Extra dienstverlening door Extranet

Student:	Pieter Visser
Studentnummer:	99004222
Opleiding:	VIA
Opleidingsinstituut:	Haagse Hogeschool
Examinatoren:	Joris Groen, Willem van Vliet
Stagebedrijf:	Xtilton
Bedrijfsmentoren:	Ron Jutte, Dick den Boef
Afstudeerperiode:	7 mei t/m 7 oktober 2005

- Afstudeerscriptie -

Referaat

Dit verslag beschrijft de werkzaamheden die door de student zijn uitgevoerd in het kader van de afstudeerstage bij de opleiding VIA aan de Haagse Hogeschool te Den Haag. Tijdens deze afstudeerstage is een extranet ontwikkeld.

Student: Pieter Visser (99004222)
Afstudeerperiode: 17 mei t/m 7 oktober 2005

Descriptoren:
Extranet
Content Management Systeem
Databases
PHP (programmeertaal)
IAD (Iterative Application Development)

Voorwoord

Ik wil hierbij op deze unieke manier aan geheel Xtilton mijn dank betuigen voor het feit dat ik er mijn afstudeerstage met plezier heb kunnen vervullen. Enkele personen wil ik met name bedanken: Theo Herpers, de directeur en opdrachtgever, voor de tijd, middelen en ruimte die ik kreeg om de opdracht te kunnen uitvoeren; Ron Jutte, mijn bedrijfsmentor, voor de excellente begeleiding tijdens het project, zijn inbreng in het ontwikkelproces en het mij bekend maken met de organisatie; Dick den Boef, latere vervanger van mijn bedrijfsmentor, voor het geduldig beantwoorden van allerlei vragen en het testen van componenten van het extranet; Joris Groen en Willem van Vliet, docenten van de Haagse Hogeschool en tevens examinatoren, voor hun informatie over en begeleiding tijdens het afstuderen.

Tot slot wil ik alle andere personen binnen Xtilton bedanken die hun medewerking hebben verleend aan het realiseren van mijn afstudeeropdracht.

Pieter Visser,
Monster, oktober 2005

Inhoudsopgave

1. INLEIDING 6

2. ORGANISATIE 7

2.1 Organisatie algemeen 7
2.2 Organisatie specifiek 7
2.3 De plaats van de afstudeerder binnen de organisatie 8

3. DE OPDRACHT 9

3.1 Opdrachtomschrijving 9
3.2 Aanleiding en probleemstelling 9
3.3 Doelstelling 9

4. HET MAKEN VAN EEN PLAN VAN AANPAK 10

4.1 Oriëntatie binnen bedrijf 10
4.2 Het bepalen van de scope van de opdracht 10
4.3 Methoden 13
4.3.1 IAD (Iterative Application Development) 13
4.3.2 DSDM (Dynamic Systems Development Method) 14
4.3.3 Verantwoording keuze voor IAD 15
4.4 De keuze voor een iteratiestrategie 16
4.5 Technieken 16
4.6 Planning, activiteiten en producten 17
4.6.1 Planning 17
4.6.2 Activiteiten 18
4.6.3 Op te leveren producten 18

5. DE FASE DEFINITIESTUDIE 19

5.1 De gebruikersgroepen analyseren 19
5.1.1 Het bepalen van de verschillende gebruikersgroepen 19
5.1.2 Informatie verzamelen over de gebruikersgroepen 20
5.1.3 De gebruikerstaken analyseren 21
5.2 Het opstellen van de systeemeisen 24
5.3 Het systeemconcept bepalen 25
5.3.1 De Inlogprocedure voor extranet bezoekers 25
5.3.2 Het beheer van het extranet 26
5.3.3 ICT problemen of vragen melden aan de servicedesk 27
5.4 De technische structuur bepalen 28
5.5 De organisatorische inrichting bepalen 31
5.6 De styleguide bepalen 31
5.7 Het pilotplan opstellen 33

6. DE FASE PILOTONTWIKKELING 35

6.1 Het bepalen van de iteratiestrategie 35
6.2 Het prioriteren van de pilots en pilotdelen 36
6.3 Pilot 1: de basisstructuur 37
6.3.1 Het opsplitsen van de pilot in pilotdelen 37

6.3.2 Pilotdeel 1: de database ... 38
6.3.3 Pilotdeel 2: het klantdeel ... 42
6.3.4 Pilotdeel 3: het Xtilton deel (CMS) ... 49
6.3.5 Pilotdeel 4: de vormgeving ... 58
6.3.6 Iteratie van de gehele pilot ... 59
6.4 Installatie op de server ... 64
6.5 Aanpassing van de planning ... 65
6.6 Pilot 2: het incidentensysteem ... 66
6.6.1 Het opsplitsen van de pilot in pilotdelen ... 66
6.6.2 Pilotdeel 1: de vormgeving ... 66
6.6.3 Pilotdeel 2: de helpfunctie ... 70
6.7 Het testen van het systeem door de klant ... 72
6.8 Pilot 3: het veelgestelde vragen systeem ... 73

7. DE FASE INVOERING ... 75

8. EVALUATIE ... 76
8.1 Productevaluatie ... 76
8.2 Procesevaluatie ... 77

BIJLAGEN
Bijlage A: Begrippenlijst ... 81
Bijlage B: Literatuurlijst ... 82
Bijlage C: Voorbeelden broncode PHP bestanden ... 83
Bijlage D: Schermafbeeldingen van het extranet ... 87
Bijlage E: Schermafbeeldingen van het CMS ... 88
Bijlage F: Cognitive Walkthrough formulier ... 89
Bijlage G: MySQL implementatiemodel ... 90

1. Inleiding

In het kader van de opleiding Vormgeving en Ontwerp van Interactie (VIA) aan de Haagse Hogeschool, heb ik in de periode van 17 mei tot 7 oktober 2005 een afstudeerstage gelopen bij het bedrijf Xtilton.

Gedurende deze periode heb ik gewerkt aan één enkele afstudeeropdracht. In dit verslag wordt beschreven wat deze opdracht was, hoe deze is uitgevoerd en wat ik er zelf van vond.

- In hoofdstuk 2 wordt kort het stagebedrijf beschreven.
- In hoofdstuk 3 wordt de opdracht beschreven.
- In hoofdstuk 4 wordt uitgelegd wat de scope van de opdracht en de planning was, welke ontwikkelmethodiek en technieken er gebruikt zijn en welke producten er opgeleverd dienden te worden.
- In hoofdstuk 5, 6 en 7 worden de verschillende fasen van het ontwikkelproces beschreven.
- In hoofdstuk 8 worden de procesgang en het eindproduct geëvalueerd.

Dit verslag heeft tot doel een helder beeld te scheppen van hoe de afstudeeropdracht doorlopen is, zodat dit getoetst kan worden aan de eisen die gesteld worden aan het afstuderen.

2. Organisatie

Dit hoofdstuk geeft een overzicht van de organisatie en mijn plek als afstudeerder hierbinnen.

2.1 Organisatie algemeen

Xtilton is een klein onafhankelijk bedrijf waar ongeveer 35 mensen werkzaam zijn. Zij verleent ICT-diensten voornamelijk in de zorgsector en één van haar grootste klanten is dan ook Parnassia, een grote GGZ-instelling in de regio Den Haag.

Tot 2001 was Xtilton een onderdeel van Parnassia, maar toen Parnassia besloot zich meer op de zorg te richten en andere activiteiten uit te besteden, is Xtilton een zelfstandige onderneming geworden. Het resultaat van deze geschiedenis is dat er bij Xtilton veel kennis aanwezig is over welke rol ICT speelt binnen de zorgsector, waardoor het kan inspelen op specifieke ICT-vraagstukken van zorginstellingen. De klanten van Xtilton blijven echter niet beperkt tot de zorgsector en omvatten bijvoorbeeld ook bibliotheken en transportbedrijven. Voor deze klanten verzorgt Xtilton de complete ICT-infrastructuur, wat afhankelijk van de klant kan variëren van vijf tot tweeduizend werkplekken.

2.2 Organisatie specifiek

Binnen Xtilton bestaan geen concrete afdelingen, maar wel groepen mensen die zich op een bepaalde taak richten. Hierbij zijn de verschillende groepen te onderscheiden:

De servicedesk

Dit is een helpdesk waar personeel van klanten terecht kan met ICT-problemen of vragen. Mocht een probleem hier niet opgelost kunnen worden, dan kan deze geëscaleerd worden naar de overige clusters waar de specialisten zich erover zullen buigen. Software die hier gebruikt wordt is Topdesk (om incidenten te registreren) en een programma om computers vanaf afstand over te nemen.

Applicatiebeheer

Applicatiebeheer is verantwoordelijk voor het goed functioneren van alle applicaties die draaien op de systemen die Xtilton ondersteunt. Als er problemen optreden in een applicatie, analyseert applicatiebeheer de situatie en zoekt er een oplossing voor. Ook het aanschaffen van nieuwe software en het updaten van software gebeurt hier.

Infrastructuur & netwerkbeheer

Deze sectie zorgt er voor dat het netwerk van Xtilton en haar klanten goed blijft functioneren. Om de week is er een persoon een week lang 24 uur per dag beschikbaar in geval van crisissituaties, als bijvoorbeeld een server met cliëntgegevens van een ziekenhuis er mee stopt, om hardwareproblemen op te lossen en nieuwe hardware te leveren en installeren bij klanten. Mocht iets volledig misgaan en Xtilton er niet meer uit komen, dan kan een extern bedrijf geraadpleegd worden.

Buitendienst

Dit zijn medewerkers die naar de klant gaan om hardware te installeren of werkplekondersteuning te bieden. De buitendienst wordt gedaan door servicedeskmedewerkers.

Telecom

Het plaatsen van telefoons, het integreren van telecom in netwerkomgevingen en het ondersteunen van de implementatie van nieuwe telefooncentrales.

Opleidingen

In Xtilton worden cursussen gegeven aan medewerkers van klanten. Deze cursussen gaan over bijvoorbeeld Windows, Word, Excel, Groupwise, etc. Ook de medewerkers van Xtilton zelf gaan van tijd tot tijd op cursus om bij te blijven met de laatste ontwikkelingen in de ICT. Deze cursussen worden echter niet door Xtilton gegeven. Het gaat hier bijvoorbeeld om cursussen om Microsoft of Novell certificaten te behalen.

Projecten

Projecten, zowel grote als kleine, worden hier ook uitgevoerd. Voorbeelden van projecten die uitgevoerd worden zijn het inrichten of verhuizen van de ICT-infrastructuur van een locatie en de selectie en implementeren van nieuwe systemen. Eigenlijk wordt alles wat niet bij de taken van de hiervoor genoemde groepen past als project beschouwd.

2.3 De plaats van de afstudeerder binnen de organisatie

Binnen Xtilton is geen afdeling die zich bezighoudt met het ontwikkelen van software. Personen die hier verstand van hebben zijn verspreid over de bovenstaande groepen. Op de plek waar ik als afstudeerder werkzaam was, waren zowel mensen van netwerkbeheer, applicatiebeheer, telecom en projecten en opleidingen werkzaam. Later kwam hier nog een andere stagiair bij die het Xtilton intranet zou gaan aanpassen. Er waren dus specialisten op verschillende gebieden aanwezig bij wie ik met vragen terecht kon. Het ontwikkelen van het extranet zou ik zelfstandig doen, waarbij mijn bedrijfsmentor als contactpersoon optrad en de directeur als opdrachtgever.

3. De opdracht

In dit hoofdstuk wordt uitgelegd wat de afstudeeropdracht inhoudt.

3.1 Opdrachtomschrijving

Kort samengevat luidt de omschrijving van de afstudeeropdracht als volgt:

- Ontwikkel een extranet waarmee de klanten van Xtilton verschillende soorten diensten en informatie kunnen raadplegen en dat door Xtilton medewerkers gemakkelijk beheerd en uitgebreid kan worden.

3.2 Aanleiding en probleemstelling

Xtilton verleent allerlei ICT diensten aan verscheidene klanten, zoals bijvoorbeeld het beheer van het netwerk, de levering en installatie van hard- en software en het oplossen van storingen en problemen.

Binnen Xtilton is er een servicedesk waar vragen en problemen van klanten afgehandeld worden. Hoewel er een grote diversiteit is in de aard van deze vragen, zitten er ook routinevragen tussen die telkens opnieuw opduiken en afgehandeld moeten worden. Ook komen deze vragen voornamelijk telefonisch binnen en moeten de medewerkers van de servicedesk deze handmatig registreren. Het gevolg hiervan is dat de servicedesk minder tijd beschikbaar heeft om andere services te bieden.

Tevens bestaat er binnen Xtilton de behoefte om verscheidene soorten informatie en diensten via een ander kanaal aan te bieden dan per email, telefoon of post.

Om deze reden is besloten een extranet te ontwikkelen, waarmee de klanten van Xtilton de servicedesk kunnen bereiken en andere diensten en informatie kunnen raadplegen. Via dit extranet dient Xtilton personeel ook gemakkelijker informatie en diensten te kunnen aanbieden aan klanten.

3.3 Doelstelling

Het doel van de afstudeeropdracht is het scheppen van kortere lijnen tussen de klanten en Xtilton en de klanten meer inzicht te bieden in de dienstverlening van Xtilton. Een ander doel is het ontlasten van de Servicedesk van Xtilton.

4. Het maken van een Plan van Aanpak

In dit hoofdstuk wordt beschreven hoe het Plan van Aanpak tot stand is gekomen.

4.1 Oriëntatie binnen bedrijf

Bij aanvang van mijn afstudeerstage heeft mijn bedrijfsmentor me uitgelegd hoe de organisatiestructuur van Xtilton in elkaar zit, wie wie is, wie wat doet, welke klanten Xtilton heeft en wat ze precies doen voor die klanten. Om hier een gedetailleerder beeld op te krijgen heeft hij mij de SLA (Service Level Agreement) tussen Xtilton en haar grootste klant (Parnassia) overhandigd.

Na afloop van dit gesprek heeft hij mij voorgesteld aan enkele collegae bij wie ik meer informatie met betrekking tot mijn afstudeeropdracht zou kunnen vinden. Ik had al reeds een opdrachtomschrijving voor school gemaakt, maar een compleet beeld van de opdracht had ik nog niet. Het onderzoeken van wat er precies van het extranet verwacht werd, maakte dan ook deel uit van de afstudeeropdracht. Tijdens deze eerste ontmoetingen met mijn collegae heb ik meteen enkele data afgesproken waarop ik hen zou kunnen interviewen.

4.2 Het bepalen van de scope van de opdracht

Om de scope van de afstudeeropdracht te bepalen (welke bij aanvang van de opdracht nog niet geheel bekend was) heb ik een onderzoek uitgevoerd naar wat men binnen het bedrijf wenste en verwachtte van het extranet. Hiertoe heb ik een aantal interviews afgenomen. Dit onderzoek was eveneens noodzakelijk om de verschillende delen van het Plan van Aanpak te kunnen opstellen.

Interview 1

Als eerste heb ik twee collegae geïnterviewd. Het doel van dit interview was het bepalen van de gewenste functionaliteiten van het extranet en de prioritering hiervan. Dit doel heb ik van tevoren ook aan hen meegedeeld zodat zij zich hierop konden voorbereiden. Bij aanvang van het interview hadden beide al enkele ideeën en is er gebrainstormd over andere mogelijkheden. Uiteindelijk kwamen we uit bij de volgende functionaliteiten.

- **Incidenten systeem**
  Hiermee kunnen gebruikers incidenten melden en inzien.
- **Formuliersysteem**
  Huidige papieren documenten tot webformulieren omzetten.
- **Veelgestelde vragen systeem**
  De antwoorden op vragen die vaak bij de servicedesk binnenkomen.

Na afloop van dit interview dacht ik een duidelijker overzicht van de afstudeeropdracht te hebben.

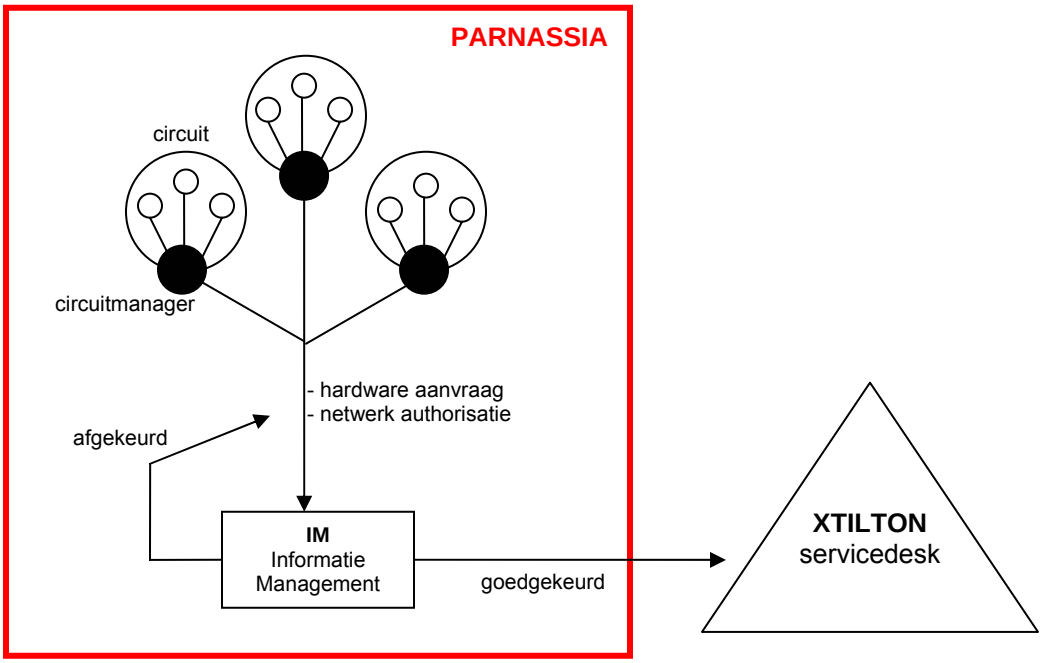
Interview 2

Voor meer informatie over deze functionaliteiten kon ik het best terecht bij het hoofd van de Servicedesk. Hij bevond zich op het operationele niveau, midden op de werkvloer, en had tevens een volledig overzicht van het geheel dat daar plaatsvond en de achterliggende processen die hierbij hoorden.

Het doel van dit tweede interview was het verkrijgen van wat meer informatie over bovenstaande functionaliteiten. Tijdens het interview kreeg ik enkele onverwachte antwoorden te horen. Er werd mij verteld dat er al een incidenten systeem bestaat (genaamd Topdesk Application Server,

oftewel TAS) en dat ik dit dus niet hoefde te ontwikkelen. Dit kwam als een verrassing voor mij en ik had op dat moment het gevoel dat mijn afstudeeropdracht in gevaar was. Na hierop doorgevraagd te hebben bleek dat dit systeem slechts door een handvol gebruikers (van de duizenden) daadwerkelijk gebruikt werd. Het was dus gewenst dat dit vaker gebruikt zou worden.

De tweede functionaliteit, het formuliersysteem, had betrekking op formulieren die bij de Servicedesk binnenkwamen via de email of post. Het bleek dat deze binnen de klant circuleerden en er bepaalde interne procedures bij gevolgd werden door de verschillende personen. Een voorbeeld:



Formulierstroom bij de klant Parnassia

In dit schema zijn de formulieren ‘netwerk autorisatie’ en ‘hardware aanvraag’ te zien. Wil een medewerker van Parnassia hardware aanvragen, dan dient hij eerst een formulier in te vullen en dit aan zijn circuitmanager te overhandigen. Deze controleert vervolgens of de prijs wel binnen de kostenplaats past en verleent al dan niet goedkeuring. Hierna stuurt hij dit formulier naar de informatiemanager, die eveneens goedkeuring dient te verlenen. Uiteindelijk komt alles via de informatiemanager bij Xtilton terecht.

Het was duidelijk dat vrijwel dit gehele proces zich intern bij de klant afspeelde. Er werd door de geïnterviewde dan ook aangegeven dat dit proces niet door Xtilton, noch mij als afstudeerder, aangepast kon worden. Hooguit kon ik de formulierstroom tussen de informatiemanager en Xtilton aanpassen, maar de geïnterviewde gaf aan dat hier geen behoefte aan was aan de kant van Parnassia. Bij de Xtilton servicedesk zou dit ook maar weinig tijdswinst opleveren per maand. Tot slot opperde hij nog twee nieuwe functionaliteiten:

- **Informatie over de Servicedesk**
  Hiermee kunnen de klanten inzien wat de servicedesk zoal doet en wie er werkt.
- **Enquêteresultaten**
  Het weergeven van de resultaten van enquêtes.

Interview 3

Als derde heb ik iemand van het tactische niveau geïnterviewd. Bij hem kon ik terecht voor informatie over de veelgestelde vragen. Voordat ik hierover begon, heb ik hem echter eerst gevraagd wat hij zelf van het extranet verwachtte en hierop terecht wilde zien komen. Dit deed ik, omdat bij mijn vorige interview al duidelijk was geworden dat niet iedereen dezelfde wensen had m.b.t. het extranet. Ik wilde nu eens zien wat ik van hem te horen kreeg als ik de keuze volledig aan hem liet en niet van te voren al aangaf wat anderen wilden. Hieruit kwam het volgende naar voren:

- **Gepland onderhoud/storingen**
  Hiermee kunnen klanten inzien wanneer onderhoud van bijv. het netwerk zal plaatsvinden of waardoor storingen optraden.
- **Gebruikerstips**
  Handige tips m.b.t. allerlei software (Windows, Word, Groupwise, etc.)
- **Management rapportages**
  Xtilton rapportages voor het management van klanten.

Prioriteren van de functionaliteiten

Ik had nu een beeld had van wat men binnen het Xtilton zoal op het extranet terecht wilde zien komen, maar wist ik dat ik dit niet allemaal zou kunnen realiseren binnen de afstudeerperiode, omdat het teveel functionaliteiten waren. Ook was door dit vooronderzoek de grote verscheidenheid aan gewenste functionaliteiten gebleken en ik vroeg me af hoe ik dit alles tot één extranet zou kunnen samenvoegen. Hierover heb ik samen met mijn bedrijfsmentor gesproken. Tijdens dit gesprek werd het duidelijk dat er een basisstructuur voor het extranet gemaakt diende te worden waar alle overige functionaliteiten aan toegevoegd moeten kunnen worden. Zelfs al zou ik slechts enkele funtionaliteiten kunnen realiseren, dan zou men binnen Xtilton met behulp van deze basisstructuur het extranet gemakkelijk kunnen uitbreiden. Deze basisstructuur diende dus ook een systeem te omvatten waarmee het extranet uitgebreid kon worden. In overleg met de directeur en bedrijfsmentor zijn er uiteindelijk drie functionaliteiten geselecteerd als scope van de opdracht.

- **Basisstructuur extranet**
- **Incidentensysteem**
- **Veelgestelde-vragen systeem**

Bij het prioriteren is gekeken hoeveel tijd de functionaliteit zou besparen voor de servicedesk medewerkers. De meeste tijdwinst zat bij het incidentensysteem, gevolgd door het veelgestelde-vragen systeem. Het belangrijkst was echter de basisstructuur, die zodanig opgezet diende te worden dat al die andere functionaliteiten in de toekomst gemakkelijk konden worden toegevoegd.

Aan mijn bedrijfsmentor heb ik gevraagd of het gewenst was om deze basisstructuur met een reeds bestaand pakket te op te zetten. Voordat ik hem deze vraag voorlegde, heb ik een klein onderzoek verricht naar enkele pakketten die beschikbaar waren voor het opzetten van een extranet. Tijdens dit onderzoek bleek dat er honderden verschillende pakketten waren en ik heb dan ook slechts een kleine selectie hieruit bekeken. Ik heb drie pakketten aan mijn bedrijfsmentor voorgelegd en deze hebben we kort besproken. Geen enkel pakket sprak hem aan. De één had een implementatietijd van 12 weken, de ander was overdreven gecompliceerd. Bovendien was reeds bekend dat er een apart systeem, TAS, binnen moest komen te draaien. Tijdens dit gesprek werd mij meegedeeld dat men een volledig door Xtilton ontworpen en beheerd extranet wilden hebben.

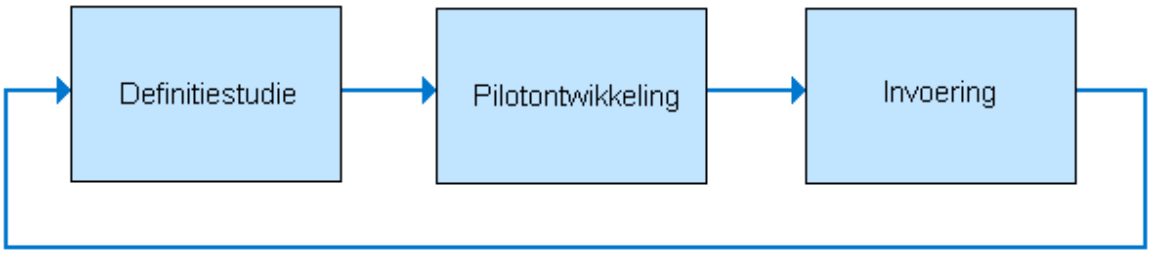
4.3 Methoden

Mijn bedrijfsmentor vertelde mij dat er binnen de organisatie geen methode gebruikt werd voor het ontwikkelen van software. Xtilton ontwikkelde namelijk slechts bij uitzondering software en richtte zich meer op andere ICT diensten als netwerkbeheer, telecom, hardware installatie en helpdesk services. Enkele voorbeelden van systemen die wel binnen Xtilton ontwikkeld zijn, zijn het urenregistratiesysteem, een systeem voor het installeren van printers bij een klant, het Xtilton intranet (dat echter grotendeels is ontwikkeld door andere stagiairs) en de Xtilton website. De keuze voor een ontwikkelmethode was dus aan mij.

Omdat ik tijdens mijn opleiding kennis had opgedaan van de IAD methode (Iterative Application Development) en ik hier al enkele malen mee gewerkt had, was ik van plan nu eens een andere methode te gebruiken. Deze methode zou echter wel voordelen moeten bieden vergeleken met IAD. Hiertoe heb ik enkele ontwikkelmodellen met elkaar vergeleken. Om niet te veel tijd te gebruiken voor het kiezen van een andere methode, heb ik twee andere geselecteerd, Extreme Programming (XP) en Dynamic Systems Development Method (DSDM). Bij XP bleek dat er meerdere programmeurs noodzakelijk waren. Omdat ik de enige programmeur zou zijn, viel deze methode af. Ik heb daarom IAD en DSDM met elkaar vergeleken.

4.3.1 IAD (Iterative Application Development)

Met IAD kan een systeem snel en effectief ontwikkeld worden. Het is hierbij belangrijk dat gebruikers een bijdrage kunnen leveren tijdens dit ontwikkelproces. IAD omvat drie fasen die elk geïtereerd kunnen worden, afhankelijk van de gekozen iteratiestrategie. Dit is tegenstelling tot de waterval-methode, waarbij het systeem sequentiëel ontwikkeld wordt. Een nadeel hiervan is dat fouten in een fase pas laat aan het licht komen en als dit eenmaal gebeurt, is het vaak noodzakelijk om terug te keren naar het begin van die fase om dit te herstellen. Ook kan deze lineaire aanpak resulteren in een eindproduct dat niet meer voldoet aan de eisen, doordat deze eisen veranderen zijn tijdens het ontwikkelproces. Het uitgangspunt van het watervalmodel is dan ook dat alle eisen vooraf foutloos te specificeren zijn. IAD ontwijkt deze valkuilen door het mogelijk te maken fasen te itereren. Fouten kunnen zo tijdig herkend en verholpen worden en op veranderingen kan snel ingespeeld worden. De drie fasen van IAD zijn:



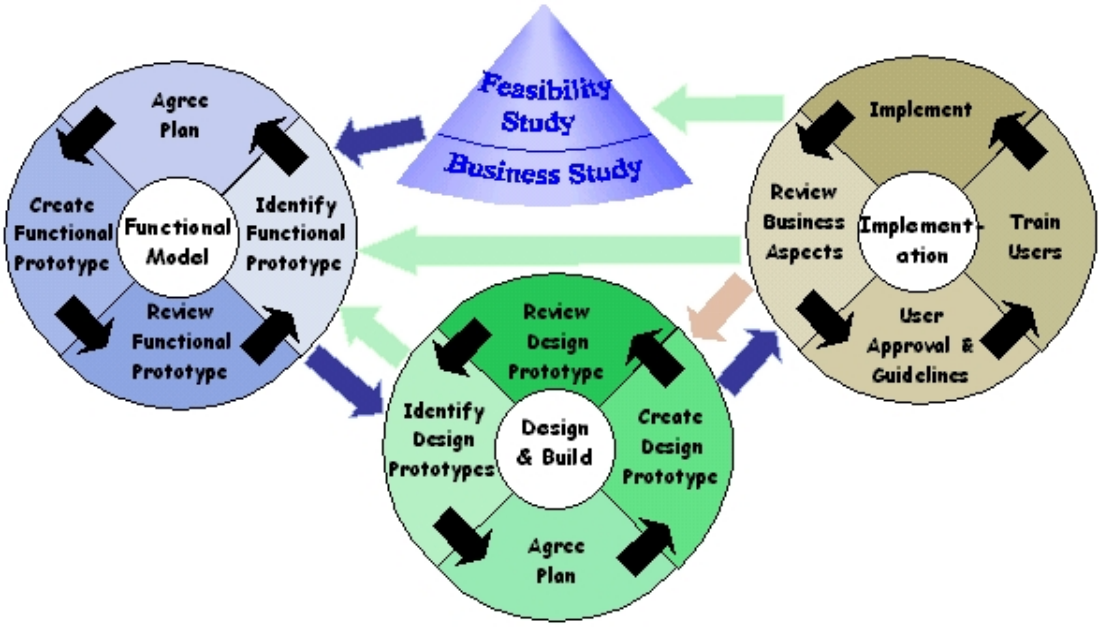
De fasen van IAD.

- **Definitiestudie**
  In deze fase wordt onderzocht aan welke eisen het systeem dient te voldoen. Hierna worden het systeemconcept opgesteld, de technische structuur en de organisatorische inrichting bepaald. Tot slot wordt het systeem in pilots opgedeeld.
- **Pilotontwikkeling**
  In deze fase worden de pilots ontworpen en gebouwd door de ontwikkelaars en getest en beoordeeld door de gebruikers.
- **Invoering**
  Het systeem of de pilot wordt ingevoerd in de organisatie en de ervaringen hiermee kunnen gebruikt worden bij de volgende iteratie.

Belangrijk is dat de gebruikers van het systeem bij elke fase betrokken worden, waardoor ze invloed kunnen uitoefenen op elke stap in het ontwikkelproces en dus ook op het eindproduct. Uiteindelijk zal dit ook de acceptatie van het systeem vergroten.

4.3.2 DSDM (Dynamic Systems Development Method)

DSDM steunt op een aantal principes, waarvan sommigen ook in IAD een rol spelen (zoals bijv. de betrokkenheid van de gebruikers en het iteratief en incrementeel ontwikkelen). De DSDM methode is toepasbaar voor zowel grote als kleine systemen. Net als bij IAD biedt DSDM de mogelijkheid de verschillende fasen van het ontwikkelproces te itereren en componenten incrementeel te ontwikkelen. Ook DSDM ontwijkt hiermee dus de valkuilen die bij een watervalmethode wel aanwezig zijn. Het uitgangspunt van DSDM is dan ook dat de specificaties van het te ontwikkelen systeem aan verandering onderhevig zijn constant bijgehouden zullen worden. De levenscyclus van DSDM bestaat uit vijf fasen, die afhankelijk van de situatie aangepast kunnen worden. In elke fase worden een aantal producten opgeleverd die aan een aantal kwaliteitscriteria dienen te voldoen. Als besloten wordt dat bepaalde producten niet noodzakelijk zijn, bestaat de kans dat het ontwikkelproces minder beheersbaar wordt.



De fasen van DSDM

- **Haalbaarheidsonderzoek**
  Hier wordt onderzocht of het project een succes kan worden en of DSDM een geschikte methode voor het project is. Ook wordt een schatting van de benodigde tijd en middelen gemaakt.
- **Bedrijfsanalyse**
  Welke processen en gebruikersgroepen ondersteund het systeem, wat is de technische basis en hoe verloopt het verdere ontwikkeltraject.
- **Functioneel Model Iteratie (FMI)**
  Hier wordt bepaald met welke functionaliteiten het systeem de processen uit de bedrijfsanalyse zal ondersteunen.

- **Ontwerp en Bouw Iteratie (OBI)**
  In deze fase wordende functionaliteiten van de FMI fase ontworpen en gebouwd. Bij aanvang van deze fase is een lijst met niet-functionele eisen en een functioneel prototype nodig. De OBI fase kan ook samengevoegd worden met FMI fase, maar hierdoor bestaat de kans dat sommige functionaliteiten niet genoeg aandacht krijgen.
- **Implementatie**
  Het systeem wordt geïmplementeerd en geëvalueerd. De documentatie wordt afgerond en de gebruikers getraind. Tot slot vindt er een evaluatie plaats.

De fasen FMI, OBI en implementatie kunnen iteratief doorlopen worden.

4.3.3 Verantwoording keuze voor IAD

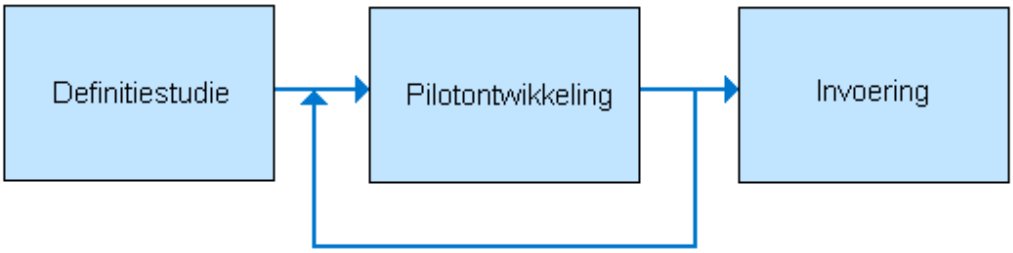
Uiteindelijk heb ik gekozen voor IAD, omdat:

- De eerste fase van DSDM was niet nodig voor de afstudeeropdracht, omdat door Xtilton al was besloten tot het ontwikkelen van een extranet. Tevens had ik zelf al een vooronderzoek uitgevoerd naar de functionaliteiten en scope van het extranet.
- DSDM is gericht op het frequent opleveren van producten. Dit was echter niet noodzakelijk voor dit project, omdat de totaliteit van het product hier van belang was. Ook had de opdrachtgever reeds aangegeven dat hij vooral de basisstructuur van het extranet van belang vond en niet allerlei tussentijdse producten hoefde te zien.
- DSDM is een relatief nieuwe ontwikkelmethode en het resultaat hiervan was dit ik weinig bruikbare informatie erover kon vinden. De fasen en uit te voeren processen worden wel beschreven op verschillende websites, maar wat er nu daadwerkelijk voor rapporten opgeleverd dienen te worden en wat hierin behoorde te staan bleef mij onduidelijk. Zonder voorbeeldrapporten zou het mij behoorlijk wat tijd en moeite kosten om dit uit te zoeken. Van IAD heb ik wel voorbeeldrapporten en kon ik dus sneller de benodigde documenten opstellen.
- Zowel IAD als DSDM maken het mogelijk om iteratief en incrementeel te ontwikkelen en verlenen een belangrijke rol aan de gebruikers. Ze vertonen dus veel overeenkomsten met elkaar. Van IAD weet ik echter hoe het in elkaar steekt en wat de gebruikte termen inhouden.

Kortom: het gebruik van DSDM bood voor dit project geen voordelen ten opzichte van IAD.

4.4 De keuze voor een iteratiestrategie

Na het vaststellen van de ontwikkelmethode is een iteratiestrategie gekozen. Er is gekozen voor de iteratiestrategie “incrementeel ontwikkelen”. Hierbij worden eerst de systeemeisen volledig in kaart gebracht en afgebeeld op een systeemconcept. Daarna wordt het gehele systeem via een aantal iteraties van de pilotontwikkelfase ontwikkeld. De fase definitiestudie en de fase invoering worden dus slechts eenmalig uitgevoerd.



De iteratiestrategie ‘incrementeel ontwikkelen’.

Ik heb om de volgende redenen gekozen voor deze iteratiestrategie:

- Er wordt een minimum aan documentatie geproduceerd.
- De eerste pilot is een zogenaamde ‘kernel-pilot’; deze vormt de basis voor alle overige pilots. Deze pilot kan na het ontwikkelen ervan niet ingevoerd worden, omdat dan eerst de overige pilots ontwikkeld moeten worden. De invoering kan dus slechts eenmalig plaatsvinden, omdat het gehele product van belang is en niet de afzonderlijke pilots.
- Dankzij mijn vooronderzoek naar de functionaliteiten van het extranet zou van alle pilots al bekend worden welke mogelijkheden deze omvatten. Door de definitiestudie eenmalig grondig uit te voeren, zouden dan ook de basis-specificaties van die pilots duidelijk worden. Als deze verder gespecificeerd dienden te worden, zou dit tijdens de pilotontwikkelfase gedaan kunnen worden.
- De eerste pilot was de grootste pilot en dus zou deze geïtereerd moeten worden in de fase pilotontwikkeling.

4.5 Technieken

De volgende technieken zijn gehanteerd tijdens de afstudeeropdracht:

Interviews

Met behulp van interviews kan informatie verzameld worden. Er wordt gekozen om een groot aantal korte interviews af te nemen, in plaats van slechts enkele grotere. Hierdoor zal de bereidbaarheid van de mensen om geïnterviewd te worden toenemen en hoeven deze interviews niet weken van tevoren ingepland te worden.

Testtechnieken

Bij het testen van de verschillende componenten van het systeem zullen enkele testtechnieken gebruikt worden, zoals de Cognitive Walkthrough en Think Aloud.

GUIDE (Graphical User Interface Design and Evaluation)

Er is voor GUIDE gekozen, omdat het hiermee de gebruikersgroepen en hun taken geanalyseerd kunnen worden. De verschillende stappen die deze taken omvatten dienen in kaart gebracht te worden, zodat bekeken kan worden of het systeem deze kan verbeteren ten opzichte van de huidige situatie en eventuele extra stappen kan automatiseren(bijv. het inloggen).

State Transition Diagrams

De inlogprocessen zullen met State Transition Diagrams weergegeven worden. Hier is voor gekozen, omdat dan de status van het systeem op verschillende momenten van de inlogprocedure in kaart gebracht kan worden. Een data flow diagram is hier niet geschikt voor, omdat hier geen beslissingsmomenten in weergegeven worden.

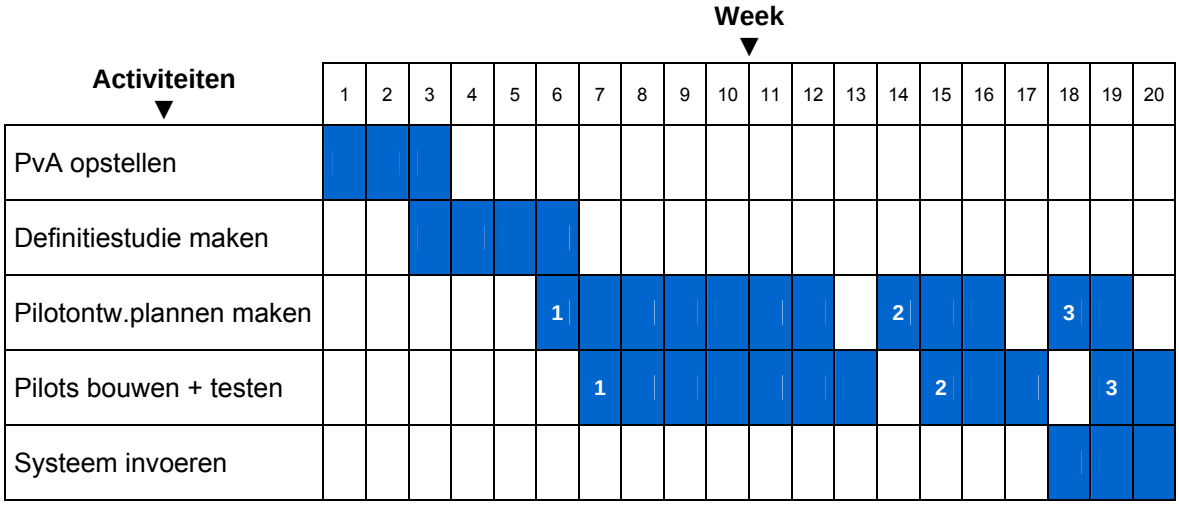
Objectmodellering

Objectmodellering is een techniek om de relaties tussen de verschillende objecten weer te geven, te bepalen welke acties er op de objecten uitgevoerd kunnen worden en dit uiteindelijk uit te werken tot een database. Deze technieken zijn geschikt voor dit project, omdat de kern van het te ontwikkelen extranet gebruik maakt van een database. Aanvankelijk zal deze database relatief klein opgezet worden, maar later dient deze wel uitgebreid te kunnen worden. Het exact vastleggen van de objecten, attributen, relaties en acties is dus vereist.

4.6 Planning, activiteiten en producten

4.6.1 Planning

In totaal duurt de afstudeerperiode 20 weken. De meeste tijd zou gaan zitten in het ontwikkelen van de basisstructuur van het extranet, omdat deze de kern van het systeem vormde en ook uitbreidbaar moest zijn. Deze basisstructuur vond de opdrachtgever dan ook het belangrijkst. Er werd gezamenlijk besloten dat als deze pilot zou uitlopen, dat de overige pilots dan later ontwikkeld zouden worden of zouden vervallen.



De keuze van de ontwikkelmethode en iteratiestrategie maakte het mogelijk de verschillende fasen af te spiegelen op de 20 weken van de afstudeerperiode.

4.6.2 Activiteiten

De uit te voeren activiteiten werden bepaald aan de hand van de ontwikkelmethode. Door mijn ervaring met IAD kon ik accuraat inschatten welke activiteiten elke fase omvatte.

Plan van Aanpak maken

- Vooronderzoek verrichten naar scope van opdracht
- Ontwikkelmethode bepalen
- Globale planning van afstudeerperiode maken

Definitiestudie uitvoeren

- De gebruikersgroepen inventariseren
- De systeemeisen en het systeemconcept bepalen
- De technische structuur bepalen
- De organisatorische inrichting bepalen
- De styleguide bepalen
- Een pilotplan maken

Pilotontwikkelplannen maken

- Functionele structuur per pilot bepalen
- Pilots onderverdelen in pilotdelen en bouweenheden
- Prioriteiten leggen m.b.t. de ontwikkeling van de pilotdelen

Pilots bouwen en testen

- De pilots realiseren in de testomgeving.
- Pilots testen en indien gewenst aanpassen.
- De pilots samenvoegen tot een geheel.

Systeem invoeren

- Installatieprocedure opstellen.
- Het systeem installeren op een server.
- Het systeem testen.
- Documentatie overhandigen.

Afstudeerslag maken

4.6.3 Op te leveren producten

In de derde week van de afstudeerperiode is met de directeur en mijn bedrijfsmentor overlegd over welke eindproducten er voor de organisatie opgeleverd dienden te worden. Hieruit kwam het volgende naar voren:

Voor de opdrachtgever

- Plan van Aanpak
- Extranet (minimaal een werkende basisstructuur)
- CMS
- Systeemdocumentatie (voor het beheer)

Voor de Haagse Hogeschool:

- Afstudeerverslag

5. De fase definitiestudie

Na het voltooien van het Plan van Aanpak ben ik begonnen met de fase definitiestudie. Dankzij mijn ervaring met IAD had ik een goed idee van de verschillende onderdelen die ik in deze fase zou gaan opstellen. Deze onderdelen waren:

- Gebruikersanalyse
- Systeemeisen
- Systeemconcept
- Technische structuur
- Organisatorische inrichting
- Styleguide
- Pilotplan

Per onderdeel heb ik een aantal activiteiten uitgevoerd. Deze worden in de volgende paragrafen besproken.

5.1 De gebruikersgroepen analyseren

5.1.1 Het bepalen van de verschillende gebruikersgroepen

Allereerst wilde ik weten welke gebruikersgroepen er van het door mij te ontwikkelen extranet gebruik zouden gaan maken. Mijn bedrijfsmentor heeft mij verteld welke klanten Xtilton heeft, hoeveel mensen er ongeveer bij die klanten werken en wat Xtilton doet voor hen. Hij vertelde mij dat het extranet was bedoeld voor de medewerkers van alle klanten van Xtilton en dat het ook door eventuele toekomstige klanten gebruikt moest kunnen worden. Omdat Xtilton voorlopig niet van plan is internationaal te gaan opereren, geldt dit alleen voor Nederlandse klanten. Deze gebruikersgroep was dus erg breed.

Er werd mij ook verteld dat het extranet gemakkelijk door Xtilton medewerkers te beheren en uit te breiden zou moeten zijn met nieuwe diensten en informatie. De tweede gebruikersgroep was dus het personeel van Xtilton dat deze taak op zich zou nemen. Toen ik vroeg over wat voor soort diensten en informatie het hier ging, kreeg ik enkele voorbeelden te horen:

- De Service Level Agreement tussen Xtilton en een klant
- Rapportages van Xtilton voor het management van een klant
- De antwoorden op veelgestelde vragen

Door deze voorbeelden werd het mij duidelijk dat op het extranet onderscheid gemaakt moest kunnen worden tussen de verschillende klanten. Binnen die klanten zelf moest echter ook onderscheid gemaakt kunnen worden tussen bijvoorbeeld managers en gewoon personeel.

Er zijn dus drie gebruikersgroepen:

1. **Personeelsleden van klanten van Xtilton**
   Dit omvat alle personeelsleden van alle klanten van Xtilton. De klant Parnassia heeft bijvoorbeeld 3500 personeelsleden, die allemaal het extranet moeten kunnen bezoeken.

2. **Xtilton personeel**
   Dit zijn degenen binnen Xtilton die het beheer en de uitbreiding van het extranet voor hun rekening gaan nemen.

3. **Specifieke gebruikersgroepen bij de klant**
   Een groep personeelsleden van een klant welke zich onderscheid van de globale gebruikersgroep(1). Dit is eigenlijk een subgroep van gebruikersgroep 1.

Om erachter te komen welke specifieke gebruikersgroepen er bij de klant zijn, heb ik van de klant Parnassia het interne telefoonboek bekeken, dat vanuit Xtilton toegankelijk was. In dit telefoonboek was een dropdownmenu zichtbaar waarin de verschillende beroepsfuncties van het Parnassia personeel alfabetisch gerangschikt zijn.

Parnassia telefoonboek: de verschillende functies

Het bleek dat er honderden verschillende beroepsfuncties binnen Parnassia zijn. Tijdens de definitiestudie heb ik dus alleen de eerste twee gebruikersgroepen geanalyseerd. De derde gebruikersgroep zou pas van belang zijn als eenmaal begonnen werd met het ontwikkelen van een functionaliteit die ook daadwerkelijk gericht was aan die gebruikersgroep. Mocht er bijvoorbeeld ooit een functionaliteit ontwikkeld worden die bedoeld is voor het keukenpersoneel van Parnassia, dan zal die gebruikersgroep dan pas geanalyseerd worden. Het Xtilton personeel moet deze functionaliteit echter gemakkelijk aan het extranet kunnen toevoegen en beschikbaar kunnen stellen aan die gebruikersgroep.

5.1.2 Informatie verzamelen over de gebruikersgroepen

Informatie over deze twee gebruikersgroepen heb ik verzameld tijdens meerdere interviews met verschillende personen binnen Xtilton.

Gebruikersgroep 1: klantpersoneel

Veel werknemers van Xtilton werkten voorheen bij Parnassia(welke nu een klant van Xtilton is en dus onder de eerste gebruikersgroep valt). Hier kon ik dus veel informatie halen over de klant Parnassia. Verder is bij de Xtilton Servicedesk veel informatie en kennis aanwezig over het personeel van alle klanten van Xtilton. Het was hier belangrijk om niet de individuele klanten te bekijken, want dan zou een vertekend beeld kunnen ontstaan dat teveel toegespitst was op die klant. Het ging hier om het totaal van alle gebruikers en degenen die hier het meest van afwisten waren de Xtilton medewerkers zelf. Men krijgt daar dagelijks telefoontjes en mailtjes met allerlei vragen en problemen, waardoor ze een goed beeld krijgen van het ICT-kennsniveau van het klantpersoneel. Ook worden al deze incidenten in een systeem (Topdesk) opgeslagen en kunnen deze op allerlei manieren gesorteerd en in grafiekvorm weergegeven worden. Informatie over de verschillende computerprogramma's die bij de klant gebruikt worden, was bekend bij de afdeling Applicatiebeheer.

Uit dit alles bleek dat, zoals ik al verwachtte, deze gebruikersgroep zeer gevarieerd was. Het belangrijkste dat hieruit naar voren kwam was dat een groot aantal mensen in deze gebruikersgroep weinig (of zelfs geen) kennis van computers hadden. Sommigen waren zelfs letterlijk bang voor muizen. Ook ICT jargon en Engelse termen waren voor hen dus onbekend. Aan de andere kant waren er ook ICT experts bij klanten aanwezig voor wie juist weer het tegenovergestelde gold.

Gebruikersgroep 2: Xtilton personeel

Informatie over deze gebruikersgroep heb ik verzameld bij enkele collegae. Eerst heb ik vastgesteld wie het beheer van het extranet waarschijnlijk zou gaan uitvoeren en vervolgens heb ik deze mensen enkele vragen gesteld over hun kennis op verschillende gebieden van ICT.

Het beeld dat hieruit naar voren kwam was een gebruikersgroep die weliswaar vaak met computers werkte en hier dus ook van veel van afwist, maar geen kennis had van programmeren en het ontwikkelen van software, op een enkele uitzondering na.

5.1.3 De gebruikerstaken analyseren

Tijdens deze interviews heb ik ook vragen gesteld over hoe de gebruikers hun taken in de huidige situatie uitvoeren, op welke volgorde ze dit doen, hoe lang ze erover doen, en waar problemen konden optreden. Aanvankelijk vond ik het lastig om de taken van de eerste gebruikersgroep in kaart te brengen, omdat ze voor veel verschillende soorten informatie en vragen bij Xtilton terecht kunnen. Door het onderzoek naar de scope van de opdracht, dat ik tijdens het maken van een Plan van Aanpak had uitgevoerd, kon ik hier echter een grens aan stellen.

De gebruikerstaken van de eerste gebruikersgroep (klantpersoneel) waren de volgende:

- Een ICT probleem of vraag melden aan de Xtilton Servicedesk
- Het opvragen van de status van hun probleem of vraag

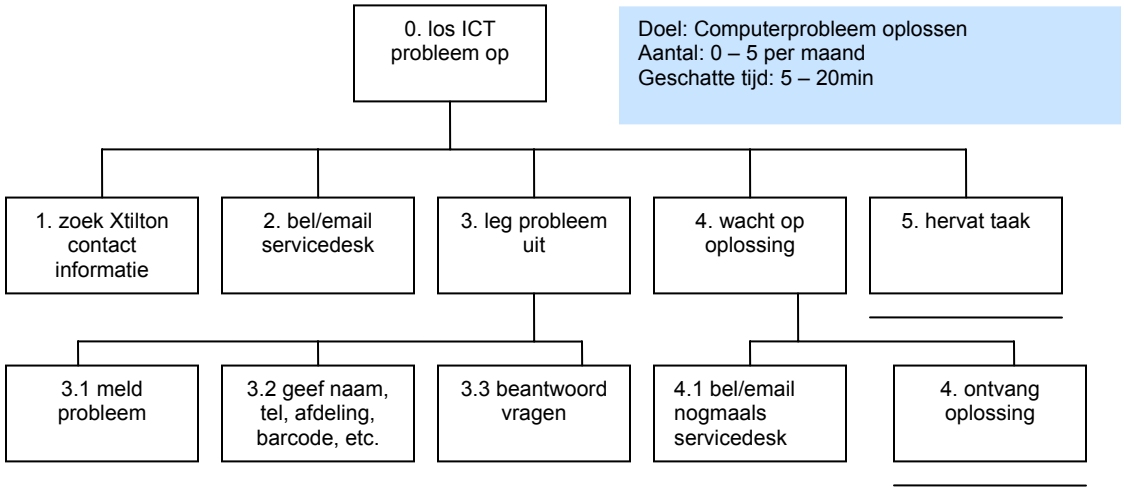
Bij de tweede gebruikersgroep waren de volgende taken te onderscheiden:

- De klant helpen met zijn ICT probleem/vraag
- Het verspreiden van informatie of diensten richting de klant

Ik besloot de taken van de gebruikersgroepen in taakdiagrammen weer te geven. Hierdoor zou precies zichtbaar worden welke stappen de gebruiker uitvoert en op welke volgorde hij dit doet. Voor de gebruikersgroep klanten vond ik dit belangrijk, omdat ik vermoedde dat het lastig zou zijn om deze stappen met het extranet te verbeteren. Als het extranet nadelen zou opleveren voor dit stappenproces bij de klant, wilde ik precies weten welke, zodat hier wellicht oplossingen voor bedacht konden worden. Ook bij de tweede gebruikersgroep, het Xtilton personeel, wilde ik op deze manier de huidige situatie met de gewenste situatie vergelijken. Bij hen was het van belang per taak en subtaak de tijdsduur te bepalen, want dit zou uiteindelijk het voordeel moeten opleveren voor Xtilton.

De informatie voor beide taakdiagrammen is verkregen door interviews met Servicedesk personeel en het doorlezen van procedures over het afhandelen van bepaalde situaties.

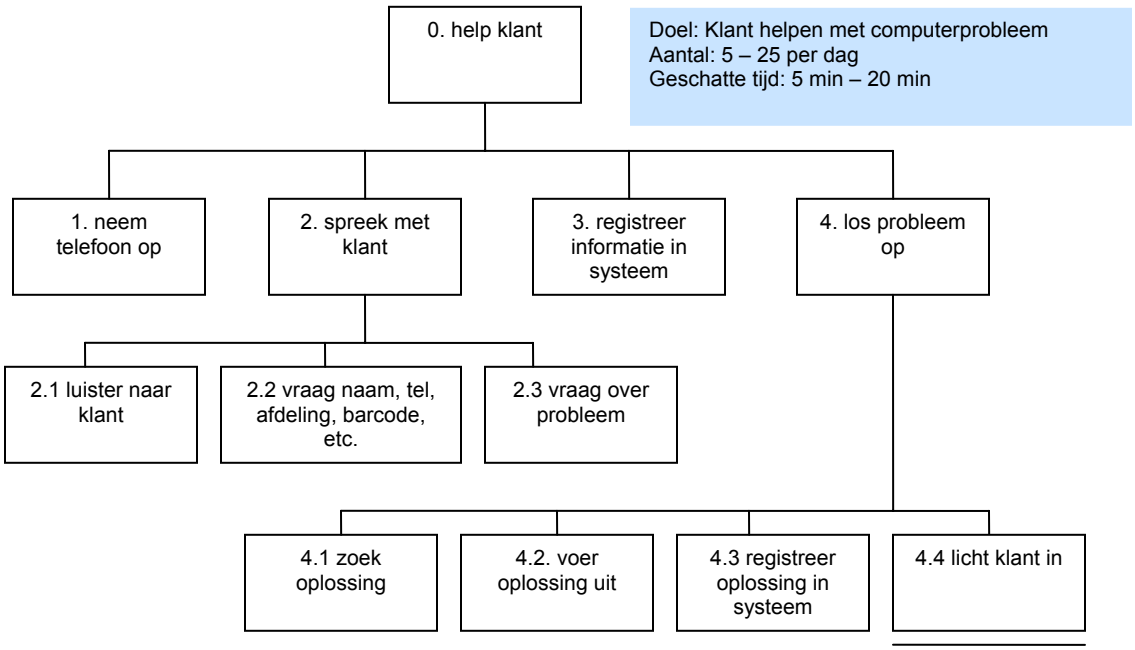
Hieronder ziet u een taakdiagram van de huidige situatie bij de klant:



Taakdiagram huidige situatie bij de klant – het oplossen van een ICT probleem

In de huidige situatie belt de klant de Xtilton servicedesk en meldt hier zijn vraag of probleem. Tijdens dit telefoongesprek zal hij enkele vragen van de servicedeskmedewerker moeten beantwoorden, zodat die weet waar het om gaat. In het gunstigste geval wordt het probleem meteen opgelost en kan de klant verdergaan waar hij mee bezig was. Als het oplossen van het probleem meer tijd kost dan wordt hij meegedeeld wanneer het ongeveer opgelost zal zijn. Wil de klant vervolgens tussentijds weten hoe het er met zijn melding voor staat, dan zal hij opnieuw contact moeten opnemen met de servicedesk.

Dit klinkt als een vrij logisch en handig proces voor de klant en het was de bedoeling dat dit in de toekomstige situatie geen nadelige extra stappen zou opleveren. Bovenstaande taak is ook vanuit het perspectief van de servicedesk medewerker bekeken:



Taakdiagram huidige situatie bij servicedesk medewerker – het helpen van een klant

In de huidige situatie neemt de servicedeskmedewerker de telefoon op en luistert naar wat de klant te melden heeft. Vervolgens stelt hij de klant enkele vragen om te weten te komen met wie hij spreekt, over welke hard- of software het gaat en wat de preciezere details van het probleem zijn. Dit alles wordt door de servicedeskmedewerker geregistreerd in het Topdesk systeem. Uiteindelijk lost hij het probleem op, licht de klant in en registreert dit in het systeem.

De meeste tijd bij deze taak gaat zitten in het voeren van het telefoongesprek, registreren van de gegevens en het oplossen van het probleem.

Een andere taak die bij Xtilton ligt is het verspreiden van informatie of diensten richting de klant. In de huidige situatie werd dit over het algemeen gedaan door een mailinglist samen te stellen, de informatie in de email te stoppen en vervolgens te versturen. Tijdens een interview kreeg ik echter te horen dat veel mensen deze emails eigenlijk nooit lezen en meteen verwijderen. Ook komen er af en toe telefoontjes bij de servicedesk binnen van mensen die zich zo’n verwijderde email herinneren en achteraf toch graag willen weten wat daar instond. Deze informatie moet dan opnieuw opgezocht worden en naar die persoon verstuurd worden.

Bij sommige taken is ook een taakscenario gemaakt waarin de details van een bepaalde taak duidelijker naar voren kwamen.

5.2 Het opstellen van de systeemeisen

Uit alle eerdere activiteiten waren inmiddels al een aantal systeemeisen duidelijk geworden. Zo moest het extranet informatie kunnen weergeven afhankelijk van de klant en beroepsfunctie van de bezoeker. Dit betekende dat er een inlogfunctionaliteit nodig zou zijn en bepaalde soorten informatie afgeschermd moesten kunnen worden. De bezoeker moest via het extranet problemen kunnen melden aan de servicedesk en de status hiervan in kunnen zien. Het Xtilton personeel moest dit alles kunnen uitbreiden met meer informatie en diensten en ook de toegankelijkheden van de verschillende klanten en beveiligingsprofielen kunnen regelen. Hieronder ziet u enkele systeemeisen:

Basissysteemeisen

- Klantpersoneel moet kunnen inloggen als de juiste klant en met het juiste beveiligingsprofiel.
- Per klant moet het systeem de informatie tonen die gericht is op die klant en zijn profiel.
- Klantpersoneel moet problemen en vragen kunnen melden aan de servicedesk.
- Klantpersoneel moet de oplossingen van veelgestelde vragen kunnen inzien.
- Xtilton medewerkers moeten het systeem gemakkelijk kunnen uitbreiden met nieuwe diensten en informatie.
- Xtilton medewerkers moeten informatie gemakkelijk beschikbaar kunnen stellen aan klanten en profielen.
- Xtilton medewerkers moeten klant en profielgegevens kunnen bewerken.
- Xtilton medewerkers moeten de onderdelen van het extranet (menu items, hyperlinks) kunnen wijzigen.

Interface en bruikbaarheids eisen

- De huisstijl van de Xtilton moet zoveel mogelijk aangehouden worden.
- Het klantgedeelte moet door computerleken te gebruiken zijn.
- Het klantgedeelte moet volledig Nederlandstalig zijn.
- Het klantgedeelte moet transparant en gemakkelijk toegankelijk zijn
- Als de gebruiker een taak uitvoert moet het systeem heldere feedback geven.

Integriteitseisen

- Er mogen geen klant/profiel gegevens voor andere klanten/profielen zichtbaar zijn.
- Er mogen geen niet-Xtilton medewerkers het CMS binnengaan.
- Er mogen geen niet-klanten het extranet binnengaan.
- Er mogen geen foutmeldingen op het scherm van de gebruiker getoond worden.

Performance eisen

- Meerdere gebruikers moeten het systeem tegelijkertijd kunnen raadplegen.
- De gebruiker moet de informatie die hij zoekt binnen 30 seconden gevonden kunnen hebben.

Operationele eisen

- Het systeem moet te bereiken zijn via Internet Explorer en Firefox.
- De gebruikers moeten het systeem zonder aanpassingen van hun eigen computer kunnen raadplegen.
- Xtilton medewerkers moeten het extranet kunnen uitbreiden en beheren zonder enige kennis van programmeren.

5.3 Het systeemconcept bepalen

5.3.1 De Inlogprocedure voor extranet bezoekers

Nadat de systeemeisen waren opgesteld, heb ik samen met mijn bedrijfsmentor gebrainstormd over het systeemconcept. Tijdens dit brainstormen kwamen we op twee mogelijkheden waarop de gebruikers het extranet zouden kunnen binnenkomen. Dit kon op twee manieren:

- **Mogelijkheid A:**
  door elke gebruiker een eigen gebruikersnaam en wachtwoord te geven en deze gebruikers vervolgens in te delen in beveiligingsprofielen

- **Mogelijkheid B:**
  door per beveiligingsprofiel een wachtwoord in te stellen en vervolgens hele gebruikersgroepen dit password te geven.

Van deze twee mogelijkheden heb ik bekeken wat de impact op het systeem(database en schermen), de bruikbaarheid en de onderhoudbaarheid zou zijn. Dit resulteerde in een lijst met voor- en nadelen.

Mogelijkheid A

Hierbij krijgt elke persoon zijn eigen gebruikersnaam en wachtwoord. Per klant kunnen dan een aantal beveiligingsprofielen bestaan waaronder deze gebruikers vallen.

Het voordeel hiervan is dat er volledige controle is over wat welke persoon kan inzien. Personen kunnen ook verplaatst worden van het ene profiel naar het andere, of er kan zelfs een nieuw niveau gemaakt worden speciaal op de persoon gericht.

Het nadeel is dat er op deze manier duizenden verschillende inlognamen en wachtwoorden in het systeem moeten komen te staan. Op zichzelf is dit geen probleem, want een MySQL database kan dit gemakkelijk aan, maar deze gebruikers en wachtwoorden moesten wel in de database ingevoerd worden. Handmatig zou dit ondoenlijk zijn, dus ik wilde weten of er al een database bestond waarin gebruikersnamen en wachtwoorden van klantpersoneel aanwezig waren. Het handigst voor de gebruiker zou natuurlijk zijn als deze wachtwoord/gebruikersnaam-combinatie overeenkomt met de wachtwoord/gebruikersnaam-combinatie waarmee hij normaliter al inlogt op de computer op zijn werkplek.

Om hier duidelijkheid over te krijgen heb ik nogmaals met het hoofd van de servicedesk en een andere servicedeskmedewerker gesproken. Ze vertelden mij dat de gebruikers bij de klant Parnassia inloggen via Groupwise, maar dat niet iedereen dit doet en er nog 2 andere databases zijn waarin wachtwoorden en gebruikersnamen opgeslagen worden. Bovendien zijn er ook klanten die weer gebruik maken van andere systemen dan Novell of waarbij Xtilton geen toegang heeft tot de inloggegevens die binnen het bedrijf gebruikt worden.

Een nadeel van deze manier op organisatorisch niveau was dat er mensen moesten zijn die deze logingegevens zouden gaan beheren. Als er nieuwe gebruikers bij zouden komen dan moesten de klanten dit doorgeven aan Xtilton. Mensen die hun wachtwoord kwijt waren zouden eveneens contact met Xtilton moeten opnemen, waardoor er weer meer telefoontjes bij de servicedesk zouden binnenkomen.

Mogelijkheid B

Hierbij krijgt elke klant(een geheel bedrijf) een aantal beveiligingsprofielen toegekend. Elk profiel heeft dan een wachtwoord. Hele groepen gebruikers kunnen dan met behulp van hetzelfde wachtwoord inloggen.

Het voordeel hiervan is dat de hoeveelheid inloggegevens zeer beperkt blijft. Hierdoor is het eenvoudiger te beheren en implementeren. Ook klanten die hun wachtwoord kwijt zijn kunnen aan de collega naast hen vragen wat het wachtwoord ook alweer was, waardoor ze hiervoor niet meer de Xtilton servicedesk hoeven te bellen.

Het nadeel hiervan is dat het moeilijk bij te houden is welke gebruiker bij welk beveiligingsprofiel hoort. Het zou immers niet meer persoonsspecifiek zijn, maar gelden voor hele gebruikersgroepen. Een ander nadeel is dat als een wachtwoord gewijzigd wordt, dat de hele gebruikersgroep dan hiervan op de hoogte gebracht moet worden. In de praktijk zou dit dus zelden mogen voorkomen. Tevens is bij deze mogelijkheid de kans groter dat ongeautoriseerden een wachtwoord bemachtigen. Het wachtwoord is hier immers een soort publiek geheim.

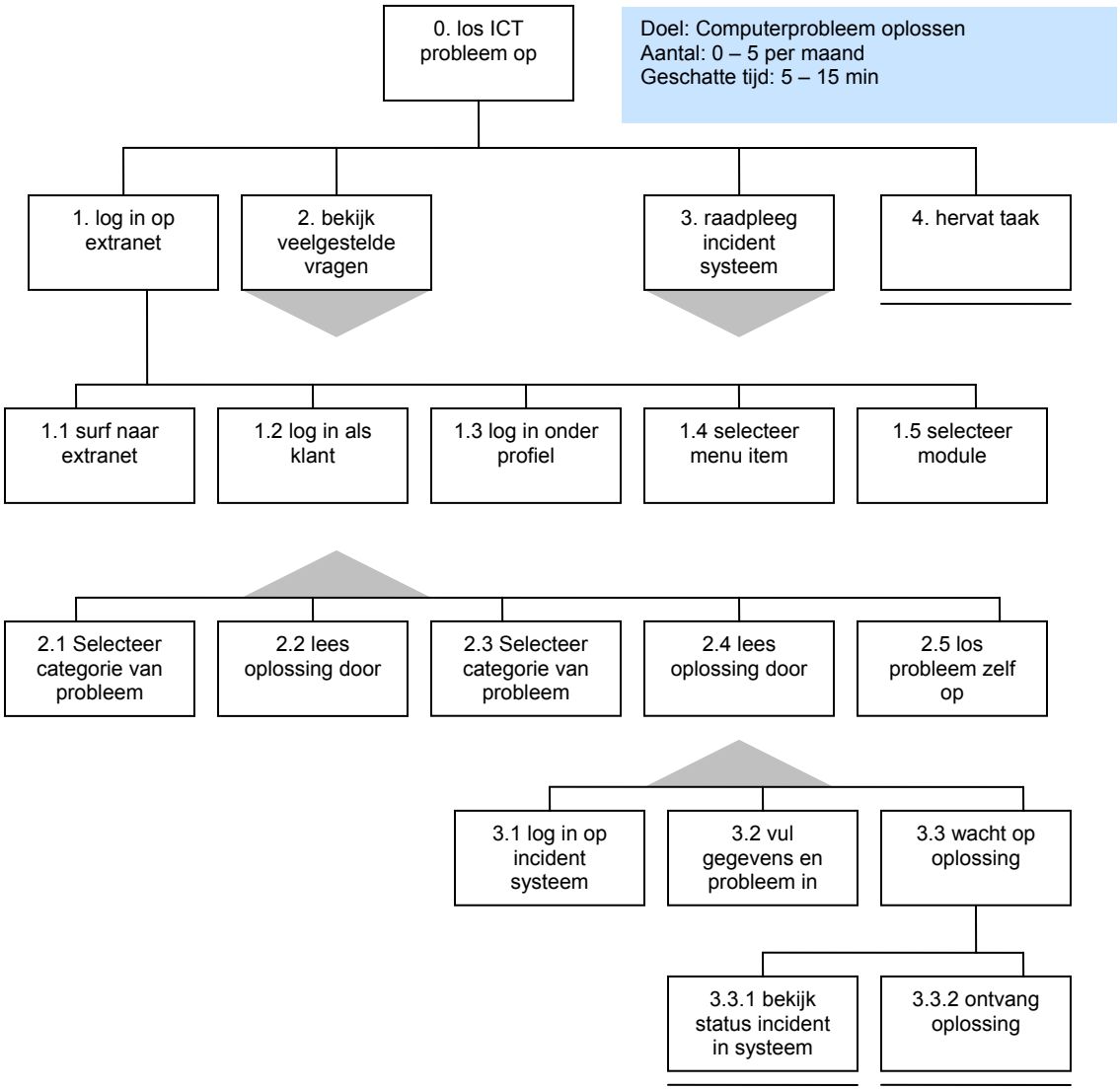
Uiteindelijk is in overleg met mijn bedrijfsmentor gekozen voor optie B, omdat bij mogelijkheid A het beheer van de gebruikersnamen en wachtwoorden te veel tijd zou kosten voor Xtilton.

5.3.2 Het beheer van het extranet

Door deze keuze werd duidelijk hoe het beheer van het extranet door Xtilton personeel zou moeten verlopen. Het was noodzakelijk dat er in het systeem klanten en beveiligingsprofielen beheerd konden worden en dat deze toegang tot bepaalde modulen verleend zouden kunnen worden. Door de grote variatie aan modulen en het feit dat ook externe systemen (zoals het incidentsysteem Topdesk Application Server)binnen het extranet zouden komen te draaien, was het noodzakelijk om het toevoegen van modulen erg flexibel te maken. Hiervoor zou het noodzakelijk zijn een Content Management Systeem (CMS) te ontwikkelen. Samen met mijn bedrijfsmentor zijn aan de hand van deze eisen de verschillende objecten van het systeem bepaald. Dit diende slechts als een eerste opzet en zou tijdens de pilotontwikkeling verder gespecificeerd worden en resulteren in een objectmodel en uiteindelijk de database. Met het CMS zouden deze objecten in de database bewerkt moeten kunnen worden.

5.3.3 ICT problemen of vragen melden aan de servicedesk

Zoals tijdens het onderzoek naar de scope van de opdracht al gebleken was, bestond er al een systeem (Topdesk Application Server) waarmee men problemen of vragen aan de servicedesk kon melden. Hieruit was ook naar voren gekomen dat dit door maar een handvol mensen gedaan werd. Ik besloot zelf eens het systeem te bekijken om te zien hoe het binnen het extranet geïmplementeerd zou kunnen worden. Aan de hand hiervan heb ik een taakdiagram opgesteld waarin de stappen zichtbaar werden die in de gewenste situatie door de gebruiker uitgevoerd zouden worden om een ICT probleem te melden aan de servicedesk.



Taakdiagram gewenste situatie bij de klant – het oplossen van een ICT probleem

Hieruit werd zichtbaar dat de gewenste situatie voor de klant zowel voor- als nadelen zou opleveren ten opzichte van de huidige situatie. Nadelen zijn dat hij moet inloggen en zelf zijn gegevens moet invullen, i.p.v. dat de servicedeskmedewerker dit voor hem doet. Voordelen zijn dat hij de antwoorden op veelvoorkomende vragen kan vinden en dat hij de status van zijn incident kan volgen. Het was de bedoeling dat de gewenste situatie meer voordelen dan nadelen zou opleveren. Tijdens de pilotonwikkeling zou hier verder naar gekeken worden.

5.4 De technische structuur bepalen

Programmeertaal

De keuze voor een programmeertaal waarmee het extranet ontwikkeld zou worden liet de opdrachtgever volledig aan mij over. Om deze beslissing te nemen heb ik binnen Xtilton rondgevraagd waar men zoal ervaring mee had. Het bleek dat er enkele mensen bij Xtilton werkzaam waren die ervaring met PHP hadden, waaronder mijn bedrijfsmentor – degene die het extranet waarschijnlijk zou gaan beheren. Tevens was binnen Xtilton, grotendeels door stagiairs, al een intranet ontwikkeld met PHP. Ik kreeg te horen dat er in de toekomst ook nog stagiairs aangenomen worden die dit intranet zouden uitbreiden, en dat ditzelfde waarschijnlijk ook gold voor het extranet.

Een vergelijkend onderzoek tussen PHP en ASP.NET had ik tijdens mijn vorige stageperiode al verricht, dus dit hoefde ik niet nogmaals te doen. Technisch gezien waren er geen factoren die de een de voorkeur boven de andere gaf met betrekking tot de afstudeeropdracht. Echter wat betreft de leercurve bleek ASP, zo vond ik zelf en kreeg ik ook van anderen te horen, wat meer tijd te vergen om onder de knie te krijgen.

Uiteindelijk heb ik om de volgende redenen gekozen voor PHP:

- binnen Xtilton was al een intranet dat ontwikkeld met PHP
- dit intranet, maar ook het extranet, zou in de toekomst uitgebreid worden door stagiairs met PHP ervaring
- mijn bedrijfsmentor had ervaring met PHP(echter ook ASP)
- ik had zelf al ervaring had met PHP
- PHP is eenvoudiger te leren

PHP zou voor dit project gebruikt worden in combinatie met:

- HTML
- CSS (stylesheets)
- Javascript
- SQL

Database

Door PHP te kiezen, werd snel duidelijk dat hierbij een MySQL database hoorde. Het is uiteraard mogelijk om PHP ook met andere databases te laten samenwerken, maar in de praktijk is het zo dat PHP het meest gebruikt wordt in combinatie met MySQL. Hiervoor is ook veruit de meeste documentatie op het internet te vinden. Bovendien draaide op de server waar het extranet op zou komen te draaien al het Xtilton intranet onder een MySQL database (versie 4.0). Nadat de keuze eenmaal op MySQL was gevallen, kon ik nog kiezen uit de verschillende versies. Er was inmiddels al een versie 4.1 en 5.0-beta beschikbaar. Ik heb niet gekozen voor versie 5.0-beta, omdat deze bedoeld is voor testdoeleinden en mogelijk kritieke fouten kan bevatten. Ook draaide op de toekomstige server van het extranet versie 4.0 dus zou ik toch geen gebruik kunnen maken van de nieuwe mogelijkheden die 5.0-beta (zoals bijv. de Foreign Key ondersteuning, die in voorgaande versies ontbreekt). De keuze viel uiteindelijk op 4.1 omdat dit de nieuwste stabiele versie was.

Later tijdens het ontwikkeltraject werd alsnog MySQL 5.0-beta op een andere server geïnstalleerd en werd het extranet hierop gezet. Omdat dit pas in de vijftiende week van de afstudeerperiode plaatsvond, kon ik geen gebruik meer maken van de nieuwe mogelijkheden die dit bood.

Om de database te bewerken wilde ik aanvankelijk een programma genaamd MySQL-Front gebruiken. Tijdens een eerder interview met de opdrachtgever kreeg ik echter te horen dat mensen die binnen Xtilton illegale software gebruiken onmiddellijk ontslagen zouden worden. Omdat MySQL-front een licentie vereist en ik wist dat er ook gratis tools beschikbaar waren, ben ik op zoek gegaan naar een ander programma. Hierbij kwam ik terecht bij PHPMYADMIN, welke ook het meest gebruikt wordt door PHP ontwikkelaars.

Server

Aan een collega van functioneel beheer heb ik gevraagd wat de technische specificaties waren van de server waarop het extranet zou komen te draaien. Dit bleek een Windows2003 systeem met daarop IIS 6.0(Internet Information Services) te zijn. Omdat ik gewend was met een Apache server te werken en deze ook van plan was lokaal te installeren, vroeg ik hem of de verschillen tussen Apache en IIS van belang waren. Er werd mij verteld dat, zolang ik maar geen Apache-specifieke functies van PHP gebruikte, een PHP applicatie precies hetzelfde zou werken onder IIS als onder Apache. Ik vermoedde dat ik geen Apache specifieke functies zou gaan gebruiken, dus dit zou geen problemen opleveren. Tevens wilde ik dat het extranet zowel op Apache als IIS zou kunnen draaien. Ik besloot dus om een Apache server te installeren, zodat ik later bij de installatie van het extranet op de webserver kon zien of het ook onder IIS goed werkte. Eventuele problemen zou ik dan wegwerken, met als resultaat een extranet dat zowel onder IIS als Apache zou werken.

Grafische tools

De grafische tools die tot mijn beschikking stonden waren:

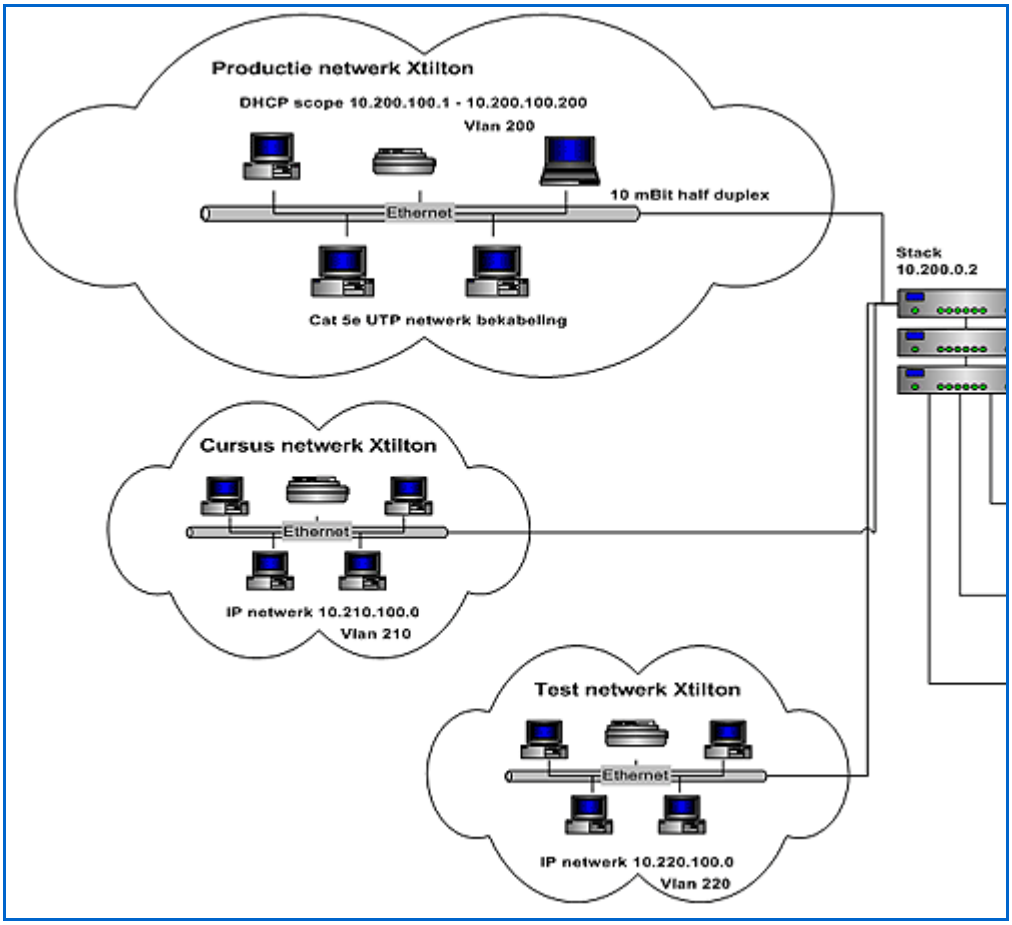
- Flash MX
- Fireworks MX
- Photoshop 9.0.

HTML/PHP-editors

Zowel Dreamweaver MX als Notepad stonden tot mijn beschikking als HTML/PHP-editors. Ik was echter niet helemaal tevreden met deze mogelijkheden. Notepad had geen kleurenweergave van PHP code en het inspringen van stukken code was lastig. Dreamweaver MX had wel een kleurenweergave en ook een handige optie voor het vervangen van stukken tekst in volledige folders. Helaas duurde het openen van dreamweaver iedere keer 10 seconden en dit zou extreem irritant en tijdrovend zijn het bewerken van veel bestanden. Ik was dus op zoek naar een klein programma, net als notepad, maar dan met kleurenweergave en de mogelijkheid om gemakkelijk code te laten inspringen. Voor massabewerkingen (het vervangen van identieke stukken code in meerdere bestanden tegelijk), die ik toch niet al te vaak zou uitvoeren, kon ik altijd nog Dreamweaver gebruiken. Op internet vond ik een programma genaamd ‘Winsyntax’, welke aan beide voorwaarden voldeed.

Netwerk

Het Xtilton netwerk als geheel maakt onderdeel uit van het Parnassia netwerk. Met sommige andere klanten heeft Xtilton Virtual Private Network verbindingen ingesteld. Binnen Xtilton is een netwerk aanwezig dat in drie verschillende Virtuele LAN's is opgedeeld: het productienetwerk, cursusnetwerk en testnetwerk.



De technische infrastructuur van Xtilton

Op het productienetwerk zijn alle werkstations binnen Xtilton aangesloten. Door mijn werkstation als server te laten functioneren, zouden alle overige werkstations binnen Xtilton ook het extranet kunnen bekijken. Dit was dus geschikt om als testsituatie tijdens de ontwikkeling van het extranet te fungeren. Later als het systeem geïmplementeerd zou worden, zou het extranet op een server geïnstalleerd worden die ook voor de klanten van Xtilton bereikbaar is.

5.5 De organisatorische inrichting bepalen

De organisatorische gevolgen van de invoering van het extranet heb ik grotendeels opgesteld door de taakdiagrammen van de huidige situatie van beide gebruikersgroepen te vergelijken met die van de gewenste situatie. Sommige organisatorische gevolgen zouden pas later aan het licht komen, bijvoorbeeld een stuk wachtwoordbeheer aan de kant van de klant. Hier zal in het hoofdstuk over de pilotontwikkelfase verder op ingegaan worden.

Klanten

Hieruit kwam voor de klanten (bezoekers van het extranet) naar voren dat het extranet voornamelijk het gebruik van bepaalde communicatiemiddelen in bepaalde situaties kan vervangen. Denk hierbij bijvoorbeeld aan iemand die het extranet gebruikt om een klacht of computerprobleem te melden aan Xtilton, in plaats van dat hij dit over de telefoon doet.

Xtilton medewerkers

Voor enkele medewerkers van Xtilton betekent het extranet dat zij nieuwe taken zullen gaan uitvoeren. Zo zal het extranet beheerd moeten gaan worden en kunnen er nieuwe functionaliteiten ontwikkeld en aan het extranet toegevoegd worden. Dit laatste zal waarschijnlijk echter door stagiairs gedaan worden. Ook zullen er binnen Xtilton personen zijn die de op het extranet aangeboden informatie up-to-date gaan houden.

Voor de Xtilton Servicedesk zal een mogelijk gevolg zijn dat zij minder routinevragen binnenkrijgen, maar wellicht weer andere vragen over mensen die een wachtwoord voor het extranet nodig hebben.

5.6 De styleguide bepalen

Als je in de buurt van het Xtilton pand komt, zie al snel verschillende uitingen van de Xtilton huisstijl. Auto’s met Xtilton kleuren en logo rijden af en aan, een Xtilton vlag wappert aan een paal en boven de ingang staat met reusachtige letters aangegeven dat er nu geen vergissing meer mogelijk is, je bent nu aangekomen bij Xtilton.

Het was me dan ook meteen duidelijk dat ik niet mijn persoonlijke stijl, een gifgroene kleur met felblauwe stippen, zou kunnen doorvoeren op het extranet.

Voor het verzamelen van informatie over de huisstijl heb ik aanvankelijk mijn bedrijfsmentor geraadpleegd. Hij heeft een collectie van bestaande Xtilton logo’s en afbeeldingen samengesteld zodat ik kon zien wat voor beeldmateriaal er aanwezig was. Ook was hij op dat moment druk in de weer met de ontwikkeling van de .NL website van Xtilton, waar hij een stylesheet voor had opgesteld. Verder stond er op de verschillende netwerkschijven nog meer huisstijl-materiaal.

Het bleek dat er drie hoofdkleuren gebruikt werden voor de huisstijl (wit – geel – blauw/paars) en 1 lettertype (arial). Hieronder ziet u een deel van de styleguide die in de fase definitiestudie is opgesteld:

Er waren geen exacte richtlijnen voor de lettergroottes van titels, hyperlinks, gewone tekst, pixelafmetingen van afbeeldingen, inspringingen, etc., dus hier had ik nog enige vrijheid in. Ook was het toegestaan tintvariaties van de hoofdkleuren te gebruiken. Er werd echter wel van het extranet verwacht dat het enigszins overeenkwam het de .NL website.

5.7 Het pilotplan opstellen

Dankzij het onderzoek naar de scope van het extranet, kon ik het systeem vrij gemakkelijk in pilots opsplitsen. Uit dit onderzoek was namelijk duidelijk geworden dat er drie functionaliteiten binnen de scope van de opdracht vielen. Ik besloot deze functionaliteiten ook als pilots te laten dienen. Ook de prioritering van deze functionaliteiten hadden de geïnterviewde medewerkers van Xtilton tijdens dit vooronderzoek al aangegeven. Bij deze prioritering werd er gekeken naar de hoeveelheid tijd een bepaalde pilot zou besparen voor de servicedesk.

De volgende pilots zijn opgesteld:

1. Basisstructuur extranet
2. Incidentensysteem
3. Veelgestelde vragen

Bovenstaande pilots heb ik bij het maken van het pilotplan al getracht in te delen in pilotdelen. Aan de hand van deze voorlopige indeling kon ik schatten hoeveel tijd het ontwikkelen ervan zou gaan kosten en hoe ik dit het best aan zou kunnen pakken.

Pilot	**Omschrijving**
1. Basisstructuur	Dit is een kernel-pilot, een pilot die functioneert als basis voor de latere pilots. Deze pilot omvat de database, extranet, content management systeem en de vormgeving. Deze pilot bestaat dus uit zowel een kant die de extranet bezoeker te zien krijgt, als een kant waarmee Xtilton personeel het extranet kan beheren.
2. Incidenten-systeem	Dit is de eerste module die in de basisstructuur geïmplementeerd zal worden. Hiermee kunnen klanten via het extranet incidenten melden en inzien.
3. Veelgestelde vragen	Hiermee kunnen klanten de oplossingen voor veelgestelde vragen via het extranet vinden.

Van bovenstaande pilots schatte ik in dat ik er ongeveer drie binnen mijn afstudeeropdracht kon afronden. Dit was voor mij moeilijk in te schatten, omdat ik nog nooit eerder een extranet had ontwikkeld. Ik heb echter de voorlopige pilotdelen vergeleken met producten die ik al eerder met PHP gemaakt had.De opdrachtgever had overigens aangegeven al tevreden te zijn met alleen pilot 1, de basisstructuur van het extranet.

Ik heb er voor gekozen om deze pilots sequentieel te ontwikkelen, omdat ze eigenlijk volledig los van elkaar staan qua functionaliteit, maar wel afhankelijk van elkaar zijn. Zo zullen pilot 2 en 3 niet ontwikkeld kunnen worden zonder dat er al een basisstructuur van het extranet is(pilot 1). Bij pilot 2 is een onderdeel van het ontwikkelen namelijk het aanpassen van de vormgeving van het bestaande incidentensysteem, aan de vormgeving van het extranet. De vormgeving van het extranet moest dus dan al voltooid zijn. Bij pilot 3 is het vereist dat de veelgestelde vragen klantspecifiek zijn. Er moet dus al een database aanwezig zijn waarin de verschillende klanten staan, zodat de veelgestelde vragen hieraan gekoppeld kunnen worden. Deze database maakt deel uit van pilot 1 en dus moet pilot 1 eerst ontwikkeld worden. Pilot 2 en 3 vervullen wel eenzelfde soort doel voor de gebruikers, namelijk het oplossen van computerproblemen of

vragen. Toch bieden beide een andere manier om dit doel te bereiken. Aan de hand van de al globaal bepaalde pilotdelen en de systeemeisen hiervan, is de ontwikkeltijd per pilot als volgt ingeschat:

Pilot	Ontwikkeltijd
1. Basisstructuur	7-11 weken
2. Incidentensysteem	4 weken
3. FAQ	4 weken

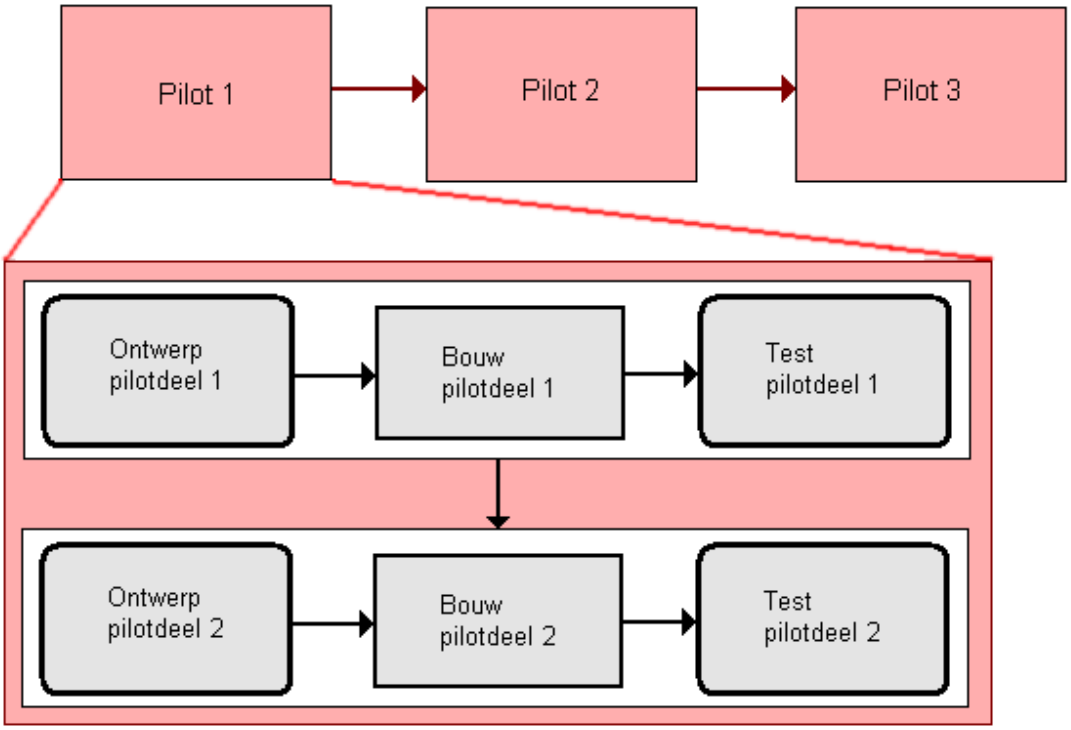
Ik hield er echter rekening mee dat de eerste pilot meer tijd zou gaan kosten dan ik hier inschatte, omdat deze pilot het grootst en belangrijkst was. Mocht er tijdens een iteratie van deze pilot of een pilotdeel ervan meer tijd nodig zijn, dan zou de deadline vooruitgeschoven worden. De overige twee pilots waren minder belangrijk dat de basisstructuur.

6. De fase pilotontwikkeling

Na de fase definitiestudie is in week 6 begonnen met de fase pilotontwikkeling. In het Plan van Aanpak is bepaald dat deze fase geïtereerd zou worden. Omdat de eerste pilot behoorlook groot was en dus in een aantal pilotdelen opgesplitst zou worden, was het noodzakelijk ook voor deze pilotdelen een iteratiestrategie te bepalen.

6.1 Het bepalen van de iteratiestrategie

Ik heb ervoor gekozen om eerst de specificaties van een enkel pilotdeel op te stellen en deze vervolgens te ontwikkelen en testen. Hierna zou het volgende pilotdeel ontwikkeld worden. Zie de afbeelding hieronder:



De iteratiestrategie binnen de fase pilotontwikkeling

Voor deze iteratiestrategie is gekozen, omdat vooral bij de eerste pilot (de basisstructuur) enkele duidelijk afzonderlijke pilotdelen met verschillende functionaliteiten zijn te onderscheiden. Deze zijn immers ook gericht op verschillende gebruikersgroepen. Zo is er een gedeelte dat de klant te zien krijgt (het daadwerkelijke extranet) en een gedeelte waarmee dit beheerd wordt (het CMS). Om te weten wat er beheerd gaat worden, en wat het CMS dus allemaal moet kunnen, moet eerst duidelijk zijn wat het klantgedeelte precies inhoudt. Dit deel moest dus eerst ontwikkeld en getest worden.

6.2 Het prioriteren van de pilots en pilotdelen

Voor het prioriteren en plannen van de pilots en pilotdelen was het mogelijk time-boxing te gebruiken. Bij time-boxing worden de pilots of pilotdelen in een aantal eenheden opgedeeld. Deze worden vervolgens geprioriteerd en per eenheid worden de factoren tijd en budget gefixeerd. Elke eenheid noemt men vervolgens een time-box. Deze time-boxes worden hierna één voor één ontwikkeld en mocht het einde van een time-box bereikt zijn, dan moet gestopt worden met de ontwikkeling ervan en begonnen worden met de ontwikkeling van de volgende time-box. Hierdoor is het de bedoeling dat de belangrijkste onderdelen van elke time-box ontwikkeld worden.

Ik heb bewust gekozen om geen time-boxing toe te passen. Dit heb ik gedaan omdat het bij dit project niet van belang was de opleverdatum van de verschillende onderdelen van het systeem vast te stellen. Sterker nog, het was zelfs zo dat de opdrachtgever de eerste pilot het belangrijkst vond en dat, indien deze uit zou lopen, hij liever zag dat de overige pilots later ontwikkeld zouden worden. Bij time-boxing zou de ontwikkeling van de eerste pilot stopgezet moeten worden als het einde van de time-box ervan eenmaal is bereikt. Omdat het vooraf niet exact in te schatten was hoeveel tijd deze pilot zou kosten, kon dit inhouden dat de basisstructuur uiteindelijk incompleet zou zijn. Een gefixeerde opleverdatum was dus niet gewenst, flexibele deadlines wel.

Bovendien was het niet nodig om snel tussentijdse producten op te leveren. Het belangrijkst was de basisstructuur van het extranet, waarvan de systeemeisen al nauwkeurig vastgelegd waren in de definitiestudie. De op te leveren functionaliteiten stonden dus vast. Bij time-boxing is het net andersom: de deadlines staan vast, en de functionaliteiten worden hieraan aangepast.

In plaats van time-boxing zou ik de pilots en pilotdelen zelf prioriteren aan de hand van de onderlinge afhankelijkheden en de geschatte tijd per onderdeel. Mocht een deel niet op tijd voltooid zijn, dan wordt de deadline verschoven zodat de functionaliteiten van dat deel toch volledig ontwikkeld kunnen worden. Het gevaar hiervan is dat er teveel tijd in de ontwikkeling van relatief onbelangrijke delen van het systeem zou gaan zitten. Hierbij is de zogenaamde 80/20-regel van toepassing: 80% van de tijd wordt besteed aan het ontwikkelen van de minder relevante 20% van het systeem. Om deze valkuil te voorkomen zou ik bij het prioriteren van de componenten van het systeem de relevantie per component in overleg met de betrokken persoon vaststellen.

6.3 Pilot 1: de basisstructuur

6.3.1 Het opsplitsen van de pilot in pilotdelen

Bij aanvang van de ontwikkeling van de eerste pilot is deze opgedeeld is verschillende pilotdelen. Er is voor gekozen om het aantal pilotdelen beperkt te houden en ze op te splitsen in duidelijk afzonderlijke delen. Deze pilotdelen zijn:

- **Database**
  De objecten, hun attributen, de relaties tussen de objecten en de verschillende databasemodellen.

- **Klant deel**
  Het deel van het systeem waar klanten gebruik van kunnen maken.

- **Xtilton deel (CMS)**
  Het deel van het systeem waar Xtilton personeel gebruik van gaat maken. Dit is het Content Management Systeem waarmee het extranet beheerd kan worden.

- **Vormgeving**
  De vormgeving van het geheel wordt uitgewerkt met als basis de styleguide uit de definitiestudie.

De database vormt de kern van deze pilot en zou daarom als eerst ontwikkeld worden. Het klant deel en Xtilton deel zijn als aparte pilotdelen ingedeeld, omdat beide voor een verschillende doelgroep bedoeld zijn en andere functionaliteiten hebben. Hierbij was gewenst het klant deel eerst te ontwikkelen, zodat het Xtilton deel beter gespecificeerd kon worden. Hiermee moest namelijk het klant deel beheerd kunnen worden.

De vormgeving zou pas in een later stadium gedaan worden. De prioriteit lag bij het ontwikkelen van de functionaliteiten van de pilot. Pas als deze functionaliteiten voldoende afgewerkt waren zou de vormgeving aandacht krijgen. Omdat ik van plan was de vormgeving centraal via stylesheets te regelen, heb ik tijdens de ontwikkeling van het klant deel en het CMS al wel in de PHP en HTML code classes benoemd en verwijzingen aangebracht naar de toekomstige stylesheets. Tijdens het ontwikkelen van het pilotdeel vormgeving zou dit veel tijd besparen.

Aan de hand van het aantal functionaliteiten en de omvang van elk pilotdeel is geschat hoeveel tijd de ontwikkeling zou kosten en hoeveel iteraties er nodig zouden zijn:

Pilotdeel	Ontwikkeltijd
1. Database	2 weken
2. Klantdeel	2 weken
3. Xtilton deel	3 weken
4. Vormgeving	1 week

6.3.2 Pilotdeel 1: de database

Voordat ik begon met het ontwerpen van deze pilot, ben ik eerst mijn kennis over databases gaan oprakelen. Ik heb de readers van de DB-modulen en het boek 'Database systemen voor de praktijk' erbij gepakt en nogmaals doorgelezen.

Het specificeren van de bouweenheden

Na afloop van deze voorstudie heb ik de verschillende stappen die bij het ontwikkelen van een database doorlopen worden op een rijtje gezet en ingedeeld in bouweenheden. Deze bouweenheden zijn:

- Conceptueel datamodel (Objectmodel)
- Relationeel representatiemodel
- Implementatiemodel
- Bouw van de database

Het bepalen van de objecten

Allereerst heb ik de verschillende objecten voor het objectmodel in kaart gebracht door samen met mijn bedrijfsmentor de verschillende onderdelen van het systeem nog eens door te nemen. Er was reeds besloten om de klanten te laten inloggen onder een bepaald profiel. Het systeem zal dus moeten weten bij welke klant een gebruiker hoort en onder welk beveiligingsprofiel hij valt. Hieruit kwamen de objecten 'klant' en 'profiel' naar voren.

Als eenmaal ingelogd is, moet het systeem de juiste functionaliteiten (ook wel modulen genoemd) tonen. De scope van de afstudeeropdracht behelsde het implementeren van twee modulen binnen de basisstructuur van het extranet, maar tijdens het onderzoek naar de scope waren er al minstens zes andere geïdentificeerd die in de toekomst geïmplementeerd zouden worden. Het confronteren van de gebruiker met een groot aantal verschillende links op één scherm zou hem kunnen afschrikken of het geheel onoverzichtelijk kunnen maken. Het was daarom noodzakelijk om deze modulen in verschillende menu-items in te delen. Uit dit proces kwamen de objecten 'menu-item' en 'module' naar voren.

Ook was uit de definitiestudie gebleken dat beheerders van het extranet verschillende soorten modulen aan het extranet moeten kunnen toevoegen. Eén daarvan, Topdesk Application Server, was een extern systeem, gemaakt door een ander bedrijf. Er is daarom besloten om in de database de hyperlinks naar de modulen op te nemen, zodat in principe elk soort module aan het extranet toegevoegd kan worden (zolang die tenminste in een browser geopend kan worden).

De attributen van de objecten

Nadat de objecten vastgesteld waren, heb ik samen met mijn bedrijfsmentor besproken welke attributen er per object nodig waren. Hierbij is per object de rol ervan in het systeem te bekeken. Vervolgens is er kort gebrainstormd over de benodigde attributen.

Het object 'module' bijvoorbeeld is een hyperlink op het extranet, welke alleen toegankelijk is als men onder het juiste profiel ingelogd is. De meest voor de handliggende attributen van dit object waren de naam, de hyperlink en de omschrijving. Als er meerdere modulen onder een menu-item vallen, dan is het echter ook handig om de belangrijkste module bovenaan te rangschikken, zodat deze het eerst in het oog springt bij de gebruiker. Van de hyperlink zelf vroeg ik me af waar deze naartoe verwijst, dus waar de module fysiek zou komen te staan. Het bleek dat de meeste extranetmodulen in de toekomst op dezelfde server als het extranet zouden komen te staan. Er zou op de server dan dus een standaard folder zijn waarin deze modulen geplaatst konden worden. Het grootste deel van de extranet-hyperlinks naar deze modulen zou dus identiek zijn, waardoor ik op het idee kwam om een 'standaard_directory' attribuut aan dit object toe te voegen.

Tijdens een latere iteratie van deze pilot zou dit attribuut in een andere tabel opgenomen worden, omdat de standaard directory voor elke module hetzelfde is en dus een gehele kolom in de tabel vol zou staan met dezelfde waarde. Tevens bleek dat sommige modulen op externe servers konden komen te staan, wat inhield dat dit attribuut niet voor alle modulen gold. Op dezelfde manier zijn ook de attributen van de andere objecten bepaald en bij iteraties aangepast.

Vervolgens is bekeken welke acties er op de verschillende objecten uitgevoerd moesten kunnen worden, afhankelijk van de functie die het object binnen het systeem zou vervullen. Hieronder ziet u als voorbeeld het object 'module':

Module
Attributen -naam -link (op extranetserver) -url (op externe server) -rang -popup -omschrijving -commentaar
Acties -toevoegen/bewerken/verwijderen (standaard acties) -rang aanpassen -popup aan/uitzetten -modulen verwisselen van menu item -modulen toegankelijk maken voor profielen

De relaties tussen de objecten

Hierna hebben we getracht de exacte relaties tussen deze objecten te bepalen, maar dit bleek wat lastiger te zijn, omdat er verschillende mogelijkheden waren. Elke mogelijkheid zou het systeem sterk beïnvloeden, en dan met name de inlogprocedure voor de klanten.

Zo kan een klant een aantal beveiligingsprofielen hebben. Aanvankelijk dachten we hierbij aan de profielniveau's <hoog> <midden> <laag>. Een klant had dan dus meerdere beveiliginsprofielen. Een profiel kon echter ook bij meerdere klanten horen, want iedere klant zou dezelfde profielindeling (hoog, midden, laag) hebben. In de praktijk zou dit resulteren in een extra tabel in de database.

Tijdens de fase definitiestudie was echter al besloten om niet iedere gebruiker een eigen loginnaam en wachtwoord te geven, maar dit per beveiligingsprofiel te doen. Een consequentie hiervan was dat elk profiel een ander wachtwoord nodig zou hebben en dat twee klanten dus niet hetzelfde profiel <hoog> met hetzelfde wachtwoord mochten gebruiken. Bovendien gaf mijn bedrijfsmentor aan dat sommige klanten misschien meer dan drie beveiligingsniveau's nodig hebben en dat niet iedere gebruiker van het hoogste niveau toegang mag hebben tot informatie van het midden-niveau. Kortom, het was vooraf niet te bepalen hoeveel profielen er per klant nodig zouden zijn en welke profielen toegang mochten hebben tot welke modulen. Hiermee verviel het <hoog> <midden> <laag> idee.

Er werd daarom besloten om per klant de profielen volledig dynamisch te maken.

Ook waren er verschillende manieren om de relatie tussen profielen en modulen in te stellen. Een profiel kan toegang hebben tot meerdere modulen en een module kan toegankelijk zijn voor meerdere profielen. Door deze N op N relatie ontstaat een nieuwe tabel waarin de relatie van een specifiek profiel met een specifieke module opgeslagen wordt.

Op dit moment kwamen we met het idee van een ‘standaardprofiel’ per klant. Dit is een profiel dat als standaard ingesteld kan worden. Alle modulen waar dit profiel toegang toe heeft worden vervolgens automatisch beschikbaar voor alle overige profielen van die klant. Dit maakte het overbodig om voor elk profiel alle toegankelijkheden in de database te zetten. Een mogelijkheid om dit op te lossen was door ook een relatie tussen klant en module te leggen waarin aangegeven werd welke modulen voor de gehele klant (alle profielen ervan) toegankelijk zouden zijn. Ook dit zou resulteren in een extra tabel.

Dit bleek echter geen goed idee, omdat er ook klanten zonder een standaardprofiel konden bestaan. Bovendien was het gewenst om de database zo simpel mogelijk te houden, omdat rekening gehouden moest worden met de uitbreiding ervan als er eenmaal andere pilots aan toegevoegd werden, zowel binnen als buiten de scope van de afstudeerperiode.

Er is vervolgens bekeken of er, door dit te versimpelen, gegevensredundantie zou ontstaan. Dit bleek zo te zijn. De relaties van elk profiel met elke module worden immers opgeslagen in de database, ongeacht of het standaardprofiel het al toegang verschaft tot deze module. Deze gegevensredundantie bleek echter zeer beperkt te zijn en bovendien zou het via PHP code te voorkomen zijn. Er is daarom bewust gekozen om deze situatie zo te ontwerpen.

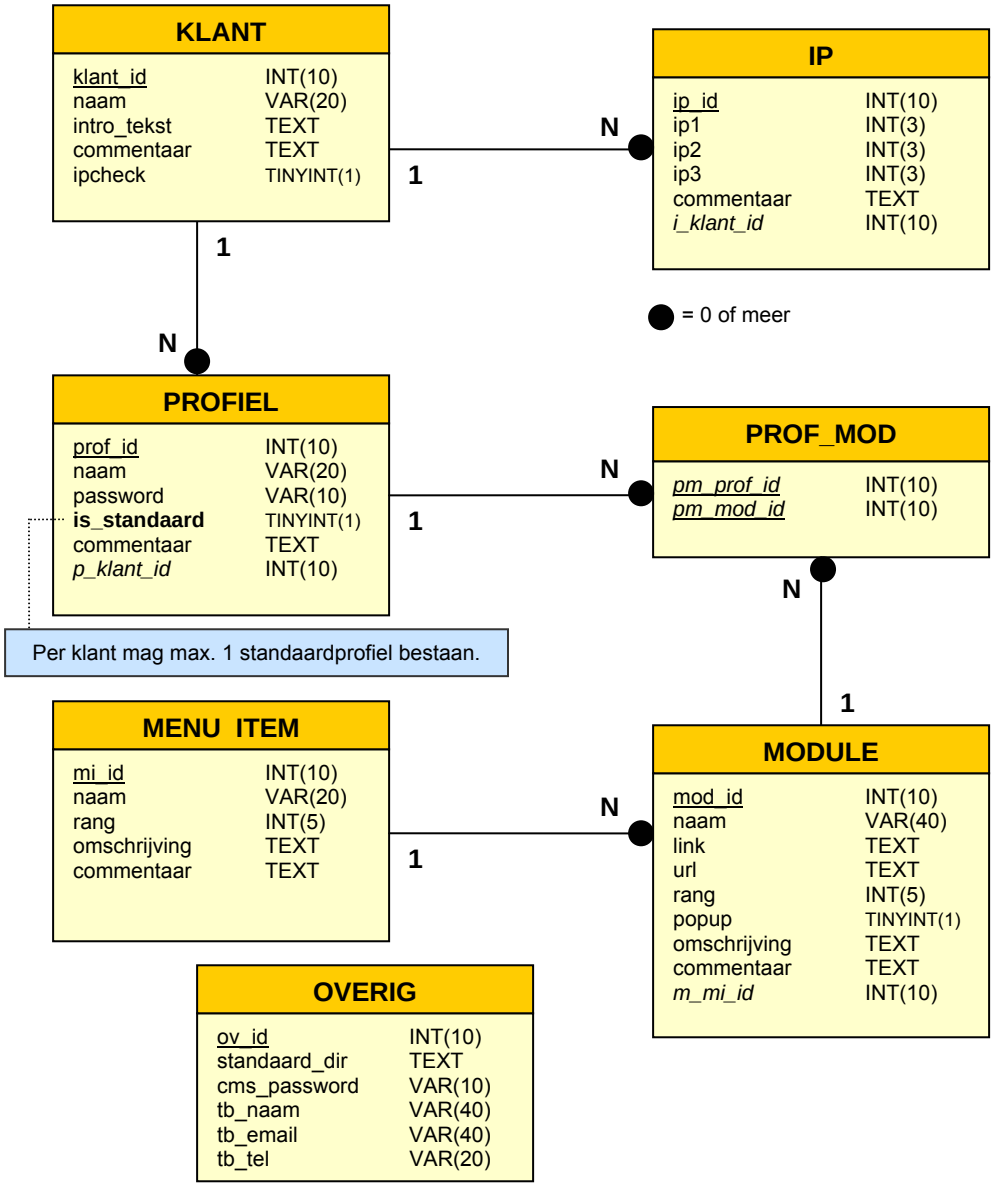
Implementatiemodel

Aan de hand van het objectmodel en relationeel representatiemodel heb ik een MySQL implementatiemodel opgesteld. De versie van MySQL die ik gebruikte bood echter geen Foreign Key ondersteuning. Alle relaties dienden dus in de PHP code geregeld te worden, evenals de ON DELETE en ON UPDATE statements. Bij het maken van het implementatiemodel zijn ook de MySQL kolomtypen van de attributen bepaald.

Databasemodel

Om te voorkomen dat ik na elke iteratie teveel tijd kwijt zou zijn aan het updaten van de documentatie en de verschillende modellen van de database, besloot ik het objectmodel en het implementatie model samen te persen in één enkel model: een objectmodel met de kolomtypen erachter vermeld. Dit model bood mij het voordeel dat ik zowel de structuur als de attributen en kolomtypen in één oogopslag kon inzien en aanpassen. Dit model zou ik dan ook tijdens de rest van de pilotontwikkeling veelvuldig gebruiken.

De uiteindelijke database van de basisstructuur ziet er als volgt uit:



Een objectmodel met zowel de relaties als de attributen en kolomtypen.

In bovenstaande afbeelding is ook de tabel ‘overig’ zichtbaar, die bij een latere iteratie van de pilot nodig bleek. Merk op dat de acties per object niet zijn weergegeven. Het attribuut ‘is_standaard’ mag per klant maar eenmaal de waarde TRUE hebben. Deze constraint zou uitgewerkt worden via de PHP code.

Het bouwen van de database

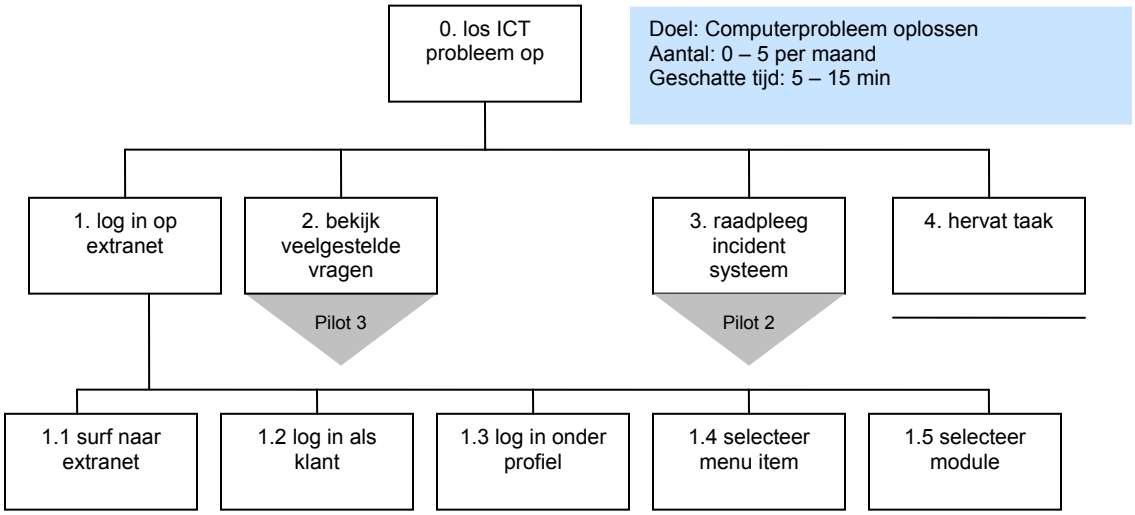
Aan de hand van dit model ben ik de database gaan realiseren in MySQL met een programma genaamd ‘phpMyAdmin’. Daar ik dit programma nog niet eerder gebruikt had, heb ik eerst een testdatabase gemaakt en enkele malen bewerkt. Toen ik eenmaal bekend was met de verschillende mogelijkheden van het programma, ben ik de extranet database gaan realiseren met behulp van het implementatiemodel. Hierbij traden aanvankelijk enkele fouten op die te

maken hadden met de specifieke kolomtypen en foreign key verwijzingen, maar dit heb ik snel opgelost door de MySQL documentatie te raadplegen. Na het invoeren van de databasestructuur in MySQL was dit eerste pilotdeel voltooid. Omdat de latere pilotdelen gegevens uit de database nodig zouden hebben om te kunnen functioneren, besloot ik elke tabel te vullen met een vijftal rijen testdata.

6.3.3 Pilotdeel 2: het klantdeel

Het specificeren van de bouweenheden

Nadat de database voltooid was, kon begonnen worden met het ontwikkelen van het klantdeel van het extranet. Tijdens de definitiestudie was een taakdiagram opgesteld waarop zichtbaar was welke stappen de extranetbezoeker diende uit te voeren om zijn taak te volbrengen:



Taakdiagram gewenste situatie bij de klant – het oplossen van een ICT probleem

Hieruit kwam naar voren dat er voor de klant in de nieuwe situatie wat stappen bijkwamen ten opzichte van de huidige situatie. De klant moet namelijk eerst inloggen als klant en onder een bepaald profiel. Van elk van deze stappen is bekeken wat het systeem moet doen en tonen en dit is uiteindelijk in een aantal bouweenheden opgesplitst.

Stap	Pilotdeel
1. surf naar extranet	▪ voorpagina extranet
2. log in als klant	▪ loginpagina1 om klant te bepalen
3. Log in onder profiel	▪ loginpagina2 om profiel te bepalen
4. Selecteer menu item	▪ layout extranet ▪ hoofdmenu ▪ startpagina (zichtbaar na inloggen, dus niet zelfde als voorpagina)
5. Selecteer module	▪ pagina waarop modulen weergegeven worden, afhankelijk van klant en profiel waarop gebruiker ingelogd is.

Ik heb gekozen om de bouweenheden ook op deze volgorde te ontwikkelen, omdat de onderdelen functioneel afhankelijk zijn van elkaar. Het is bijvoorbeeld pas mogelijk om de tweede loginpagina te maken als de eerste voltooid is, want de gegevens op de tweede loginpagina worden gehaald uit het resultaat van de eerste pagina.

Het verbeteren van de inlogprocedure

Omdat het inloggen voor de klant een extra taak is en dus een nadeel vormt ten opzichte van de huidige situatie, ben ik op zoek gegaan naar manieren om dit nadeel weg te werken. Het zou gebruikers van het systeem kunnen afschrikken. Zeker omdat een groot deel van deze gebruikersgroep weinig ervaring heeft met computers, kon dit inloggen een te hoge drempel vormen.

Hiertoe heb ik een collega geïnterviewd die ervaring heeft met het ontwerpen van PHP applicaties. Ik heb hem het probleem voorgelegd en gevraagd of hem gevraagd oplossingen te bedenken om inlogprocedures zo gemakkelijk mogelijk te maken voor gebruikers. Tijdens dit interview hebben ik het vraagstuk uiteindelijk in twee delen opgesplitst:

- Het systeem toont modulen afhankelijk van de klant en het profiel waaronder de gebruiker ingelogd is.
    - Hoe weet het extranet bij welke klant de gebruiker hoort?
    - Hoe weet het extranet bij welk profiel het extranet hoort?

Door deze twee vragen vanuit het perspectief van het extranet te bekijken was het gemakkelijker te bepalen welke gegevens van de gebruiker nodig zijn voor het systeem om de gebruiker te herkennen. Het doel van inloggen is namelijk het herkennen van de inlogger.

Uit dit interview kwamen enkele ideeën naar voren. Het zou mogelijk zijn om per klant een laagste beveiligingsprofiel in te stellen waar alle medewerkers van een klant standaard onder vallen. Iemand die naar het extranet toegaat wordt dan ook automatisch ingelogd onder dit laagste profiel, zonder een wachtwoord in te hoeven vullen. Als men vervolgens alsnog een module wil raadplegen die voor een hoger beveiligingsprofiel bedoeld is, kan men op dat moment het wachtwoord van dit hogere profiel invullen.

Hierbij bleef echter nog de kwestie open van hoe het extranet kan weten van welke klant een gebruiker afkomstig is. Een medewerker van de ene klant mag namelijk niet automatisch ingelogd worden onder het laagste beveiligingsprofiel van een andere klant. Ook hiervoor is een oplossing bedacht. Het bleek dat elke klant van Xtilton een aantal IP-ranges heeft (afhankelijk van de locatie van de klant) en het met PHP mogelijk is het IP-adres van een extranetbezoeker te detecteren. Door deze IP-adressen in de database te zetten kan het systeem deze vergelijken met het gedetecteerde IP-adres van de bezoeker. De inlogprocedure zou dan als volgt verlopen:

1. De bezoeker surft naar het extranet.
2. Het extranet detecteert zijn IP-adres.
3. Het extranet vergelijkt zijn IP-adres met de IP-ranges in de database.
4. Het extranet weet nu bij welke klant de bezoeker hoort.
5. Het extranet logt de gebruiker in onder het laagste beveiligingsprofiel van deze klant.

Op deze manier wordt een gebruiker dus volledig automatisch ingelogd.

Iteratie pilotdeel 1: de database

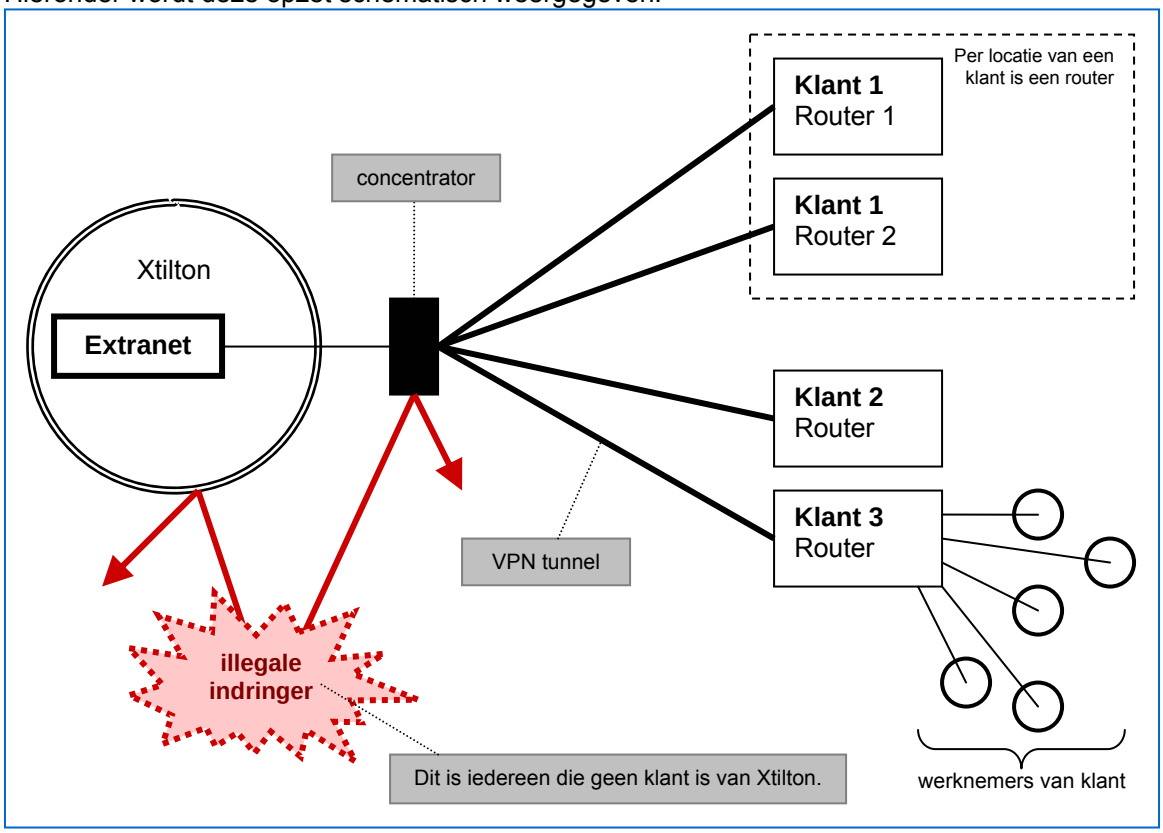
Door deze vernieuwde inlogprocedure waren er wijzigingen in de database nodig. Het pilotdeel database is dan ook geïtereerd en het object ‘IP-adres’ en het attribuut ‘is_standaard’ zijn toegevoegd aan de database.

Beveiliging tegen IP-spoofing

Het IP-detectie systeem was niet ongevoelig voor fraude. De PHP functie die hiervoor beschikbaar is kan namelijk gemanipuleerd worden door de gebruiker door middel van 'IP-spoofing', waarbij zij computer doet alsof het een ander IP-adres heeft. Er bestond dus de mogelijkheid dat gebruikers die niet bij een klant horen, toch zouden kunnen inloggen als een bepaalde klant.

Aan een collega van netwerkbeheer heb ik gevraagd of hier iets tegen gedaan kon worden. Uit dit gesprek bleek dat er tussen Xtilton en sommige klanten VPN (Virtual Private Network) tunnels opgezet waren. Hierbij wordt gebruik gemaakt van de publieke telecommunicatie infrastructuur, maar is de onderlinge datastroom beveiligd zodat alleen degenen in het virtuele netwerk deze kunnen bekijken. Door het extranet via VPN tunnels voor klanten beschikbaar te stellen, zou de rest van het internet hier niet in kunnen binnenkomen. IP-spoofing heeft dan ook geen effect meer, tenzij dit door iemand binnen het virtuele netwerk gedaan wordt.

Hieronder wordt deze opzet schematisch weergegeven:



Beveiliging d.m.v. VPN tunnels

Cookies of sessions

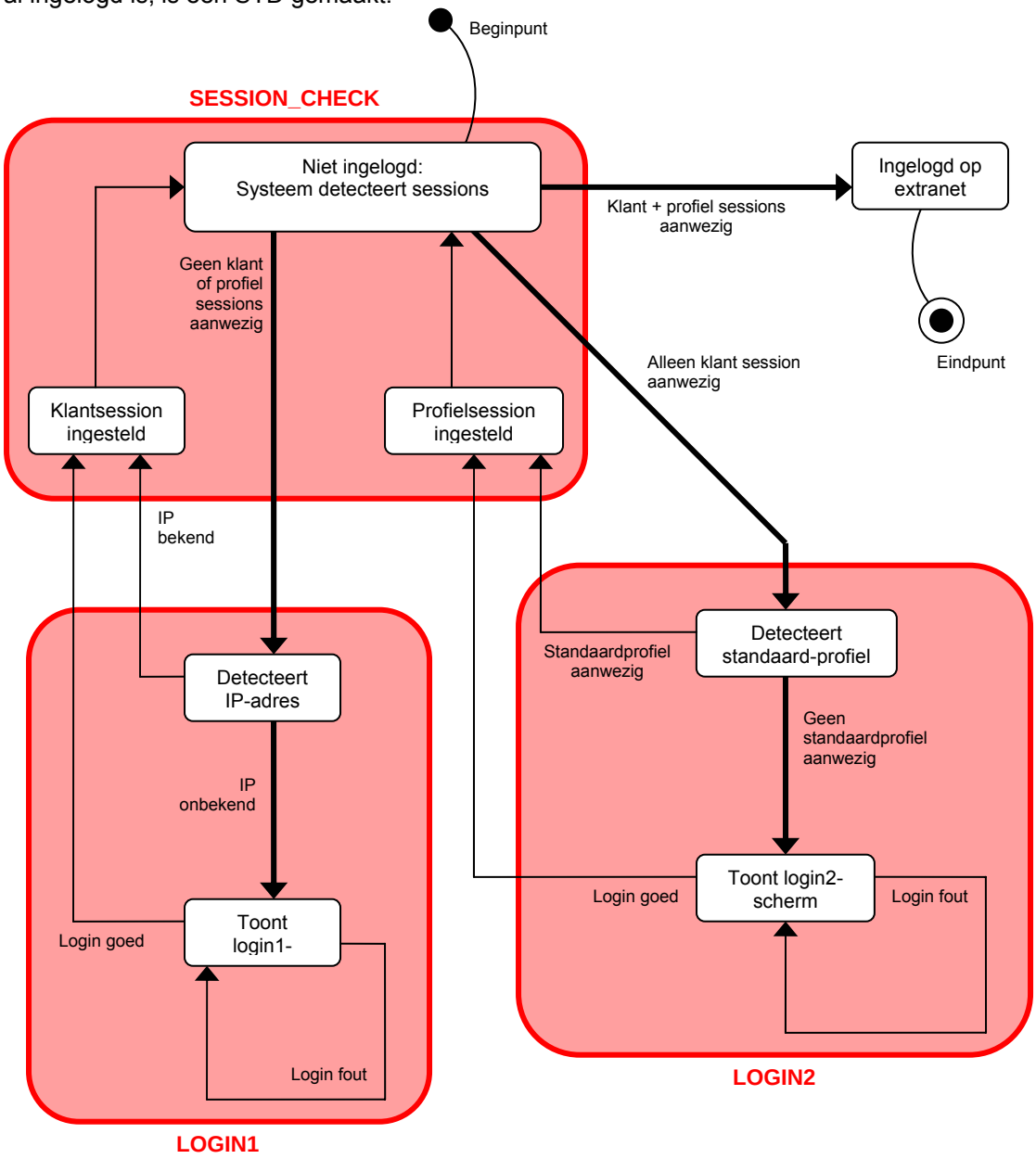
Het systeem dient na het inloggen van de gebruiker op elk moment te weten bij welke klant en profiel deze hoort. Aan de hand van deze gegevens zullen namelijk bepaalde modulen wel en niet toegankelijk zijn. Er waren twee mogelijkheden om deze gegevens op elk moment bekend te houden:

- Cookies
- Sessions

Er is gekozen voor sessions, omdat deze op de server opgeslagen worden. Cookies worden op de harde schijf van de gebruiker opgeslagen en kunnen dus door hem aangepast worden.

State Transition Diagram van de inlogstatus

Voor het geheel van de inlogprocedure heb ik eerst zelf geprobeerd een schema op te stellen met de beslissingspunten per proces. Dit schema mislukte, omdat ik verschillende zaken door elkaar haalde, zoals het weergeven van de status van het systeem en de beslissingsmomenten. Uiteindelijk heb ik hier een State Transition Diagram(STD) gemaakt van gemaakt. Hiermee wilde ik duidelijk maken welke gebeurtenissen er in de inlogprocedure plaatsvinden, welke informatie er hiervoor nodig was en in welke staten dit resulteert. Hoewel 'inlogstatus' eigenlijk geen class of object was, kon je dit geheel toch als een soort class zien. Ook van het wisselen van profiel, als al ingelogd is, is een STD gemaakt.



State Transition Diagram van inlogstatus

Op dit State Transition Diagram heb ik de verschillende staten en gebeurtenissen ingedeeld in PHP pagina's (login1, login2, session_check). Dit heb ik gedaan zodat ik kon bepalen uit hoeveel pagina's de bouweenheid zou bestaan en welke functies er in de PHP code per pagina nodig waren.

Het bouwen van de inlogprocedure

Er was inmiddels bekend welke PHP pagina's er nodig waren en welke functies deze dienden te bevatten, dus kon begonnen worden met de bouw van het pilotdeel.
Hiervoor waren enkele simpele schetsen nodig van de inlogschermen. De gebruiker zou tijdens de inlogprocedure waarschijnlijk echter geen enkel scherm zien, omdat dit automatisch gebeurde.

Inlogscherm (dit krijgt de gebruiker te zien als zijn IP-adres niet herkend wordt)

Nadat de schetsen voltooid waren, is begonnen met het schrijven van de PHP code. Hieronder is de IP-detectie functie van de pagina login1.php te zien:

```
//*****************************************************************
//1.
//BEGIN IP DETECTIE
//*****************************************************************
  //detecteer IP
  $ip = $_SERVER['REMOTE_ADDR'];

  //splits IP op
  $octets = explode(".", $ip);

  $ip1 = $octets[0];
  $ip2 = $octets[1];
  $ip3 = $octets[2];
  $ip4 = $octets[3];

  //=============================
  //haal de gegevens op uit IP met een query
  $query_ip =

      "SELECT i.i_klant_id
      FROM ip i, klant k
      WHERE ip1 = ' ".$ip1." '
      AND ip2 = ' ".$ip2." '
      AND ip3 = ' ".$ip3." '
      AND ipcheck = '1'
      AND k.klant_id = i_klant_id
      GROUP BY i_klant_id
      ";

  $result_ip = mysql_query($query_ip) or
  die ("Ongeldige query: " . mysql_error() );
```

Gaat rechts verder >>

<< Vervolg van links

```
//=============================
  $detectie_succesvol = "nee";
  $aantal_gedetecteerde_klanten = "0";

  //controleer of IP in database staat
  while(list($i_klant_id) = mysql_fetch_row($result_ip))
        {
        $detectie_succesvol = "ja";
        $gedetecteerde_klant = $i_klant_id;
        $aantal_gedetecteerde_klanten++;
        }
  //=============================
  //als gedetecteerde IP in database staat
  //EN als er maar 1 klant gedetecteerd is
  if ($detectie_succesvol == "ja" AND
$aantal_gedetecteerde_klanten == "1")
     {
     //log in als klant
?>
<HTML><BODY>
<FORM name="goed" ACTION="session_check.php"
METHOD="POST">
<INPUT TYPE="hidden" NAME="ingelogde_klant_id"
VALUE= "<? echo $gedetecteerde_klant; ?>">
<SCRIPT TYPE="text/javascript">document.goed.submit();
</SCRIPT>
<INPUT TYPE="submit" VALUE="Ga verder">
</FORM></BODY></HTML>
<?
     }
//EIND IP DETECTIE
//*****************************************************************
```

De IP-detectie in het bestand login1.php

In dit figuur is te zien dat er bij de verschillende stukken code commentaar is neergezet (de stukken met '//' ervoor). Dit is gedaan om de code begrijpelijk te maken voor mensen die het in de toekomst zouden moeten aanpassen, maar ook voor mezelf wanneer ik bij een volgende iteratie wijzigingen zou moeten aanbrengen. Hetzelfde geldt voor de variabelen in de code. Deze heb ik voorzien van begrijpelijke namen.

Omdat het systeem binnen het Xtilton productienetwerk getest zou worden, zou deze IP-detectie aanvankelijk het volledige IP-adres van de extranetbezoeker detecteren. Tijdens de invoering van het systeem waren echter slechts de eerste drie IP-octetten van de klant van belang, omdat deze per locatie van de klant uniek waren. Als er in die situatie toch vier octetten gedetecteerd worden, dan zouden er duizenden verschillende IP-adressen in de database ingevoerd moeten worden, omdat bij iedere computer het vierde IP-octet anders is. Met drie octetten zou dit beperkt blijven tot ongeveer veertig verschillende IP-adressen.

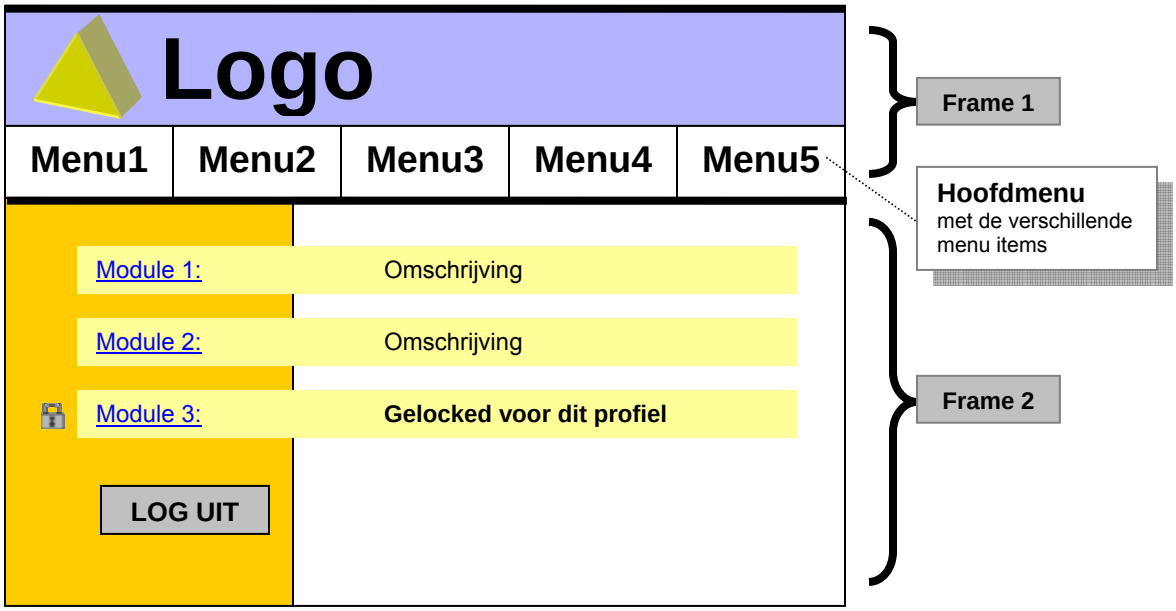
Op deze manier zijn alle pagina's van de inlogprocedure in elkaar gezet.

Het hoofdscherm van het extranet

De verschillende onderdelen van de inlogprocedure waren nu voltooid en een bezoeker zou dus het daadwerkelijke extranet kunnen bereiken. Ik heb daarom een navigatieschema gemaakt van wat de gebruiker binnen het extranet zou zien en hoe hij tussen de verschillende schermen van deze binnenkant kan navigeren. Het was de bedoeling deze navigatie zo simpel mogelijk te houden, zodat computerleken het ook konden begrijpen.

Het bepalen van de layout

Van de layout van het extranet heb ik enkele schetsen gemaakt. Uit de styleguide was naar voren gekomen dat het extranet op de internetsite moest lijken. Deze eis heb ik verwerkt in de schetsen. De schetsen heb ik vervolgens getoond heb aan mijn bedrijfsmentor, die de internetsite gedeeltelijk ontworpen had. Na enkele aanpassingen is uiteindelijk een schets geselecteerd:



Schets van extranet hoofdscherm

Er is gekozen voor frames, omdat het hierdoor voor de beheerders van het extranet erg gemakkelijk wordt om modulen toe te voegen. Bij een niet-frames oplossing was het noodzakelijk dat op iedere pagina het hoofdmenu ingevoegd wordt in de PHP code. Wil een beheerder dan een nieuwe module toevoegen, dan moet hij dus in de PHP code gaan wroeten. Voor sommige

modulen is dit zelfs niet eens mogelijk, omdat dit volledig externe systemen zijn waaraan Xtilton niets kan veranderen.

Het was mij bekend dat er enkele nadelen aan frames kleven, zoals bijvoorbeeld dat de gebruiker niet de exacte URL van de pagina kan zien, en dat bij het verversen van de pagina (F5) niet de laatst geopende pagina in het frame geladen wordt. Ook wanneer iemand via een zoekmachine een aparte pagina binnenkomt, wordt het frame met het hoofdmenu niet weergegeven. Toch is gekozen om frames te gebruiken, omdat deze nadelen niet opwogen tegen de voordelen.

Op de schets is tevens een module te zien met een slot icoontje ervoor. Dit houdt in dat de module niet toegankelijk is voor het profiel waarop de gebruiker ingelogd is, maar wel toegankelijk is voor andere profielen van de klant waar de gebruiker toe behoort. Als de gebruiker toch op de module klikt, zal hij een wachtwoord moeten invoeren. Voor deze structuur is gekozen om het extranet zo transparant mogelijk te houden. Het wordt hierdoor namelijk voor alle gebruikers duidelijk welke modulen er zijn, of deze nu toegankelijk zijn of niet. Een manager bijvoorbeeld hoeft dan niet meteen onder zijn profiel in te loggen om erachter te komen dat een module bestaat.

Het bouwen van de extranetschermen

Hierna zijn de verschillende schermen van de laatste twee bouweenheden gemaakt in PHP, zoals het hoofdmenu, de startpagina en de pagina met daarop de modulen. De startpagina is klantspecifiek gemaakt, zodat elke klant een andere voorpagina te zien krijgt.

Het startscherm van het extranet

De meeste tijd ging echter zitten in het weergeven van de toegankelijke en niet-toegankelijke modulen, omdat hier de gegevens uit de sessions gehaald werden en aan de hand daarvan twee arrays met databasegegevens met elkaar vergeleken werden: de toegankelijke profielen van de gehele klant, vergeleken met de toegankelijke modulen van het profiel waarop ingelogd is.

Het extranetscherm dat de verschillende modulen toont.

Hierbij kreeg ik te maken met een aantal PHP foutmeldingen. Hier ziet u een voorbeeld een foutmelding waar ik tegenaan liep:

Parse error: syntax error, unexpected ',', expecting T_PAAMAYIM_NEKUDOTAYIM in **C:\web\p\extranet\CMS\functionaliteiten\func_edit.php** on line **21**

Aanvankelijk dacht ik dat het hier om een Indonesische tekst ging, maar het bleek Hebreeuws te zijn. Om dergelijke foutmeldingen op te lossen heb ik de PHP documentatie geraadpleegd en mezelf geregistreerd voor een online PHP forum. Op dit forum kon ik mijn problemen voorleggen aan anderen die vervolgens tips gaven hoe het beter aan te pakken was. Door deze stappen te zetten werden de foutmeldingen opgelost.

Uiteindelijk heb ik met behulp van enkele collegae de inlogprocedure getest binnen het netwerk van Xtilton. Enkele fouten zijn hiermee opgespoord en opgelost. Omdat er vanuit de gebruikerskant nog weinig te testen was (slechts inloggen, menu item aanklikken, module aanklikken) besloot ik voor dit pilotdeel geen aparte gebruikerstests uit te voeren. Pas wanneer het volgende pilotdeel (het CMS) voltooid was, zou ik beide delen tegelijk laten testen.

6.3.4 Pilotdeel 3: het Xtilton deel (CMS)

Dit is het deel dat door Xtilton medewerkers gebruikt wordt om het extranet te beheren. Alle objecten in de database moeten hiermee bewerkt kunnen worden, afhankelijk van de taak de beheerder probeert uit te voeren. De verschillende taken die men moet kunnen uitvoeren zijn tijdens de ontwikkeling van deze pilot wat verder gespecificeerd.

Het specificeren van de bouweenheden

Ik besloot om de eerste opzet van het CMS zodanig te maken dat ieder object van de database in eigen onderdeel van het CMS te bewerken zou zijn. Pas als deze onderdelen voltooid waren, zou

ik bij de volgende iteratie kijken naar hoe de gebruiker zijn taak het best kan uitvoeren in het systeem en vervolgens de navigatiestructuur aanpassen en de onderdelen met elkaar verbinden. Hier is om de volgende redenen voor gekozen:

- Het levert herbruikbare stukken code op.
- De relaties tussen de onderdelen van het CMS kunnen uiteindelijk beter uitgewerkt worden als alle componenten al aanwezig zijn.
- Er zijn minder schetsen nodig.
- De componenten zullen gemakkelijker te bewerken zijn.

Het object ‘profiel’ bijvoorbeeld zou op deze manier aanvankelijk als apart onderdeel te bewerken, verwijderen en toe te voegen zijn. Echter een taak van de gebruiker is bijvoorbeeld het toevoegen van een profiel aan een klant. Om dit te doen, wil de gebruiker eerst de klant selecteren, vervolgens bekijken welke profielen deze allemaal al heeft, en tot slot een nieuw profiel toevoegen. Om dit proces te versoepelen zou de navigatiestructuur pas in een later stadium specifiek op dergelijke gebruikerstaken aangepast worden. Dit pilotdeel is dan ook opgedeeld in bouweenheden afhankelijk van de objecten in de database die bewerkt moesten kunnen worden:

- Klanten
- Profielen
- IP-adressen
- Menu-items
- Modulen

De overige bouweenheden zijn:

- Inlogstructuur
- Layout
- Toegangsbeheer
- Overig

Vervolgens zijn de functionele afhankelijkheden van deze bouweenheden bepaald. Aan de hand hiervan zijn de bouweenheden geprioriteerd.

De inlogstructuur

De inlogstructuur van het CMS was een stuk eenvoudiger dan bij het klantdeel. Er waren slechts enkele mensen van Xtilton die het CMS zouden mogen gebruiken, en geen van hen had meer bevoegdheden dan de ander. Het enige wat dus nodig was, was een enkel wachtwoord. Aanvankelijk was dit wachtwoord in de broncode van de loginpagina opgeslagen, maar later bleek dat men dit ook wilde kunnen aanpassen. Hiertoe is het CMS wachtwoord in de database opgenomen.

Het bepalen van de layout

Er is gekozen om de layout vrijwel hetzelfde te maken als die van het klantdeel, zodat onderdelen uit het klantdeel bruikbaar zouden zijn voor dit pilotdeel. Tevens zou dit een consistente vormgeving opleveren.

De navigatiestructuur bepalen

De navigatiestructuur was in eerste instantie zodanig opgesteld dat deze voor vrijwel elke bouweenheid hetzelfde was. Hierdoor was de navigatie per scherm consistent en hoefden er weinig schermschetsen gemaakt te worden. Later, bij de iteratie van dit pilotdeel, is de navigatiestructuur meer gericht op de volgorde waarop de gebruikergroep zijn taken uitvoert.

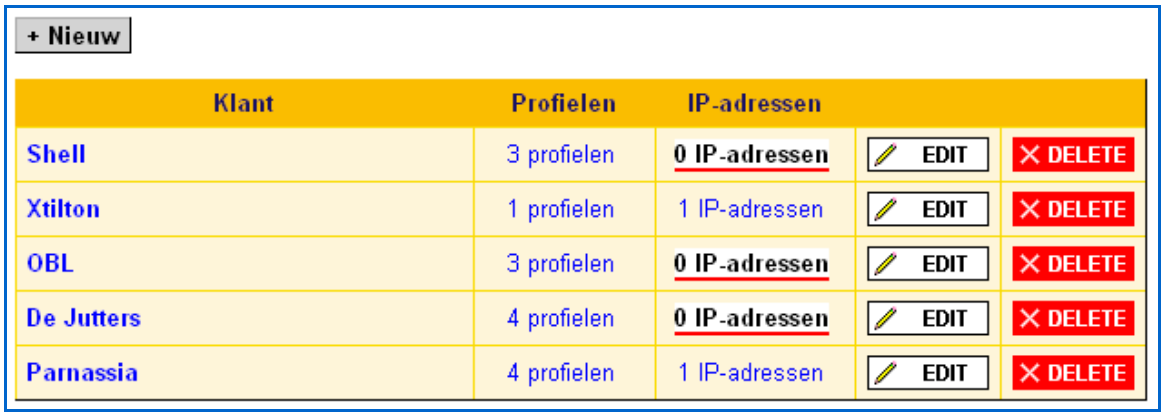
Herbruikbare componenten bepalen

De volgende bouweenheid die gemaakt werd, was die van de klanten. Deze bouweenheid was het minst complex en zou dus een goede basis vormen om herbruikbare componenten uit te kunnen afleiden.

De functionaliteiten uit de bouweenheid ‘klanten’ zijn:

- Klanten weergeven
- Klanten toevoegen
- Klanten bewerken
- Klanten verwijderen

De meeste van deze functionaliteiten zouden ook in andere bouweenheden gebruikt worden. Het zijn dus een soort standaardfunctionaliteiten.



Standaardfunctionaliteiten: toevoegen/bewerken/verwijderen

De klanten worden in een lijst weergegeven. Bovenin staat een knop om nieuwe klanten toe te voegen. Rechts staan knoppen om een klant te bewerken of te verwijderen. Om te zorgen dat er geen lege plekken in de lijst terechtkomen, wordt er in de PHP code gecontroleerd of er geen lege waarde, nul of spatie als klantnaam ingevuld wordt. Al deze functionaliteiten zouden ook in de rest van het CMS vaak gebruikt worden en dienden dus als herbruikbare componenten.

Op deze afbeelding zijn ook al enkele dingen zichtbaar die bij de iteratie van het pilotdeel ontwikkeld zijn, namelijk de kolommen ‘profielen’ en ‘IP-adressen’. Deze geven weer hoeveel profielen en IP-adressen een klant bevat. Als een klant geen enkel profiel heeft, of geen enkel IP-adres heeft, dan geeft het systeem aan dat er iets niet klopt. Klikt de gebruiker op ‘3 profielen’, dan ziet hij welke profielen dit zijn en kan hij deze toevoegen, bewerken, verwijderen en de relaties ervan met andere objecten zien (bijv. toegankelijke modulen voor een profiel).

Bij het verwijderen van een klant wordt eerst weergegeven welke klant de gebruiker zal verwijderen. De gebruiker moet zijn actie bevestigen om de klant daadwerkelijk te verwijderen.

Om te voorkomen dat gebruikers per ongeluk op ‘verwijderen’ klikken in plaats van op ‘bewerken’, zijn er voor beide functionaliteiten verschillende afbeeldingen gebruikt. Aanvankelijk waren dit gewone hyperlinks, maar deze leken teveel op elkaar.

Dit is nog verder uitgewerkt door tevens het gehele verwijderscherm een rode kleur te geven, zodat dit zich duidelijk onderscheidt van het bewerkscherm.

Omdat een profiel en een IP-adres altijd bij een klant moeten horen, zullen deze ook verwijderd worden als de klant verwijderd wordt. Deze verwijder queries zijn via de PHP code geregeld, omdat het niet mogelijk was dit via MySQL te doen. De versie van MySQL die ik gebruikte bood immers geen ondersteuning voor vreemde sleutels.

De roze stukken tekst op het verwijderscherm zijn hyperlinks waarmee de gebruiker kan zien welke profielen en IP-adressen meeverwijderd worden als hij de klant verwijdert.

Objecten bewerken

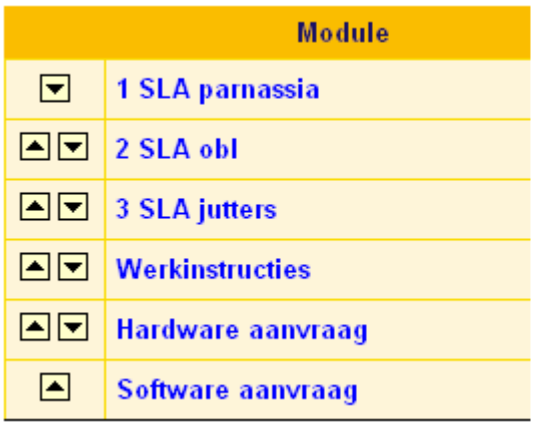
Tijdens de pilotontwikkeling van de database waren de verschillende acties per object bepaald en weergegeven in een objectmodel. Hieronder ziet u het object 'module':

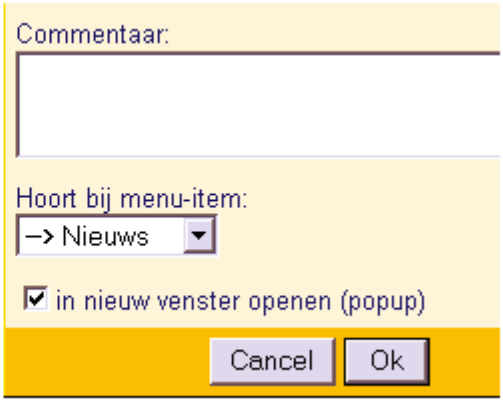
Module
Attributen -naam -link (op extranetserver) -url (op externe server) -rang -popup -omschrijving -commentaar
Acties -modulen toevoegen/bewerken/verwijderen -rang aanpassen -popup aan/uitzetten -modulen verwisselen van menu item -modulen toegankelijk maken voor profielen

Het object 'module'uit het datamodel

Het toevoegen, bewerken en verwijderen van modulen waren standaard acties die binnen de herbruikbare componenten vielen. Er waren echter ook acties die specifiek voor het object module golden.

Zo moest het mogelijk zijn de modulen op het extranet te rangschikken, zodat in het klantdeel de belangrijkste modulen bovenaan verschijnen wanneer de gebruiker op een menu-item klikt.



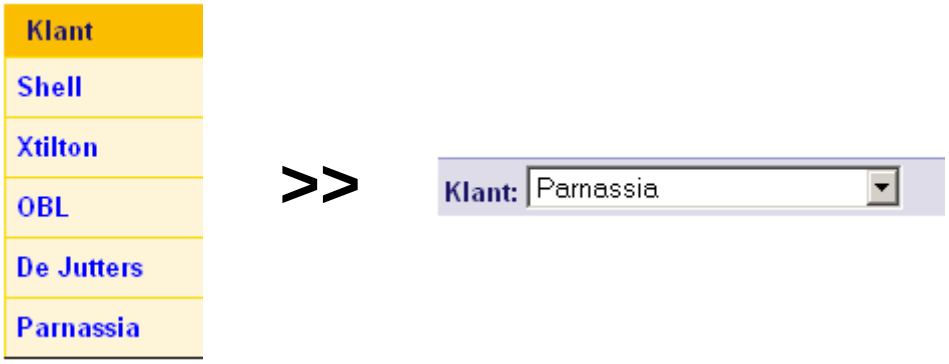


Op de linker afbeelding zijn pijltjes te zien waarmee de beheerder de volgorde van modulen kan wijzigen. Op de rechterafbeelding is te zien hoe de beheerder modulen kan verwisselen van menu-item, en ervoor kan zorgen dat ze in een nieuw scherm geopend worden.

Bij de andere objecten zijn de acties ook op deze manier uitgewerkt in het systeem.

Dropdownmenu's

Aanvankelijk werden klanten weergegeven met behulp van hyperlinks, in plaats van een dropdownmenu. Deze hyperlinks namen echter teveel ruimte in beslag, waardoor ze op een aparte pagina moesten worden gezet. Bij de gebruikerstests bleek vervolgens dat het navigeren tussen deze aparte pagina en de pagina's met beveiligingsprofielen te omslachtig was.
Er is toen gekeken of deze navigatie door een dropdownmenu te vervangen was.



Enkele voordelen van dropdownmenu's zijn:

- Ze kosten kost weinig ruimte op de pagina, dus overzichtelijk
- De navigatie kost minder stappen, omdat schermen weggelaten kunnen worden
- Dropdownmenu's kunnen gemakkelijk met het toetsenbord gebruikt worden (goed tegen RSI).
- De navigatie op de verschillende schermen wordt consistenter

Enkele nadelen van dropdownmenu's zijn:

- De gebruiker kan niet in één oogopslag zien welke waarden het dropdownmenu bevat.
- Er zijn geen mooie buttonplaatjes van te maken.

Er is uiteindelijk besloten om dropdownmenu's te gebruiken, omdat de voordelen opwogen tegen de nadelen. Door Javascript te gebruiken is het mogelijk om met een dropdownmenu naar verschillende pagina's te navigeren. De waarden van het dropdownmenu worden opgehaald uit de database en bevatten tevens variabelen die meegegeven worden aan de volgende pagina.

Het testen van het pilotdeel

Toen dit pilotdeel zich eenmaal in een testbaar stadium bevond besloot ik het door verscheidene mensen binnen Xtilton te laten testen. Met behulp van de MI-33 sheets en IP-project documentatie heb ik de testtechnieken bepaald.

Voor de tests zou ik de volgende testtechnieken gebruiken:

- **Cognitive Walkthrough**
  Door taken in stappen onder te delen en de gebruiker deze stappen te laten uitvoeren, kunnen onduidelijkheden in de User Interface van het systeem ontdekt worden.
- **Think Aloud**
  Bij Think Aloud wordt de testpersoon aangeraden hardop te zeggen wat hij ervaart tijdens het doorlopen van de scenario's. Hierdoor kan de observator direct horen hoe de testpersoon reageert op het systeem en eventuele misverstanden opsporen. Onduidelijkheden in de User Interface die tijdens de Cognitieve Walkthrough misschien niet waargenomen zijn, kunnen dan toch aan het licht komen.

Er hebben in totaal twee tests plaatsgevonden met per test één gebruiker. Tijdens de tests heb ik de gebruikers enkele scenario's laten doorlopen, bestaande uit verschillende stappen. Terwijl ze deze scenario's doorliepen, noteerde ik mijn bevindingen op een CWT-formulier. Dit formulier heb ik op zo opgesteld het voor mij snel en eenvoudig in te vullen was. Uit ervaring met eerdere gebruikerstests, die ik tijdens mijn studie in verschillende projecten heb uitgevoerd, wist ik namelijk dat dit een knelpunt tijdens de test kon zijn. Bij enkele van deze projecten waren overdreven uitgebreide formulieren gebruikt, wat als gevolg had dat de formulieren grotendeels onbeschreven bleven, omdat er geen tijd was om ze in te vullen of op alle aandachtspunten te letten. Dit wilde ik voorkomen dus heb ik het formulier simpel gehouden. Door een visuele weergave van het systeem op dit formulier te plaatsen zou ik tevens ogenblikkelijk gebruikersacties kunnen noteren. Naast deze twee testtechnieken heb ik de testpersonen na afloop van de tests ook enkele vragen gesteld over de observaties die ik tijdens de test deed.

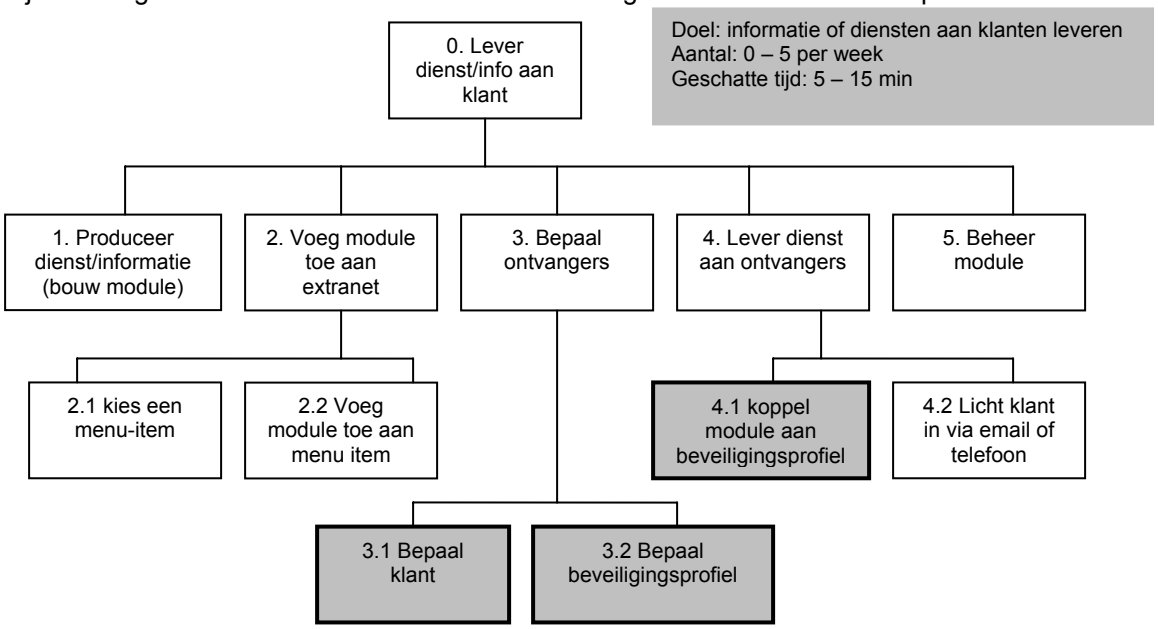
De resultaten van de gebruikerstests zijn in verbeterpunten uitgewerkt, welke in de iteraties van dit pilotdeel gerealiseerd zouden worden.

Enkele van het dozijn knelpunten die uit deze tests naar voren kwamen waren:

- De navigatie tussen de verschillende schermen verliep niet intuïtief. Het was per object apart ingedeeld en er waren geen hyperlinks die ze met elkaar verbonden.
- De gebruiker kreeg geen feedback van het systeem na het uitvoeren van een actie.
- Van sommige onderdelen was niet duidelijk wat de bedoeling was (met name het onderdeel toegangsbeheer).

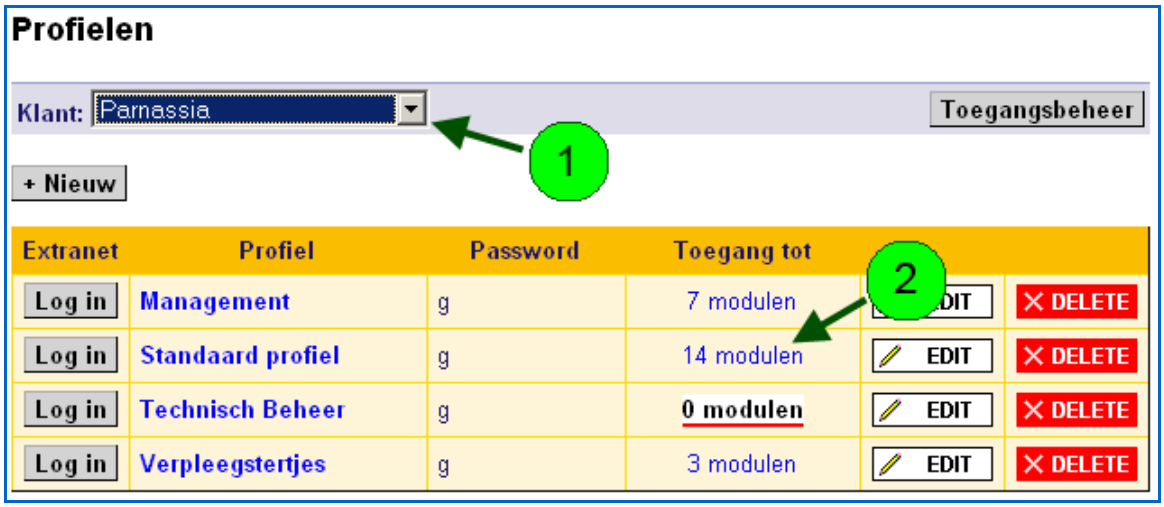
Iteratie van het pilotdeel

Bij de iteratie van het pilotdeel werd de navigatiestructuur op verschillende plekken aangepast. De noodzaak deze dergelijke aanpassingen bleek uit de taakdiagrammen van de definitiestudie. Tijdens de gebruikerstests is de behoefte hieraan nog eens verder onderstreept.



In bovenstaand taakdiagram is bijvoorbeeld te zien in welke volgorde een Xtilton medewerker de stappen uitvoert die nodig zijn om een module aan het extranet toe te voegen, en vervolgens een bepaalde groep gebruikers binnen een klant toegang te verlenen tot deze module.

In onderstaande schermen wordt weergegeven hoe dit in het systeem is uitgewerkt.
Er wordt hier aangenomen dat de dienst al bestaat en als module in het systeem beschikbaar is.



Stap 3.1 van het taakdiagram wordt op dit scherm uitgevoerd bij punt 1. Hier selecteert de extranetbeheerder een klant, waarna onderin het scherm een lijst met beveiligingsprofielen van die klant tevoorschijn komt. Per profiel kan de beheerder direct zien tot hoeveel modulen deze toegang heeft. Stap 3.2 van het taakdiagram wordt uitgevoerd bij punt 2.

Zodra men op de hyperlink bij punt 2 klikt, wordt namelijk een scherm zichtbaar waarop wordt weergegeven tot welke modulen het profiel toegang heeft.

Het koppelen van de module(n) aan het beveiligingsprofiel (stap 4.1 van het taakdiagram) is vervolgens zeer eenvoudig te doen door de checkboxes aan te vinken. Dit wordt gedaan bij punt 3 in de afbeelding. Tot slot kan de gebruiker zijn keuzes bevestigen door bij punt 4 het geheel op te slaan. De module is nu toegankelijk voor de klant.

Het inlichten van de klant over het feit dat er een nieuwe module beschikbaar is, is niet door het extranet geautomatiseerd. De individuele gebruikersgegevens staan immers niet in de extranetdatabase opgeslagen, maar in verschillende andere systemen binnen Xtilton.

Uit de tests was tevens naar voren gekomen dat het systeem te weinig feedback gaf na gebruikersacties. Hier heb ik tijdens de iteratie van het pilotdeel verandering in gebracht.

Als een handeling van de gebruiker succesvol door het systeem wordt afgehandeld, verschijnt er een stuk tekst met een groene achtergrond. Als de gebruiker een fout maakt (hij vult bijvoorbeeld niet alle IP-octetten in bij het toevoegen of bewerken van een IP-adres, dan verschijnt er een stuk tekst met een rode achtergrond. Het zou hierdoor duidelijk zijn wanneer er iets goed of fout gaat.

Bij de onderdelen in de User Interface waarvan tijdens de tests bleek dat ze niet geheel duidelijk waren, heb ik korte stukjes relevante uitleg erbij gezet. Het ging hier bijvoorbeeld om het aangeven van het verschil tussen standaard- en gewone profielen.

Uiteindelijk heeft deze iteratie twee weken in beslag genomen. Aan het eind van deze iteratie is het pilotdeel opnieuw getest door een gebruiker. Inmiddels waren de belangrijkste verbeterpunten van de vorige iteraties al uitgewerkt. Als resultaat van deze gebruikerstest kwamen er nog enkele verbeterpunten naar voren. Het project bevond zich inmiddels echter al in de twaalfde week en er was volgens de planning nog anderhalve week beschikbaar om de pilot af te ronden. Ik besloot eerst het laatste pilotdeel, de vormgeving, te gaan ontwikkelen. Als dat pilotdeel voltooid was, zou ik de gehele pilot (zowel de database, klantdeel, CMS en vormgeving) laten testen en de verbeterpunten vervolgens tijdens de iteratie van de gehele pilot uitwerken.

6.3.5 Pilotdeel 4: de vormgeving

Nu de ontwikkeling van deze pilotdelen grotendeels voltooid was, kon er begonnen worden met het verbeteren van de vormgeving. Dit pilotdeel was bewust als laatst gepland, omdat de prioriteit gesteld was op de functionaliteiten van het systeem.

Tijdens de gebruikerstests van de verschillende pilotdelen had ik al vragen gesteld over wat men van de vormgeving vond. Ook gaven de gebruikers zelf spontaan veel commentaar hierop, dat ik vervolg. Van begin af aan had ik de vormgeving van de pilotdelen al centraal geregeld door middel van een stylesheet. Deze stylesheet was echter nog bijna geheel leeg en de classes die er al wel instonden waren niet opgesteld volgens de richtlijnen van de styleguide.

Om deze styleguide te realiseren heb ik de al aanwezige classes aangepast aan de Xtilton huisstijl. Hierdoor kwamen de kleuren van en lettertypen van het geheel overeen met de huisstijl. Met enkele onderdelen van de huisstijl van de internetsite was ik het echter niet eens. Zo waren de hyperlinks als dikgedrukte donkerblauwe tekst weergegeven. Ook de gewone tekst was donkerblauw en af en toe stonden hier ook dikgedrukte woorden tussen. Het zou voor de gebruiker dus onduidelijk zijn wat er nu wel en niet een hyperlink is. Dit wilde ik op het extranet voorkomen. Uiteindelijk heb ik hier dan ook enkele wijzigingen in aangebracht. Ook wat betreft de schermindeling zou afgeweken worden van de internetsite, omdat er op het extranet andersoortige informatie weergegeven werd. Dit vond men overigens geen probleem, want de extranet site hoefde niet identiek te zijn aan de internetsite. Beide hebben immers een andere doelgroep.

Aanvankelijk gebruikten zowel het CMS als het klantdeel dezelfde stylesheet. Later is deze opgesplitst in twee verschillende stylesheets, zodat duidelijker werd welke classes bij welk systeem horen. Hierdoor kunnen beide onafhankelijk van elkaar aangepast worden.

Tevens zijn de voorpagina's en de inlogschermen van het CMS en klantdeel aangepast en is een logo gemaakt. Er werd verwacht dat het logo de naam en de pyramide van Xtilton zou bevatten. Tevens moest het in het bovenste frame van het extranet passen en mocht het niet te groot zijn. Om het geheel bij de Xtilton huisstijl te laten passen, heb ik het daarom zo simpel mogelijk gehouden:

Voor de voorpagina werd een grotere afbeelding gemaakt:

Tijdens het gehele vormgevingsproces heb ik continu feedback gehad van verschillende Xtilton medewerkers, zodat ik kon inspelen op de verschillende wensen die er waren.

6.3.6 Iteratie van de gehele pilot

Nu het pilotdeel vormgeving afgerond was, heb ik de gehele pilot (database, klantdeel, CMS, vormgeving) laten testen door mijn bedrijfsmentor en de opdrachtgever. Hierbij zijn dezelfde testtechnieken gebruikt, maar de testscenario's zijn wat aangepast en toegespitst op het geheel.

Als resultaat van deze gebruikerstest verwachtte ik geen substantiële verbeterpunten te zullen ontdekken, omdat die tijdens de tests van de individuele pilotdelen al naar voren waren gekomen naar voren. Deze verwachting bleek niet te kloppen.

Enkele van het dozijn knelpunten die uit de tests naar voren kwamen waren:

Knelpunten van pilotdeel 2: het klantdeel

- Bij het inloggen worden teveel verschillende pagina's geladen.
- Bij gebruik van de 'terug' knop van de browser treedt een errorpagina op.
- Bij gebruik van de 'refresh' knop van de browser komt de gebruiker in een ander scherm terecht dan waar hij was.
- Er waren Engelse en dubbelzinnige termen aanwezig.
- Het was mogelijk om beveiligde modulen in te zien.

Knelpunten van pilotdeel 3: het CMS

- Het toegankelijk stellen van modulen aan meerdere profielen was omslachtig.
- De IP-detectie zou uitgeschakeld moeten kunnen worden.

Behalve dit werd er door één van de testpersonen ook om een helpfunctie bij gevraagd, of in ieder geval een document waarin beschreven werd wat de mogelijkheden van het systeem waren. Iets dergelijks was ik ook al van plan om te maken.

Aanpassingen van pilotdeel 1: het klantdeel

Aan de hand van de knelpunten die uit de tests zichtbaar werden, zijn een aantal verbeterpunten opgesteld. Om deze verbeterpunten uit te werken was het noodzakelijk het pilotdeel te itereren. De meeste verbeterpunten hadden te maken met de inlogprocedure en de sessions en zouden dus een grote invloed hebben op het gehele pilotdeel.

Beveiligingsproblemen
Eén van de testpersonen, die ervaring had met PHP, probeerde tijdens de test beveiligde informatie te bereiken. Het bleek dat de hyperlink naar beveiligde modulen gewoon in de broncode te vinden was. Ook was het mogelijk om variabelen via een zelfgemaakt formulier naar de loginpagina's te versturen, waardoor ingelogd kon worden onder elk willekeurig profiel, zonder dat hier een wachtwoord voor nodig was. Tijdens de iteratie van dit pilotdeel zijn dergelijke beveiligingsproblemen opgelost.

Hier ziet u bijvoorbeeld een script dat controleert of de bezoeker niet van buitenaf een module probeert te bereiken:

```
<?
//AANROEP CHECK
//controleert of de bezoeker de pagina via het extranet bereikt heeft
//*****************************************************************************
//pagina moet via deze url aangeroepen worden
$vereiste_url = $_SERVER["HTTP_HOST"];
//*****************************************************************************
//haal daadwerkelijk aangeroepen url op
$ref = $_SERVER["HTTP_REFERER"];
$url = explode("?", $ref);
$www = explode("/", $url[0]);
$ontvangen_url = $www[2];
//*****************************************************************************
//vergelijk de urls
if($ontvangen_url != $vereiste_url)
{
echo
"Foutmelding.";
exit;
}
?>
```

PHP script: aanroep check

Dit script moest op elke pagina geplaats worden waar deze beveiliging nodig is. Om dit zo efficiënt mogelijk te doen, is dit via de include() functie gedaan, welke slechts een verwijzing binnen de PHP code geeft naar de locatie van het bestand dat dit script bevat. Op dezelfde manier zijn ook andere beveiligingsproblemen opgelost.

Bruikbaarheidsproblemen

Een probleem dat uit de test naar voren kwam, was dat er bij gebruik van de ‘terug’ knop van de browser een errorpagina te voorschijn kwam. Dit kwam doordat in Internet Explorer een beveiliging ingebouwd zit die voorkomt dat formulieren opnieuw verzonden worden bij gebruik van de ‘terug’ knop. Dit is opgelost door tussen de formulier-pagina’s en de eindpagina’s nog een tussenpagina te zetten en formulierinformatie via sessions naar de volgende pagina over te brengen.

Een ander probleem was dat er bij het herladen van een pagina de gebruiker telkens bij de startpagina van het extranet terecht kwam, in plaats van bij de laatste pagina die hij bekeek. Dit is opgelost door het adres van de laatste pagina in de session op te slaan, en deze in het onderste frame te laten laden.

Ook zaten er enkele Engelse en dubbelzinnige termen in het klantdeel. Als een gebruiker bijvoorbeeld wilde inloggen, dan kon het voorkomen dat hij een profiel zou moeten selecteren. De term ‘profiel’ heeft echter bij de klant Parnassia ook een andere betekenis, namelijk die van het psychiatrisch profiel van een patiënt. De term ‘profiel’ is dan ook vervangen door ‘beveiligingsprofiel’.

Aanpassingen van pilotdeel 2: het CMS

De toegankelijkheid van modulen
Bij de taak van het beschikbaar stellen van modulen aan klanten, bleek tijdens de gebruikerstests dat het ook vanuit een ander perspectief bekeken kon worden. Het was in de huidige situatie namelijk lastig om een bepaalde module weer ontoegankelijk te maken voor alle(of sommige) profielen. Er moest dan immers voor elk profiel bekeken worden of die module wel of niet aangevinkt stond.

Het was daarom gewenst om een bouweenheid te ontwikkelen waarmee het mogelijk was om dit vanuit het perspectief van de module te bekijken, in plaats van per profiel:

In bovenstaande afbeelding ziet u de uitwerking van deze bouweenheid. Hiermee kan de beheerder per module bekijken welk profiel toegang heeft, en hier wijzigingen in aanbrengen. Als het nodig is om een profiel voor alle klanten toegankelijk of ontoegankelijk te maken, dan kunnen simpelweg alle checkboxes aan- of afgevinkt worden.

IP-detectie
Ook andere nieuwe bouweenheden zijn bij de iteratie ontwikkeld, zoals de mogelijkheid om de IP-detectie per klant aan of uit te zetten. Dit bleek noodzakelijk, omdat bijvoorbeeld bij de klant OBL (Openbare Bibliotheek Leiden) het kon voorkomen mensen vanuit bibliotheekcomputers het extranet zouden binnenkomen. Door het mogelijk maken IP-detectie per klant aan of uit te zetten, kon de beveiliging van het systeem beter beheerd worden.

IP-detectie aan/uit zetten

Klant	IP-detectie
Parnassia	Aan Uit
De Jutters	Aan Uit
OBL	Aan Uit
Xtilton	Aan Uit
Shell	Aan Uit

Inloggen onder een profiel van een klant

Als beheerder inloggen in het klantdeel
Een andere bouweenheid bleek toen tijdens de gebruikerstest een gebruiker wilde inzien welke resultaten zijn acties in het CMS op het klantdeel zouden hebben. Hiervoor moest hij telkens tussen het CMS en het extranet wisselen. Om dit te vergemakkelijken is een inlog-knop bij elk beveiligingsprofiel geplaatst, waarmee de CMS beheerder kan inloggen onder elk profiel, zodat hij kan zien wat de klant ook ziet. Om dit te realiseren moesten er ook enkele wijzigingen in de inlogprocedure van het klantdeel aangebracht worden.

Beheerdersmodule voor de klant
Tijdens de iteratie van de pilot is ook een beheerdersmodule voor de klanten gebouwd, waarmee ze de wachtwoorden van de verschillende profielen kunnen inzien en aanpassen. De noodzaak hiertoe bleek tijdens een gesprek met mijn bedrijfsmentor. Ik vroeg hem met wie een extranetbezoeker contact dient om te nemen om te weten te komen bij welk profiel hij hoort en welk wachtwoord hij dus nodig heeft. Er waren hiervoor twee plekken waar hij terecht zou kunnen:

- Bij Xtilton
- Bij iemand van de klant zelf

Voor Xtilton was het natuurlijk het meest gunstig als de klant zelf kon bijhouden welke mensen bij welk profiel horen en wat de wachtwoorden zijn. De klant zat er echter niet op te wachten om iemand te belasten met deze nieuwe taken. Xilton zou dit dan ook niet als vereiste kunnen stellen voor het gebruik van het extranet, maar wel als extra optie voor de klant. Er is toen besloten een bescheiden module te bouwen, waarmee klanten de wachtwoorden van de verschillende profielen kunnen inzien en aanpassen.

De module 'technisch beheer'

In bovenstaande afbeelding is rechts nog een knop 'toegankelijkheden' zichtbaar. Hiermee kon de technisch beheerder instellen waar een profiel toegang toe heeft. Deze optie is later weer verwijderd, omdat dit eigenlijk alleen door Xtilton bepaald mag worden.

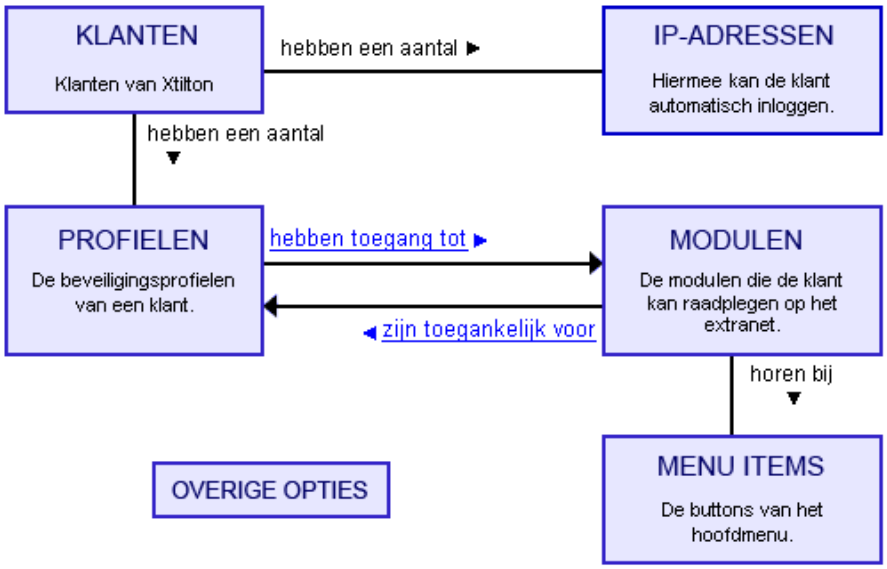
Deze module 'technisch beheer' is vervolgens ook als module aan het extranet toegevoegd(dit was inmiddels mogelijk dankzij het CMS), waardoor het via een menu-item in het extranet toegankelijk zou zijn voor het profiel 'technisch beheer' van klanten.

Gebruikshandleiding
Wanneer een gebruiker het CMS binnenkwam, bleek het niet ogenblikkelijk duidelijk te zijn wat de mogelijkheden van het CMS waren en welke invloed het CMS had op het extranet. Er werd bijvoorbeeld nergens uitgelegd dat als men een standaardprofiel aan een klant toekent, dat klanten hier dan automatisch onder ingelogd werden. Dit proces werd alleen in de

systeemdocumentatie beschreven. Er kon echter niet van CMS gebruikers verwacht worden dat zij de systeemdocumentatie bij de hand moesten hebben om het CMS te kunnen gebruiken.

Om de structuur van het extranet duidelijk te maken en aan te geven welke gevolgen acties in het CMS voor het extranet hadden, heb ik een korte gebruikershandleiding geschreven. Hierin werd per object van de database aangegeven wat de rol ervan binnen het geheel was. Op deze manier zou men ook direct inzien wat voor consequenties een bewerking van een object in het CMS op het gehele systeem zou hebben. Deze handleiding is als een soort helpfunctie binnen het CMS bereikbaar gemaakt.

Tevens is er een voorpagina gemaakt die de gebruiker ziet zodra hij het CMS binnenkomt.



Voorpagina van het CMS

Op deze voorpagina is schematisch weergegeven wat in woorden in de gebruikershandleiding is te lezen. Op deze manier zou een gebruiker bij binnenkomst ogenblikkelijk de structuur van het CMS inzien. Door op één van de onderdelen van de afbeelding te klikken kon de gebruiker tevens direct naar het betreffende deel van het CMS gaan waar hij bewerkingen erop kan uitvoeren.

Afronden van de pilot

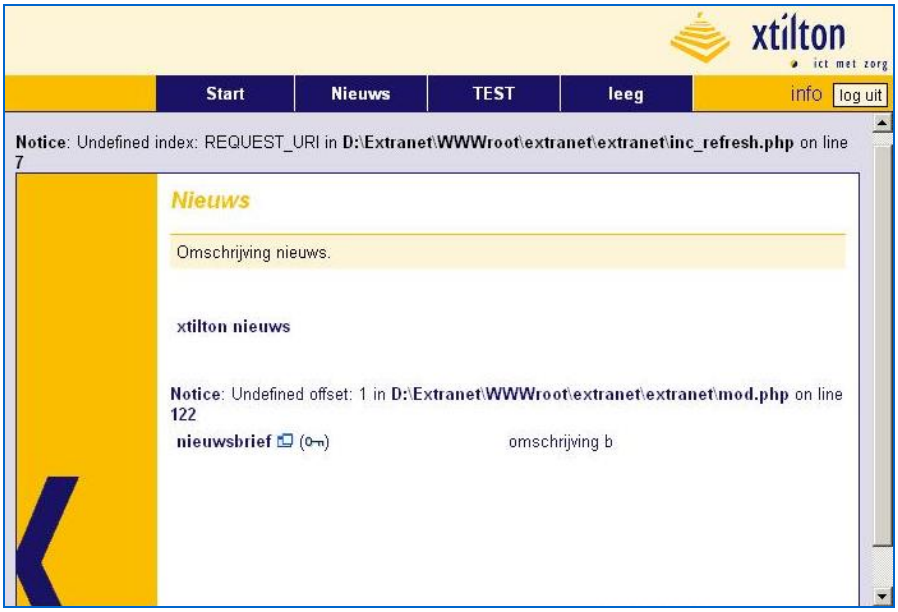
Aan het eind van deze iteratie van de gehele pilot, is de pilot nogmaals getest. Als resultaat van deze gebruikerstest kwamen slechts kleine verbeterpunten naar voren die betrekking hadden op de vormgeving en slordigheidsfoutjes. Na deze uitgewerkt te hebben besloot ik de pilot op de server te installeren.

6.4 Installatie op de server

Bij aanvang van het project was besloten het extranet op dezelfde server te zetten als die waar het Xtilton intranet ook op draaide. Na het voltooien van de eerste pilot, de basisstructuur van het extranet, heb ik opnieuw gesproken met een collega over de installatie van het extranet op de server. Het bleek dat dit hier toch een andere server voor gebruikt zou worden. Op deze server werd door hem IIS6.0, PHP en MySQL 5.0-beta geïnstalleerd.

Voor de exacte configuratie van PHP op de server heb ik samen met hem de documenten doorgenomen die ik tijdens de installatie van het extranet in de lokale omgeving had opgesteld. In deze documenten had ik iedere stap van de installatieprocedure in de testomgeving volledig beschreven. Hierdoor was het mogelijk deze procedure met enkele kleine wijzigingen(IIS in plaats van Apache) te hanteren bij de installatie van het extranet op de uiteindelijke server.

Na het voltooien van deze installatie, ben ik de verschillende onderdelen van de basisstructuur gaan uitproberen om te zien of alles werkte onder IIS. Hierbij kwamen er een aantal problemen aan het licht. Sommige PHP functies die gebruikt werden, bleken niet te werken onder IIS, waardoor er behoorlijk wat foutmeldingen tevoorschijn kwamen.



Foutmelding die zichtbaar werd na installatie op de IIS server

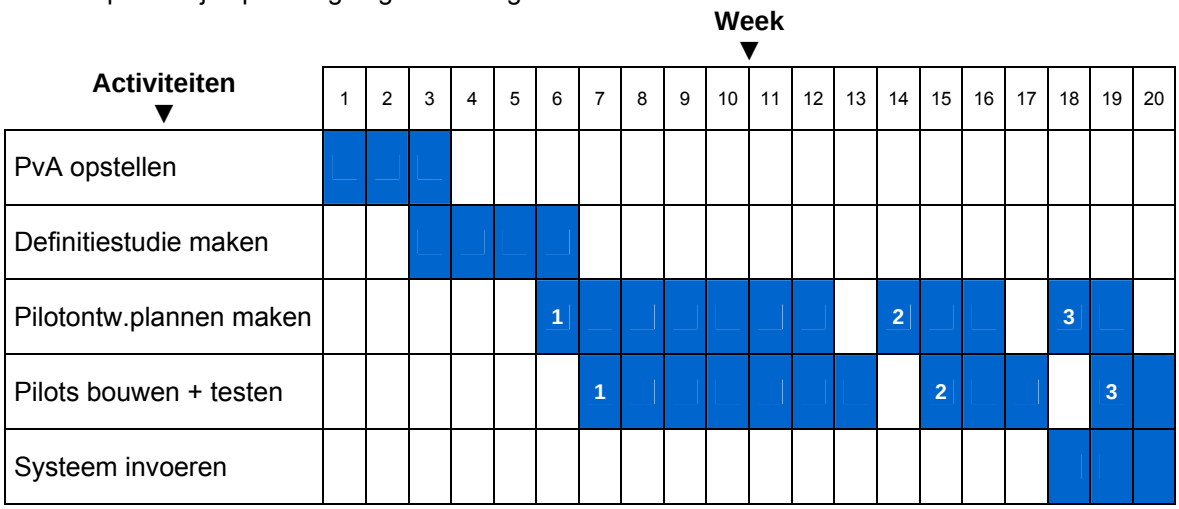
De bovenste foutmelding in bovenstaand scherm bijvoorbeeld werd veroorzaakt door de PHP functie REQUEST_URI, welke het adres van de huidige pagina opvraagt. Deze functie werd gebruikt om ervoor te zorgen dat bij het herladen van de pagina deze waarde in het onderste frame getoond wordt. IIS bleek deze functie niet te ondersteunen.

Uiteindelijk bleek het er erger uit te zien dan het in werkelijkheid was. Door het raadplegen van de PHP documentatie en door de verschillende foutmeldingen op te zoeken op het internet, kon ik de problemen binnen enkele dagen oplossen. Deze oplossingen heb ik vervolgens ook op de lokale Apache server getest, zodat ik er zeker van zou zijn dat de PHP code onder beide servers zou werken.

6.5 Aanpassing van de planning

Het project bevond zich inmiddels in de vijftiende week en liep dus achter op de planning. Er was echter van begin af aan al aangegeven dat de overige twee pilots verschoven konden worden naar een latere datum.

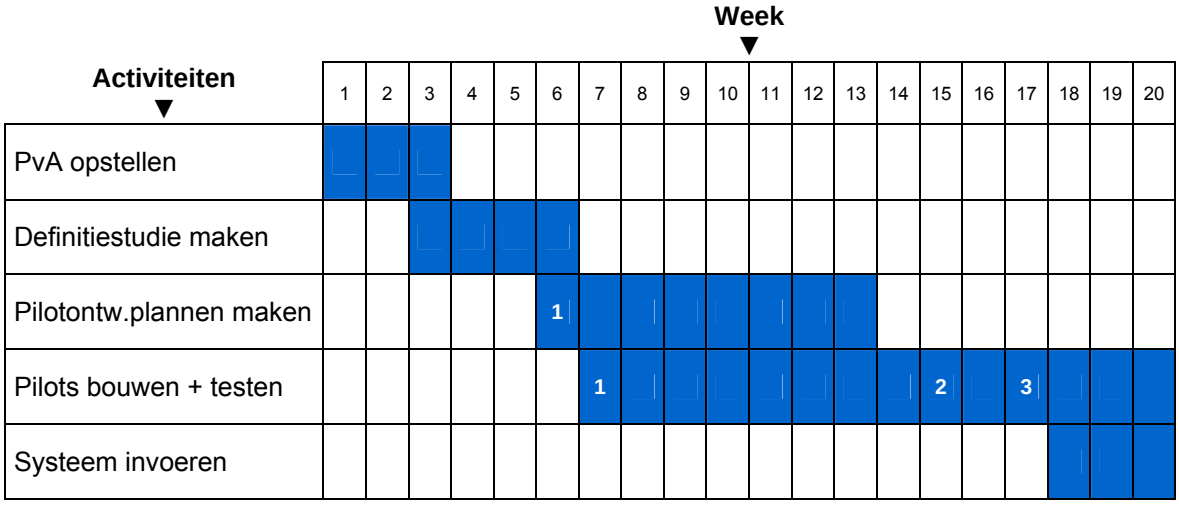
De oorspronkelijke planning zag er als volgt uit:



Oorspronkelijke planning

Omdat er nog maar enkele weken overbleven, besloot ik geen volledige pilotontwikkelplannen hiervoor te produceren, maar slechts documenten die echt noodzakelijk waren voor het product zelf. Hiermee hoopte ik tijd te besparen, waardoor de pilots toch op tijd voltooid zouden kunnen worden. Tevens besloot ik de invoering wat te vervroegen, zodat dit meteen na het voltooien van pilot twee, het incidentensysteem gedaan kon worden.

De gewijzigde planning kwam er toen als volgt uit te zien:



Gewijzigde planning

6.6 Pilot 2: het incidentensysteem

Nu de basisstructuur afgerond was, kon begonnen worden met de volgende pilot. Deze pilot zou, evenals pilot 3, als module binnen de basisstructuur aangeboden worden aan klanten. Met deze pilot kunnen extranetbezoekers computerproblemen vragen of melden aan de Xtilton servicedesk.

6.6.1 Het opsplitsen van de pilot in pilotdelen

Bij aanvang van de ontwikkeling van de pilot is deze opgedeeld in verschillende pilotdelen. De pilotdelen zijn:

- **Vormgeving**
  Het aanpassen van de vormgeving van het huidige incidentensysteem, zodat het binnen het extranet past.

- **Helpfunctie**
  Het maken van een helpfunctie waarin wordt uitgelegd hoe het incidentensysteem werkt.

De helpfunctie zou als laatst ontwikkeld worden, omdat hierbij schermafbeeldingen van het systeem gebruikt dienden te worden. Het was dus noodzakelijk eerst de vormgeving te voltooien zodat deze schermafbeeldingen gemaakt konden worden.

Voordat begonnen is met het ontwikkelen van de pilotdelen, heb ik echter eerst met een collega gesproken over hoe het gebruik van dit systeem vergroot zou kunnen worden. Het systeem bestond immers al, maar werd nog maar weinig gebruikt. Uit dit gesprek kwamen enkele ideeën naar voren:

- Bij het opnemen van de telefoon, zouden servicedeskmedewerkers de opbeller kunnen melden dat vragen/problemen ook via het extranet gemeld kunnen worden.
- Door het opstellen van een lijst met voordelen van het systeem op het extranet(pilot 2) en deze te emailen naar de klanten, konden ze gemotiveerd worden dit te gaan gebruiken.

6.6.2 Pilotdeel 1: de vormgeving

Het was reeds bekend dat er al een systeem (Topdesk Application Server, oftewel TAS) aanwezig was waarmee personeel van klanten problemen en vragen aan de Xtilton servicedesk kon melden. Het was de wens van de opdrachtgever dat dit systeem binnen het extranet kwam te draaien. Hiertoe was het noodzakelijk om de vormgeving van dit systeem aan te passen aan de vormgeving van het extranet.

Informatie verzamelen bij de leverancier

Omdat dit systeem door een extern bedrijf was ontwikkeld, OGD software, wist ik niet in hoeverre ik dit zou kunnen realiseren. Ik heb eerst de website van Topdesk geraadpleegd, waarop ik las dat het uiterlijk van het systeem flexibel was en aangepast kon worden aan de huisstijl. Om hier meer over te weten te komen heb ik OGD software opgebeld met de vraag hoe dit aanpassen van de huisstijl precies in zijn werk gaat.

Het bleek dat er twee mogelijkheden waren om de vormgeving aan te passen:

- Het zelf doen
- OGD software het laten doen

Het was mogelijk om de vormgeving zelf aan te passen door de stylesheet, .jsp (Java Server Pages) pagina's en afbeeldingen aan te passen. Hief kleefde echter een nadeel aan. Als OGD software namelijk een update zou uitbrengen, dan was het goed mogelijk dat de volledige stylesheet en overige bestanden aangepast werden. Een gevolg hiervan kon zijn dat systeem er niet alleen als een rotzooi uit zou komen te zien, maar dat het zelfs niet meer zou werken.

Het was ook mogelijk om de vormgeving door OGD software zelf te laten aanpassen aan de Xtilton huisstijl. Bij een update van Topdesk Application Server zou door hen dan ook telkens de vormgeving bijgehouden worden. Aan deze optie kleefde echter een prijskaartje.

Uiteindelijk is in overleg met mijn bedrijfsmentor gekozen voor de eerste mogelijkheid, het zelf aanpassen van de vormgeving. Het zou dan wel noodzakelijk zijn om deze vormgeving goed te documenteren, zodat precies duidelijk zou zijn waar wijzigingen aangebracht waren en hoe deze na eventuele OGD updates hersteld konden worden.

Het analyseren van de User Interface

Na deze beslissing heb ik eerst de verschillende onderdelen binnen TAS bekeken, zoals het menu en de verschillende pagina's. Ik had dit systeem nog nooit eerder gezien en kon me dus goed verplaatsen in andere gebruikers die het voor het eerst te zien zouden krijgen. Het viel mij direct op dat er bij het ontwikkelen van dit systeem erg weinig aandacht besteedt was aan de gebruikerskant ervan. Van tevoren wist ik wel dat dit systeem als doel had het melden en inzien van incidenten, maar toen ik het systeem eenmaal met eigen ogen zag, was ik hier opeens niet zo zeker meer van.

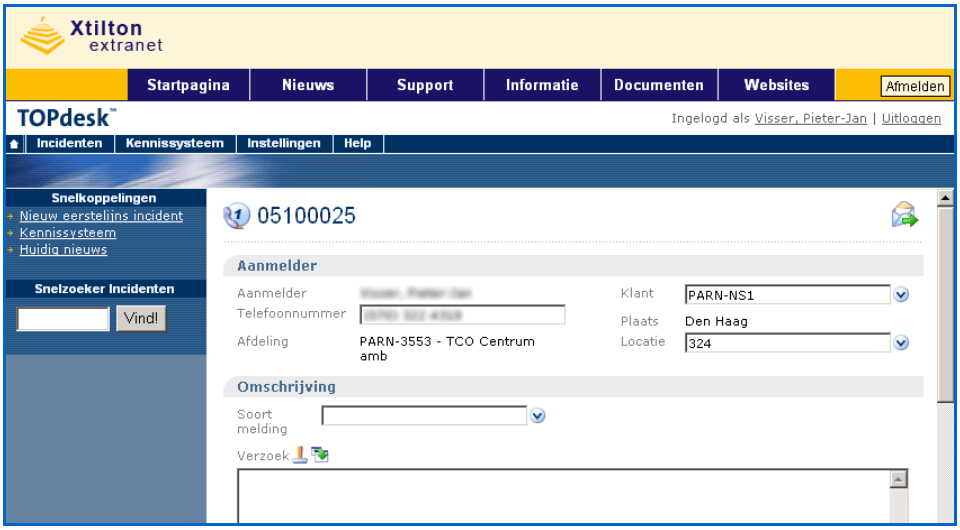
Topdesk Application Server

Een voorbeeld: als je een probleem of vraag wilde melden, dan was er in het menu een hyperlink beschikbaar genaamd 'nieuw 1^{e} lijns incident'. Ten eerste weet een gebruiker niet eens wat een incident is en ten tweede weet hij ook niet wat de 1^{e} lijn is. Al deze termen zijn van betekenis binnen de servicedesk en Xtilton zelf waar men de ITIL processen kent, maar niet voor een gebruiker bij de klant die niets van ITIL afweet.

Ik heb op deze manier elke pagina doorgenomen en genoteerd welke onderdelen ik niet begreep, met als resultaat een behoorlijk lange lijst met knelpunten. Tijdens het aanpassen van de vormgeving zou ik trachten zoveel mogelijk punten op die lijst weg te werken. Ik realiseerde me echter dat het voor mij niet mogelijk was de daadwerkelijke broncode van de pagina's aan te passen, omdat alleen OGD software hier veranderingen in kon aanbrengen. De punten die ik dus niet zou kunnen wegwerken tijdens het aanpassen van de vormgeving, zou ik dan in de helpfunctie aan bod laten komen.

Het vervangen van het menu

Bij elk onderdeel heb ik tevens de broncode doorgenomen om beeld te krijgen van hoe het systeem in elkaar zit. Hierbij kwam ik erachter dat het systeem gebruik maakt van HTML frames. Elke hyperlink in het menu verwees naar een .jsp pagina die in het andere frame geladen zou worden.



TAS binnen het extranet – oude versie

Het feit dat dit TAS gebruik maakte van frames bood mij de mogelijkheid het menu ervan te vervangen door een zelfgemaakt menu dat binnen het extranet paste. Het eerder genoemde voorbeeld over de hyperlink genaamd 'nieuw 1e lijns incident' zou ik hiermee dus kunnen wegwerken, evenals alle andere punten van de lijst die betrekking hadden op het menu. Bovenin TAS was tevens een horizontale menubalk te zien die niet binnen het extranet paste, omdat het extranet zelf ook al een horizontale menubalk had. Dit zou verwarring kunnen veroorzaken bij de gebruiker. Bovendien waren sommige van de hyperlinks overbodig. Het 'huidig nieuws' bevatte bijvoorbeeld totaal geen informatie.

ITIL termen vervangen

In overleg met de TAS kenners binnen Xtilton en het hoofd van de servicedesk heb ik de relevante delen van het systeem voor de eindgebruikers uitgezocht en deze in een nieuw menu geplaatst. De relevante delen waren:

- Het melden van een vraag/probleem
- Het inzien van reeds gemelde vragen/problemen
- Het wijzigen van het wachtwoord (dit is niet het extranet wachtwoord, maar het TAS wachtwoord)

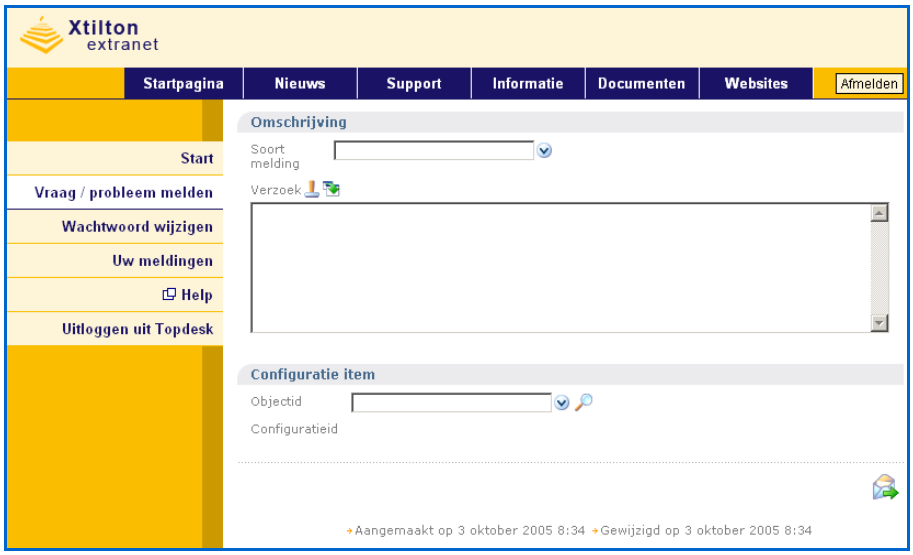
De overige opties binnen TAS, zoals enkele tussenpagina's, het kennissysteem, het huidige nieuws en een zoekoptie, bleken niet relevant te zijn voor de eindgebruiker, omdat ze vrijwel geheel leeg waren.

Bij het uitzoeken van de relevante delen is tevens gediscussieerd over de te gebruiken termen. Zo wilde de één dat er ITIL termen gebruikt werden, terwijl ik liever termen wilde gebruiken die voor gewone gebruikers te begrijpen zijn. De term 'probleem' betekent binnen ITIL bijvoorbeeld wat anders dan wat de eindgebruiker hieronder verstaat. Als een gebruiker bijvoorbeeld een foutmelding in MS Excel krijgt die hij niet kan oplossen, dan denkt hij dat hij een 'probleem' heeft en meldt dit als zodanig aan de servicedesk.

Bij ITIL worden dergelijke ‘problemen’ echter als ‘incidenten’ benoemd. Het wordt pas een probleem genoemd als er een structurele fout achter zit. Als er een structurele fout bestaat, dan zal dit resulteren in aantal incidenten.

Uiteindelijk wordt deze structurele fout dan als een probleem benoemd en valt het onder het probleem management. Hier wordt het probleem een bekende fout, wat betekent dat de oorzaak ervan bekend is en er een tijdelijke oplossing, een workaround, voor is bedacht. Vervolgens wordt er een Request For Change procedure uitgevoerd zodat de oorzaak van het probleem kan worden weggewerkt.

Dit gehele achterliggende ITIL proces, inclusief alle gebruikte termen, is voor de gebruiker echter onbekend. De gebruiker confronteren met dergelijke termen zou dus alleen maar verwarring zaaien. Een eventueel voordeel voor de servicedesk zou kunnen zijn dat gebruikers die een probleem/incident melden via het extranet de juiste termen hanteren in hun omschrijving van het probleem/incident. Er kon echter niet van ze verwacht worden dat ze bekend waren met ITIL. Er is dus besloten om geen ITIL termen te hanteren bij de interface voor de eindgebruiker.



TAS binnen het extranet – gewijzigde versie

Uiteindelijk is het menu van TAS zodanig aangepast dat het binnen de Xtilton huisstijl paste en de gebruikte termen begrijpelijk zouden zijn voor de eindgebruiker. Helaas waren er op de verschillende pagina’s van TAS nog een groot aantal punten die voor onduidelijk waren, maar die ik niet kon wijzigen. Door deze punten in de helpfunctie aan te kaarten zou het voor de eindgebruiker toch mogelijk zijn er duidelijkheid over te krijgen.

Tot slot is er een introductiepagina gemaakt waarop wordt uitgelegd wat TAS is, wat je ermee kan doen, en wat er gebeurt nadat je hiermee een probleem gemeld hebt. Hierdoor zou de gebruiker een beter overzicht van de gehele meldingsprocedure krijgen, waardoor de drempel tot gebruik van het systeem lager zou worden. Na al deze aanpassingen bleek het systeem qua vormgeving al goed binnen het extranet te passen. Er is toen besloten dat de stylesheet van TAS niet meer aangepast zou worden, zodat dit in de toekomst ook niet meer bijgehouden hoefde te worden door een Xtilton medewerker.

6.6.3 Pilotdeel 2: de helpfunctie

Tijdens het uitwerken van het eerste pilotdeel ben ik op talloze onduidelijkheden in de TAS-interface gestuit. Deze onduidelijkheden had ik reeds genoteerd en sommigen ervan waren door het aanpassen van het menu, het maken van de introductiepagina en het weglaten van andere onderdelen, al verbeterd. Andere onduidelijkheden konden alleen via een helpfunctie aangekaart worden.

Binnen TAS was al een knop aanwezig met het woordje 'help' erop, maar als hierop geklikt wordt, krijg je geen helpfunctie te zien. In plaats daarvan verschijnt het scherm dat u rechts ziet.

Als helpfunctie heeft een dergelijk scherm natuurlijk weinig nut. Iemand die een computerprobleem wil melden en niet begrijpt hoe dit moet, zal hoogstwaarschijnlijk geen brief gaan schrijven aan de producent van het softwarepakket dat dit mogelijk moet maken.

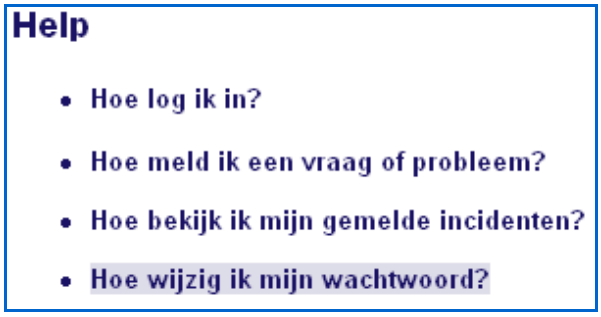
Het was dus de bedoeling dat dit scherm vervangen zou worden door een geschiktere helpfunctie.

De helpfunctie van TAS

In het eerste pilotdeel waren al de relevante delen van het systeem geselecteerd. Voor de gebruikers kwamen deze delen op drie verschillende taken neer:

- het melden van een probleem
- het bekijken van een gemeld probleem
- het wijzigen van het wachtwoord.

Hier kwam nog een vierde taak bij, namelijk het inloggen op TAS, dat noodzakelijk was om één van bovenstaande taken te kunnen uitvoeren. Deze vier taken zijn als aparte secties in de helpfunctie uitgewerkt:



Elk van deze taken heb ik enkele malen doorlopen, waarbij ik van elk scherm genoteerd heb waar onduidelijkheden zaten en welke stappen er precies uitgevoerd dienden te worden om het uiteindelijke resultaat te bereiken.

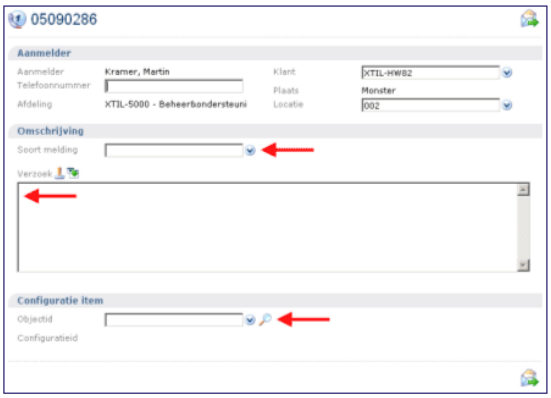
Het melden van een probleem bestond bijvoorbeeld uit drie stappen:

- Stap 1: het invullen van het probleem
- Stap 2: het bevestigen van de gegevens
- Stap 3: het afsluiten van TAS

Deze stappen heb ik in nog gedetailleerdere acties opgedeeld. Per stap heb ik een screenshot van het systeem gemaakt en hierop aangegeven waar de gebruiker welke actie dient uit te voeren.

Stap 1: invullen

Klik links in het menu op **<Vraag/probleem melden>**.
U ziet nu dit scherm:



U hoeft slechts drie velden in te vullen (zie rode pijltjes):

- Bij **Soort melding** selecteert u met het pijltje de aard van uw melding.
- Bij **Verzoek** typt u uw vraag of probleem.
- Bij **Objectid** typt u de barcode van uw PC, of het nummer van uw PRINTER *(deze staat op de zijkant van uw computer/printer en begint met 'PARN-').*

Klik vervolgens op

Een scherm van de nieuwe TAS-helpfunctie

Tot slot heb ik de navigatie tussen de verschillende help-onderdelen uitgewerkt.

Door de helpfunctie zorgvuldig uit te werken, hoopte ik dat bij de test van het systeem door de eindgebruikers zo min mogelijk verbeterpunten naar voren zouden komen.

6.7 Het testen van het systeem door de klant

Het project was inmiddels in week 16 aangekomen. Omdat de basisstructuur nu voltooid was en er ook een werkende module binnen deze basisstructuur draaide, kon het systeem door de gebruikers aan de klantzijde getest worden. Op advies van een collega heb ik contact opgenomen met de Service Level Manager van Parnassia. Hij gaf er de voorkeur aan om zelf samen met een andere collega, een internet/intranetspecialist, het extranet te bekijken en vervolgens feedback hierover te geven.

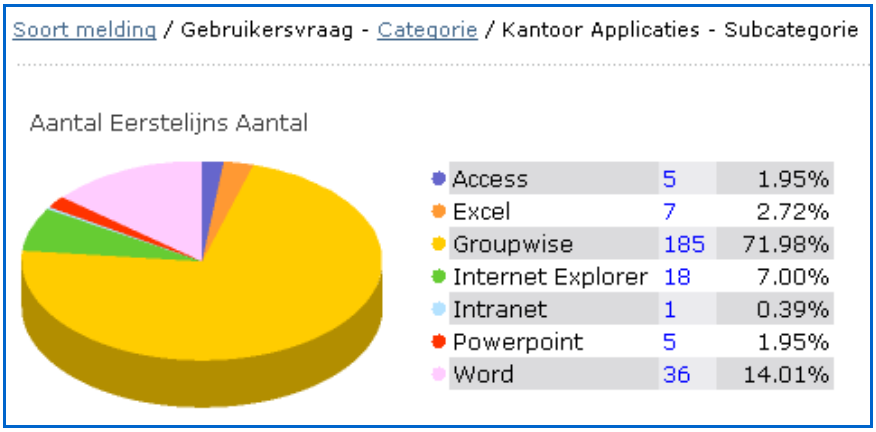
Hoewel ik er zelf de voorkeur aan gaf ook aanwezig te zijn bij de test, bleek dit pas op een latere datum mogelijk te zijn. Om het project niet te vertragen besloot ik dus met deze testopzet in te stemmen. Aan de ene kan zou ik hierdoor misschien wat spontane reacties mislopen, aan de andere kant bood dit beide testpersonen wel de mogelijkheid om eerst uitgebreid en zorgvuldige over hun feedback na te denken.

Om feedback op de juiste onderdelen van het systeem te krijgen, heb ik een formulier met enkele scenario’s en een formulier met vragen opgesteld. Deze scenario’s waren vrij simpel en gingen bijvoorbeeld over het inloggen, uitloggen en het melden van een probleem. Ik heb hierbij expres enkele tussenstappen weggelaten, zodat de gebruiker zelf een beetje moet zoeken en uitproberen wat de bedoeling is. In het vragenformulier werd er vervolgens gevraagd of men op een bepaald moment niet meer wist wat te doen. Ook over de vormgeving zijn vragen opgesteld, evenals vragen over de vergelijking tussen het oude systeem (TAS) en de nieuwe versie op het extranet.

Via de email heb ik deze formulieren tezamen met de internetadres van het extranet naar de Service Level Manager en de internet/intranetspecialist toegestuurd. Er werd afgesproken dat ze binnen een week de feedback zouden terugsturen. Enkele dagen later ontving ik deze formulieren met feedback terug. Hieruit kwamen slechts enkele verbeterpunten naar voren die betrekking hadden op de vormgeving. Hoewel geen van deze punten belangrijk was, heb ik ze toch uitgevoerd, omdat dit maar weinig tijd kostte.

6.8 Pilot 3: het veelgestelde vragen systeem

Na het voltooien van de tweede pilot, was er nog maar weinig tijd beschikbaar voor het ontwikkelen van de derde pilot. Terwijl ik wachtte op de resultaten van de test bij Parnassia, ben ik bij een collega gaan informeren of er ergens een lijst met meestgestelde vragen aan de servicedesk te vinden was. Dit bleek helaas niet het geval te zijn. Om erachter te komen welke vragen klanten zoal aan de servicedesk stellen en in welke categorieën deze onder te delen zijn, ben ik in het incidentensysteem gaan kijken. Hier kun je als een 'behandelaar' op inloggen, waarna je alle incidenten die ooit gemeld zijn kunt inzien, sorteren, weergeven in grafieken, etc. Ik heb alle incidenten van de laatste twee maanden in een rapport samengevoegd en deze gesorteerd op soort melding en soort software. Vervolgens heb ik deze in een taartdiagram weergegeven, zodat meteen duidelijk werd welk soort incidenten het meest binnenkwamen.



De verschillende soorten incidenten die gemeld zijn

Per applicatie kunnen vervolgens de individuele meldingen bekeken worden, evenals de oplossingen ervan. Het paste echter niet meer binnen het tijdsbestek van de afstudeeropdracht om deze allemaal door te nemen en de meestvoorkomende vragen te selecteren.

Eerste opzet database

Door de verschillende soorten gebruikersvragen te bekijken, was het echter wel mogelijk al een eerste opzet voor de database te op te stellen. Uit het systeem bleken er zowel categorieën als subcategorieën te bestaan.

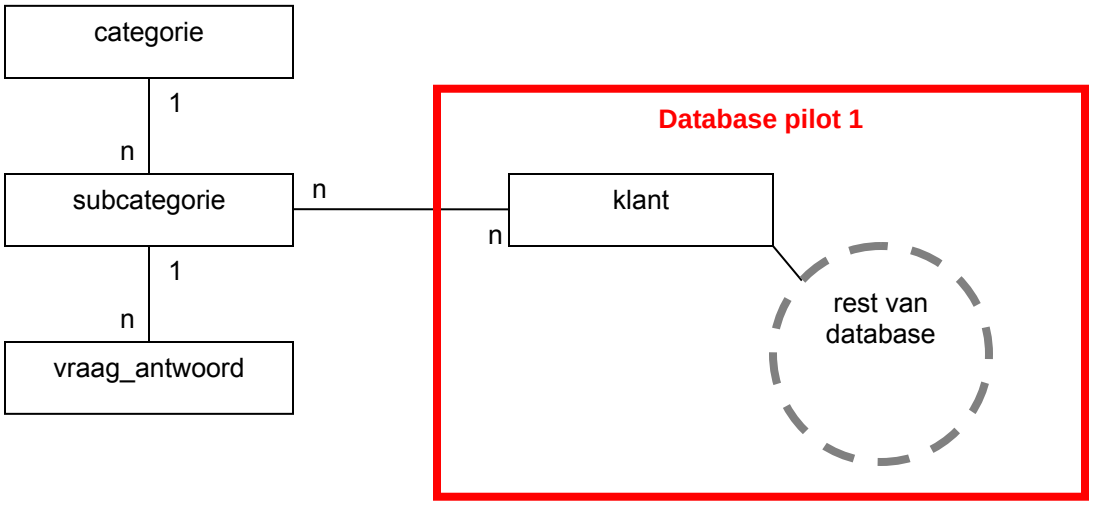
Voorbeelden van categorieën zijn:

- Kantoor applicaties
- Netwerk Hardware
- Telefonie

Enkele subcategorieën van bijvoorbeeld 'kantoor applicaties' zijn:

- Access
- Excel
- Groupwise

In elke subcategorie waren vervolgens vragen en bijbehorende gehanteerde oplossingen te vinden. Uiteindelijk is er van deze situatie een eerste opzet voor een database gemaakt.



Eerste opzet database pilot 3: het veelgestelde-vragen systeem

Door deze opzet al enigszins uit te werken, wilde ik te weten komen of er daadwerkelijk een database nodig was om het veelgestelde-vragen systeem te realiseren. Uit de definitiestudie was al gebleken dat er per klant verschillende soorten software waren, en dus ook verschillende soorten vragen gesteld werden. Een klant die bijvoorbeeld geen gebruik maakt van Groupwise, hoeft niet de veelgestelde vragen en oplossingen hierover in te zien. Hierdoor was het dus noodzakelijk om de database van het veelgestelde-vragen systeem te koppelen aan de database van pilot 1. Door de simpele opzet van de database van pilot 1 zou dit te realiseren zijn zonder de structuur ervan aan te passen.

Als de vragen en oplossingen in de database kwamen te staan, dan moest er ook een Content Management Systeem gemaakt worden om deze gegevens te kunnen bewerken. Een dergelijk systeem zou echter niet meer binnen de afstudeerperiode gerealiseerd kunnen worden, omdat het project zich inmiddels in de zeventiende week bevond. Ik besloot daarom nog eens met een collega te spreken over mogelijke andere manieren om dit uit te werken. Hieruit kwam naar voren dat, omdat er geen beveiligde informatie in dit systeem staat, het eigenlijk niet erg is als klanten ook de andere categorieën kunnen inzien. Bovendien was het mogelijk dit hele systeem met statische pagina's in elkaar te zetten, door gewoon per vraag een HTML pagina te maken met daarop de oplossing. Uit dit gesprek bleek de voorkeur uitging naar de statische HTML uitwerking. Omdat er binnen Xtilton mensen aanwezig waren met kennis van HTML of HTML-bewerkingsprogramma's zou deze optie uiteindelijk gemakkelijker beheerd kunnen worden. Ook bood deze optie meer vrijheid in het gebruik van afbeeldingen en animaties.

Uiteindelijk is deze pilot niet meer gerealiseerd, omdat het project ten einde liep.

7. De fase invoering

De invoering van het extranet heeft plaatsgevonden in de laatste twee weken van de afstudeerperiode. Tijdens de invoering zijn de volgende activiteiten uitgevoerd:

- Het verwijderen van overbodige bestanden en documenten
- Het opschonen van de broncode
- Het opstellen van een beknopte gebruikershandleiding
- Het overhandigen van alle systeemdocumentatie
- Het digitaal opleveren van alle systeembestanden

Het extranet zelf stond inmiddels reeds op de juiste server geïnstalleerd. Deze server was via een IP-adres te bereiken door klanten. Voordat het daadwerkelijk bij de klanten aangeboden zou gaan worden, zouden er eerst nieuwe modulen ontwikkeld en binnen het extranet geplaatst moeten worden. Als het extranet namelijk op al op dit moment door klanten gebruikt zou worden, dan troffen ze hier alleen nog maar het incidentensysteem aan. De kans was dan groot dat ze nooit meer terug zouden komen, omdat ze denken dat er toch vrijwel niets op staat.

Het CMS was voor iedereen binnen Xtilton bereikbaar en dankzij de gebruikershandleiding was het duidelijk hoe men het extranet kon uitbreiden. Binnen Xtilton was het systeem dus volledig ingevoerd.

8. Evaluatie

In dit hoofdstuk worden de opgeleverde producten en het doorlopen proces geëvalueerd.

8.1 Productevaluatie

Plan van Aanpak

Bij aanvang van de stage is, na het onderzoek naar de scope van de opdracht, een Plan van Aanpak opgesteld. Het doel hiervan was een overzicht van de planning en uit te voeren activiteiten van de projectperiode te verkrijgen. Aanvankelijk werden hierbij beslissingen genomen die nog helemaal niet genomen konden worden, zoals het bepalen van de benodigde ontwikkeltijd per pilot en de exact uit te voeren activiteiten. Tijdens de definitiestudie is de omvang van de pilots duidelijker geworden, waarna de planning in het Plan van Aanpak is bijgesteld. De kwaliteit van het document was goed, want het is als leidraad voor het verdere project gebruikt. Enkele beschreven activiteiten uit het Plan van Aanpak zijn uiteindelijk niet nodig gebleken, maar globaal gezien bood het me een goed overzicht van het gehele project.

Definitiestudie

Omdat de fase definitiestudie eenmalig doorlopen zou worden, is hier enkele weken de tijd voor genomen. Deze fase is grondig uitgevoerd en het definitiestudierapport bevatte dan ook de informatie die nodig was om de fase pilotontwikkeling te kunnen beginnen. De gebruikers- en taakanalyse maakte het mij mogelijk de verschillende functionaliteiten van het systeem in kaart te brengen en de styleguide is tijdens het uitwerken van de vormgeving handig gebleken. Het systeemconcept heeft als opzet voor de verdere pilotontwikkeling gediend. Een minpunt van de definitiestudie was dat er misschien teveel de nadruk is gelegd op de taken van de gebruikersgroep klanten, terwijl het grootste deel van de opdracht uit het ontwikkelen van een systeem voor de Xtilton medewerkers (het CMS) zou bestaan.

Pilotontwikkelplan

Er is uiteindelijk maar één pilotontwikkelplan opgesteld. Deze beschreef de grootste pilot: de basisstructuur. In dit pilotontwikkelplan zijn de verschillende aspecten van de basisstructuur uitgebreid beschreven en ik heb het dan ook gebruikt bij de realisatie van de verschillende pilotdelen. Voor Xtilton zal dit document ook geschikt zijn om informatie over de basisstructuur in terug te vinden.

Extranet

Dit is het deel van het extranet dat door de klanten gebruikt gaat worden om de verschillende soorten informatie en diensten van Xtilton te raadplegen. Over de opzet van dit systeem ben ik tevreden. Het werkt goed, ziet er duidelijk uit en is toegankelijk en transparant voor de doelgroep. Met name over het automatiseren van het inloggen en het fraudebestendig maken hiervan ben ik tevreden. Waar ik minder tevreden over ben, is dat er nog maar weinig modulen daadwerkelijk op het extranet draaien. Dit was echter ook niet de bedoeling van de afstudeeropdracht, want de nadruk lag op de uitbreidbaarheid van het systeem. Dit is dus dit eigenlijk geen minpunt van het door mij opgeleverde product. Wel vind ik het jammer dat het veelgestelde-vragen systeem in de ontwikkelfase is blijven steken en niet afgerond is. Uiteindelijk zal een bezoeker die op dit moment het extranet binnenkomt hierop dus nog maar weinig aantreffen.

Content Management Systeem

Hiermee kan Xtilton personeel het extranet uitbreiden en beheren. Tijdens het project is hier de meeste tijd aan besteed en dit heeft een degelijk systeem opgeleverd. De grootste pluspunten van dit systeem zijn dat het erg gemakkelijk is om nieuwe modulen aan het extranet toe te

voegen en toegankelijk te stellen aan de juiste klanten en profielen. Iemand zonder enige programmeerervaring kan binnen enkele minuten een module aan het extranet toevoegen. De overige modulen die tijdens het vooronderzoek aan het licht kwamen, kunnen op deze manier dus zeer eenvoudig geïmplementeerd worden binnen het extranet. Ook de eigenschappen en toegankelijkheden van de verschillende klanten en profielen zijn gemakkelijk aan te passen, en door het aan- of uitzetten van IP-detectie en standaardprofielen kan de inlogprocedure per klant versoepeld of verscherpt worden. Het enige minpunt van dit systeem is dat de gebruikershandleiding niet erg uitgebreid is. Er wordt niet gedetailleerd besproken wat men op elk scherm kan doen. Daar staat weer tegenover dat het CMS zodanig opgezet is dat deze handleiding eigenlijk niet nodig is. Over het CMS ben ik van alle opgeleverde producten het meest tevreden.

8.2 Procesevaluatie

Oriëntatie op de opdracht

Tijdens het eerste sollicitatiegesprek was mij kort uitgelegd wat de opdracht inhield. Er werden toen allerlei ideeën genoemd en uiteindelijk kwam het er op neer dat er een extranet ontwikkeld diende te worden waarin deze ideeën uitgewerkt konden worden. Het was voor mij op dat moment dus nog niet geheel duidelijk wat dit precies inhield en daarom was het lastig een opdrachtomschrijving te formuleren. In overleg met een docent is toen afgesproken dat ik tijdens de stageperiode een onderzoek zou uitvoeren om te bepalen wat nou precies de bedoeling van dit extranet was. Dit onderzoek zou een zeer goede beslissing blijken te zijn.

Bij aanvang van de afstudeerperiode ben ik vrijwel meteen met dit onderzoek begonnen door verschillende mensen te interviewen. Ik wilde simpelweg weten welke functionaliteiten dit extranet moest bevatten. Tijdens deze interviews bleek dat iedereen eigenlijk wat anders wilde. Dit had ik niet verwacht. Ik had aangenomen dat binnen Xtilton duidelijk was wat de bedoeling van het extranet was en dat ik van iedereen ongeveer hetzelfde verhaal te horen zou krijgen. Dit bleek niet het geval en per persoon waren de wensen totaal verschillend. Het resultaat was een lijst met zeer diverse functionaliteiten.

Op deze manier kwam ik er uiteindelijk achter dat het belangrijkste onderdeel van de afstudeeropdracht een basisstructuur was waarin al deze verschillende functionaliteiten konden functioneren. Dit onderdeel had ik aanvankelijk over het hoofd gezien, omdat ik me teveel richtte op de functionaliteiten die uit de interviews bleken.

Ik realiseer me dat als ik dit vooronderzoek niet had uitgevoerd, dat ik dan later tijdens de afstudeerperiode in de problemen was gekomen. Als ik bijvoorbeeld gewoonweg begonnen was met de ontwikkeling één van de functionaliteiten, dan zou het later lastig worden om hier een andere, totaal verschillende functionaliteit aan toe te voegen. Uiteindelijk zou ik me wel realiseren dat er een basisstructuur noodzakelijk was, maar dit zou dan waarschijnlijk te laat in de afstudeerperiode gebeurd zijn. Het gevolg hiervan kon dan zijn dat er te weinig tijd over was om de basisstructuur op te zetten. De beslissing tot dit vooronderzoek bleek dus verstandig geweest te zijn.

Het bepalen van de gebruikersgroepen

Tijdens de fase definitiestudie had ik inmiddels een beter beeld van de afstudeeropdracht. Ik zat echter nog steeds met het idee in mijn hoofd dat de belangrijkste gebruikersgroep die van de klanten van Xtilton was. Dit kwam doordat uit het vooronderzoek allerlei functionaliteiten naar voren waren gekomen die voor die klanten bedoeld waren. Het grootste gedeelte van de opdracht zou echter besteed worden aan het ontwikkelen van een systeem voor Xtilton medewerkers. Het gevolg hiervan was dat ik me tijdens de fase definitiestudie teveel op de doelgroep klanten heb gericht, terwijl dat eigenlijk niet nodig was. Hierdoor heb ik uiteindelijk enkele dagen tijd aan activiteiten besteed, zonder dat dit later enig resultaat opleverde.

Informatie over het systeem verzamelen

Voor het vaststellen van de systeemeisen en functionaliteiten van het extranet is met verschillende personen gesproken. Uiteindelijk was het de opdrachtgever die hier laatste beslissing over kon nemen, maar in de praktijk kon ik hiervoor soms beter bij mijn bedrijfsmentor en andere collegae terecht. Bij Xtilton waren er meerdere personen die elk op een bepaald onderdeel van het systeem beslissingen konden nemen, omdat de kennis en expertise van die onderdelen bij hen lag. Hierdoor ontstond bij mij soms het gevoel dat ik niet één, maar meerdere opdrachtgevers had en dit gevoel werd in het laatste deel van mijn afstudeerperiode nog eens versterkt doordat mijn bedrijfsmentor ziek werd. Voor mij was dit een nieuwe situatie, omdat ik in de projecten tijdens mijn studie, evenals bij mij vorige studie, gewend was om met één opdrachtgever te maken te hebben die de beslissingen nam. Het was voor mij tijdens de afstudeerstage dan ook een uitdaging om toch een duidelijk beeld van de opdracht te krijgen/behouden en hiervoor de juiste mensen te raadplegen.

Het indelen van het systeem in pilots

Tijdens de fase definitiestudie heb ik het systeem in pilots opgesplitst. De basisstructuur koos ik als één pilot, omdat dit als één geheel de kern van het extranet vormde, terwijl de overige pilots duidelijk afzonderlijke onderdelen waren. Bij het ontwikkelen van deze pilot in de fase pilotontwikkeling is deze pilot in vier pilotdelen opgesplitst. Tijdens het realiseren van deze pilotdelen, bleek de pilot als geheel toch wat te groot te zijn om als één pilot te ontwerpen, bouwen en testen. Dit bleek doordat de verschillende bouweenheden per pilotdeel duidelijk werden en dus ook de omvang van de gehele pilot en de benodigde ontwikkeltijd. Ik heb op dat moment besloten om de pilotdelen zelf als een soort pilots te beschouwen en deze volgens een iteratiestrategie te ontwikkelen binnen de fase pilotontwikkeling. Op deze manier was het toch mogelijk de afzonderlijke pilotdelen te testen en te itereren, i.p.v. de volledige pilot. Ik merkte echter dat deze situatie voor mezelf minder overzichtelijk was, omdat het pilotontwikkelplan te groot werd en het onderscheid tussen de componenten minder benadrukt werd. Het was beter geweest om gewoon kleinere pilots te definiëren en voor elk een apart pilotontwikkelplan op te stellen. In dit geval betekent dit dat ik dus elk pilotdeel van pilot 1 (de basisstructuur) als een geheel aparte pilot zou benoemen.

Het ontwikkelen van de pilots

Het ontwikkelen van de basisstructuur ging aanvankelijk vlot. Pas toen de verschillende pilotdelen getest werden, bleken er toch wat mankementen in het systeem te zitten. Het aanpassen van de inlogstructuur heeft bij de ontwikkeling van het klantdeel bijvoorbeeld een volledige week gekost. Hieruit werd voor mij nog eens onderstreept wat het belang is van het systeem in een vroeg stadium te laten testen door de gebruikers.

Ik had bij het ontwikkelen van de pilots ervoor gekozen om geen time-boxing toe te passen. Van tevoren wist ik al dat ik dan op moest letten om niet teveel tijd te besteden aan de minder relevante delen van het systeem (de zogenaamde 80/20-regel). Toen na de gebruikerstests allerlei verbeterpunten naar voren kwamen, ben ik in de meeste gevallen daarna samen met de testpersoon deze punten door gaan nemen om de belangrijkste eruit te halen. Ik merkte echter dat ik het moeilijk vond om de minder belangrijke verbeterpunten links te laten liggen. Op enkele momenten heb ik dan ook teveel tijd besteed aan het uitwerken van minder relevante delen van het systeem. Toch is een 80/20 situatie succesvol voorkomen.

De gebruikerstests

Tijdens vorige studieprojecten en stageopdracht had ik al enkele malen tests laten uitvoeren door gebruikers. Meestal gebeurde dit echter slechts één of tweemaal. Tijdens dit project wilde ik hier meer ervaring mee opdoen. Ik besloot de verschillende onderdelen van het systeem dan ook meerdere malen door verschillende gebruikers te laten testen. En dan niet alleen aan het eind van de ontwikkeling van een gehele pilot, maar ook na het afronden van de tussentijdse

pilotdelen. Uit deze tests kwamen veel verbeterpunten naar voren, die ik anders niet of te laat ontdekt zou hebben. Door deze betrokkenheid van de gebruikers te combineren met de iteratiestrategie, zijn de verbeterpunten succesvol en op het juiste moment gerealiseerd. Als ik een waterval-methode had gekozen in plaats van een iteratieve ontwikkelmethode, dan zouden deze verbeterpunten waarschijnlijk niet uitgewerkt zijn.

Invoering

De invoering van het extranet heeft pas in de laatste twee weken plaatsgevonden. Ik heb gemerkt dat hier eigenlijk wat meer tijd voor nodig was, zodat de gebruikshandleiding voor de Xtilton medewerkers wat uitgebreider had kunnen worden uitgewerkt. Het bleef nu bij een beknopte uitwerking. Het systeem zelf is wel succesvol ingevoerd binnen Xtilton en doordat het tijdens de ontwikkeling al door verscheidene mensen getest was, zouden die de handleiding waarschijnlijk niet nodig hebben. Natuurlijk zou ik het extranet ook bij de klanten ingevoerd willen zien, maar dit bleek niet binnen de scope van de opdracht te vallen.

Bijlagen

Bijlage A: Begrippenlijst

Apache	Een webserver
CMS	Content Management System Een applicatie waarmee de inhoud van een computersysteem bewerkt kan worden.
DBMS	Database Management System Systeem voor het beheren van databases
DSDM	Dynamic Systems Development Method Software ontwikkelmethode
Extranet	Het deel van het intranet dat beschikbaar is voor anderen, buiten de organisatie.
GUI	Graphical User Interface Het deel van een programma dat de gebruiker visueel waarneemt.
GUIDE	Graphical User Interface Design and Evaluation Een proces voor het ontwerpen van Grafische User Interfaces.
HTML	Hypertext Markup Language Een taal waarmee statische webpagina's gemaakt worden.
IAD	Iterative Application Development Software ontwikkelmethode
IIS	Internet Information Services/Server Een verzameling services voor het beheren van websites
MySQL	Een database management systeem
PHP	Hypertext Preprocessor Een scripttaal die op de server uitgevoerd wordt en dynamische websites mogelijk maakt.
SLA	Service Level Agreement Een contractuele overeenkomst tussen een dienstverlener en haar klant.
SQL	Structured Query Language Een standaardtaal waarmee gegevens in databases bewerkt kunnen worden.
STD	State Transition Diagram Een diagram dat de status van een klasse toont in bepaalde situaties
TAS	Topdesk Application Server Een web-applicatie waarmee waarmee incidenten gemeld kunnen worden aan helpdesks/servicedesks
VLAN	Een Local Area Netwerk waarbij het netwerk door software (virtueel) in verschillende delen is opgesplitst, in plaats van door hardware.
VPN	Virtual Private Network Een (beveiligde) koppeling tussen (bedrijfs)netwerken via het internet door middel van tunnels

Bijlage B: Literatuurlijst

IAD, Het evolutionair ontwikkelen van informatiesystemen
R.J.H. Tolido
Uitgeverij: Academic Service, 1997.

GUIDE, Graphical User Interface Design and Evaluation, a practical process
David Redmond-Pyle & Alan Moore
Uitgeverij: Prentice Hall Europe, 1995

Obect-Oriented Modeling And Design
James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen
Uitgeverij: Prentice Hall International, 1991

Databasesystemen voor de praktijk
Prof.dr.J.A.Vandenbulcke
Uitgeverij: Kluwer bedrijfswetenschappen, 1997

SQL en relationele databases
Leo Wiegerink, Jeanot Bijpost, Marco de Groot
Uitgeverij: Academic Service, 2001

Bijlage C: Voorbeelden broncode PHP bestanden

Deze bijlage bevat een aantal stukken broncode zoals die te vinden zijn in de PHP bestanden.

Session_check.php

Deze pagina controleert of een klantmedewerker die het extranet bezoekt wel ingelogd is:

```
<?
include ("inc_session.php");
include ("inc_db.php");
include ("inc_referer.php");
//-----------------------------------------------------------------------
//REDIRECT NAAR DE JUISTE PAGINAS
if(!isset($_SESSION['klant_id']))
   {
   //geen sessies -> redirect naar login1
   header("location:login1.php");
   exit;
   }
   else
   {
   if(!isset($_SESSION['prof_id']))
      {
      //alleen klantsessie aanwezig -> redirect naar login2
      header("location:login2.php");
      exit;
      }
      else
      {
      //zowel klant- als profielsessies aanwezig -> redirect naar frame
      header("location:login1_redirect.php");
      exit;
      }
   }
//-----------------------------------------------------------------------
?>
```

Inc_refresh.php

Deze pagina haalt het pad van de pagina op de server op en slaat dit op in een session. Als de gebruiker op F5(refresh) drukt, zal dit pad in het onderste frame geladen worden, zodat altijd de huidige pagina te zien is. Hiermee vervalt een nadeel van frames: dat bij refreshen de startpagina wordt getoond, i.p.v. de laatst bezochte pagina.

```
<?
//-----------------------------------------------------------------------
//IIS versie:
unset($_SESSION['refresh_url']);

$bottom_url = $_SERVER['SCRIPT_NAME'];
if ($_SERVER['QUERY_STRING'] != "")
   {
   $bottom_url .= '?'.$_SERVER['QUERY_STRING'];
   }
$_SESSION['refresh_url'] = $bottom_url;
//-----------------------------------------------------------------------
//bovenstaand script werkt voor IIS en Apache
//in Apache kan bovenstaand script vervangen worden door:
//unset($_SESSION['refresh_url']);
//$_SESSION['refresh_url'] = $_SERVER['REQUEST_URI'];
//-----------------------------------------------------------------------
?>
```

Prof_mod.php

Dit is een gedeelte van de code van de pagina prof_mod.php. Deze pagina toont een scherm waarop de CMS gebruiker checkboxes kan selecteren om aan te geven tot welke modulen een bepaald profiel toegang heeft.

```
<?
//-------------------------------------------------------------------
//haal gegevens op uit MODULE voor dit PROFIEL
$query_mod =   "SELECT m.mod_id as mod_id_prof
                FROM module m, prof_mod pm, profiel p
                WHERE pm.pm_prof_id = ' ".$prof_id." '
                AND pm.pm_mod_id = m.mod_id
                ORDER BY m.naam ASC";
$result_mod = mysql_query($query_mod);
//-------------------------------------------------------------------
//haal alle MENU_ITEMS op
$query_mi =     "SELECT mi_id, naam
                FROM menu_item
                ORDER BY rang ASC";
$result_mi = mysql_query($query_mi);
//-------------------------------------------------------------------
?>

<FORM name="set_pm" ACTION="set_prof_mod.php" METHOD="POST">
<INPUT TYPE="reset" NAME="cancel" VALUE="Reset">
<INPUT TYPE="submit" NAME="set_prof_mod" VALUE="Opslaan">
> selecteer de modulen die u toegankelijk wilt maken voor dit profiel:<BR><BR>

<?
echo $dikgedrukte_modulen;
//-------------------------------------------------------------------
//genereer de rijen van de tabel
while(list($mi_id, $naam) = mysql_fetch_row($result_mi))
   {
?>
<TABLE class="table_main" WIDTH="100%" BORDER="0" CELLSPACING="0" CELLPADDING="5">
<TR class='td_main_title' ALIGN="center">
<TD WIDTH="280" ALIGN="left"><? echo $naam; ?> </TD><TD WIDTH="*">Commentaar</TD></TR>
<?
//-------------------------------------------------------------------
//haal all MODULEN van dit menu_item op
$query_mod_mi= "SELECT mod_id, naam, commentaar
                FROM module
                WHERE m_mi_id = '".$mi_id."'
                ORDER BY rang ASC";
$result_mod_mi= mysql_query($query_mod_mi);
//-------------------------------------------------------------------
$checked = "";
$standaard1 = "";
$standaard2 = "";
while(list($mod_id, $naam, $commentaar) = mysql_fetch_row($result_mod_mi))
        {
//-------------------------------------------------------------------
        //dit stuk selecteert checkboxes
        while(list($mod_id_prof) = mysql_fetch_row($result_mod))
           {
           //controleer welke modulen al bij profiel horen
           if ($mod_id_prof == $mod_id)
              {
              //checkbox aanvinken
              $checked = "checked";
              break;
              }
              else
              {
```

```
                $checked = "";
                }
            }
        //reset de array als deze niet leeg is, zodat deze opnieuwe vergeleken kan worden
        if(mysql_num_rows($result_mod)!=0)
            {
            mysql_data_seek ($result_mod,0);
            }
//-------------------------------------------------------------------
        //dit stuk geeft aan welke modulen standaard zijn
        while(list($pm_mod_id) = mysql_fetch_row($result_stanmod))
            {
            //controleer welke modulen al bij profiel horen
            if ($pm_mod_id == $mod_id)
                {
                //standaard toegankelijke modulen dikgedrukt maken
                $standaard1 = "<B>";
                $standaard2 = "</B>";
                break;
                }
                else
                {
                $standaard1 = "";
                $standaard2 = "";
                }
            }
        //reset de array als deze niet leeg is, zodat deze opnieuw vergeleken kan worden
        if(mysql_num_rows($result_stanmod)!=0)
            {
            mysql_data_seek ($result_stanmod,0);
            }
//-------------------------------------------------------------------
        echo ("<TR>
                <TD class='td_main' ALIGN='left'><LABEL FOR='text$mod_id'>
                <INPUT TYPE=checkbox NAME=checkbox[] VALUE=$mod_id $checked
                id='text$mod_id'>
                $standaard1 $naam $standaard2  </LABEL></TD>
                <TD class='td_main'>$commentaar  </TD>
                </TR>\n");
        }
//-------------------------------------------------------------------
?>
</TABLE></BR>
```

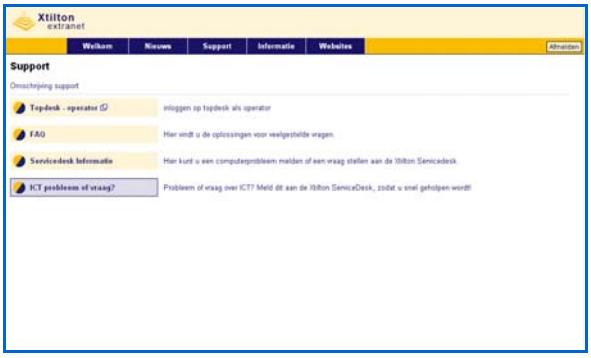
Rangschik.php

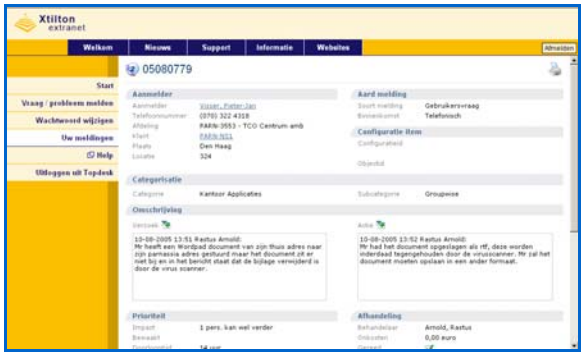
Deze pagina rangschikt de modulen, wanneer de CMS gebruiker de volgorde ervan aanpast.

```
<?
$mod_id = $_REQUEST['mod_id'];
$mi_id = $_REQUEST['mi_id'];
$rang = $_REQUEST['rang'];
$move = $_REQUEST['move'];
//-------------------------------------------------------------------
if ($move == "up")
{
   //als op PIJLTJE OMHOOG is geklikt
   $query_up =

      "SELECT MAX(rang) as rang_boven2
      FROM module
      WHERE rang < '".$rang."'
      AND m_mi_id = '".$mi_id."'";

   $result_up = mysql_query($query_up);
   while(list($rang_boven2) = mysql_fetch_row($result_up))
      {
      $rang_boven = $rang_boven2;
      }
```

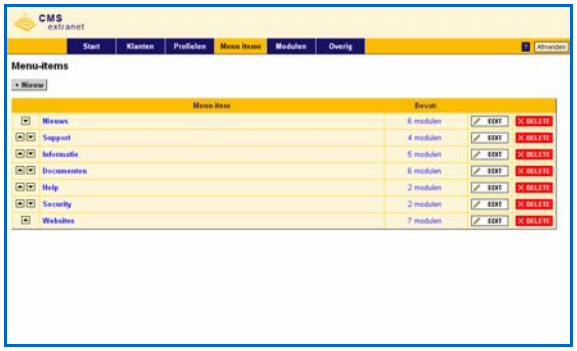
```
   //beweeg hogere rang omlaag
   $query_move_down =

      "UPDATE module
      SET rang = '".$rang."'
      WHERE rang = '".$rang_boven."'
      AND m_mi_id = '".$mi_id."'";

   $result_move_down= mysql_query($query_move_down);
   //beweeg lagere rang omhoog
   $query_move_up =

      "UPDATE module
      SET rang = '".$rang_boven."'
      WHERE rang = '".$rang."'
      AND m_mi_id = '".$mi_id."'
      AND mod_id = '".$mod_id."'";

   $result_move_up= mysql_query($query_move_up);
//-----------------------------------------------------------------------
}else{
//-----------------------------------------------------------------------
   //als op PIJLTJE OMLAAG is geklikt
   $query_down =

      "SELECT MIN(rang) as rang_onder2
      FROM module
      WHERE rang > '".$rang."'
      AND m_mi_id = '".$mi_id."'";

   $result_down = mysql_query($query_down);
   while(list($rang_onder2) = mysql_fetch_row($result_down))
      {
      $rang_onder = $rang_onder2;
      }
   //beweeg lagere rang omhoog
   $query_move_up =

      "UPDATE module
      SET rang = '".$rang."'
      WHERE rang = '".$rang_onder."'
      AND m_mi_id = '".$mi_id."'";

   $result_move_up= mysql_query($query_move_up);
   //beweeg hogere rang omlaag
   $query_move_down =

      "UPDATE module
      SET rang = '".$rang_onder."'
      WHERE rang = '".$rang."'
      AND m_mi_id = '".$mi_id."'
      AND mod_id = '".$mod_id."'";

   $result_move_down= mysql_query($query_move_down);
}?>
```

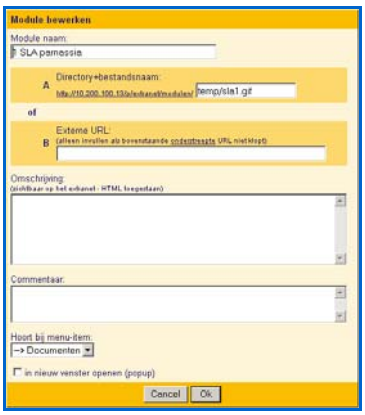
Bijlage D: Schermafbeeldingen van het extranet

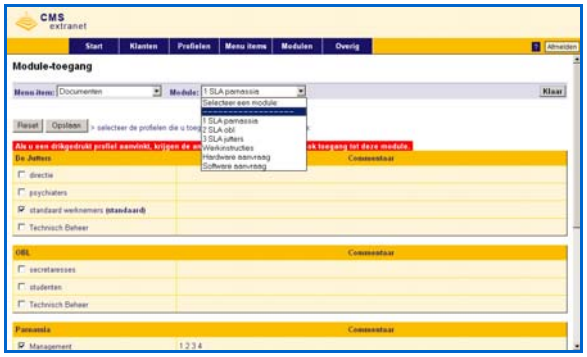


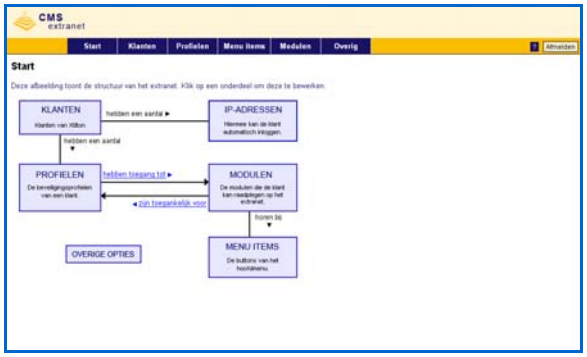


Bijlage E: Schermafbeeldingen van het CMS









Bijlage F: Cognitive Walkthrough formulier

Onderstaand formulier heb ik als observator gebruikt bij de gebruikerstests.

Testpersoon: -naam-
Scenario: -nr-

Fouten	Acties van de gebruiker	Think Aloud kreten
	1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. **Geschatte totaaltijd:**	

Extra notities observator:

Visuele weergave CMS scherm:

start	Klanten	profielen	Menu-item	modulen	overig	

Bijlage G: MySQL implementatiemodel

Het MySQL implementatiemodel van de database:

```
CREATE TABLE klant (
klant_id        INT(10)   NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
naam            VARCHAR(20)  NOT NULL,
intro_tekst     TEXT,
commentaar      TEXT,
ipcheck         TINYINT(1)  NOT NULL);

CREATE TABLE profiel (
prof_id         INT(10)  NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
naam            VARCHAR(20)  NOT NULL,
password        VARCHAR(10),
is_standaard    TINYINT(1)  NOT NULL,
commentaar      TEXT,
p_klant_id      INT(10)  NOT NULL REFERENCES klant(klant_id));

CREATE TABLE ip (
ip_id           INT(10)  NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
ip1             INT(3),
ip2             INT(3),
ip3             INT(3),
commentaar      TEXT,
i_klant_id      INT(10)  NOT NULL  REFERENCES klant(klant_id));

CREATE TABLE prof_mod (
pm_prof_id      INT(10)  NOT NULL  REFERENCES profiel(prof_id),
pm_mod_id       INT(10)  NOT NULL  REFERENCES module(mod_id_id));

CREATE TABLE menu_item (
mi_id           INT(10)  NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
naam            VARCHAR(20)  NOT NULL,
rang            INT(5)  NOT NULL,
omschrijving    TEXT,
commentaar      TEXT);

CREATE TABLE module (
mod_id          INT(10)  NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
naam            VARCHAR(40)  NOT NULL,
link            TEXT,
url             TEXT,
rang            INT(5)  NOT NULL,
popup           TINYINT(1)  NOT NULL,
omschrijving    TEXT,
commentaar      TEXT,
m_mi_id         INT(10)  NOT NULL  REFERENCES menu_item(mi_id));

CREATE TABLE overig (
ov_id           INT(10)  NOT NULL  AUTO_INCREMENT  PRIMARY KEY,
standaard_dir   TEXT,
cms_password    VARCHAR(10)  NOT NULL);
```