

Eindverslag

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

Afstudeerder:

Bram de Jager

20006776

Bedrijf:

eProjects B.V.

Rotterdam

Examinatoren:

dhr. B. van Strien

dhr. H.M.C. Lachman

Afstudeerperiode:

2004-1.1

9 februari 2004 t/m 11 juni 2004

## **Voorwoord**

Voor u ligt het resultaat van de afgelopen vijf maanden. Zij hebben voor mij in het teken van afstuderen gestaan. Alle bevindingen en ervaringen zijn in dit verslag beschreven. Ik heb met veel plezier gewerkt aan het eindresultaat. Enkel resteert mij het bedanken van een aantal die dit mogelijk hebben gemaakt.

Mijn dank gaat uit naar de directie van eProjects, dhr. R.R. Dolle en mevr. H.V. de Ryck, waar ik mijn opdracht uit heb mogen voeren. Ik heb veel nieuwe .NET ervaring opgedaan tijdens de ontwikkeling van de bouw van pilots. Maar ook op het vlak van het totale proces ben ik met raad en daad bijgestaan.

De projectgroep wil ik bedanken voor hun deelname aan besprekingen en het enthousiasme dat zij voor het project toonden. Hun bijdrage heeft op verschillende momenten voor een heldere kijk gezorgd. Speciaal wil ik dhr. D. Roos bedanken voor het bekijken en bedenken van oplossingen voor problemen die de kop opstaken.

Verder bedank ik de twee examinatoren, dhr. B. van Strien en dhr. H.M.C. Lachman, voor hen snelle reacties op vragen en het geven van adviezen.

## **Samenvatting**

In de periode van 9 februari 2004 tot 11 juni 2004 heb ik bij eProjects te Rotterdam afgestudeerd. Het afstuderen op de sector Informatica en Informatiekunde gaat in de vorm van de ontwikkeling van een applicatie. De applicatie die ik ontwikkeld heb is een Content Management Systeem (Cms). Het systeem wordt gebruikt voor het onderhoud van websites. Een belangrijk aandachtspunt was het generieke karakter van het Cms. Het Cms moet onafhankelijk (generiek) zijn voor verschillende websiteontwerpen.

Het proces is volgens de IAD methode doorlopen. Er hebben een aantal iteratieve slagen plaats gevonden tijdens de ontwikkeling van het systeem. Om optimaal gebruik te maken van de gebruikers van het systeem is de pilotontwikkeling geheel doorlopen. Aan het einde van een pilot is de pilot in gebruik genomen door eProjects of klanten.

Het Cms is opgedeeld in vier pilots. De pilots zijn pilot Basis Cms, pilot Gebruikers, pilot Integratie ProSuite en pilot Modules en modulaire opbouw. Per pilot is de definitie studie, pilotontwikkeling en invoering uitgevoerd. Dit proces is per pilot in een hoofdstuk beschreven.

## Inhoudsopgave

Voorwoord .....	i
Samenvatting .....	ii
1 Inleiding .....	1
2 Context .....	3
2.1 Het bedrijf eProjects.....	3
2.2 Markten .....	3
2.3 Organisatiestructuur en cultuur .....	4
2.4 Plaats afstudeerder in organisatie .....	5
3 Opdracht .....	6
3.1 Aanleiding afstudeeropdracht .....	6
3.2 Doelstelling van de opdracht .....	6
3.3 Op te leveren producten .....	7
4 Plan van aanpak .....	8
4.1 Projectorganisatie .....	8
4.2 Bepalen projectgroep .....	8
4.3 Ontwikkelstrategie en -technieken .....	9
4.3.1 Methoden .....	9
4.3.2 Technieken .....	11
4.4 Planning .....	12
4.5 Programmeertaal .....	13
5 Pilotplan .....	14
5.1 Inleiding .....	14
5.2 Pilots.....	14
5.3 Volgorde pilots .....	15
6 Algemene beschouwingen ten aanzien van programmeer-bouweenheden .....	18
6.1 Inleiding .....	18
6.2 Applicatiearchitectuur .NET .....	18
6.3 Database .....	20
6.3.1 Inleiding.....	20
6.3.2 Eén of meerdere databases.....	20

6.3.3	MS Access en/ of MS SQL-server .....	21
6.4	Bepalen systeemeisen .....	22
7	Pilot 1: Basis Cms.....	24
7.1	Definitiestudie .....	24
7.1.1	Systeemeisen .....	24
7.1.2	Systeemconcept .....	27
7.2	Pilotontwikkeling.....	28
7.2.1	Pilotontwikkelplan.....	28
7.2.2	Bouw softwareprogrammeereenheden.....	29
7.3	Invoering .....	35
8	Pilot 2: Gebruikers.....	36
8.1	Definitiestudie .....	36
8.1.1	Systeemeisen .....	36
8.1.2	Systeemconcept .....	38
8.2	Pilotontwikkeling.....	39
8.2.1	Pilotontwikkelplan.....	39
8.2.2	Bouw softwareprogrammeereenheden.....	40
8.3	Invoering .....	45
9	Pilot 3: Integratie ProSuite.....	46
9.1	Definitiestudie .....	46
9.1.1	Systeemeisen .....	46
9.1.2	Systeemconcept .....	48
9.2	Pilotontwikkeling.....	49
9.2.1	Pilotontwikkelplan.....	49
9.2.2	Bouw softwareprogrammeereenheden.....	50
9.3	Invoering .....	56
10	Pilot 4: Modules en modulaire opbouw .....	57
10.1	Definitiestudie .....	57
10.1.1	Systeemeisen .....	57
10.1.2	Systeemconcept.....	59
10.2	Pilotontwikkeling.....	60
10.2.1	Pilotplan.....	60

10.2.2	Bouw softwareprogrammeereenheden .....	60
10.3	Invoering .....	63
11	Evaluatie afstuderen .....	64
11.1	Productevaluatie .....	64
11.2	Procesevaluatie .....	65
	Verklarende woordenlijst .....	67
	Bronnen .....	69
	Boeken .....	69
	Web .....	69
	Overig .....	70
	Interne bijlage .....	71
	Externe bijlage .....	71

## Lijst van afbeeldingen

afbeelding 1: Organigram eProjects.....	4
afbeelding 2: Scenario's IAD.....	10
afbeelding 3: Volgorde concept 1 .....	17
afbeelding 4: Volgorde concept 2 .....	17
afbeelding 5: Applicatiearchitectuur .NET .....	18
afbeelding 6: Opbouw Cms.....	24
afbeelding 7: Eerste weergave menustructuur .....	30
afbeelding 8: Tweede weergave menustructuur .....	31
afbeelding 9: Eerste concept structuur Cms.....	34
afbeelding 10: Koppeling gebruiker en gebruikersgroepen.....	41
afbeelding 11: Weergave gebruikersgroepen .....	42
afbeelding 12: Klassendiagram koppeling artikel en gebruikersgroep .....	43
afbeelding 13: Databasediagram koppeling artikel en gebruikersgroep.....	44
afbeelding 14: Schermverdeling ProSuite; menu, werkbalk en proces .....	47
afbeelding 15: Mappenweergave in bestandsbeheer .....	52
afbeelding 16: Tweede concept structuur Cms .....	53
afbeelding 17: Weergave verschillende locaties, maar hetzelfde artikel .....	54
afbeelding 18: Weergave menustructuur van CmsPro in ProSuite .....	55
afbeelding 19: Databasediagram koppelingen van tabel banner .....	61
afbeelding 20: Scherm status beheer.....	62



## **Lijst van tabellen**

tabel 1: Samenstelling projectgroep .....	9
tabel 2: Klanten die deelnemen aan het testen van het Cms.....	9
tabel 3: Planning afstudeerperiode .....	13
tabel 4: Resultaat onderzoek websiteontwerp .....	26
tabel 5: Stakeholders .....	37



## 1 Inleiding

Met de komst van internet zijn websites niet meer weg te denken uit ons leven. Vrijwel iedereen in Nederland is wel eens in aanraking gekomen met het internet. Voor de een is het vertier en voor een ander is het zijn inkomsten. Websites worden met verschillende regelmaat bezocht. Sommige website worden dagelijks door duizenden bezocht en de website van jantje op de hoek van de straat is blij met één bezoeker per week.

In het begin van internet was het maken van websites alleen weggelegd voor programmeurs. Hedendaags kan iedereen een website maken. Met de introductie van applicaties voor het maken van een website is het maken van een website eenvoudiger geworden. Een website is niet alleen een website maken, maar ook regelmatig veranderen van de inhoud van de website. Door het grote aanbod van websites worden websites met regelmaat bezocht als de inhoud net regelmaat veranderd. Dit is te vergelijken met een reclameblad wat ook één keer gelezen wordt en daarna in de prullenbak verdwijnt. De inhoud van websites moet dus aan te passen zijn.

Samenvattend: de inhoud van een website moet veranderd kunnen worden. Dit moet snel en met regelmaat uitgevoerd kunnen worden. Om het updaten van websites snel, eenvoudig en voor iedereen toegankelijk te maken is er een Content Management Systeem (Cms) ontwikkeld.

Een Content Management Systeem is een systeem waarmee het uiterlijk, de functionaliteit en de inhoud van een site beheerd kan worden. Alle informatie wordt opgeslagen in een centraal systeem. Van daaruit kan dit via de webbrowser beheerd worden. Geen enkele kennis is daarbij nodig van HTML, XML of welke programmeervaardigheid dan ook; iedereen kan het. Voor geavanceerde gebruikers is er altijd de mogelijkheid gebruik te maken van HTML. Middels een autorisatiesysteem kan aan beheerders een stukje van de site toegankelijk worden gemaakt voor aanpassingen, zonder dat dit van enige invloed is op de vormgeving van de site of de rest van de inhoud.

Het Cms maakt het mogelijk een website te beheren, maar dat is niet uniek. Dit product bestaat al langer, maar als het beheer van de website nu onderdeel uitmaakt



van het bedrijf als zijnde een proces. Niet alleen die folder waar alle producten in staan die het bedrijf levert. Maar een systeem wat een website verdeeld in delen voor verschillende geïnteresseerde groepen. Een groep mensen/ werknemers is zelf verantwoordelijk voor hun deel van de website. Zij bepalen de inhoud. Wellicht is de inhoud van dit gedeelte alleen zichtbaar voor die mensen die geabonneerd zijn op de website. Leden die maandelijks lidmaatschap betalen voor informatie wat voor hen interessant is. Misschien kunnen de leden zelf een hun bijdrage leveren door artikelen aan te maken die later na goedkeuring gepubliceerd kunnen worden. Een Cms kan zoveel meer bieden dan alleen een folder...

## 2 Context

### 2.1 Het bedrijf eProjects

eProjects Advanced Software is een in Rotterdam gevestigd ICT bedrijf. Het is opgericht in 1999 en houdt zich bezig met e-commerce en e-business projecten voor het midden- en grootbedrijf. eProjects telt op dit moment 25 medewerkers en er lopen diverse projecten, waaronder een web-based Customer Relationship Management (Crm) systeem, een financieel pakket en een webshop.

Naast generieke projecten voor organisaties als KPMG en Memisa, richt eProjects zich ook op een aantal specifieke branches, zoals de verhuizerbranche of beveiligingsbedrijven.

### 2.2 Markten

eProjects richt zich voornamelijk op:

- Branchemarkten zoals verhuizers- en beveiligingsmarkt
- Klantspecifieke websiteprojecten zoals [www.zuidhollandslandschap.nl](http://www.zuidhollandslandschap.nl) en [www.memisa.nl](http://www.memisa.nl)
- Producten zoals een Customer Relationship Management, Webshop, Boekhoudpakket, etc.

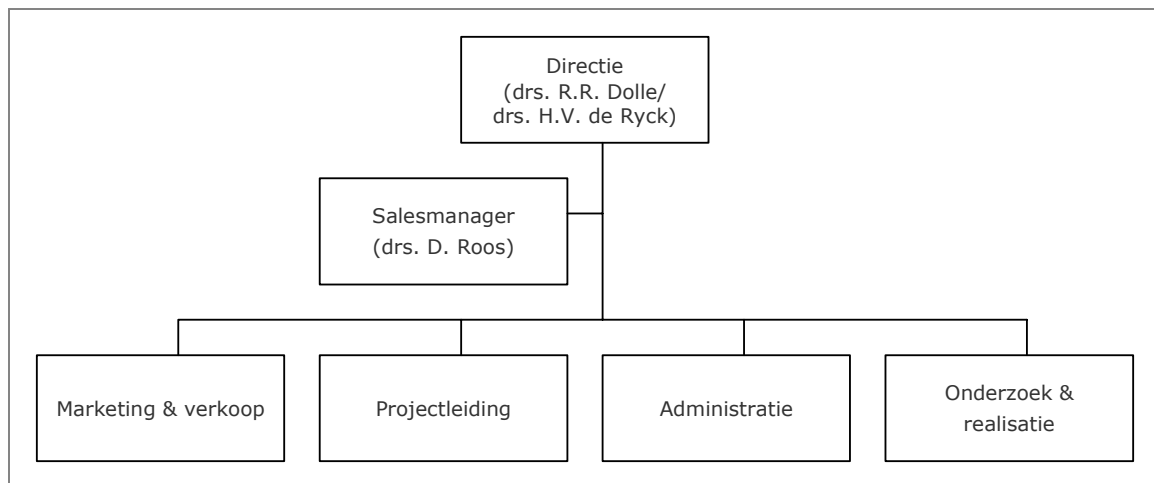
Producten als MoversPro en SecurityPro zijn specifiek ontworpen voor de verhuizers- en beveiligingsmarkt. Beide zijn internetapplicaties. Dit houdt in dat zij toegankelijk zijn via het internet en het enige vereiste is een computer met internet en een browser. Hierdoor ben je niet meer gebonden aan een computer met daarop geïnstalleerde software.

Het niet meer gebonden zijn aan een computer met specifieke software geldt niet alleen voor MoversPro en SecurityPro. Alle applicaties bij eProjects worden op deze manier ontworpen. Dat zulk soort internetapplicaties toekomst hebben, bewijst het

feit dat talloze huidige toepassingen op dit moment geschikt worden gemaakt voor het internet.

### 2.3 Organisatiestructuur en cultuur

Het bedrijf wordt door een tweehoofdige directie geleid. Zij houden zich bezig met het aansturen van het bedrijf. Zij worden ondersteund door de heer D. Roos die zorgdraagt voor de afdeling Sales. Bij afwezigheid van de directie neemt hij waar. Er bevinden zich een viertal afdelingen bij eProjects. Dit zijn Marketing & Verkoop, Projectleiding, Administratie en Onderzoek & Realisatie.



afbeelding 1: Organigram eProjects

Marketing & Verkoop werft klanten en onderhoudt contacten met hen. De afdeling Administratie verzorgt de administratie van het bedrijf. De afdeling Onderzoek & Realisatie draagt zorg voor de ontwikkeling van de applicaties. De projectleiding is verantwoordelijk voor de uit te voeren projecten.

De cultuur binnen het bedrijf is open. Iedereen kent elkaar en vraagstukken zijn adhoc op te lossen. Door de aanwezigheid van weinig hiërarchie binnen het bedrijf kunnen vragen ook direct aan de betreffende persoon gesteld worden zonder tussenkomst van afdelingschefs of andere personen.

## **2.4 Plaats afstudeerder in organisatie**

Als afstudeerder heb ik plaats genomen op de afdeling Onderzoek & Realisatie. Deze afdeling bestaat uit ongeveer 10 medewerkers. Dit zijn voornamelijk ontwikkelaars die zich bezig houden met het ontwikkelen van nieuwe applicaties en het onderhouden van bestaande applicaties. Naast ontwikkelaars wordt deze afdeling versterkt door een aantal webdesigners. Zij werken aan het ontwerp van websites en applicaties.

Als afstudeerder heb ik met betrekking tot de opdracht veel contact gehad met de heer R.R. Dolle en de heer D. Roos. Zij waren mijn bedrijfsmentoren. Voornamelijk in het begin en op het einde van de periode is er veel contact geweest met de heer R.R. Dolle. Hij heeft een tijd in het buitenland doorgebracht en hierdoor zijn de contactmomenten verminderd. In die periode heb ik veel contact met de heer D. Roos gehad.

### **3 Opdracht**

Voor aanvang van het afstudeerproject was er al een Cms aanwezig. Het CmsPro (versie 3.0) is een "standaard" Cms, wat inmiddels niet meer aan de behoefte voldoet van de veel vragende markt. Een eerdere versie van het CmsPro (versie 2.5) is het meest verkochte Cms. Deze is in ASP ontwikkeld en niet in ASP.NET (de standaard voor eProjects).

Hierin is de aanleiding van de opdracht te vinden. Met de kennis van CmsPro versies 2.5 & 3.0 en een wensenpakket van de opdrachtgever is er een goede uitgangssituatie voor de nieuwe versie van het Cms.

Voor aanvang van het afstudeerproject is eProjects bezig geweest met de ontwikkeling van een totaal pakket, waarin alle applicaties van eProjects geïntegreerd zijn. De ProSuite, zoals de naam van de applicatie luidt, heeft onder andere een CrmPro (Customer Relation Management), AccountPro (Boekhoudpakket), MoversPro (Verhuispakket), HrmPro (Human Resources Management) en PrmPro (Proces Management). Het Cms, wat de afstudeeropdracht beslaat, zal naast deze applicaties onderdeel worden van de ProSuite.

#### **3.1 Aanleiding afstudeeropdracht**

Het huidige CmsPro versie 3.0 van eProjects is een "standaard" Cms. Het product heeft alle basisfuncties van een Cms, maar eProjects wil de afzetmarkt vergroten van CmsPro.

#### **3.2 Doelstelling van de opdracht**

Het ontwikkelen van CmsPro 4.0 in de website-ontwikkelomgeving ASP.NET.

### **3.3 Op te leveren producten**

Hieronder staan de op te leveren producten zoals deze in de opdrachtomschrijving vermeld staan. De resultaten voor de opdrachtgever (op te leveren producten):

- Definitiestudie
- Ontwerpdocumentatie
- Beheerdershandleiding (t.b.v. de implementatie van het Cms)
- Testrapport
- CmsPro 4.0

## 4 Plan van aanpak

### 4.1 Projectorganisatie

De afstudeerder

Van de afstudeerder wordt verwacht dat hij volgens het plan van aanpak te werk gaat. In het plan van aanpak staat een globale planning. De planning is een leidraad voor het project. Waar nodig zal de planning aangepast moeten worden.

De afstudeerder zal regelmatig de voortgang van het project rapporteren bij de opdrachtgever en examinatoren.

De opdrachtgever

De opdrachtgever zal beschikbaar moeten zijn voor vraaggesprekken en voor voldoende terugkoppeling. Terugkoppeling heeft betrekking op de op te leveren producten.

De examinatoren

Examinatoren zullen tijdig reageren op de terugkoppeling van de voortgang van de afstudeerder. Waar nodig zullen zij zorgen voor bijsturing.

### 4.2 Bepalen projectgroep

Om alle informatie in te winnen betreffende de systeemeisen moet er een projectgroep worden samengesteld. Deze projectgroep zal bestaan uit de opdrachtgever, programmeurs en gebruikers. Onderstaand een tabel met deelnemers van de projectgroep.

Naam	Functie	Taak
R. Dolle	Opdrachtgever	Voorziet in de opdracht en benodigde informatie m.b.t. eisen van het systeem
D. Roos	Gebruiker/ verkoop	



W. Sussenbach	Webdesign/ gebruiker	Webdesign, maakt interfaces voor Cms
M. van Wijk	Webdesign/ gebruiker	Webdesign voor site voor het Cms
H. Bal	Usability/ Medewerker	Doet onderzoek naar gebruiksvriendelijkheid en de gebruikersinterface
J. Veldhuizen	Programmeur/ gebruiker	
B. de Jager	Afstudeerder	

tabel 1: Samenstelling projectgroep

Mogelijk worden er klanten gevraagd of zij deel willen nemen aan het testen van de gebruikersinterface. Deze klanten vallen onder de categorie gebruikers.

Organisatie	Naam	Functie
Economische Faculteit Rotterdam	Dhr. E. Miedema	President
Erasmus Magazine	Dhr. Hofstede	Hoofdredacteur
eProjects/ MoversPro.com	Dhr. Roos	Sales
Nederlandse Vereniging van Hoofdpijn Patiënten	Dhr. P. Vriezen	Penningmeester

tabel 2: Klanten die deelnemen aan het testen van het Cms

## 4.3 Ontwikkelstrategie en –technieken

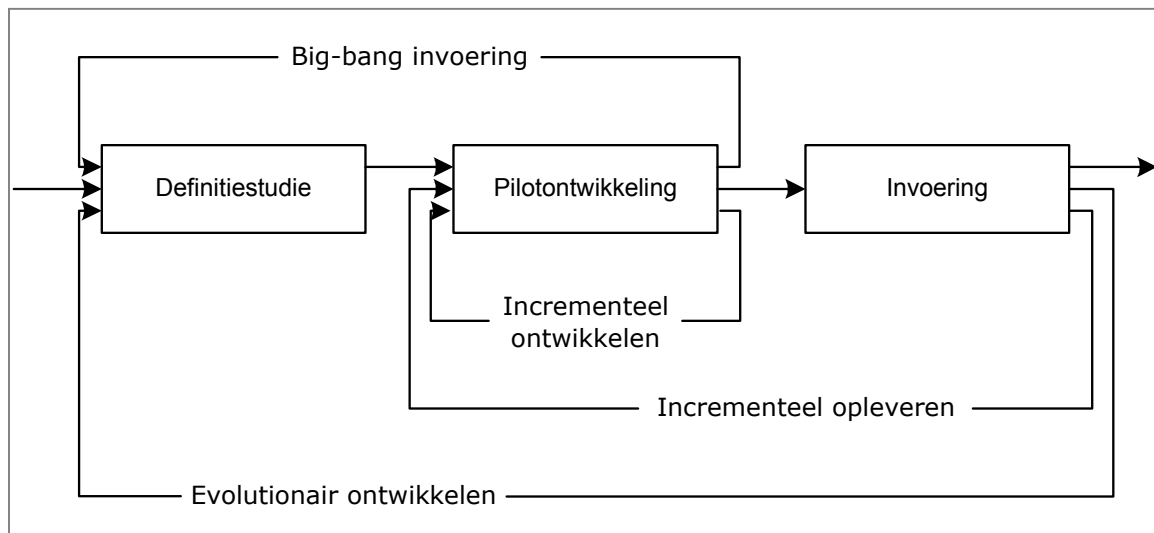
### 4.3.1 Methodes

Een belangrijke scheiding in verschillende ontwikkelmethoden is het doorlopen van de cycli van de methode. Het doorlopen van een cyclus voor de ontwikkeling van een applicatie kan lineair of iteratief worden uitgevoerd.

IAD biedt de mogelijkheid om informatiesystemen op een iteratieve manier evolutionair te ontwikkelen. Door het iteratieve karakter van IAD kan er snel ingesprongen worden op veranderingen. Er zal veel samen gewerkt worden met de

opdrachtgever, gebruikers en ontwikkelaars. Deze verschillende groepen hebben allen een eigen kijk op de applicatie en een eigen belang bij het informatiesysteem. Door de grote diversiteit van mensen die deelnemen in het proces is de kans op veranderingen hoog. Aanpassingen moeten snel doorgevoerd kunnen worden en met IAD is dat mogelijk.

Ik heb besloten om IAD te gebruiken, omdat deze methode een goede aansluiting heeft met het proces. IAD gaat uit van drie scenario's. Elk scenario is een andere manier van het doorlopen van het project.



afbeelding 2: Scenario's IAD

Het product zal tijdens ontwikkeling door eProjects en klanten in gebruik worden genomen. Het in gebruik nemen van tussentijdse producten (pilots) sluit big-bang invoering en incrementeel ontwikkelen uit. De kans op het veranderen van functionele eisen tijdens ontwikkeling van het Cms is aanwezig. Door het in gebruik nemen van het Cms zal er veel feedback zijn van klanten. En tijdens de ontwikkeling kunnen functionele eisen worden aangepast aan de hand van de bevindingen van een voorgaande pilot. Ik heb besloten de definitiestudie bij elke pilot te doorlopen, zodat ik een volledig beeld heb van de te ontwikkelen pilot.

Het scenario 'evolutionair ontwikkelen' sluit hierop aan. De systeemeisen en het systeemconcept worden elke keer herzien, bijgesteld of verder uitgebreid. Alle pilots

worden ingevoerd in de organisatie. En de ervaringen van de ingevoerde pilots vormen belangrijke invoer voor de volgende iteraties van de ontwikkelcyclus.

IAD werkt met pilots. Pilots zijn volledige operationele deelsystemen van het uiteindelijke beoogde systeem. Een pilot is een op zichzelf staand project met als resultaat een werkende applicatie. De pilots worden in het begin van het project gedefinieerd. Hierdoor zal er een afbakening ontstaan van het informatiesysteem.

#### **4.3.2 Technieken**

IAD is een methode, maar elke methode heeft technieken nodig om tot een product op papier te komen. Ik heb kennis van twee verschillende technieken. Yourdon en UML. Yourdon is samen met SDM besproken. De techniek van Yourdon is goed bruikbaar in combinatie met SDM. IAD geeft voorkeur aan UML (Geüniformeerde modelleertaal, Eng.: Unified Modeling Language), dit is een objectgeoriënteerde taal.

Applicaties van eProjects worden op objectgeoriënteerde wijze ontwikkeld. Dit is een standaard van het bedrijf. Objectoriëntatie duidt op een bepaalde structuur in programmatuursystemen. Het kenmerk van objecttechnologie is dat gegevens en functie die deze gegevens bewerken, samengevoegd worden in een eenheid die object genoemd wordt. Elk object is een zelfstandige entiteit binnen het totale systeem. De structuur van het systeem bestaat uit objecten die met elkaar verbonden zijn en met elkaar communiceren.

Objecten zijn een weerspiegeling van de werkelijkheid. Veel fysieke dingen zoals auto, tafel en stoel zijn eenvoudig als object te onderscheiden. Door de nauwe relatie tussen het objecten en werkelijkheid is het model beter begripbaar voor gebruikers.

Dit in tegenstelling tot traditionele methoden, zoals Yourdon, die in alle fasen van het ontwikkelproces verschillende concepten gebruiken. Bij Yourdon is bij elke faseovergang een transformatie van het ene naar het andere concept nodig. Van een fase naar een voorafgaande fase gaan, is daarom niet goed mogelijk.

Principes uit de software-engineering zijn een uitgangspunt voor objectoriëntatie. Een object verbergt zijn binnenkant voor alle andere objecten in je systeem. Dit principe van inkapseling (encapsulation) is welhaast een perfecte vorm van information hiding. Inkapseling voorkomt een belangrijk software probleem. Het aanpassen van software heeft meestal een domino-effect tot gevolg. Kleine aanpassingen hebben vaak ongewenste neveneffecten op grote delen van het systeem. Door middel van inkapseling is het mogelijk deze aanpassingen te lokaliseren. Dit is dus minder werk en maakt de kans op fouten kleiner. Dit geeft grote flexibiliteit en het informatiesysteem kan geleidelijk meegroeien met de behoeften. Dit is een belangrijk voordeel van UML (objectoriëntatie) ten opzichte van Yourdon (traditionele techniek).

Met deze voordelen en in combinatie met IAD, heb ik UML gekozen als de te gebruiken techniek.

#### 4.4 Planning

Tijdens het opzetten van het plan van aanpak heb ik er voor gekozen een globale planning op te zetten. Na het maken van de globale planning heb ik per pilot een gedetailleerde planning gemaakt. Dit is verwerkt in een totale planning<sup>1</sup>.

Onderdeel	Activiteit	Dagen
Opstart		5
Pilot 1		26
Pilot 2		21
Pilot 3		10

---

<sup>1</sup> In de bijlage staat een uitgebreide planning

Pilot 4		10
Verslag		18
<b>Totaal</b>		<b>90 dagen</b>

tabel 3: Planning afstudeerperiode

## 4.5 Programmeertaal

De standaard voor de programmeertaal bij eProjects is het Microsoft .NET framework. Het framework is een raamwerk voor een nieuwe generatie applicaties. De .NET definitie volgens Microsoft: "Het framework ondersteunt 20 verschillende programmeertalen en maakt het mogelijk goedbeveiligde, bedrijfszekere, hoogwaardige applicaties te ontwikkelen, te implementeren en te beheren."

Een van de programmeertalen is C#.NET (uitgesproken als "C-sharp"). C# is door Microsoft ontwikkeld om de concurrentie met Java aan te gaan. C# is gebaseerd op de kracht en controle waar C en C++ om befaamd zijn. C# is de standaard voor nieuwe applicaties die bij eProjects worden ontwikkeld.

Enkele eigenschappen van C#:

- een objectgeoriënteerd programmeermodel dat overerving en exception-handling ondersteunt;
- een algemeen gestandaardiseerd typesysteem;
- een uitgebreide set classes die zaken omvat als gegevenstoegang en de Win32 API;
- geavanceerde ondersteuning voor debugging en profilering (profiling);

## 5 Pilotplan

### 5.1 Inleiding

Het pilotplan omschrijft de verschillende pilots algemeen. Daarna gaat het pilotontwikkelplan verder in op de eisen en technische invulling. Wie doet wat etc.

De methode IAD maakt gebruik van pilots. Pilots zijn volledig operationele deelsystemen van het uiteindelijke beoogde systeem. Een pilot is een op zichzelf staand project met als resultaat een werkende applicatie.

Er worden meerdere pilots gedefinieerd voor de indeling van het project. De verdeling van de pilots is gemaakt aan de hand van MoSCoW<sup>2</sup> (Must have, Should have, Could have en Won't have).

Voordat de definitieve pilotindeling vastgesteld was, hebben meerdere verdelingen de revue gepasseerd. Deze wijzigingen waren te herleiden naar de prioriteit van de ProSuite<sup>2</sup>. In het begin van de afstudeerperiode waren er ideeën om de ProSuite te ontwikkelen. Deze ontwikkelingen zijn snel verlopen en enkele weken later was de ProSuite een werkend pakket. Het Cms moest een onderdeel van de ProSuite worden waarmee veranderingen in het Cms verbonden waren.

### 5.2 Pilots

Om duidelijkheid te creëren in de te ontwikkelen applicatie en het toepassen van de methode IAD, zijn er pilots gedefinieerd. In een gesprek met de opdrachtgever is besloten om 4 pilots<sup>3</sup> te ontwikkelen.

De eerste pilot was duidelijk. Een Cms is een samenhang van één website met verschillende menu-items die elk artikelen bevatten. Dit vormt een basis voor een Cms. Omdat dit een basis is waar de werking van een Cms op gebaseerd is hebben de opdrachtgever en ik besloten hier pilot 1: Basis Cms van te maken.

Tijdens het gesprek werd een duidelijke wens uitgesproken over de mogelijkheid van gebruikers in het Cms. Het moet mogelijk worden voor gebruikers om in te

---

<sup>2</sup> Zie verklarende woordenlijst pag. 67

<sup>3</sup> In de bijlage is het pilotplan, met de verdeling van de pilots, te vinden

loggen op de website en hierdoor meer rechten te krijgen. Naast de mogelijkheid voor inloggen, moet met het oog op de toekomst er een koppeling komen met de ProSuite. In de ProSuite zit een relatiebeheerpakket (Crm) waar relaties een gebruikersnaam en wachtwoord kan krijgen. Deze gebruikersnaam en wachtwoord kan hij gebruiken voor inlog op de website. De integratie van gebruikers vormt pilot 2: Gebruikers.

Een basissysteem met één website, menu-items en artikelen is een goed voor een simpele website, maar veel meer is het niet. Er zullen meer mogelijkheden voor een klant moeten komen. Mogelijkheden zoals het kunnen plaatsen van een poll, evenementenkalender, enquêteformulier, etc. Dit zijn allerlei uitbreidingen van de pilot Basis Cms. Een belangrijk onderdeel is de flexibiliteit van het plaatsen van deze uitbreidingen in de website. Uitbreidingen verrijken het systeem, maar ook in de toekomst zullen er uitbreidingen blijven komen. Deze moeten gemakkelijk te implementeren zijn in het te ontwikkelen systeem. Pilot 3: Modules en modulaire opbouw gaat in op de uitbreidingen, ook wel modules genoemd.

Pilot 4: Integratie ProSuite is de laatste van de vier pilots. Al eerder in dit verslag is er gesproken over de ProSuite en de snelle groei van de applicatie. Tijdens het opzetten van de opdrachtomschrijving waren de plannen voor de ProSuite nog niet bekend bij mij. Maar de integratie in de ProSuite werd tijdens het gesprek duidelijk en is een pilot geworden.

### **5.3 Volgorde pilots**

Tijdens het definiëren van de pilots wordt vrijwel tegelijkertijd de volgorde van de pilots bepaald. De pilot Basis Cms is de eerste pilot die ontwikkeld zal worden. Deze pilot moet een solide basis vormen voor de opvolgende pilots.

De tweede pilot is de pilot Gebruikers. Aan het eind van de afstudeerperiode is het belangrijk voor het bedrijf dat er een werkend systeem staat. De integratie van de ProSuite is op zich belangrijk, maar de gebruikers en uitbreidingen met modules zijn belangrijker. Beide pilots Gebruikers en Modules en modulaire opbouw dragen bij aan een totaal werkend systeem. De uitbreidingen van het Cms vormen een proces dat niet alleen tijdens mijn afstudeerperiode zal lopen, maar daarna zullen

uitbreidingen blijven komen. Pilot Gebruikers is een afgerond geheel en vormt een belangrijk onderdeel van het totaal werkend systeem. Om deze redenen hebben de opdrachtgever en ik pilot Gebruikers tot de tweede pilot gekozen.

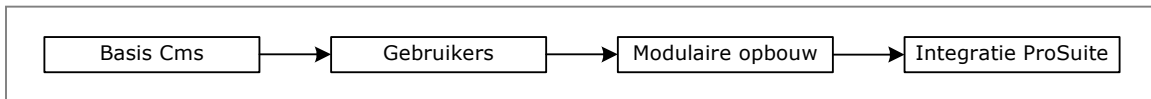
Zoals hierboven al is aangehaald, is de uitbreiding van het systeem, op het moment van bepalen van de volgorde van de pilots, belangrijker dan de integratie in de ProSuite. Pilot drie wordt de uitbreiding van de applicatie met meerdere modules: pilot Modulaire opbouw. Pilot Integratie ProSuite wordt de vierde pilot.

Maar de volgorde van de pilots drie en vier is veranderd tijdens het proces. Door de ontwikkelingen van de ProSuite is er een prioriteitsverandering van de pilots opgetreden. De pilot Integratie ProSuite is een belangrijk onderdeel geworden. eProjects is meer aandacht op de ProSuite gaan richten. De voordelen van het systeem zijn veel groter dan de bestaande systemen. Alle applicaties moeten samen gaan werken in de ProSuite en het Cms dus ook. De integratie van het Cms in de ProSuite is naar voren geschoven in belangrijkheid. Samen met de opdrachtgever heb ik besloten pilot drie en vier om te wisselen.



### *Concept 1*

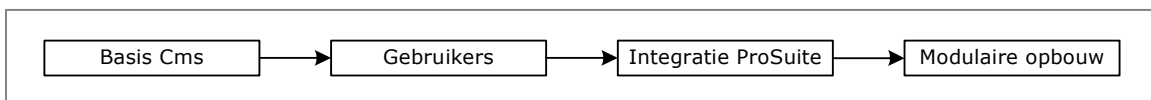
Eerste opzet van de volgorde in de pilots.



afbeelding 3: Volgorde concept 1

### *Concept 2*

Tweede opzet van de volgorde in de pilots.



afbeelding 4: Volgorde concept 2

## 6 Algemene beschouwingen ten aanzien van programmeer- bouweenheden

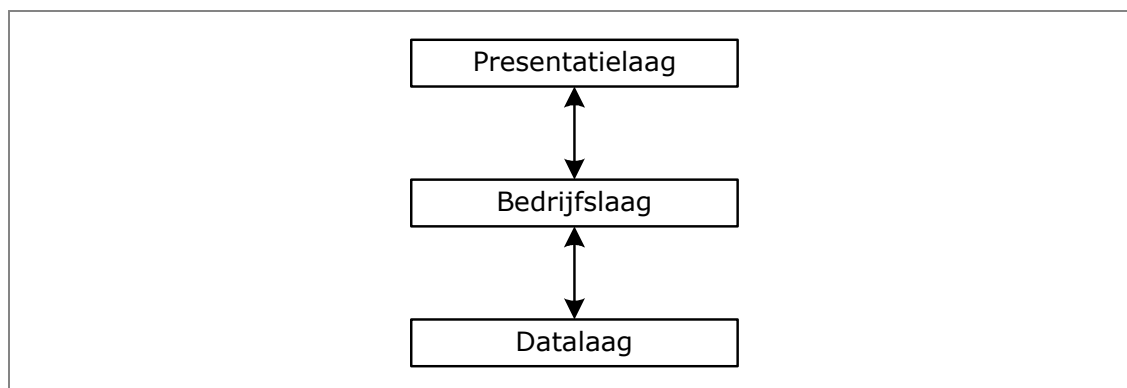
### 6.1 Inleiding

In dit hoofdstuk zal kort een stuk techniek besproken worden

### 6.2 Applicatiearchitectuur .NET

#### *Inleiding*

De opbouw van de code bestaat uit drie lagen. Elke laag heeft haar eigen verantwoordelijkheid. De lagen zijn: presentatielaag, bedrijfslaag en data laag.



afbeelding 5: Applicatiearchitectuur .NET

#### *Presentatielaag*

Deze laag zorgt voor de presentatie van de applicatie. Een andere benaming zou zijn: grafische interface. De grafische interface is op te delen in twee delen: de opmaak van de schermen en het opvangen van de gebeurtenissen op de schermen. Met het opvangen van gebeurtenissen bedoel ik het uitvoeren van een stuk code na het klikken van een knop.

#### *Bedrijfslaag*

In de bedrijfslaag zit de bedrijfslogica. De bedrijfslogica is de middelste laag van de applicatie. Het opvangen van gebeurtenissen wordt door de presentatielaag

afgehandeld, maar functies en berekeningen uitvoeren gebeurt door objecten. Objecten bevinden zich in de bedrijfslaag. Hierin bevindt zich de objectenlogica. Elk object dat is ontworpen in de definitiefase wordt hier omgezet naar code.

Er wordt vanuit de presentatielaag een object gemaakt van de klasse in de bedrijfslaag. Na het initialiseren van een klasse kunnen attributen of functies benaderd worden. Het daadwerkelijk uitvoeren van de functie gebeurt op de bedrijfslaag.

Voor het benaderen van attributen zal er contact met de database gemaakt moeten worden. Dit gebeurt in de datalaag. Vanuit de bedrijfslaag weet een object welke opgeslagen procedure hij nodig heeft, maar het uitvoeren van de opgeslagen procedure gebeurt in de datalaag.

#### *Data laag*

De data laag zorgt voor de connectie met de database. Vanuit de bedrijfslaag wordt een aanvraag gedaan bij de data laag voor het uitvoeren van een opgeslagen procedure. De naam en parameters worden meegegeven en de logica voor het benaderen en uitvoeren van de opgeslagen procedure bevindt zich in deze laag.

Er werd bij eProjects met verschillende datalagen gewerkt. Deze waren ontstaan tijdens ontwikkeling van verschillende applicaties. Omdat er geen standaard was voor de data laag is deze ontwikkeld. De nieuwe data laag is een samensmelting van verschillende bestaande datalagen. Ik had voor aanvang van het afstuderen een eigen data laag ontwikkeld. Ook deze data laag is meegenomen als voorbeeld voor de nieuwe data laag. Met een projectgroep is besloten om de data laag transparant te maken voor de bedrijfslaag. De objecten in de bedrijfslaag weten nooit met wat voor soort database zij werken. Dit kan MS Access, SQL-server of Oracle zijn.

#### *Voor- en nadelen*

De applicatiearchitectuur voorziet in een standaard opbouw van applicaties. Door deze standaardisatie kan een programmeur snel bekend raken met een de programmacode van een bestaande applicatie. Bij het ontwikkelen van een nieuwe applicatie kan direct gebruik worden gemaakt van de data laag.

Specifieke kennis van ADO.NET, voorziet communicatie met database in .NET, is niet nodig. Dit vormt ook direct een nadeel. Specifieke functies van de database kunnen niet gebruikt worden, omdat de data laag transparant moet zijn voor alle databases. Specifieke functies behoren niet tot transparantie.

De bedrijfslaag bevat de klassen van een applicatie. Door de structuur van de applicatiearchitectuur en objectgeoriënteerde eigenschap van .NET is het software principe inkapseling (encapsulation) hier van toepassing. Alle klasse gerelateerde functies bevinden zich in de klasse. Hierdoor wordt het domino-effect van softwareaanpassingen voorkomen en fouten zijn sneller te lokaliseren.

## **6.3 Database**

### **6.3.1 Inleiding**

De database bevat de data van de applicatie. Hierin wordt de website opgebouwd en de database wordt vanuit twee kanten benaderd. Vanuit de website voor het opbouwen van de website en vanuit de ProSuite voor het onderhoud van de website.

### **6.3.2 Eén of meerdere databases**

De ProSuite is zo ontworpen dat alle klanten inloggen op hetzelfde systeem. Door de inloggegevens zijn alleen die gegevens toegankelijk waar de gebruiker recht op heeft. De data laag is zo ontworpen dat elke inlog een eigen of gezamenlijke database kan gebruiken. Na de inlog wordt er voor elk bedrijf een aparte connectiestring opgehaald. In de connectiestring staat welke server, database en gebruiker er moet worden gebruikt voor de connectie. Hierdoor is het mogelijk iedereen dezelfde connectiestring te laten gebruiken of iedereen een eigen.

Ik heb ervoor gekozen om gebruik te maken van één database. Zeker omdat tijdens de ontwikkeling er veel aanpassingen moeten worden doorgevoerd. Als er bij elke aanpassing verschillende databases aangepast moeten worden is dat tijdrovend.

Het Cms moet zo worden opgebouwd dat een klant één of meerdere sites kan beheren. Deze sites zijn ieder weer opgebouwd uit menu's, artikelen, banners, etc.

Dit moet aan elkaar verbonden zijn, want alleen dan is het mogelijk een scheiding van verschillende klanten in één database te maken.

Het aanpassen van de database gebeurt met een script wat uitgevoerd moet worden in de SQL-server. Het script wordt per database uitgevoerd en zorgt ervoor dat aanpassingen aan tabellen en opgeslagen procedures worden doorgevoerd. Het uitvoeren van een dergelijk script op één database of meerdere databases is enkel herhalen van een handeling.

Er zijn meerdere nadelen aan dit systeem. Bij het aanmaken van een site voor een klant kan een script uitgevoerd worden. Er wordt dan een aantal samenhangende gegevens ingevoerd in de database. Daarna kan een gebruiker inloggen in de ProSuite en heeft hij toegang tot zijn site. De hoeveelheid gegevens zal alleen maar groeien. In de hoeveelheid gegevens kan mogelijk een probleem liggen. De gegevens in de database zullen exponentieel groeien. Dit zou kunnen resulteren in het traag worden van het systeem.

Een ander nadeel van een database tegenover meerdere databases is het verhuizen van klanten. In één database met meerdere sites van meerdere klanten is het moeilijk een klant eruit te filteren en deze dan in zijn geheel te verhuizen naar een nieuwe locatie.

Het terugzetten van een back-up van één klant is bij één database niet mogelijk. Het terugzetten van de gegevens van een klant kan alleen wanneer de klant een eigen database heeft. Want bij het terugzetten van een back-up wordt de gehele database overschreven. Doordat er meerdere klanten in een database zitten zullen deze allen worden overschreven.

Mocht er in de toekomst zich verschillende problemen zich voordoen kan er alsnog worden bekeken wat dan de beste oplossingen zijn.

### **6.3.3 MS Access en/ of MS SQL-server**

Het Cms gaat gebruikt worden voor simpele en geavanceerde websites. Voor simpele websites is de MS SQL-server een zware database en zijn de mogelijkheden van MS Access toereikend. Bij integratie in de ProSuite moet het Cms op een SQL-

server werken, want de ProSuite is een op SQL-server werkende applicatie. Het Cms moet dus ontwikkeld worden voor twee soorten databases.

Door de transparantie van de nieuwe datalaag zou de overgang van een MS Access database naar een SQL-server database zonder problemen moeten verlopen. Het verschil in de databases zit voornamelijk in de uitgebreide mogelijkheden van SQL-server. Om ervoor te zorgen dat het Cms met beide databases werkt zullen de queries in MS Access en de opgeslagen procedures in SQL-server hetzelfde moeten uitvoeren. De beperking ligt in de MS Access database en is daarom de uitgangspunt voor het Cms.

Later tijdens de integratie van de ProSuite is besloten het Cms alleen aan te bieden met een SQL-server database. Dit zal toegelicht worden in de beschrijving van de pilot Integratie ProSuite.

## 6.4 Bepalen systeemeisen

Voor het vormen van een beeld van het eindresultaat ben ik begonnen met het bestuderen van CmsPro versie 2.5 en versie 3.0. Deze twee bestaande Cms-systemen vormen de basis voor de systeemeisen. De mogelijkheden van beide systemen zijn opgesomd in een Excelbestand. Dit Excelbestand vormt een samenvatting van de systeemeisen en is voor eProjects overzichtelijk.

Voor aanvang van de afstudeerperiode had ik al kennis van een Cms. Dit systeem heeft de naam DotNetNuke en is een open source applicatie. Na verdieping in de applicatie is bekeken wat de mogelijkheden waren voor het gebruik van de applicatie bij eProjects. Is het systeem bruikbaar voor het nieuwe Cms in plaats van het geheel opnieuw te ontwikkelen? Ik heb de site van eProjects als voorbeeld genomen en deze omgezet in een DotNetNuke site. Hierdoor werden de mogelijkheden van DotNetNuke duidelijk. Het systeem zoals het wordt geleverd door DotNetNuke is uitgebreid en werkt goed. Voor particulieren en bedrijven die weinig eisen stellen aan hun website is het systeem bruikbaar. Maar om een website een eigen uitstraling te geven is het systeem alleen toereikend als men veel verstand heeft van de mogelijkheden van DotNetNuke, maar ook dan blijkt dit nog problemen te geven. Een echt generieke eigenschap heeft DotNetNuke dus niet.



Tevens bij een open source applicatie is er geen garantie dat de applicatie volledig 100% werkt en het is niet in eigen beheer. Dit zou eProjects afhankelijk van anderen maken. De nadelen wegen niet op tegen het zelf ontwikkelen van een Cms. In plaats van het verder ontwikkelen op DotNetNuke kan DotNetNuke een aanvulling zijn op de systeemeisen voor het te ontwikkelen Cms.

Na het opstellen van alle eisen in het Excelbestand is aan alle eisen een prioriteit toegekend. Het toekennen van de prioriteit is eerst door mijzelf uitgevoerd. Ik heb op eigen inzicht de prioriteiten toegekend. Hierna is dit besproken met de opdrachtgever en waar nodig aangepast.

Als de lijst nader bekeken wordt zijn de meeste eisen aan Pilot Modules en modulaire opbouw toe te wijzen. Veel eisen kunnen vertaald worden naar een module. Een voorbeeld is: Eis "Module: Nieuws berichten". Dit is een onderdeel van het Cms, maar is een uitbreiding van het bestaande systeem. Worden die eisen eruit gefilterd, dan zijn er 20 van de 58 over. De meeste hiervan zijn te herleiden naar een basis Cms. Het bestand heeft voornamelijk voor een duidelijk beeld gezorgd in het begin van de afstudeerperiode.

## 7 Pilot 1: Basis Cms

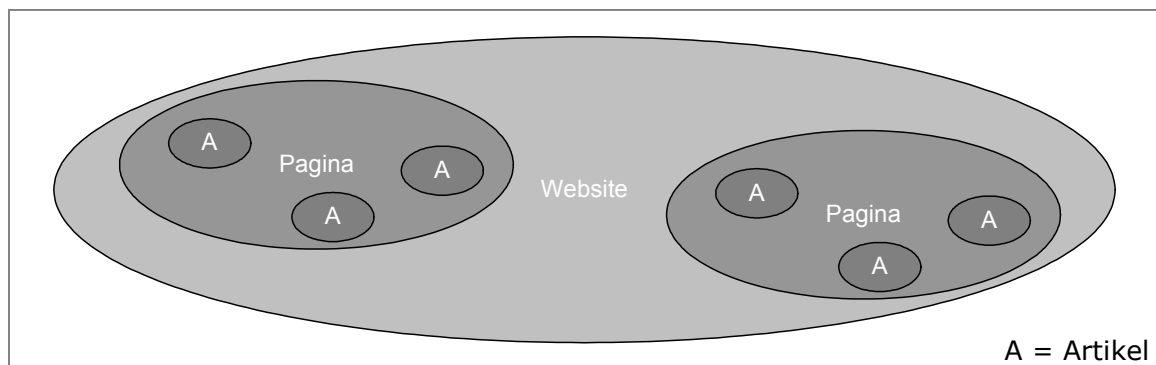
Deze pilot vormt de basis voor het systeem. De pilot moet het mogelijk maken een website te beheren. Het beheer van de website heeft betrekking op het beheer van het menu en plaatsen van artikelen.

### 7.1 Definitiestudie

#### 7.1.1 Systeemeisen

Het bestaansrecht van deze pilot ligt bij de basis van een Cms. De systeemeisen van deze pilot zijn de basisfuncties van een Cms. Hierbij was al snel duidelijk wat een basis Cms was.

Een Cms is een gebruiksvriendelijk, webbased onderhoudssysteem voor het beheren van een website. De basis voor een Cms-website zijn artikelen. Een artikel is een stuk tekst wat op de site wordt geplaatst. Het kan vergeleken worden met een artikel in een magazine. Een aantal artikelen samen vormt een pagina. De website waar de bezoeker op komt is opgedeeld in verschillende pagina's.



afbeelding 6: Opbouw Cms

Hierboven is de opbouw van een Cms te zien. Dit vormt de basis voor pilot Basis Cms. Dus de systeemeisen voor het Basis Cms is het aanmaken, wijzigen en verwijderen van artikelen en menu's (pagina's). Deze eisen zijn tijdens het gesprek over pilotindeling met de opdrachtgever duidelijk geworden.



De moeilijkheid van deze pilot zat in de generieke eigenschap, wat een functionele eis is van het Cms. Het Cms beheert een website, maar er wordt nergens gesproken over het ontwerp (opmaak, uiterlijk) van de website. Elke website heeft een ander ontwerp. Om een generiek Cms te maken moest ik weten of het mogelijk is de verschillende ontwerpen te categoriseren en daarna een selectie te maken voor welke categorieën het Cms geschikt moet worden.

Tijdens een eigen onderzoek naar verschillende ontwerpen heb ik een opsomming gemaakt van verschillende soorten ontwerpen. Na het samenstellen van deze lijst is dit besproken met de projectgroep en resulteerde dit in onderstaande tabel.

Soort	Voorbeeld	Opmerkingen
Webwinkel	<a href="http://www.advancedhosting4u.net">www.advancedhosting4u.net</a>	Hoofdmenu met artikelen en categorieënmenu met subcategorieën met onderliggende producten
Krant	<a href="http://www.volkskrant.nl">www.volkskrant.nl</a>	Hoofdmenu, artikelkoppen en inleiding. Hierna link naar inhoud artikel
Portal	<a href="http://www.ilse.nl">www.ilse.nl</a>	Hoofdmenu, link naar verschillende portals. Ieder eigen uiterlijk en menustructuur
Folder, dropdown menu	<a href="http://www.portal.com">www.portal.com</a>	Home ander uiterlijk als rest van website
Folder, hoofd-/submenu	<a href="http://www.memisa.nl">www.memisa.nl</a>	Continu hoofdmenu met dynamisch submenu
Zoekmachine	<a href="http://www.google.nl">www.google.nl</a>	
Meerdere sites gekoppeld	<a href="http://www.download.com">www.download.com</a>	Een centraal menu met verschillende sites met globale gelijke indeling
Startpagina	<a href="http://www.startpagina.nl">www.startpagina.nl</a>	Geen menustructuur en alleen links

Hybride systemen	<a href="http://www.voelspriet.nl">www.voelspriet.nl</a>	Zoekmachine en links naar allerlei sites
Specifiek		Maatwerk ontwerp
Flash sites		Maatwerk flash sites

tabel 4: Resultaat onderzoek websiteontwerp

Belangrijkste functionele eis die uit de bespreking kwam is het streven naar een mogelijkheid voor het onderhouden van alle ontwerpen. Maatwerk websites zijn moeilijk te onderhouden en vallen daarom buiten de functionele eis.

Tijdens de bespreking werd duidelijk dat de verschillen tussen bovenstaande ontwerpen onder te verdelen zijn in elementen die op een ontwerp te vinden zijn. Voorbeelden van elementen zijn een menu, blok met links of banners. Naast het onderzoek naar verschillende ontwerpen heb ik een onderzoek gedaan met als doel de verschillende elementen op een ontwerp te inventariseren.

In het onderzoek zal gekeken worden welke elementen overlappend zijn met betrekking tot de verschillende ontwerpen. In een nieuw overzicht zijn alle ontwerpen onder elkaar gezet en alle elementen naast elkaar. Zodoende kan er gekeken worden welke elementen daadwerkelijk een ontwerp bepalen.

Uit het onderzoek blijkt dat navigatie is een van de belangrijkste elementen van een website is, maar alle elementen zijn op zich zelf staande onderdelen van een website. In de bijlage staat de Excelsheet met de elementen. Door een website te bekijken vanuit het perspectief dat het bestaat uit pagina's met elementen zouden bovenstaande ontwerpen allemaal onderhoudbaar moeten zijn met het Cms.

Functionele eisen voor pilot Basis Cms:

- Beheren van meerdere sites in een Cms
- Menu's aanmaken
- Menu's wijzigen
- Menu's verwijderen
- Artikelen aanmaken
- Artikelen wijzigen
- Artikelen verwijderen
- Artikelen aan een menu toewijzen
- Volgorde van artikelen in een menu bepalen
- Artikelen in archief plaatsen
- Inlog voor beheerder
- Generiek opzetten van systeem, zodat het Cms op verschillende websites toepasbaar is

### 7.1.2 Systeemconcept

#### *Klassendiagram*

Het klassendiagram geeft de klassen en hun relaties weer. Deze klassen worden aangemaakt in de programmacode en dienen als basis voor het programmeren. Het klassendiagram vormt een belangrijke basis voor de pilotontwikkeling. Een klasse heeft attributen en methodes. Klassen worden aan elkaar geassocieerd als er een relatie tussen beide bestaat.

Ik heb ervoor gekozen de klassen te ontwikkelen op basis van een minimum aantal attributen. Het voordeel is dat er snel resultaat kan worden geboekt met de werking van het systeem zonder te blijven hangen bij specifieke attributen. Het uitbreiden van een klasse met meer attributen is op een later tijdstip relatief weinig werk. De werking van het systeem heeft een hogere prioriteit dan de hoeveelheid attributen.

Bij de iteratieslagen is elke keer het klassendiagram aangepast. Aanpassingen waren voornamelijk van toepassing op het toevoegen van attributen wat extra informatie bevat van een klasse. Men kan denken aan een extra attribuut dat een

extra beschrijving is van de klasse. Een voorbeeld is een attribuut auteur van een artikel toevoegen aan de klasse artikel.

### *Use-cases*

Use-cases zijn voor pilot Basis Cms niet gemaakt. Ik heb hiervoor gekozen omdat de systeemeisen vrijwel voor zich zelf spreken. Zij zijn gebaseerd op CmsPro versie 2.5 en versie 3.0. Deze versies zijn bekend bij de opdrachtgever en de afstudeerder en bij onduidelijkheden kan er worden bekeken hoe dit wordt afgehandeld in CmsPro versie 2.5 en versie 3.0.

### *Sequentiediagrammen*

Door de eenvoudige werking van de pilot Basis Cms was het nodig een sequentiediagram te maken. De communicatie tussen klassen was voornamelijk tussen de klassen van de data laag en de objecten. Deze communicatie verloopt op dezelfde manier. De klasse spreekt de data laag aan en geeft aan dat hij een query wil uitvoeren op de database. De data laag voert dit uit en geeft het resultaat terug aan het object. Hiervoor hoeft geen sequentiediagram gemaakt te worden.

## **7.2 Pilotontwikkeling**

### **7.2.1 Pilotontwikkelplan**

In een pilotontwikkelplan wordt de pilot opgedeeld in pilotdelen, waarna een pilotdeel wordt opgedeeld in bouweenheden. De pilot Basis Cms heb ik opgedeeld in een aantal pilotdelen. Per pilotdeel heb ik een aantal bouweenheden gedefinieerd.

In de systeemeisen van deze pilot staat een inlog voor de beheerder. Deze functionele eis heb ik naar de pilot Gebruikers verschoven. Dit omdat deze pilot enkel over de basis gaat van een Cms en niet de gebruikers.

- Menu
  - Aanmaken databasetabellen
  - Overzicht van menu's
  - Menu aanmaken, wijzigen en verwijderen

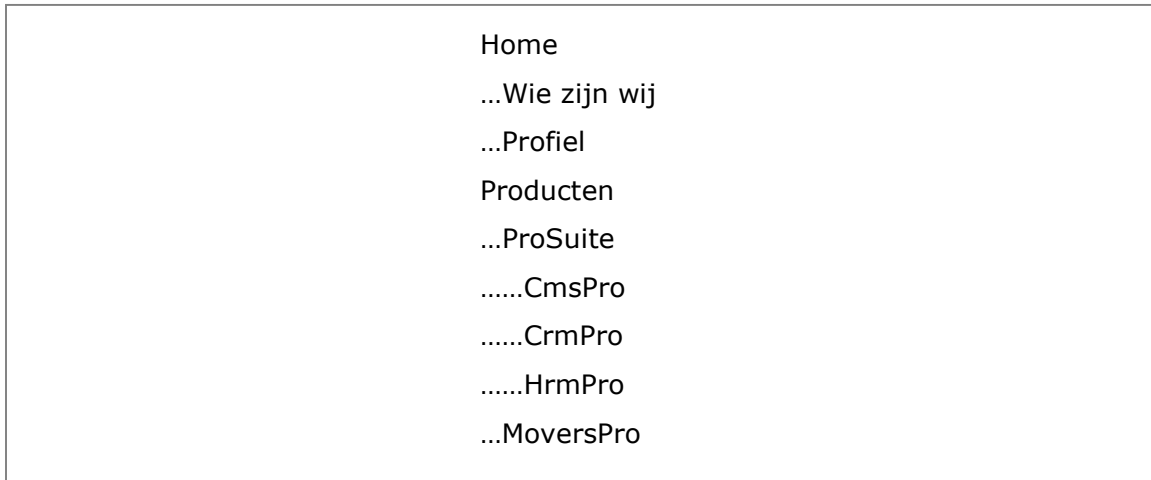
- Volgorde menu wijzigen
- Website
  - Website ontwikkelen (frontoffice)
- Artikel en nieuwsberichten
  - Aanmaken databasetabellen
  - Overzicht van artikelen
  - Artikel aanmaken, wijzigen en verwijderen
  - Overzicht van het archief
  - Overzicht van nieuwsberichten
  - Nieuwsbericht aanmaken, wijzigen en verwijderen

### **7.2.2 Bouw softwareprogrammeereenheden**

#### *Pilotdeel Menu*

In een eerder onderzoek was duidelijk geworden dat de navigatie in de backoffice voor verschillende soorten uiterlijk gelijk was. Een eis van het menu was dat er een oneindig aantal onderliggende menu's aangemaakt kan worden. Door menu's naar elkaar te laten verwijzen kan dit worden gerealiseerd. Een probleem wat hierdoor ontstond was de weergave van de menustructuur. Door het oneindig aantal onderliggende menu's moet de weergave dus dynamisch zijn.

















































Ik ben begonnen met een weergave waarbij de menu's onder elkaar werden weergegeven. Als er een hoofdmenu was met onderliggende menu's dan werden de onderliggende menu's voorzien van een rij puntjes voor de naam. Hierdoor ontstaat een soort boomstructuur.



afbeelding 7: Eerste weergave menustructuur

Dit overzicht voldeed, maar door het dynamische karakter van de menu's ontstonden er problemen met het beheer van de volgorde van de menu's. Dit werd onoverzichtelijk en de veranderingen die samengingen met het verplaatsen van een menu waren groot.

Om verder te ontwikkelen en niet vast te lopen op deze functionaliteit heb ik in overleg met de opdrachtgever de weergave van de menustructuur veranderd. Hierdoor worden de mogelijkheden van het oneindig aanmaken van onderliggende menu's beperkt tot 3 niveaus van onderliggende menu's.

Menu		
Hoofdmenu	Submenu	Sub-submenu
<u>Toevoegen</u> <a href="#">Home</a>     <a href="#">Organisatie</a>     <a href="#">Nieuws</a>     <a href="#">Producten</a>     <a href="#">Diensten</a>     <a href="#">Technologie</a>     <a href="#">Partners</a>     <a href="#">Hosting</a>     <a href="#">Referenties</a>     <a href="#">Vacatures</a>     <a href="#">Contact</a>     <a href="#">Login</a>    	<u>Toevoegen</u>	<u>Toevoegen</u>

afbeelding 8: Tweede weergave menustructuur

Door wel gebruik te blijven maken van de tabelstructuur in de database zoals hij opgezet was, is er op een later moment de mogelijkheid om een oneindig aantal onderliggende menu's aan te maken.

### *Pilotdeel Website*

Nu de structuur van de navigatie van de website bepaald kan worden kan er een demosite worden ontwikkeld. De inhoud van de demosite kan door het Cms worden gewijzigd. Tevens kan er getest worden of het systeem naar behoren werkt.

Het doel van de website is om te testen. Het uiterlijk van de demosite is zeer eenvoudig opgezet. De menu's worden weergegeven en tijdens de ontwikkeling van de artikelen zullen artikelen weergegeven worden. Dit pilotdeel zal steeds verder ontwikkeld worden naarmate de weer te geven items veranderen.

In een later stadium zal een complete website worden ontwikkeld met een volwaardig uiterlijk. Het uiterlijk is voor elke website anders. Maar de elementen (zie § 7.1.1 Systeemeisen) komen op alle ontwerpen terug. Menu vormt het eerste element. Voor de generieke eis zal er een oplossing bedacht moeten worden zodat

het mogelijk is het element qua uiterlijk aan te passen, terwijl het technische gedeelte hetzelfde blijft. Mochten er veranderingen optreden op technisch niveau dan kunnen deze geüpdated worden zonder dat het uiterlijk ervan verandert.

Ik heb in overleg met webdesigners besproken welk gedeelte van een ontwerp door hen ontworpen kan worden en waar geen beperkingen mogen zijn. Er is een duidelijk beeld geschetst van de delen van een ontwerp welke wel en welke niet vast staan. Door een website op te bouwen uit elementen kan een ontwerper deze elementen in zijn website opnemen. Als hij het uiterlijk van deze elementen kan bepalen heeft hij genoeg vrijheid. Deze elementen zullen dynamisch hun inhoud moeten laden.

In de praktijk blijkt deze opbouw niet voldoende. De opbouw voldoet voor een simpele site, welke één soort uiterlijk heeft als de gehele site. Omdat dit Cms een stap verder ontwikkeld moest worden dan voorgaande versies hoort daar een ander uiterlijk ook bij. Verschillende menu's kunnen verschillende soorten uiterlijk hebben.

Een voorbeeld: "Home" geeft een overzicht van samenvattingen van artikelen en aan de rechterkant is een rij met sponsors te zien. Menu "profiel" geeft een indicatie van het profiel van het bedrijf. Dit is een zakelijk stuk en hier zijn alleen artikelen te zien. Deze artikelen zijn in drie koppen verdeeld.

Dit voorbeeld geeft aan dat binnen een ontwerp een menu een ander uiterlijk heeft. Er zal een stap verder gedacht moeten worden. Ik heb veel overleg gehad met andere ontwikkelaars. Deze gesprekken heb ik voornamelijk een op een gehouden in plaats van met de projectgroep. In eerste instantie heeft deze keus alleen betrekking op het uiterlijk van een ontwerp, maar met deze keus verandert er veel in het ontwerp van de database en klassen. Door de moeilijkheidsgraad van de database veranderingen heb ik dit één op één besproken.

De gesprekken resulteerden in een uitbreiding van het databaseontwerp. Na alles te hebben geïnventariseerd heb ik het nieuwe concept besproken met de opdrachtgever. Tijdens het onderzoeken van het nieuwe concept ben ik door gaan ontwikkelen met deze pilot en de volgende pilot. Ik wilde graag een aantal programmeurs spreken die niet altijd aanwezig zijn. Hierdoor liep het onderzoek vertraging op. Door het lange tijdsbestek en het vele malen bekijken was het



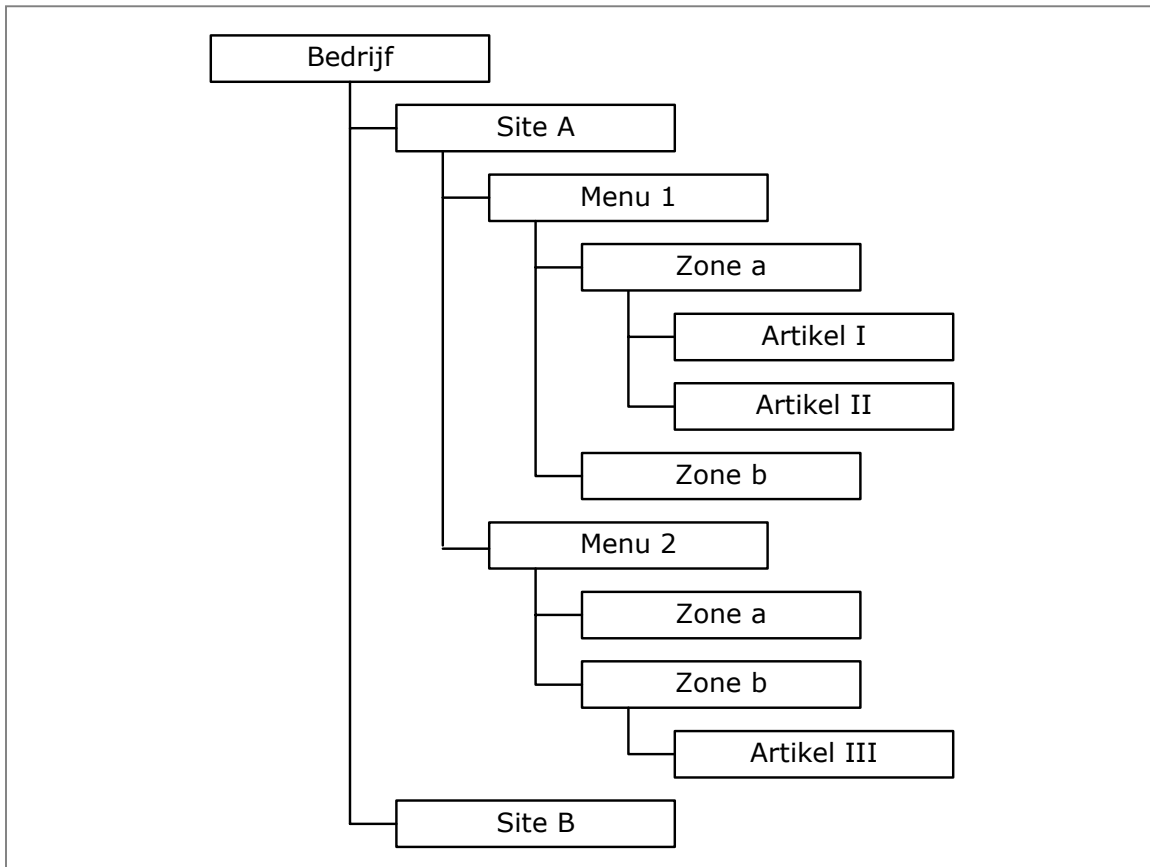
concept naar mijn idee goed uitgedacht en zou dit de oplossing zijn voor zowel kleine als grote sites.

Door de opgelopen vertraging was, gelijktijdig met het nieuwe concept voor de opbouw van een website, pilot 2: Gebruikers bijna afgerond. Ik heb besloten het nieuwe concept gelijktijdig met de pilot 3: Integratie ProSuite door te voeren. Omdat de pilot Integratie ProSuite in tijd naar voren was geschoven. Pilot 1 en 2 vormden nu een samenwerkend geheel en konden deze getest worden. De aanpassingen voor de opbouw kunnen direct met de aanpassingen van het uiterlijk in pilot 3 meegenomen worden.

#### *Pilotdeel artikel en nieuwsitem*

Een artikel is een losstaand onderdeel in het Cms. Het is niet verplicht een artikel direct te koppelen aan een menu. Wellicht maakt men een artikel aan en plaatst het een week later pas op de site. Er mag verondersteld worden dat een artikel op een bepaald moment in de website geplaatst wordt. Dan zal er een koppeling gemaakt moeten worden. De koppeling speelt nogal een grote rol voor de module Artikel, maar ook voor de rest van het Cms.

Op dit moment wordt de opbouw van het Cms een belangrijk onderdeel. De opbouw van een website is in afbeelding 9 te zien. Het betreft hier een artikel, maar dit is eenvoudig te vervangen door een ander element. Voorbeelden zijn banner, polls, etc. Artikel is te beschouwen als een element/ module voor het ontwerp. Artikel is de eerste te ontwikkelen module, maar hierna zullen meerdere volgen. De werking van het plaatsen van het element artikelen op een ontwerp zal de basis vormen voor alle andere te ontwikkelen modules.



afbeelding 9: Eerste concept structuur Cms

Deze opbouw is aan de hand van een aantal iteratieslagen ontstaan. Een artikel wordt direct gekoppeld aan een menu en een zone/ locatie. Dit zal zijn locatie op de website bepalen. Wordt het menu opgevraagd in de frontoffice dan worden per locatie de artikelen opgehaald die erbij horen.

Een voorbeeld: Er bevinden zich artikelen in het midden van de site, maar aan de rechterkant zijn korte nieuwsberichten te zien. Dit zijn ook artikelen. Twee locaties met dezelfde inhoud.

Hier geldt dat het concept goed werkt voor een simpele site, maar na het gebruik van het Cms door gebruikers werd uit de feedback duidelijk dat dit niet voldoende was. Er kwam een vraag naar het plaatsen van één artikel op meerdere locaties. Dit zou voor kunnen komen wanneer op meerdere plaatsen één samenvatting van een artikel wilt plaatsen om de aandacht erop te vestigen. De veranderingen worden doorgevoerd in de pilot Integratie ProSuite.

## 7.3 Invoering

### *Invoeren*

EProjects werkt met Tortoise CVS. Hierdoor wordt voorkomen dat wijzigingen verloren gaan als meerdere personen tegelijkertijd bestanden wijzigen. Tijdens het programmeren maak je lokaal wijzigingen. Als het stuk code werkt en getest is kan het geüpdated worden. Hierna zal in een centrale update het doorgevoerd worden naar de online versie van de ProSuite.

### *Pilotacceptatie*

Voor het in gebruik nemen van de pilot heb ik een gebruikershandleiding geschreven, omdat er voornamelijk andere programmeurs met het Cms gaan werken en zij hier weinig kennis van hebben. De beheerdershandleiding beschrijft hoe je een nieuwe website in combinatie met het Cms moet ontwikkelen.

Tussentijds heb ik veel contact gehad met de webdesigners. Zij gaan werken met het Cms. Zij zullen niet alleen een ontwerp maken met het Cms, maar hebben ook losse modules gemaakt.

### *Feedback*

De feedback van deze pilot is samen met de volgende pilot uitgevoerd en zal in de volgende pilot besproken worden. Dit vanwege het vrijwel gelijktijdig afronden van pilot 1: Basis Cms en pilot 2: Gebruikers.

## 8 Pilot 2: Gebruikers

De pilot Gebruikers gaat het Basis Cms uitbreiden met gebruikersbeheer: het toevoegen van gebruikers en gebruikersgroepen. En deze rechten geven of ontnemen op menu's, artikelen, etc.

### 8.1 Definitiestudie

#### 8.1.1 Systeemeisen

Aan de hand van het gebruikers- en gebruikersgroepenmodel, waar de ProSuite gebruik van maakt, zullen de systeemeisen voor deze pilot worden gedefinieerd. In pilot 3: Integratie ProSuite zal het Cms en onderdeel van de ProSuite is. Mogelijk kan het gebruikers- en gebruikersgroepenmodel van de ProSuite als richtlijn genomen worden.

Ik ben het gebruikers- en gebruikersgroepenmodel in de ProSuite gaan onderzoeken. Doel van het onderzoek was hoe het gebruikers- en gebruikersgroepenmodel werkt en hoe het eventueel toe te passen is in het Cms. Na het onderzoek heb ik dit besproken met de opdrachtgever om tot de systeemeisen te komen.

In het gesprek werd duidelijk dat de integratie van de ProSuite een belangrijk aandeel in het Cms zou hebben. Relaties die in het Crm van de ProSuite aanwezig zijn moeten in de toekomst de mogelijkheid krijgen in te kunnen loggen op de frontoffice van het Cms. Er zijn dus verschillende groepen mensen met verschillende belangen die mogelijk willen gaan inloggen op de frontoffice. Deze belangengroepen zijn stakeholders. In de paragraaf Bepalen stakeholders meer hierover.

#### *Bepalen stakeholders*

Wat zijn de betrokken partijen van het Cms? Met het oog op de toekomstige ontwikkelingen zullen niet alleen klanten inloggen op de website, maar verschillende partijen met eigen belangen en toegang tot delen van de website.

De verschillende belangenpartijen worden stakeholders genoemd. Ik heb een lijst gemaakt met mogelijke stakeholders. Hierna heb ik in overleg met de projectgroep de lijst aangevuld waar nodig was. De verschillende partijen kunnen zijn:

Stakeholder	Website gedeelte
Klanten	Wijzigen gegevens, inschrijven maillist
Advocaten	
Politie	Opvragen vluchtplannen gebouw, werknemerslijst (voor calamiteiten)
Ambulance	Route en alternatieve routes naar bedrijf
Overheid (belasting)	Opvragen jaarcijfers
Sponsors	Overzicht van publicaties van hun advertenties
Accountant	Opvragen jaarcijfers
Afstudeerders/ Stagiaires	Wijzigen gegevens en CV invullen uploaden
Schoolinstellingen	
Gemeente	
Geregistreerde gebruikers/ klanten	Wijzigen gegevens
Potentiële klanten	
Concurrenten	
Bakkerij	
Postbedrijf/ pakketbezorgdienst	

tabel 5: Stakeholders

Bij een aantal bedrijven staat een websitegedeelte ingevuld. Dit zijn voorbeelden van delen van websites waar de stakeholder specifiek alleen voor hem recht op heeft. De gegevens die te vinden zijn op deze gedeeltes kunnen tekst zijn, maar ook gegevens die uit de ProSuite komen.

Een voorbeeld: Als een afstudeerder interesse heeft in het bedrijf kan hij inloggegevens aanvragen. Nadat hij heeft ingelogd vult hij zijn gegevens in. Deze gegevens worden opgeslagen. Als hij later komt werken bij het bedrijf staan deze

gegevens in het systeem. Hij kan direct in HrmPro worden gevoerd voor urenregistratie, krijgt in SrvPro rechten om zijn eigen projecten te beheersen, etc.

Deze koppelingen van verschillende systemen worden pas interessant om te ontwikkelen als systemen volledig werken. Dit zijn luxe eisen. Voor pilot Gebruikers is geen extra aandacht nodig voor deze groep. Het is mogelijk dit in een later stadium te implementeren.

Functionele eisen voor pilot Gebruikers:

- Gebruikers toevoegen
- Gebruikers wijzigen
- Gebruikers verwijderen
- Indelen van gebruikers in gebruikersgroepen
- Gebruikersgroepen toevoegen
- Gebruikersgroepen wijzigen
- Gebruikersgroepen verwijderen
- Toewijzen van rechten aan gebruikersgroepen
- Koppelen van gebruikersgroepen aan menu's, artikelen, etc

### **8.1.2 Systeemconcept**

#### *Klassendiagram*

De uitbreidingen van het klassendiagram brachten veel complicaties met zich mee. Omdat een klassendiagram en databasemodel sterk verbonden zijn beschrijf ik hier de problematiek van beide.

Binnen het Cms is alles dynamisch te bepalen. Voorbeelden zijn: menu-indeling, artikelen en niet te vergeten de gebruikers. Een volledig Cms zonder gebruikers heeft al tal van koppelingen. Een site heeft menu's, menu's hebben weer onderliggende menu's, menu's hebben locaties en locaties hebben artikelen, banners, etc. Alle koppelingen worden opgeslagen in de database en genereert bij het ophalen dus veel databaseverkeer. Bij de ontwikkeling van de gebruikers wordt het aantal koppelingen in de database vergroot. Hierdoor zal het databaseverkeer toenemen. Dit probleem heb ik met een aantal andere programmeurs besproken en

de conclusie was dat dit een onvermijdelijke situatie was. In de praktijk zal moeten blijken of het een acceptabele situatie is. Er ontstaat een mogelijkheid dat het een traag systeem kan worden. Een oplossing kan zijn de SQL-queries aan te passen. Ondanks deze problematiek heb ik gekozen om de werking van het systeem als prioriteit een te beschouwen. Bij een traag systeem kunnen naderhand aanpassingen uitgevoerd worden.

### *Use-cases*

De use-cases vormen de basis voor de sequentiediagrammen. Omdat ik de sequentiediagrammen wil gebruiken voor duidelijkheid in communicatie tussen de klassen zal ik eerst de use-cases moeten maken.

### *Sequentiediagrammen*

In deze pilot gaan klassen elkaar gebruiken en onderling aanroepen om gebruik te maken van elkaars methoden. De onderlinge samenhang heb ik voor mezelf verduidelijkt met de sequentiediagrammen.

## **8.2 Pilotontwikkeling**

### **8.2.1 Pilotontwikkelplan**

De pilot Gebruikers heb ik verdeeld in pilotdelen. Ik heb voor twee pilotdelen gekozen: de Gebruikers en Gebruikersgroepen. Binnen de pilotdelen maak ik ook de koppelingen met de menu's, artikelen, etc. Ik had hier een apart pilotdeel van kunnen maken, maar het was voor mij overzichtelijker om dit onder twee pilots te verdelen en het aantal bouweenheden per pilotdeel te vergroten.

- Gebruikers
  - Aanmaken databasetabellen
  - Overzicht van gebruikers
  - Gebruikers aanmaken, wijzigen en verwijderen
- Gebruikersgroepen
  - Aanmaken databasetabellen

- Overzicht gebruikersgroepen
- Gebruikersgroepen aanmaken, wijzigen en verwijderen
- Gebruiker aan een gebruikersgroep toevoegen en verwijderen
- Gebruikersgroep koppelen aan menu's, artikelen, etc
- Inlogmogelijkheid voor backoffice maken in het Cms
- Inlogmogelijkheid voor frontoffice maken als module

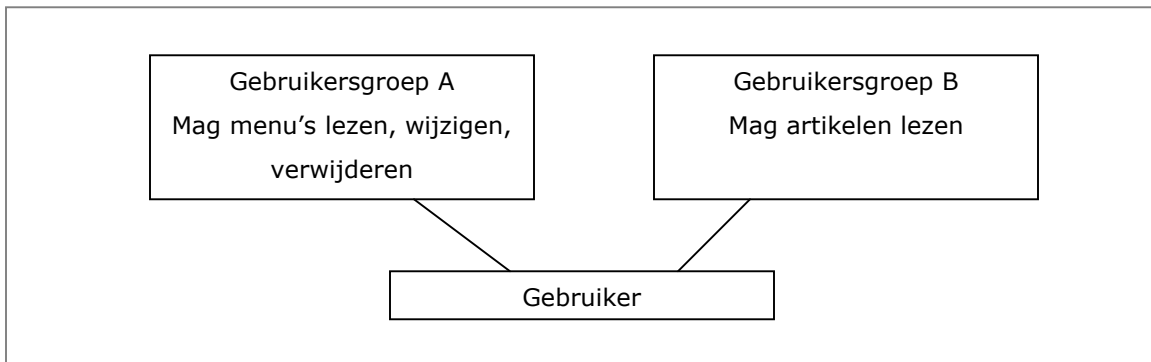
Gebruikersgroepen vormen een groter geheel in de verhouding van gebruikers en gebruikersgroepen. Gebruikersgroepen bevatten meerdere gebruikers. Pilotdeel Gebruikers vormt daarmee een losstaand geheel. Daarom wordt eerst pilotdeel gebruikers ontwikkeld en daarna gebruikersgroepen.

### **8.2.2 Bouw softwareprogrammeereenheden**

Het gebruikers- en gebruikersgroepenmodel van de ProSuite zou als voorbeeld dienen voor het Cms. De werking van gebruikers- en gebruikersgroepenmodel kon ik bekijken in de ProSuite. Het gebruikers- en gebruikersgroepenmodel werkt met een principal. Dit is een functie van ASP.NET voor het inloggen van gebruikers op sites. Het werkt met een rollensysteem.

Een gebruiker logt in op een website. Deze gebruiker heeft rollen toegewezen gekregen door de beheerder van de website. Een rol is te verdelen in het recht tot toegang op bepaalde menu's, artikelen, etc of het recht om menu's, artikelen, etc te mogen lezen, wijzigen of verwijderen. De rechten voor een gebruiker worden bepaald door de gebruikersgroepen waar hij lid van is, want de gebruikersgroepen zijn gekoppeld aan de rechten. Tijdens het inloggen wordt er gekeken naar de rechten van de gebruikersgroepen waar de gebruiker lid van is.





afbeelding 10: Koppeling gebruiker en gebruikersgroepen

Een voorbeeld: De gebruiker, zoals in **Fout! Verwijzingsbron niet gevonden.**, is lid van twee gebruikersgroepen. Hij heeft het volledige recht op menu's, maar mag artikelen alleen lezen en dus niet wijzigen of verwijderen.

#### *Pilotdeel gebruikers*

Voor dit pilotdeel heb ik weinig onderzoek gedaan naar de technische eisen van het gebruikers- en gebruikersgroepenmodel. Omdat de hierop volgende pilot de integratie van de ProSuite inhoudt was het niet interessant om het gebruikers- en gebruikersgroepenmodel zelf te ontwikkelen. Gebruikers krijgen standaardattributen, zoals een gebruikersnaam en wachtwoord. Verdere specifieke gegevens worden tijdens de integratie van de ProSuite toegevoegd.

Ik heb veel tijd gestoken in het bekend raken met het principal van .NET. Omdat het gebruikers- en gebruikersgroepenmodel tijdens de ontwikkeling aangepast moest worden, ging dit samen met de ontwikkeling van de nieuwe data laag.

#### *Pilotdeel gebruikersgroepen*

Dit pilotdeel vormt de koppeling tussen de gebruikers en de pilot Basis Cms. Hier worden de koppelingen tussen menu's, artikelen, etc gemaakt. Gebruikers kunnen inloggen en krijgen rechten. Gebruikers kunnen aan menu's, artikelen, etc gekoppeld worden.

Tabellen en klassen vormen voor deze pilot de basis, net als elke andere pilot. De tabellen en klassen zijn als eerste ontworpen. Na het bouwen van de klassen kan er een overzicht van gebruikersgroepen gemaakt worden. Dit overzicht is hetzelfde als

dat overzicht van artikelen: alle gebruikersgroepen onder elkaar met de mogelijkheid om een nieuwe aan te maken, te wijzigen of te verwijderen.

Gebruikersgroepen			
Naam	Omschrijving		
Beheerders	Heeft alle rechten	<a href="#">edit...</a>	<a href="#">del...</a>
Medewerkers		<a href="#">edit...</a>	<a href="#">del...</a>
Studenten		<a href="#">edit...</a>	<a href="#">del...</a>
<a href="#">Nieuwe Gebruikersgroep...</a>			

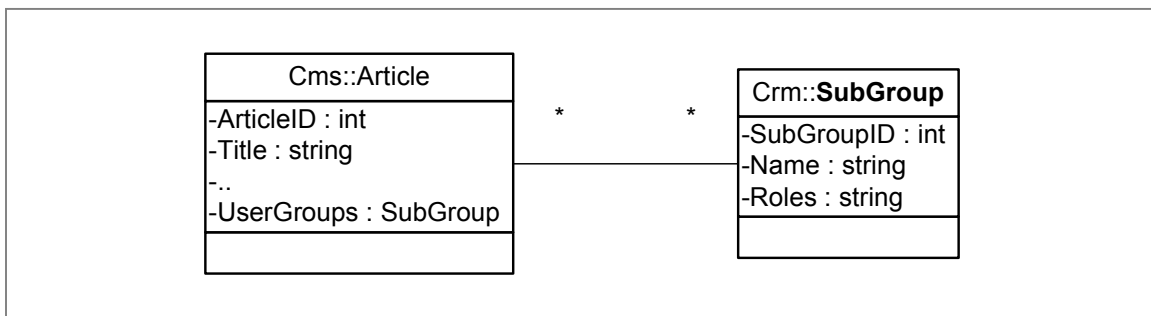
afbeelding 11: Weergave gebruikersgroepen

Vanuit dit overzicht van gebruikersgroepen kunnen de functies voor toevoegen, wijzigen en verwijderen worden ingebouwd. Het bouwen van deze schermen heeft geen problemen opgeleverd.

Nu de gebruikersgroepen in het Cms zijn gebouwd moeten gebruikers lid gemaakt kunnen worden van een gebruikersgroep. Ik heb ervoor gekozen om vanuit de gebruiker te werken. Om een gebruiker lid te maken van een gebruikersgroep moet de beheerder naar het wijzig scherm van een gebruiker gaan. Daar kan de gebruikersgroep aan de gebruiker toegevoegd worden. Om de overzichtelijkheid te behouden kan in het scherm van een gebruikersgroep de gebruikers opgevraagd worden die lid zijn van die gebruikersgroep.

Gebruikers zijn ingedeeld in gebruikersgroepen en krijgen via de gebruikersgroepen hun rollen. Niet alleen op de backoffice moeten gebruikers in kunnen loggen, maar tevens op de frontoffice. Een gebruiker kan zich inloggen en krijgt daarmee het recht om een beschermd gedeelte te lezen. Om geen dubbele gebruikers te krijgen is het eenvoudig om het gebruikers- en gebruikersgroepenmodel zowel voor de frontoffice als de backoffice te gebruiken. De eis om menu's, artikelen, etc te koppelen aan gebruikersgroepen heeft hier betrekking op. Hier komt het probleem wat beschreven is in de paragraaf

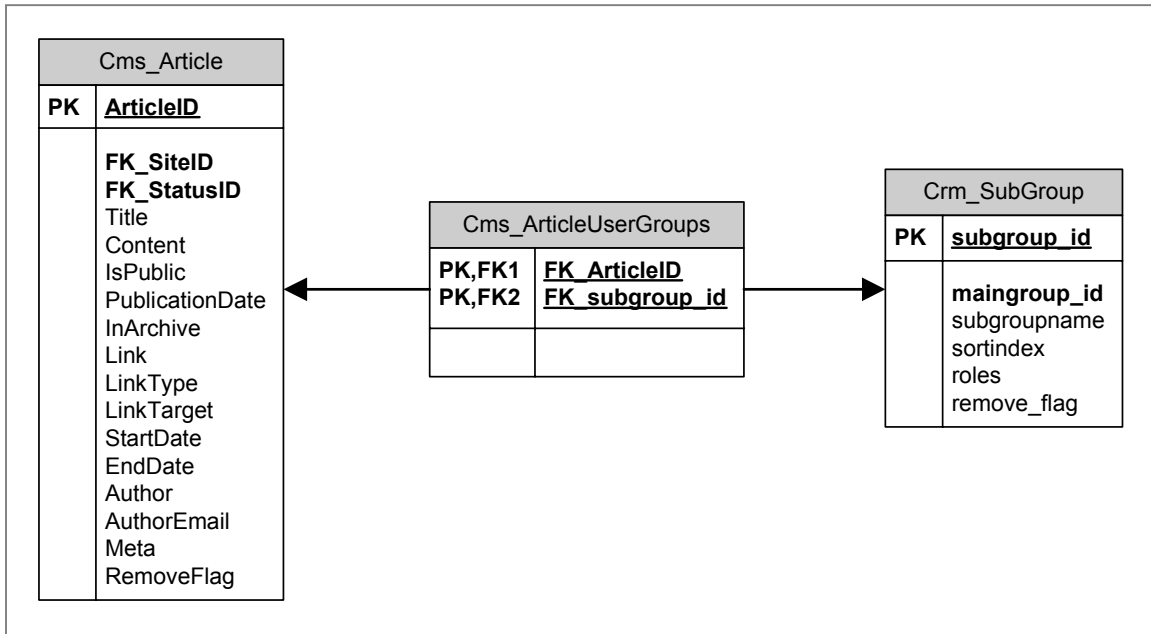
Klassendiagram van deze pilot aan de orde. Ik had besloten om koppeltabellen te gebruiken. In het klassendiagram ziet het er als volgt uit.



afbeelding 12: Klassendiagram koppeling artikel en gebruikersgroep

Een artikel kan aan meerdere gebruikersgroepen gekoppeld zijn. Maar een gebruikersgroep kan ook aan meerdere artikelen gekoppeld zijn. Dit is een veel op veel relatie. Deze veel op veel relatie is in de database niet direct te koppelen. Dit is wel mogelijk met een koppeltabel. Meeste klassen kunnen direct één op één overgenomen worden uit de database. Maar voor dit soort koppelingen is dat niet mogelijk.

In afbeelding 13 is een van de voorbeelden van de koppeltabellen in het Cms te zien. Maar deze koppeling van een module (artikel) met een gebruikergroep zal alleen maar groeien. Alle modules die ontwikkeld worden zullen gekoppeld worden aan gebruikersgroepen. Modules moeten soms wel en soms niet zichtbaar zijn voor gebruikersgroepen.



afbeelding 13: Databasediagram koppeling artikel en gebruikersgroep

Door een koppeling zoals hierboven is het mogelijk om een gebruiker in te laten loggen op de frontoffice en dan zijn artikelen waar hij recht op heeft op te halen. In de backoffice werkt de koppeling op een andere manier. Een gebruiker die inlogt op de backoffice kan daar alle artikelen zien. Als hij het recht heeft om artikelen te wijzigen dan kan hij deze wijzigen. Door feedback en gesprekken werd duidelijk dat dit niet voldoende was. Alleen het recht hebben om artikelen te wijzigen was niet voldoende. De volgende stap was het kunnen wijzigen van artikelen als de gebruiker lid was van de groep die het recht had om dat artikel te wijzigen. Dit was een stuk uitgebreider dan alleen het recht hebben op het wijzigen van artikelen in het algemeen.

Een voorbeeld: De kuierlatten is een website met verschillende wandelroutes. De wandelroutes worden onderhouden door verschillende mensen. Iedereen houdt zijn eigen wandelroute(s) bij en heeft dus niet het recht om andere wandelroutes te lezen, te wijzigen of te verwijderen.

Om de gebruikersinterface overzichtelijk te houden heb ik ervoor gekozen om alle artikelen weer te geven. De artikelen waar de gebruiker recht op heeft krijgen een wijzigknop en andere niet.

Het gebruikers- en gebruikersgroepenmodel is door het gehele Cms doorgevoerd: vanaf het hoogste niveau, de site, waar een gebruiker recht op heeft tot het laagste niveau, een artikel. Hierdoor is alles geheel te beheersen door het bedrijf zelf.

### **8.3 Invoering**

#### *Invoeren*

Pilot Gebruikers is vrij snel na pilot Basis Cms ingevoerd. Doordat beide pilots gereed waren kon het Cms getest worden met klanten. Tijdens mijn afstudeerperiode heb ik meerdere gesprekken met klanten gevoerd. Een van deze klanten was de Economische Faculteit Rotterdam (EFR). Zij wilden gebruik gaan maken van het Cms. Het Cms is door hen als eerste klant ingezet.

#### *Pilotacceptatie*

De beheerdershandleiding is uitgebreid met het gebruikers- en gebruikersgroepenmodel. Tussentijds heb ik veel contact gehad met webdesigners. Zij gaan werken met het Cms. Zij zullen niet alleen een ontwerp maken met het Cms, maar hebben ook losse modules gemaakt.

#### *Feedback*

Na het afronden van de pilot is er een testsite gemaakt. De testsite was een mogelijk nieuwe site voor eProjects. Vanwege het geslaagde uiterlijk van de testsite is besloten om hier de nieuwe website van eProjects van te maken. Dit was een goede testcase voor het Cms. Hierna heb ik feedback gekregen van de opdrachtgever en dit is verwerkt in het Cms. Zie in de bijlage het testrapport.

## **9 Pilot 3: Integratie ProSuite**

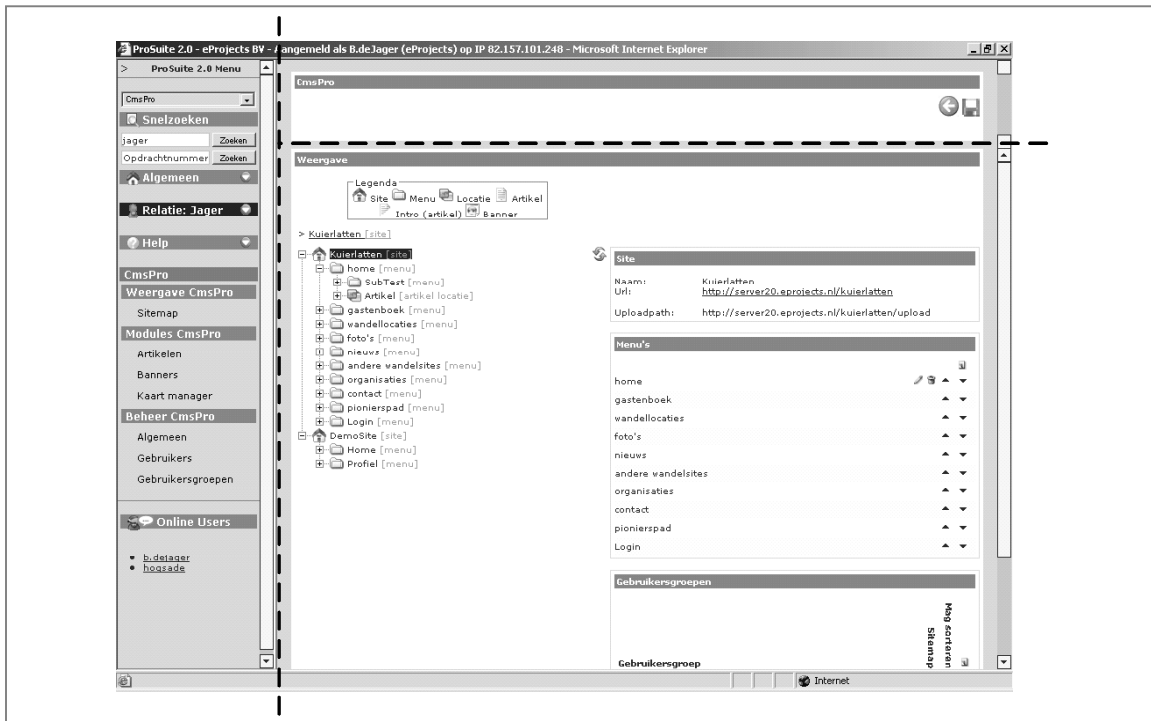
De ProSuite is het totaalpakket van eProjects. Dit product koppelt alle applicaties van eProjects. Het systeem werkt met een gebruikers- en gebruikersgroepenmodel. Als gebruiker krijg je rechten voor het gehele systeem. Het Cms maakt onderdeel uit van de ProSuite.

### **9.1 Definitiestudie**

#### **9.1.1 Systeemeisen**

De belangrijkste eisen voor deze pilot liggen in het uiterlijk van de ProSuite. Het streven naar één uiterlijk voor alle overeenkomstige functies in de ProSuite is het uitgangspunt voor deze pilot.

De opbouw van de ProSuite bestaat uit drie delen. Aan de linkerkant is het menu constant aanwezig voor de navigatie in de ProSuite. Vanuit het menu zijn de belangrijkste functies benaderbaar. De rechterkant is opgedeeld in twee delen. Bovenaan is een soort werkbalk met de knop voor opslaan. Onder de werkbalk is de weergave het proces. Hieronder is een weergave van het uiterlijk van de ProSuite. In een applicatie is altijd een aantal overzichten, functies en weergave van items hetzelfde. Het uiterlijk van een overzicht van relaties kan vrijwel hetzelfde zijn als dat van artikelen. Naast overzichten is de functie voor opslaan gestandaardiseerd in de ProSuite.



afbeelding 14: Schermverdeling ProSuite; menu, werkbalk en proces

Het Cms moet onderdeel worden van de ProSuite en dat wordt bijgestaan door hetzelfde uiterlijk. Dus het overnemen van het uiterlijk wordt een functionele eis. Een onderdeel daarvan is de knop opslaan, welke zich rechts bovenaan bevindt in afbeelding 14, moet door de gehele applicatie zorgen dat alles opgeslagen wordt.

Tijdens de ontwikkeling van de pilot Basis Cms zijn er problemen geweest met de opbouw van de menu's. Tijdens de overstap naar een nieuwe interface voor de backoffice kan tevens de menustructuur opnieuw worden ontwikkeld. De basis voor de structuur is al aanwezig. Ik heb besloten nogmaals het probleem te bekijken en een mogelijke oplossing te zoeken.

Tevens was uit pilot Basis Cms gebleken dat de opbouw van een website moest veranderen. Door de vertraging die opgelopen was en de daarmee samenhangende beslissing dit door te voeren in deze pilot is dit een van eisen van deze pilot.

Het gebruikers- en gebruikersgroepenmodel dat in de pilot Gebruikers is ontwikkeld, moet vervangen worden door het gebruikers- en

gebruikersgroepenmodel wat in de ProSuite aanwezig is. De gebruikersgroepen die aan artikelen, banners, etc toegevoegd kunnen worden moeten de gebruikersgroepen in de ProSuite worden.

Bedrijven die hun relaties in de ProSuite beheren, beheren ook hun relatiegroepen in het Crm van de ProSuite. Deze gegevens zijn al aanwezig in de ProSuite en bij het aanmaken van nieuwe items in het Cms kunnen snel desbetreffende relaties worden toegevoegd.

Functionele eisen voor pilot Integratie ProSuite:

- Het Cms moet hetzelfde uiterlijk als de ProSuite krijgen
- De opbouw van de website moet veranderen en een betere weergave van de structuur van de website
- Het gebruikers- en gebruikersgroepenmodel moet vervangen worden met het gebruikers- en gebruikersgroepenmodel van de ProSuite

### 9.1.2 Systeemconcept

#### *Klassendiagram*

Door de wijzigingen in de structuur van de website moesten er aanpassingen worden gemaakt in het klassendiagram. Er zijn klassen toegevoegd en gewijzigd. Ik had het nieuwe klassendiagram gemaakt samen met een databaseontwerp. Tijdens het ontwerpen van het klassendiagram waren wezenlijke verschillen tussen database en klassendiagram terug te vinden. De één op één relatie tussen klassendiagram en database die in het begin van de ontwikkeling aanwezig was verdween compleet. Toen het klassendiagram en het databaseontwerp af waren, heb ik dit uitvoerig besproken met andere programmeurs. Tijdens de gesprekken werd duidelijk dat met de huidige eisen een ander ontwerp niet mogelijk was.

Voor wat betreft het gebruikers- en gebruikersgroepenmodel is een aantal tabellen verwijderd uit de Cms-database en vervangen door de tabellen die al in de database van de ProSuite aanwezig waren. De tabellen waren die van de gebruikers en gebruikersgroepen. De koppeltabellen met de modules zijn blijven bestaan. Alleen de vreemde sleutels zijn ergens anders naar verwezen.



### *Use-cases*

De veranderingen in de structuur van de website hebben nogal wat wijzigingen te weeg gebracht. De use-case diagrammen zouden bijna gelijk zijn aan die van de pilot Basis Cms. Omdat ik deze niet gemaakt heb voor pilot Basis Cms en deze pilot toch wat veranderingen had heb ik besloten de use-cases te maken voor deze pilot. Tevens zijn de use-cases voorbereidingen op de sequentiediagrammen.

### *Sequentiediagrammen*

De sequentiediagrammen heb ik gemaakt. Zij geven een goed overzicht in de communicatie van de klassen. Dit kwam erg van pas om duidelijkheid te creëren in de opbouw van de structuur.

Ik heb de sequentiediagrammen alleen gebruikt voor de structuur van de website en niet voor het gebruikers- en gebruikersgroepenmodel. Omdat het gebruikers- en gebruikersgroepenmodel al bestond heb ik besloten dat de sequentiediagrammen niet nodig waren.

## **9.2 Pilotontwikkeling**

### **9.2.1 Pilotontwikkelplan**

Deze pilot is opgedeeld in drie pilotdelen. De pilotdelen zijn aan de hand van de functionele eisen gedefinieerd. Hierna zijn de pilotdelen uitgesplitst in bouweenheden.

- Uiterlijk ProSuite
  - Overzetten van Access database naar SQL-server database
  - Eventuele data laag aanpassingen
  - Aanpassen uiterlijk artikelen
  - Aanpassen uiterlijk bannermodule
- Menustructuur
  - Weergave menustructuur
  - Doorvoeren nieuwe opbouw website
- gebruikers- en gebruikersgroepenmodel

- Database en klassen aanpassen
- Schermen ontwerpen en aanpassen waar nodig

Uit voorgaande pilots is gebleken dat de modules beschouwd kunnen worden als losstaande delen die een klein aantal koppelingen hebben met het Cms. Omdat de modules gedeeltelijk losstaan van het Cms heb ik besloten deze als eerste over te zetten naar het uiterlijk van de ProSuite. Voordat ik aan het uiterlijk van de modules kon werken moest de database overgezet worden. Met het overzetten van de database van MS Access naar SQL-server is een andere connectie nodig. De nieuwe data laag zou dit moeiteloos moeten oppakken. De klassen zouden het niet mogen merken dat het op een MS Access of SQL-server database werkt. Omdat dit met de nieuwe data laag nog niet getest is heb ik hier een aparte bouweenheid van gemaakt.

Na de klassen en het uiterlijk van de modules kan de structuur van de website worden doorgevoerd. Naast het aanpassen van de structuur kan tevens een andere manier van weergave onderzocht worden.

Als laatste zullen de gebruikersgroepen zoals die in de ProSuite bestaan geïntegreerd worden met het Cms.

### **9.2.2 Bouw softwareprogrammeereenheden**

#### *Pilotdeel uiterlijk ProSuite*

Omdat de ProSuite een op SQL-server gebaseerde applicatie is moest het Cms worden omgezet naar een SQL-database. Ik heb alle tabellen opnieuw aangemaakt in de database. De queries zoals deze in MS Access aangemaakt waren konden niet overgezet worden en moesten ook opnieuw aangemaakt worden. De queries zijn te vergelijken met de opgeslagen procedures in SQL-server.

Nadat de tabellen en opgeslagen procedures waren aangemaakt kon de connectie met de database veranderd worden. De MS Access database wordt vanaf dit moment niet meer gebruikt. Een belangrijk punt om niet te vergeten was dat op een later tijdstip de applicatie mogelijk naar MS Access teruggezet moest worden. Om te voorkomen dat dit niet meer kan zullen de opgeslagen procedures niet te uitgebreid mogen worden. Opgeslagen procedures kunnen allerlei geavanceerde functies

uitvoeren die queries in MS Access niet kunnen. Dit heb ik van te voren besproken met de opdrachtgever. De opdrachtgever heeft te kennen gegeven dat het Cms in combinatie met de ProSuite verkocht zal worden. Dit houdt in dat het Cms met een SQL-server zal werken en niet meer met MS Access.

Door de transparantie die de data laag voorziet zouden de klassen moeiteloos moeten werken met de nieuwe database. Dit kon getest worden samen met de ontwikkeling van de gebruikersinterface voor de modules. Met enkele kleine aanpassingen van de data laag werkte hij volledig.

Voor het ontwikkelen van schermen met het uiterlijk van de gebruikersinterface van de ProSuite ben ik me gaan verdiepen in de opbouw van schermen in ProSuite. Elk scherm heeft een aantal standaard javascriptfuncties voor het afhandelen van standaardfuncties, zoals het opslaan van gegevens of het bijhouden van wijzigingen. De opbouw van het uiterlijk van een scherm gebeurt veel aan de hand van style sheets. Het heeft veel tijd gekost deze functies goed te leren kennen. Sommige functies waren niet volledig en moesten aangepast worden. Een ander bijkomend probleem was mijn kennis van javascript. Ik had nog weinig tot niet met javascript gewerkt en dat kostte tijd. Dit resulteerde in vertraging die niet gepland was. Ik heb besloten de functies wel aan te passen, omdat meerdere ontwikkelaars daar profijt van hadden en ik zelf de functies optimaal kon benutten voor het Cms. Op het moment dat ik de gebruikersinterface van de ProSuite goed beheerste werkte dit vele malen sneller, maar dat had ik niet kunnen bereiken zonder daar aandacht aan besteed te hebben.

#### *Pilotdeel menustructuur*

De menustructuur zoals deze bestond in pilot Basis Cms was opgezet om een oneindig aantal menu's aan te maken. Dit was mogelijk, maar werd beperkt door problemen met de gebruikersinterface en specifieke functies.

In de ProSuite wordt op verschillende plaatsen gebruik gemaakt van een treeviewcontrol. Dit is een control, net als een textbox, dat gebruikt kan worden voor de weergave van een structuur. Deze treeview is in Windows terug te vinden in de

Windowsverkenner als de mappenstructuur. In afbeelding 15 is het treeviewcontrol te zien, zoals hij gebruikt wordt in het bestandsbeheer in de ProSuite.



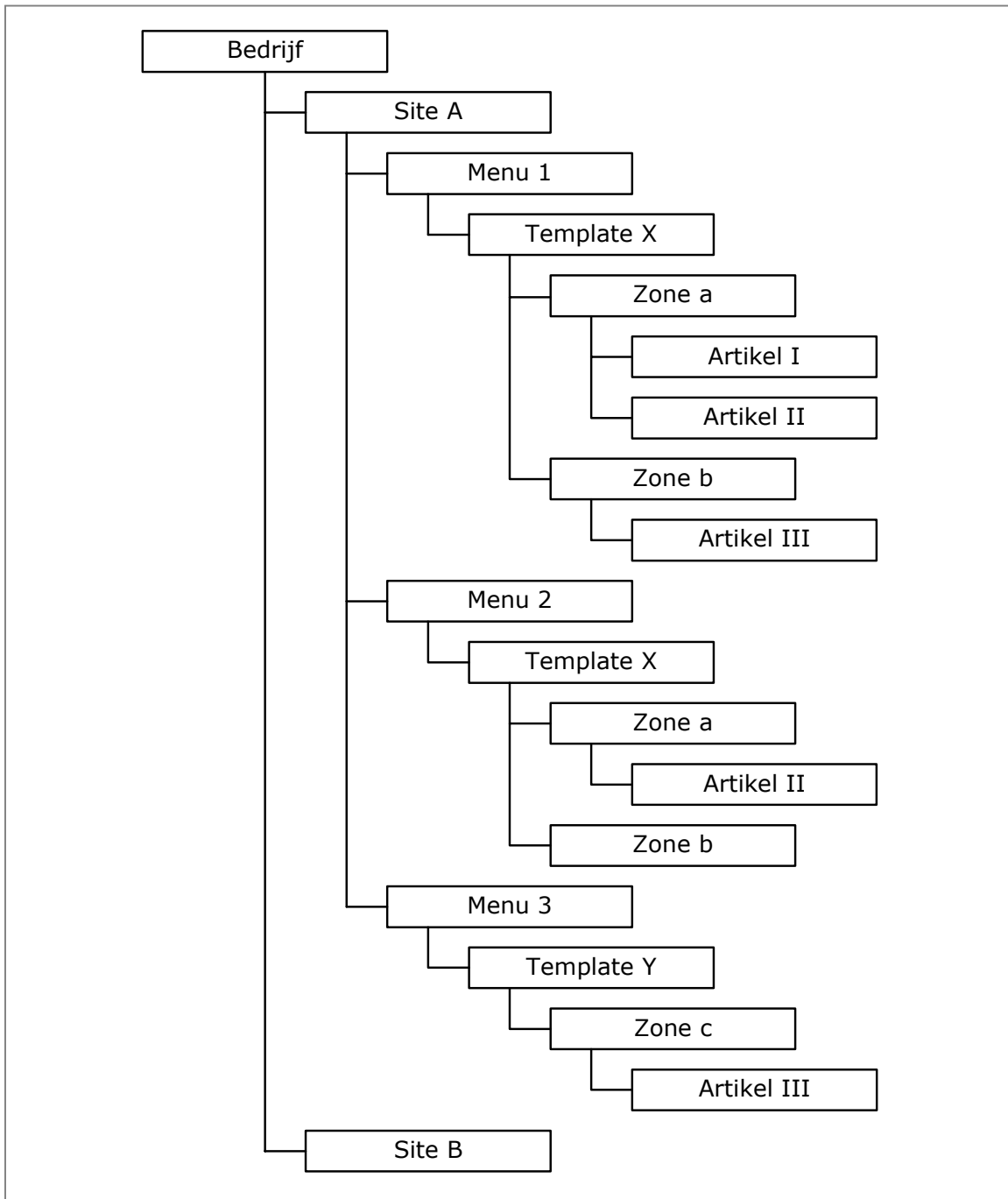
afbeelding 15: Mappenweergave in bestandsbeheer

Het treeviewcontrol biedt de mogelijkheid om een oneindig aantal onderliggende items aan te maken. Omdat menu's ook oneindig aangemaakt moeten kunnen worden is dit een passende oplossing. De treeview is een duidelijke weergave van een structuur en veel mensen zijn bekend met de weergave van het treeviewcontrol. In overleg met de opdrachtgever heb ik besloten het control te gaan gebruiken voor de weergave van de structuur van de website.

De structuur, zoals deze besproken is in pilot 1: Basis Cms paragraaf 7.2.2, voldeed niet meer en op de bestaande structuur is verder ontwikkeld. Dit resulteerde in onderstaande structuur (afbeelding 16). In eerste instantie is er weinig veranderd. Er is een extra koppeling ingekomen. Een menu wordt eerst aan een template gekoppeld en daarna worden aan een zone artikelen toegevoegd.

Essentieel verschil één: Een artikel kon aan één menu gekoppeld worden. In de oude situatie werd door middel van een vreemde sleutel een artikel aan een menu gekoppeld. De nieuwe situatie biedt de mogelijkheid een artikel op meerdere plaatsen te tonen. Dit kan door middel van een koppeltabel.

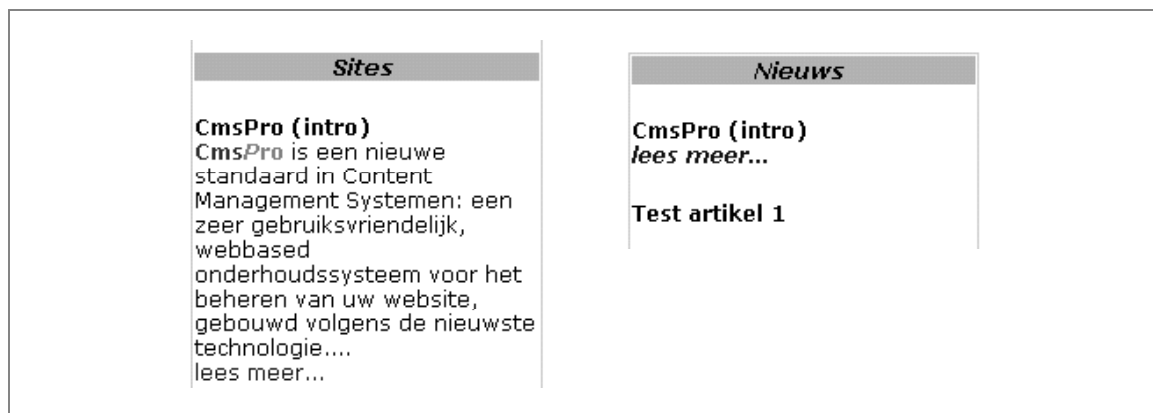
Essentieel verschil twee: Een menu wordt aan een template gekoppeld. Een template is een vooraf gedefinieerd uiterlijk van een menu. Bij een site worden meerdere op maat gemaakte templates geleverd. Een template bepaalt het uiterlijk van een menu. Een template heeft bepaalde zones en per zone kunnen er artikelen, banners, polls, etc worden toegevoegd. Dit is afhankelijk van het type zone.



afbeelding 16: Tweede concept structuur Cms

Een template bepaalt het uiterlijk van een menu. Dit gebeurt in twee fases. Het uiterlijk van een template is fase één. Een template kan bestaan uit een of meerdere modules. De plaatsen van de modules bepalen het uiterlijk van de template. Fase twee: een module heeft een eigen uiterlijk.

Een voorbeeld: Artikelen kunnen op verschillende wijze worden weergegeven. De combinatie van een titel, inhoud met link is mogelijk. Maar alleen titel en link geeft een ander uiterlijk van een module. In afbeelding 17 zijn twee locaties te zien met artikelen, maar andere weergave.

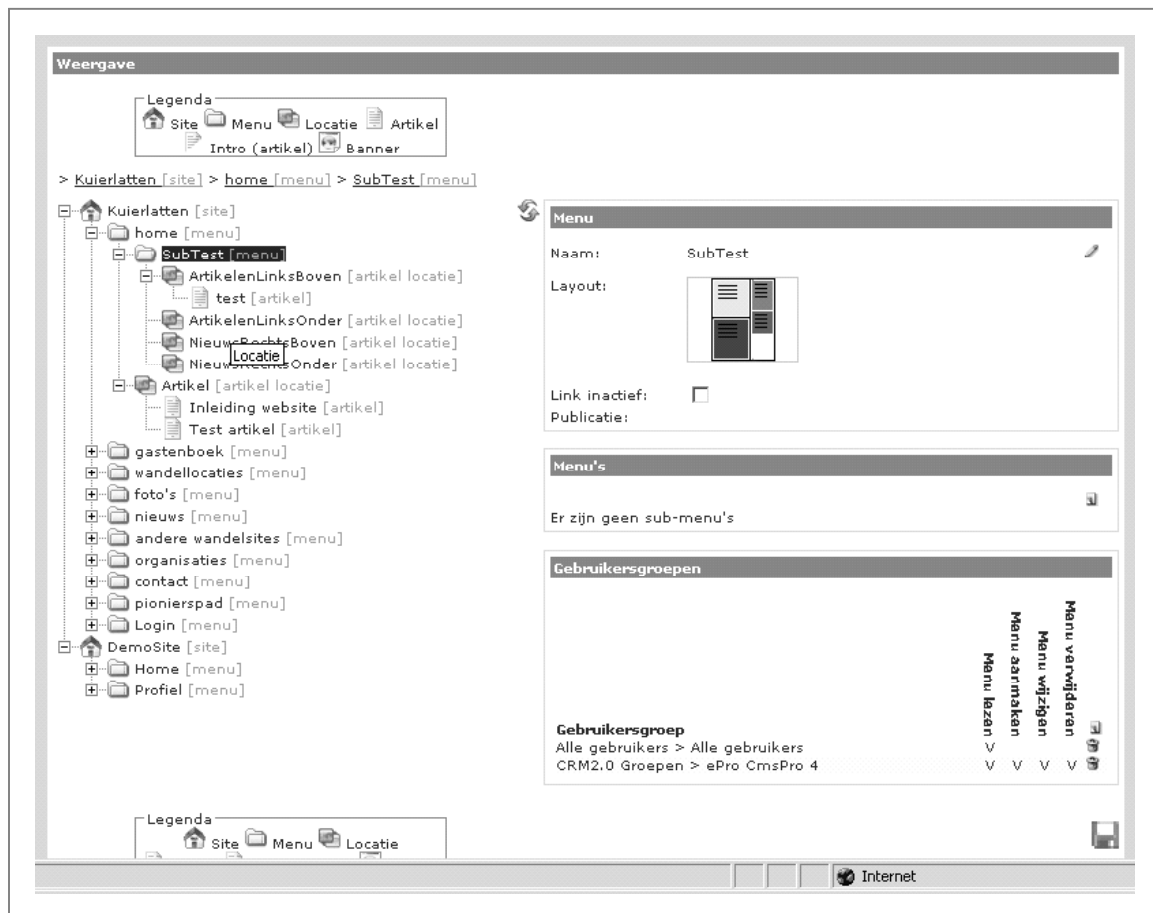


afbeelding 17: Weergave verschillende locaties, maar hetzelfde artikel

Deze opbouw heeft belangrijke veranderingen aangebracht in het Cms. De mogelijkheden van het Cms zijn hiermee stukken vergroot. Het laden van pagina's gaat sneller, omdat alleen die zones geladen worden die op de pagina staan. En niet zoals bij de oude situatie een groot aantal zones dat niet geladen wordt.

Dit alles moet weergegeven worden in de backoffice. Het treeviewcontrol biedt de mogelijkheid deze opbouw systematisch weer te geven. Stapsgewijs wordt de structuur van de site getoond. Het resultaat was dat de treeview werd geladen met een soortgelijk model als afbeelding 15. Aan de linkerzijde van de weergave van de websitestructuur is de treeview te zien. Hier kunnen de verschillende items geselecteerd worden. Aan de rechterkant wordt de bijpassende informatie van het selecteerde item weergegeven.

Nu is het mogelijk de gehele site te beheren vanuit deze structuurweergave van de website. Met het in gebruik nemen van deze treeview is de beheersbaarheid van de website verbeterd. Hieronder staat afbeelding 18 van de weergave van de structuur.



afbeelding 18: Weergave menustructuur van CmsPro in ProSuite

### *Pilotdeel gebruikers- en gebruikersgroepenmodel*

Het gebruikers- en gebruikersgroepenmodel in de ProSuite was al bekend bij mij. Dit omdat ik het model al tijdens de pilot Gebruikers had onderzocht. Dit zorgde voor snelle vooruitgang. In de ProSuite zaten al schermen waar gebruikersgroepen gekozen konden worden en met kleine aanpassingen waren deze schermen voor het Cms te gebruiken.

Het proces van gebruikers toevoegen aan sites, menu's, artikelen, etc was niet veranderd ten opzichte van de pilot Gebruikers. Hierdoor was de ontwikkeling van dit pilotdeel snel doorlopen.

### **9.3 Invoering**

#### *Invoeren*

Deze pilot is via veel tussentijdse terugkoppeling binnen eProjects ingevoerd. Er zijn verschillende testsites geweest voor deze pilot. Dit resulteerde in feedback van de afdeling sales wat verwerkt is in de pilot.

#### *Pilotacceptatie*

Bij het naderen van het eind van de pilot is er veel contact geweest met de webdesigners. Deze hebben test websites ontwikkeld voor klanten. Door de samenwerking met de webdesigners hebben zij een langzame introductie van het Cms gehad.

#### *Feedback*

Na implementatie van de pilot Integratie ProSuite waren uitgebreide mogelijkheden met het Cms. De mogelijkheden waren zelfs zo uitgebreid dat dit leidde tot een onduidelijke gebruikersinterface. Gebruikers die niet bekend met het systeem waren, hadden een uitgebreide beschrijving nodig voordat zij de werking snaptten. Dit probleem is op te lossen door een stapsgewijze 'wizard' in te bouwen voor het aanmaken van verschillende onderdelen. Dit is niet verwerkt in het systeem vanwege tijdsgebrek.



## **10 Pilot 4: Modules en modulaire opbouw**

Deze pilot zorgt voor de uitbreidingen van het Cms. Het doel van de pilot is het ontwikkelen van nieuwe elementen/ modules voor het Cms. Deze pilot stond in de planning als laatste pilot. Dit omdat het aantal te ontwikkelen modules in theorie oneindig is.

Tijdens de ontwikkeling van het Cms is het systeem op verschillende plaatsen in gebruik genomen. Op een aantal plaatsen waren artikelen alleen niet voldoende en moesten er modules ontwikkeld worden voordat deze pilot gestart werd.

Voorbeelden zijn bannermodule, pollmodule, kaartmanager. De koppelingen die een artikel heeft met het systeem hebben alle modules. Het maken van een module is op zich losstaand, want het heeft maar een paar koppelingen met het systeem. De modules zijn deels door mezelf , deels door anderen ontwikkeld.

### **10.1 Definitiestudie**

#### **10.1.1Systeemeisen**

De modules die ontwikkeld moesten worden kwamen voort uit de vraag van klanten. De modules die ontwikkeld zijn voor het Cms zijn bannermodule, pollmodule en kaartmanager.

De bannermodule maakt het mogelijk banners (reclame afbeeldingen) aan te maken. De eisen voor de bannermodule zijn door de afdeling sales voornamelijk gedefinieerd. Ik heb eerst een opzet gemaakt en geïmplementeerd in het Cms, daarna zijn er opties toegevoegd. Het voordeel hiervan is dat de handelingen voor het aanmaken, wijzigen en verwijderen zichtbaar en tastbaar zijn. Dit is duidelijker voor de gebruiker en voor de afdeling Sales. Ik heb gemerkt dat je door een klein stuk van te voren te ontwikkelen betere eisen kan definiëren tijdens een gesprek.

De systeemeisen van de bannermodule:

- Banner aanmaken
- Banner wijzigen
- Banner verwijderen
- Een banner moet uit een plaatje bestaan

- Een banner kan een link bevatten

De andere modules zijn ontwikkeld door collega's. Zij hebben zelf de systeemeisen van de modules bepaald. Ik heb nauw samengewerkt met hen om een feilloze aansluiting met het Cms te creëren. Ik bespreek in dit hoofdstuk alleen de bannermodule.

Tijdens mijn afstudeerperiode heb ik een gesprek gehad met het Erasmus Magazine. Het is de redactie van een magazine dat op de Erasmus in Rotterdam wordt verspreid. Zij wilden het magazine niet alleen als hardcopy aanbieden, maar ook online gaan uitbrengen dit vormde een nieuwe functionele eis.

Op een redactie worden verschillende artikelen geschreven door journalisten. Zij werken aan een artikel en als het af is kan het gepubliceerd worden. Voor publicatie zal het artikel gecontroleerd moeten worden. Na goedkeuring kan het artikel in het magazine geplaatst worden.

Als er naar dit voorbeeld wordt gekeken dan maakt het artikel een aantal fases door. Aan een fase kan een status worden gekoppeld. In bovenstaand voorbeeld zouden de fases (in chronologische volgorde) kunnen zijn: Nieuw, Te publiceren, Goedgekeurd, Online, Archief. Door een status te koppelen aan een gebruikersgroep kan het bovenstaande voorbeeld in de praktijk worden toegepast.

#### Systeemeisen voor status

- Status aanmaken
- Status wijzigen
- Status verwijderen
- Gebruikersgroep toevoegen aan status
- Status samenvoegen met bestaande modules (artikelen, banner)

### 10.1.2Systeemconcept

#### *Klassendiagram*

Het klassendiagram is uitgebreid met een statusklasse en bannerklasse.

De bannerklasse moet toegevoegd worden aan het klassendiagram. Dit is de eerste module die ontwikkeld word als module. Hierna zullen nog vele volgen en dus is het van belang dat de implementatie van de bannermodule eenvoudig moet verlopen. Want dan is dit ook van toepassing op toevoegen van nieuwe modules. De bannerklasse heeft dezelfde koppelingen als de artikelklasse met andere klassen in het diagram. Bij het toevoegen van nieuwe modules zullen deze ook dezelfde koppelingen hebben als artikel en banner. Er zullen drie koppelingen gemaakt moeten worden. Die met site, status en locatie.

De statusklasse heeft enkele attributen: een naam, volgorde en de gekoppelde gebruikersgroepen. De verschillende modules krijgen ieder een attribuut erbij. Dit is het attribuut status. Een artikel kan dan van een status worden voorzien.

#### *Use-cases*

De Use-cases zijn niet gemaakt voor deze pilot. Er was genoeg inzicht in de werking van de klassen. Het klassendiagram was voldoende voor de ontwikkeling van de pilot.

#### *Sequentiediagrammen*

Hetzelfde als de use-cases zijn de sequentiediagrammen niet gemaakt.

## 10.2 Pilotontwikkeling

### 10.2.1 Pilotplan

De pilot is opgedeeld in twee delen. De status vormt pilotdeel één en banner is het tweede pilotdeel. Ik heb beide pilotdelen opgedeeld in bouweenheden.

- Pilotdeel Banner
  - Aanmaken database tabellen
  - Banners overzicht
  - Banner aanmaken, wijzigen en verwijderen
  - Interface maken voor frontoffice
- Pilotdeel Status
  - Aanmaken database tabellen
  - Status overzicht
  - Status aanmaken, wijzigen en verwijderen
  - Gebruikersgroepen aan status koppelingen
  - Status verwerken in aanwezige modules

Pilotdeel banner is ontwikkeld tijdens pilot Gebruikers. Omdat dit wenselijk was voor oplevering van de website voor EFR. Hierdoor is de chronologische volgorde van deze pilot niet kloppend.

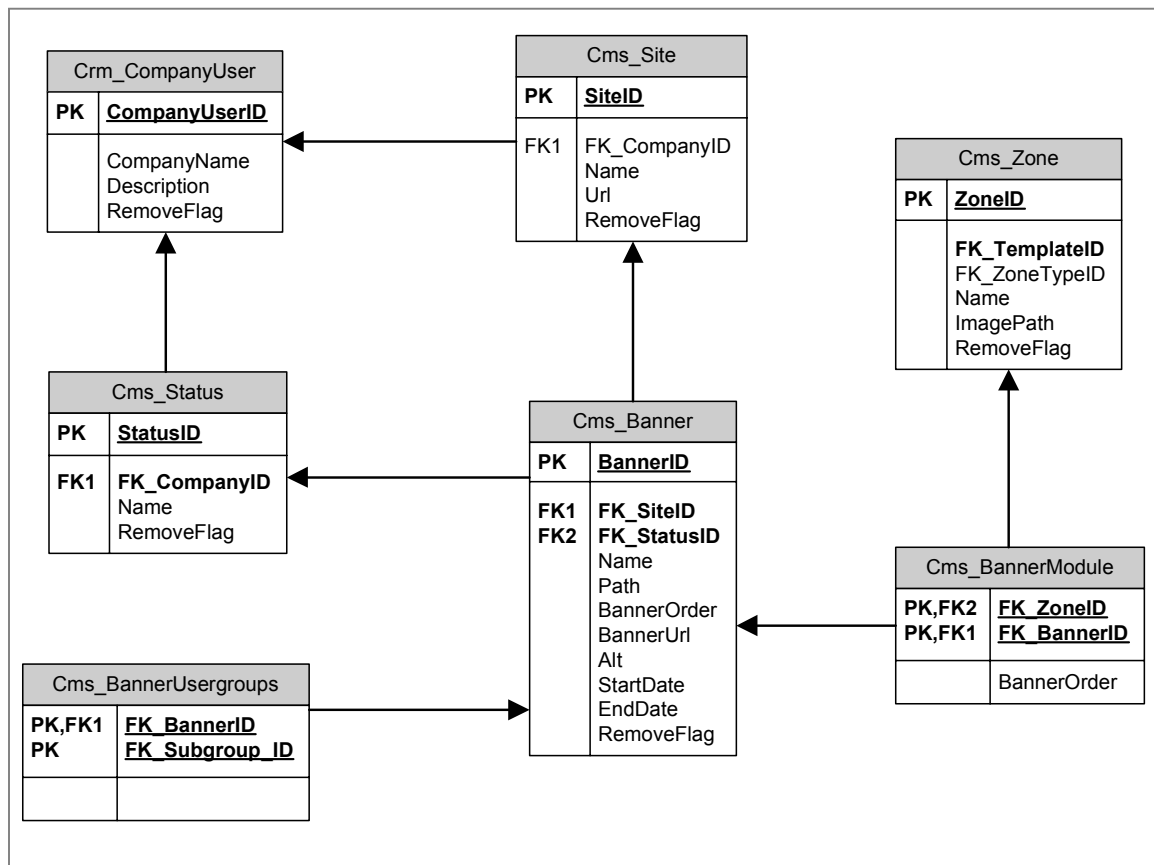
### 10.2.2 Bouw softwareprogrammeereenheden

#### *Pilotdeel Banner*

Het maken van een module als uitbreiding van het bestaande systeem is een losstaand geheel in eerste instantie. Pas als er koppelingen worden gemaakt wordt het een onderdeel van het Cms.

Ik heb ervoor gekozen om de bannermodule direct in de ProSuite te maken. Na het opzetten van de tabellen en klassen kon een overzicht van banners weergegeven worden. Vanuit het overzicht ben ik gaan uitbreiden met de functies voor het aanmaken, wijzigen en verwijderen van banners.

Het Cms is zo ontwikkeld dat het eenvoudig moet zijn om modules toe te voegen. Een module wordt door koppelingen in het Cms toegevoegd. Deze koppelingen bestaan uit koppelingen met de site, status, gebruikersgroepen en locatie.



afbeelding 19: Databasediagram koppelingen van tabel banner

De tabellen Cms\_Banner, Cms\_BannerUsergroups en Cms\_BannerModule zijn de tabellen die nodig zijn voor de koppelingen van een module in het Cms. Cms\_Banner is de tabel die de informatie van een banner opslaat. Cms\_BannerModule bepaalt de locatie van een banner in de website. Een zone bevindt zich op een pagina en in de zone worden de betreffende banners geladen. Cms\_BannerUsergroups zorgt voor de koppeling van de gebruikersgroepen aan een banner.

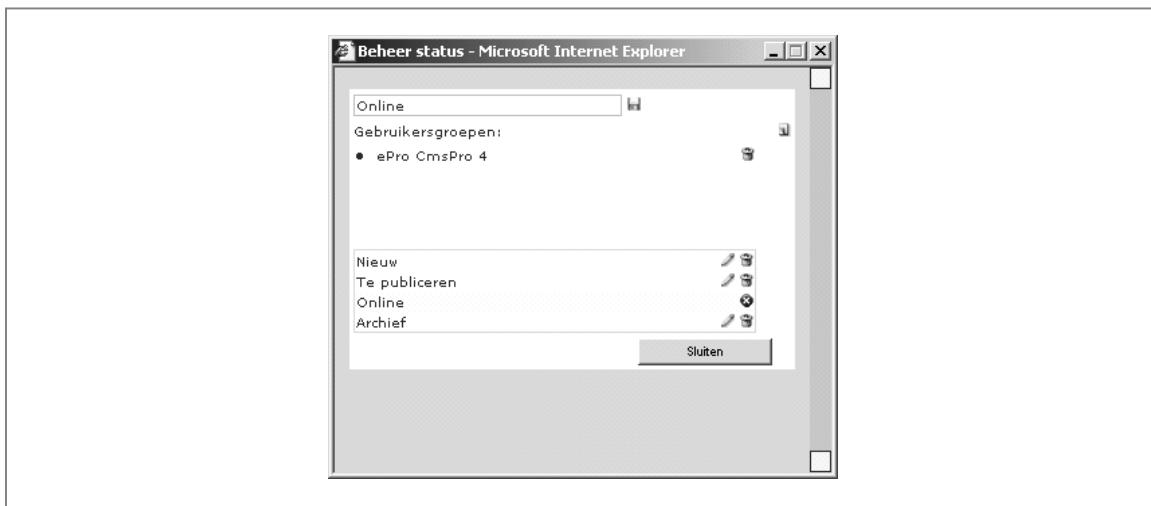
Het ontwikkelen van banners bracht geen problemen met zich mee. Tijdens de ontwikkeling van het beheren van artikelen zijn mogelijke problemen al een keer

naar voren gekomen. Het ontwikkelen van de module Banners en de module Artikelen is vrijwel hetzelfde.

### *Pilotdeel Status*

De status is alleen afhankelijk van een bedrijf. Een bedrijf kan meerdere statussen hebben. De statussen worden eenmalig aangemaakt, daarna zullen enkel kleine veranderingen optreden. Omdat dit op beheerdersniveau gaat zal de status onder de verantwoordelijkheid van de beheerder vallen.

In afbeelding 19 met de tabellen voor banner is de status tabel te zien. De status is direct aan een module gekoppeld. Dit geldt voor artikelen, banners en alle opvolgende modules.



afbeelding 20: Scherm status beheer

Status kan een extra laag inbrengen in het beheer van de website. De status hoeft niet gebruikt te worden, maar kan beperkingen opleggen aan gebruikersgroepen. Bij het juist toepassen van het gebruikersmodel in combinatie met de status wordt het een geavanceerd Cms. De vraag van de redactie van Erasmus Magazine is een goed voorbeeld van dit gebruik.

### **10.3 Invoering**

#### *Invoeren*

Invoering voor deze pilot is geleidelijk gebeurd. De updates die tijdens de ontwikkelingen zijn doorgevoerd hebben geleid tot invoering van deze pilot. Updates zijn pas doorgevoerd als deze getest waren op mijn lokale computer.

#### *Pilotacceptatie*

Deze pilot heeft weinig impact gehad op het Cms. Het waren uitbreidingen van het bestaande systeem. De uitbreidingen behoeven geen uitgebreide handleidingen. Ik heb besloten om alleen de aanpassing te vermelden binnen eProjects aan ontwikkelaars en sales. Zij hebben dit ervaren als een verbetering van het bestaande Cms en dit heeft geen gevolgen gehad.

#### *Feedback*

Tot nu toe is er geen feedback van klanten voor deze pilot. De status is in het Cms geïmplementeerd, maar de website voor het Erasmus Magazine is nog niet opgeleverd. Dus de feedback voor status zal op een later moment pas beschikbaar zijn.

## 11 Evaluatie afstuderen

### 11.1 Productevaluatie

De doelstelling van afstudeeropdracht het ontwikkelen van CmsPro 4.0 in de website-ontwikkelomgeving ASP.NET is behaald. Het resultaat is een werkende applicatie voor het beheer van een website.

De structuur van het Cms is uitgebreid onderzocht en besproken. Hierdoor heeft een gebruiker grote flexibiliteit in het uiterlijk van de website. De gebruiker kan een aantal templates definiëren die door eProjects gemaakt worden. Mocht na verloop van tijd behoefte zijn naar een extra template dan is deze in een zeer korte tijd te ontwikkelen. De menustructuur is dynamisch op te bouwen en de mogelijkheid voor het oneindig aanmaken van menu's biedt een uitgebreide navigatie voor grote websites. De structuurweergave van de website (afbeelding 18: Weergave menustructuur van CmsPro in ProSuite) geeft de gebruiker een totaaloverzicht op één scherm. Vanuit dit scherm kan de gehele website beheerd worden. Dit is een groot voordeel ten opzichte van andere Cms-applicaties.

Het Cms zal uitgebreid moeten worden met verschillende modules. Maar de eenvoud waarmee modules gekoppeld kunnen worden met het Cms biedt veel mogelijkheden voor snelle implementatie.

Het Cms kan niet alleen gebruikt worden voor simpel beheer, maar het is ook mogelijk meerdere gebruikers delen van websites te laten onderhouden. Door het rechten systeem zijn de mogelijkheden per gebruikersgroep te beheren. Door integratie met de ProSuite worden gebruikers en -groepen niet apart beheerd, maar gebeurt dit centraal in de ProSuite.

Algemene voordelen van een Cms:

- Onderhoud van meerdere websites in een Cms.
- Onderhoud en 'updaten' van webpagina's is makkelijk en snel uit te voeren, gewoon via de webbrowser.
- Geen kennis van HTML nodig.
- Iedere pagina wordt consequent, volgens vooraf vastgestelde criteria opgebouwd aan de hand van sjablonen (templates).



- Pagina's zijn systematisch gerangschikt op hiërarchische wijze, zoals bestanden en mappen in een bestandssysteem.
- Vrijwel nooit meer gebroken links.
- Dure design- en programmeeruren zijn niet meer nodig na invoering van het Cms.

## 11.2 Procesevaluatie

Tijdens de afstudeerperiode heb ik IAD als methode gebruikt. De methode was nieuw voor mij, maar door zelfstudie heb ik veel geleerd en IAD kunnen toe passen. In het begin van de afstudeerperiode was onduidelijkheid met het kiezen van een passend scenario. Nu terugkijkend is het scenario een goede keus geweest voor het ontwikkelproces van het Cms. Tussentijdse producten (afgeronde pilots) zijn ingevoerd en in gebruik genomen door verschillende klanten. Feedback leverde input voor de volgende pilot.

Naast feedback van de klanten heb ik veel feedback gekregen van afdeling Sales en de projectgroep. Het opzetten van de projectgroep was voornamelijk in het begin van de afstudeerperiode een toegevoegde waarde. Door eigen ideeën te bespreken met de projectgroep ontstaat er een compleet beeld. In een later stadium ben ik de projectgroep minder gaan gebruiken, maar heb dit verlegd naar direct contact met ontwikkelaars. Dit vanwege het technische karakter van de problemen. Dit heb ik als prettig ervaren. Meeste problemen hadden een verloop van enkele dagen, omdat een aantal programmeurs parttimer is. Door het meerdere malen bekijken van een probleem kan een volledig beeld van alle aspecten gevormd worden en kunnen weloverwogen beslissingen worden genomen. Een voorbeeld hiervan is de ontwikkeling van de structuur van het Cms.

Feedback van de afdeling Sales had voornamelijk betrekking op de gebruikersinterface. De feedback kwam voornamelijk na de ontwikkeling van functies. Hierdoor moesten achteraf vaak aanpassingen gemaakt worden. Ik heb dit deels kunnen voorkomen door vooraf de ontwikkeling een simpele gebruikersinterface te maken van de te ontwikkelen functie. Hierdoor was er meer inzicht en aanpassingen achteraf namen af.

De planning zoals deze in het begin is opgesteld wijkt af met het daadwerkelijke proces. Tijdens pilot 1: Basis Cms is er veel tijd gestoken in de ontwikkeling van de menustructuur en de weergave hiervan. De beslissing om de opbouw van het menu te beperken naar 3 niveaus heeft gezorgd voor voortgang van het proces. Hieruit heb ik geleerd niet te diep in het technische probleem te duiken, maar het probleem opnieuw te bekijken en waar mogelijk in overeenstemming een andere oplossing toe te passen.

De ontwikkeling van de structuur van de opbouw van het Cms is een langdurig proces geweest. Door de vele gesprekken met verschillende programmeurs en gebruikers heeft dit geresulteerd in een robuuste oplossing. Er is gekozen om deze structuur tijdens pilot 3: Integratie ProSuite in te bouwen. Door de veranderingen van structuur, uiterlijk ProSuite en gebruikers- en gebruikersgroepenmodel samen te ontwikkelen tijdens pilot 3 heeft veel tijdwinst opgeleverd. De vooruitgang van het Cms pilot 1 en 2 naar het Cms pilot 3 was hierdoor groot.

Mijn kennis van .NET is weer vergroot en heb veel geleerd van IAD. Het is jammer dat de methode IAD niet werd aangeboden tijdens mijn schooljaren. Deze methode zal meer voor de hand liggen voor objectgeoriënteerde applicaties dan een lineaire methode. Ik hoop komende tijd verder te werken met .NET en hier meer kennis van op te doen. Naar mijn idee is voor .NET een lange toekomst weggelegd door zijn mogelijkheden voor het internet. Zoals in de inleiding aangehaald is, is internet meer dan alleen die folder van producten.



## **Verklarende woordenlijst**

### **Backoffice:**

De backoffice is de beheerderskant van het Cms. Dit is de 'achterkant' van een website.

### **Frontoffice:**

Dit is de voorkant van een website. Het ontwerp dat beheer wordt in de backoffice.

### **Cms – Content Management Systeem:**

Een Cms is een gebruiksvriendelijk, webbased onderhoudssysteem voor het beheren van een website. Het enige wat een gebruiker nodig heeft is een webbrowser (zoals Internet Explorer) en toegang tot internet.

### **CmsPro:**

CmsPro is een van de applicaties die eProjects ontwikkelt. Het is een Content Management Systeem waarmee websites onderhouden kunnen worden. Van dit CmsPro zijn inmiddels meerdere versies ontwikkeld. (vb. versies 2.5 en 3.0)

### **Crm – Customers Relationship Management**

Dit is een relatiebeheerpakket. In een relatiebeheerpakket worden relaties van een bedrijf opgeslagen.

### **MoSCoW-principe:**

Must have = vereiste functies;

Should have = gewenste functies;

Could have = luxe functies;

Would have = extra functionaliteit indien alle overige zaken zijn gerealiseerd.

Opgeslagen procedure:

Een opgeslagen procedure is een term die vanuit de SQL-server van Microsoft komt. De opgeslagen procedure is te vergelijken met een query. Binnen SQL-server kan je meerdere opgeslagen procedures maken. Hieronder staat een voorbeeld:

```
Create stored procedure dbo.Cms_ArticleSelectAll()  
AS  
SELECT *  
FROM dbo.Cms_Article
```

ProSuite:

De ProSuite is een van de producten van eProjects. Het is een totaal pakket van de verschillen producten. ProSuite maakt gezamenlijk gebruik van documenten, een centrale agenda en een relatiebeheersysteem mogelijk. ProSuite bestaat uit een relatiebeheersysteem, boekhoudpakket, Cms, webshop, human resource management, ProcesPro, ServicePro.

Stored Procedure:

Zie opgeslagen procedure.

## Bronnen

### Boeken

IAD; Het evolutionair ontwikkelen van informatiesystemen

R.J.H. Tolido

ISBN: 90-395-0401-6, Academic Service, eerste druk, oplage 6, 2003

Practische UML 2<sup>de</sup> editie

Jos Warmer & Anneke Kleppe

ISBN: 90-430-0494-4, Addison Wesley, 2001

Fundamentals of Software Engineering

Carlo Ghezzi, Mehdi Jazayeri & Dino Mandriolo

ISBN: 0-12-820432-2, prentice-Hall Inc., 1991

### Web

Onderzoek voor interface:

Naam	Url
Advanced hosting 4 u	<a href="http://www.advancedhosting4u.net">www.advancedhosting4u.net</a>
De volkskrant	<a href="http://www.volkskrant.nl">www.volkskrant.nl</a>
Ilse	<a href="http://www.else.nl">www.else.nl</a>
Portal B.V.	<a href="http://www.portal.com">www.portal.com</a>
Memisa	<a href="http://www.memisa.nl">www.memisa.nl</a>
Google zoekmachine	<a href="http://www.google.nl">www.google.nl</a>
Download.com	<a href="http://www.download.com">www.download.com</a>
Startpagina	<a href="http://www.startpagina.nl">www.startpagina.nl</a>
KPN Telecom	<a href="http://www.kpn.nl">www.kpn.nl</a>
Voelspriet	<a href="http://www.voelspriet.nl">www.voelspriet.nl</a>
Geurtzen	<a href="http://www.geurtzen.nl">www.geurtzen.nl</a>

Overige sites:

Naam	Url
Microsoft Developers Network	<a href="http://msdn.microsoft.com">msdn.microsoft.com</a>
DotNetNuke	<a href="http://www.dotnetnuke.com">www.dotnetnuke.com</a>
Asp.NET	<a href="http://www.asp.net">www.asp.net</a>
Typo	<a href="http://www.typo.com">www.typo.com</a>

## Overig

Help documentatie van Visual Studio .NET

## Interne bijlage

- Opdrachtschrijving
- Plan van aanpak
- Gedetailleerde planning (Excel)
- Pilotplan
- Pilotontwikkelplan pilot Basis Cms
- Pilotontwikkelplan pilot Gebruikers
- Pilotontwikkelplan pilot Integratie ProSuite
- Pilotontwikkelplan pilot Modules en modulaire opbouw
- Testrapport [Rapporten besprekingen, 15 pag. zie hieronder]
- Excel-bestand met systeemeisen?
- Excel-bestand met onderzoek naar element (tabblad interface frontoffice)

## Externe bijlage

- CmsPro versie 4.0
- Beheerderhandleiding (t.b.v. de implementatie van het Cms)
- Rapporten besprekingen [15 pag.]

Eindverslag: **Bijlagen**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

Afstudeerder:

Bram de Jager

20006776

Bedrijf:

eProjects B.V.

Rotterdam

Examinatoren:

dhr. B. van Strien

dhr. H.M.C. Lachman

Afstudeerperiode:

2004-1.1

9 februari 2004 t/m 11 juni 2004





## **Inhoudsopgave bijlagen**

Opdrachtschrijving  
Plan van aanpak  
Gedetailleerde planning  
Definitie van eisen  
Excel onderzoek elementen  
Pilotplan  
Ontwerpdocumentatie  
Rapporten besprekingen  
Beheerdershandleiding



## Opdrachtschrijving afstudeeropdracht CmsPro 4.0

Afstudeerblok: 2004-1.1

Studentnummer: 20006776

Afstudeerrichting: OSTI

Achternaam: Jager

Voorletters: B

Roepnaam: Bram

Adres: Van Maanenkaade 199

Postcode: 2516 LP

Woonplaats: Den Haag

Telefoon: 06 - 41 49 50 15

Naam bedrijf: eProjects BV

Afdeling bedrijf: Softwareontwikkeling

Bezoekadres bedrijf: Meent 93

Postcode bezoekadres: 3011 JG

Postbusnummer:

Postcode postbusnummer:

Plaats: Rotterdam

Telefoon bedrijf: +31 (0)10 240 96 50

Telefax bedrijf: +31 (0)10 213 68 71

Achternaam bedrijfsmentor1: Dolle

Voorletters bedrijfsmentor1: R.R.

Titulatuur bedrijfsmentor1: drs.

Doorkiesnummer bedrijfsmentor1:

Achternaam bedrijfsmentor2: Roos

Voorletters bedrijfsmentor2: D.

Titulatuur bedrijfsmentor2: drs.

Doorkiesnummer bedrijfsmentor2:



Doorkiesnummer afstudeerder:

Titel afstudeeropdracht: CmsPro 4.0

## Inhoudsopgave

Inhoudsopgave .....	i
Omschrijving afstudeeropdracht.....	4
Inleiding (organisatorische omgeving, kader, historie) .....	4
Aanleiding afstudeeropdracht.....	4
Doelstelling van de opdracht.....	4
Uitgangssituatie .....	5
Benodigde software .....	5
Benodigde hardware .....	5
Beschikbare rapporten .....	5
Aanwezige ideeën.....	5
Concrete werkzaamheden .....	6
Uit te voeren activiteiten .....	6
Te hanteren methoden .....	7
Te gebruiken technieken .....	7
Planning .....	8
Te vermelden nadrukken .....	8
Resultaten voor de opdrachtgever (op te leveren producten) .....	8
Verantwoording .....	8

## **Omschrijving afstudeeropdracht**

Ontwikkeling van een nieuwe versie van het huidige Content Management Systeem (CMS) bij eProjects (CmsPro 4.0).

## **Inleiding (organisatorische omgeving, kader, historie)**

eProjects Advanced Software is een in Rotterdam gevestigd ICT bedrijf. Het is opgericht in 1999 en houdt zich bezig met e-commerce en e-business projecten voor het midden- en grootbedrijf. eProjects telt op het moment 25 medewerkers en er zijn diverse projecten aan de gang, waaronder een web-based Customer Relationship Management (CRM) systeem, een financieel pakket en een webshop.

Naast generieke projecten voor organisaties als KPMG en Memisa, richt eProjects zich ook op een aantal specifieke branches, zoals die van de verhuizer of beveiligingsbedrijven.

## **Aanleiding afstudeeropdracht**

Het huidige CmsPro versie 3.0 van eProjects is een "standaard" CMS. Het product heeft alle basisfuncties van een CMS, maar eProjects wil de afzetmarkt vergroten van CmsPro.

## **Doelstelling van de opdracht**

Het ontwikkelen van CmsPro 4.0 in de website-ontwikkelomgeving ASP.NET.

## **Uitgangssituatie**

### **Benodigde software**

Microsoft Visual Studio .NET;  
Microsoft Developer Network (MSDN)  
Microsoft Visio;  
Microsoft Internet Information Services (IIS);  
Microsoft SQL Server;  
Microsoft Access;  
Microsoft Word;  
Microsoft .NET-framework;  
Microsoft Web Matrix.

### **Benodigde hardware**

Een workstation en een testserver

### **Beschikbare rapporten**

-

### **Aanwezige ideeën**

Basis van CmsPro 4.0

Vanuit de functionaliteiten van CmsPro 3.0, welke als basis dient voor CmsPro 4.0, zal er een inventarisatie worden gemaakt voor de functionaliteiten van CmsPro 4.0.

### ProSuite

De ProSuite wordt een pakket waarin meerdere modules/ applicaties zijn opgenomen. De suite is een overkoepelende applicatie die zorgdraagt voor de koppeling van de verschillende modules/ applicaties. Een voorbeeld is een relatie in het CRM welke ook bekend is in de boekhoudapplicatie AccountPro. Het CMS moet ook onderdeel van de ProSuite worden.

### Dynamische content/ automatische content

Er zal onderzocht worden wat de mogelijkheden zijn met betrekking tot automatische content. In de praktijk blijkt dat sites met CMS-systemen niet met regelmatige terugkeer worden upgedate. Terwijl een CMS het juist mogelijk maakt een site eenvoudig veranderen wordt dit niet regelmatig gedaan. Waardoor veranderingen aan de site minimaal zijn. Om nu als nog te zorgen dat de inhoud van de site wijzigd is automatische content een oplossing.

Automatische content zal zelf content genereren a.d.h.v. ingevoerde gegevens die hij elders vergaart. Deze mogelijkheid zal worden uitgediept en beschreven in een kort verslag. Dit idee heeft lage prioriteit en zal uitgevoerd worden als de tijd hiervoor beschikbaar is.

## Concrete werkzaamheden

### **Uit te voeren activiteiten**

Vorbereiding

Maken plan van aanpak;

Bestuderen CmsPro 3.0;

Vorbereiden vraaggesprek om gewijzigde eisen te ontdekken;

Vraaggesprek houden;

Vraaggesprek uitwerken

#### Definitiestudie

Functionele eisen opstellen;  
Klassediagram maken;  
Use-cases maken;  
Use-case-diagram maken

#### Applicatieomschrijving

Pilotplan opstellen;  
Pilotontwikkelplan opstellen;  
Pilotontwerp verschillende pilots maken (zie hoofdstuk Pilots)

#### Ontwikkeling CmsPro 4.0

Database aanmaken, inclusief tabellen;  
Schermenlay-out;  
Functionaliteit per scherm implementeren;  
Testen op functionaliteit

### **Te hanteren methoden**

IAD

### **Te gebruiken technieken**

UML

Interviewtechnieken (Deze techniek worden gebruikt voor de definitiestudie)  
Klantgesprekken (Deze techniek worden gebruikt voor de definitiestudie)  
Enquetes (Deze techniek worden gebruikt voor de definitiestudie)



## **Planning**

Vorbereitung - 2 weken

Definiestudie - 4 weken

Applicatieontwikkeling - 9 weken

Verslag - 3 weken

## **Te vermelden nadrukken**

-

## **Resultaten voor de opdrachtgever (op te leveren producten)**

Definiestudie;

Ontwerpdokumentatie;

Beheerdershandleiding (t.b.v. de implementatie van het CMS);

Testrapport;

CmsPro 4.0

## **Verantwoording**

SO-08 Objectgeoriënteerde systeemontwikkeling

CmsPro zal worden ontworpen met behulp van UML. Tevens wordt het pakket geprogrammeerd in een objectgeoriënteerde taal in .NET.

SO-07 Methoden en technieken van systeemontwikkeling

Met de kennis van methoden en technieken kan er gekeken worden naar de beste techniek voor het ontwerpen van de applicatie.

DB-02 Fysiek database-ontwerp *en* DB-05 Concurrency en recovery

De applicatie CmsPro zal gaan werken met een database. De database zal naar behoefte uitgebreid worden voor de applicatie. Kennis van database-ontwerp is een vereiste.

SW-10 Software-ontwikkeling in een objectgeoriënteerde omgeving

CmsPro zal worden ontworpen met behulp van UML. Tevens wordt het pakket geprogrammeerd in een objectgeoriënteerde taal in .NET.

Bijlage: **Plan van aanpak**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

## Inhoudsopgave

Inhoudsopgave .....	ii
Inleiding .....	1
Opdracht.....	2
Aanleiding opdracht.....	2
Doelstelling van de opdracht.....	2
Doelgroep van de opdracht.....	2
Standaarden, richtlijnen en procedures .....	3
Documentatie .....	3
Gegevensbank .....	4
Programmeren .....	4
Versiebeheer.....	5
Taal .....	5
Aanpak .....	6
Projectorganisatie.....	6
Te gebruiken methoden en technieken .....	7
Methoden .....	7
Technieken .....	7
Risico's .....	8
Planning.....	8
Implementatie .....	9
Kwaliteit.....	9
Afsluiting.....	10



## Inleiding

eProjects Advanced Software is een in Rotterdam gevestigd ICT bedrijf. Het is opgericht in 1999 en houdt zich bezig met e-commerce en e-business projecten voor het midden- en grootbedrijf. eProjects telt op het moment 25 medewerkers en er lopen diverse projecten, waaronder een web-based Customer Relationship Management (CRM) systeem, een financieel pakket en een webshop.

Naast generieke projecten voor organisaties als KPMG en Memisa, richt eProjects zich ook op een aantal specifieke branches, zoals die van de verhuizer of beveiligingsbedrijven.

Een van de producten van eProjects is CmsPro. Dit is een Content Management Systeem (CMS). Een CMS is een tool die "achter een website wordt geplakt". Met een CMS kan de eindgebruiker op eenvoudige wijze een complexe en beheersbare website maken en beheren. Enkele voorbeeld functies zijn:

- Menustructuur wijzigen
- Gemakkelijke lay-outwijzigingen door te voeren
- Documenten beheren



## **Opdracht**

### **Aanleiding opdracht**

Het huidige CmsPro versie 3.0 van eProjects is een "standaard" CMS. Het product heeft alle basisfuncties van een CMS, maar eProjects wil de afzetmarkt vergroten van CmsPro.

### **Doelstelling van de opdracht**

Het ontwikkelen van CmsPro 4.0 in de website-ontwikkelomgeving ASP.NET.

### **Doelgroep van de opdracht**

De opdrachtgever is eProjects, maar dit is niet de doelgroep van het CMS. Het CMS moet door iedereen te beheren zijn. De gebruikers van het systeem zullen vanuit allerlei verschillende achtergronden komen. Dus de diversiteit van de gebruikers is groot.

Kortom de doelgroep strekt zich van de huisvrouw, die voor haar vrijwilligerswerk de website moet onderhouden, tot de ICT-beheerder van een wereldwijde organisatie.



## **Standaarden, richtlijnen en procedures**

### **Documentatie**

Voor documentatie zal gebruik worden gemaakt van Microsoft Office. De te gebruiken documentstijlen zijn:

Standaard:

Lettertype: Verdana

Tekenstijl: Normaal

Puntgrootte: 10pt

Tekstkleur: Zwart

Regelafstand: Anderhalf

Uitlijning: Links

Kop 1:

Lettertype: Verdana

Tekenstijl: Vet

Puntgrootte: 12pt

Tekstkleur: Grijs 50%

Regelafstand: Anderhalf

Uitlijning: Links

Kop 2:

Lettertype: Verdana

Tekenstijl: Vet

Puntgrootte: 11pt

Tekstkleur: Zwart

Regelafstand: Anderhalf

Uitlijning: Links

Kop 3:

Lettertype: Verdana



Tekenstijl: Vet

Puntgrootte: 10pt

Tekstkleur: Zwart

Regelafstand: Anderhalf

Uitlijning: Links

Kop- en voetteksten zullen bestaan uit een logo van eProjects in de koptekst. In de voettekst staat de documentnaam en titel samen met het paginanummer.

## **Gegevensbank**

Het CMS zal voor twee verschillende gegevensbanken worden ontwikkeld. Dit zijn Microsoft Access en Microsoft SQL-server. Naast deze twee gegevensbanken zou in de toekomst het mogelijk moeten zijn om tevens andere gegevensbanken te gebruiken.

## **Programmeren**

De ontwikkelomgeving voor het programmeren is Microsoft Visual Studio .NET. Visual Studio maakt gebruik van de .NET programmeertaal. .NET bestaat uit een aantal verschillende talen bijvoorbeeld C#.NET, VB.NET en J#.NET.

Webapplicaties worden in ASP.NET geschreven. De webapplicaties zijn op te splitsen in twee delen. De delen zijn opmaak en codebehind.

De opmaak is een combinatie van HTML-controls en ASP.NET-webcontrols. Dit zorgt voor het uiterlijk van de website. Codebehind wordt gebruikt voor het opvangen van events en het uitvoeren van bewerkingen op data. De events treden op als er bijvoorbeeld op een knop wordt geklikt.

De programmeertaal die gebruikt wordt voor de codebehind-bestanden is C#.NET (C-sharp).





## **Versiebeheer**

Versies van de te maken programmacodebestanden worden beheerd door de toepassing TortoiseCVS. Hierdoor wordt voorkomen dat wijzigingen verloren gaan als meerdere personen tegelijkertijd bestanden wijzigen.

## **Taal**

Voor het benoemen van modellen, tabellen, veldnamen, klassen, functies, eigenschappen, variabelen en commentaar in de code wordt de Engelse taal gebruikt. Dit is een standaard van eProjects.

## **Aanpak**

### **Projectorganisatie**

De afstudeerder

Van de stagiaire wordt verwacht dat hij volgens het plan van aanpak te werk gaat. In het plan van aanpak staat een globale planning. De planning is een leidraad voor het project. Waar nodig zal de planning aangepast moeten worden.

De stagiaire zal regelmatig de voortgang van het project rapporteren bij de opdrachtgever en examinatoren.

De opdrachtgever

De opdrachtgever zal beschikbaar moeten zijn voor vraaggesprekken en voor voldoende terugkoppeling. Terugkoppeling heeft betrekking op de op te leveren producten.

De examinatoren

Examinatoren zullen tijdige reageren op de terugkoppeling van de voortgang van de stagiaire. Waar nodig zullen zij zorgen voor bijsturing.

### **Projectgroep**

De projectgroep is samengesteld uit mensen die allen werkzaam zijn binnen eProjects. Deze groep zal meerdere malen benaderd worden voor het inwinnen van informatie. Het inwinnen van de informatie zal via persoonlijke benadering en groepsoverleg zijn.

Onderstaande tabel geeft een overzicht van de projectleden:

Naam	Functie	Taak
R. Dolle	Opdrachtgever	Voorziet in de opdracht en benodigde informatie m.b.t. eisen van het systeem
D. Roos	Gebruiker/ verkoop	
W. Sussenbach	Webdesign/ gebruiker	Webdesign, maakt interfaces voor Cms
Marcel	Webdesign/ gebruiker	Webdesign?
H. de Ryck	Usability/ Medewerker	Doet onderzoek naar gebruikersvriendelijkheid en de gebruikersinterface
J. Veldhuizen	Programmeur/ gebruiker	
B. de Jager	Afstudeerder	

## **Te gebruiken methoden en technieken**

### **Methoden**

De methode die binnen eProjects veel wordt toegepast is IAD (Iterative Application Development). De karakteristiek van IAD is het iteratief ontwikkelen. Aan de hand van verschillende pilots kan er snel iteratief worden ontwikkeld. Een pilot is een applicatie die als tussenproduct dient. Definitiestudie en pilotontwikkeling zullen meerdere malen doorlopen worden.

### **Technieken**

De techniek die gebruikt gaat worden is UML (Unified Modeling Language). UML is een object georiënteerde modelleringstechniek. Deze kan gebruikt worden voor de ontwikkeling van een object georiënteerde systeem. UML is een veel gebruikte techniek binnen IAD en wordt veel gebruikt door eProjects.

Gebruikte diagrammen van UML zijn afhankelijk van de te ontwikkelen pilot. De keuze voor een diagram zal worden toegelicht in het eindverslag. De te gebruiken diagrammen zijn:

- Klassendiagrammen
- Use-case-diagrammen
- Sequentiediagrammen

### Risico's

De afstudeerder heeft geen kennis van IAD. Er zal tijd moeten worden gereserveerd voor het bekendmaken met de methode. Door gebrek aan kennis kan tijdens ontwikkeling vertraging worden opgelopen. Mocht dit plaatsvinden dan zal de planning aangepast moeten worden.

### Planning

Onderdeel	Activiteit	Dagen
<i>Opstart</i>		<i>5</i>
<i>Pilot 1</i>		<i>26</i>
<i>Pilot 2</i>		<i>21</i>
<i>Pilot 3</i>		<i>10</i>
<i>Pilot 4</i>		<i>10</i>
<i>Verslag</i>		<i>18</i>
<b><i>Totaal</i></b>		<b><i>90</i></b>

Het proces van de ontwikkeling van de 4 pilots zullen voornamelijk allemaal dezelfde indeling hebben. In onderstaande tabel zijn de activiteiten, welke voor elke pilot gelden, te zien.

Onderdeel	Activiteit	Procentuele tijdsindeling
<i>Pilot</i>		
	Klassendiagram maken	7 %
	Use-cases maken	7 %
	Use-case-diagram maken	8 %
	Sequence-diagrammen maken	8 %
	Programmeren van klassen	30 %
	Functionaliteit per scherm implementeren	30 %
	Schrijven programmadocumentatie. (beheerdershandleiding)	5 %
	Testen op functionaliteit en fouten	5 %

## Implementatie

Het CMS zal "achter" verschillende websites geplaatst moeten worden. Websites hebben verschillende lay-out en deze lay-out is deels statisch en deels dynamisch. Het dynamische deel zal worden aangestuurd door het CMS. Implementatie is per oplevering van een website. Het dynamische deel zal gekoppeld moeten worden aan het CMS.

## Kwaliteit

De ontwikkeling van de applicatie gaat aan de hand van pilotontwikkeling. Aan het einde van het traject van een pilotontwikkeling vindt een testfase plaats. Deze zal samen met de opdrachtgever en eventuele gebruikers uitgevoerd worden.

De vooraf vastgestelde eisen worden tijdens de test vergeleken met het eindresultaat. Na de test zal er een beeld ontstaan van de eisen waaraan de pilot wel en/ of niet voldoet.

### **Afsluiting**

De afsluiting is te verdelen in twee onderdelen inleveren verslagen, opleveren applicatie.

Vrijdag 11 juni 2004 tussen 10.30 uur en 13.00 uur worden 4 kopieën van de verslagen op school ingeleverd. Naast het inleveren op school wordt er bij eProjects een kopie ingeleverd.

De applicatie zal aan het eind van de periode worden overgedragen aan eProjects.

Bijlage: **Gedetailleerde planning**

## **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

Bijlage: **Definitie van Eisen**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**





## Inhoudsopgave

Inhoudsopgave .....	i
Functionele eisen .....	1
Totale functionele eisen in tabelvorm .....	1
Pilot 1: Basis Cms.....	4
Pilot 2: Gebruikers.....	5
Pilot 3: Integratie ProSuite .....	5
Pilot 4: Modules en modulaire opbouw .....	5
Beperkingen ontwerp.....	7
Database.....	7
Inleiding .....	7
Eén of meerdere databases.....	7
MS Access en/ of MS SQL-server .....	8
Software beperkingen .....	9
Hardware beperkingen .....	9
Attributen .....	10
Beveiliging.....	10
Onderhoudbaarheid .....	10

## Functionele eisen

### Totale functionele eisen in tabelvorm

Onderstaande tabel geeft een totaal overzicht van alle eisen die te stellen zijn aan een Cms. Dit is een lijst van functies die door iedereen is aangevuld tijdens het project. Bij het bedenken van nieuwe eisen tijdens ontwikkeling, weekendje weg of thuis konden deze hier worden toegevoegd.

Per functie is een prioriteit aangegeven. Hierna zullen de eisen worden opgezet in eisen per pilot.

<b>Prioriteit</b>	<b>Functie</b>	<b>Beschrijving</b>
1	Menu's aanmaken	Op verschillende plaatsen in de site
	Meertalig	
1	Snel aanpassen capture (topframe, kleur, vorm)	
2	Automatische aanpassing vormgeven a.d.h.v. criteria	timer aanpassing aan kleurstelling en gadgets zoals sneeuw aan winter lente etc.
1	Duidelijk structuur dynamisch aan te maken	Hoofdstructuur/sub/sub/submenu's
3	Autocontent	
1	Artikelroulator	
1	Forum	
2	Chat	
2	Polling	
2	Sharepoint	MS SharePoint functies overnemen
1	Aanpassen artikelen, popup met deels backoffice	aanpassen van artikelen door middel van popup met backoffice mogelijkheden en WYSIWYG tool
1	Browserspecifiek	Moet het op alle browsers werken, kan dat met ASP.NET

1	Gebruikers en gebruikersgroepen	Gebruikers moeten/ kunnen in een groep geplaatst worden, deze groep heeft toegang tot bepaalde artikels op de site. Maar mogelijk heeft de gebruiker ook toegang tot specifieke delen
1	Modulair opzetten	CmsPro is een onderdeel van een groter geheel, het beheert niet alleen de website, maar ook intranet en documenten beheer. Wellicht meer andere aansturingen van modules
2	Module: Statistieken	Statistieken --> IP, Gebruikers login, Browsers, Tabs, Per pagina - hoeveelheid, per dag, per uur, per week, per maand, per land
1	Module: Nieuws berichten	
2	Module: Banners	
1	Module: Contactgegevens (CRM)	
1	Module: Forum	
1	Module: Documenten	
2	Module: Evenementen/ kalender	
1	Module: FAQ's	
1	Module: feedbackformulier	
	Module: Iframe	Een andere site kan worden weergegeven binnen deze site
1	Module: Image	
1	Module: links	Links binnen huidige venster, nieuw venster en nieuw venster zonder titelbalk, knoppenbalk en adresbalk
1	Module: RSS (news feed)	Via RSS-feed kunnen verschillende gegevens van andere sites weergegeven worden op deze site. Voorbeeld is Marktplaats aanbiedingen of

		FOK nieuwsberichten.
2	Module: Zoeken op site/ Google	
1	Module: Polls	
1	Module: Tekst/ HTML input	
2	Module: Tabel	De gebruiker kan zelf een tabel aanmaken en de verschillende velden definiëren
2	Module: XML/XSL	
2	Mogelijkheid om Mobilepagina's te genereren	
1	Tabs toevoegen	
1	Site instellingen	
1	Gebruikers beheren	
1	Gebruikersrechten	
1	Module: email	Het mailen van een nieuwsbrief aan geregistreerde gebruikers, per gebruikersrecht, per aanmelding
1	Module: file manager	
2	Module: Host settings	
2	Module: SiteMap	
1	Module: Reviews	
3	Module: Vertel een vriend	
3	Module: Webcam	
2	Module: fotoalbum	
3	Module: quotes	
3	Module: export	
2	Module: Errorhandling	
1	Module: Support	Dezelfde module als reageer van eprojects.nl
2	Module: Quote of the day	

2	Module: task	
3	Module: media player	
2	Module: berichten centrum/ System messenger	
1	Module: gastboek	
1	Module: Formcreator	
2	Module: Sollicitatiemodule	Post jobs on your Website using this module. Allow applicants post their Word formatted resume and send it to you via email. Define an autoresponder for the applicant when they apply for your job. Private assembly file and source code included.
2	Module: online users	

## Pilot 1: Basis Cms

Functionele eisen voor pilot Basis Cms:

- Beheren van meerdere sites in een Cms
- Menu's aanmaken
- Menu's wijzigen
- Menu's verwijderen
- Artikelen aanmaken
- Artikelen wijzigen
- Artikelen verwijderen
- Artikelen aan een menu toewijzen
- Volgorde van artikelen in een menu bepalen
- Artikelen in archief plaatsen
- Inlog voor beheerder
- Generiek opzetten van systeem, zodat het Cms op verschillende websites toepasbaar is

## **Pilot 2: Gebruikers**

Functionele eisen voor pilot Gebruikers:

- Gebruikers toevoegen
- Gebruikers wijzigen
- Gebruikers verwijderen
- Indelen van gebruikers in gebruikersgroepen
- Gebruikersgroepen toevoegen
- Gebruikersgroepen wijzigen
- Gebruikersgroepen verwijderen
- Toewijzen van rechten aan gebruikersgroepen
- Koppelen van gebruikersgroepen aan menu's, artikelen, etc

## **Pilot 3: Integratie ProSuite**

Functionele eisen voor pilot Integratie ProSuite:

- Het Cms moet hetzelfde uiterlijk als de ProSuite krijgen
- De opbouw van de website moet veranderen en een betere weergave van de structuur van de website
- Het gebruikers- en gebruikersgroepenmodel moet vervangen worden met het gebruikers- en gebruikersgroepenmodel van de ProSuite

## **Pilot 4: Modules en modulaire opbouw**

De systeemeisen van de Bannermodule:

- Banner aanmaken
- Banner wijzigen
- Banner verwijderen
- Een banner moet uit een plaatje bestaan
- Een banner kan een link bevatten

#### Systeemeisen voor status

- Status aanmaken
- Status wijzigen
- Status verwijderen
- Gebruikersgroep toevoegen aan status
- Status samenvoegen met bestaande modules (artikelen, banner)

## **Beperkingen ontwerp**

### **Database**

#### **Inleiding**

De database bevat de data van de applicatie. Hierin wordt de website opgebouwd en de database wordt vanuit twee kanten benaderd. Vanuit de website voor het opbouwen van de website en vanuit de ProSuite voor het onderhoud van de website.

#### **Eén of meerdere databases**

De ProSuite is zo ontworpen dat alle klanten inloggen op hetzelfde systeem. Door de inloggegevens zijn alleen die gegevens toegankelijk waar de gebruiker recht op heeft. De data laag is zo ontworpen dat elke inlog een eigen of gezamenlijke database kan gebruiken. Na de inlog wordt er voor elk bedrijf een aparte connectiestring opgehaald. In de connectiestring staat welke server, database en gebruiker er moet worden gebruikt voor de connectie. Hierdoor is het mogelijk iedereen dezelfde connectiestring te laten gebruiken of iedereen een eigen.

Ik heb ervoor gekozen om gebruik te maken van één database. Zeker omdat tijdens de ontwikkeling er veel aanpassingen moeten worden doorgevoerd. Als er bij elke aanpassing verschillende databases aangepast moeten worden is dat tijdrovend. Het Cms moet zo worden opgebouwd dat een klant één of meerdere sites kan beheren. Deze sites zijn ieder weer opgebouwd uit menu's, artikelen, banners, etc. Dit moet aan elkaar verbonden zijn, want alleen dan is het mogelijk een scheiding van verschillende klanten in één database te maken.

Het aanpassen van de database gebeurt met een script wat uitgevoerd moet worden in de SQL-server. Het script wordt per database uitgevoerd en zorgt ervoor dat aanpassingen aan tabellen en opgeslagen procedures worden doorgevoerd. Het uitvoeren van een dergelijk script op één database of meerdere databases is enkel herhalen van een handeling.



Er zijn meerdere nadelen aan dit systeem. Bij het aanmaken van een site voor een klant kan een script uitgevoerd worden. Er wordt dan een aantal samenhangende gegevens ingevoerd in de database. Daarna kan een gebruiker inloggen in de ProSuite en heeft hij toegang tot zijn site. De hoeveelheid gegevens zal alleen maar groeien. In de hoeveelheid gegevens kan mogelijk een probleem liggen. De gegevens in de database zullen exponentieel groeien. Dit zou kunnen resulteren in het traag worden van het systeem.

Een ander nadeel van een database tegenover meerdere databases is het verhuizen van klanten. In één database met meerdere sites van meerdere klanten is het moeilijk een klant eruit te filteren en deze dan in zijn geheel te verhuizen naar een nieuwe locatie.

Het terugzetten van een back-up van één klant is bij één database niet mogelijk. Het terugzetten van de gegevens van een klant kan alleen wanneer de klant een eigen database heeft. Want bij het terugzetten van een back-up wordt de gehele database overschreven. Doordat er meerdere klanten in een database zitten zullen deze allen worden overschreven.

Mocht er in de toekomst zich verschillende problemen zich voordoen kan er alsnog worden bekeken wat dan de beste oplossingen zijn.

### **MS Access en/ of MS SQL-server**

Het Cms gaat gebruikt worden voor simpele en geavanceerde websites. Voor simpele websites is de MS SQL-server een zware database en zijn de mogelijkheden van MS Access toereikend. Bij integratie in de ProSuite moet het Cms op een SQL-server werken, want de ProSuite is een op SQL-server werkende applicatie. Het Cms moet dus ontwikkeld worden voor twee soorten databases.

Door de transparantie van de nieuwe datalaag zou de overgang van een MS Access database naar een SQL-server database zonder problemen moeten verlopen. Het verschil in de databases zit voornamelijk in de uitgebreide mogelijkheden van SQL-server. Om ervoor te zorgen dat het Cms met beide databases werkt zullen de queries in MS Access en de opgeslagen procedures in SQL-server hetzelfde moeten uitvoeren. De beperking ligt in de MS Access database en is daarom de uitgangpositie voor het Cms.



Later tijdens de integratie van de ProSuite is besloten het Cms alleen aan te bieden met een SQL-server database. Dit zal toegelicht worden in de beschrijving van de pilot Integratie ProSuite.

### **Software beperkingen**

Er zijn vrijwel geen softwarebeperkingen aan het Cms. Het Cms wordt vanuit de webbrowser gestart. Software matig zijn de eisen voor de gebruiker dat hij een browser heeft.

De technische beperkingen zijn groter. Voor de te ontwikkelen website zal er rekening gehouden moet worden met de verschillende browsers. Momenteel zijn de meest toonaangevende browser: Internet Explorer, Mozilla en Netscape. De website zal voor alle browser geschikt moeten zijn.

De ProSuite heeft als eis dat deze alleen met Internet Explorer 5.5 of hoger werkt. Omdat de ProSuite geavanceerde functies heeft die alleen met Internet Explorer samenwerken.

### **Hardware beperkingen**

Hardware beperkingen hebben enkel te maken met de computer van de gebruiker en zijn internetverbinding. Het is voor een gebruiker wenselijk dat hij een snelle computer en internetverbinding heeft. Dit bevordert de snelheid van de applicatie.



## **Attributen**

### **Beveiliging**

De beveiliging van het Cms wordt moet op gebruikersniveau worden geregeld. Er zal een gebruikers- en gebruikersgroepenmodel in het Cms worden gebouwd. In de ProSuite zit al een gebruikers- en gebruikersgroepenmodel en kan eventueel gebruikt worden.

### **Onderhoudbaarheid**

EProjects werkt met Tortoise CVS. Hierdoor wordt voorkomen dat wijzigingen verloren gaan als meerdere personen tegelijkertijd bestanden wijzigen. Tijdens het programmeren maak je lokaal wijzigingen. Als het stuk code werkt en getest is kan het geüpdate worden.

Bijlage: **Onderzoek ontwerp elementen**

## **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

Bijlage: **Pilotplan**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**



## Inhoudsopgave

Inhoudsopgave .....	i
Pilots .....	1
Pilot Basis Cms.....	1
Pilot Gebruikers.....	3
Pilot Modules en modulaire opbouw .....	4
Pilot Integratie ProSuite .....	5
Tijdsindeling .....	7
Pilot volgorde.....	10
Schatting benodigde tijd.....	10
Globaal invoeringsplan.....	12

## Pilots

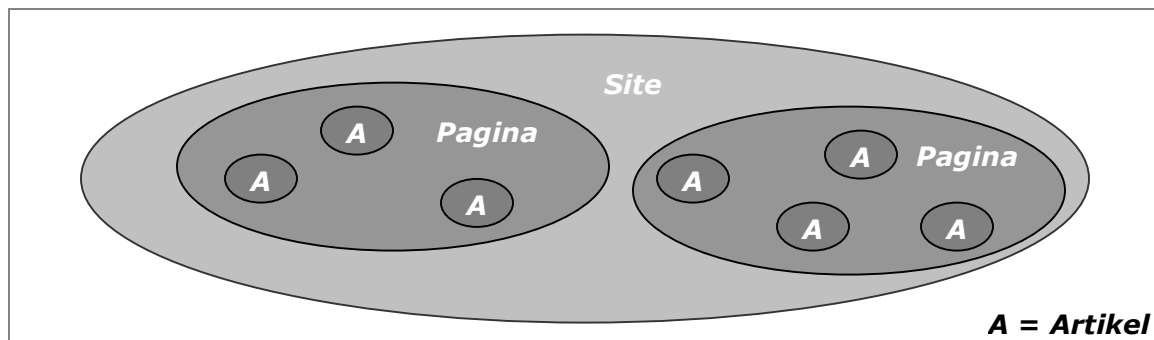
In de bespreking van 3 maart '04 is besloten met de opdrachtgever om pilot 3 en pilot 4 te wisselen. De moeilijkheidsgraad van pilot 4 was hoger als die van pilot 3. Met de koppelingen met bestaande systemen ben ik afhankelijk van de systemen en de programmeurs die eraan werken.

Er moeten verschillende koppelingen worden gemaakt met databases (accountpro, moverspro, etc.).

Pilot 3 is een op zich staande pilot en kan later worden geïmplementeerd. De invoering van de modulaire werking is eenvoudiger. En kan worden gerealiseerd op elk moment.

## Pilot Basis Cms

Een Cms is een gebruiksvriendelijk, webbased onderhoudssysteem voor het beheren van een website. De basis voor een Cms-website zijn artikelen. Een artikel is een stuk tekst wat op de site wordt geplaatst. Het kan vergeleken worden met een artikel in een magazine. Een aantal artikelen samen vormen een pagina. De website waar de bezoeker op komt is opgedeeld in verschillende pagina's.



Hierboven is de opbouw van een Cms te zien. Dit vormt de basis voor pilot Basis Cms.

Een belangrijke eigenschap van deze pilot is het generieke karakter. Elke website heeft een ander uiterlijk. Elke website heeft een ander doel en doelgroep. Het Cms moet generiek worden opgebouwd zodat het mogelijk is te gebruiken bij verschillende ontwerpen voor sites.

Het beheer van de site zal voor onbevoegde mensen afgesloten moeten worden. Dit kan bijvoorbeeld door een inlog. De gebruiker moet voordat hij wijzigen kan aanbrengen inloggen op een beveiligd gedeelte. Na het inloggen krijgt de gebruiker pas toegang tot de backoffice. In de backoffice kunnen aanpassingen gemaakt worden voor de website. De backoffice vormt het Cms.

Functionele eisen voor pilot Basis Cms:

- Beheren van meerdere sites in een Cms
- Menu's aanmaken
- Menu's wijzigen
- Menu's verwijderen
- Artikelen aanmaken
- Artikelen wijzigen
- Artikelen verwijderen
- Artikelen aan een menu toewijzen
- Volgorde van artikelen in een menu bepalen
- Artikelen in archief plaatsen
- Inlog voor beheerder
- Generiek opzetten van systeem, zodat het Cms op verschillende websites toepasbaar is

Moeilijkheid: Koppeling interfaces



## **Pilot Gebruikers**

Na het afronden van pilot Basis Cms is er een werkend Cms. Voor het beheren van de site zal er een inlog nodig zijn. Dit wordt in pilot Basis Cms ingebouwd. Anders is het mogelijk dat iedereen de inhoud van de site kan wijzigen.

Het inloggen voor de beheerder is het begin van het gebruikersmodel en pilot Gebruikers. Het gebruikersmodel is op te splitsen in twee delen. Er zijn gebruikers op de website (bezoekers met inlogrechten) en de gebruikers die recht hebben in te loggen op de backoffice en daar aanpassingen te maken die doorgevoerd worden in de website.

Gebruikers en gebruikersgroepen worden gebruikt voor het toekennen van rechten aan de gebruikers. Rechten op bepaalde delen van de site. Hierbij kan gedacht worden aan het mogen bekijken van artikelen (publieke artikelen en afgeschermd artikelen). Andere rechten hebben te maken met de mogelijkheden van de backoffice. Er kunnen beperkingen worden opgelegd aan een gebruiker in de backoffice. Hierbij kan gedacht worden aan het wel of niet mogen bewerken van artikelen.

Functionele eisen voor pilot Gebruikers:

- Gebruikers toevoegen
- Gebruikers wijzigen
- Gebruikers verwijderen
- Indelen van gebruikers in gebruikersgroepen
- Gebruikersgroepen toevoegen
- Gebruikersgroepen wijzigen
- Gebruikersgroepen verwijderen
- Toewijzen van rechten aan gebruikersgroepen

Moeilijkheid: Duidelijke rechtenstructuur die overal inzetbaar en begripbaar is. (stakeholders en prosuite, pilot Integratie Prosuite)

## **Pilot Modules en modulaire opbouw**

Het Basis Cms, welke in Pilot Basis Cms is besproken, heeft enkel artikelen als inhoud voor de website. Dit is een basis en beperk systeem. Uitbreiding van verschillende mogelijkheden is de inhoud van deze pilot.

Het systeem zal zo worden opgebouwd dat het mogelijk is verschillende modules toe te voegen. Een module is een onderdeel van het Cms. Een module geeft een extra functionaliteit en is een op zichzelf werkend onderdeel. De modules kunnen nu gemaakt worden, maar ook in de toekomst. Modules zullen generiek zijn, maar het is ook mogelijk dat een module als maatwerk voor een klant gemaakt wordt.

Voorbeelden voor modules zijn banners, polls, evenementen, kalender, email/feedback formulier en uitgebreide enqueteformulieren.

Functionele eisen voor Pilot Modules en modulaire opbouw:

- Maken van modules (aantal is afhankelijk van tijdsbestek)
- Toevoegen van modules aan Cms
- Het in de backoffice kunnen beheren van modules
- Beheerdershandleiding opstellen voor ontwerpen modules

Moeilijkheid: Modulair opzetten van modules die zonder of weinig aanpassingen aan het Cms toegevoegd kunnen worden.



## **Pilot Integratie ProSuite**

De ProSuite is een van de produkten van eProjects. Het is een totaal pakket van de verschillen produkten. ProSuite maakt gezamenlijk gebruik van documenten, een centrale agenda en een relatiebeheersysteem mogelijk. ProSuite bestaat uit een relatiebeheersysteem, boekhoudpakket, CmsPro, WebshopPro, HrmPro (Human Resource Management), ProcesPro, ServicePro.

Het Cms gaat een onderdeel van de ProSuite worden. De integratie van het Cms houdt in dat het uiterlijk van de backoffice van het Cms hetzelfde uiterlijk moet hebben als de ProSuite. Maar niet alleen uiterlijk is bepalend, maar gebruik van hetzelfde rollensysteem, database, etc.

Functionele eisen voor

Pilot Integratie ProSuite:

- Integratie ProSuite
- Overeenkomstig uiterlijk als ProSuite
- Rollensysteem gebruiken en uitbreiden van ProSuite
- Koppeling met de relaties en gebruikersgroepen die zich in de ProSuite (CRM) bevinden

Moeilijkheid: Uiterlijk overnemen en daarbij gebruik te maken van bestaande functies voor opslag en gebruikersvriendelijkheid. (melding als er een verandering is opgetreden en deze verloren gaat als men naar een andere functie gaat).

## Tijdsindeling

De volgorde van de pilots wordt bepaald door de verdeling van prioriteit. Omdat het systeem opnieuw wordt ontwikkeld zal er een begin worden gemaakt met een basis systeem. De Pilot Basis Cms zal de eerste pilot in dit proces worden. Vanuit deze pilot kan er verder worden gewerkt aan de ontwikkeling van uitbreidingen. Andere pilots zijn een aanvulling op de Pilot Basis Cms.

Overige pilots kunnen in verschillende volgorden worden ontwikkeld. De keuze voor de volgorde is gebaseerd aan de hand van het MoSCoW-principe. Volgens dit principe moest er worden gekeken naar vereiste functies, gewenste functies, luxe functies en extra functionaliteit indien alle overige zaken zijn gerealiseerd.

Gebruikers en gebruikersgroepen vormen een belangrijke onderdeel van het Cms.

Voor het generiek opzetten van het Cms zijn gebruikers en gebruikersgroepen een belangrijk onderdeel. Een eis voor het Cms is het tegelijkertijd onderhouden van de website met meerdere gebruikers. De gebruikers moesten wel verschillende rechten hebben voor het toevoegen/ wijzigen en verwijderen van berichten. Niet iedereen zou beheerdersrechten mogen hebben.

Naast het rechtensysteem voor de backoffice zijn de rechten ook toepasbaar voor de website. Er kan gedacht worden aan het inloggen op de website en hierdoor recht krijgen op het lezen van meer artikelen.

Het rechten systeem heeft niet alleen betrekking op artikelen, maar bij de ontwikkeling van de Pilot Modules en modulaire opbouw komt dit weer terug. Elke module zou ook rechten kunnen krijgen. Een voorbeeld is wanneer polls zichtbaar zijn voor iedereen, maar het resultaat is alleen beschikbaar voor gebruikers die ingelogd zijn of je kan alleen je stem uitbrengen wanneer je ingelogd bent.

Het ontwikkelen van de Pilot Gebruikers heeft dus invloed op het gehele systeem en de vervolg pilots. Hierdoor is dit de volgende te ontwikkelen pilot.

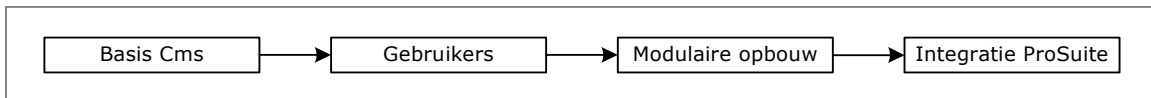


Voor afronding van het systeem en belangrijkheid van de pilots is gekozen om als eerste de Pilot Modules en modulaire opbouw en pas daarna de

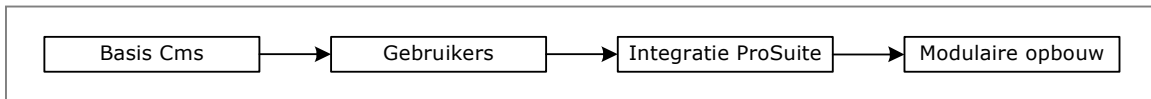
Pilot Integratie ProSuite uit te voeren. Maar door de ontwikkelingen van de ProSuite en de tijdsplanning van de invoering van de ProSuite is hier verandering in gekomen. Modules worden niet alleen nu ontwikkeld, maar zullen altijd ontwikkeld blijven worden. Hierdoor kan het systeem uitgebreid worden naar de wensen van de gebruikers. Modules zullen door verschillende programmeurs ontwikkeld worden en de implementatie moet eenvoudig zijn. De integratie van de ProSuite is een afgesloten geheel dit in tegenstelling tot de Pilot Modules en modulaire opbouw, want het aantal modules dat ontwikkeld wordt, wordt bepaald door de resterende tijd van het project.

Pilot Integratie ProSuite zal de derde pilot worden. Als dit punt bereikt is, is er een volledig werkend Cms. Hierna kan Pilot Modules en modulaire opbouw het Cms uitbreiden met nieuwe modules.

*Concept 1:*



*Concept 2:*



## **Pilot volgorde**

De volgorde van de pilots zal als volgt zijn:

1. Pilot Basis Cms
2. Pilot Gebruikers
3. Pilot Integratie ProSuite
4. Pilot Modulaire opbouw

## **Schatting benodigde tijd**

Per pilot is er een schatting gemaakt met betrekking tot de tijd die nodig is voor ontwikkeling van de pilot. Pilot Basis Cms is de eerste pilot die ontwikkeld zal worden en vormt hiermee de basis voor het systeem. Het is van groot belang dat dit een solide basis vormt. Hierna zullen hier de andere pilots op gebouwd worden. Mocht zich in deze pilot een fout voordoen kan dit gevolgen hebben op de andere pilots. Er is gekozen om 25 dagen uit te trekken voor het ontwikkelen van deze pilot.



De tweede pilot is het gebruikersmodel. Hier zijn 20 dagen voor ingepland. Door de ontwikkeling van het gebruikersmodel is het mogelijk dat er aanpassingen gemaakt moeten worden aan pilot 1. Deze aanpassingen kosten tijd en het gebruikersmodel moet geheel in kaart worden gebracht. Alle verschillende aspecten van het gebruikersmodel moeten in kaart worden gebracht. Hierna moeten deze verschillende model te verwerken zijn in het systeem.

Pilot Integratie ProSuite heeft 7 dagen gekregen en zal geen moeilijke pilot zijn. Het gebruik maken van de gebruikersgroepen uit het Crm gaat met objectorientatie relatief makkelijk. De intelligentie van de relaties en gebruikersgroepen zitten in de klassen van het Crm en zouden toegankelijk moeten zijn.

De overgebleven tijd wordt verdeeld aan rapportage en pilot 4. Volgens een totale planning zal er 10 dagen zijn voor Pilot Modules en modulaire opbouw. Naar gelange kan gekeken worden naar het aantal te ontwikkelen modules.

### **Globaal invoeringsplan**

Door in globale termen de processen van installatie en overdracht te beschrijven, rollen en verantwoordelijkheden te definiëren en procedures te schetsen voor het tussentijds onderhouden van de pilot en voor het verzamelen van feedback.

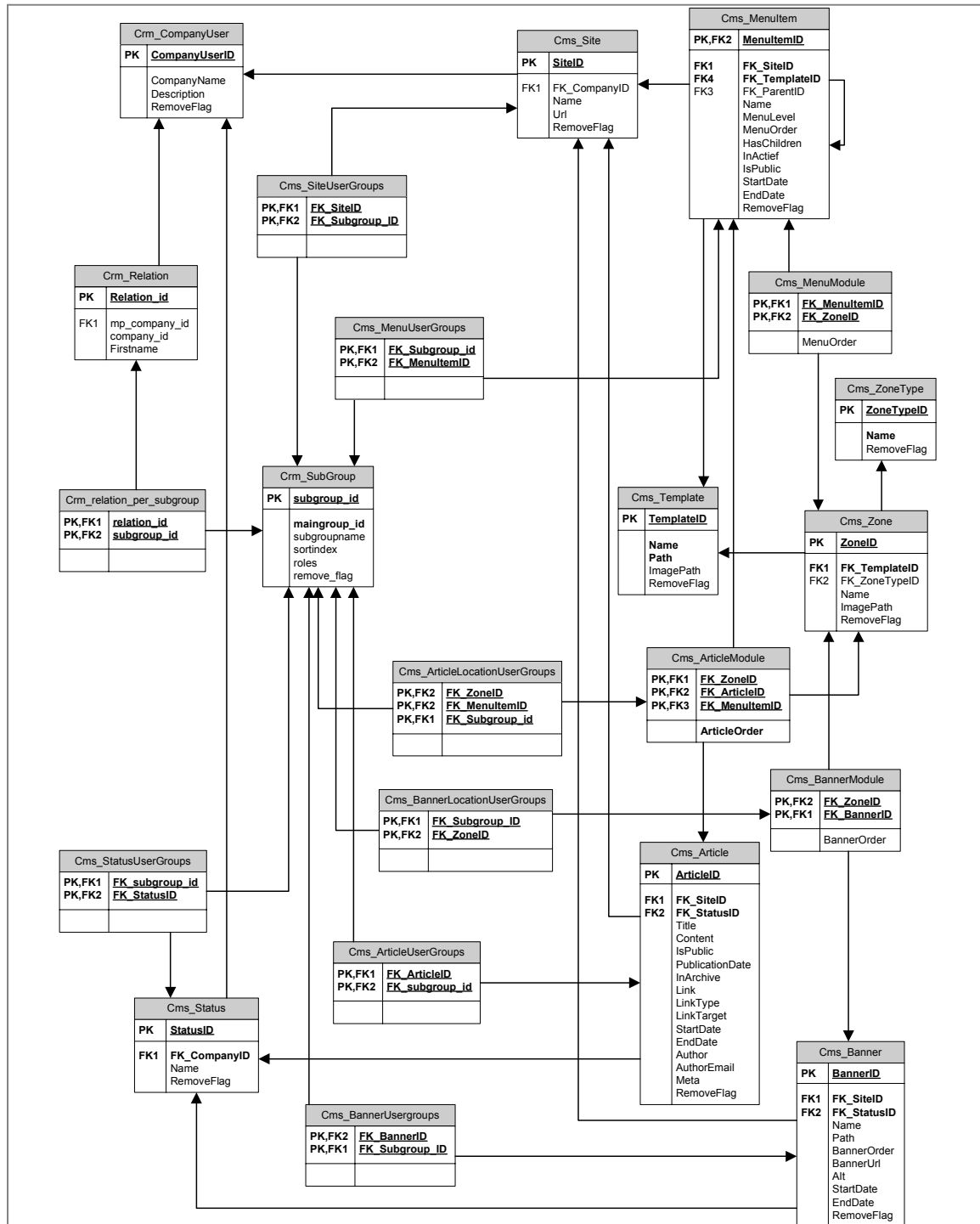
Bijlage: **Ontwerpdocumentatie**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**

## Inhoudsopgave

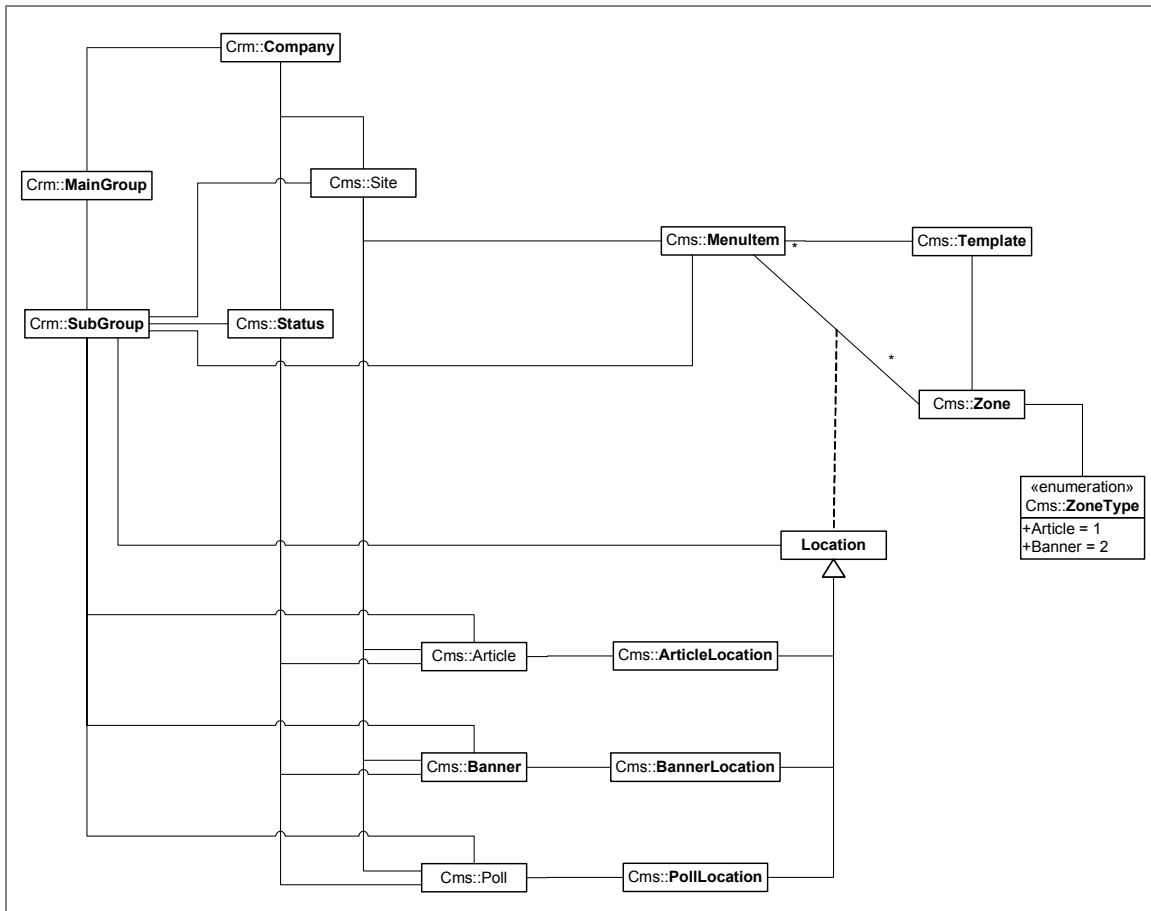
Inhoudsopgave .....	i
Databaseontwerp .....	1
UML.....	2
Klassendiagram.....	2
Objectdiagram voor bepalen locatie .....	3
Use-Cases .....	4
Sequentiediagrammen .....	7
Rechtenstructuur Cms .....	10

## Databasontwerp

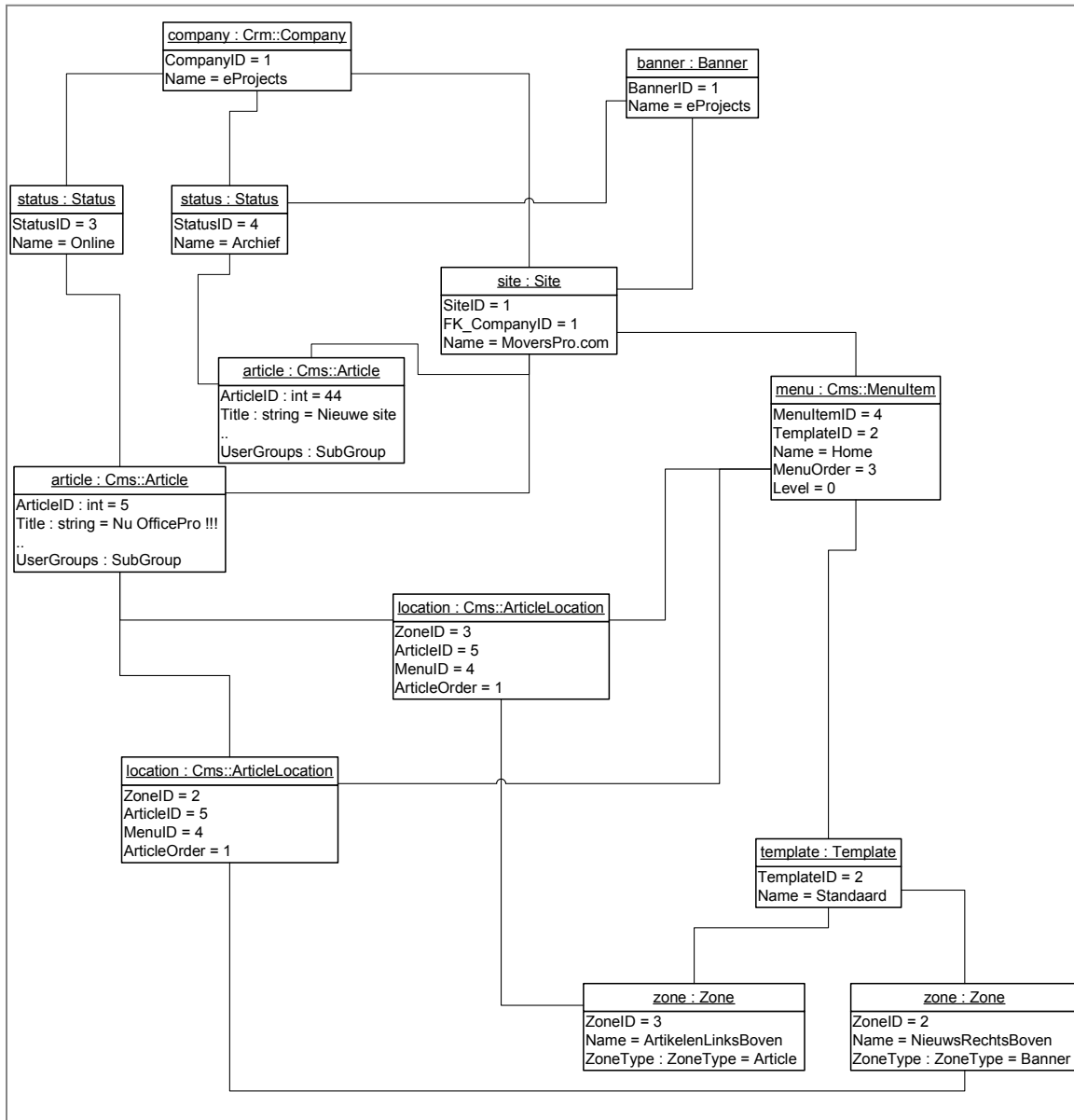


## UML

### Klassendiagram

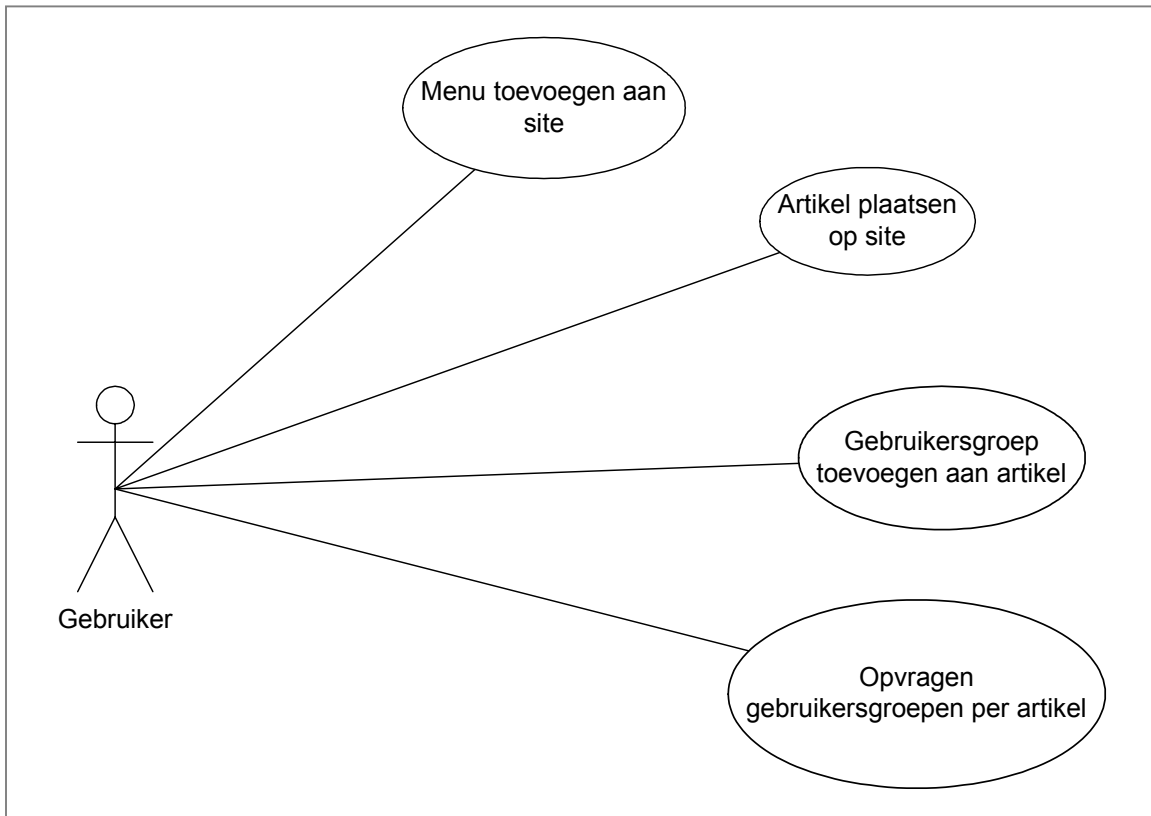


## Objectdiagram voor bepalen locatie



## Use-Cases

De use-cases die ik gemaakt heb voor de verschillende pilots zijn hier samen weergegeven.



Use-cases:

use-case 1: Menu toevoegen aan site

use-case 2: Een bestaand artikel plaatsen op site

use-case 3: Gebruikersgroep toevoegen aan artikel

use-case 4: Opvragen gebruikersgroepen per artikel



Naam	Menu toevoegen aan site
Samenvatting	Een gebruiker kan een nieuw menu aanmaken. Het menu is afhankelijk van een site
Actoren	Gebruiker in de ProSuite
Aannamen	De gebruiker is binnen het systeem geregistreerd. De gebruiker heeft de rechten: Mag sitemap lezen, Mag menu's lezen en Mag menu's aanmaken.
Beschrijving	De gebruiker heeft het scherm sitemap voor zich en gaat vanuit hier naar een menu aanmaken bij een site. Er zal een site gekozen moeten worden waarna het mogelijk is een menu aan te maken. Als het menu aangemaakt wordt moet de gebruiker een naam invullen, een layout kiezen. Hierna kan hij meerdere attributen aangeven.
Uitzondering	Als de gebruiker niet de juiste rechten heeft.
Resultaat	Er is een nieuw menu aangemaakt onder een site.

#### **use-case 1: Menu toevoegen aan site**

Naam	Een bestaand artikel plaatsen op site
Samenvatting	Een gebruiker kan een artikel op een locatie plaatsen
Actoren	Gebruiker in de ProSuite
Aannamen	De gebruiker is binnen het systeem geregistreerd. De gebruiker heeft de rechten: Mag sitemap lezen, Mag menu's lezen, Mag artikelen toevoegen aan menu's, Mag artikelen lezen.
Beschrijving	De gebruiker zal op de sitemap de betreffende locatie moeten kiezen. De stappen voor het kiezen zijn: site → menu → locatie. Na het kiezen van een locatie kan een artikel toegevoegd worden aan de locatie. Dit gebeurt door de knop 'nieuw', waarna een dropdownlist met artikelen zal verschijnen. Na het kiezen van een artikel kan deze worden opgeslagen.

Uitzondering	Als de gebruiker niet de juiste rechten heeft.
Resultaat	Het artikel is op de site geplaatst en is te zien in de frontoffice

### **use-case 2: Een bestaand artikel plaatsen op site**

Naam	Gebruikersgroep toevoegen aan artikel
Samenvatting	Een gebruikersgroep uit de ProSuite kan worden toegevoegd aan een artikel.
Actoren	Gebruiker in de ProSuite
Aannamen	De gebruiker is binnen het systeem geregistreerd. De gebruiker heeft de rechten: Mag artikelen lezen, Mag artikelen wijzigen, Mag gebruikersgroepen opvragen
Beschrijving	Na een artikel geselecteerd te hebben kan de gebruiker een overzicht van gebruikersgroepen oproepen en hier een selectie maken. Na de knop toevoegen te klikken zal de geselecteerde gebruikersgroep aan het artikel worden toegevoegd.
Uitzondering	Als de gebruiker niet de juiste rechten heeft.
Resultaat	De gebruikersgroep is aan het artikel toegekend

### **use-case 3: Gebruikersgroep toevoegen aan artikel**

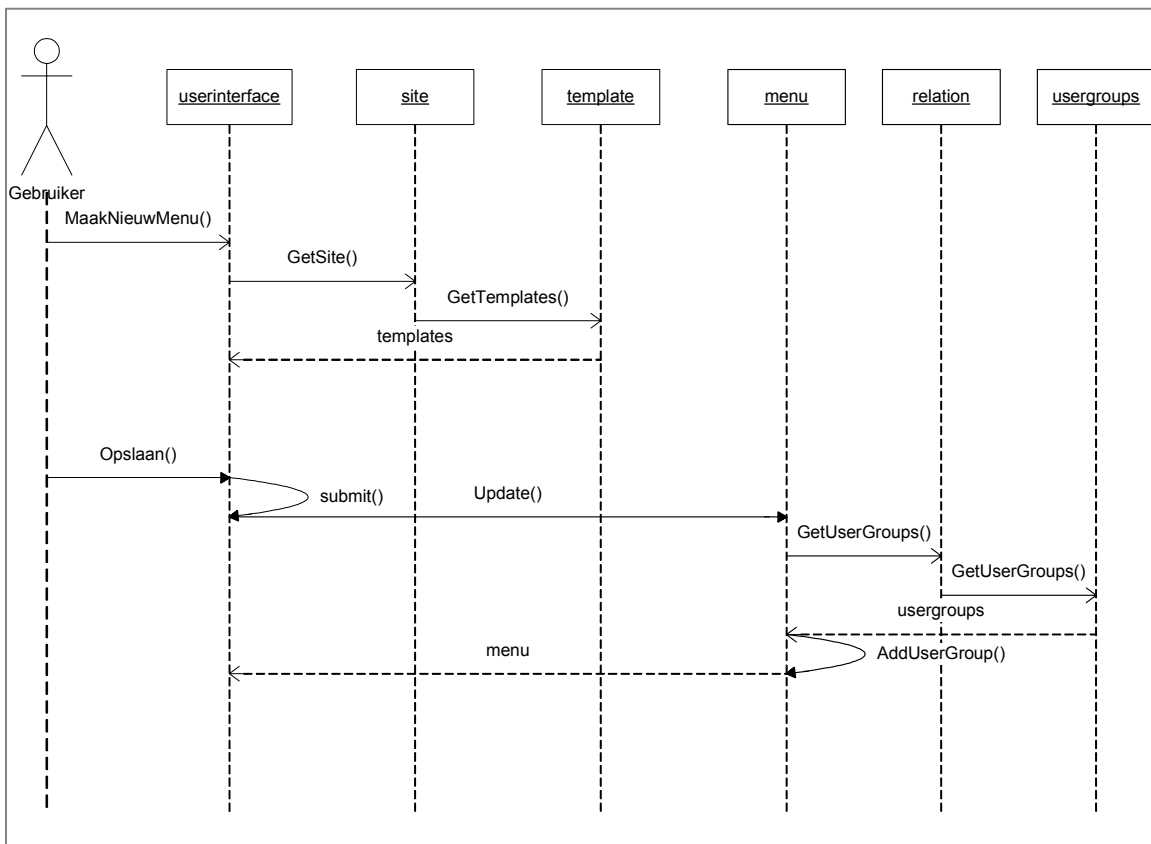
Naam	Opvragen gebruikersgroepen per artikel
Samenvatting	Een gebruiker kan de gebruikersgroepen per artikel opvragen
Actoren	Gebruiker in de ProSuite
Aannamen	De gebruiker is binnen het systeem geregistreerd. De gebruiker heeft de rechten: Mag artikelen lezen en Mag gebruikersgroepen opvragen
Beschrijving	Als een gebruiker een artikel bekijkt dan kan hij ook de gekoppelde gebruikersgroepen raadplegen. Dit wordt in een overzicht weergegeven.
Uitzondering	Als de gebruiker niet de juiste rechten heeft.

Resultaat	Een overzicht van gebruikersgroepen
-----------	-------------------------------------

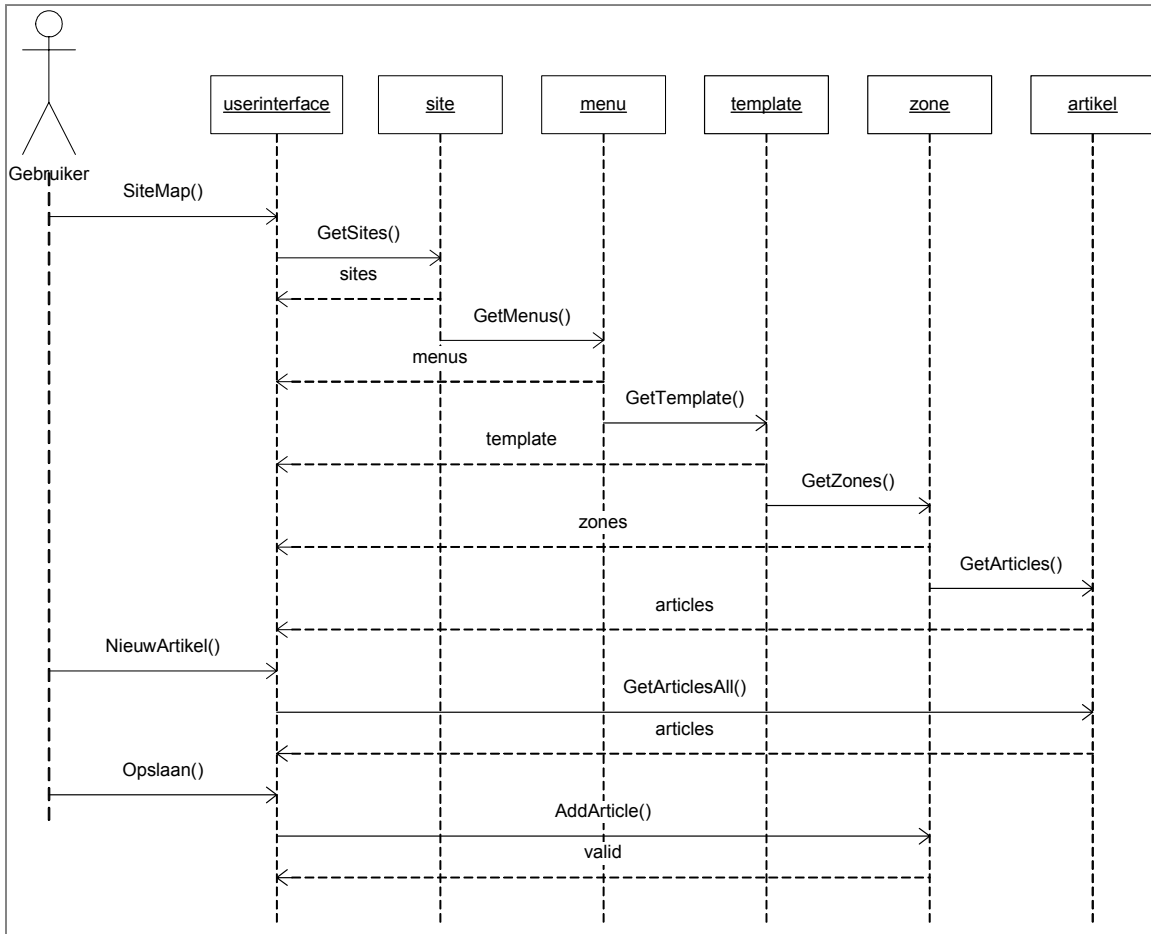
## use-case 4: Opvragen gebruikersgroepen per artikel

### Sequentiediagrammen

Sequentiediagram van: use-case 1: Menu toevoegen aan site

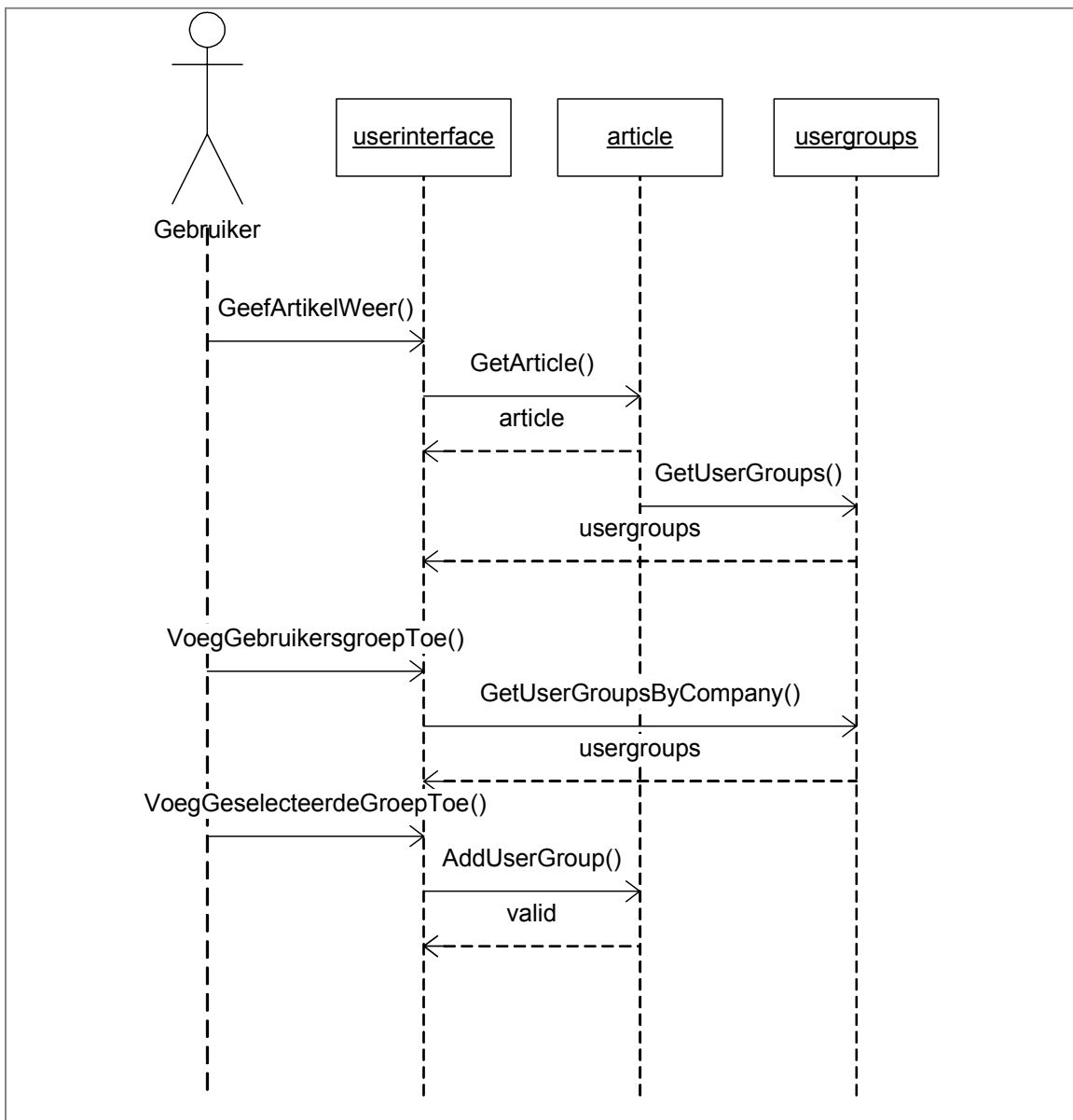


## Sequentiedigram van: use-case 2: Een bestaand artikel plaatsen op site



Sequentiediagram van: use-case 3: Gebruikersgroep toevoegen aan artikel en use-case 4: Opvragen gebruikersgroepen per artikel

Ik heb use-case 3 en use-case 4 samen in een sequentiediagram verwerkt, omdat deze dicht bij elkaar liggen



## Rechtenstructuur Cms

Het gebruikers- en gebruikersgroepenmodel dat het Cms gaat gebruiken werkt aan de hand van een principal. Dit is een functie in ASP.NET die voorziet in een systeem voor het inloggen van gebruikers en toekennen van rechten aan een gebruiker. Een recht wordt gezien als het recht op het wijzigen van een artikel. De rechten die een gebruiker heeft kan gebruikt worden om te bepalen wat hij wel en niet mag in het Cms. Veel van deze rechten worden ingesteld in de gebruikersinterface door middel van het wel of niet weergeven van knoppen om bijv. te wijzigen of te verwijderen.

De rechten die het Cms kent zijn:

- Menu
  - Mag menu's lezen
  - Mag menu's aanmaken
  - Mag menu's wijzigen
  - Mag menu's verwijderen
  - Mag volgorde menu's wijzigen
  - Mag artikelen toevoegen aan menu's
- Banners
  - Mag banners lezen
  - Mag banners aanmaken
  - Mag banners wijzigen
  - Mag banners verwijderen
- Artikel
  - Mag artikelen lezen
  - Mag artikelen aanmaken
  - Mag artikelen wijzigen
  - Mag artikelen verwijderen
  - Mag volgorde artikelen wijzigen
- Kaartmanager

- Mag locatiepunt lezen
  - Mag locatiepunt aanmaken
  - Mag locatiepunt wijzigen
  - Mag locatiepunt verwijderen
- Sitemap
  - Mag sitemap lezen
- Admin algemeen
  - Mag status lezen/ wijzigen/ verwijderen
- Gebruikers
  - Mag gebruikers lezen
  - Mag gebruikers aanmaken
  - Mag gebruikers wijzigen
  - Mag gebruikers verwijderen
- Gebruikersgroepen
  - Mag gebruikersgroepen opvragen
  - Mag gebruikersgroepen aanmaken
  - Mag gebruikersgroepen wijzigen
  - Mag gebruikersgroepen verwijderen

Bijlage: **Rapporten besprekingen**

## **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**



## Inhoudsopgave

Inhoudsopgave .....	i
Rapport pilotverdeling.....	1
Rapport projectmeeting Interface & stakeholders .....	3
Rapport Interface veranderingen a.d.h.v. review .....	5
Rapport verschillende soorten uiterlijk .....	7
Rapport gesprek NVvHP .....	10
Rapport review na terugkomst R. Dolle .....	12

## Rapport pilotverdeling

Datum: 1 maart 2004

Aanwezig: Ritsaert Dolle, Bram de Jager

### - Pilot verdeling

Welke pilots worden er ontwikkeld? En wat is het doel van de pilots.

### - Pilot 1 (Basis Cms met transparante interface)

Functionaliteit: Menu aanmaken/ wijzigen/ verwijderen, Artikelen aanmaken/ wijzigen/ verwijderen, Inlog mogelijkheid ADMIN.

Dit is een basis CMS en kan de "folder" worden onderhouden.

Functionaliteit: Het kunnen plakken van het CMS achter elke interface

Welke interfaces zijn er. Onderzoek doen via Internet naar verschillende interfaces.

Opsomming maken van 10 verschillende interfaces en daaruit wellicht een selectie maken voor CMS. A.d.h.v. de selectie kunnen verschillende interfaces vormgegeven worden.

Moeilijkheid: Koppeling interfaces

### - Pilot 2 (Gebruikers en gebruikersgroepen)

Functionaliteit: Gebruikers aanmaken/ wijzigen/ verwijderen, Gebruikersgroepen aanmaken/ wijzigen/ verwijderen

Stakeholder definiëren voor duidelijke opbouw van rechtenstructuur. Rekening houden met pilot 4 (koppeling Portal)

Moeilijkheid: Duidelijke rechtenstructuur die overal inzetbaar is.

### - Pilot 3 (Modulaire opbouw; modules voor CMS ZONDER koppeling Portal\*)

Functionaliteit: Losse modules moeten toegevoegd



Modules kunnen los ontwikkeld worden voor CMS en moeten zonder aanpassingen van totale systeem toegevoegd aan het CMS. Voorbeeld modules zijn: poll, feedback-formulier (via mail), evenementen kalender.

Moeilijkheid: Modulair opzetten van modules die ZONDER aanpassingen toegevoegd kunnen worden

- Pilot 4 (Modulaire opbouw; modules voor CMS MET koppeling Portal\*)

Functionaliteit: Modules van AccountPro/ MoversPro/ SecurityPro/ HrmPro/ ... kunnen in CMS worden toegevoegd. Dit wordt CmsPro 5.1.

Het is mogelijk voor stakeholders om toegang te krijgen tot delen van applicaties. Er wordt een koppeling gemaakt met de applicaties.

Moeilijkheid: Koppelingen moeten mogelijk zijn en interface verschil van Cms (website) en applicatie

Planning:

- Eindverslag:

Maken van hoofdstukindeling voor eindverslag. Op basis van verslag Jonathan.

- Bepalen stakeholders en interfaces:

Overleggen plannen voor definitie stakeholders/ interfaces waarop gebaseerd moet worden.

- Projectsgroep samenstellen:

Wie waarvoor en waarom. Welke klanten gebruiken voor testen. Bespreken met Ritsaert.

- Testplan opstellen:

Testplan voor pilot 1 maken dat deze getest kan worden voor deel projectgroep

\*) Definitie portal: CMS als basis voor alle applicaties samengevoegd in een portal. Dus CmsPro/ AccountPro/ MoversPro/ SecurityPro/ HrmPro/ ...

## Rapport projectmeeting Interface & stakeholders

Datum: 3 maart 2004

Aanwezig: Ritsaert Dolle, Hanife Bal, Jonathan Veldhuizen, Wesley Sussenbach,  
Marcel van Wijk, Bram de Jager

Afwezig: Derek Roos

### Stakeholders:

Er zijn geen uitbreidingen op de stakeholders. Er was wat onduidelijkheid van de definitie stakeholder en discussie is dan gesloten zonder toevoeging. Lijst was compleet. Zoals nu rekening gehouden is met de verschillende rechten die er bestaan of komende tijd worden toegevoegd hoeft het Cms niet aangepast worden.

### Interfaces:

Zelf had ik er 7 verschillende opties (Webwinkel, Krant, Portal, Folder, dropdown menu, Folder, hoofd-/ submenu, Zoekmachine, Meerdere sites gekoppeld), daar zijn er 6 aan toegevoegd (Startpagina/ geen menu, Flash pagina's, Homepage + verder pagina, Hybride systemen, Specifiek, Dolles S model).

Nu zal er een duidelijk beeld moeten ontstaan a.d.h.v. de verschillende interfaces. Er is een selectielijst gemaakt van website elementen. Welke elementen op welke interfaces aanwezig zijn. Daarna kan er een keuze worden gemaakt welke elementen onderdeel van het Cms moeten vormen.

### Pilots:

Pilots zijn gewijzigd. Eerste planning had pilot 3: Modulair website systeem en pilot 4: koppeling met systemen. Deze zijn omgedraaid omdat er meer problemen verwacht worden met de koppeling als het maken van modulaire pilots.

### Instellingen voor interface:

Basic en advanced instellingen! Het probleem van Pilot 2 met de gebruikers en gebruikersrechten is de gebruikersinterface. Is het model duidelijk voor de

webmaster. Kan hij rechten aanmaken verwijderen etc. Er is in overleg met de opdrachtgever besloten een verdeling te maken in de instellingen. Basic en advanced instellingen. Zowel in de menustructuur van de backoffice (menu, artikelen, gebruikers, gebruikersrechten) als per artikel, menu, gebruikersrechten, etc. We kunnen EFR gebruiken als test voor de basic en advanced instellingen. Wellicht mijn ouders als leken voor de test.

#### Planning:

- Interface-elementen inventariseren
- Basic en advanced instellingen verwerken in het Cms ? testen met groep
- Interface van de backoffice aanpassen aan die van MoversPro
- EFR gebruiken als testcase
- Plan van aanpak afmaken met planning
- Verslag pilotplan 1 documenteren globaal verloop

## Rapport Test pilot 1 en pilot 2

Datum : 8 maart 2004

Aanwezig: Derek Roos, Bram de Jager

Afwezig: -

Het gebruik van het Cms is eenvoudig. Hierdoor kunnen veel mensen werken met het systeem. De grens van kennis voor een gebruiker ligt laag. Menu weergave van het Cms (menu's, artikelen, gebruikers, gebruikersgroepen, archief) zal fraaier gemaakt moeten worden. D.m.v. iconen kan het vriendelijker overkomen. Er werken een aantal functies niet of zij ontbreken, deze staan hieronder.

### HOT ISSUES (in mate van urgentie)

- admin login is voor beide admins hetzelfde?
- kan niet uitloggen in het CMS.
- menu-admin werkt niet; hoe maak ik een nieuw item aan?
- link-edit + link aanmaken geeft error.
- meertaligheid.
- fileupload (generieke functionaliteit).
- contactformulier (generieke functionaliteit).
- plaatsen van artikelen in een webarchief (dus niet meer zichtbaar maar wel nog in de database en via zoekfunctie).
- zoekmachine: zoeken op content. Genereert lijst met artikelen. Klikken op artikel opent artikel in de site. Gekoppeld aan rechten.
- webwinkel: even overleg over mogelijkheden.

## **Rapport Interface veranderingen a.d.h.v. review**

Datum: 8 maart 2004

Aanwezig: Derek Roos, Bram de Jager

Afwezig: -

Momenteel zijn de menu items en hun artikelen visueel gescheiden. Het is duidelijker om deze bij elkaar te voegen. Door aan de linkse kant van het scherm een menustructuur weer te geven en daar rechts van de onderliggende artikelen. Dit zelfde is door te voeren met de gebruikers.

Het toewijzen van een artikel kan dan door een archief van artikelen te hebben en dan onder een tab te schuiven. Maar dit kan misschien ook met drag en drop. Maar of dit mogelijk is en haalbaar is een vraag. Om dit uit te zoeken zal eerst bekeken moeten worden is het haalbaar en in te passen in mijn product. Drag en drop is wel interessant voor verdere ontwikkelingen van alle applicaties, maar dat is niet voor mij interessant als proces.

Artikelen moeten aan nieuwsitems te koppelen zijn. Elk artikel moet een samenvatting hebben als nieuwskop. Maar het moet wel te activeren zijn vanuit de web.config bijvoorbeeld voor de verschillende interfaces. Dit moet dan ook terug te vinden zijn in de elementen van de sites. Zie onderzoek.

## Rapport verschillende soorten uiterlijk

Datum: 16 maart 2004

Aanwezig: Bram de Jager

Afwezig: -

Voor de grafische weergave van menu's zijn er meerdere soorten menu's te gebruiken. Voorbeelden zijn dropdownmenu, uitschuifmenu, etc.

Dropdownmenu:

Zie onderstaand voorbeeld.





Uitschuifmenu:

Zie onderstaand voorbeeld.



Hoofdmenu's en submenu's



Meerdere plaatsen:

Of een combinatie van verschillende soorten menu's op verschillende plaatsen.

De belangrijkste conclusie die momenteel wordt gemaakt a.d.h.v. bespreking met de projectgroep is er voor gekozen maximaal een hoofdmenu en maximaal een submenu te gebruiken. Althans dit alles heeft betrekking op de grafische locatie van de menu's. Er zal namelijk geen derde locatie zijn van een subsubmenu.

Voorbeeld:



Mocht het voorkomen dat een derde locatie nodig is als afhankelijke van een tweede locatie dan wordt dit maatwerk.

## Rapport gesprek NVvHP

Datum: 2 april 2004

Aanwezig: Peter Vriezen, Kees van der Wees, Derek Roos, Bram de Jager

Afwezig: -

Er heeft een gesprek plaats gevonden met een NVvHP/Hoofdpijn.

Aan de hand van het gesprek kan ik verder de uitbreiding van mijn functionele eisen en het is tevens een test voor de huidige functionele en technische eisen.

Het systeem is al in gebruik en zou ook voor deze klant toepasbaar moeten zijn.

Voor een complete beschrijving van de klant kan op

//Service/\_Sales/Ondersteuningsbureau/Aangeleverde bestanden worden gekeken.

Hieronder een listing van de belangrijkste functionele eisen en idee tijdens het gesprek opgedaan.

- Browser ondersteuning (min. IE 5.0 en andere browsers)
- Via een printknop een artikel kunnen printen
- Upload: bij meerdere bedrijven moet je meerdere directories hebben. Hoe kom je bij de locatie op de site in de upload vanuit de ProSuite die ergens anders gehost word.
- Beveiliging: virus probleem. Bestanden die geupload worden moet misschien gescand worden. Of een beperking van het soort bestanden dat je kan uploaden.
- Simpele boekhoudmodule: frontoffice declaratiemodule, backoffice moet balans, totaal bedrag begroting, etc bevatten
- Login: registreer --> admin moet recht aan gebruiker geven voordat deze op de site mag inloggen
- Login: backoffice/ frontoffice: wanneer wil je op de backoffice inloggen en wanneer moet je terug geleid worden naar de frontoffice.
- Agenda module: meerdere soorten agenda's. Meldingen bij het inplannen van activiteiten als er meerdere activiteiten gepland staan op die dag.

- Lees meer: als hierop wordt geklikt en men wil naar een artikel in het beveiligde gedeelte dan moet er gelinkt worden naar de login.
- Zij willen een koppeling met SOAP. Een CRM wat zij nu gebruiken. Wat voor een database is dat en kan die koppeling worden gemaakt met de nieuwe datalaag.
- Declaratieformulier gaat altijd gepaard met fysieke bonnen. Deze zullen opgestuurd moeten worden.
- Discussiegedeelte moet achter het beveiligde gedeelte. En misschien gepaard met het recht van de gebruiker. Kan hij lezen, etc etc
- Documenten beheer module: op verschillende niveaus.  
Organisatie/Deelorganisatie/groepen/subgroepen/gebruiker
- Feedbackformulier dynamisch genereren staat in .NET magazine #3
- FCK-editor moet aangepast worden. Formulier erin.
- FAQ-module
- Statistieken

## Rapport review na terugkomst R. Dolle

Datum: 7 mei 2004

Aanwezig: Ritseart Dolle, Derek Roos, Bram de Jager

Afwezig: -

Er is een demo gegeven door mij van CmsPro. Nu zijn zowel Ritseart als Derek weer op de hoogte van de laatste veranderingen.

Belangrijkste wijzigingen waren:

- TreeView weergave van de sitemap
- Het gehele proces van het plaatsen van artikelen/ banners in de site

Ik had een aantal vragen welke tijdens het ontwerpen naar boven waren gekomen. Deze vragen zullen hieronder puntsgewijs besproken worden.

[Article] Weergave van locaties in de artikel edit

Locaties werden toegevoegd vanuit de artikel edit. Dit wordt nu gedaan in de sitemap. De vraag of de locaties nu wel of niet weergegeven moeten worden in de edit. Ja, voorlopig wel. Als het onduidelijk wordt dan kan het verwijderd worden. Het is enkel een overzicht en de edit functie ervan is niet meer mogelijk.

[Archive/ status] Archief is nu een van de modules, moet deze voor alle modules worden?

Archief heeft nu alleen betrekking op artikelen. Maar wellicht moeten banners ook gearchiveerd gaan worden.

Banners moeten inderdaad ook gearchiveerd gaan worden dus er kan een dropdownbox geplaatst worden waarin gekozen kan worden: "in archief" / "actueel" / "alles".

Maar een stap verder is een status aan een artikel te koppelen. Deze vraag kwam van klanten om een redactie-systeem op te zetten. Een werknemer maakt een artikel [status: nieuw]. Na bewerken wordt is het klaar voor controle [status: te

publiceren]. Het wordt gecontroleerd en kan gepubliceerd worden [status: publicatie]. Archiveren kan dan een van de statussen zijn [status: archief]. Deze status moet dynamisch te beheren zijn. Daarnaast is een bepaalde status te koppelen aan gebruikersgroepen.

[Banners] Weergave banners kan in drie vormen, maar welke te gebruiken?

Mogelijkheid 1: Een banner wordt gekoppeld aan een zone. Is de zone aanwezig op een template dan is de banner zichtbaar ongeacht het menu wat geselecteerd is.

Mogelijkheid 2: Een banner wordt gekoppeld aan een zone en menu. Nu heeft de banner dezelfde eigenschappen als een artikel. Afhankelijk van de zone en het gekozen menu.

Mogelijkheid 3: Een banner wordt gekoppeld aan een zone en menu. Dezelfde eigenschappen als mogelijkheid 2, maar nu kan er bij een banner aangegeven worden of deze altijd zichtbaar is. Een groot probleem hiervan is de koppeling in de database.

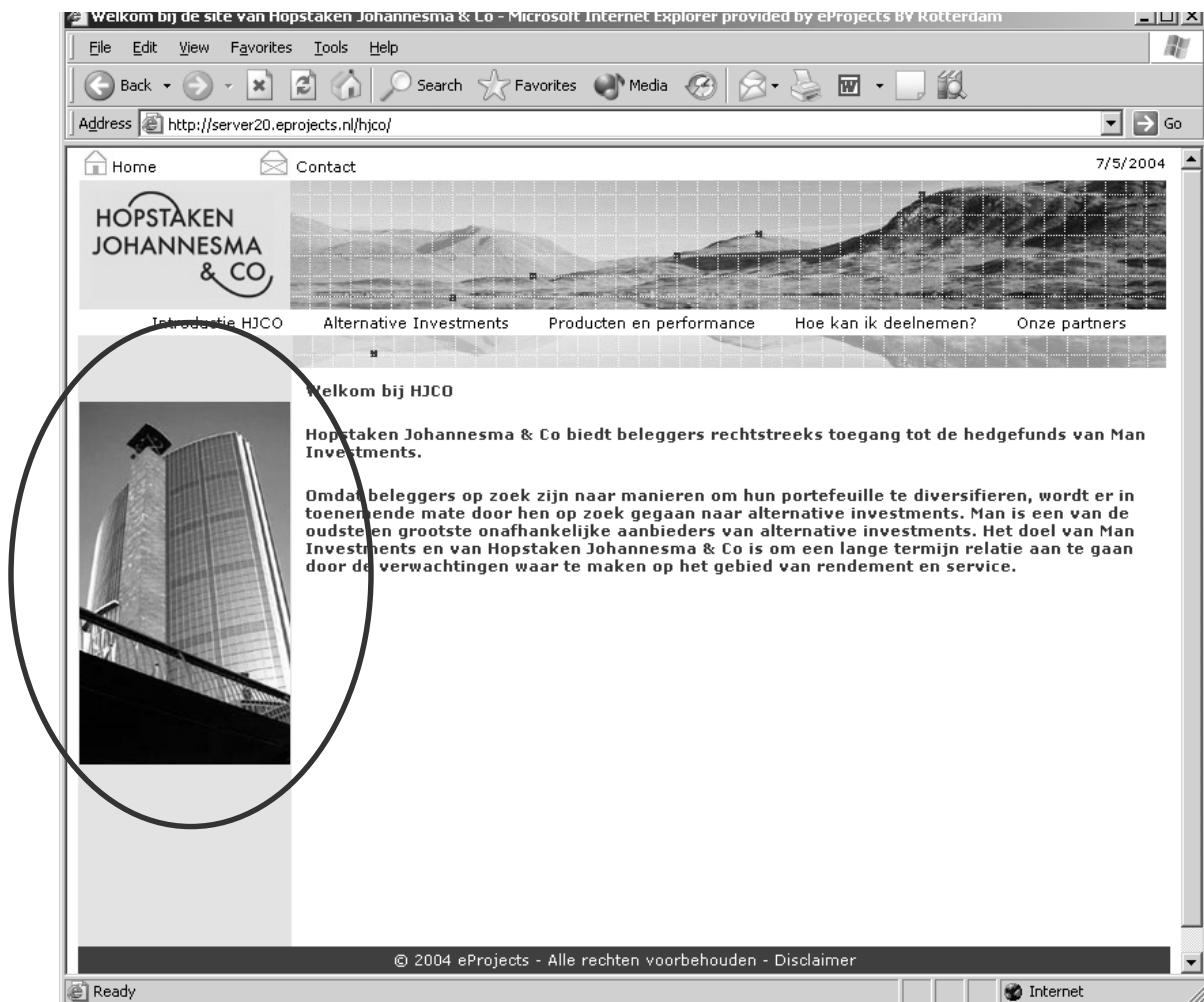
Oplossing probleem mogelijkheid 3:

MenuID	ZoneID	BannerID	Order	AltijdZichtbaar
6	1	12	1	No
6	1	5	2	No
	1	23	2	Yes
24	1	51	1	No
	1	7	1	Yes

Door gebruik te maken van een extra kolom kan aangegeven worden wanneer de banner zichtbaar is. Maar een moeilijkheid treed op met de volgorde van de banners. Altijd zichtbaar houdt in dat wanneer de zone (zoneID = 1) aanwezig is op de template de banner met AltijdZichtbaar = yes geladen moet worden. Maar deze banner heeft een volgorde samen met andere banners van die zone en altijd zichtbaar. De andere volgorde is van de banners die aan die zone en een menu zijn gekoppeld.

De keus is gevallen op mogelijkheid 1. Een banner is zichtbaar wanneer de zone aanwezig is op de template. Banners dienen als sponsoring/ reclame en is dus zichtbaar op een site. De extra koppeling met menu is overbodig.

Probleem bij HJCO. Deze willen een plaatje bij een menu, maar dat plaatje staat op een vaste plaats.



Dat plaatje moet veranderen als er een menu wordt gekozen. Het wordt een zone van het type artikel (met weergave van de content). Voor elk plaatje maak je een



nieuw artikel aan met enkel een plaatje. Dit plaatje plaats je per menu in die zone (locatie)

[Meta] Uit zoeken hoe het mogelijk is meta informatie mee te geven aan een artikel. Het moeilijke is hier dat zoekmachines zoeken in de HTML en de artikelen zitten in de database.

[Site] Actieve site

Bij het bewerken van meerdere sites in het Cms moet er ergens worden aangeven in welke site je aan het werken bent.

Dit kan a.d.h.v. een dropdownlist. Dit is een makkelijke oplossing voorlopig die werkt, waarna eventueel dit anders opgelost kan worden.

Dropdown in article.aspx

[SitePath] Is het mogelijk dat in sitemap.aspx op de verschillende menu's geklikt kan worden om zodoende hier direct naartoe te gaan. ("u bent hier")

Het teken "\" vervangen door ">"

[Legenda] Is de plaats van de legenda goed?

Ja

[Bestandsbeheer] Het beheren van een site is geheel database driven. Het benaderen van een database kan vanuit overall. Maar bij bestanden is dit anders. Willen we bestandsbeheer gaan toepassen dan moet er toegang zijn op die directory. Het benaderen van directories kan alleen als je op dezelfde server zit.

Aangezien dat de prosuite (beheer) voornamelijk op dezelfde server zal staan als de site is dit geen probleem.

Een oplossing zou zijn bestanden via een webservice uploaden. Maar dat is een luxe eis.



[SaveContent] Bij het aanmaken van een nieuw item (artikel/ menu) en daarna opslaan, wijziging maken en wederom opslaan dan wordt het item voor een tweede keer in de database opgeslagen.

[Kopie artikel] Het moet mogelijk zijn een kopie te maken van een artikel. Dit moet een extra knop worden welke een nieuw venster opent met daarin de mogelijkheid dit artikel naar een site te kopiëren en de status van het artikel tegelijk in te stellen.

[SiteMap] Naast het toevoegen van een locatie moet het mogelijk zijn een nieuw item aan te maken in de database.

Dit wordt een extra icoon met toevoegen item (artikel/ banner/ menu)

[SiteMap] Toevoegen van locaties van een item wordt een probleem als er 1000 artikelen bevinden in de site. En welke site is geselecteerd?

Dit is op te lossen door de dropdown te verwijderen en te vervangen door een popup met daarin een duidelijke structuur voor de listing van de artikelen.

[Module, pathcontrol] Voor uiterlijk site moet er een module geplaatst kunnen worden met daarin het path van het menu.

[Toolbox] Is het mogelijk een toolbox te maken met alle iconen voor acties.(Ritseart) Gebruik maken van de header.aspx voor alle iconen. Vanuit hier alle acties aanroepen. Kan dat en alleen uitvoeren als het weinig werk is.

Niet besproken:

[Weergave zones in templates] Bij het aanmaken van een menu selecteer je een template. Aan de hand van de template krijgt het menu een bepaalt uiterlijk. Per zone kan je bepalen wat het uiterlijk is, maar dit wordt vastgelegd in de template. Dit is dus niet dynamisch! Het kan dynamisch als je na het vastleggen de template de gebruiker per zone laat instellen wat het uiterlijk van deze zone zou moeten zijn.

Bijlage: **Beheerdershandleiding**

# **Ontwikkeling van een nieuwe generatie Content Management Systeem bij eProjects**



## Inhoudsopgave

Inhoudsopgave .....	i
Solution in Visual Studio .....	1
Files downloaden uit CVS Tortoise .....	1
Solution aanmaken in Visual Studio .....	2
Aanmaken virtual directory .....	2
Visual studio .....	3
Design maken.....	6
Opbouw .....	6
Modules.....	6
Standaard bestanden .....	6
Default.aspx .....	6
DefaultHeader.aspx .....	7
DefaultFooter.aspx .....	7



## Solution in Visual Studio

### Files downloaden uit CVS Tortoise

Je maakt op je harde schijf een directory aan met de naam van je project. In deze directory maak je een directory aan met de naam HTML.

Vb: D:\projecten\eProjects\HTML

Rechtermuisknop op de HTML directory en selecteer CVS → Make new module

Je stelt de volgende velden in:

Protocol:	Locally mounted folder (:Local)
Repository folder:	\\strauss\Afdelingen\Development\Projecten\CMSPro 4.0\source-repository
Module:	HTML

Door op OK te drukken worden de bestanden opgehaald uit het CVS.

Herhaald dit voor de database:

Maak een directory aan in de project directory met de naam "Database". Ook hier gebruik je CVS en maak je een nieuwe module aan.

Je stelt de volgende velden in:

Protocol:	Locally mounted folder (:Local)
Repository folder:	\\strauss\Afdelingen\Development\Projecten\CMSPro 4.0\source-repository
Module:	Database



## **Solution aanmaken in Visual Studio**

### **Aanmaken virtual directory**

De volgende stappen zijn in Internet Information Service.

Stap 1: Start je IIS server op door rechtermuisknop op "My computer", waarna "Manage" te selecteren. Te volgen pad in "Computer management": Services and applications → Internet Information Service → Web sites → Default web site

Stap 2: Hier moet een virtuele directory aangemaakt worden. Rechtermuisknop op "Default web site" en "New → Virtual directory".

Geef de alias van de website op. LET OP: Gebruik dezelfde naam als de zodirect te gebruiken projectnaam in Visual Studio.

Vb: eProjectsCms

Stap 3: De directory waar naar verwezen moet worden is de HTML directory in je project directory.

Vb: D:\projecten\eProjects\HTML

## Visual studio

De volgende stappen zijn in Visual Studio.

### Stap 1: Aanmaken van de solution

Start Visual Studio en maak een nieuwe solution aan. File → New → Blanc solution.

Bij "name" vul je de naam van de project directory in en "Location" geef je de directory aan waar de project directory zich bevind.

Vb: Name: eProjects en Location: D:\projecten\ → De solution wordt nu in de directory D:\projecten\ aangemaakt

### Stap 2: Aanmaken van een project

Maak een nieuw project aan in je Solution Explorer. "Solution → Add → New project". Selecteer "Visual C# projects" en geef aan dat je een "ASP.NET Web application" wilt maken. Bij location geef je de eerder aangemaakte virtual directory aan.

Na het klikken van de OK-knop krijg je een melding dat de "Global.asax" al bestaat. Er wordt gevraagd of het bestaande global.asax bestand overschreven mag worden.

**NEE**, je hebt het oude bestand nodig. Daarna een bevestiging met "**OK**"

Vb: <http://localhost/eProjectsCms>

### Stap 3: Instellen projecteigenschappen

In de Solution Explorer kan je de eigenschappen van het project opvragen. Stel de volgende eigenschappen in:

In common properties → General

- Assembly name: eProjects.Web.*[naam van het project]*
- Default namespace: eProjects.Web.*[naam van het project]*

In Configuration properties → Debugging

- Start page: Default.aspx



#### Stap 4: Referenties toevoegen

Er moeten verschillende referenties toegevoegd worden aan het project. In de solution explorer kan je bij je project referenties toevoegen "References → Add reference". Je moet browsen naar je bin map van je HTML directory. Hierin selecteer je de volgende bestanden:

- eProjects.Cms.dll
- eProjects.Common.dll
- eProjects.Web.Cms.dll
- FredCK.FCKeditor.dll
- solpartwebcontrols.dll

Vb: D:\projecten\eProjects\HTML\bin\

#### Stap 5: Include bestanden

Er zijn in praktijk maar drie bestanden nodig om te includen. Dit zijn de bestanden die het uiterlijk van de site bepalen.

- Default.aspx
- DefaultHeader.aspx
- DefaultFooter.aspx

Als je deze bestanden include in het project dan kan gaan werken aan het design.



## Stap 6: Instellen web.config

In de web.config stel je verschillende variabelen in die nodig zijn voor het laten werken van de site.

DatabaseSettingsPath: is een referentie naar het XML-bestand met de verschillende settings.

- Connectionstring: Geeft aan waar de database van het project staat.
- Submenuactive: Hiermee kan je het submenu activeren. Het submenu is de module die afhankelijk van het hoofdmenu geladen wordt. Staat deze op false dan wordt gebruik gemaakt van het hoofdmenu met dropdown.
- ZonesActive: Zet deze standaard op false, kan je gebruiken als je op meerdere plaatsen dezelfde module hebt staan. Elke module staat in een zone.
- UploadPath: Dit is het path naar de upload directory voor de upload van bestanden in de backoffice
- SendMailAddress: Geeft aan waar het feedbackformulier zijn response naar moet sturen.

```
<appSettings>
  <add key="DatabaseSettingsPath" value="D:\Projecten\eProjects\Html\" />
  <add key="connectionstring" value="D:\Projecten\eProjects\Database\cmspro.mdb" />
  <add key="submenuActive" value="true"/>
  <add key="zonesActive" value="true"/>
  <add key="uploadPath" value="/Upload/" />
  <add key="sendMailAddress" value="service@moverspro.com" />
</appSettings>
```



## Design maken

### Opbouw

Een site bestaat uit een opbouw in tabellen. Deze layout kan je geheel zelf bepalen, maar als je gebruik wil maken van modules dan moeten deze geplaatst worden in panels. (zie Default.aspx)

Als je de site wilt bekijken zal je eerst de site moeten compileren/ build in Visual Studio.

### Modules

- Artikelen: Hierin worden artikelen in een repeater opgebouwd
- Nieuws: Verschillende nieuwsberichten worden weergegeven met een mogelijke link naar de site of andere site.
- Hoofdmenu: Dit bevat de hoofdmenuitems. Als in de web.config de submenuactive = false staat worden hier de onderliggende menu's getoond in een dropdown.
- Submenu: Dit zijn de submenu's afhankelijk van het gekozen hoofdmenu.  
Kan je alleen gebruiken i.c.m. de web.config instelling submenuactive = true.

## Standaard bestanden

### Default.aspx

De default.aspx wordt al eerste geladen en kan twee user controls bevatten, te weten defaultheader.ascx en defaultfooter.ascx.

In dit bestand maak je de opmaak van de site. Je kan deze file helemaal veranderen en opnieuw opbouwen.

Door verschillende modules erop te slepen uit de directory "modules" kan je artikelen, nieuwsberichten, etc plaatsen op je site. Deze modules moeten in panel geplaatst worden, met als naam de naam van een zone. Deze naam kan je zelf verzinnen, maar moet overeen komen met de naam in de database-tabel "Zones". Zodoende kan je bijv. artikelen toewijzen aan één of meerdere zones.

**DefaultHeader.aspx**

Deze file kan je gebruiken voor het hoofdmenu.

**DefaultFooter.aspx**

Deze file kan je gebruiken voor een standaard footer. Maar deze kan je ook op de default plaatsen.