

#### 3.3.1. Tijdsriteria over verschillende Oplossingen

Hoe wordt een tijdsriteria verrekend over de verschillende oplossingen?

Het was nog onduidelijk hoe de verschillende criteria tegenover elkaar verrekend moesten worden. Om achter deze informatie te komen is er gekozen om een diepte interview te houden met een expert op het gebied van het opzetten van software projecten. Het doel is dat er tijdsindicaties gemaakt kunnen worden volgens een vaste methode, zodat de oplossingen eerlijk tegenover elkaar vergeleken kunnen worden.

Als expert is Leo Hexspoor ( Software Architect bij NetMatch ) geïnterviewd.

De volgende vragen zijn als leidraad opgesteld:

1. Hoe meet je hoelang het kost om een applicatie te maken?
2. Hoeveel tijd kost het om een bestaand softwarepakket te leren kennen die je zelf nog moet gaan uitbreiden?
3. Hoeveel tijd kost het om communicatie te realiseren met een nog onbekende applicatie?

Ook deze uitkomsten zijn opgenomen in bijlage II.

### 3.4. Deelvraag: Oplossingen

Welke oplossingen zijn er die aan de eisen en criteria voldoen?

Om deze vraag te beantwoorden worden eerst de 2 sub deelvragen beantwoord. De applicaties uit deze deelvragen zullen hier verder geanalyseerd worden. De top 3 applicaties uit de “Bestaande Oplossingen” deelvraag zijn opgenomen. En de relevante applicaties uit de “Huidige Tools” deelvraag zijn opgenomen.

Er is eerst een takenlijst gemaakt op basis van de requirements uit bijlage III, hierin staan de taken die binnen het systeem aanwezig moeten zijn. Voor elk van deze taken is genoteerd wat de verwachte complexiteit is om deze taak te maken. Hieruit kan opgemaakt worden hoe complex zelf bouwen van de applicatie is.

Complexiteit is gemeten van: 1 Zeer Simpel, 2 Simpel, 3 Niet Simpel, 4 Complex, 5 Zeer Complex.

De verschillende oplossingen zijn gemeten tegenover de eerder vernoemde takenlijst. Elk van de taken zit nog gekoppeld aan de Must Have (MH), Should Have (SH), Could Have (CH) en Would Have (WH) categorie van hun oorspronkelijke eis, zodat er per applicatie gekeken kan worden hoe ze scoren in elk van deze prioriteitscategorieën.

De oplossingen zijn gemeten op de volgende criteria(gebaseerd op bijlage II):

1. Complexiteit om aan te passen van 1 tot 5 van zeer simpel tot zeer complex.
  - a. Documentatie over de code aanwezig 0 tot 2 van afwezig tot in overvloed.
  - b. Community over de code aanwezig 0 tot 2 van afwezig tot in overvloed.
  - c. Is het opensource Ja of Nee.
  - d. Is het modulair Ja of Nee.

*Formule(Nee = 0, Ja = 1):  $5 - d - c - (b + a) / 2$*

2. Bevat het alle kerntaken?
3. Jaarlijkse kosten onder € 5000,- (De jaarlijkse kosten in euro's).

### 3.4.1. Bestaande oplossingen

Welke bestaande oplossingen zijn er?

De volgende 4 zoekopdrachten worden ingevoerd in de google search engine ( [www.google.nl](http://www.google.nl) ):

- server administration software
- license administration software
- datacenter administration software
- administration software

Bij elk van deze 4 zoekopdrachten worden de eerste 10 links die google terug geeft genoteerd. Bij elk van de links zal er gekeken worden of er relevante software op de pagina's staat met betrekking tot de “Server Administration Tool”.

Er is twee keer gekeken of de applicatie relevant is. De eerste keer is er gekeken binnen de Google resultaatpagina of deze voldoet aan de criteria van relevante software. De tweede keer is de oorspronkelijke website van het softwarepakket opgezocht en wordt hier nogmaals gekeken of het relevante software is.

Bepaling relevante software:

De software moet voldoen aan de volgende criteria:

1. Het software pakket is instaat 1 van de volgende acties uit te voeren:
  - a. Gebruiker moet meerdere servers aan het systeem kunnen toevoegen.

- b. Gebruiker moet meerdere licenties aan het systeem kunnen toevoegen.
2. Het is een software applicatie.
3. De applicatie moet webbased zijn.

De software mag niet voldoen aan de volgende criteria:

4. In de software zijn de servers die bijgehouden kunnen worden niet platform onafhankelijk( de software houdt dus bijvoorbeeld alleen maar Windows servers bij).
5. In de software wordt maar ondersteuning geboden voor 1 type server, bijvoorbeeld alleen SQL servers of exchange servers.
6. In de software kan alleen van specifieke software of software fabrikanten data bijgehouden worden. bijvoorbeeld alleen van Microsoft en/of Adobe.

De overgebleven applicaties zijn gemeten tegenover de must have requirements van bijlage III. Het meten of de applicaties deze requirements bevatten is gedaan door het analyseren van screenshots, video's of demo's van het programma. Tekstuele beweringen of presentaties die geen beelden van de applicatie bevatten zijn hierin niet meegenomen.

In bijlage IV ( Bestaande Oplossingen ) zijn alle zoekresultaten en vergelijkingen genoteerd.

### 3.4.2. Huidige tools

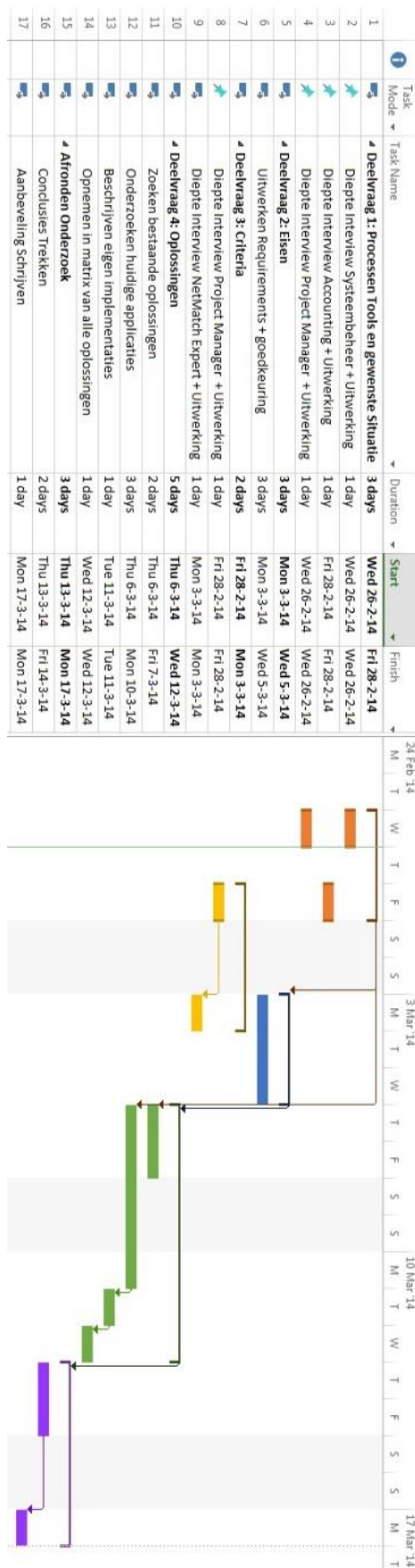
Op welke manier kunnen de huidige tools ingezet worden?

De applicaties / documenten die in de "Processen, Tools en gewenste Situatie" deelvraag naar voren zijn gekomen zijn verder uitgewerkt in bijlage V ( Huidige Tools ). Opmerkingen over de tools en documenten die de tools verder uitlichten zijn ook doorgenomen en genoteerd in bijlage V.

Voor de applicaties is er gekeken of er mogelijkheden zijn voor het uitbreiding, data gebruik of communicatie van en naar de applicatie. Hiervoor zijn de volgende vragen beantwoord:

1. Welke data is te gebruiken binnen de tools? ( Gekeken naar alle tools uit bijlage V )
2. Hoe is deze data te importeren / uit te wisselen? ( Gekeken naar alle overgebleven tools )
3. Hoeveel data is er en met welke regelmaat moet de data gebruikt worden? ( Gekeken naar alle overgebleven tools )
4. Wenselijk om de applicatie uit te breiden / om te bouwen naar "Server Administration Tool"? ( Gekeken naar alle overgebleven applicaties )

## 4. Onderzoeksplanning





## 5. Resultaten

### 5.1. Processen, Tools en gewenste situatie

Welke processen en tools worden momenteel gebruikt en wat is de gewenste situatie?

Voor elk van de 3 partijen is hieronder hun processen, tools en gewenste situatie beschreven. Deze partijen zijn hieronder nogmaals vernoemt:

- **Systeembeheer** vertegenwoordigd door: Erwin Kooij, Riny Jacobs en Niels Abspoel.
- **Accounting** vertegenwoordigd door: Ronald Wisse.
- **Management** vertegenwoordigd door: Jeroen Havermans.

Hieronder staat omschrijvingen van de processen, tools en gewenste situatie per partij:

#### 5.1.1. Systeembeheer

##### Processen

In het gesprek blijkt dat systeembeheer momenteel de volgende processen doorloopt bij het beheren van de server administratie:

- Het maandelijks updaten van de Excel server administratie met de nieuwe SPLA ([Microsoft Corporation, 2012](#)) licenties.
- Het maandelijks terug rapporteren naar de SPLA leverancier van de af te nemen licenties.
- Bij het aanschaffen van nieuwe servers deze in de Excel server administratie toe voegen en in RackTables toevoegen.

PARAFRASERING VAN BIJLAGE I -> BRAINSTROM SYSTEEMBEHEER -> HUIDIGE SITUATIE.

##### Tools

In het gesprek met systeembeheer komen er een aantal tools naar voren:

- **Excel server administratie:** De huidige tool voor het bijhouden van alle licentie en server administratie.
- **License admin:** Een applicatie die gemaakt is voor het vervangen van de excelsheet maar niet goed werkt ( wordt niet gebruikt ).
- **Racktables:** Hierin worden de volgende gegevens opgeslagen: de locatie gegevens van de datacenters, de serverracks binnen de datacenters en de hardware en servers binnen de serverracks.
- **Hosting admin:** Een applicatie die gebruikt wordt voor het bijhouden van de hosting gegevens van NetMatch. Wordt niet gebruikt voor de huidige server administratie.

PARAFRASERING VAN BIJLAGE I -> BRAINSTROM SYSTEEMBEHEER -> HUIDIGE APPLICATIES.

##### Gewenste Situatie

Wat voornamelijk naar voren komt is dat systeembeheer zoveel mogelijk gegevens bij willen houden van de servers, hardware en licenties. Met de voornaamste reden om te kunnen zien wanneer een server of hardware element oud is, of wanneer een garantie termijn dreigt te verlopen en hoeveel licenties ze in kunnen zetten op welke servers. Verder willen ze inzicht hebben in welke servers waarvoor ingezet zijn en waar deze zich op dit moment fysiek bevinden.

SAMENVATTING VAN BIJLAGE I -> BRAINSTROM SYSTEEMBEHEER -> GEWENSTE FUNCTIONALITEITEN.

### 5.1.2. Accounting

#### Processen

Na aanleiding van het gesprek met accounting blijkt dat accounting voornamelijk verantwoordelijk is voor het opsturen van de facturen naar de klanten toe en het betalen van de facturen van de leveranciers.

Eén keer per maand per klant wordt er een factuur opgesteld en verstuurd. In deze facturen staan de werkzaamheden van systeembeheer, de ingekochte hardware, de rackspace gegevens en hun licenties. Zodra accounting geen updates krijgt over de ingekochte hardware Rackspaces en licenties worden de getallen over genomen uit de vorige maand. Voor de werkzaamheden wordt wel maandelijks gecommuniceerd tussen accounting en systeembeheer.

Verder krijgt systeembeheer de facturen binnen van de SPLA leverancier en de rackspace leveranciers. Deze betaald accounting dan ook als ze binnen komen.

PARAFRASERING VAN BIJLAGE I -> BRAINSTROM ACCOUNTING -> HUIDIGE SITUATIE.

#### Tools

Accounting gebruikt vrijwel geen tools buiten Excel om de facturen op te stellen en de mail om de gegevens op te vragen en binnen te krijgen.

PARAFRASERING VAN BIJLAGE I -> BRAINSTROM ACCOUNTING -> TOOLS.

#### Gewenste Situatie

Aan alle eisen die accounting stelt kan je concluderen dat ze vooral maandoverzichten van de ingekochte spullen (servers, hardware, Rackspaces en licenties) voor de klant zowel als voor intern kunnen zien. Verder willen ze deze data kunnen vergelijken met voorgaande maanden en met de facturen die ze van de leveranciers binnen krijgen. Ze willen graag meer controle en overzicht in wat ze zelf factureren en wat er betaald moet worden.

PARAFRASERING VAN BIJLAGE I -> BRAINSTROM ACCOUNTING -> GEWENSTE FUNCTIONALITEITEN + BELANGRIJK VOOR ACCOUNTING.

### 5.1.3. Management

#### Processen en Tools

In het gesprek met Jeroen Havermans komt naar voren dat hij verantwoordelijk is voor systeembeheer. Verder laat hij weten dat als management hij de volgende taken binnen het project heeft:

- Hardware van klanten vervangen ( bij garantie verval ).
- Bij houden van SLA contracten ( klant/project ).
- Controleren hoe de ingekochte licenties tegenover de aangewezen licenties staan.

PARAFRASERING VAN BIJLAGE I BRAINSTROM PROJECT MANAGER -> ROL BINNEN DE SERVER ADMINISTRATIE

#### Gewenste Situatie

Jeroen Havermans laat in het Brainstorm gesprek blijken dat voornamelijk het implementeren en rapporteren van de SPLA licentie lijsten heel belangrijk is en de overzicht van servers en licenties zowel historisch als huidige gegevens.

PARAFRASERING VAN BIJLAGE I BRAINSTROM PROJECTMANAGER -> KERN FUNCTIONALITEITEN

Jeroen Havermans laat blijken dat hij hoopt dat de applicatie ervoor zorgt dat er een transparantere en duidelijkere overdracht van informatie komt tussen Systeembeheer en Accounting. En verder hoopt hij dat systeembeheer goed gebruik zal maken van de “Server Administration Tool”.

PARAFRASERING VAN BIJLAGE I BRAINSTROM PROJECTMANAGER -> GEWENSTE SITUATIE

## 5.2. Eisen

Wat zijn de eisen van NetMatch aan de “Server Administration Tool”?

De definitieve eisen van NetMatch staan omschreven in bijlage III ( Requirements ).

## 5.3. Criteria

Wat zijn NetMatch zijn criteria waarmee de verschillende aanpakken vergeleken moeten worden?

Uit het gesprek met Jeroen Havermans zijn de volgende criteria naar voren gekomen:

- **Code Onderhoud:** De code moet benaderbaar en aanpasbaar zijn.
- **Hosting:** Er wordt gewenst dat de applicatie intern te hosten is.
- **Applicatie Uitbreiding:** Het wordt geprefereerd dat de applicatie in .NET geschreven is, zodat NetMatch hem zelf kan onderhouden en uitbreiden.
- **Haalbaar:** De volgende kern functionaliteiten moet de applicatie binnen de duur van de stage periode kunnen bevatten:
  - Inladen van SPLA lijst.
  - Overzicht tonen van alle SPLA licenties.
  - Moet een maandelijkse uitdraai mogelijk zijn van SPLA licenties.
  - Historie bij houden minimaal per maand van veranderingen in licenties.
  - Moet servers hun hoofdtaak en de gekoppelde licenties bij kunnen houden.
  - Overzicht mogelijkheid van de licenties die klanten hebben per maand inzichtelijk.
- **Budget:** De gekozen oplossing mag niet meer dan € 5000,- per jaar kosten.

PARAFRASERING VAN BIJLAGE II -> DIEPTE INTERVIEW PROJECT MANAGER

## 5.4. Tijdsriteria over verschillende Oplossingen

Hoe wordt een tijdsriteria verrekend over de verschillende oplossingen?

Uit het gesprek met Leo Hexspoor blijkt dat tijd niet gemeten moet worden, aangezien dit veel te onnauwkeurig is. Wel kan er over verschillende ranges van de programma's vastgelegd of er een eigen of bestaande oplossing gekozen moet worden.

De volgende punten en methoden zijn gegeven voor het meten van de complexiteit van de software:

### *Complexiteit berekenen project*

- Maak een work breakdown structure ( WBS ) op basis van je requirements.
  1. Stel vast hoe complex het is om de WBS items te maken.
  2. Noteer voor elk van deze elementen je zekerheidsniveau.
  3. Maak bij de onzekerste eisen een Proof of Concept ( PoC ).

*Complexiteit berekenen bestaande oplossingen:*

- Hoeveel gewenste functionaliteiten bevat het programma?
- Open-source of closed-source?
  - Open-source:
    - Is de taal binnen je expertise gebied?
    - Hoe goed is het programma gedocumenteerd?
    - Hoe groot is de community die je ondersteuning biedt?
    - Is de applicatie modulair opgebouwd?
  - Closed-source & commercieel:
    - Nodig het bedrijf uit en kijk wat de mogelijkheden zijn door middel van kritische vragen te stellen.

*Complexiteit berekenen eigenbouw:*

- Valt het binnen het expertise gebied qua soort software / is er veel kennis aanwezig voor het maken van deze software?

PARAFRASERING VAN BIJLAGE II -> DIEPTE INTERVIEW NETMATCH EXPERT

## 5.5. Bestaande Oplossingen

Welke bestaande oplossingen zijn er?

De volledige resultaten van de zoekopdracht en analyse van de verschillende bestaande applicaties staat gedocumenteerd in bijlage IV ( Bestaande Oplossingen ).

*Aantal gevonden oplossingen vanuit de zoekresultaten:*

41 ( voor een lijst met alle applicaties zie bijlage IV -> Lijst mogelijke matches ).

*Overgebleven applicaties na tweede filtering:*

17 ( voor een lijst met alle overgebleven applicaties zie bijlage IV -> Requirement voldoening ).

*De top 5:*

<i>Software Naam</i>	<i>Must-Haves %</i>
<a href="#">Desktop Central 8</a>	44
<a href="#">Spiceworks</a>	41
<a href="#">OpManager</a>	39
<a href="#">WhatsUp Gold</a>	39
<a href="#">Certo AssetStudio</a>	39

Tabel 1 Parafrasering van bijlage IV -> Requirement voldoening

## 5.6. Huidige Tools

Op welke manier kunnen de huidige tools ingezet worden?

In bijlage V ( Huidige Tools ) staan alle applicaties en documenten omschreven van NetMatch die iets te maken hebben met Server en Licentie administratie. Hieronder worden in het kort alle applicaties en documenten genoemd:

*Applicaties*

- RackTables
- Hosting Admin

- License Admin

#### *Documenten*

- Exceladministratie
- Facturen leveranciers rackspaces
- Facturen leveranciers SPLA licenties
- SPLA prijslijst
- Excel Facturen voor de klanten

#### 5.6.1. Welke data is te gebruiken binnen de tools?

##### *RackTables*

- Servers en Hardware:
  - Naam
  - Type
  - Model
  - Positie binnen server rack
- Server racks:
  - Naam
  - Rij positie binnen datacenter
  - Aantal racks
- Datacenters:
  - Naam
  - Locatie

##### *Hosting Admin*

- Servers:
  - Naam
  - Doel
  - Of de server virtueel is

##### *Excel Administratie*

- Servers:
  - Ordernummer
  - Leverdatum
  - Aanschafprijs
  - Service tag
  - Model
  - Naam
  - Garantie afloop datum

##### *SPLA prijslijst*

- SPLAs
  - Unieke code
  - Naam
  - Aantal waarin de licentie wordt verkocht
  - Prijs
  - Aantal aangeschaft

*Facturen leveranciers SPLA en Rackspaces + Excel facturen voor de klanten.*

Deze documenten bevatten geen nieuwe informatie die binnen de scope van het project opgeslagen behoort te worden.

Facturen van leveranciers worden hooguit vergeleken met de informatie binnen de “Server Administration Tool”. De facturen van klanten worden op basis van de interne informatie opgesteld, dit zal echter niet door de applicatie gebeuren ( zie bijlage I & III ).

*License Admin*

Systeembeheer beweert dat de informatie van License Admin hetzelfde is als in de Excel administratie (Kooij, 2014), maar een rapport van Jeroen Havermans legt vast dat er fouten in de gegevens en applicatie zitten van License Admin zitten (Havermans, 2013). Om niet het risico te lopen dat er corrupte data wordt geïmporteerd, is deze applicatie buiten beschouwing gelaten.

*5.6.2. Hoe is deze data te importeren / uittewisselen?**RackTables*

Er is toegang tot de database. De data kan geëxporteerd en geïmporteerd worden of er kan een directe communicatie op worden gezet met de database. (Kooij, 2014)

*Hosting Admin*

Er is toegang tot de database. De data kan geëxporteerd en geïmporteerd worden of er kan een directe communicatie op worden gezet met de database. (Kooij, 2014)

*Excel Administratie*

De Excelsheet kan zelf geïmporteerd en uitgelezen worden om bij de data te kunnen.

*Maandelijkse SPLA prijslijst*

De Excelsheet kan zelf geïmporteerd en uitgelezen worden om bij de data te kunnen. Er moet dan nog wel aangegeven worden voor welke maand deze prijslijst is.

*5.6.3. Wat is de hoeveelheid data en hoe vaak moet die ingevoerd worden?**Hosting Admin*

Eenmalig in te voeren.

- 238 Servers ( Niet allemaal gewenst binnen de “Server Administration Tool” (Kooij, 2014) ).

*RackTables*

Eenmalig in te voeren.

- 4 Datacenter locaties.
- 7 Server racks.
- 61 Servers.
- 90 Hardware elementen (die in een server rack geplaatst kunnen worden oa. UPC's, switches, routers, modems, patchpanels etc.).

*Excel Administratie*

Eenmalig in te voeren.

- 130 Servers.

*SPLA prijslijst*

Maandelijks in te voeren.

- +/- 97 SPLA Licenties.

#### 5.6.4. Wenselijk om de applicatie om te bouwen naar “Server Administration Tool”?

##### Hosting Admin

Deze applicatie wordt gebruikt voor het bijhouden en onderhouden van alle websites onder NetMatch zijn beheer, het is niet wenselijk dat deze applicatie ook nog eens licenties en server locaties bij gaat houden. (Kooij, 2014)

##### RackTables

Deze applicatie zou eventueel kunnen gebruikt worden, maar bevat veel functionaliteiten die we niet willen en weinig van de functionaliteiten die we wel willen (Kooij, 2014)

### 5.7. Oplossingen

Welke oplossingen zijn er die aan de eisen en criteria voldoen?

Uit de vergelijkingen die gemaakt zijn in bijlage VI ( Oplossingen ) blijkt dat SpiceWorks het hoogste scoort op basis van Totaal aantal taken gebaseerd op eisen zie onderstaand diagram. Voor de complete takenlijst zie bijlage VI



Diagram 1 Toont per applicatie aan hoeveel taken er voldaan wordt per prioriteitsniveau. (Totaal aantal taken 52 )

Hieronder is een tabel opgenomen uit bijlage VI die de belangrijkste waarden laat zien van de vergelijkingen tussen de verschillende oplossingen ( voor de overige data zie bijlage VI ).

Oplossingen	Taken Totaal %	Taken MH %	OpenSource	Jaarlijkse kosten	Complexiteit om aan te passen(0-5)
Desktop Central 8	36,5	59,5	Nee	€ 9148,13	5
SpiceWorks	40,4	59,5	Nee	€ 359,79	2,5
OpManager	26	38,1	Nee	€ 14387,27	5
RackTables	25	28,6	Ja	€ 0	3

Tabel 2 Relevantste data gefilterd uit bijlage VI

## 6. Conclusie

Welke oplossing voldoet het best aan de eisen en criteria van NetMatch voor één “Server Administration Tool”?

Uit het hoofdstuk 5.7. is al gebleken dat er geen oplossing is die aan alle gestelde eisen en/of criteria voldoet.

Als er gefilterd wordt op welke applicaties er onder het budget blijven van € 5000,- per jaar, dan blijven alleen SpiceWorks en Racktables over. Waarbij Spiceworks de meeste taken bevat.

Het probleem is echter dat de beste applicatie ( SpiceWorks ) die naar voren komt geen opensource applicatie is en maar 40,4% van alle taken bevat en 59,5% van alle Must Have taken bevat. Om deze applicatie zover te krijgen dat het de uiteindelijke “Server Administration Tool” is die NetMatch wilt zal er heel wat aangepast moeten worden en dan is de vraag hoeveel aan te passen is met het modulaire plug-in systeem van SpiceWorks.

De enige opensource applicatie ( RackTables ) bevat maar liefst 25% van het totaal aantal taken en maar 28,6% van de Must Have taken. Deze applicatie zal dus aanzienlijk aangepast moeten worden om aan alle eisen voor de “Server Administration Tool” te voldoen.

De conclusie is dan ook dat geen van gevonden en geanalyseerde applicaties goed genoeg voldoen aan de eisen en criteria om deze ook daadwerkelijk te gebruiken als basis voor de te maken: “Server Administration Tool”.

## 7. Aanbevelingen

Er wordt aanbevolen dat geen van de gevonden applicaties gebruikt worden als basis van de “Server Administration Tool”. Dit aangezien er niet voldoende voldaan wordt aan de taken en door de twijfel over de aanpasbaarheid van de applicaties.

Er wordt dan ook aanbevolen dat er een eigen applicatie ontwikkeld wordt.

Verder wordt er aanbevolen voor het implementeren van servers de data van de Server Excel Administratie te gebruiken. Voor het maandelijkse SPLA prijzen toevoegen moet in overleg met de SPLA leverancier gekeken worden welke mogelijkheden er nog meer zijn voor het aan leveren van de prijslijst informatie.



## Bronnen

Andriyanov, A., Dummer, A., & Ovsienko, D. (2013). *About*. Opgehaald van RackTables:  
<http://racktables.org/about.php>

Fayad, M., & Schmidt, D. C. (1997, Oktober). *Object-Oriented Application Frameworks*. Opgehaald van Distributed Object Computing (DOC) Group for Distributed Real-time and Embedded (DRE) Systems: <http://www.dre.vanderbilt.edu/~schmidt/CACM-frameworks.html>

Havermans, J. (2013). *License Admin improvements*. Tilburg.

International Institute of Business Analysis. (2009). *A Guide to the Business Analysis Body of Knowledge*.

Kooij, E. (2014, 03 12). Onderzoek interne beschikbare applicatie data. (C. Duivis, Interviewer)

Microsoft Corporation. (2012). *SPLA-licentiemodel*. Opgehaald van Microsoft:  
<https://partner.microsoft.com/belux-nl/licensing/licensingprograms/40059322>

*Service level agreement*. (2013, 12 8). Opgehaald van Wikipedia:  
[https://nl.wikipedia.org/wiki/Service\\_level\\_agreement](https://nl.wikipedia.org/wiki/Service_level_agreement)

# Bijlage III: Requirements

Server Administration Tool

*Versie 1.0*



*net*match

## Document Informatie

Datum voltooid:

11-3-2014

Status:

Definitief

Geschreven door:

Auteur

Colin Duivis

Naam:

Bedrijf

NetMatch B.V.

Plaats:

Tilburg

## Document Historie

Datum	Versie	Auteur	Omschrijving
<b>2014-03-03</b>	0.1	Colin Duivis	Opzet van het document
<b>2014-03-07</b>	0.2	Jeroen Havermans	Review
<b>2014-03-10</b>	0.3	Colin Duivis	Verwerking review + nieuwe eisen
<b>2014-03-11</b>	1.0	Colin Duivis	Definitieve versie

## Goedkeuring Versie 1.0

*Project Manager*

Naam:

Datum:

Paraaf:

*Systeembeheer + Accounting*

Naam:

Datum:

Paraaf:

## Inhoudsopgave

Document Historie .....	1
Inleiding.....	3
Methode .....	3
Werkwijze .....	3
Rollen.....	3
Functionele Eisen .....	4
Must Haves.....	4
Should Haves .....	5
Could Haves.....	6
Would Haves .....	6
Niet Functionele Eisen.....	7
Must Haves.....	7

## Inleiding

In dit document staan de requirements voor de NetMatch Server Administration Tool.

## Methode

De requirements zijn genoteerd in de **MoSCoW** methode. Deze methode maakt gebruik van vier niveaus van requirements:

- **Must Have:** Aan deze eis moet voldaan worden om tot een volledig product te komen die geaccepteerd word door de klant.
- **Should Have:** Deze functionaliteit is zeer gewenst, maar ingeval van tijdsnood worden deze achterwegen gelaten.
- **Could Have:** Het zou mooi zijn als deze functionaliteit in de applicatie zit, maar hier zal waarschijnlijk geen tijd voor zijn.
- **Would Have:** Dit zijn leuke toevoegingen maar het is onwaarschijnlijk dat hier tijd voor is, deze functionaliteiten worden normaliter bewaard voor een volgend project om te implementeren.

De eisen aan het systeem zullen verder nog verdeeld worden onder functionele eisen en niet functionele eisen:

- **Functioneel:** een eis die iets zegt over de functionaliteit van de te ontwikkelen applicatie.
- **Niet Functioneel:** een eis die iets zegt over de omgeving waar het systeem mee te maken krijgt. Denk hierbij aan een software systeem waarop de applicatie op moeten kunnen draaien, een specifiek taal die gebruikt moet worden voor het ontwikkelen van de applicatie en/of een globale tijdsgebonden eis aan de applicatie.

## Werkwijze

Alle requirements binnen dit document hebben een unieke code. Deze codes worden gebruikt als referenties naar de requirements. Na goedkeuring van de requirements zullen de codes niet meer veranderen of verwijderd worden. Vervallen of vervangen codes zullen doorstreept worden met de datum waarop ze vervallen zijn en indien ze een vervangende requirement hebben gekregen staat de code van de nieuwe requirement erbij. Vervallen requirements sinds de vorige versie zijn rood aangegeven. Nieuwe requirements sinds de vorige versie zijn blauw aangegeven. Voorbeeld codes:

- **[F-CH-10]** De 10<sup>de</sup> opgestelde functionele “Could Have” requirement aan de Server Administration Tool.
- **[NF-MH-03]** De 3<sup>de</sup> opgestelde niet functionele “Must Have” requirement aan de Server Administration Tool.

## Rollen

Er worden 5 rollen beschreven binnen dit systeem. Hieronder wordt omschreven welke rollen aan wie bedeeft zijn:

- **Administrator:** Jeroen Havermans.
- **Accounting:** Ronald Wisse.
- **Systeembeheer:** Erwin Kooij, Riny Jacob, Niels Abspoel.
- **Management:** Jeroen Havermans, Eric van der Hout, Bart Hemmer.
- **Gebruiker:** Alle bovenstaande.

## Functionele Eisen

### Must Haves

<b>[F-MH-01]</b>	Om toegang te krijgen tot de applicatie dient er ingelogd te worden met je NetMatch Active Directory account. De toegang wordt pas verleend als blijkt dat de gebruiker een rol binnen het systeem heeft.
<b>[F-MH-02]</b>	De gebruiker moet een overzicht van de huidige licenties in kunnen zien.
<b>[F-MH-03]</b>	De gebruiker moet uit het verleden een maand overzicht op kunnen vragen over de licentie zijn kosten en klanten. Hierbij moeten de toen geldige prijzen worden getoond en niet de huidige prijzen.
<b>[F-MH-04]</b>	De gebruiker moet een details pagina van een licentie kunnen bekijken. Hierin staan alle specificaties van de licenties en zijn historische data.
<b>[F-MH-05]</b>	De gebruiker moet een overzicht van de huidig actieve servers in kunnen zien.
<b>[F-MH-06]</b>	De gebruiker moet uit het verleden een maand overzicht op kunnen vragen over de server zijn klanten en licenties.
<b>[F-MH-07]</b>	De gebruiker moet een detail pagina van de server kunnen bekijken. Hierin staan alle specificaties van de server en zijn historische data.
<b>[F-MH-08]</b>	De gebruiker moet een overzicht van de klanten in kunnen zien.
<b>[F-MH-09]</b>	De gebruiker moet een detail pagina van de klant kunnen bekijken. Hierin staan alle specificaties van de klant en zijn historische data.
<b>[F-MH-10]</b>	De gebruiker moet uit het verleden een maand overzicht op kunnen vragen over de klant zijn rackspaces, servers, hardware en licenties.
<b>[F-MH-11]</b>	De gebruiker moet een overzicht van de hardware in kunnen zien.
<b>[F-MH-12]</b>	De gebruiker moet een detail pagina van de hardware kunnen bekijken.
<b>[F-MH-13]</b>	Alle rackspaces, servers, hardware en licenties overzichten moeten te filteren zijn op klant.
<b>[F-MH-14]</b>	De gebruiker moet zich aan en af kunnen melden op automatische mails/notificaties.
<b>[F-MH-15]</b>	Elke maand moet het systeem automatisch een uitdraai maken van de SPLA licentie data per klant en deze door sturen naar de personen die zich hierop aangemeld hebben.
<b>[F-MH-16]</b>	Het moet voor accounting mogelijk zijn om in te zien wat de originele SPLA inkoop prijzen zijn per maand.
<b>[F-MH-17]</b>	Het moet voor accounting en management mogelijk zijn inkoop maand overzichten in te kunnen zien voor servers, hardware, rackspace en licenties.
<b>[F-MH-18]</b>	Accounting, systeembeheer en management moeten klanten kunnen toevoegen aan het systeem.
<b>[F-MH-19]</b>	Bij het toevoegen van servers moet systeembeheer kunnen aangeven of het een hardware server is of een virtuele server.
<b>[F-MH-20]</b>	De Sequint prijslijst voor de SPLA's moet door systeembeheer maandelijks automatisch aan de applicatie toegevoegd kunnen worden ( door een handeling van maximaal 10 muisklikken ).
<b>[F-MH-21]</b>	Het moet voor systeembeheer mogelijk zijn om handmatig licenties toe te voegen.

<b>[F-MH-22]</b>	Systeembeheer moet de licentie aantallen kunnen veranderen en de aantallen lijst maandelijks uit kunnen draaien voor het opsturen naar Sequint.
------------------	---

### Should Haves

<b>[F-SH-01]</b>	Er dient een administrator account te zijn die gebruikers rollen toe kan bedelen en kan ontnemen.
<b>[F-SH-02]</b>	Accounting moet per klant een maand op verwerkt kunnen zetten. Deze data wordt daarna als Historische data beschouwd.
<b>[F-SH-03]</b>	Historische data mag niet gewijzigd of verwijderd worden.
<b>[F-SH-04]</b>	Management moet bij klanten SLA overeenkomsten kunnen beheren.
<b>[F-SH-05]</b>	Er dient bijgehouden te worden als een gebruiker mutaties binnen de applicatie maakt. Hierbij dient de tijd waarop de mutatie voorkwam, de gebruikersnaam en de actie opgeslagen te worden.
<b>[F-SH-06]</b>	Systeembeheer, management en accounting moeten de mutaties door kunnen zoeken en kunnen filteren op start en eind datum.
<b>[F-SH-07]</b>	Het moet voor systeembeheer visueel zichtbaar gemaakt worden in het hardware en server overzicht als de hardware oud is en wanneer de garantie verlopen is.
<b>[F-SH-08]</b>	Systeembeheer moet per hardware aan kunnen passen wanneer deze oud wordt bevonden (standaard 5 jaar na aanschafdatum).
<b>[F-SH-09]</b>	Systeembeheer moet per hardware aan kunnen passen wanneer deze een notificatie/mail geeft van het verloop van de garantie (standaard 6 maanden voor verloop).
<b>[F-SH-10]</b>	Systeembeheer moet per licentie aan kunnen passen wanneer deze een notificatie/mail geeft van verloop (standaard 3 maanden voor verloop).
<b>[F-SH-11]</b>	Systeembeheer moet bij hardware een extra garantie termijn kunnen toevoegen in jaren of in maanden.
<b>[F-SH-12]</b>	Wanneer de prijs van een SPLA handmatig aangepast wordt dient er altijd een notitie ingevuld te worden waarom. Het systeem dient deze SPLA in het overzicht aan te geven met een ander/extra icoontje.
<b>[F-SH-13]</b>	De gebruiker moet in het systeem kunnen zien als de SPLA prijzen meer als een maand niet up-to-date gehouden zijn.
<b>[F-SH-14]</b>	De gebruiker moet de laatste maand zien waarin de SPLA prijzen zijn geüpload.
<b>[F-SH-15]</b>	Systeembeheer moet in kunnen zien op welke locaties hun serverracks staan en welke servers en hardware op welke posities zitten binnen het serverrack.
<b>[F-SH-16]</b>	Systeembeheer moet alle hardware (serverframes, servers, routers, switches etc.) die in het een serverrack geplaatst kan worden op kunnen nemen in het systeem en een positie toe kunnen wijzen in het serverrack.
<b>[F-SH-17]</b>	Systeembeheer moet een server binnen een serverframe een positie kunnen geven. Er is hier de mogelijkheid tussen de volgende formaten [ ¼   ½   1 verticaal   1 vierkant ]. Met de posities beschikbaar voor 8 ( 1 ) of 16 ( ½ ) of 32 ( ¼ ) of een combinatie van deze formaten.
<b>[F-SH-18]</b>	Systeembeheer moet een serverrack toe kunnen voegen aan een data center.

<b>[F-SH-19]</b>	Systeembeheer moet datacenters, serverracks, servers, hardware en serverframes aanpassen verwijderen en voegen aan het systeem
<b>[F-SH-20]</b>	Het systeem moet een waarschuwing geven bij het uploaden van de nieuwe SPLA gegevens als blijkt dat er een licentie vervalt waar nu nog aantallen op staan die besteld dienen te worden.
<b>[F-SH-21]</b>	Systeembeheer en management moeten de huidige SPLA prijslijst door kunnen zetten naar de volgende maand.

### Could Haves

<b>[F-CH-01]</b>	Systeembeheer moet bij de servers en hardware commentaar kunnen plaatsen gekoppeld aan een tijdstip en datum.
<b>[F-CH-02]</b>	Het moet voor accounting en management mogelijk zijn om 2 maanden met elkaar te kunnen vergelijken qua Rackspaces en licenties.
<b>[F-CH-03]</b>	Als een server of hardware uit garantie dreigt te gaan of uit garantie is moet management en systeembeheer mails/notificaties kunnen krijgen.
<b>[F-CH-04]</b>	Systeembeheer moet de standaard tijden aan kunnen passen van: wanneer hardware en servers oud worden bevonden, wanneer garantieverloop een waarschuwing geeft voor hardware en servers en wanneer licentieverloop waarschuwingen geeft.

### Would Haves

<b>[F-WH-01]</b>	Bij het toevoegen van een server moet het voor systeembeheer mogelijk zijn het servicetag nummer van een DELL machine op te geven waarbij het systeem automatische via de DELL API de server gegevens vult. Deze gegevens worden eerst ter controle aangeboden voor dat ze door het systeem opgenomen worden.
<b>[F-WH-02]</b>	Systeembeheer moet bij kunnen houden dat hardware en/of een server op een datum een reparatie gehad heeft.
<b>[F-WH-03]</b>	Systeembeheer moet bij kunnen houden wanneer hardware en/of een server een nieuw component heeft gekregen.
<b>[F-WH-04]</b>	Het systeem moet maandelijks een uitdraai maken van de servers, hardware, licenties en Rackspaces die voor die klant in de betreffende zijn aangeschaft. Deze uitdraai dient door gestuurd te worden naar accounting.
<b>[F-WH-05]</b>	Systeembeheer moet de real-time informatie van de servers kunnen zien.



## Niet Functionele Eisen

### Must Haves

<b>[NF-MH-01]</b>	De taal van de applicatie moet Engels zijn.
<b>[NF-MH-02]</b>	De applicatie wordt geschreven in C# mits zelf de applicatie daadwerkelijk zelf geschreven wordt.
<b>[NF-MH-03]</b>	De pagina's / content die de applicatie laat zien moet binnen 3 seconden geladen zijn. Wanneer dit niet mogelijk is dient de gebruiker hier visueel van op de hoogte gesteld te worden. Bij deze tijden wordt uit gegaan van een standaard werkstation en internet verbinding zoals die bij NetMatch aanwezig zijn.
<b>[NF-MH-04]</b>	De applicatie dient ondersteund te worden in IE9+, Chrome vanuit 2013 en nieuwer en Firefox vanuit 2013 en nieuwer.
<b>[NF-MH-05]</b>	De applicatie dient webbased te zijn, dit wil zeggen dat de applicatie via de browser benaderbaar is en dat vanuit de browser alle functionaliteiten uitgevoerd kunnen worden.

# Bijlage IV: Requirement Gesprekken

Server Administration Tool



*net*match

*In dit document staan de resultaten van de gesprekken vastgelegd, die zijn afgenomen om achter de eisen van de applicatie te komen.*

*De gesprekken binnen dit document zijn afgenomen en uitgewerkt door Colin Duivis.*

## Brainstorm Systeembeheer

Begin van gesprek: 26-02-2014 10:15

Aanwezigen: Colin Duivis, Erwin Kooij, Riny Jacobs, Niels Abspoel

### Huidige applicaties

- **License admin:** Poging tot een applicatie om de Excelsheet te vervangen, maar deze applicatie werkt niet naar behoren.
- **Hosting admin:** Hierin worden de hosting gegevens en prijzen vastgehouden, niet van belang voor de server administratie, maar wel een systeem die veel gebruikt wordt en word vaak als referentie gebruikt in dit gesprek voor gewenste functionaliteiten.
- **Excelsheet Administratie:** Hierin wordt de licentie en server administratie bijgehouden.
- **RackTables:** Hierin wordt bijgehouden in welke datacenters alle serverracks van NetMatch staan en per rack wordt er bijgehouden welke hardware erin zit zoals: servers, switches, routers etc.

### Huidige situatie

Maandelijks komt er een SPLA lijst binnen van SEQUINT met prijzen en licenties. Deze worden nu handmatig ingevoerd in de Excelsheet administratie.

Maandelijks moet er ook terug gerapporteerd worden welke licenties afgenomen worden aan SEQUINT, dit gebeurt nu via de zelfde Excelsheet als dat ze binnen krijgen.

De huidige servers zijn altijd van DELL en het is voor systeembeheer dan ook niet denkbaar dat hier in de toekomst verandering in gaat komen.

Als er nieuwe servers aangeschaft worden, worden deze nu in RackTables ingevoerd voor de locatie en in de administratie Excelsheet om de klant en de licenties bij te houden.

### Gewenste functionaliteiten

1. De SPLA gegevens moeten automatisch in de applicatie ingevoerd kunnen worden.
2. Buiten de SPLA moeten er voor andere software licenties ook informatie bijgehouden kunnen worden.
3. Naast hardware servers moeten er ook virtuele servers toegevoegd kunnen worden aangezien deze soms ook software licenties bevatten die specifiek bij die Virtual Machine horen.
4. DELL Server gegevens moeten automatisch ingevuld worden d.m.v. het opgeven van de server artikelnummer en mits nodig de expresscode en op basis hiervan de DELL API aan te roepen.
5. Alle hardware die in de racks geplaatst kunnen worden moeten in het systeem gezet kunnen worden ( servers, switches, routers ).
6. Er moeten serverframes weergegeven kunnen worden binnen de serverracks welke op zichzelf weer servers kunnen bevatten in een combinatie van (32 kwart servers | 16 halve servers | 8 hele servers) of een combo van deze verschillende formaten servers (zie figuur 1).
7. Er moet bij gehouden kunnen worden of bepaalde hardware nieuwe componenten heeft gekregen en wanneer dit gebeurd is
8. Er moet bij gehouden kunnen worden of bepaalde hardware reparaties gehad heeft.
9. Automatische export maandelijkse SPLA licentie gegevens in CSV formaat.
10. Automatische mail bij dreiging verloop van garantie hardware.
11. Automatische mail bij dreiging verloop licenties van software.

12. Taal van de software moet Engels zijn.
13. Er moet een login mogelijk zijn met de Active Directory gegevens van NetMatch. (Hiervoor is verwezen naar de huidige Hosting Admin applicatie, als voorbeeld).
14. Het moet visueel aangegeven worden als een hardware component "oud" is (standaard is dit bij 5 jaar).
15. Het moet aanpasbaar zijn wanneer een hardware component oud gevonden wordt(standaard 5 jaar).
16. Het moet aanpasbaar zijn wanneer er een notificatie komt dat de garantie van hardware (standaard 6 maanden van te voren).
17. Het moet aanpasbaar zijn wanneer er een notificatie mail komt als de licentie gaat verlopen (standaard 3 maanden van te voren).
18. Het moet opgeslagen kunnen worden dat garantie van hardware verlengt is ( in jaar of in maanden ).
19. Er moet mutatiegegevens bij gehouden worden van wie wat wanneer heeft aangepast.
20. De mutatiegegevens moeten doorzoekbaar zijn d.m.v. begin en eind datum door te geven.
21. Alleen specifieke gebruikers moeten toegang hebben tot het systeem ( systeembeheer, Ronald Wisse, Jeroen Havermans ).

### Bij te houden gegevens

Hieronder staat per entiteit aangegeven welke data hierbij opgeslagen kan worden.

#### Datacenter

De racks binnen het data-center.	De locatie van het data-center
Naam van het bedrijf van het data-center	

#### Serverrack

De hardware binnen het serverrack.	Het aantal rijen dat het serverrack heeft.
De tellingsvolgorde van het serverrack: oplopend of aflopend.	Klant die behoort bij het serverrack.
Prijs per maand	Datacenter waar het server rack instaat

#### Server

Klant waar de server bij hoort	Expres code
Groote van server binnen serverframe	Positie van server binnen serverframe
Historische gegevens licenties	Historische gegevens klanten
Is het een virtuele server	Prijs per maand
Huidige licenties	

---

*Plus alle gegevens van Hardware, zie volgend hoofdstuk.*

---

## Hardware

Naam	Artikelnummer
Typenummer	Model
Merk	Aanschafdatum
Ordernummer	Duur garantie in maanden / jaar
Extra aanschafte garantie in maanden / jaar	Aanschafprijs
Commentaar	Positie van hardware binnen het serverrack (voorkant   achterkant   midden   alle).
Status van de hardware ( Active   Spare   Defunct ).	Aantal rijen die in de serverrack worden bezet.
Huidige serverrack	

## Licentie

Licentienummer	E-mailadres
Aantal licenties	Aanschafdatum
Aanschafprijs	Prijs per maand
Ordernummer	Merk
Model	Afloopdatum
Aantal maanden voor afloop waarop notificatie gestuurd wordt.	Is het een SPLA
Telling afname per: PC, gebruiker, CPU cores	Welke servers deze licentie momenteel gebruiken.

## Klant

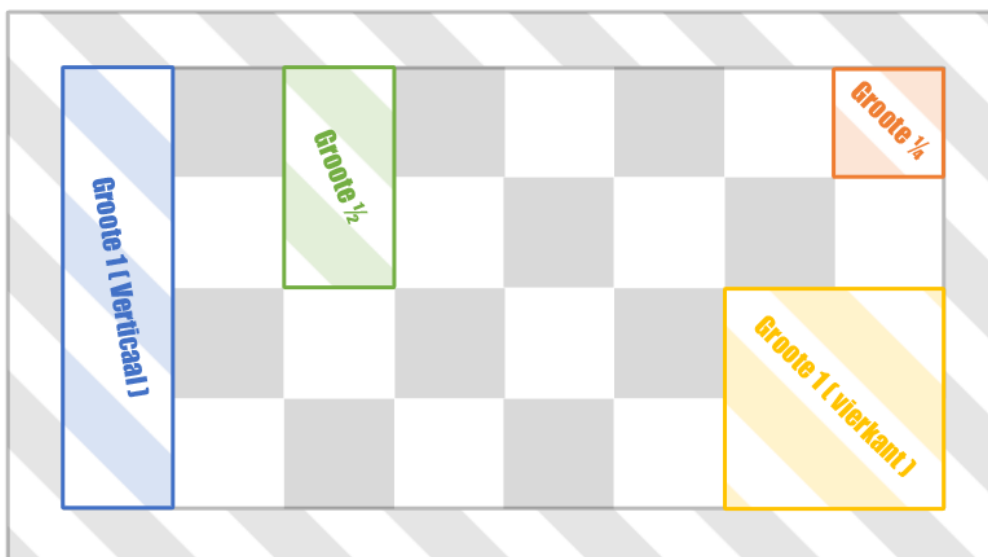
Naam klant	Huidige licenties
Huidige servers	Historische server gegevens
Historische licentie gegevens	

## Extra's

Voorgesteld om visueel de stijl van de Hosting Admin aan te houden.

Voor de toekomst is het leuk om werkstations ook toe te voegen in dit systeem.

Voor de toekomst is het leuk om contact te maken met VM API's om zo gegevens uit te kunnen lezen.



Figuur 1 Schematische vooraanzicht van een serverframe, elk gekleurd blok representeert een van de mogelijke server formaten.

## Brainstorm Accounting

Datum van gesprek: 28-02-2014 13:35

Aanwezigen: Colin Duivis, Ronald Wisse

### Huidige situatie

Systeembeheer stuurt door wanneer er wijzigingen zijn op de SPLA prijzen van SEQUINT, echter gebeurt dit niet altijd aangezien er vanuit systeembeheer weleens iets over het hoofd wordt gezien.

Ook de Rackspace's voor de klanten worden nu vanuit systeembeheer doorgestuurd naar accounting.

Om achter de werkzaamheden van systeembeheer te komen, wordt er meestal een mailtje gestuurd vanuit accounting naar systeembeheer. De werkzaamheden worden dan vervolgens doorgegeven door systeembeheer, echter laat dit soms wel even op zich wachten, omdat systeembeheer deze gegevens dan nog op moet zoeken.

Maandelijks komen er facturen binnen van de verschillende providers waar Rackspace's gehuurd worden en van SEQUINT voor de SPLA licenties. Voor beide van de facturen geldt dat hier de totaal bedragen en aantallen ontstaan van de aangeschafte licenties en gehuurde Rackspace's. Deze zijn voor accounting momenteel niet terug te koppelen naar de klanten, omdat ze alleen maar inzicht hebben in wat er naar maandelijks naar de klanten gefactureerd moet worden en niet weten welke van deze licenties en Rackspace's bij NetMatch zelf horen. Dit maakt controleren en verifiëren van de facturen onmogelijk. Accounting gaat er dan ook maar gewoon vanuit dat deze gegevens altijd goed zijn aangeleverd vanuit systeembeheer.

Vervolgens worden de werkzaamheden van systeembeheer, de licenties vanuit SEQUINT en de ingehuurde Rackspace's bij de datacenters in 1 factuur doorgestuurd naar de klant, dit gebeurt elke maand.

### Huidige tools

Er wordt momenteel alleen gebruik gemaakt van Excel voor klantfacturen op te stellen en mail om met systeembeheer te communiceren wat de juiste gegevens zijn.

### Gewenste functionaliteiten

1. Van elke maand per klant in kunnen zien wat de inkoop prijzen voor de licenties zijn.
2. Inzicht krijgen in de maandelijkse prijslijst die systeembeheer van SEQUINT binnen krijgt.
3. Alle licentie gegevens in kijken van een bepaalde maand.
4. Overzicht van welke hardware is ingekocht voor welke klant en wanneer deze aangeschaft is.
5. Overzicht van welke servers zijn ingekocht voor welke klant en wanneer deze aangeschaft is.
6. Overzicht van welke Rackspace's zijn ingehuurd voor welke klant en wanneer deze aangeschaft is.
7. 2 maanden met elkaar kunnen vergelijken en kijken of er wijzigingen zijn in aantallen / prijzen voor zowel Rackspace's als voor licenties.
8. Notificaties kunnen krijgen wanneer er prijs wijzigingen of aantal wijzigingen plaats vinden bij licenties en Rackspace's.
9. Automatisch kunnen genereren van facturen.

### Belangrijk voor accounting

1. Controle kunnen uitvoeren op inkoop t.o.v. binnen gekregen facturen.
2. Controle kunnen uitvoeren op verkoop t.o.v. inkoop prijzen.

### Bij te houden gegevens

Hieronder staat per entiteit aangegeven welke data hierbij opgeslagen kan worden.

#### Rackspace

De naam van de Provider	De locatie van de rackspace
Naam van het rack	Prijs per maand
Naam van het rack door Provider	Type rack door provider

#### Factuur

De klant waar voor het factuur bedoelt is	Btw
Maand	De systeembeheer activiteiten voor de klant voor gekozen maand
Systeembeheer calamiteiten opslag voor gekozen maand	Opslag licenties
De licenties van de klant voor gekozen maand	Rackspaces voor de klant voor gekozen maand
Opslag rackspace	Factuur nummer

#### Klant

Naam klant	Servers die klant heeft
Rackspaces die klant heeft	Licenties die klant heeft
Factuur geschiedenis van klant	Licentie geschiedenis van klant
Server geschiedenis van klant	NAW gegevens
Debiteurnummer	

#### Hardware / Servers

Naam	Aanschafprijs
Aanschafdatum	Behorend bij klant
Klant geschiedenis	

#### Licentie(SPLA)

Naam	Prijs per maand
Prijs geschiedenis	Afnemende klanten
Geïnstalleerd op servers	Artikelnummer van SEQUINT
Omschrijving van SEQUINT	Aantal in gebruik



Aantal aangeschaft	Aantal aan te schaffen deze maand
--------------------	-----------------------------------

## Brainstrom Projectmanager

Begin van gesprek: 28-02-2014 9:55

Aanwezigen: Colin Duivis, Jeroen Havermans

### Rol binnen de server administratie

Projectmanager en tevens verantwoordelijke systeembeheer.

#### Taken

- Hardware van klanten vervangen ( bij garantie verval ).
- Bij houden van SLA contracten ( klant/project ).
- Controleren hoe de ingekochte licenties tegen over de aangewezen licenties staan.

### Toegang "Server Administration Tool"

CEO: Bart Hemmer, Eric van der Hout.

Managers: Jeroen Havermans.

Accounting: Ronald Wisse.

Heel systeembeheer.

### Gewenste functionaliteiten

1. Bij de klanten de SLA contracten bij houden.
2. Een rapport per maand per klant op kunnen vragen over wat hun licenties en Servers voor die maand waren.
3. Notificatie geven als er prijzen veranderen (zowel voor servers als voor licenties).
4. De inkoop van licenties kunnen vergelijken tegen over de aangewezen licenties over de servers.
5. Notificatie half jaar van te voren als een server uit garantie dreigt te gaan.
6. Notificatie als licentie niet meer leverbaar is.
7. Mogelijkheid om de garantie notificaties voor servers uit te kunnen zetten.

### Bij te houden gegevens

Hieronder staat per entiteit aangegeven welke data hierbij opgeslagen kan worden.

#### Server

Naam	Model
Aanschafdatum	Aanschafprijs
Service contract	Servicetag
Aantal CPU's	Aantal CPU cores
Doel	Huidige klant
Huidige licenties	Historie klant
Historie licenties	Status[active deprecated]
Log van installaties / reparaties	

### Licenties

Naam	Titel
Type [SPLA OEM VSPP MSDN Other]	Key
Prijs	Status[Out of Order Active Expired]
Aantal	Licentietermijn
Prijs Historie	

### Serverrack

Locatie	Hardware binnen rack
---------	----------------------

### Klant

Naam	SLA ( support contracten )
Huidige Servers	Huidige Licenties
Huidige ServerRacks	Historie Servers
Historie Licenties	Historie ServerRacks

### Niet functionele eisen

1. Moet binnen een browser draaien.
2. Moet op de meest recente major browsers draaien maximaal jaar oude versies ondersteunen voor Firefox en Chrome, bij IE vanaf versie 9 ondersteunen.
3. Een maximale laadtijd van 3 seconden ( op een normale pc, met normale internet verbinding ).

### Kern functionaliteiten

1. Tonen overzicht van SPLA licenties en geschiedenis(per maand) in kunnen zien van SPLA licenties.
2. Server overzicht tonen en geschiedenis(per maand) in kunnen zien van de servers.
3. Overzicht van ingezette servers onder klant en licenties onder klant.
4. SPLA inkoop per klant inzien.
5. Rapporteren van SEQUINT aanschaf lijst per klant.
6. Importeren van SEQUINT SPLA gegevens per maand.

### Gewenste Situatie

- De applicatie moet er voor zorgen dat er een transparantere en duidelijkere overdracht van informatie komt tussen Systeembeheer en Accounting.
- Van systeembeheer wordt verwacht dat ze de applicatie actief gaan gebruiken en dat ze de informatie actief up-to-date gaan houden.