

KCC Klantenmap:
Een Internetsite voor
Korton Computer Communication B.V.

Marcel Stokman

Periode:	9 februari 2004 t/m 11 juni 2004
Plaats:	Hoofddorp
Auteur:	Marcel Stokman
Opleiding:	Informatiekunde van de Haagse Hogeschool sector Informatica & Informatiekunde
Afstudeerrichting:	OSTI
Examinatoren:	Dhr. A.W.W.M. Biegstraaten Dhr. J. van Peski
Gastbedrijf:	Korton Internet
Bedrijfsmentor:	Dhr. M. Huijkman

Referaat

Dit verslag beschrijft het tot stand komen van de website voor Korton Computer Communication. De volgende termen komen binnen dit verslag naar voren.

- XML
- PHP 5
- Fusebox
- Adalon
- PEAR

Voorwoord

Dit rapport handelt over de realisering van de afstudeerppdracht die ik overeenkwam met Korton Computer Communication B.V. te Hoofddorp (verder KCC). KCC is een werkmaatschappij die onderdeel uitmaakt van de Korton Group B.V., het bedrijf waar ik in het kader van het afstuderen aan de opleiding Informatiekunde van de Haagse Hogeschool sector Informatica en Informatiekunde, mijn stage heb vervuld. Gedurende de uitvoering van de stage was ik in dienst bij Korton Internet B.V., eveneens een werkmaatschappij van Korton Group.

Ik dank de heren B. Arendsen en R. van Harskamp, medewerkers van KCC en dhr. M. Huijckman, medewerker van Korton Internet, voor hun begeleiding, hulp en ondersteuning bij het uitvoeren van de opdracht. Naast de begeleiding vanuit het bedrijf hebben de heren A.W.W.M. Biegstraaten en J. van Peski mij in de loop van het hele traject begeleid vanuit de hogeschool. Ik dank hen voor de support bij het voltooien van de opdracht en het samenstellen van dit verslag. Een extra dankwoord gaat uit naar drs H. Hardebol voor zijn bijdrage aan dit verslag.

Marcel Stokman
Hoofddorp, juni 2004

Inhoudsopgave

Referaat.....	2
Voorwoord.....	3
1. Inleiding.....	6
1.1 Een schets van de organisatie.....	6
1.2 Een snuifje cultuur van KCC.....	7
1.3 De inrichting van het verslag	7
2. Problematiek en afstudeeropdracht.....	8
2.1 Problematiek	8
2.2 De afstudeeropdracht	8
2.3 Randvoorwaarden	9
3. Plan van eisen.....	10
3.1 Eisen en wensen KCC.....	10
3.2 Eisen en wensen van de medewerkers.....	11
4. Ordening van de werkzaamheden.....	12
4.1 Voorbereidende werkzaamheden.....	12
4.2 Werkzaamheden die betrekking hebben op de realisatie	13
4.3 Werkzaamheden in het kader van evaluatie en verificatie.....	14
4.4. Precedentieschema.....	15
5. Vooronderzoek.....	17
5.1 Opslagstructuur.....	17
5.2 Keuze voor programmeertaal	20
5.3 Template engines.....	22
5.4 Rechtensysteem	24
5.5 Pear DB.....	25
5.6 Veilig inloggen	25
5.7 Locken van XML bestanden	26
6. Ontwerp.....	27
6.1 Databaseontwerp	27
6.2 Fusebox ontwerp met Adalon	29
6.3 Pilot ontwerp.....	33
6.4 Schermontwerpen.....	34
6.5 Structuur ontwerp (XML bestanden).....	35

7. Implementatie	36
7.1 Opzetten structuur.....	36
7.2 Inlogpagina	36
7.3 Klantenpagina	37
7.4 Detailpagina	38
7.5 Urendecclaratie	42
7.6 Todo	44
7.7 Uploaden van bestanden.....	46
7.8 Zoeken.....	46
8. Testen.....	48
8.1 Debuggen	48
8.2 Testsets.....	48
8.3 Gebruikerstest.....	49
8.4 Acceptatietest.....	50
8.5 Conclusie.....	50
9. Evaluatie.....	51
10. Slotwoord.....	54
Literatuurlijst	55
Figurenlijst.....	56
Bijlage A: Plan van aanpak	57

1. Inleiding

Korton Computer Communication is op 1 maart 1997 als eenmanszaak onder de naam "KCC" gestart met het aanbieden van ICT dienstverlening. Vanaf 1 januari 1999 is Korton Computer Communication een besloten vennootschap geworden. Vanwege de groei van de bedrijfsactiviteiten is per 1 januari 2000 besloten de onderscheiden activiteiten in aparte B.V.'s onder te brengen en om de bedrijfsnaam te veranderen in Korton Group B.V.

1.1 Een schets van de organisatie

De Korton Group B.V. bestaat uit drie werkmaatschappijen:

Korton Computer Communication B.V.

Korton Computer Communication verzorgt de installatie, configuratie, training, levering van soft- en hardware en het dagelijkse beheer van de computernetwerken en communicatiesystemen van haar klanten, waaronder veel makelaarskantoren.

Voor de klanten zijn er onderhoudscontracten beschikbaar, voor periodiek onderhoud aan soft- en hardware en voor ondersteuning van de medewerkers van de klanten.

Korton Software B.V.

Korton Software verzorgt de ontwikkeling, verkoop, implementatie en ondersteuning van WinTax, maatwerk software producten alsook de aanbod Wizard van de Nederlandse Vereniging van Makelaars o.g. en vastgoeddeskundigen.

Korton Software heeft zich voornamelijk gespecialiseerd in objectgeoriënteerde programmeertechnieken en maakt veelal gebruik van Delphi en C++ als ontwikkelomgeving.

Korton Internet B.V.

Korton Internet is gespecialiseerd in het aanbieden van totale Internet oplossingen. Als totaal ondersteunende Internet partner stemmen ze de bedrijfsvoering van de klanten geheel af op het gebruik van Internet oplossingen.

Hierbij kan worden gedacht aan het ontwikkelen van dynamische database gestuurde Internet sites met shopping- en backoffice functionaliteit. Ook ontwikkelt men toepassingen op maat die verbetering en dimensie aanbrengen binnen de huidige bedrijfsvoering van een klant en geven ze aan bestaande en/of nieuwe Internet sites een extra meerwaarde in functionaliteit voor online doelgroepen.

Daarnaast levert Korton Internet facilitaire Internet diensten zoals "dedicated server hosting", "Router en Firewall" oplossingen en vaste Internet lijnverbindingen voor een permanente aansluiting op het Internet.

1.2 Een snuifje cultuur van KCC

KCC maakt gebruik van papieren Excel sheets als klantenmappen voor het beheer van klant gegevens. Deze klantenmappen bevinden zich in het kantoor en worden regelmatig gebruikt voor systeembeheer. Na te zijn gebruikt en bijgewerkt met de laatste wijzigingen dienen de mappen opgeborgen te worden. Slechts een deel van de medewerkers houdt zich aan deze instructie.

1.3 De inrichting van het verslag

De opbouw van dit verslag is als volgt. In hoofdstuk 2 wordt de afstudeeropdracht behandeld waarna de gestelde eisen in hoofdstuk 3 worden besproken. Aan de hand van deze hoofdstukken is er een ordening aangebracht die terug te vinden is in hoofdstuk 4. Hoofdstuk 5 beschrijft het traject dat vooraf gegaan is aan het ontwerp die terug te vinden is in hoofdstuk 6. De implementatie van de website wordt beschreven in hoofdstuk 7 en het testverslag bevindt zich in hoofdstuk 8. Het verslag wordt afgesloten met een procesevaluatie in hoofdstuk 9 en een slotwoord in hoofdstuk 10.

2. Problematiek en afstudeeropdracht

Dit hoofdstuk omvat de beschrijving van de problematiek die ten grondslag ligt aan de afstudeeropdracht (2.1) alsmede de formulering van de afstudeeropdracht als zodanig (2.2).

2.1 Problematiek

De technische medewerkers van KCC beschikken niet over de mogelijkheden om klantgegevens vanaf de werkplek dan wel vanaf locatie op te vragen, te raadplegen of te wijzigen. Deze gegevens behelzen naast de algemene klant gegevens ook de server instellingen, wachtwoorden, "To Do's" etc.

In de huidige situatie zijn de klant gegevens opgeslagen in zgn. klantenmappen, dossiers in papieren alsook in digitale vorm, die zich in het kantoor van Korton bevinden.

Het opvragen en raadplegen van deze dossiers zowel als het onderhouden en bijwerken ervan wordt door een deel van de medewerkers als omslachtig en verouderd ervaren. Met name op momenten dat de dossiers moeten worden geraadpleegd buiten standaardsituaties om resulteert de van kracht zijnde gang van zaken regelmatig in ongewenst gedrag bij een deel van de medewerkers. Het niet juist of niet volledig bijwerken van dossiers en het niet tijdig of in het geheel niet retourneren van dossiers zijn daar voorbeelden van. Een en ander leidt tot gebrekkige managementinformatie, irritaties bij medewerkers die later gebruik maken van sub-optimaal bijgehouden mappen, tot fouten in orders en tot klachten van klanten.

Op management niveau zowel als (deels) op de werkvloer is de geschetste situatie als een probleem aangemerkt. Vanuit het management is nagegaan of en zo ja waarom het deel uitmaakt van een meer omvattende problematiek. Het valt buiten het bestek van dit werkstuk om daar verder op in te gaan.

In het overleg over mijn stageopdracht met vertegenwoordigers vanuit het management, is mij gevraagd of ik met mijn informatiekundige kennis en kunde een hedendaags, en bij het bedrijf passend stuk gereedschap zou kunnen ontwikkelen dat er toe zou kunnen bijdragen dat de problematiek zoals geschetst, vermindert of zelfs verdwijnt.

'Passend bij het bedrijf' kan dan opgevat worden als "het imago van het bedrijf ondersteunend" zowel als "in lijn te zijn met wensen van management en eindgebruikers".

Ik heb overwogen dat gereedschap vanuit zichzelf nooit een bijdrage kan leveren aan organisatieontwikkeling. De oude volkswijsheid "goed gereedschap is het halve werk" zou evenwel ook bij Korton kunnen opgaan. Gereedschap dat het werk van de professional zonder weerstand ondersteunt, verhoogt beslist het plezier in het werken. In afgeleide zin is derhalve niet ondenkbaar dat een gereedschap dat het soepel raadplegen van klantgegevens mogelijk maakt, bijdraagt aan het voorkomen van ongewenste effecten.

2.2 De afstudeeropdracht

De afstudeeropdracht luidt:

Ontwikkel een Internetsite waarop de medewerkers van de technische dienst kunnen inloggen om zo de klantgegevens te kunnen raadplegen en te wijzigen.

2.3 Randvoorwaarden

Korton stelt voor de uitvoering van de opdracht de volgende uitrusting ter beschikking:

2.3.1 Hardware

- Laptop (Fujitsu Siemens E serie lifebook, PIII 700mhz, 512Mb intern geheugen, 20gb harddisk)
- Extern beeldscherm, toetsenbord en muis

2.3.2 Software

- Microsoft Windows 2000 professional
- Microsoft Office 2000
- PHP 5
- Sambar 6.0 (webserver) en Microsoft IIS
- MySQL 4 en Microsoft SQL server 2000
- Adalon (case tool voor Fusebox)

2.3.3 Methoden en Technieken

Voor de ontwikkeling zal er gebruik gemaakt gaan worden van de Fusebox Lifecycle Process (FLiP of gewoon Fusebox). Dit is een ontwikkelmethode die door Korton gebruikt wordt. Deze methode zorgt ervoor dat er gemakkelijk met meerdere personen aan een ontwikkeltraject gewerkt kan worden. Daarnaast is het een wijze van werken die ervoor zorgt dat de implementatie gestructureerd verloopt. Er zal gebruik gemaakt gaan worden van de aanwezige technieken binnen de Fusebox ontwikkel methode.

3. Plan van eisen

3.1 Eisen en wensen KCC

De opdrachtgever heeft een aantal eisen en wensen geformuleerd waaraan de te ontwikkelen website moet voldoen.

- De website moet afgeschermd worden door middel van een inlog systeem
- In de toekomst moet het mogelijk zijn dat een klant zijn of haar gegevens bekijkt en mogelijk muteert*
- De KCC medewerkers moeten toegang hebben tot het systeem
- Gegevens kunnen niet door de gebruikers verwijderd worden
- De gebruikers dienen gedurende het ontwikkeltraject bijvoorbeeld d.m.v. presentaties op de hoogte gehouden te worden van de voortgang van het project
- Medewerkers kunnen bij tussenevaluaties aanvullende wensen te berde brengen

Naast de opdrachtgever heeft ook de "werkgever" in de rol van opdrachtnemer eisen en wensen neergelegd waaraan de te ontwikkelen site behoort te beantwoorden.

- De website moet in de toekomst ook voor andere doeleinden gebruikt kunnen worden naast het bijhouden van de KCC klanten. Denk hieraan bijvoorbeeld aan het bijhouden en documenteren van de Wintax klanten met hun bijbehorende gegevens
- Leesbaarheid en onderhoudbaarheid van data opslag moet van hoge kwaliteit zijn
- Leesbaarheid en onderhoudbaarheid van de source code moet van hoge kwaliteit zijn
- De website moet het imago van Korton ondersteunen
- De website moet ontwikkeld worden in overeenstemming met de huisstijl van Korton
- Gebruik maken van de standaarden die aanwezig zijn binnen de afdeling
 - o Het wordt ontwikkeld en geïmplementeerd in de Fusebox structuur
 - o De variabelen volgen de Hongaarse notatie
De prefix wordt bepaald aan de hand van het type variabele
 - o Directory opbouw van de website is gelijk aan de standaard websites van Korton Internet
- Inloggen moet gebeuren door middel van een beveiligde verbinding naar de server (HTTPS)
- De data moeten dynamisch beschikbaar zijn. Dat wil zeggen dat als op dit moment er bijvoorbeeld een floppy drive in een server zit (floppy ja/nee) die op enig moment zonder problemen vervangen moet kunnen worden door een dvd-recorder (dvd-rec ja/nee). Een mogelijkheid hiervoor is het gebruik van XML

* Opmerking:

In het ontwerp zal er rekening gehouden worden met de mogelijkheden om klanten hun eigen gegevens te laten bekijken en eventueel te beheren. De implementatie hiervan valt buiten de opdracht.

3.2 Eisen en wensen van de medewerkers.

Na kennis te hebben genomen van de eisen en wensen van opdrachtgever en werkgever heb ik geprobeerd om door middel van een vragenlijst de medewerkers te bewegen mij hun eisen en wensen kenbaar te maken. De respons bleef, ook na twee herinneringen, nihil. Aanvankelijk heb ik dit, naar al spoedig bleek geheel ten onrechte, geïnterpreteerd als desinteresse. In de dagelijkse omgang op de werkvloer bemerkte ik dat een vragenlijst in de druk van een productieomgeving al gauw onderop de stapel van todo's wordt gelegd. In de contacten met de medewerkers kon ik evenwel steeds mijn vorderingen toetsen aan hun ideeën.

4. Ordening van de werkzaamheden

De in het kader van de afstudeeropdracht uit te voeren werkzaamheden kunnen verdeeld worden in drie groepen:

4.1 Voorbereidende werkzaamheden

Werkzaamheden die uitgevoerd dienen te worden zodat de opdracht overeenkomstig de afspraken met de opdrachtgever uitgevoerd kan worden.

- *Plan van aanpak*
Het plan van aanpak geldt als handleiding voor het project, hierin staat onder andere beschreven welke producten er worden opgeleverd en bevat het een planning van de uit te voeren activiteiten. Aan de hand van het plan van aanpak kan de voortgang van het project gecontroleerd worden. Het plan van aanpak bevat in ieder geval: probleemstelling, doelstelling, te gebruiken methoden en technieken, planning en de mogelijke risico's.
- *Interviewen toekomstige gebruikers*
De toekomstige gebruiker zullen geïnterviewd worden om een duidelijk beeld te krijgen van de huidige situatie en de gewenste situatie.
- *Bepalen welke (tussen)producten moeten worden opgeleverd*
 - o Documentatie van het vooronderzoek
Het vooronderzoek vormt de basis van de te nemen beslissingen. De resultaten alsmede de onderzochte mogelijkheden moeten gedocumenteerd worden voor later gebruik. Deze documentatie is voor Korton Internet.
 - o Ontwerp volgens de Fusebox ontwikkelmethode
Het ontwerp vormt de basis voor eventuele aanpassingen in de toekomst. Het is een product voor Korton Internet.
 - o Database
De database bevat essentiële gegevens voor de werking van de website, deze zal tezamen met de website opgeleverd worden aan KCC.
 - o Implementatie in PHP
Dit omvat de volledige website.
 - o Documentatie met betrekking tot de implementatie
Deze documentatie bevat implementatie specifieke onderdelen, deze wordt opgeleverd aan Korton Internet.

- Planning

De gehele opdracht is op te delen in een aantal afzonderlijke activiteiten namelijk: vooronderzoek, ontwerp, implementatie en samenstellen verslag. Voor het uitvoeren van de activiteiten binnen de beschikbare tijd is er een planning opgesteld. De planning is gebaseerd op de beschikbare weken binnen de afstudeerperiode waarbij er rekening is gehouden met het samenstellen van het eindverslag. De volgende tijd is gereserveerd voor een activiteit.

Vooronderzoek

Voor het gehele vooronderzoek zijn 4 weken gereserveerd. Het vooronderzoek bestaat uit het bepalen van de te gebruiken opslagstructuur en te gebruiken software. Om de juiste opslag structuur te kunnen bepalen wordt er gekeken naar de mogelijkheden die een database biedt en mogelijkheden van XML. Tevens wordt er gekeken naar de mogelijkheden met betrekking tot de template engines en gebruik van PEAR pakketten. Het geheel wordt uitgebreid besproken in hoofdstuk 5.

Ontwerp

Het ontwerp zal worden samengesteld in 2 weken. Het ontwerp bestaat uit de volgende deelonderdelen: Inloggen, Klanten, Details, Urendecclaratie, Todo's en Zoeken. Het ontwerp wordt gemaakt volgens de Fusebox methode en ik zal gebruik maken van Adalon.

Implementatie

De implementatie zal worden uitgevoerd in 9 weken tijd. Het implementeren van de gehele website is opgedeeld in kleinere stukken zodat er een duidelijk overzicht van de voortgang gekregen kan worden. De volgende deelonderdelen zijn aanwezig: Inloggen, klanten, details, urendecclaratie, todo's en zoeken.

Verslag

Voor het samenstellen van het verslag zijn 3 weken gereserveerd. Dit zal niet voldoende zijn om het verslag volledig samen te stellen daarom zal ik ook buiten de werktijden om aan het verslag werken. Tijdens het schrijven van het verslag zullen de toekomstige gebruikers op de testomgeving gaan werken. Er is in deze periode ook tijd gereserveerd om hierbij begeleiding te bieden opdat mogelijke fouten zo snel mogelijk te kunnen herstellen.

4.2 Werkzaamheden die betrekking hebben op de realisatie

- Vooronderzoek

Het vooronderzoek bestaat uit het onderzoeken van de diverse mogelijkheden. De volgende onderdelen zullen zeker in het vooronderzoek behandeld worden: opslag structuur en de versies en mogelijkheden van PHP.

- Precedentieschema

Het precedentieschema beschrijft de procesgang gedurende de gehele afstudeerperiode. Zowel de processen als de producten komen erin naar voren.

- Pilot ontwikkeling
Het pilot ontwerp zal de mogelijke acties van het systeem gaan beschrijven, deze kan gebruikt worden voor de implementatie alsmede voor het testen.
- Fusebox ontwerp
Voordat er werkelijk met de implementatie begonnen kan worden zal er eerst een gestructureerd ontwerp volgens de fusebox ontwikkel methode gemaakt worden. Fusebox is een methode welke sinds lange tijd binnen Korton gebruikt word. Fusebox stelt de programmeur in staat om gestructureerd te ontwerpen en programmeren.
- Ontwikkelen XML structuur
De structuur van de XML bestanden* is van belang om een correcte implementatie te krijgen. De structuur zal uit twee delen bestaan. Eén gedeelte voor de werkelijke gegevens en één gedeelte voor de definitie van de werkelijke gegevens.
- Database ontwerp
De database zal de website gaan ondersteunen, om een correcte en consistent geheel te krijgen zal er een database ontwerp gemaakt worden.
- Implementatie
De implementatie zal geschieden in PHP**. PHP is een serverside scripting taal waarmee het mogelijk is om dynamische webpagina's te maken. PHP is zeer geschikt voor deze opdracht omdat het ook beschikt over een XML parser.

* De keuze voor XML is verantwoord in paragraaf 5.1.

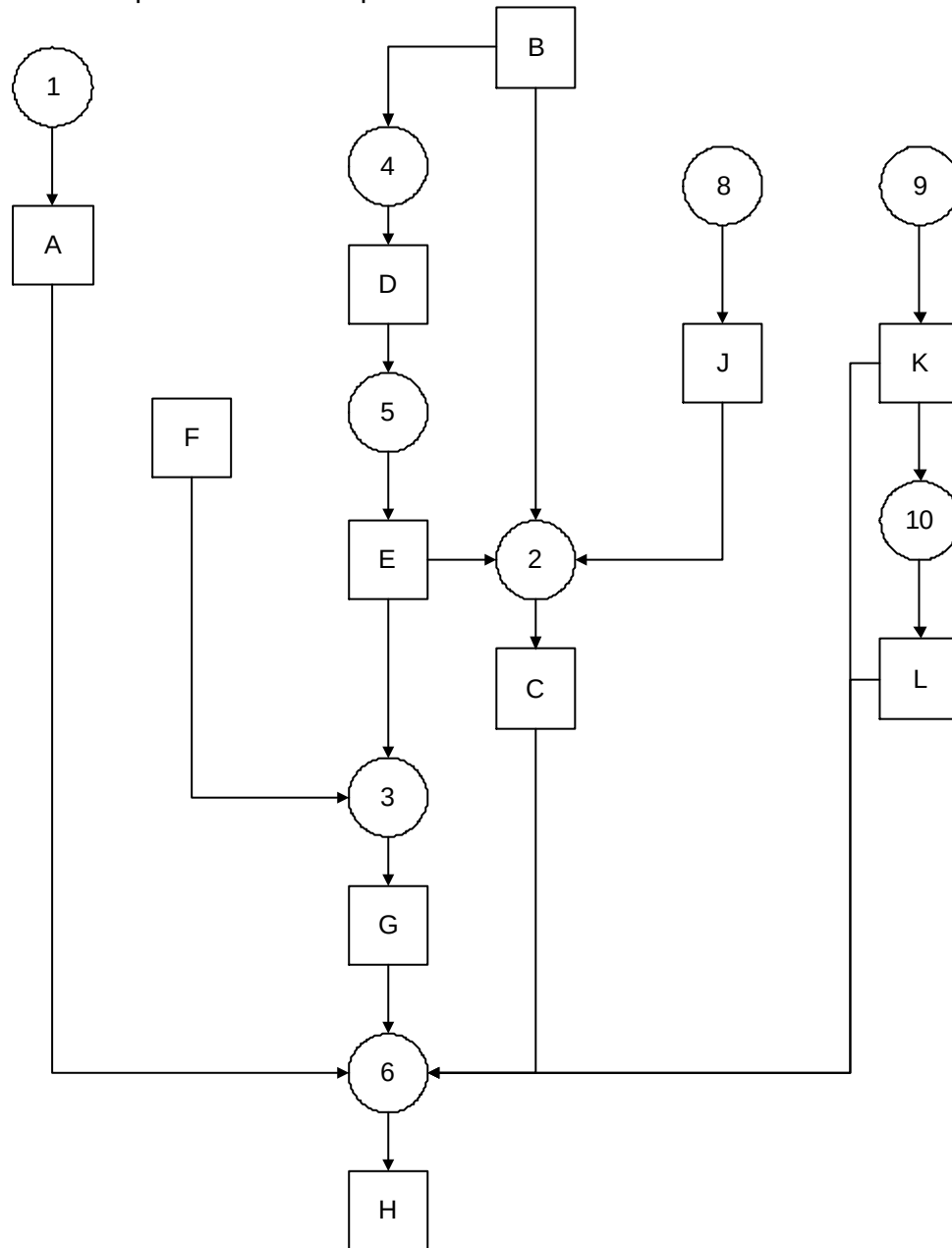
** De keuze voor PHP is verantwoord in paragraaf 5.2.

4.3 Werkzaamheden in het kader van evaluatie en verificatie

Zijn de producten van de uitgevoerde werkzaamheden volgens afspraak uitgevoerd door middel van whitebox en blackbox testing.

4.4. Precedentieschema

De ordening van de werkzaamheden is in een precedentie schema tot in detail uitgewerkt. Het precedentieschema beschrijft de procesgang gedurende de gehele afstudeerperiode. Zowel de processen als de producten komen erin naar voren.



Figuur 1: Precedentieschema

Toelichting bij het precedentieschema

1. Onderzoek naar het gebruik van XML als opslagstructuur
 2. Ontwikkelen XML datastructuur (vragen, inhoud vraag gedeelte)
 3. Samenstellen fusebox ontwerp
 4. Samenstellen vragenlijst
 5. Interviewen toekomstige gebruikers
 6. Implementatie van het fusebox ontwerp
 8. Ontwikkelen XML Meta structuur (structuur xml files)
 9. Ontwikkelen lock structuur XML bestanden
 10. Ontwikkelen database structuur
-
- A. Resultaten XML onderzoek
 - B. Huidige klantenmappen
 - C. XML Datastructuur
 - D. Vragenlijst
 - E. Resultaten vragenlijst
 - F. Adelon
 - G. Fusebox ontwerp
 - H. KCC Klantenmap
 - J. XML Meta structuur
 - K. Lock structuur XML bestanden
 - L. Database ontwerp

5. Vooronderzoek

Het doel van het vooronderzoek is het vergaren van informatie over geschikte structurerende tools.

Om de juiste beslissingen te kunnen nemen met betrekking tot de manier van data opslaan en de te gebruiken software moeten de meest in aanmerking komende alternatieven met elkaar worden vergeleken. Voor het opslaan van de gegevens is eenvoud van groot belang. Ook de gevolgen voor de leesbaarheid en onderhoudbaarheid van de source code is van gewicht. De snelheid speelt een minder belangrijke rol omdat er een beperkt aantal gebruikers tegelijk gebruik zal maken van de website. Om ergernissen van de gebruikers te voorkomen is het wel van belang dat de snelheid niet irriterend laag is. Wanneer een gebruiker enkele seconden moet wachten voordat de gegevens op het scherm staan zal dit als irritant beschouwd worden. De exacte tijdsgrens is niet bekend. Om te voorkomen dat aan het eind van het project de snelheid tegen valt en de gebruikers het daarom liever niet gebruiken zal de snelheid van te voren getest moeten worden, dit is meegenomen in het vooronderzoek.

Als eerste bespreek ik de keuze voor het opslaan van de data, waarna ik verder ga met de te gebruiken software. Tevens behandel ik mijn keuze voor PHP 5 en beschrijf ik het gebruik van template engines om de opmaak en logica van elkaar gescheiden te houden. In de opdrachtoomschrijving is vastgesteld dat er een rechten systeem ontwikkeld zal worden. Ik heb onderzocht welke mogelijkheden er zijn met betrekking tot het gebruik van rechten binnen de website, deze resultaten zullen gebruikt worden bij de verdere ontwikkeling en toekomstige implementatie van de rechten binnen de website.

5.1 Opslagstructuur

Tijdens de eerste gesprekken met het management van de afdeling zijn de mogelijke manieren van opslaan besproken. Twee mogelijke oplossingen komen in aanmerking, een database of XML bestanden. XML kan gebruikt worden als free format database en is hierdoor flexibeler dan een DBMS. Aan de hand van deze ideeën ben ik gaan onderzoeken welke van de twee oplossingen het beste zou zijn en of er mogelijk nog alternatieven waren.

5.1.1 XML

XML staat voor eXtensible Markup Language. Het is een formattering van tekst op een dusdanige manier dat het voor mens en machine te lezen is. Tevens moet het zichzelf beschrijven, er zijn dus geen andere tools nodig om de inhoud te kunnen lezen. Het kan gebruikt worden om gegevens mee op te slaan of te versturen via een netwerk.

De XML specificatie is ontwikkeld door W3C. De specificatie is opgesteld in het Engels daarnaast zijn er verschillende vertalingen onder andere in het Nederlands. De Nederlandse omschrijving geeft het volgende weer over XML.

W3C:

Extensible Markup Language, afgekort als XML, omschrijft een klasse van data objecten die XML documenten genoemd worden en deels het gedrag omschrijven van computer programma's die deze documenten verwerken. XML is een toepassing profiel of beperkte vorm van SGML, de Standard Generalized Markup Language [ISO 8879]. Standaard (bv construction) zijn XML documenten conforme SGML documenten.

XML documenten worden opgemaakt uit opslageenheden die entiteiten genoemd worden welke ofwel parsed of unparsed data bevatten. Parsed data wordt opgemaakt uit karakters, waarvan sommige karakters data vormen en sommige opmaak. Opmaak encodeert een omschrijving van de opslag van een document lay-out en logische structuur. XML biedt een mechanisme om beperkingen op te leggen op de opslag lay-out en logische structuur.

[Definitie: Een software module die een XML processor genoemd wordt, wordt gebruikt om XML documenten te lezen en biedt toegang tot hun inhoud en structuur.] [Definitie: Er wordt verondersteld dat een XML processor zijn werk doet voor een andere module, die de toepassing genoemd wordt.] Deze specificatie omschrijft het vereiste gedrag van een XML processor in termen van hoe het XML data moet lezen en de informatie die het moet aanbieden aan de toepassing.

De design doelen voor XML zijn:

- 1. XML zal overal algemeen bruikbaar zijn over het Internet.*
- 2. XML zal een grote variatie van toepassingen ondersteunen.*
- 3. XML zal compatibel zijn met SGML.*
- 4. Het zal gemakkelijk zijn om programma's te schrijven die XML documenten verwerken.*
- 5. Het aantal optionele eigenschappen in XML dient op het absolute minimum gehouden te worden, ideaal nul.*
- 6. XML documenten zouden door-de-mens-begrijpbaar en aanzienlijk duidelijk moeten zijn.*
- 7. Het XML design zou snel moeten voorbereid worden.*
- 8. Het design van XML zal formeel en consequent zijn.*
- 9. XML documenten zullen gemakkelijk te maken zijn.*

Bron Nederlands: <http://www.playgarden.com/w3c/TR/REC-xml/REC-xml.html>

Origineel Engels: <http://www.w3.org/TR/2004/REC-xml-20040204/>

Deze beschrijving is nogal technisch maar geeft wel aan met welk doel XML in eerste instantie ontwikkeld is. Een andere omschrijving is die van Wikipedia.org. Wikipedia.org is de grootste online encyclopedie waarbij de gebruikers invloed hebben op de correctheid van de informatie.

Wikipedia

Extensible Markup Language (XML) is een standaard voor het definiëren van formele markup-talen voor de representatie van gestructureerde gegevens in de vorm van platte tekst. Deze representatie is zowel machineleesbaar als leesbaar voor de mens.

Met andere woorden: XML is een bepaalde manier om gegevens gestructureerd vast te leggen (bijvoorbeeld in een bestand, maar ook voor bijvoorbeeld doorsturen van informatie over het internet). Deze manier is gedefinieerd en mag iedereen gebruiken. Het is ontworpen om zowel door een programma als door een mens leesbaar en schrijfbaar te zijn.

Bron: <http://nl.wikipedia.org/wiki/XML>

Deze omschrijving van XML komt overeen met mijn gedachten over XML en het gebruik hiervan. XML is net als HTML afgeleid van SGML, alleen in het geval van XML gaat het niet om presentatie van de data maar de structuur van de informatie.

De snelheid is niet direct van belang voor de applicatie, maar een te lange response tijd kan irriterend zijn voor de gebruikers. Om te voorkomen dat achteraf de snelheid onder de maat is, heb ik de snelheid bepaald van het inlezen en verwerken van XML. De snelheid heb ik in de beoordeling meegenomen.

Om de snelheid te kunnen bepalen, is gebruik gemaakt van simpleXML in PHP 5 (zie 5.2) en het boek "A Tale of Two Cities" van Charles Dickens. Dit boek is in het XML formaat geschreven en is 927 kB groot. De grootte van dit bestand en de structuur zijn van belang voor de snelheid. Om een goede indicatie te krijgen, zijn er meerdere tests uitgevoerd. Uit de test kwam naar voren dat het 50 maal openen, doorlopen en sluiten van het gehele boek en het tonen van alle hoofdstuk titels net onder de 10 seconden uit kwam. De test is uitgevoerd op een ontwikkel machine, in verhouding tot de uiteindelijke webserver geeft dit een vertekend beeld qua prestaties. De webserver zal niet alleen voor de KCC klantenmap gebruikt worden maar daarnaast zal hij een stuk sneller zijn in verhouding tot de ontwikkelmachine. Daarnaast is te verwachten dat de in te voeren XML bestanden maximaal 50 kB zullen zijn. Hierdoor zal het verwerken van de gegevens sneller verlopen dan in de test. Aan de hand van deze gegevens is het geloofwaardig om te concluderen dat de snelheid geen struikelblok zal zijn.

Behalve naar de snelheid, is er ook gekeken naar de mogelijkheden van XML. Omdat XML een free format hanteert zijn de mogelijkheden bijna eindeloos. Het gebruik van XML binnen PHP is onbeperkt en bovendien is het mogelijk om eigen functies te schrijven. Ik heb gekeken naar de beschikbare standaard functies binnen PHP, in paragraaf 5.2 ga ik hier verder op in.

5.1.2 Database

Voor de website is een database een mogelijke oplossing om de gegevens te bewaren. De database kent meerdere voordelen, de twee belangrijkste zijn: veiligheid en snelheid. Hoewel de snelheid in dit geval niet het belangrijkste is, bevat een database een sterk geoptimaliseerde manier van gegevensbeheer en opslag. De snelheid is bij een correct databaseontwerp zeer goed te noemen. Het DBMS beheert de toegang tot de gegevens, hierdoor kan de ontwikkelaar voorkomen dat meerdere gebruikers tegelijk dezelfde gegevens gaan wijzigen.

Een database kenmerkt zich door een statische structuur, voor de in ontwikkeling zijnde website is dat minder gewenst. Het zoeken is dan ook gericht op het vinden van een structurerend instrument met een acceptabele flexibiliteit.

5.1.4 Conclusie

Er is gekozen om de gegevens op te slaan in XML bestanden. Om de keuze te maken is er gebruik gemaakt van een beslissingstabel, deze tabel geeft de belangrijkste criteria weer en hier kan een gewicht aan gekoppeld worden. Tijdens het onderzoeken van de mogelijkheden is er een voorkeur naar voren gekomen, deze voorkeur samen met de beslissingstabel heeft tot de uiteindelijke keuze geleid.

Criteria	Gewicht	XML	Database
Moet gebruikt kunnen worden als free format database	9/63=0,143	9=1,286	3=0,429
Moet leesbaar zijn door mens en machine	6/63=0,095	7=0,667	3=0,286
Moet zichzelf kunnen beschrijven	8/63=0,127	9=1,143	3=0,381
Moet aanvaardbaar snel zijn	7/63=0,111	7=0,778	9=1,000
Moet zeer breed toepasbaar zijn	5/63=0,079	8=0,635	6=0,476
Moet de beschikking hebben over standaardfuncties	4/63=0,063	5=0,317	5=0,317
Moet veilig zijn	9/63=0,143	3=0,429	9=1,286
Moet eenvoudig van structuur zijn	5/63=0,079	9=0,714	6=0,476
Moet beperkte complexiteit hebben	2/63=0,032	6=0,190	5=0,159
Moet makkelijk onderhoudbaar zijn	8/63=0,127	5=0,635	8=1,016
Totaal	1,0	6,794	5,826

Figuur 2: Beslissingstabel opslagstructuur

5.2 Keuze voor programmeertaal

5.2.1 Wat is PHP

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

Bron: www.php.net

PHP valt onder de categorie scripting talen. Het is een Open-source project, dit zorgt ervoor dat er continue doorontwikkeld wordt. Ook kan de gebruiker eigen aanpassingen in de source aanbrengen. Het grote voordeel van PHP is het feit dat het wereldwijd veel gebruikt wordt waardoor er veel informatie te vinden is op het Internet. Ondanks het feit dat het niet als een volledige programmeertaal wordt beschouwd, zijn de mogelijkheden bijna onbeperkt. De ontwikkelaars zijn bezig met een Object Georiënteerde versie waardoor het steeds meer richting een volwassen programmeertaal gaat.

5.2.2 PHP 5 (beta)

De ontwikkeling van PHP volgt twee trajecten. De PHP 4 versie is de stabiele versie en wordt aanbevolen voor productie omgevingen, op dit moment is versie 4.3.7 de laatste stabiele versie. Deze versie wordt langzaam doorontwikkeld en alleen uitgebreid geteste onderdelen en functionaliteiten worden toegevoegd.

Daarnaast wordt er aan een nieuwe versie gewerkt, namelijk PHP 5.0. Deze bevat nieuwe functionaliteiten en grote veranderingen vergeleken met de stabiele versie. Ook bevat deze versie veel niet ontdekte bugs of nog niet opgeloste bugs. De aanwezige documentatie is beperkt en wordt minimaal bijgewerkt. Tevens kan een onderdeel volledig veranderen met de uitkomst van een nieuwe subversie (bv van Beta 3 naar Beta 4). Het spreekt dan ook voor zich dat deze versie afgeraden wordt om in een productie omgeving in te zetten.

PHP 5 beta heeft echter wel mogelijkheid om PEAR pakketten te installeren. PEAR staat voor “PHP Extension and Application Repository”. De site van PEAR (pear.php.net) stelt:

The purpose of PEAR is to provide:

- *A structured library of open-sourced code for PHP users*
- *A system for code distribution and package maintenance*
- *A standard style for code written in PHP*
- *The PHP Foundation Classes (PFC)*
- *The PHP Extension Community Library (PECL)*
- *A web site, mailing lists and download mirrors to support the PHP/PEAR community*

Op dit moment bestaat PEAR uit 228 pakketten. Een aantal pakketten zitten standaard bij PHP 5 (zoals PEAR DB, PEAR System, PEAR Mail) overige pakketten zijn te downloaden en te installeren. Een niet onbelangrijke aspect van PHP 5 is het onderdeel simpleXML.

SimpleXML is een XML parser. Het vergemakkelijkt het verwerken van een XML bestand. Het kan een XML bestand inlezen in het geheugen waardoor men de data kan benaderen als een object. Daarnaast bevat het een ingebouwde XML controle, die zorgt ervoor dat de bestanden alleen ingelezen worden indien deze correct XML zijn.

Naast het vereenvoudigde inlezen van bestanden ondersteunt het XPATH. XPATH is een zoek functie die reguliere expressie ondersteunt. De versie binnen simpleXML ondersteund een beperkte vorm van reguliere expressie.

Met simpleXML is het ook mogelijk om het gehele object terug te schrijven naar een XML bestand.

5.2.4 De hulpmiddelen DOM en XSL

Er zijn enkele hulpmiddelen gescreend voor het inlezen van XML bestanden en lay-out beschrijving die compatible zijn met PHP.

DOM staat voor Document Object Model en is een hulpmiddel om XML in te lezen in het geheugen. DOM lijkt op veel punten op simpleXML. De support in PHP 5 met betrekking tot DOM is minimaal. SimpleXML overklast op bijna alle benodigde onderdelen DOM. Alleen het toevoegen van een nieuw element ondersteunt het niet, iets wat DOM wel doet.

XSL staat voor Extensible Stylesheet Language en beschrijft hoe de gepresenteerde data eruit moet komen te zien. XSL bestaat uit twee onderdelen namelijk: XSL transformations (XSLT) en XSL Formatting Objects (XSL-FO).

XSLT is een transformatie taal die speciaal ontwikkeld om XML te transformeren. De meest gebruikte transformatie is die van XML naar HTML. Aan de hand van het XSL bestand wordt de uiteindelijke opmaak bepaald. Aangezien er gebruik gemaakt wordt van templates om de lay-out te bepalen is het gebruiken van XSLT niet nodig. XSL-FO is de voorganger van XSLT.

5.2.5 Onderbouwing keuze

Ik heb gekozen om PHP 5 te gaan gebruiken, ondanks de mogelijke risico's die een beta release met zich mee brengt. Tijdens de afstudeerperiode is Release Candidate 2 uitgebracht, met deze versie is PHP 5.0 op een stabiele versie aangekomen. De keuze is gemaakt op de aanwezigheid van simpleXML, hierdoor zijn XML bestanden eenvoudig als object aan te spreken. Het zorgt ervoor dat men geen eigen parser hoeft te schrijven of een ander standaard pakket te zoeken dat dezelfde functionaliteiten bevat als simpleXML. De risico's van PHP 5 zijn aanzienlijk, bij elke nieuwe subversie kunnen onderdelen aangepast of verwijderd zijn zonder dat dit van tevoren is aangekondigd. De meeste aanpassingen zijn overigens wel terug te vinden in de changelog.

Er is voor gekozen om geen gebruik te maken van DOM. Ook wordt voor de lay-out bepaling geen gebruik gemaakt van XSL en XSLT. Hiervoor is de keuze gevallen op het gebruik van templates (zie 5.3).

De hier besproken keuze is mede tot stand gekomen met behulp van onderstaande beslissingstabel.

Criteria	Gewicht	PHP 4	PHP 5
Moet een stabiele structuur hebben	7/48=0,15	8=1.20	5=0.65
Moet een goede pers hebben	8/48=0,17	8=1.36	8=1.36
Moet zeer veel functionaliteiten bevatten	9/48=0,19	6=1.14	8=1.52
Moet Pear pakketten kunnen installeren	6/48=0,13	0=0	10=1.30
Moet goed gedocumenteerd zijn	8/48=0,17	9=1.53	4=0.68
Moet XML bestanden kunnen inlezen	10/48=0.21	0=0	10=2.1
Totaal:	48/48=1.0	5.23	7.61

Figuur 3: Beslissingstabel PHP

5.3 Template engines

5.3.1 Wat zijn het?

Een template engine is een stukje software dat ervoor zorgt dat de logica en de opmaak van een pagina gescheiden gehouden kan worden. Alle opmaak onderdelen staan in de template, de logica staat in aparte bestanden. De template engine zorgt ervoor dat de uitvoer van de logica in de template terecht komt waarna deze getoond wordt. De template bevat speciale tags welke uiteindelijk vervangen zullen worden door de uitvoer van het logica gedeelte.

Het grote voordeel van template engines is de duidelijke scheiding tussen logica en opmaak. Dit zorgt ervoor dat een grafisch ontwerper of web designer de template kan maken en een programmeur de logica. Een nadeel van sommige template engines is dat ze te veel logica toe staan in de template.

5.3.3 Welke heb ik getest?

Er zijn veel verschillende template engines beschikbaar voor PHP. Naast de standaard engines is het mogelijk een eigen template engine te schrijven. Het grote voordeel van een eigen geschreven template engine is dat de functionaliteiten volledig naar eigen smaak kunnen worden ingericht.

Ik heb totaal 4 template engines geprobeerd en bekeken, over het algemeen is de werking hetzelfde alleen de functionaliteiten zijn verschillend. Om aan testvarianten te komen heb ik op een breed forum gezocht naar de populairste template engines.

De volgende engines zijn het meest in gebruik.

- Savant
- PatTemplate
- Smarty
- Templatepower

PatTemplate

PatTemplate is een template engine die is ontwikkeld voor PHP 4. Het is een engine met veel functionaliteiten en goede documentatie. Volgens de documentatie is het een van de snelste engines, daarnaast zijn de functionaliteiten bijna onbeperkt. Tijdens het testen van de engine bleek hij niet compatible te zijn met PHP 5.

Templatepower

Templatepower is een template engine welke eenvoudig in het gebruik is en redelijk veel functionaliteiten ondersteunt. Er zijn drie verschillende versie's beschikbaar verschillend van eenvoudig, zeer uitgebreid en een versie met database ondersteuning. De snelheid is goed en de templates zijn snel een eenvoudig samen te stellen. De ondersteuning van herhalende blokken is een groot voordeel. Het gebruik van Templatepower is zeer eenvoudig in het gebruik en de templates zijn redelijk overzichtelijk.

Smarty

De reviews op Internet zijn erg positief over Smarty. Alles is mogelijk, hij is snel, goed en duidelijk gedocumenteerd. Het nadeel is de complexiteit van het gebruik. De syntax is complex en de templates zijn niet overzichtelijk. Smarty heeft een overload aan functionaliteiten.

Savant

Savant is een template engine die is geënt op Smarty maar zonder de hoeveelheid overload die Smarty heeft. Daarnaast zijn de templates van Savant gebaseerd op PHP code. Hierdoor zijn er geen aparte tags nodig binnen een template. Tevens is Savant hierdoor compatible met de toekomstige versies van PHP. De mogelijkheden zijn uitgebreid, mede dankzij het gebruik van PHP in de template. Savant maakt gebruik van aparte plugins, dit zorgt ervoor dat de standaard versie geen onnodige onderdelen bevat. Savant maakt gebruik van PEAR (zie 5.2.2), dit is geen nadeel omdat we PEAR ook gebruiken voor de database verbinding (zie 5.5).

5.3.4 Onderbouwing van de template keuze

Zoals uit de vorige paragraaf is gebleken, zijn er vier template engines bekeken en getest. Smarty (lastig in gebruik) en PatTemplate (niet compatible met PHP 5) vallen af. Overgebleven waren Savant en Templatepower. Uiteindelijk is er gekozen voor Savant. Vooral het gebruik van PHP in de template is een groot voordeel. De mogelijkheden van Templatepower en Savant zijn in grote lijnen gelijk. Savant ondersteunt een beter gebruik van geneste blokken en tevens is het mogelijk om de template elementen te controleren op inhoud. Een nadeel hieraan is dat de templates met een overkill aan logica zouden kunnen worden belast.

Om mijn keuze te onderbouwen heb ik gebruik gemaakt van een beslissingstabel. In deze tabel kwam direct naar voren dat PatTemplate en Smarty geen keuze mochten zijn. Savant kwam als beste naar voren. Het voldoet het beste aan de gestelde criteria.

Criteria	Wegingsfactor	PatTemplate	TemplatePower	Smarty	Savant
Functionaliteiten moeten naar eigen hand gezet kunnen worden	6/82=0.072	4=0.288	4=0.288	6=0.432	6=0.432
Moet goed gedocumenteerd zijn	7/82=0.085	7=0.595	7=0.595	9=0.765	8=0.68
Moet brede functionaliteit hebben	8/82=0.098	6=0.588	7=0.686	9=0.882	8=0.784
Moet compatible zijn met de gehanteerde programmeertaal	10/82=0.122	0=0	5=0.61	5=0.61	5=0.61
Moet eenvoudig zijn in gebruik	9/82=0.110	3=0.33	7=0.77	1=0.11	9=0.99
Moet aanvaardbaar snel zijn	6/82=0.073	7=0.511	7=0.511	6=0.438	8=0.584
Moet herhalende blokken ondersteunen	10/82=0.122	5=0.61	5=0.61	5=0.61	5=0.61
Moet overzichtelijke templates leveren	10/82=0.122	7=0.854	7=0.854	3=0.366	8=0.976
Moet een simpele syntax hebben	8/82=0.098	6=0.588	6=0.588	3=0.294	7=0.686
Moet geen overbodige rompslomp hebben	8/82=0.098	5=0.49	5=0.49	3=0.294	5=0.49
Totaal:	82/82=1.0	4.854	6.002	4.801	6.842

Figuur 4: Beslissingstabel template engines

5.4 Rechtensysteem

Het systeem zal verschillende soorten gebruikers gaan dienen. Onder de gebruikers zullen in de toekomst ook de klanten van KCC vallen. Om te voorkomen dat gebruikers gegevens te zien krijgen die vertrouwelijk zijn, moet het gebruik van rechten voorbereid worden. Omdat het nog niet bekend is welke rechten een klant exact krijgt, is er besloten wel rekening te houden met een rechtensysteem in het ontwerp maar dat nog niet te implementeren. Om een correct ontwerp te maken heb ik gezocht naar de verschillende manieren om rechten te beheren, ik heb gekeken naar de mogelijkheden van een gratis standaardpakket en de mogelijkheden van zelfontwikkeling.

Op het internet is een goed standaardpakket aanwezig (PatUser), dit is van dezelfde makers als PatTemplate. Helaas is dit pakket nog niet compatible met PHP 5. Andere standaardpakketten zijn niet gratis, bevatten te veel functionaliteiten of zijn te lastig te implementeren. De keuze is dan ook gevallen op het ontwikkelen van een eigen rechtensysteem dat wel compatible is met PHP 5.

Voor de ontwikkeling van het rechtensysteem is er gebruik gemaakt van bitwise operatoren. Dat wil zeggen dat de rechten op een binaire manier gedefinieerd worden.

5.5 Pear DB

PEAR DB is een module die standaard bij PEAR (zie 5.2.2) geleverd wordt. Ze stelt de programmeur in staat om DBMS onafhankelijke programmatuur te schrijven. Korton Internet B.V. gebruikt voor grote klanten het DBMS Microsoft SQL server, daarnaast gebruiken ze voor de kleinere klanten Microsoft Access databases. Aan het begin van de stageopdracht was nog niet zeker of de website gebruik gaat maken van de Microsoft SQL server of dat er gekozen gaat worden voor een MySQL Server. Dankzij Pear DB geeft dit geen problemen tijdens het opzetten van de productieomgeving aan het eind van de afstudeeropdracht. PEAR DB heeft een eenmalige instelling waarin staat met welke host er verbinding gemaakt wordt en wat voor DBMS dat is. Overal in de website kan er gebruik gemaakt worden van de standaard PEAR DB functies om de data te verwerken.

5.6 Veilig inloggen

De website zal vertrouwelijke informatie gaan bevatten. Daarom moet hij goed afgeschermd zijn tegen ongewenst gebruik. Om dit te kunnen uitvoeren is onderzocht welke mogelijkheden er binnen PHP zijn. Het meest gebruikte variant is een database met de gegevens van de gebruiker in combinatie met sessies en cookies om te controleren of een gebruiker nog steeds ingelogd is. Bij deze gegevens wordt naast gebruikersID en ipadres ook een token bewaard. Dit token staat zowel in de cookie als in de database en zorgt ervoor dat de cookie niet nagemaakt kan worden. De gebruikersnaam en wachtwoord worden uit veiligheid niet opgeslagen in de cookie.

Het gebruiken van cookies houdt een veiligheids risico in. Een cookie zorgt ervoor dat de laatste login gegevens bewaard blijven. Wanneer een medewerker vanaf een klant zijn pc inlogt op de website zal deze login bewaard blijven. Wanneer de klant daarna naar de website suft zal hij automatisch ingelogd worden onder de naam van de medewerker. Dit zorgt ervoor dat personen zonder toegang "ineens" toegang krijgen.

Een inlog systeem zonder cookies maakt het geheel al veiliger, in combinatie met het gebruiken van een sessie wordt de veiligheid nog groter. Een sessie is een speciaal soort cookie. Het kan gegevens bewaren en zal deze gegevens altijd vernietigen wanneer de webbrowser gesloten wordt. Daarnaast is de levensduur van de sessie instelbaar.

Om de beveiliging verder te verhogen is er besloten gebruik te maken van HTTPS verbindingen. HTTPS is een HTTP verbinding via SSL. Alle data die door deze verbinding gaan wordt geencrypt door middel van een private key. Deze key is alleen op de webserver bekend. Decryptie gebeurt door middel van een sessie key, deze is per sessie verschillend en wordt uitgegeven aan de hand van een certificaat. De encryptie kan verschillen van 40 bit, 128 bit of 256 bit. Het kraken van deze encryptie door middel van brute kracht kost maanden

HTTPS is een techniek die veel op Internet gebruikt wordt om (credit card) betalingen te verwerken, grote banken zoals de Rabobank gebruiken het voor hun Internet Bankieren.

De HTTPS support zal worden ingeschakeld op het moment dat de applicatie naar de productieomgeving verhuist.

5.7 Locken van XML bestanden

Om te voorkomen dat er meerdere gebruikers tegelijk dezelfde gegevens gaan bewerken is een locking mechanisme bedacht. Dit mechanisme houdt in een database bij welke bestanden in gebruik zijn. Hierdoor kan het systeem zien wanneer een gebruiker wel of niet mag bewerken. Er is gekozen voor een database omdat op die manier de gegevens met hoge snelheid te bereiken zijn. Daarnaast wordt de locking van deze gegevens door het DBMS afgehandeld. Een andere mogelijkheid is om een extra bestand in de directory te zetten met dezelfde naam en een andere extensie dan het gelockte bestand. Om te voorkomen dat een lock ongewenst lang blijft staan worden bij elke hit alle locks van de gebruiker verwijderd. Wanneer het systeem elke aanwezige klant directory moet langsgaan om te controleren op lock bestanden zorgt dit voor een grote vertraging binnen het systeem. Nog een andere mogelijkheid is het bewaren van alle aanwezige locks in één XML bestand. Het probleem bij deze mogelijkheid is dat twee gebruikers tegelijkertijd het bestand kunnen bijwerken waardoor er een lock verdwijnt.

6. Ontwerp

In dit hoofdstuk is beschreven welke onderdelen ik ontworpen heb en welke keuzes daarbij gemaakt zijn. Het vooronderzoek zoals beschreven in hoofdstuk 5 heeft de bouwstenen van het ontwerp aangedragen. Als eerste wordt het databaseontwerp besproken, daarna volgt het ontwerp in Fusebox.

6.1 Databaseontwerp

Voor het bijhouden van de data is gekozen voor het gebruik van XML bestanden. Desalniettemin zullen er een aantal onderdelen in de database bijgehouden worden. Dit zijn:

- Gebruikers
- Gebruikersgroepen
- Rechten
- Klanten
- Locks
- Todo's
- Prioriteiten

De gebruikers tezamen met hun gebruikersgroep en de bijbehorende rechten worden bijgehouden om te zorgen dat de gebruikers kunnen inloggen. Deze gegevens staan in een database vanwege de eenvoud en snelheid. De drie tabellen zijn aan elkaar gekoppeld door middel van een vreemde sleutel. Tussen de rechten en de gebruikersgroepen zit geen directe koppeling, een gebruikersgroep kan meerdere rechten bezitten en een recht kan bij meerdere groepen horen. Zoals te lezen is in 5.4 is gekozen voor een manier die gebruik maakt van binaire waarden van een geheel getal. Door deze koppeling zijn snel de rechten te controleren en is er geen koppeltabel nodig.

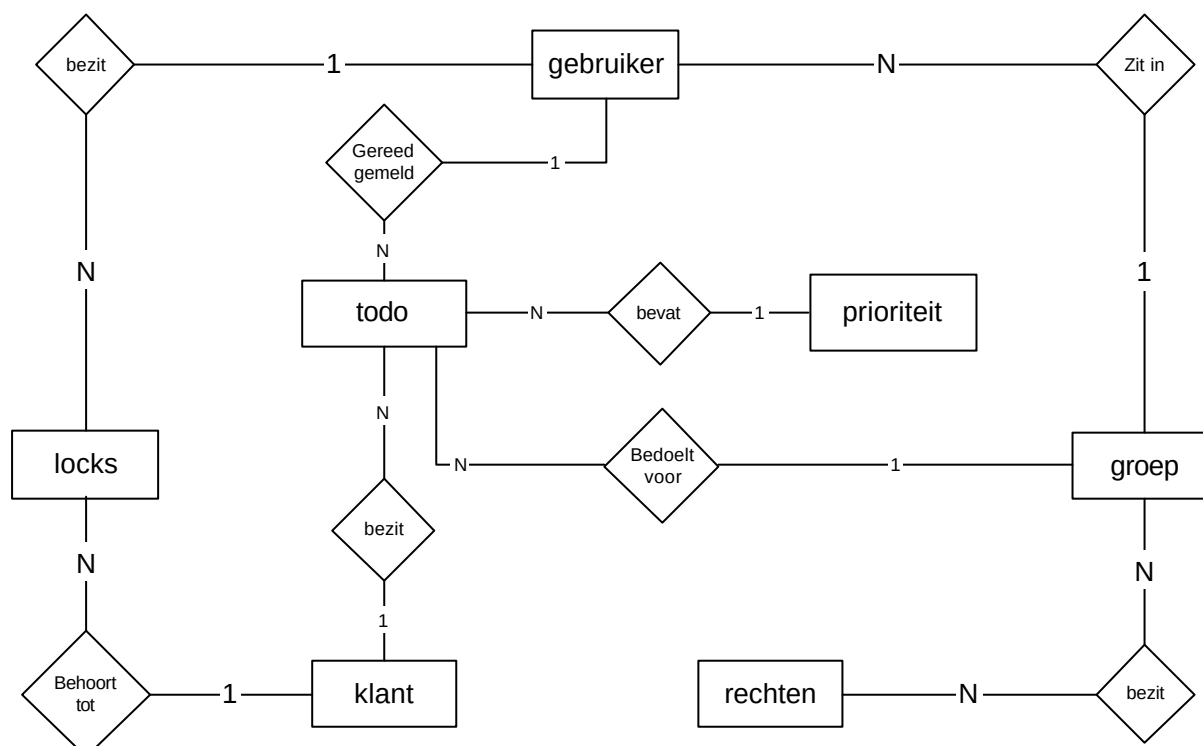
De gebruikerstabel bevat naast de inlognaam en wachtwoord ook de volledige naam van de gebruiker. Ook geeft de tabel aan in welke gebruikersgroep de gebruiker zit en of de gebruiker geblokkeerd is. Hierdoor kan gemakkelijk de toegang worden onzegd zonder problemen met de referentiele integriteit te krijgen.

Naast de gebruikers worden ook de klanten in de database bijgehouden. Hierdoor is het snel te identificeren welke klanten er bekend zijn in het systeem en waar de gegevensbestanden staan. Tevens kunnen klantnamen meerder verschillende tekens bevatten zoals ë, ï, â etc. Per klant wordt er ook bijgehouden welke gebruiker er als laatste een aanpassing gemaakt heeft.

Om te voorkomen dat meerdere gebruikers tegelijk dezelfde gegevens gaan wijzigen is er een lock systeem ontwikkeld. Dit systeem maakt gebruik van een database om bij te houden welke bestanden gelockt zijn. Hier wordt bijgehouden welk detail bestand van een klant in gebruik is. De gebruiker wordt bijgehouden zodat er gecontroleerd kan worden welke gebruiker de lock bezit.

Een van de onderdelen die later aan de database toegevoegd zijn, zijn de todo's. Deze zouden aanvankelijk in een XML bestand opgeslagen worden net als de rest van de detail gegevens. Omdat de verwachting is dat de todo's niet van structuur zullen wijzigen en dat ze veel gebruikt zullen worden is er alsnog voor een database gekozen. Tevens is het maken van een filter eenvoudiger, dit filter zorgt ervoor dat er snel een selectie gemaakt kan worden van de aanwezige todo's. Een todo heeft een bepaalde prioriteit, aan de hand van deze prioriteit kan een KCC medewerker bepalen welke todo het belangrijkste is. De prioriteiten staan in een aparte tabel zodat er later eenvoudig een extra prioriteit toegevoegd kan worden zonder de code te hoeven aanpassen. De tabel todo staat gekoppeld met gebruikersgroepen, gebruikers, klanten en prioriteiten.

Een totaal overzicht van de database bevindt zich in het ERD (zie figuur 5).



Figuur 5: ERD

Locks	Klanten	Gebruiker	Groepen
ID	ID	ID	ID
gebruikerId	naam	gebruikersnaam	naam
bestand	datumTijdWijziging	wachtwoordMD5	rechten
datumTijd	directory	naam	
klantId	gebruikerID	groepId	
		geblokkeerd	

Todo	Prioriteit	Rechten
ID	ID	ID
klantID	omschrijving	omschrijving
omschrijving		
prioriteit		
aanmaakdatum		
gereeddatum		
verloopdatum		
gereed		
groepID		
oplossing		
gebruikerID		

Figuur 6: Toelichting ERD

6.2 Fusebox ontwerp met Adalon

6.2.1 Beknopte uitleg Fusebox

De Fusebox methode werkt als het ware als een zekeringen kast in een huis. Op het moment dat er één zekering uitvalt blijven de andere hun werk doen. Deze metafoor is doorgetrokken op de applicatie, op het moment dat een gedeelte niet meer werkt zal dit geen invloed hebben op de werking van de andere gedeelten.

Fusebox bevat uit een aantal onderdelen die hierna kort worden toegelicht omdat ze van essentieel belang zijn voor het ontwerp.

- Circuits
- Fuseactions
- Switch
- Fuses
- XFA's

Daarnaast zijn er nog een enkele onderdelen binnen Fusebox die voor het ontwerp niet van belang zijn:

- Settingsfile (fbx_settings)
- layoutfile (fbx_Layouts)

Circuits

Een circuit is een directory binnen de applicatie die zo is ontworpen dat deze een specifieke functie uitvoert. Een voorbeeld is een onderdeel "webwinkel" op een website. Deze handelt alles met betrekking tot de webwinkel af. Een ander circuit is verantwoordelijk voor de nieuwsberichten op (dezelfde) site. De aanwezige circuits staan in het fbx_circuits bestand.

Fuseactions

Een fuseaction roept een aantal fuses aan binnen de switch. Een fuseaction is te vergelijken met een method. Een voorbeeld van een fuseaction is:

```
Index.php?fuseaction=toonGebruikers
```

Het is ook mogelijk om een fuseaction van een ander circuit aan te roepen. Bijvoorbeeld:

```
Index.php?fuseaction=forum.toonGebruikers
```

XFA's

Op elke plek in een fuse waar een fuseaction aangeroepen wordt, moet de fuseaction een variabele zijn. Om dit te bereiken worden er in de switch XFA's gedefinieerd. XFA staat voor eXitFuseAction en deze worden als volgt gedefinieerd in de switch.

```
switch($Fusebox["fuseaction"]) {  
  
    case ("toonGebruikers") :  
        $XFA["gebruiker_wijzigen" = "forum.wijzigGebruiker"  
        include ("qry_gebruikers.php");  
        include ("act_gebruikers.PHP");  
        include ("dsp_gebruikers.PHP");  
        break;  
}
```

Op dit moment is de XFA 'gebruiker_wijzigen' bekend binnen de fuseaction en zal deze verwijzen naar de fuseaction 'wijzigGebruiker' in het circuit 'forum'. Deze XFA is te verwerken in bijvoorbeeld een link op de pagina.

```
echo "<a href='".$GLOBALS['self']."'?fuseaction="  
    . $XFA["gebruiker_wijzigen"].">Wijzig gebruiker</a>"
```

PHP zet deze regel om naar HTML en dat ziet er dan zo uit:

```
<a href='index.php?fuseaction=forum.wijzigGebruiker'>Wijzig  
gebruiker</a>
```

Het grote voordeel hiervan is het centraal beheren van alle XFA's hierdoor is een XFA op meerdere locaties te gebruiken.

Switch

Elk circuit heeft zijn eigen switch, in deze switch staat beschreven welke fuses er bij een fuseaction uitgevoerd moeten worden. Tevens staan hierin de XFA's gedefinieerd. Een deel van een switch ziet er als volgt uit:

```
switch($Fusebox["fuseaction"]) {  
    case ("toonGebruikers") :  
        include ("qry_gebruikers.php");  
        include ("act_gebruikers.php");  
        include ("dsp_gebruikers.php");  
        break;  
}
```

Fuses

Een fuse bevat de uiteindelijk uit te voeren code. Het belangrijkste van een fuse is het meervoudige gebruik. Er zijn drie verschillende soorten fuses. Afhankelijk van hun taak krijgen ze de volgende prefix:

- *qry*: een 'query' file. Deze voert een query uit. De uitvoer van de query wordt in een action file afgehandeld. Een voorbeeld is het ophalen van de gebruikers. Een action kan deze dan verder verwerken.
- *act*: een 'action' file. Deze voert een of meerdere acties uit zonder output te geven naar de gebruiker. Een voorbeeld is het bijwerken van de gegevens op de harddisk.
- *dsp*: een 'display' file. Deze toont de gegevens uiteindelijk op het scherm. Daarnaast kan hij gebruikt worden om informatie aan de gebruiker te vragen. Indien er gebruik wordt gemaakt van template's zullen deze hier gevuld worden.

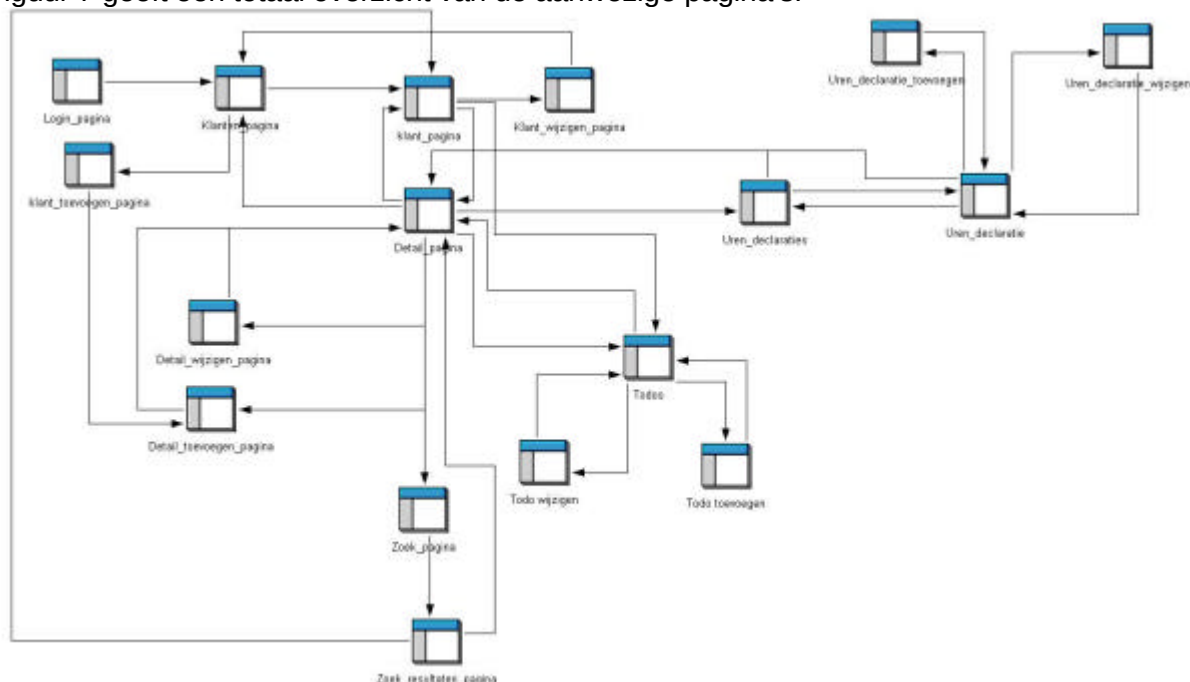
6.2.2 Wat is Adalon?

Adalon is een case-tool voor Fusebox. Met Adalon is gemakkelijk de Fusebox structuur ontwikkelen en direct het te bekijken. Het is mogelijk om circuits te definiëren en fuseactions te maken, aan deze fuseactions zijn fuses te koppelen en XFA's te definiëren. De HTML output stelt in staat om de structuur van de site te bekijken zonder dat één woord aan code hoeft te worden geschreven. Deze output kan dan als uitgangspunt gebruikt worden in een overleg met de opdrachtgever.

6.2.3 Ontwerp

Het uiteindelijke Adalon ontwerp is ontstaan door diverse deelontwerpen te maken. Het eerste deelontwerp was de algemene structuur. Dit ontwerp geeft alle pagina's voor de uiteindelijke website. Aan de hand van dit ontwerp zijn de links op een pagina bepaald. Tijdens deze ontwikkeling werd het gehele ontwerp erg onduidelijk door alle aanwezige links tussen de pagina's. Daarom is besloten om niet alle links in het ontwerp te zetten. Zo is het bijvoorbeeld mogelijk om vanaf elke pagina uit te loggen en daarmee terug te keren naar de inlog pagina. Deze links zijn niet in het ontwerp meegenomen. Tevens zijn er andere links niet in het ontwerp gezet, terwijl deze wel in de applicatie komen. In de uitgewerkte deelontwerpen zijn de links wel terug te vinden. Tevens is besloten om het commentaar bij elke link te verwijderen zodat het geheel beter leesbaar werd. Als resultaat is er een diagram waarin alle te maken pagina's staan evenals een aantal van de aanwezige koppelingen tussen de pagina's. Aan de hand van dit ontwerp konden de afzonderlijke delen ontworpen worden.

Figuur 7 geeft een totaal overzicht van de aanwezige pagina's.



Figuur 7: Structuur overzicht

Aan de hand van de algemene structuur is begonnen met het uitwerken van het inlog gedeelte. Het inlog gedeelte is de kleinste van de gehele site qua structuur. Hierdoor kon ik de werking van Adalon en de opbouw van Fusebox beter onder de knie krijgen.

Inlog pagina

De *inlogpagina* handelt het inloggen af. Het bestaat uit één pagina en twee fuseactions. Eén fuseaction zorgt ervoor dat de pagina op het scherm komt en de andere controleert de inloggegevens van de gebruiker. De pagina heeft twee XFA's, één wanneer de inlog gegevens correct zijn en één wanneer deze incorrect zijn. Aan de hand van deze XFA wordt uiteindelijk de gebruiker naar de juiste pagina doorgestuurd.

Klanten

Op de *klantenpagina* komen alle aanwezige klanten te staan. De gebruiker kan altijd naar deze pagina gaan om een andere klant te kiezen. Wanneer een gebruiker een klant kiest zal hij de betreffende *klantpagina* te zien krijgen. Net als de *inlogpagina* bestaat de *klantenpagina* uit één pagina en meerdere fuseactions. Vanaf de *klantenpagina* kan de gebruiker de volgende acties ondernemen: zoeken, uitloggen, klant toevoegen en klant kiezen.

Klant

De *klantpagina* is een afgeleide van een *detailpagina*. Op de *klantpagina* wordt het detail algemeen getoond. Hiervoor is gekozen opdat elke klant altijd minstens één *detailpagina* bevat. Vanaf de *klantpagina* is het mogelijk *detailpagina's* te bekijken. Tevens kan een klant gewijzigd worden, dit wijzigen heeft alleen betrekking op de naam in de database.

Detail

De *detailpagina* toont de aanwezige detail gegevens van een detail bestand. Er kunnen meerdere detail bestanden aanwezig zijn, de werking is per bestand identiek. Het ontwerp hoeft hierdoor geen rekening te houden met welk bestand getoond wordt. De gebruiker kan een detail bekijken, wijzigen of toevoegen. KCC staat het niet toe dat gebruikers gegevens kunnen verwijderen, hierdoor is het ook niet mogelijk om een detail te verwijderen.

Todo

De todo's zijn later toegevoegd aan het ontwerp. In eerste instantie zouden de todo's een *detailpagina* worden, dit is later gewijzigd naar een apart onderdeel. De eerste pagina waarop de gebruiker komt is de *todopagina*, hierop worden alle todo's getoond en kunnen ze gereed gemeld worden. Op de *todo wijzigen pagina* kan een todo gewijzigd worden en op de *todo toevoegen pagina* kan een todo toegevoegd worden.

Urendecclaratie

Net als de todo's zijn de urendecclaraties losgekoppeld van de detail gegevens. De rede van loskoppeling is het gebruik van een aparte lay-out en de beperkte mogelijkheden tot wijzigen. Zodra een urendecclaratie gereed gemeld is, mag deze in het systeem niet meer gewijzigd worden. De todo's en urendecclaratie zijn qua werking gelijk aan elkaar. Achter de schermen zit het werkelijke verschil tussen de beide onderdelen.

Zoeken

Het deelontwerp van zoeken is eenvoudig van opbouw. Eén pagina om de gegevens in te voeren en één pagina waar de resultaten op komen te staan. Vanaf de resultaten pagina is het mogelijk om direct naar de *klantpagina* of naar het *detailpagina* toe te gaan.

Alle diagrammen zijn terug te vinden in hoofdstuk 5 van de externe bijlage.

6.3 Pilot ontwerp

In het pilot ontwerp zijn de mogelijke acties van het systeem beschreven. Per actie staan de verwachte stappen die uitgevoerd moeten worden. Daarnaast staan er eventuele aannames genoteerd. Een algemene aanname is dat de gebruiker ingelogd is. Tevens worden de uitzonderingen genoemd en de verwachte uitvoer. Tijdens het ontwikkelen zijn zoveel mogelijk de onderdelen uitgesplitst in losse functies.

Tijdens de implementatie kan het pilot ontwerp worden geraadpleegd om te controleren of het onderdeel aan de verwachtingen voldoet of indien er een fout optreedt, die fout ook correct wordt afgehandeld. Tevens vormt het pilot ontwerp een goede basis voor het testen.

Een voorbeeld van een pilot omschrijving.

Inloggen	
Omschrijving:	De gebruiker logt in op het systeem
Procedure:	1) Gebruiker voert zijn gebruikersnaam en wachtwoord in 2) De gebruiker drukt op de knop inloggen 3) Het systeem controleert de volgende gegevens: Gebruikersnaam, wachtwoord, vergrendeld en groep. <i>Bij correcte gegevens</i> 4) Het systeem maakt de sessie aan 5) Het systeem zet de gegevens in de sessie Het systeem stuurt de gebruiker door naar de klanten pagina indien deze geen klant is. <i>Bij incorrecte gegevens</i> 4) Het systeem stuurt de gebruiker terug naar de inlog pagina
Aannames:	Geen
Uitzonderingen:	De gebruiker heeft foutieve gegevens ingevoerd of is geblokkeerd.
Resultaat:	De gebruiker is ingelogd.

Aan de hand van deze beschrijving kan bij de implementatie gekeken worden naar de uit te voeren stappen.

6.4 Schermontwerpen

Voor er met de werkelijke implementatie begonnen kon worden, is er prototyping toegepast. De karakteristieke kenmerken van prototyping zijn:

- *Prototyping actually works*
- *Built quickly and cheaply*
- *Subset of requirements*
- *Integral to life cycle; includes evaluation and modification*
- *Purpose is risk reduction/improved quality/knowledge increase*

Bron: http://www.cc.gatech.edu/classes/AY2003/cs6300_fall/slides/prototyping/sld002.htm

De schermontwerpen zijn immers een eerste opzet en worden tijdens de ontwikkeling van de website verder aangepast zodat het geheel beter op elkaar afgestemd raakt. Het ontwikkelen van een eerste opzet zorgt voor een duidelijker overzicht van de onderdelen die op het scherm moeten komen. Naar mate de website duidelijker vormen aan neemt, zal het uiterlijk worden aangepast aan de bruikbaarheid en leesbaarheid en tevens aan de huisstijl van Korton.

6.5 Structuur ontwerp (XML bestanden)

Er is gekozen voor XML bestanden als opslagmedium. Zoals eerder vermeld in hoofdstuk 5, zijn XML bestanden erg flexibel te gebruiken. De structuur van de XML bestanden is van belang voor de optimale werking. Daarom is gekozen voor een eenvoudige platte structuur. De bestanden zijn door elk persoon met geringe kennis te lezen zonder aparte software. Dankzij de eenvoudige opzet is het niet direct nodig om een editor voor de bestanden te ontwikkelen, ze zijn gemakkelijk in een teksteditor te lezen of te wijzigen. Er worden twee bestanden gebruikt. In het bron bestand zijn de data beschreven alsmede alle velden van één detail onderdeel. Er staat bijvoorbeeld omschreven hoe de velden naam, adres, woonplaats, postcode en telefoonnummer eruitzien. Daarnaast is er een doel bestand, dit bestand bevat de data aan de hand van het bron bestand. Aansluitend op het vorige voorbeeld zou het doel bestand de volgende gegevens kunnen bevatten. Marcel Stokman, Jadelaan 99, Hoofddorp, 2123XZ, 023-5555060. Deze zijn dan in het XML formaat.

De volledige structuur van het bron bestand is als volgt.

- *input*, een input is één record.
 - o *id*, een unieke combinatie van letters met als doel het record te identificeren.
 - o *text*, beschrijving van een kritisch kenmerk van een record. Bijvoorbeeld: naam.
 - o *hint*, specificatie van een record.
 - o *type*, specificatie van het soort record. Dit zijn: text, textarea, pulldown
 - o *size*, het maximaal aantal tekens dat een text record mag omvatten.
 - o *cols*, het maximaal aantal kolommen dat een texarea record groot is.
 - o *rows*, het maximaal aantal rijen dat een texarea record groot is.
 - o *values*, bepaalt de mogelijke keuzes binnen een pull down record. Elke keuze wordt gedefinieerd in een *val*.
 - o *val*, specificatie van een waarde binnen een pull down record.
 - o *required*, geeft aan of de invoer van een record verplicht is of niet. De mogelijke waarden zijn *y* of *n*.
 - o *defaultvalue*, geeft de advies waarde van een record aan.
 - o *displayonly*, geeft aan dat een record alleen getoond mag worden.
 - o *validation*, geeft aan welke invoer controle('s) er uitgevoerd moet worden op een record.
 - *cannotBeBlank*: veld mag niet leeg zijn.
 - *mustBeDate*: het veld moet datum formaat zijn.
 - *mustBeEmail*: het veld moet e-mail formaat zijn.
 - *cannotContainAlfa*: het veld mag geen letters bevatten.
 - *cannotContainNumber*: het veld mag geen letters bevatten.
 - *cannotContainSpecial*: het veld mag bepaalde speciale tekens niet bevatten.

Het bestand dat de werkelijke gegevens bevat bestaat uit de volgende onderdelen.

- *output*, een output is een blok met alle gegevens van het bron bestand.
 - o *Identifier*, een *identifier* identificeert het output blok. Er kunnen meerdere output blokken voorkomen in het bestand de *identifier* zorgt voor het verschil tussen de blokken
 - o *Antwoord*, een antwoord blok met daarin de waarde van één *input* blok.
 - *id*, de unieke id van het record.
 - *text*, de ingevoerde waarde

7. Implementatie

Dit hoofdstuk beschrijft de aanpak en het verloop van de implementatie van de website in PHP. Elk onderdeel krijgt aparte behandeling waarbij zoveel mogelijk de gemaakte keuzes gemotiveerd en beargumenteerd zijn.

7.1 Opzetten structuur

Alvorens te beginnen met de implementatie van de onderdelen heb ik eerst de structuur opgezet. Daartoe is de source code generator van Adalon gebruikt. Deze genereert de belangrijkste bestanden en samen met de core Fusebox files levert dit de basis voor de structuur. Tijdens de implementatie zullen er nog aanpassingen gemaakt worden in deze bestanden zoals het toevoegen van instellingen in de fbx_settings file en het uitbreiden van de XFA's in de fbx_switch files.

7.2 Inlogpagina

De inlog pagina is de eerste die de gebruikers te zien krijgen, het is dan ook meest logisch om hiermee te beginnen. De pagina zelf is eenvoudig opgebouwd, de werking achter de schermen was een stuk lastiger om te ontwikkelen. Er zijn een aantal stappen die uitgevoerd moeten worden waarna de gebruiker op de klanten pagina uit komt in het geval zijn gebruikersnaam en wachtwoord correct zijn of op de inlog pagina blijft indien deze incorrect zijn. De volgende stappen worden achter de schermen uitgevoerd:

- Het versturen van de ingevoerde gegevens naar de server
- Het maken van een database verbinding
- Het ophalen van de gegevens uit de database aan de hand van de ingevoerde gegevens
- Het aantal resultaten controleren
- Het doorsturen naar de inlog pagina indien geen resultaten gevonden
- Het instellen van de sessie gegevens indien één resultaat gevonden
- Het doorsturen naar de klanten pagina

Dankzij mijn vooronderzoek zijn de meeste van bovenstaande punten eenvoudig te implementeren. De implementatie binnen de Fusebox structuur is lastiger. Het belangrijkste struikelpunt is het bepalen in welk onderdeel van Fusebox de database verbinding wordt gemaakt. Dit kan men op het moment doen dat een query wordt uitgevoerd, de verbinding wordt gesloten op het moment dat de resultaten zijn verwerkt. Een andere mogelijkheid is het initialiseren van de verbinding bij elke hit op de website. Deze optie heeft mijn voorkeur omdat er dan sprake is van centrale afhandeling. Tijdens de implementatie werd duidelijk dat PEAR DB niet direct een verbinding maakt met het DBMS maar wacht totdat de verbinding aangesproken wordt. Tevens verbreekt hij de verbinding zodra deze niet meer nodig is. Hierdoor blijven er geen onnodige verbindingen met de server openstaan.

7.3 Klantenpagina

7.3.1 Tonen klanten

De klantenpagina toont alle binnen het systeem bekende klanten. Standaard worden alle klanten opgehaald uit de database, dit kan resulteren in een lange onoverzichtelijke lijst met klanten. Daarom is het ook mogelijk om per begin letter de klanten te ophalen. Wanneer er een klant gekozen wordt, wordt de naam, id en directory in de sessie gezet. De directory bevat de detail bestanden van de specifieke klant. Per klant kunnen er andere detail bestanden staan, het is immers mogelijk dat één klant wel remote login mogelijkheden heeft en een andere klant niet.

Na de keuze van de klant wordt de gebruiker doorgestuurd naar de detail pagina met daarop de algemene klantgegevens. Omdat dit onder het detail onderdeel valt wordt dit besproken in 7.4.

Kies een klant:		
Alles <u>A</u> <u>B</u> <u>C</u> <u>D</u> <u>E</u> <u>F</u> <u>G</u> <u>H</u> <u>I</u> <u>J</u> <u>K</u> <u>L</u> <u>M</u> <u>N</u> <u>O</u> <u>P</u> <u>Q</u> <u>R</u> <u>S</u> <u>T</u> <u>U</u> <u>V</u> <u>W</u> <u>X</u> <u>Y</u> <u>Z</u>		
Naam	Datum	Gebruiker
Test gebruiker	01-06-2004 15:24:01	Test gebruiker
Test gebruiker	01-06-2004 16:38:11	Test gebruiker
Test gebruiker	12-05-2004 09:37:17	Test gebruiker
Test gebruiker	12-05-2004 10:04:54	Test gebruiker
test	01-06-2004 09:46:01	Test gebruiker
Test gebruiker	11-05-2004 09:44:45	Test gebruiker
Test gebruiker	17-05-2004 16:22:08	Test gebruiker

Figuur 8: Klantenpagina

(Uit privacy overwegingen zijn de klantnamen onleesbaar gemaakt)

7.3.2 Toevoegen van een klant

Het toevoegen van een klant bestaat uit meer onderdelen dan alleen de klant toevoegen binnen de database. Het toevoegen van een klant bestaat globaal uit de volgende stappen.

- Controleren of klant al bestaat
- Directory voor de gegevensbestanden genereren
- Controleren of directory al bestaat
- Klant toevoegen in de database
- Aanmaken van directory's op het bestandsysteem
- Instellen van de sessie gegevens (oa klanternaam, id, directory en detailbestand)
- Doorsturen naar het toevoegen van een detail (voor de verdere gegevens)

Voordat de klant toegevoegd wordt, wordt er eerst een directory naam gegenereerd. Deze directory naam bevat alle letters vanuit de klanternaam, alle overige tekens worden genegeerd. Beide onderdelen zijn van belang voor een correcte werking van het systeem. Daarom is er eerst gecontroleerd of één van beide al bekend is in het systeem. Indien dit het geval is zal het toevoegen van de klant worden afgebroken om zo de data consistentie te kunnen waarborgen.

Nadat de invoer gecontroleerd is, worden de gegevens verwerkt. De eerste stap is het aanmaken van de directory's op het bestandssysteem, waarna de klant toegevoegd kan worden in de database. Als laatste stap worden de gegevens weggeschreven naar de sessie gegevens. Vervolgens wordt de gebruiker doorgestuurd naar de detail pagina waar hij de algemene klantgegevens kan toevoegen.

Tijdens de implementatie van dit onderdeel zijn er een aantal kleine problemen naar voren gekomen. De volgorde van de handelingen is van belang om een correcte foutafhandeling te hebben. Indien er een directory niet gemaakt kan worden mag de klant nog niet bekend zijn in het systeem. Bij de eerste opzet werd eerst de klant toegevoegd in de database, op het moment dat het aanmaken van een directory niet lukte kon de klant wel gekozen worden. Dit resulteerde in foutmeldingen omdat de benodigde bestanden niet gevonden konden worden. Dit probleem is opgelost door eerst alle directory's aan te maken en indien dit succesvol is pas de klant toe te voegen in de database.

Een groter probleem kwam naar voren bij de invoer van een klant waar het teken ' in de naam voor kwam. Omdat binnen PHP strings weergegeven worden door ze tussen ' of tussen " te zetten resulteert dit in fouten. Op de meeste plekken binnen de applicatie maakte ik gebruik van ' om een string aan te geven, dit heb ik overal omgezet naar ". Dit loste het probleem van ' wel op maar het probleem van " niet, de uiteindelijke oplossing bestaat uit het escapen van tekens. Door het escapen van " en ' worden deze omgezet naar \" en \" waardoor ze als speciaal karakter worden gezien binnen PHP. Theoretisch zou dit alle problemen met betrekking tot ' en " moeten oplossen maar helaas bleek dit niet het geval. De XPATH functie van SimpleXML kan niet goed overweg met \". Als oplossing is er besloten om overal waar een gebruiker " invoert dat om te zetten naar ". Hierdoor is het probleem met " opgelost.

7.3.3 Wijzigen van een klant

Het wijzigen van een klant heeft alleen betrekking op de naam in de database, vanwege de dynamische inhoud van de detail bestanden is het niet bekend waar de klanternaam nog meer voorkomt en of deze ook gewijzigd moet worden. Het wijzigen van de algemene klant gegevens valt onder het wijzigen van een detail gegeven, deze wordt besproken in paragraaf 7.4.

7.4 Detailpagina

De *detailpagina* is het deel van de applicatie dat door de gebruikers het meest zal worden gebruikt. Een belangrijk punt is dan ook de snelheid waarmee de gegevens op het scherm komen. De drie onderdelen (tonen, toevoegen en wijzigen) worden hierna besproken.

7.4.1 Tonen detail gegevens

Het tonen van de detail gegevens volgt een aantal stappen. Aan de hand van een voorbeeld worden deze stappen beschreven.

Een gebruiker kiest welk detail hij wil bekijken, bijvoorbeeld de server gegevens. Deze gegevens kunnen meerdere servers bevatten. De eerste opzet toonde alle servers op het scherm inclusief alle niet ingevulde velden. Tijdens een korte demonstratie aan de toekomstige gebruikers werd de voorkeur uitgesproken om eerst een overzicht te krijgen van de aanwezige onderdelen in een detail bestand. Alsook om de niet ingevulde velden niet weer te geven tijdens het tonen van een detail maar wel bij het toevoegen of wijzigen. In het geval van servers komt er eerst een overzicht met alle aanwezige servers waarna de gebruiker er één kan uitkiezen. Indien er maar één in het systeem bekend is dan wordt deze stap overgeslagen.

De opgehaalde gegevens worden doorgegeven aan de template engine die er een overzichtelijk geheel van maakt. Een voorbeeld van een deel van een detail pagina is te vinden in figuur 9.

ntserver	
Datum laatst bijgewerkt:	1-06-2004
Datum installatie:	1-06-2004
Medewerker code:	marcel
Server locatie:	in serverhok
Servernaam:	server
Domein:	server
DNS Naam:	server.localhost
Administrator wachtwoord:	geheimpje
KCC Wachtwoord:	geheimpje
IP Adres intern:	192.168.0.1
IP Adres extern:	123.123.123.123
DHCP:	192.168.0.2 - 192.168.0.254
DHCP opties:	geen
Gereserveerde IP-nummers:	geen
Server merk:	compac
Server type:	dinges zoveel

Figuur 9: Gedeelte detailpagina

7.4.2 Toevoegen detail gegevens

Het toevoegen van een detail kent twee mogelijkheden, het toevoegen van een detail aan een bestaand bestand of het toevoegen van een detail aan een niet bestaand bestand. In beide gevallen is de basis hetzelfde. De eerste actie van het systeem is het controleren van de lock. Indien een andere gebruiker in het detail onderdeel aan het toevoegen of wijzigen is zal het systeem een melding geven en wordt het toevoegen van een detail gestopt. Wanneer er geen andere gebruiker bezig is op het systeem wordt er een lock geplaatst waarna de gebruiker het detail kan toevoegen.

De gebruiker krijgt hetzelfde scherm te zien als bij het tonen van de detail gegevens, alleen zijn de detail gegevens invoervelden en bevatten ze geen gegevens. Tevens zijn er twee knoppen zichtbaar namelijk Opslaan en Annuleren.

Na het drukken op de Opslaan knop, wordt er begonnen met het opslaan van de gegevens. Voordat er opgeslagen gaat worden, wordt er gecontroleerd of de ingevulde gegevens voldoen aan de gestelde invoer criteria.

Na de controle van de invoer wordt er onderscheid gemaakt tussen het toevoegen van een detail aan een nieuw bestand of aan een bestaand bestand.

Nieuw bestand

Wanneer er een detail toegevoegd wordt aan een nieuw bestand kan er een nieuw XML object gemaakt worden. Dit wordt gecontroleerd op correctheid waarna het in één geheel weggeschreven wordt naar de schijf.

Bestaand bestand

Indien er reeds een detail bestand aanwezig is, zal dit ingelezen moeten worden. Het nieuw toegevoegde detail wordt omgezet naar een object waarna de twee objecten samengevoegd worden tot één geheel. Dit wordt gecontroleerd op correctheid. Voordat dit object weggeschreven wordt naar de schijf wordt er een back-up gemaakt van de huidige versie waarna de nieuwe versie weggeschreven wordt.

Wanneer het detail succesvol is toegevoegd wordt het lock verwijderd en de gebruiker doorgestuurd naar de tonen pagina zodat de zojuist ingevoerde gegevens op het scherm staan.

7.4.3 Wijzigen detail gegevens

Het wijzigen van een detail gebruikt dezelfde lay-out als het tonen ervan. Voordat een gebruiker een detail gaat wijzigen wordt er gecontroleerd of een andere gebruiker dit detail reeds aan het aanpassen is. Indien dit het geval is, wordt het gemeld aan de gebruiker. De detail gegevens kunnen dan niet gewijzigd worden.

Er wordt zoveel mogelijk gebruik gemaakt van de source code van het detail tonen onderdeel, dit verhoogt de onderhoudbaarheid van de website. Om onderscheid te kunnen aanbrengen tussen het tonen en het wijzigen van een detail onderdeel zijn er parameters aan de te gebruiken functies toegevoegd. Door middel van deze parameters wordt bepaald of de lege velden getoond moeten worden en of de tekst ingevuld moet worden in een invoerveld of niet. Figuur 10 geeft een voorbeeld weer van de detail wijzigen pagina, in dit voorbeeld worden de gegevens uit figuur 9 gewijzigd.

ntserver

Datum laatst bijgewerkt:	1-06-2004
Datum installatie:	1-06-2004
Medewerker code:	marcel
Server locatie:	in serverhok
Servernaam:	server
Domein:	server
DNS Naam:	server.localhost
Administrator wachtwoord:	geheimpje
KCC Wachtwoord:	geheimpje
IP Adres intern:	192.168.0.1
IP Adres extern:	123.123.123.123
DHCP:	192.168.0.2 - 192.168.0.254
DHCP opties:	geen
Gereserveerde IP-nummers:	geen
Server merk:	compaq

Figuur 10: Detail wijzigen

Wanneer de detail gegevens gewijzigd zijn, moeten deze nog opgeslagen worden in het detail bestand. Voordat de gegevens weggeschreven worden, wordt er gecontroleerd of ze aan de gestelde invoereisen voldoen. Deze invoercontrole wordt op een centraal punt afgehandeld. De uit te voeren controles staan in het meta bestand, zodoende is er per record te bepalen welke controles er uitgevoerd moeten worden. Indien er incorrecte invoer wordt gevonden, wordt de gebruiker op de hoogte gesteld.

Wanneer alle invoer correct ingevoerd is, wordt die opgeslagen op de schijf. Voordat er een wijziging doorgevoerd wordt, wordt er een back-up gemaakt van de huidige gegevens. Dit zorgt ervoor dat de gebruikers geen mogelijkheid hebben om de gegevens te "verwijderen". Dit "verwijderen" is het vervangen van de gegevens door lege of foutieve invoer waarna deze opgeslagen worden. Tevens zorgt dit ervoor dat bij een mogelijke fout tijdens het wegschrijven naar schijf de gegevens niet verloren gaan.

Het opslaan van de gewijzigde gegevens gebeurt door middel van een standaard SimpleXML functie, die maakt van het XML object een tekstuele weergave waarna die in het bestand geschreven wordt.

7.4.4 Automatisch aanpassen detail bestand

De inhoud van het detail bestand is afhankelijk van de inhoud van het meta bestand. Op het moment dat de beheerder het meta bestand aanpast, moeten de wijzigingen automatisch doorgevoerd worden in de detail bestanden. Een wijziging zal vooral bestaan uit het toevoegen van een veld. Dit moet tevens in alle detail bestanden beschikbaar zijn. Om de bestanden aan de nieuwe meta bestand aan te passen is er een functie geschreven. Die functie wordt bij elke toon actie aangeroepen, hierdoor bevatten de bestanden altijd de correcte structuur.

Indien er een aanpassing aan het bestand gemaakt wordt, wordt er een backup gemaakt. Het detail wordt gevuld met de standaard waarde of leeg gelaten indien er geen standaard waarde bekend is. Wanneer er een veld verwijderd wordt, wordt het niet uit het detail bestand verwijderd. Omdat er geen bijbehorend meta gegeven gevonden wordt, wordt het detail ook niet op het scherm getoond, dit komt overeen met de wens van Korton om geen gegevens te verwijderen maar slechts onzichtbaar te maken voor de gebruiker. Een Andere optie in plaats van het verwijderen van de gegevens is het aanzetten van het displayonly vlaggetje. Dit zorgt ervoor dat de gegevens alleen toegevoegd kunnen worden en niet gewijzigd, bij het tonen van de gegevens op het scherm zijn deze gegevens ook beschikbaar.

7.5 Urendeclaratie

De urendeclaratie is een speciaal detail gedeelte, in vergelijking tot een detail pagina zitten er enkele verschillen in. Het grootste verschil is het uiterlijk, de urendeclaratie heeft een vooraf bepaalde indeling en een vaste veld structuur. De overeenkomst met een detail pagina zit in het gebruik van (bijna) dezelfde functies en het feit dat een urendeclaratie intern dezelfde structuur heeft als een detail pagina. Bij een detail pagina kan men eerst een keuze maken welk onderdeel men van een detail wil bekijken. Bij de urendeclaratie kiest men eerst een urendeclaratie. Een groot verschil is dat een urendeclaratie bestand exact één urendeclaratie bevat. Tevens is de structuur van het bron bestand voor een urendeclaratie vaststaand en mag niet meer gewijzigd worden omdat anders de template niet meer klopt met het afdruk formaat.

7.5.1 Tonen urendeclaratie

Om de urendeclaraties netjes op het scherm te tonen heb ik een aparte template gemaakt. Deze template is volledig afgestemd op de urendeclaratie formulier en ziet er hetzelfde uit als de papieren versie en wanneer die afgedrukt wordt past hij exact op een A4 pagina.

Het tonen van een urendeclaratie maakt gebruik van dezelfde functies als tonen van een detail pagina. Het verschil zit hem in de toewijzing naar de template, bij de urendeclaratie moeten de juiste velden de correcte waarden krijgen anders ontstaat er een vertekend beeld. In figuur 11 op de volgende pagina staat een voorbeeld urendeclaratie.

**Urendeclaratieformulier**

Firma: Test BV
Aanspreekpartner: Jantje
Straat: Jadelaan 99B
Postcode/Plaats: 2123XZ Hoofddorp

Medewerker: Marcel S
Datum: 1 juni 2004
Melding: Fax

Doorgevoerde werkzaamheden:

Installatie mailserver Configuratie cisco router Aansluiten xs4all ADSL basic

Geleverde / uitgeleende onderdelen + s/n. Voor installatie (soort/plaats):

geen

Gewerkte uren per persoon:

Marcel S 9.00 - 17.00 pauze: 30 min

Materiaal- en voorrijkosten en BTW zijn niet inbegrepen tenzij expliciet in de service-overeenkomst hierover een regeling is getroffen. Als Cliënt een service-overeenkomst heeft geldt bij overschending van de maandelijkse uren het Extra-uren tarief volgens de service-overeenkomst. Anders gelden de prijzen volgens de prijslijst van Korton Groep bv. Het bedrag wordt gefactureerd of d.m.v. incasso verrekend.

De werkzaamheden werden tot onze tevredenheid uitgevoerd.

Datum:
1 juni 2004

Handtekening:

Gereed melden

Figuur 11: Voorbeeld urendeclaratie formulier

7.5.2 Toevoegen van een urendeclaratie

Het toevoegen van een urendeclaratie lijkt in sterke mate op het toevoegen van een detail bestand. Er zijn echter belangrijke verschillen tussen de beide onderdelen. Een groot verschil is het feit dat een urendeclaratie bestand altijd exact één urendeclaratie bevat. Het aanmaken van het bestand begint met het genereren van de bestandsnaam, die bestaat uit de huidige datum en tijd. Na het genereren van de bestandsnaam wordt er begonnen met het controleren van de invoer. Bij correcte invoer zal de urendeclaratie samengesteld worden aan de hand van de invoer. Indien die correct is zal hij worden opgeslagen op schijf. Bij een mogelijke foutmelding wordt de gebruiker op de hoogte gesteld met een nette foutmelding. De beheerder krijgt een e-mail met daarin uitgebreide informatie met betrekking tot de fout.

7.5.3 Wijzigen van een urendeclaratie

Het wijzigen van een urendeclaratie verschilt slechts op 1 punt van het wijzigen van een detail pagina. Voordat een urendeclaratie gewijzigd wordt, moet er gecontroleerd worden of hij gewijzigd mag worden. Indien hij gewijzigd mag worden, is de werking gelijk aan het wijzigen van een detail onderdeel.

7.5.4 Gereed melden van een urendeclaratie

De urendeclaratie wordt online ingevoerd waarna hij afgedrukt wordt. Deze uitgeprinte versie wordt ondertekend door een medewerker van het bedrijf. Op het moment dat deze handtekening gezet is mag de urendeclaratie niet meer gewijzigd worden. Dit wordt voorkomen door de urendeclaratie gereed te melden. Wanneer hij gereed gemeld is, is de urendeclaratie definitief en kan hij gefactureerd worden.

Het gereed melden wordt gedaan door een veld in het bestand op waar te zetten. Bij het wijzigen wordt hierop gecontroleerd. Indien een urendeclaratie al gereed is dan kan deze niet meer gereed gemeld worden.

7.6 Todo

De todo's zijn een apart onderdeel binnen de website geworden, in de eerste opzet zouden de todo's een detail onderdeel vormen. Hiervan is afgestapt, de reden hiervoor is het overstappen van XML naar database als dataopslag. In deze paragraaf wordt de implementatie van de todo's binnen de website besproken.

7.6.1 Tonen van de todo's

Standaard worden bij het tonen alle todo's getoond. Door middel van een filter kan de gebruiker instellen welke todo's voor hem van belang zijn. Dit filter biedt de keuze uit filteren op: gereed of niet gereed, prioriteit, aanmaakdatum, verloopdatum en datum gereed gemeld. Tevens kan hij kiezen op welk veld er gesorteerd moet worden, verloopdatum of prioriteit.

Indien er gefilterd wordt op een datum kan er een bereik aangegeven worden, dit wil zeggen dat de gebruiker een begin datum en een einddatum op kan geven. Deze zijn niet verplicht. Als er alleen een begin datum opgegeven wordt, zullen alle todo's vanaf die datum opgehaald worden. Indien er alleen een einddatum opgegeven wordt zullen alle todo's tot en met de einddatum getoond worden.

Zoals besproken in hoofdstuk 6 maken de todo's gebruik van een database voor de dataopslag. Het filter wordt direct toegepast op de database, er worden dus geen overbodige gegevens opgehaald en verwerkt.

Gereed

Geen voorkeur ▼

Prioriteit

Geen voorkeur ▼

Aanmaakdatum tussen
 en
Verloopdatum tussen
 en
Datum gereed gemeld tussen
 en
Gesorteerd op:

Geen voorkeur ▼

Toon

Omschrijving	Prioriteit	Verloopdatum	Oplossing	Gereed gemeld door	Gereed melden	Wijzigen
Windows 2000 updaten naar SP4	standaard	11-06-2004			☒	Wijzigen

Figuur 12: Todo pagina

7.6.2 Toevoegen van een todo

Het toevoegen gedeelte maakt gebruik van dezelfde template als het tonen van de todo's. Door aan de template door te geven dat het filter uitgeschakeld moet worden en dat het om toevoegen van een todo gaat, komen de juiste velden en knoppen op het scherm. Door deze keuze blijft de lay-out van het gehele todo gedeelte gelijk aan elkaar en is het geheel makkelijk te wijzigen omdat er maar één template gewijzigd hoeft te worden.

De gebruiker geeft een omschrijving van de uit te voeren todo, een prioriteit en een datum wanneer hij uiterlijk gereed moet zijn. Daarnaast is het mogelijk om een (mogelijke) oplossing in te voeren, dit veld is niet verplicht. Het systeem vult deze gegevens aan met de klant gegevens en de aanmaak datum. Dit geheel wordt opgeslagen in de database waarna de gebruiker doorgestuurd wordt naar de todo pagina. Hierop staat de zojuist ingevoerde todo.

Als het toevoegen van de todo om één of andere reden niet lukt, krijgt de gebruiker een leesbare foutmelding, de detail gegevens over de fout worden naar de beheerder geë-mailed.

7.6.3 Wijzigen van een todo

Het wijzigen van de todo heeft dezelfde lay-out als het toevoegen van een todo. Bij het wijzigen van een todo kunnen alle de door de gebruiker ingevoerde velden gewijzigd worden.

7.6.4 Gereed melden van een todo

Wanneer een todo afgerond is, wordt hij gereed gemeld. Dit kan gedaan worden door de medewerker van KCC maar in de toekomst moet een klant dit ook kunnen. Wanneer een todo gereed gemeld wordt, wordt er bijgehouden op welke datum dit gebeurde en door welke gebruiker. Wanneer een todo gereed gemeld is kan deze niet meer gewijzigd worden.

Het is niet mogelijk om een todo te verwijderen, mocht een todo foutief ingevuld zijn dan kan hij gewijzigd of gereed gemeld worden.

7.7 Uploaden van bestanden

Tijdens de demo werd het verzoek uitgesproken om bestanden te kunnen uploaden, het betreft dan voornamelijk router configuratie bestanden en netwerk tekeningen. Het uploaden van de bestanden moet via de website kunnen en mag geen gebruik maken van extra externe software. De bestanden moeten weer gedownload kunnen worden om te kunnen bekijken of te gebruiken.

De implementatie van dit onderdeel bestaat uit één template en drie fuseactions. Eén om de bestanden te uploaden, één om ze te downloaden en één om een overzicht van de bestanden te krijgen. Een mogelijke beveiligingsfout zou kunnen zitten in het uploaden van webscripts. Deze zouden uitgevoerd in plaats van gedownload kunnen worden. Op de ontwikkelomgeving bleek dit niet het geval, op de uiteindelijke testomgeving zal dit nogmaals gecontroleerd moeten worden omdat het een server instelling is.

Wanneer er een voor de browser bekend bestand gedownload wordt, zal dit direct geopend worden, zo worden foto's direct getoond.

Het is niet de bedoeling dat de medewerkers het uploaden van bestanden gaan gebruiken om grote bestanden met elkaar uit te wisselen, om dit te voorkomen staat er een maximale bestandsgrootte van 10Mb in gesteld. Dit is in de server instelling vrij aan te passen, mocht het nodig blijken dan kan hij hoger gezet worden.

Kies een bestand:

Het bestand mag niet groter zijn dan 10MB

Indien het bestand reeds bestaat zal deze overschreven worden.

Klik op een bestand om dit te downloaden, of gebruik "doel opslaan als"

 [Eindverslag.doc](#)

Figuur 13: Upload / download pagina

7.8 Zoeken

Het zoeken is als laatste onderdeel aan de website toegevoegd. De zoekmachine is een eenvoudige implementatie, hij zoekt naar de opgegeven zoekterm binnen de detail bestanden. Om de snelheid en eenvoud te verhogen kan de gebruiker eerst een keuze maken uit de te doorzoeken klanten en detail bestanden. De gebruiker wordt aangeraden om een zo'n gedetailleerd mogelijke keuze van klanten en detail bestanden te maken om snel resultaat te krijgen en het systeem niet overmatig te belasten.

Klanten: 

Details: 

Zoekterm:

Figuur 14: Voorbeeld zoekpagina

(Uit privacy overwegingen zijn de klantnamen onleesbaar gemaakt)

De resultaten komen op het scherm te staan, een zoek resultaat bestaat uit een link naar de klantpagina en een link naar de detail pagina('s). Indien de zoekterm in meerdere detailbestanden gevonden is, zullen er meerdere links onder de klant staan.

Gezocht op: Marcel

7 klanten en 33 bestanden doorzocht in 0.686 seconden.

3 resultaten gevonden.

test

algemeen

hosting

server

Figuur 15: Zoekresultaten

Het zoeken is een intensief proces voor het systeem. De gebruikers zijn geadviseerd om zo gespecificeerd mogelijk te zoeken.

8. Testen

Dit hoofdstuk beschrijft de uitgevoerde werkzaamheden met betrekking tot verificatie en evaluatie van de website.

Korton Internet maakt geen gebruik van standaard testmethoden zoals unit testing. Korton test zijn in opdracht voltooide producten met de methode van valide/invalid invoer. Wanneer deze tests zijn voltooid en de mogelijke fouten zijn hersteld, wordt de applicatie in een testomgeving gezet zodat de opdrachtgever het kan testen.

8.1 Debuggen

Gedurende de implementatie van de door mij gebouwde website is de website zowel als de delen daarvan continue getest op correctheid, betrouwbaarheid, robuustheid, gebruikersvriendelijkheid en onderhoudbaarheid. Deze vorm van testen noemt men debuggen. Elke keer wanneer er een stuk source code aan de website toegevoegd wordt, zal deze beproefd worden op correctheid en foutafhandeling. De meeste fouten zullen door deze methode uit de applicatie gefilterd worden.

8.2 Testsets

De test zijn uitgevoerd volgens het whitbox principe zoals Ghezzi deze beschrijft. De specificaties zijn opgemaakt aan de hand van de sourcecode.

De correctheid, betrouwbaarheid en robuustheid is getest door middel van testcases, de testcases zijn gebaseerd op het pilot ontwerp. Een voorbeeld van zo'n testcase bevindt zich in figuur 16.

Onderdeel:	Details wijzigen
Invoer:	Algemene gegevens wijzigen van de zojuist ingevoerde klant. Correcte invoer.
Verwachte uitvoer:	Detail pagina met gewijzigde gegevens
Gekregen uitvoer:	Detail pagina met de gewijzigde gegevens
Opmerking:	

Onderdeel:	Details wijzigen
Invoer:	Algemene gegevens wijzigen van de zojuist ingevoerde klant. Foutieve invoer.
Verwachte uitvoer:	Error bericht op het scherm
Gekregen uitvoer:	Error bericht op het scherm
Opmerking:	

Figuur 16: Voorbeeld testcase

In figuur 16 wordt het wijzigen van een detail getest op correctheid, er worden twee cases onderscheiden. Eén wanneer de invoergegevens correct zijn volgens de gestelde invoer eisen en één waarbij de invoergegevens incorrect zijn. In beide gevallen reageert het systeem zoals gewenst. De test wordt meerdere malen uitgevoerd waarbij de invoer afgewisseld wordt door correct in incorrect. Indien het scherm meerdere invoervelden bevat moet elk veld minstens één maal correcte invoer bevatten en één maal incorrecte invoer. Om de betrouwbaarheid van de applicatie te testen, is elke testcase meerdere malen uitgevoerd waarbij de invoer elke keer exact dezelfde was. Bij een betrouwbaar systeem zal de uitvoer gelijk blijven. De website doorstond deze test goed.

De gebruikersvriendelijkheid is getest door enkele van de toekomstige gebruikers. Voordat deze tests uitgevoerd werden is de werking naar eigen inzicht opgebouwd en getest. De uiteindelijke gebruikers hebben de gebruikersvriendelijkheid positief beoordeeld.

De onderhoudbaarheid is van belang voor Korton Internet. Deze zullen de uiteindelijke website in onderhoud nemen. Het testen van de onderhoudbaarheid wordt gedaan door te kijken naar de fusebox structuur. Een correct opgebouwde structuur verhoogd de onderhoudbaarheid, tevens zorgt het gebruik van de Hongaarse notatie voor een onderhoudbaarder geheel. Tijdens de test kwam naar voren dat er beter gebruik gemaakt had kunnen worden van circuits binnen de website. De onderhoudbaarheid van de website is matig, het uitbreiden van de circuits zou het geheel onderhoudbaarder maken.

8.3 Gebruikerstest

Enkele gebruikers hebben de mogelijkheid gekregen om de website te testen. De gebruikers hebben de opdracht gekregen om een van hun eigen klantenmappen in het systeem te zetten en zich er een beeld van te vormen of de website tegemoet komt aan hun persoonlijke verwachtingen en doelstellingen.

De reacties van de gebruikers waren op veel punten positief. De lay-out van de website is duidelijk en eenvoudig. Er kwamen voornamelijk opmerkingen met betrekking tot de aanwezige data, over de aanwezige functionaliteiten waren ze allen erg te spreken.

De tests die door de gebruikers uitgevoerd zijn vallen onder de categorie black box testing. Ze kijken naar de verwachte werking van de website. Aan de hand van hun eigen verwachtingen bepalen ze of de website voldoet. Aan de hand van de test konden ze feedback geven met betrekking tot de aanwezige gegevens (bijvoorbeeld het ontbreken van een gegevensveld) en de algemene werking. De feedback is, na overleg met het management, zoveel mogelijk direct in de website verwerkt.

In de laatste week is de website in de productieomgeving geplaatst. Hier kunnen de gebruikers en het management de website testen waarna de officiële overdracht plaats zal vinden.

8.4 Acceptatietest

Op maandag 7 juni 2004 zijn de website en de bijbehorende nevenproducten voor eerste oplevering en acceptatietest overgedragen. Vanaf die dag kunnen de medewerkers en het management werken met de website. Op maandag 14 juni 2004 zullen de resultaten hiervan geëvalueerd worden waarna ter voorbereiding op de tweede oplevering de fouten zullen worden hersteld. Na de definitieve overdracht zal de website uitgebreid gaan worden met een lay-out die binnen de huisstijl van Korton past. Tevens zal dan de rechtenstructuur geïmplementeerd gaan worden.

8.5 Conclusie

Bij het testen van de onderdelen van de website ben ik uitgegaan van de basis begrippen van C. Ghezzi. Voor elk van deze begrippen maakte ik een testinstrument van het "ja/nee type". Nadeel hiervan is dat er op die manier geen gevalideerde testmethode is gebruikt; voordeel is dat de instrumenten passen op de testen producten.

Test	1 ^e keer correct	incorrect	2 ^e keer correct
Correctheid	16	6	22
Betrouwbaarheid	88	0	-
Robuustheid	88	0	-
Onderhoudbaarheid	2	1	-
Gebruikersvriendelijkheid	4	0	-

Figuur 17: Testresultaten

Bovenstaand tabel geeft de testresultaten weer. Voor de betrouwbaarheid en robuustheid is de correctheid test als basis genomen. Bij de betrouwbaarheid is de test meerdere malen uitgevoerd om te controleren of de output constant bleef. Voor de robuustheid is de invoer gewijzigd, hierbij werd zoveel mogelijk correcte en incorrecte invoer gebruikt.

Over de acceptatietests valt op het moment van inleveren van dit verslag nog niets te melden. Op grond van de bovengenoemde testuitslagen, met het voorbehoud voor de acceptatietests, mag ik concluderen dat de website en de nevenproducten in goede mate voldoen aan de gestelde eisen en verwachtingen.

9. Evaluatie

Met de methodiek van de SWOT analyse kijk ik met een kritisch blik terug op het proces dat ik heb doorgemaakt bij het ontwikkelen van de website. Ook kijk ik terug op de producten die ik heb vervaardigd. Ik stel mezelf de volgende vragen:

- Wat waren de sterke punten?
- Wat waren de zwakke punten?
- Welke lessen heb ik ervan geleerd?

Ik doe dit aan de hand van de onderwerpen die in de inhoudsopgave in de opeenvolgende hoofdstukken aan de orde zijn geweest.

Problematiek en afstudeeropdracht:

Sterk: Duidelijke omschrijving probleem.

Zwak: Geen afbakening in de opdrachtoomschrijving.

Geen motivering bij de te gebruiken methode en te korte uitleg.

Les: Baken een opdracht zo vroeg mogelijk al af. Omschrijf duidelijk welke onderdelen er wel binnen vallen en welke niet.

Plan van eisen:

Sterk: Duidelijk de eisen van opdrachtgever en opdrachtnemer genoteerd.

Zwak: De interview methode die gebruikt is past niet goed binnen de cultuur van Korton.

Website beantwoord niet aan alle door de opdrachtgever gestelde eisen.

Les: Ik moet meer oog hebben voor welke interview techniek passend is binnen een organisatie onderdeel.

Ordering van de Werkzaamheden:

Sterk: Goed en duidelijk beschreven welke activiteiten ik moet uitvoeren.

Beschreven wat er ontwikkeld zal worden.

Precedentieschema om structuur aan te brengen in de werkzaamheden.

Zwak: Geen gevalideerde testmethode beschreven.

De planning was voornamelijk gebaseerd op intuïtie. Achteraf gezien is dat redelijk geslaagd. Toch zou ik de planning rationeler moeten opzetten.

Les: Om een project rationeler te kunnen organiseren moet ik time management methoden hanteren.

Vooronderzoek:

Sterk: Programmastructuur, templates en opslagstructuur alternatieven aan de hand van beslissingstabellen met elkaar vergeleken.

Ik heb alternatieven gezocht als opslagstructuur. XML was aangedragen door Korton, ik heb gekeken naar alternatieven.

Ook zijn er verschillende template engines bekeken en uitgetprobeerd. Er is verder gekeken dan de eerste de beste die aan de wensen voldeed.

Ik heb aandacht besteed aan de veiligheid van de website en de mogelijkheden om de veiligheid te verhogen.

Zwak: De rechtenstructuur van gebruikers is onderbelicht gebleven. Ik heb me te veel laten leiden door de wetenschap dat de rechtenstructuur geen deel zou uitmaken van de implementatie.

Gebruik van de ontwikkel versie van PHP in een productieomgeving houdt risico's in. Omdat simpleXML als standaard onderdeel van PHP 5 mij veel gebruikersmogelijkheden bood, heb ik met instemming van de opdrachtgever toch gekozen voor PHP 5.

Les: Een beslissingstabel is een belangrijk hulpmiddel bij het nemen van beslissingen. Als opdrachtnemer mag ik mijzelf niet toestaan om een instabiel instrument toe te passen in een website die ik in opdracht van derden ontwikkel. Indien een onderdeel niet geïmplementeerd hoeft te worden maar wel ontworpen, moet aan dat ontwerp dezelfde aandacht geschonken worden als aan onderdelen die wel geïmplementeerd worden.

Ontwerp:

Sterk: Het gebruiken van Adalon als casetool, het vergemakkelijkt het ontwerpen en verhoogt de consistentie.

Ontwerp opgedeeld in kleinere deelontwerpen zodat er een duidelijk overzicht blijft.

Gebruik gemaakt van een pilot om de mogelijke functies binnen de implementatie weer te geven.

Gebruik gemaakt van prototyping bij het samenstellen van de schermen.

Zwak: Te weinig gebruik gemaakt van circuits binnen Fusebox. Hierdoor is de structuur te weinig gedifferentieerd.

Les: Gebruik een casetool om het werk te vergemakkelijken maar zorg dat je de casetool goed beheerst.

Splits het ontwerp in betekenisvolle onderdelen zodat de structuur van het ontwerp transparant wordt.

Implementatie:

Sterk: Goed voorbereid dmv vooronderzoek.

Directory structuur gelijk gemaakt aan de standaard van Korton Internet.

Modulair geprogrammeerd.

Zwak: De website is informeel.

Ik werd me in een laat stadium bewust dat de manier van foutafhandeling niet de gewenste functionaliteit bezat. Daarbij hadden de fouten geen standaard lay-out. Het inbouwen van centrale foutafhandeling bleek evenwel de geïnvesteerde extra tijd meer dan waard.

Les: In de ontwerpfase moet ik meer aandacht besteden aan de implementatie en aan de structuur daarvan.

Het direct implementeren van een foutafhandelingsmodule zorgt direct voor een nette en centrale foutafhandeling.

Testen:

Sterk: Ik heb de gebruikers vroeg betrokken bij het testen van de website.

Het gebruiken van de basis begrippen van Ghezzi heeft geresulteerd in gestructureerde manier van testen.

Zwak: Er is geen gebruik gemaakt van gevalideerde testmethoden zoals unit testing.

Les: Het is aan te bevelen om aan het onderwerp verificatie tijdens het ontwerpen van een software programma meer aandacht te schenken.

Ik ben zeer tevreden met het behaalde resultaat, toch zijn er enkele aandachtspunten waarmee ik rekening moet houden bij volgende projecten. De belangrijkste is de afbakening van de opdracht, bij een groot project zou dit problemen kunnen geven. De opdrachtgever zou steeds meer eisen kunnen stellen omdat de opdrachtomschrijving daar ruimte toe laat.

Het vooronderzoek heeft me op vele punten geholpen om de juiste beslissingen te nemen, een negatief punt in het geheel is het rechtensysteem. Deze is minder goed onderzocht dan de overige punten. Dit komt voornamelijk door de wetenschap dat de rechten niet binnen de afstudeeropdracht geïmplementeerd hoefde te worden.

Het gebruiken van Fusebox is me goed bevallen, hoewel het door vele niet als een volledige methode gezien wordt zorgt het er wel voor dat er methodisch gewerkt wordt. Daarnaast brengt het een duidelijke structuur aan binnen een website. Het grote minpunt aan mijn gebruik van Fusebox is het niet goed gebruiken van de circuits. Hierdoor is de website minder gestructureerd dan hij zou kunnen zijn.

De implementatie in PHP is goed gelukt, ondanks de mogelijke risico's die PHP 5 met zich mee bracht heb ik hier geen problemen mee gekregen. De huidige versie is release candidate 3 en dit is de laatste ontwikkelversie voordat er een stabiel verklaarde versie uitgebracht wordt. Tijdens de afstudeerperiode is PHP 5 gegroeid vanaf beta 2 naar beta 4 waarna de eerste release candidate uitgebracht is. Een belangrijke les die hieruit voortvloeit is dat het gebruiken van in ontwikkeling zijnde software beter niet gebruikt kan worden voor belangrijke opdrachten. Eventueel kan in overleg met de opdrachtgever nadrukkelijk worden afgesproken dat het gebruik van in ontwikkeling zijnde software toegestaan is omdat het bepaalde belangrijke extra mogelijkheden biedt. De verantwoordelijkheid voor de risico's dienen in die afspraak bepaald te worden.

10. Slotwoord

Terugkijkend naar het resultaat van het werken aan de afstudeeropdracht ben ik zeer tevreden. De website beantwoordt voor zover dat op dit moment kan worden overzien aan de verwachtingen. De ervaring die voor mijzelf de meeste indruk maakte, is dat ik bemerkte dat ik mijn werk steeds beter ging organiseren en structureren.

Het schrijven van het afstudeerverslag vond ik een helse klus. Na veel vallen en opstaan is het me evenwel toch gelukt. De truc die ik toepaste bij het werken aan de website ging ook hierop. Organiseren, opdelen, structureren en stap voor stap verder gaan.

Al met al; lekker en hard gewerkt, veel geleerd en ook veel toegepast van wat ik eerder in de opleiding heb opgestoken.

Tot slot bedank ik de medewerkers van Korton Group B.V. voor de prettige werksfeer en de vele positieve opmerkingen en complimenten met betrekking tot de website. Ik ben blij dat ik de komende tijd een bijdrage kan leveren als Jr Coldfusion Developer binnen Korton Internet. Ik kijk uit naar de toekomstige samenwerking met mijn collega's.

Literatuurlijst

Zwet, C. van der en Warnink, M. *Reader: Rapportage techniek, interne reeks 908*. Den Haag, juli 1997

Humphrey, Watts S, *Introduction to the Personal Software Process*, Sarasota Florida USA: Addison-Wesley 2004

Pear DB handleiding : http://pear.php.net/distributions/manual/pear_manual_nl.html.gz

Hill, H. Percy, *Making decisions, a multidisciplinary Introduction*. University Press of America inc. 1986

Ghezzi, Carlo, *Fundamentals of Software Engineering* Prentice-hall inc. New Jersey, USA.

www.php.net

<http://www.playgarden.com/w3c/TR/REC-xml/REC-xml.html>

<http://www.w3.org/TR/2004/REC-xml-20040204/>

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

Figurenlijst

Figuur 1: Precedentieschema.....	15
Figuur 2: Beslissingstabel opslagstructuur	20
Figuur 3: Beslissingstabel PHP	22
Figuur 4: Beslissingstabel template engines	24
Figuur 5: ERD.....	28
Figuur 6: Toelichting ERD.....	29
Figuur 7: Structuur overzicht.....	32
Figuur 8: Klantenpagina	37
Figuur 9: Gedeelte detailpagina.....	39
Figuur 10: Detail wijzigen	41
Figuur 11: Voorbeeld urendeclaratie formulier	43
Figuur 12: Todo pagina	45
Figuur 13: Upload / download pagina.....	46
Figuur 14: Voorbeeld zoekpagina.....	47
Figuur 15: Zoekresultaten	47
Figuur 16: Voorbeeld testcase.....	48
Figuur 17: Testresultaten	50

Bijlage A: Plan van aanpak

Inhoudsopgave:

1. Inleiding.....	58
2. Opdrachtoomschrijving.....	58
2.1 Inleiding	58
2.2 Het probleem	58
2.3 Doelstelling	58
2.4 Activiteiten	59
2.5 Eindsituatie	59
3. Producten.....	61
4. Methoden en technieken.....	62
4.1 Methoden	62
4.2 Technieken.....	62
5. Risico's.....	63
6. Precedentieschema	63
7. Planning	66

1. Inleiding

Voor u ligt het plan van aanpak zoals gedurende de gehele afstudeerperiode gebruikt zal worden. Deze periode omvat 18 weken, gedurende deze 18 weken zal er aan de opdracht gewerkt worden zoals in dit plan van aanpak beschreven staat.

2. Opdrachtomschrijving

Dit hoofdstuk beschrijft de opdrachtomschrijving zoals goedgekeurd door alle betrokken partijen.

2.1 Inleiding

Korton B.V. heeft een afdeling systeembeheer, deze afdeling verzorgt de server installatie en configuratie bij klanten op locatie. Om deze instellingen en gegevens te beheren maken ze gebruik van Excel sheets in zowel digitale als papieren vorm.

2.2 Het probleem

De technische medewerkers van KCC beschikken niet over de mogelijkheden om klantgegevens vanaf de werkplek dan wel vanaf locatie op te vragen, te raadplegen of te wijzigen. Deze gegevens behelzen naast de algemene klant gegevens ook de server instellingen, wachtwoorden, "To Do's" etc.

In de huidige situatie zijn de klant gegevens opgeslagen in zgn. klantenmappen, dossiers in papieren alsmede digitale vorm, die zich in het kantoor van Korton bevinden.

Het opvragen en raadplegen van deze dossiers zowel als het onderhouden en bijwerken ervan wordt door een deel van de medewerkers als omslachtig en verouderd ervaren. Met name op momenten dat de dossiers moeten worden geraadpleegd buiten standaardsituaties om resulteert de van kracht zijnde gang van zaken regelmatig in ongewenst gedrag bij een deel van de medewerkers. Het niet juist of niet volledig bijwerken van dossiers en het niet tijdig of in het geheel niet retourneren van dossiers zijn daar voorbeelden van. Een en ander leidt tot gebrekkige managementinformatie, irritaties bij medewerkers die later gebruik maken van sub-optimaal bijgehouden mappen, tot fouten in orders en tot klachten van klanten.

Op management niveau zowel als (deels) op de werkvloer is de geschetste situatie als een probleem aangemerkt. Vanuit het management is nagegaan of en zo ja waarom het deel uitmaakt van een meer omvattende problematiek. Het valt buiten het bestek van dit werkstuk om daar verder op in te gaan.

In het overleg over mijn stageopdracht met vertegenwoordigers vanuit het management, is mij gevraagd of ik met mijn informatiekundige kennis en kunde een hedendaags, en bij het bedrijf passend stuk gereedschap zou kunnen ontwikkelen dat er toe zou kunnen bijdragen dat de problematiek zoals geschetst, vermindert of zelfs verdwijnt.

'Passend bij het bedrijf' kan dan opgevat worden als "het imago van het bedrijf ondersteunend" zowel als "in lijn te zijn met wensen van management en eindgebruikers".

Ik heb overwogen dat gereedschap vanuit zichzelf nooit een bijdrage kan leveren aan organisatieontwikkeling. De oude volkswijsheid "goed gereedschap is het halve werk" zou evenwel ook bij Korton kunnen opgaan. Gereedschap dat het werk van de professional zonder weerstand ondersteunt, verhoogt beslist het plezier in het werken. In afgeleide zin is derhalve niet ondenkbaar dat een gereedschap dat het soepel raadplegen van klantgegevens mogelijk maakt, bijdraagt aan het voorkomen van ongewenste effecten.

2.3 Doelstelling

Ontwikkel een Internetsite waarop de medewerkers van de technische dienst kunnen inloggen om zo de klantgegevens te kunnen raadplegen en te wijzigen.

2.4 Activiteiten

Globaal gezien zullen er de volgende activiteiten uitgevoerd worden.

- **Samenstellen van het Plan van Aanpak.**
In het plan van aanpak wordt het gehele traject vastgelegd alvorens te beginnen met de opdracht. Binnen dit plan van aanpak zal een gedetailleerde omschrijving van de activiteiten staan.
- **Interviewen toekomstige gebruikers**
De toekomstige gebruiker zullen geïnterviewd worden om een duidelijk beeld te krijgen van de huidige situatie en de gewenste situatie.
- **Onderzoek mogelijkheden XML**
Voordat er aan een ontwerp en implementatie begonnen zal worden dient er eerst vooronderzoek uitgevoerd te worden. Dit vooronderzoek zal informatie opleveren over de mogelijkheden met XML en een mogelijke structuur.
- **Onderzoek mogelijkheden PHP**
Er zal onderzocht worden welke mogelijkheden PHP biedt met betrekking tot het veilig inloggen, DBMS onafhankelijk verbinding maken, scheiden van PHP code van de HTML code en de mogelijkheden van XML binnen PHP.
- **Ontwerpen database**
De database zal gebruikt worden om statische gegevens te beheren.
- **Ontwerpen XML structuur**
De structuur van de XML bestanden is van belang om een correcte implementatie te krijgen. De structuur zal uit twee delen bestaan. Eén gedeelte voor de werkelijke gegevens en één gedeelte voor de definitie van de werkelijke gegevens.
- **Ontwerp in fusebox**
Voordat er werkelijk met de implementatie begonnen kan worden zal er eerst een gestructureerd ontwerp volgens de fusebox ontwikkel methode gemaakt worden. Fusebox is een methode welke sinds lange tijd binnen Korton gebruikt word. Fusebox stelt de programmeur in staat om gestructureerd te ontwerpen en programmeren.
- **Implementatie in PHP**
De implementatie zal geschieden in PHP. PHP is een serverside scripting taal waarmee het mogelijk is om dynamische webpagina's te maken. PHP is zeer geschikt voor deze opdracht omdat het ook beschikt over een XML parser.

2.5 Eindsituatie

De opdracht is voltooid indien Korton een goed werkend systeem heeft die de huidige manier van werken kan overnemen. Naast het product moet de bijbehorende documentatie ook juist zijn. Het gehele systeemontwerp moet door alle medewerkers van Korton Internet te begrijpen zijn.

3. Producten

De volgende producten zullen worden opgeleverd aan Korton.

- Ontwerp en onderzoek documentatie
- Database
- Systeemontwerp in Fusebox
- Volledige sourcecode van de implementatie in PHP
- Gebruikershandleiding.
- Technische documentatie

4. Methoden en technieken

4.1 Methoden

Binnen de opdracht zal er gebruik gemaakt gaan worden van de FuseBox ontwikkel methode. Deze ontwikkelmethode stelt de ontwikkelaar in staat om gestructureerd een systeemontwerp te maken. Daarnaast sluit deze methode goed aan op de te gebruiken implementatie taal.

4.2 Technieken

Er zal gebruik gemaakt gaan worden van de beschikbare technieken binnen de FuseBox ontwikkelmethode. Alleen de technieken welke van belang zijn binnen het project zullen gebruikt gaan worden.

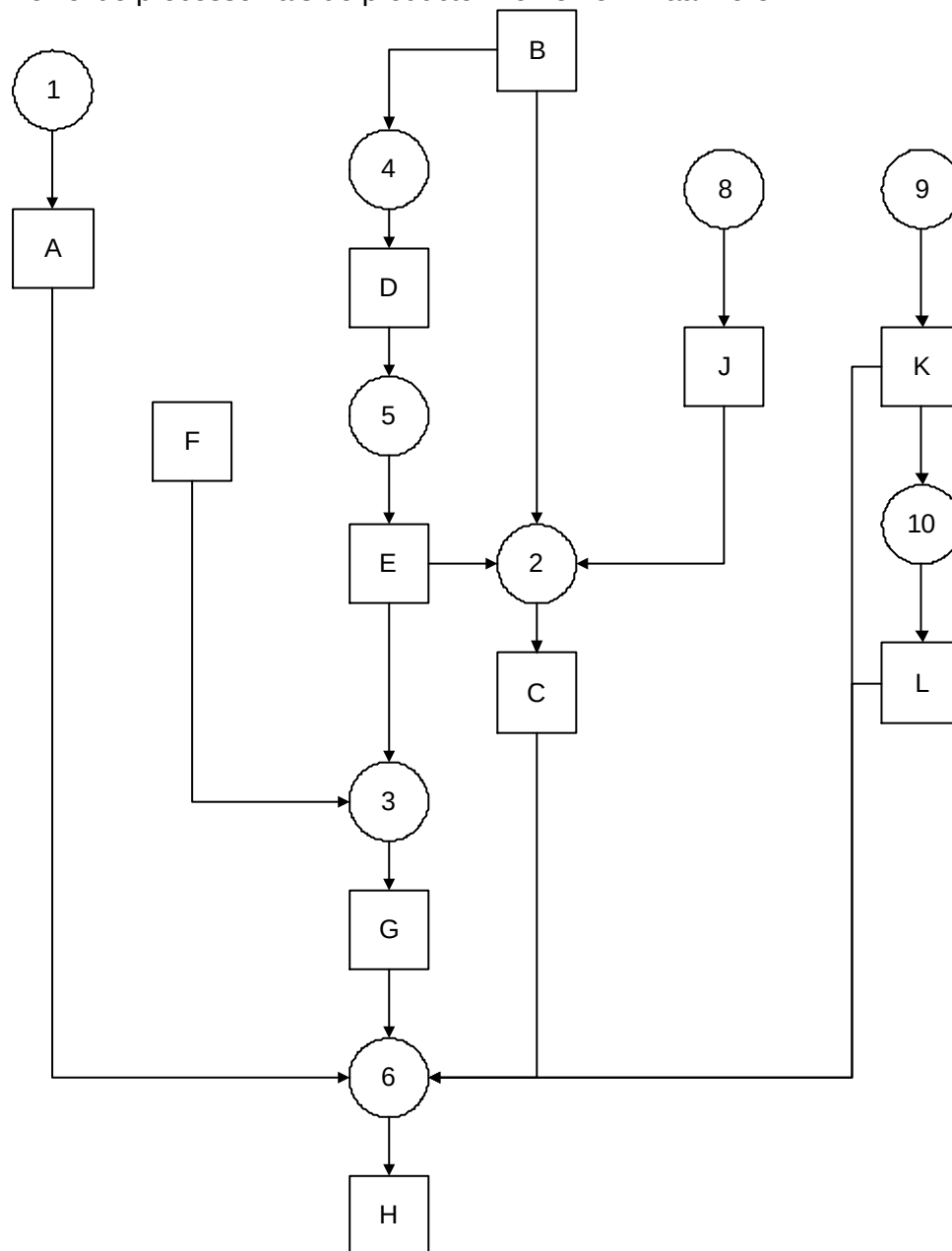
5. Risico's

De opdracht brengt enkele risico's met zich mee. Deze risico's dienen voor zover mogelijk vermeden te worden. De volgende risico's worden erkend.

- Te weinig kennis bij de afstudeerder.
De afstudeerder kan te kort kennis hebben over het onderwerp en over de gebruikte methoden / technieken en programmeertalen. Door middel van extra hulp zal Korton proberen zoveel mogelijk kennis over te dragen zodat de afstudeerder zo snel mogelijk op het juiste kennis niveau zit.
- Te kort tijd voor de opdracht
De opdracht kan achteraf meer tijd blijken te kosten dan van tevoren bekend is. Door onvoorziene omstandigheden zoals het extra tijd nodig hebben om te wennen aan de werkwijze binnen Korton kan er minder tijd overblijven voor de werkelijke opdracht.
- Gebruik maken van PHP 5
Het gebruik maken van de beta versie van PHP 5 is een mogelijk risico. Deze versie bevat nog vele bugs en wordt doorontwikkeld. Tijdens de doorontwikkeling kunnen onderdelen compleet veranderen met alle mogelijke gevolgen van dien. Dit risico is te vermijden door het lezen van de changelogs voordat de lokale versie geupdate gaat worden.

6. Precedentieschema

Het precedentieschema beschrijft de procesgang gedurende de gehele afstudeerperiode. Zowel de processen als de producten komen erin naar voren.



Figuur 18: Precedentieschema

Toelichting precedentieschema

1. Onderzoek naar het gebruik van XML als opslagstructuur
 2. Ontwikkelen XML datastructuur (vragen, inhoud vraag gedeelte)
 3. Samenstellen fusebox ontwerp
 4. Samenstellen vragenlijst
 5. Interviewen toekomstige gebruikers
 6. Implementatie van het fusebox ontwerp
 8. Ontwikkelen XML Meta structuur (structuur xml files)
 9. Ontwikkelen lock structuur XML bestanden
 10. Ontwikkelen database structuur
-
- A. Resultaten XML onderzoek
 - B. Huidige klantenmappingen
 - C. XML Datastructuur
 - D. Vragenlijst
 - E. Resultaten vragenlijst
 - F. Adelon
 - G. Fusebox ontwerp
 - H. KCC Klantenmap
 - J. XML Meta structuur
 - K. Lock structuur XML bestanden
 - L. Database ontwerp

7. Planning

Datum	Week	Taak	School
9 – 13 februari	1	Plan van aanpak Mogelijke opslagstructuur Mogelijkheden XML	
16 – 20 februari	2	Mogelijkheden XML Mogelijkheden Database Database ontwerp (inc erd) (ondersteuning van XML)	
23 – 27 februari	3	Login onderzoek (veilig inloggen) Template engines Onderzoek PEAR DB (onafhankelijk van je dbms) Samenstellen inhoud nieuwe klantenmap opzet van de nieuwe informatie	
1 – 5 maart	4	Interviewen gebruikers + verwerken resultaten Leren fusebox structuur Adalon ontwerp	
8 – 12 maart	5	Leren fusebox structuur Adalon ontwerp (login, klanten) Verslag (eerste structuur opzetten) Functie specificaties Scherm ontwerpen Samenstellen systeemontwerp	
15 – 19 maart	6	Adalon ontwerp (details, uren, todo, zoeken)	Bezoek examinatoren
22 – 26 maart	7	Verslag Implementatie (inloggen)	Bezoek examinatoren
29 maart – 2 april	8	Implementatie (inloggen) Opdrachtomschrijving	Definitieve opdrachtomschrijving
5 – 9 april	9	(voortgang) verslag Implementatie (klanten)	Voortgangsverslag
12 – 16 april	10	Implementatie Verslag (klanten)	Mijlpaal : Klanten werkend
19 – 23 april	11	Verslag (details) Implementatie	
26 – 30 april	12	Implementatie (details) Verslag	Mijlpaal : Details werkend
3 – 7 mei	13	Verslag (urendeclaratie) Implementatie	Mijlpaal: Uren werkend
10 – 14 mei	14	Implementatie (todo's) Opzetten testomgeving lokaal	Relevante delen eindverslag voor examinatoren Mijlpaal: ToDo werkend:

17 - 21 mei	15	Verslag Technische documentatie bijwerken Implementatie (zoeken) Oplossen van gevonden problemen tijdens test door gebruikers	Gesprek met examinatoren + bedrijfsmentor, doorspreken leerplan
24 – 28 mei	16	Implementatie (zoeken) Oplossen van gevonden problemen tijdens test door gebruikers Verslag Opzetten productieomgeving	Gesprek met examinatoren + bedrijfsmentor, doorspreken leerplan Mijlpaal: Zoeken afgerond.
31 mei – 4 juni	17	Verslag Laatste mogelijke bugs oplossen	
7 – 11 juni	18	Verslag	

Externe Bijlagen

KCC Klantenmap: Een Internetsite voor Korton Computer Communication B.V.

Externe Bijlage

Marcel Stokman



Periode:	9 februari 2004 t/m 11 juni 2004
Plaats:	Hoofddorp
Auteur:	Marcel Stokman
Opleiding:	Informatiekunde van de Haagse Hogeschool sector Informatica & Informatiekunde
Afstudeerrichting:	OSTI
Examinatoren:	Dhr. A.W.W.M. Biegstraaten Dhr. J. van Peski
Gastbedrijf:	Korton Internet
Bedrijfsmentor:	Dhr. M. Huijkman

Inhoudsopgave

1. XML.....	71
1.1 Locking	71
1.2 Backups.....	72
1.3 Directe toegang	72
1.4 Structuur.....	72
1.5 SimpleXML	74
2. Databaseontwerp.....	77
2.1 ERD	77
2.2 Tabellen.....	78
2.3 Indexen.....	80
3. Pilot.....	81
4. Schermontwerpen.....	81
4.1 Inlogschermb.....	89
4.2 Klant kiezen.....	89
4.3 Klant	90
4.4 Detail	90
4.5 Zoeken.....	91
4.6 Zoekresultaten	91
4.7 Urendecclaratie	92
5. Fusebox ontwerp.....	93
6. Vragenlijst KCC Klantenmap	100
7. Handleiding opzetten meta bestand.....	101
8. Testrapport	104

1. XML

1.1 Locking

Voor de opslag van de gegevens zal er gebruik gemaakt gaan worden van XML bestanden. Om te voorkomen dat er updates over elkaar heen gemaakt worden zullen de bestanden beschermt moeten worden door middel van een locking mechanisme. Voor dit locking mechanisme zal er gebruik gemaakt worden van een database. De volgende tabel is daarvoor ontwikkeld.

Tabel: locks : houdt bij welke xml bestanden in gebruik zijn.			
Veld	Type	Null toestaan	Opmerkingen
ID	Int(8)	Nee	Primaire sleutel, auto increment
GebruikerID	Int(8)	Nee	Vreemde sleutel → gebruiker
Bestand	Varchar(200)	Nee	
Tijd	Datetime	Nee	
KlantID	Int(8)	Nee	Vreemde sleutel → klant

Deze tabel houdt bij welk bestand er door een gebruiker in gebruik is. Een lock wordt op bestandsniveau neergezet en verschilt per klant. De tijd van plaatsing zal worden gebruikt om de lock automatisch na een aantal uur te verwijderen. Dit om te voorkomen dat een bestand altijd gelockt blijft.

Er zijn een aantal mogelijk gebeurtenissen mogelijk op een bestand wanneer er een lock op staat.

Gebruiker logt in

- Alle locks worden verwijderd

Gebruiker logt uit

- Alle locks worden verwijderd

De gebruiker klikt een willekeurige link

- Indien de gebruiker een lock bezit worden deze verwijderd

Onaardige logout dmv kruisje browser

- Na inloggen zijn alle locks verwijderd

Gebruiker Z probeert de gegevens te lezen terwijl deze gelockt is.

- Lezen is toegestaan.

Gebruiker Z probeert de gegevens te wijzigen terwijl er een lock op staat.

- Dit is niet toegestaan. De gebruiker krijgt een melding dat het bestand al bewerkt wordt door gebruiker Y

1.2 Backups

Om te voorkomen dat de data door een foutieve update verdwijnt zal er voor elke update een backup gemaakt worden van het bestand. Elke backup zal een versienummer mee krijgen om zo onderscheid te kunnen maken tussen de verschillende versies.

Het bestand zal de volgende syntax hebben: JJJJ-MM-DD UU-MM-SS Naam.xml

Waarbij JJJJ-MM-DD de datum is, UU-MM-SS de tijd en Naam het originele bestandsnaam.

1.3 Directe toegang

De bestanden staan op een webserver welke vanaf het Internet te bereiken is. Deze server is beveiligd om te voorkomen dat kwaadwillende alle bestanden kunnen zien. Er zal gebruik gemaakt worden van deze beveiliging om te voorkomen dat een XML bestand direct opgevraagd kan worden. De XML bestanden zullen daarom niet direct in de WWW root van de server staan maar in een aparte directory welke alleen door de scripts op de webserver te benaderen is.

1.4 Structuur

De gehele structuur is opgebouwd uit twee bestanden. Een bestand waarin de 'vragen' staan. Deze vragen bevatten gegevens over de structuur van het 'antwoord' en welke controle eroverheen moet. Het 'antwoord' gedeelte bevat het 'antwoord' op bijbehorende 'vraag'. Aan de hand van het ID kunnen 'vraag' en 'antwoord' aan elkaar gekoppeld worden.

1.4.1 Definitie 'vraag' structuur

De volledige structuur van het bron bestand is als volgt.

- *input*, een input is één record.
 - o *id*, een unieke combinatie van letters met als doel het record te identificeren.
 - o *text*, beschrijving van een kritisch kenmerk van een record. Bijvoorbeeld: naam.
 - o *hint*, specificatie van een record.
 - o *type*, specificatie van het soort record. Dit zijn: text, textarea, pulldown
 - o *size*, het maximaal aantal tekens dat een text record mag omvatten.
 - o *cols*, het maximaal aantal kolommen dat een texarea record groot is.
 - o *rows*, het maximaal aantal rijen dat een texarea record groot is.
 - o *values*, bepaalt de mogelijke keuzes binnen een pull down record. Elke keuze wordt gedefinieerd in een *val*.
 - o *val*, specificatie van een waarde binnen een pull down record.
 - o *required*, geeft aan of de invoer van een record verplicht is of niet. De mogelijke waarden zijn *y* of *n*.
 - o *defaultvalue*, geeft de advies waarde van een record aan.
 - o *displayonly*, geeft aan dat een record alleen getoond mag worden.
 - o *validation*, geeft aan welke invoer controle('s) er uitgevoerd moet worden op een record.
 - *cannotBeBlank*: veld mag niet leeg zijn.
 - *mustBeDate*: het veld moet datum formaat zijn.
 - *mustBeEmail*: het veld moet e-mail formaat zijn.
 - *cannotContainAlfa*: het veld mag geen letters bevatten.
 - *cannotContainNumber*: het veld mag geen letters bevatten.
 - *cannotContainSpecial*: het veld mag bepaalde speciale tekens niet bevatten (oa \$%^).

Een voorbeeld:

```
<input>
  <id>server_merk</id>
  <text>wat is het merk van de server</text>
  <hint>noem alleen het merk, niet het type. bv compac</hint>
  <type>text</type>
  <size>50</size>
  <required>y</required>
  <defaultvalue>HP</defaultvalue>
  <validation></validation>
</input>
```

1.4.2 Definitie 'antwoord' structuur

Opmerking:

Het bestand dat de werkelijke gegevens bevat bestaat uit de volgende onderdelen.

- *output*, een output is een blok met alle gegevens van het bron bestand.
 - o *Identifier*, een *identifier* identificeert het output blok. Er kunnen meerdere output blokken voorkomen in het bestand de *identifier* zorgt voor het verschil tussen de blokken
 - o *Antwoord*, een antwoord blok met daarin de waarde van één *input* blok.
 - *id*, de unieke id van het record.
 - *text*, de ingevoerde waarde

```
<output>
  <identifier>server 1</identifier>
  <antwoord>
    <id>server_merk</id>
    <text>hp</text>
  </antwoord>
</output>
```

1.5 SimpleXML

SimpleXML is een module binnen PHP voor het gebruik van XML bestanden. Deze module bevat functies welke het gebruik van XML vergemakkelijken. Enkele functies zijn het inlezen, het wegschrijven en het zoeken in bestanden.

1.5.1 Inlezen

Door middel van SimpleXML kan een XML bestand direct in het geheugen ingelezen worden. De functie `simpleXML_load_file` zorgt ervoor dat een XML bestand in het geheugen gezet wordt waarna deze direct aan te spreken is.

```
$xml_vraag = simpleXML_load_file("../xml bestanden/server.xml");
```

De functie `simpleXML_load_string` doet hetzelfde als de `load_file` variant alleen leest deze een string in. Deze kan gebruikt worden als het xml document geïnclude als variabele is.

De inhoud van het XML bestand kan direct aangesproken worden door middel van de variabele (`$xml_vraag` in het voorbeeld).

```
echo $xml_vraag->INPUT->TEXT;
```

Dit drukt de inhoud van de tag `<TEXT>` af op het scherm. Let wel op dat XML hoofdletter gevoelig is. Er zit een wezenlijk verschil tussen `<TEXT>` en `<text>`.

1.5.2 Zoeken

Het zoeken binnen een XML bestand (welke in het geheugen staat) gaat door middel van de functie xpath. Bij het zoeken in een resultaat moet de // weggehaald worden.

```
$resultaat = $xml_vraag->xpath('//*[ID[contains(text(),"MERK")]]');
```

Dit voorbeeld resulteert een object wat aan de eis voldoet. De gestelde eis is dat de tag <ID> een tekst staat waarin "MERK" voorkomt.

Voorbeeld:

```
<INPUT>
<ID>SERVER_MERK</ID>
</INPUT>
<INPUT>
  <ID>MERKLOOS</ID>
</INPUT>
```

Dit zouden twee mogelijke resultaten van de functie kunnen zijn. Xpath geeft altijd het moederobject terug.

Andere mogelijke implementaties van xpath zijn.

Selecteer het record waarvan het element 'ROEPNAAM' gelijk is aan 'Marcel'
 //*[ROEPNAAM = 'Marcel']

Selecteer het record waarvan het element 'ROEPNAAM' ergens de waarde 'Mar' heeft.
 //*[ROEPNAAM[contains(text(),'Mar')]]

Selecteer alle records met een element 'PRIVE'
 //*[PRIVE]

Selecteer alle records waarvan het element 'PRIVE' gevuld is.
 //*[PRIVE[string-length(text()) > 0]]

Onderdelen zijn ook te combineren tot complexere zoekopdrachten, de koppeling gaat door middel van | (of) of and.

1.5.3 Aanpassen

Het aanpassen van (een deel van) een XML kan doormiddel van een toekenningstatement.

```
foreach ($xml_vraag->xpath ('//INPUT') as $vraag) {
  $antwoord->NAAM = 'Jansen';
}
```

In dit voorbeeld zal elke naam gewijzigd worden naar 'Jansen'.

Na het aanpassen van een waarde zal het gehele bestand opnieuw weggeschreven moeten worden naar de schijf. Dit zal worden besproken in de volgende paragraaf.

1.5.4 Wegschrijven naar schijf

Nadat een XML bestand aangepast is (in het geheugen). Kan het bestand weer teruggeschreven worden naar de schijf. Voordat er weggeschreven gaat worden zal er eerst een bestand geopend (of gemaakt) moeten worden.

```
$file = fopen("./server.xml", "w");
```

Deze regel zorgt ervoor dat er een bestand (server.xml) geopend wordt op in te schrijven. De "w" geeft aan dat het om overschrijven gaat, andere opties zijn "a" voor append (toevoegen onderaan het bestand) en "r" (alleen lezen). Nadat het bestand geopend is kan er begonnen worden met het wegschrijven.

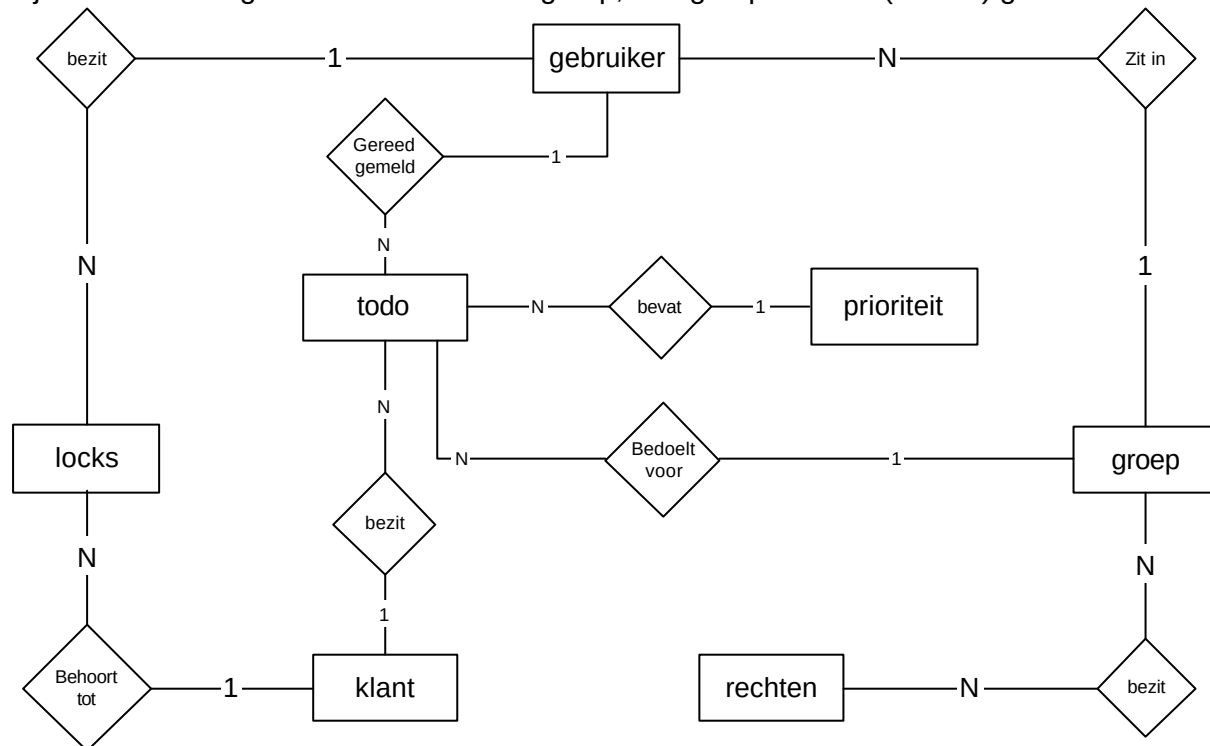
```
fputs($file, $xml_vraag->asXML());  
fclose($file);
```

Het wegschrijven wordt gedaan door middel van de functie *fputs* (*bestand*, *tekst*). Er zijn twee parameters vereist, namelijk het bestand waarin hij moet schrijven en de tekst welke hij moet wegschrijven. Het bestand is een file pointer naar het zojuist geopende bestand. De weg te schrijven tekst is een string. De functie *asXML()* zorgt ervoor dat het gehele ingelezen (en eventueel aangepaste) XML bestand weergegeven wordt als string, inclusief opmaak. Na het wegschrijven dient het bestand netjes afgesloten te worden, dit wordt gedaan door middel van de functie *fclose*(*bestand*).

2. Databaseontwerp

2.1 ERD

Een ERD geeft de onderlinge relaties tussen tabellen weer. Voor de leesbaarheid zijn de attributen van een tabel hier weggelaten. Er wordt van rechts naar links gelezen, bijvoorbeeld elke gebruiker zit in exact 1 groep, een groep bevat N ($N \geq 0$) gebruikers.



2.2 Tabellen

De volgende tabellen zullen de werking van het systeem ondersteunen, beveiligen en versnellen.

tabel: locks : houdt bij welke xml bestanden in gebruik zijn.			
Veld	type	null toestaan	opmerkingen
ID	Int(8)	nee	primaire sleutel, auto increment
GebruikerID	Int(8)	nee	vreemde sleutel → gebruiker
Bestand	varchar(200)	nee	
DatumTijd	datetime	nee	tijd+datum begin lock
KlantID	Int(8)	nee	vreemde sleutel → klant

Deze tabel houdt de gelockte bestanden bij.

tabel: klanten : houdt de klanten en hun directory bij			
Veld	type	null toestaan	opmerkingen
ID	Int(8)	Nee	primaire sleutel, auto increment
Naam	varchar(100)	Nee	
datumTijdWijziging	datetime	Nee	datum + tijd van de laatste wijziging
Directory	varchar(200)	Nee	
GebruikerID	Int(8)	Nee	Gebruiker laatste wijziging

Deze tabel houdt de klanten bij. Naast de volledige naam van de klant word ook de directory waar alle bestanden in staan bijgehouden. Tevens wordt er bij gehouden wanneer en door wie de laatste wijziging plaats vond.

tabel: gebruikers : houdt de gebruikers bij welke mogen (en kunnen) inloggen			
Veld	type	null toestaan	opmerkingen
ID	Int(8)	Nee	primaire sleutel, auto increment
Gebruikersnaam	varchar(50)	Nee	
WachtwoordMD5	varchar(32)	Nee	md5 versie wachtwoord
Naam	varchar(200)	Nee	volledige naam medewerker
GroepID	Int(8)	Nee	vreemde sleutel → groep
Geblokkeerd	bool	Nee	default false

In deze tabel staan alle mogelijke gebruikers van het systeem. Het wachtwoord wordt gecodeerd opgeslagen om te voorkomen dat deze vrij beschikbaar zijn tijdens een mogelijke inbraak. Het veld geblokkeerd geeft aan of een gebruiker mag inloggen of niet. Elke gebruiker zit in exact één groep.

tabel: groepen : verdeelt de gebruikers in groepen zodat de rechten eenvoudiger werken			
Veld	Type	null toestaan	opmerkingen
ID	Int(8)	Nee	primaire sleutel, auto increment
Naam	varchar(30)	Nee	
Rechten	Bigint	Nee	de waarde van de rechten. $7 = 4+2+1$

De tabel groepen groepeerde de gebruikers en hun rechten. Een groep kan nul of meer gebruikers bevatten en heeft een aantal rechten. De rechten worden via een binarie telling bijgehouden. Zo bestaat het recht 7 uit de rechten 4, 2 en 1. Deze rechten staan vermeld in onderstaande tabel.

tabel: rechten : geeft de rechten weer, per pagina.			
Veld	Type	Null toestaan	opmerkingen
ID	Bigint	Nee	bitwaarde, primaire sleutel
Omschrijving	varchar(30)	Nee	

In deze tabel staan de mogelijke rechten vermeld. Het id bestaat uit de bitwaarde van het getal. Er zijn totaal 64 mogelijke rechten beschikbaar. Dit is meer dan genoeg in de huidige omgeving omdat er gecombineerde rechten mogelijk zijn.

tabel: todo : hier worden alle todo's bijgehouden.			
Veld	Type	Null toestaan	Opmerkingen
Id	Int(8)	Nee	Primaire sleutel, auto increment
KlantID	Int(8)	Nee	Vreemde sleutel naar tabel klanten
Omschrijving	Text	Nee	Omschrijving van de uit te voeren werkzaamheden / klachten
Prioriteit	Int(8)	Nee	Vreemde sleutel naar tabel prioriteit
Aanmaakdatum	Date	Nee	Datum van aanmaak todo
Gereeddatum	Date	Ja	Datum wanneer de todo gereed gemeld is
Verloopdatum	Date	Ja	Datum wanneer de todo gereed moet zijn
Gereed	Tinyint(1)	Ja	Geeft aan of een todo gereed is (true/false)
Groepid	Int(8)	Nee	Geeft aan welke groep de toto aangemaakt heeft. Is aanwezig als voorbereiding op de rechten.
Oplossing	Text	Ja	De oplossing van het probleem / klacht
Gebruikerid	Int(8)	Ja	De gebruiker die de todo gereed gemeld heeft.

In deze tabel worden de todo's bijgehouden. Een todo hoort altijd bij een klant.

Er is nog geen rekening gehouden met de wens dat een klant alleen de todo's ziet die hijzelf ingevoerd heeft en een kcc medewerker alle todo's ziet. In het ontwerp wel maar binnen de implementatie niet.

tabel: prioriteit : geeft de verschillende prioriteiten weer.			
Veld	Type	Null toestaan	Opmerkingen
ID	Int(8)	Nee	Primaire sleutel, auto increment
Omschrijving	varchar(30)	Nee	

De prioriteiten welke bekend zijn. Deze zijn afgeleid van de applicatie Q/A welke gebruikt wordt binnen Korton om de te maken aanpassingen aan de software bij te houden.

2.3 Indexen

Om de snelheid te verhogen zijn er indexen aangebracht op de lock tabel. De lock tabel zal vaak benaderd worden en niet altijd direct op de primaire sleutel. De velden waarop een secundaire index gezet is zijn: gebruikerid, klantid en bestand. Op deze drie velden zal het meest gezocht worden. Na de implementatie zal er gekeken worden naar andere mogelijke langzame queries en de mogelijkheden om deze met een index te versnellen.

3. Pilot

De volgende algemene procedures zullen binnen het systeem gebruikt worden. De procedures zijn algemeen opgeschreven. Er zijn procedures welke intern andere procedures gebruiken. De exacte werking wordt niet omschreven maar slechts de globale werking om een idee te schetsen wat een procedure doet. De beschreven procedures vormt de basis voor de werking van het systeem. Tijdens de implementatie zullen de procedure beschrijving zoveel mogelijk gehandhaafd en indien nodig bijgewerkt worden.

Inloggen	
Omschrijving:	De gebruiker logt in op het systeem
Procedure:	7) Gebruiker voert zijn gebruikersnaam en wachtwoord in 8) De gebruiker drukt op de knop inloggen 9) Het systeem controleert de volgende gegevens: Gebruikersnaam, wachtwoord, vergrendeld en groep. <i>Bij correcte gegevens</i> 10) Het systeem maakt de sessie aan 11) Het systeem zet de gegevens in de sessie 12) Het systeem stuurt de gebruiker door naar de klanten pagina indien deze geen klant is. <i>Bij incorrecte gegevens</i> 4) Het systeem stuurt de gebruiker terug naar de inlog pagina
Aannames:	Geen
Uitzonderingen:	De gebruiker heeft foutieve gegevens ingevoerd of is geblokkeerd.
Resultaat:	De gebruiker is ingelogd.

Uitloggen	
Omschrijving:	De gebruiker logt zichzelf uit.
Procedure:	1) De gebruiker drukt op de knop om uit te loggen. 2) Het systeem vernietigt de sessie gegevens. 3) Het systeem stuurt de gebruiker naar de inlog pagina.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Geen
Resultaat:	De gebruiker is uitgelogd.

IsIngelogd	
Omschrijving:	Er wordt gecontroleerd of de gebruiker ingelogd is.
Procedure:	1) Het systeem kijkt of de sessie bij de gebruiker bestaat en de juiste gegevens bevat. 2) Het systeem geeft terug of de gebruiker ingelogd is of niet.
Aannames:	Geen
Uitzonderingen:	Geen
Resultaat:	Er wordt terug gegeven of de gebruiker ingelogd is of niet.

ToonKlanten	
Omschrijving:	De klanten worden op het scherm getoond
Procedure:	1) Het systeem controleert de rechten 2) Het systeem haalt alle klanten op. 3) Het systeem toont de klanten op het scherm.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten.
Resultaat:	De klanten staan op het scherm

ToonKlant	
Omschrijving:	De gekozen klant wordt getoond.
Procedure:	1) Het systeem haalt de gegevens op. 2) Het systeem controleert de integriteit. 3) Het systeem toont deze gegevens. 4) Het systeem haalt de aanwezige detail onderdelen op. 5) Het systeem toont een link naar deze detail pagina's.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De integriteit klopt niet, dit wordt weergegeven op het scherm
Resultaat:	De klantgegevens staan op het scherm.

WijzigenKlant	
Omschrijving:	De gekozen klant zal worden gewijzigd.
Procedure:	1) Het systeem controleert de rechten. 2) De gebruiker kan de klantnaam wijzigen.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Gebruiker heeft onvoldoende rechten.
Resultaat:	De klant kan worden gewijzigd.

ToevoegenKlant	
Omschrijving:	De gebruiker kan een nieuwe klant aanmaken.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem controleert of er een lock op staat 3) Het systeem haalt de gegevens op. 4) De vragen worden getoond waarna ze kunnen worden gevuld met gegevens.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Het bestand is al gelockt.
Resultaat:	De klant kan worden opgeslagen.

OpslaanWijzigenKlant	
Omschrijving:	De gewijzigde klantnaam wordt opgeslagen.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem wijzigt de klantnaam in de database.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft geen rechten op de pagina.
Resultaat:	De klant is opgeslagen.

OpslaanToevoegenKlant	
Omschrijving:	De klant wordt toegevoegd.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem controleert of de verplichte gegevens ingevuld zijn 3) Het systeem maakt de klant aan in de database 4) Het systeem maakt de directory en subdirectory aan op het bestandssysteem. 5) Het systeem stuurt de gebruiker door naar de detail toevoegen pagina
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft geen rechten op de pagina.
Resultaat:	De klant is toegevoegd en de gebruiker kan de detail gegevens toevoegen.

ToonDetail	
Omschrijving:	De gekozen detail pagina wordt getoond.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem haalt de gekozen detail gegevens. 3) Het systeem controleert de integriteit. 4) Het systeem toont deze gegevens.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. De integriteit is niet correct, dit wordt weergegeven op het scherm.
Resultaat:	De detail pagina wordt getoond.

WijzigDetail	
Omschrijving:	De gekozen detail zal worden gewijzigd.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem controleert of er een lock aanwezig is 3) Het systeem zet een lock 4) Het systeem haalt de gegevens op. 5) Het systeem controleert de integriteit. 6) De gegevens worden getoond op het scherm en kunnen worden aangepast.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is al een lock aanwezig. De integriteit is niet correct, dit moet verplicht aangepast worden.
Resultaat:	De detail kan worden aangepast.

ToevoegenDetail	
Omschrijving:	De gebruiker kan een nieuwe detail aanmaken.
Procedure:	1) Het systeem controleert of de rechten 2) Het systeem controleert of er een lock aanwezig is. 3) Het systeem haalt de gegevens op. 4) Het systeem toet de vraag gegevens waarna de gegevens ingevoerd kunnen worden.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is al een lock aanwezig.
Resultaat:	De details kunnen worden opgeslagen.

OpslaanWijzigenDetail	
Omschrijving:	De gewijzigde details worden opgeslagen.
Procedure:	1) Controleren of de gebruiker de lock bezit. 2) lock neerzetten. 3) Het systeem controleert de rechten. 4) Het systeem controleert de invoer. 5) Het systeem maakt een backup van de huidige versie. 6) Het systeem slaat de details op. 7) Het systeem verwijdert de lock
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is iets niet correct ingevuld. Het bestand is gelockt door een andere gebruiker.
Resultaat:	De details zijn opgeslagen

OpslaanToevoegenDetail	
Omschrijving:	De toegevoegde details worden opgeslagen.
Procedure:	1) Controleren of de gebruiker de lock bezit. 2) lock neerzetten. 3) Het systeem controleert de rechten. 4) Het systeem controleert de invoer. 5) Het systeem maakt een backup van de huidige versie indien het bestand al aanwezig is. 6) Het systeem maakt het bestand aan indien deze niet aanwezig is 7) Het systeem voegt de detail toe aan het bestand 8) Het systeem slaat het bestand op 9) Het systeem verwijdert de lock
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is iets niet correct ingevuld. Het bestand is gelockt door een andere gebruiker.
Resultaat:	De details zijn opgeslagen

Zoeken	
Omschrijving:	De gebruiker kan in de klanten zoeken
Procedure:	1) Het systeem controleert de rechten. 2) De gebruiker voert in op welk woord hij wilt zoeken 3) De gebruiker kiest waarin hij wilt zoeken 4) De gebruiker drukt op de zoek knop. 5) Het systeem bepaald in welke klanten er gezocht moet worden 6) Het systeem bepaald in welke details er gezocht moet worden 7) Het systeem zoekt in de bepaalde onderdelen 8) Het systeem resulteert de gevonden resultaten.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten.
Resultaat:	De gevonden resultaten (0 of meer) worden getoond.

ZetLock	
Omschrijving:	Een bestand wordt gelockt.
Procedure:	1) Systeem controleert of het bestand al gelockt is. 2) Het systeem zet de lock neer 3) Het systeem geeft terug dat het gelukt is (of mislukt)
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Het bestand is al gelockt door een andere gebruiker. Het bestand is niet gevonden.
Resultaat:	Het bestand is gelockt.

VerwijderLock	
Omschrijving:	Een lock wordt verwijderd.
Procedure:	1) Het systeem controleert of het bestand gelockt is 2) Het systeem verwijdert de lock 3) Het systeem geeft terug of het gelukt is (of mislukt)
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	
Resultaat:	Het bestand is niet meer gelukt.

ControleerLock	
Omschrijving:	Er wordt gecontroleerd of er een lock aanwezig is op het bestand.
Procedure:	1) Het systeem controleert of er een lock aanwezig is op het bestand. 2) Het systeem geeft terug of het bestand gelockt is (of niet)
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Gevraagde bestand niet gevonden.
Resultaat:	True of false

IsEigenLock	
Omschrijving:	Er wordt gecontroleerd de aanwezige lock van jezelf is of van een andere gebruiker.
Procedure:	1) Het systeem controleert of er een lock aanwezig is op het bestand. 2) Het systeem controleert wie de lock bezit. 3) Het systeem geeft true terug indien het je eigen lock is, false indien het een lock van een andere gebruiker is.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Gevraagde bestand niet gevonden.
Resultaat:	True of false

VerwijderEigenLock	
Omschrijving:	Alle locks van de gebruiker worden verwijderd.
Procedure:	1) Het systeem verwijdert alle locks van de ingelogde gebruiker.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	
Resultaat:	

EigenaarLock	
Omschrijving:	De eigenaar van de lock wordt opgehaald
Procedure:	1) Het systeem haalt de eigenaar van de lock op 2) Het systeem geeft de naam van de eigenaar terug indien bestand gelockt is 3) Het systeem geeft niets terug indien hij niet gelockt is.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	
Resultaat:	

controleerInvoer	
Omschrijving:	De invoer wordt gecontroleerd op de integriteit.
Procedure:	1) Het systeem controleert of elke vraag een antwoord (al dan niet leeg) bevat. 2) Het systeem controleert of de invoer voldoet aan de gestelde types (datum etc). 3) Het systeem controleert of verplichte velden lege invoer bevat. 4) Het systeem retourneert de resultaten.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De invoer is niet correct.
Resultaat:	True indien correct, false indien fouten

ToonDeclaraties	
Omschrijving:	De gevonden declaraties worden getoond
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem haalt de aanwezige declaraties op. 3) Het systeem toont de declaraties
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten.
Resultaat:	De declaraties pagina wordt getoond.

ToonDeclaratie	
Omschrijving:	De gekozen declaratie wordt getoond
Procedure:	1) Het systeem controleert de rechten 2) Het systeem haalt de gekozen declaratie op. 3) Het systeem toont de declaratie
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten.
Resultaat:	De declaratie pagina wordt getoond.

WijzigDeclaratie	
Omschrijving:	De gekozen urendeclaratie zal worden aangepast
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem controleert of de declaratie geblokkeerd is. 3) Het systeem haalt de gegevens op. 4) De gegevens worden getoond op het scherm en kunnen worden aangepast.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. De declaratie is geblokkeerd
Resultaat:	De declaratie kan worden aangepast.

ToevoegenDeclaratie	
Omschrijving:	De gebruiker kan een nieuwe declaratie aanmaken.
Procedure:	1) Het systeem controleert of de rechten 2) Het systeem haalt de gegevens op. 3) Het systeem toet de vraag gegevens waarna de gegevens ingevoerd kunnen worden.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten.
Resultaat:	De declaratie kan worden opgeslagen.

OpslaanToevoegenDeclaratie	
Omschrijving:	De details worden opgeslagen.
Procedure:	1) Het systeem controleert de rechten. 2) Het systeem controleert de invoer. 3) Het systeem maakt de declaratie bestand aan 4) Het systeem schrijft de declaratie gegevens weg naar het bestand.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is iets niet correct ingevuld.
Resultaat:	De declaratie is opgeslagen

OpslaanWijzigenDeclaratie

Omschrijving:	De details worden opgeslagen.
Procedure:	1) Het systeem controleert of de declaratie niet vergrendeld is. 2) Het systeem controleert of de declaratie gelockt is 3) Het systeem zet een lock 4) Het systeem controleert de rechten. 5) Het systeem controleert de invoer. 6) Het systeem slaat de declaratie op 7) Lock verwijderen
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	De gebruiker heeft onvoldoende rechten. Er is iets niet correct ingevuld.
Resultaat:	De declaratie is opgeslagen

VergrendelenDeclaratie

Omschrijving:	De declaratie wordt vergrendeld zodat deze niet meer gewijzigd kan worden.
Procedure:	1) Controleren of de declaratie al vergrendeld is 2) Controleren of de declaratie gelockt is 3) Declaratie locken 4) Declaratie op vergrendeld zetten 5) Declaratie wegschrijven naar schijf 6) Lock verwijderen
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Bestand is gelockt of vergrendeld.
Resultaat:	De declaratie is vergrendeld

AfdrukkenDeclaratie

Omschrijving:	De declaratie wordt afgedrukt.
Procedure:	1) Het systeem controleert of de declaratie geblokkeerd is 2) Het systeem blokkeert de declaratie 3) Het systeem drukt de declaratie af.
Aannames:	De gebruiker is ingelogd.
Uitzonderingen:	Er is geen printer geïnstalleerd
Resultaat:	De declaratie is afgedrukt

4. Schermontwerpen

Dit hoofdstuk bevat het eerste concept van de schermontwerpen. De uiteindelijke implementatie zal nog aangepast worden naar de wensen de uiteindelijke gebruikers. Tevens zal de grafische expert er nog een kritische blik op werpen om tot een consistent en duidelijk geheel te komen.

4.1 Inlogscherf

KCC Klantenmap

Gebruikersnaam:

Wachtwoord:

[© Korton Internet 2004](#)

4.2 Klant kiezen

iets van een KCC klantenmap logo

Uitloggen
Zoeken

Kies een klant:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A-TAX
Amsterdamse taxicentrale
etc.. (alles met een A als eerste letter)

4.3 Klant

iets van een KCC klantenmap logo

	Klantnaam	
Uiloggen		
Zoeken		
klant	Adres	: ergens 1234
kiezen	Postcode	: 1234rp
	Woonplaats	: hoofdorp
	Contactpersoon	: Jan
	Telefoon	: 061234567
Algemeen		
Servers		
ToDo		
etc..		
etc..		

4.4 Detail

iets van een KCC klantenmap logo

	Klantnaam : Detail	
Uiloggen		
Zoeken		
klant	Server 1	
kiezen	Merk : merk a	
	Type : type b	
	etc	
Algemeen		
Servers		
ToDo		
etc..		
etc..		

4.5 Zoeken

iets van een KCC klantenmap logo

Zoeken

Uiloggen	Trefwoord:	
Zoeken		
Klant kiezen	Klanten	Alle Klanten klant a klant b etc
Algemeen	Details	Alle details Algemeen Servers ToDo Urenverantwoording
Servers		
ToDo		
etc..		
etc..		
	Zoeken	

4.6 Zoekresultaten

iets van een KCC klantenmap logo

Zoekresultaten

Uiloggen	
Zoeken	
Klant kiezen	<Klantnaam>
	<Detail>
	<Detail>
Algemeen	Atax
Servers	Servers
ToDo	ToDo
etc..	Urenverantwoording

4.7 Urendeclaratie



Uren declaratieformulier

Firma: korton internet

Medewerker: MarcelS

Aanspreekpartner: Bart

Datum: 16-03-2004

Straat: Jadelaan 99

Melding: fax

Postcode / Plaats: 2123 XZ

Doorgevoerde werkzaamheden:

Installatie wintaxserver	▲
Installatie mailserver	▼
Installatie 3 werkstations	▼
◀ ▶	▶

Geleverde / uitgeleende onderdelen +s/n

Voor installatie (soort/plaats):

is reeds gefactureerd	▲
◀ ▶	▶

Gewerkte uren per persoon:

Marcel stokman. Van: 9.00 Tot: 17.30 Pauze: 0.30 Totaal: 8.00	▲
Marcel Huijkman. Van: 9.00 Tot: 17.30 Pauze: 0.30 Totaal: 8.00	▼
◀ ▶	▶

Materiaalen voorrijkosten en BTW zijn niet inbegrepen tenzij expliciet in de service-overeenkomst hierover een regeling is getroffen. Als Cliënt een service-overeenkomst heeft geldt bij overschrijding van de maandelij kse uren het Extra-uren tarief volgens de service-overeenkomst. Anders gelden de prijzen volgens de prijslijst van Korton Groep bv.

Het bedrag wordt gefactureerd of d.m.v. incasso verrekend.

De werkzaamheden werden tot onze tevredenheid uitgevoerd

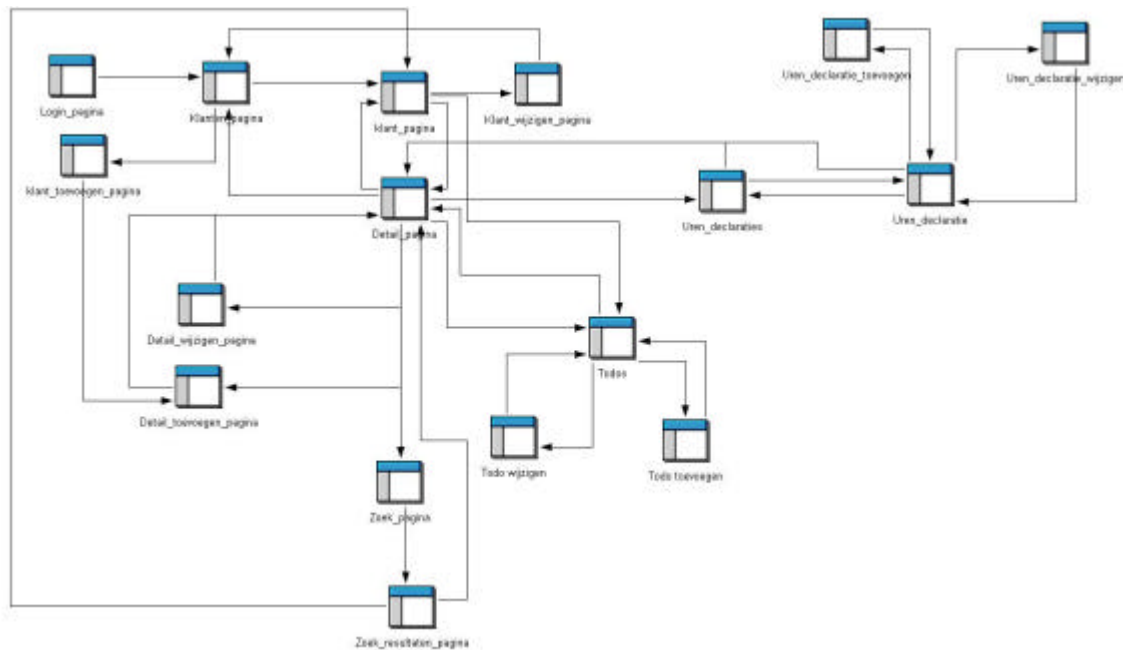
Datum:

Handtekening:

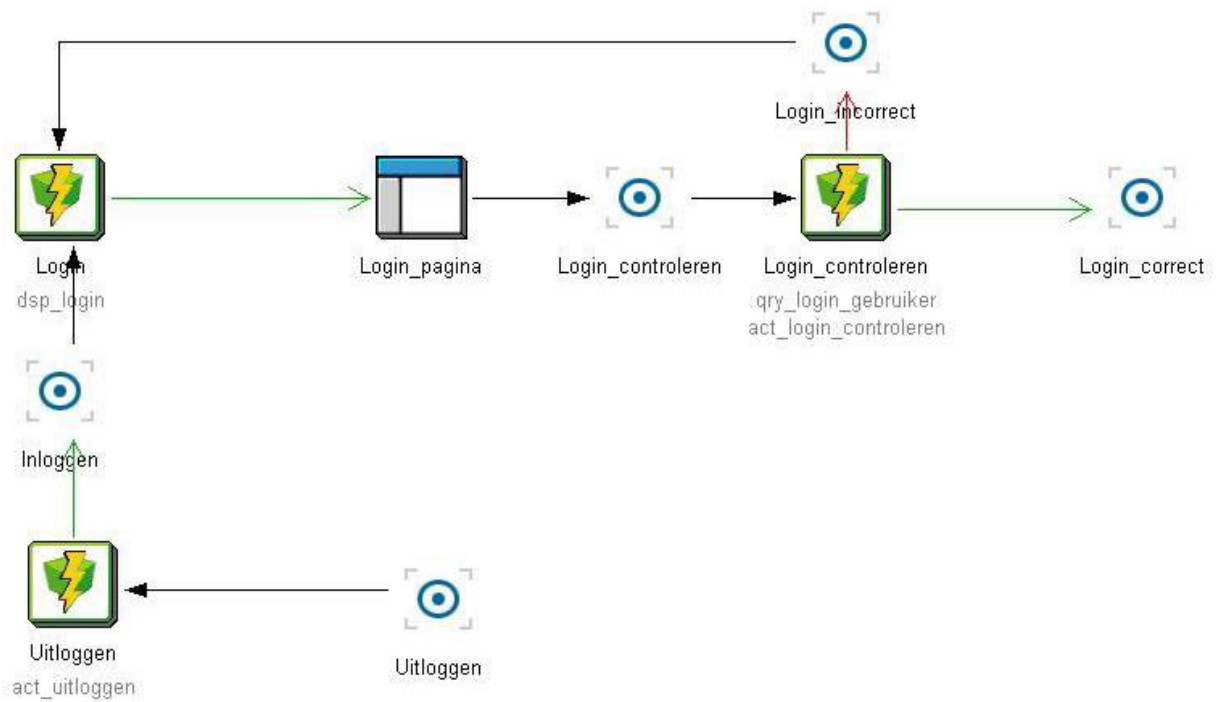
17-03-2004

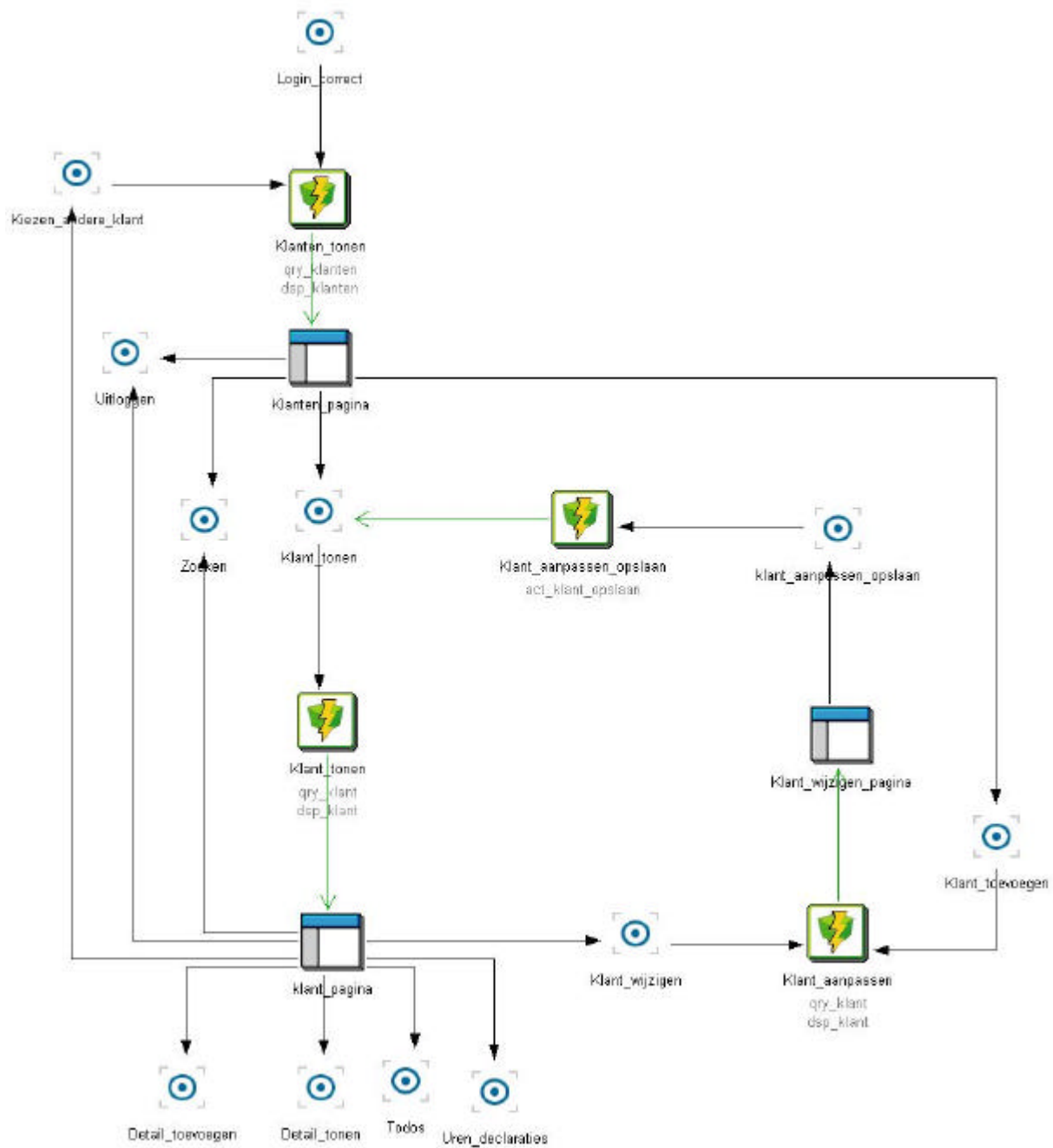
5. Fusebox ontwerp

Voor het maken van het Fusebox is er gebruik gemaakt van Adalon. Adalon is een case tool welke gemaakt is voor Fusebox. Adalon biedt de mogelijkheid om je framework te genereren en de structuur van je site alvast door te lopen zonder zelf (veel) code te schrijven. Vanaf de volgende pagina bevinden zich enkele prints van het Adalon ontwerp. Het gehele ontwerp is beschikbaar binnen Korton.

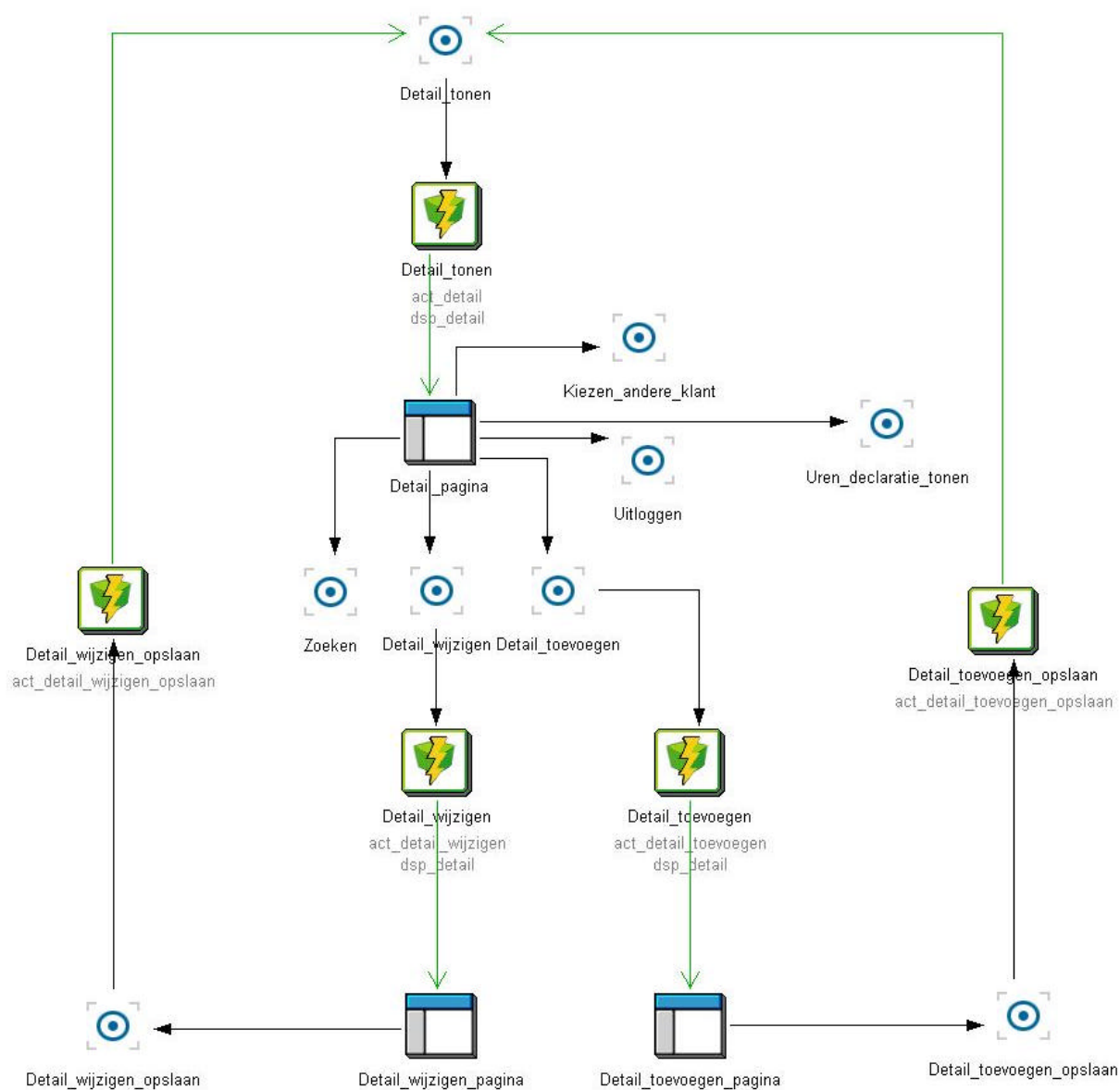


Figuur 19: Structuur

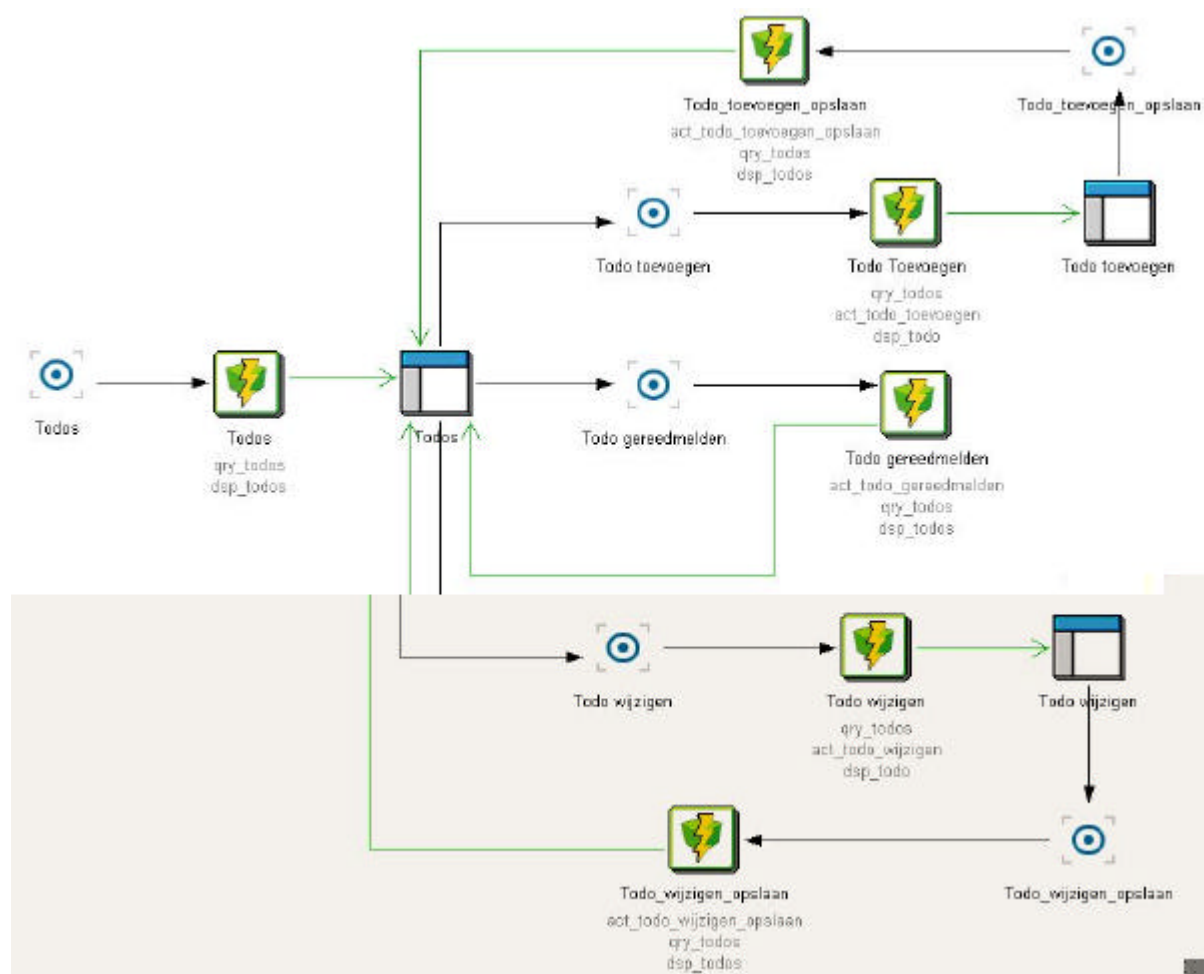
**Figuur 20: Login pagina**



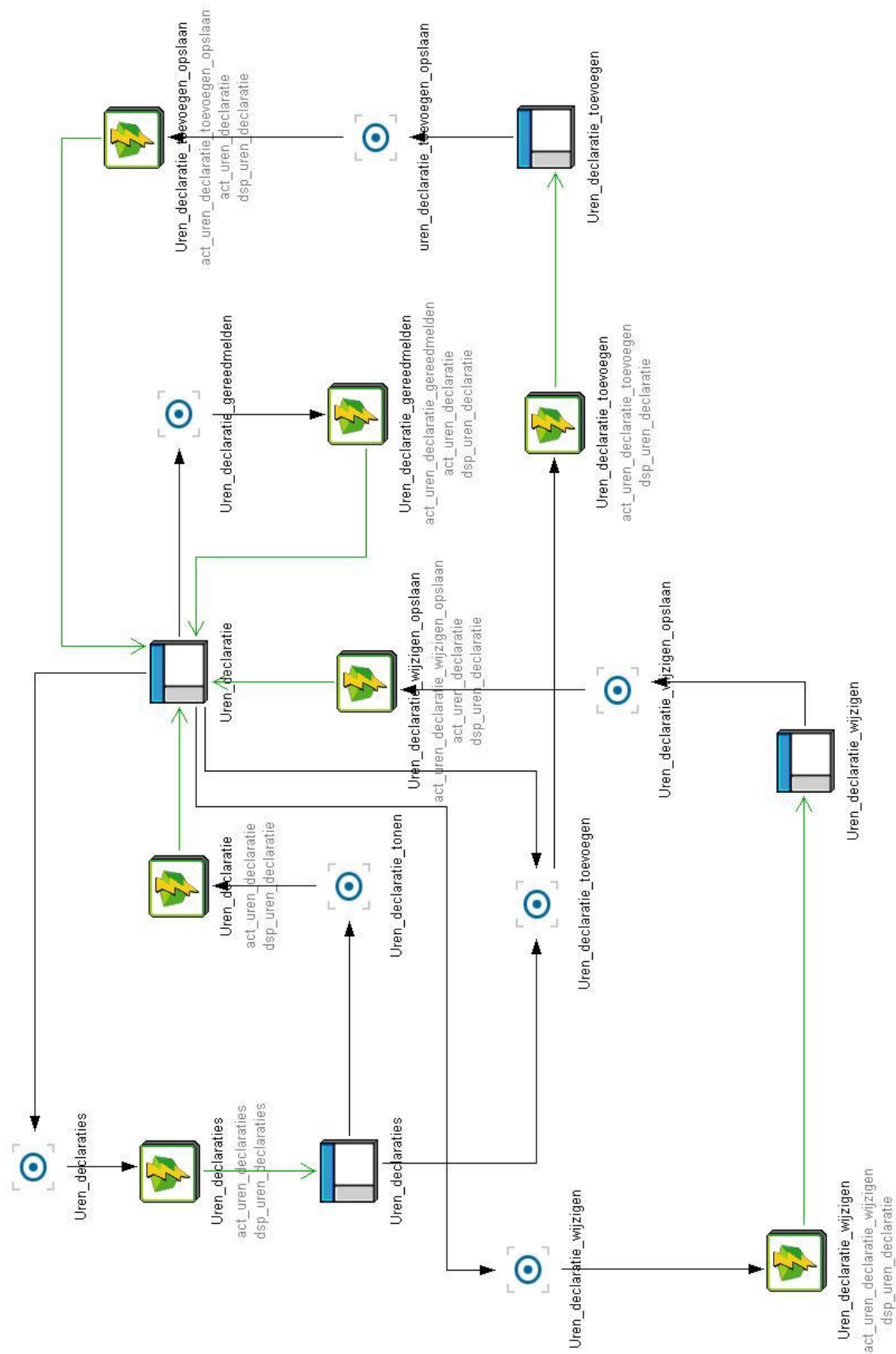
Figuur 21: Klantenpagina



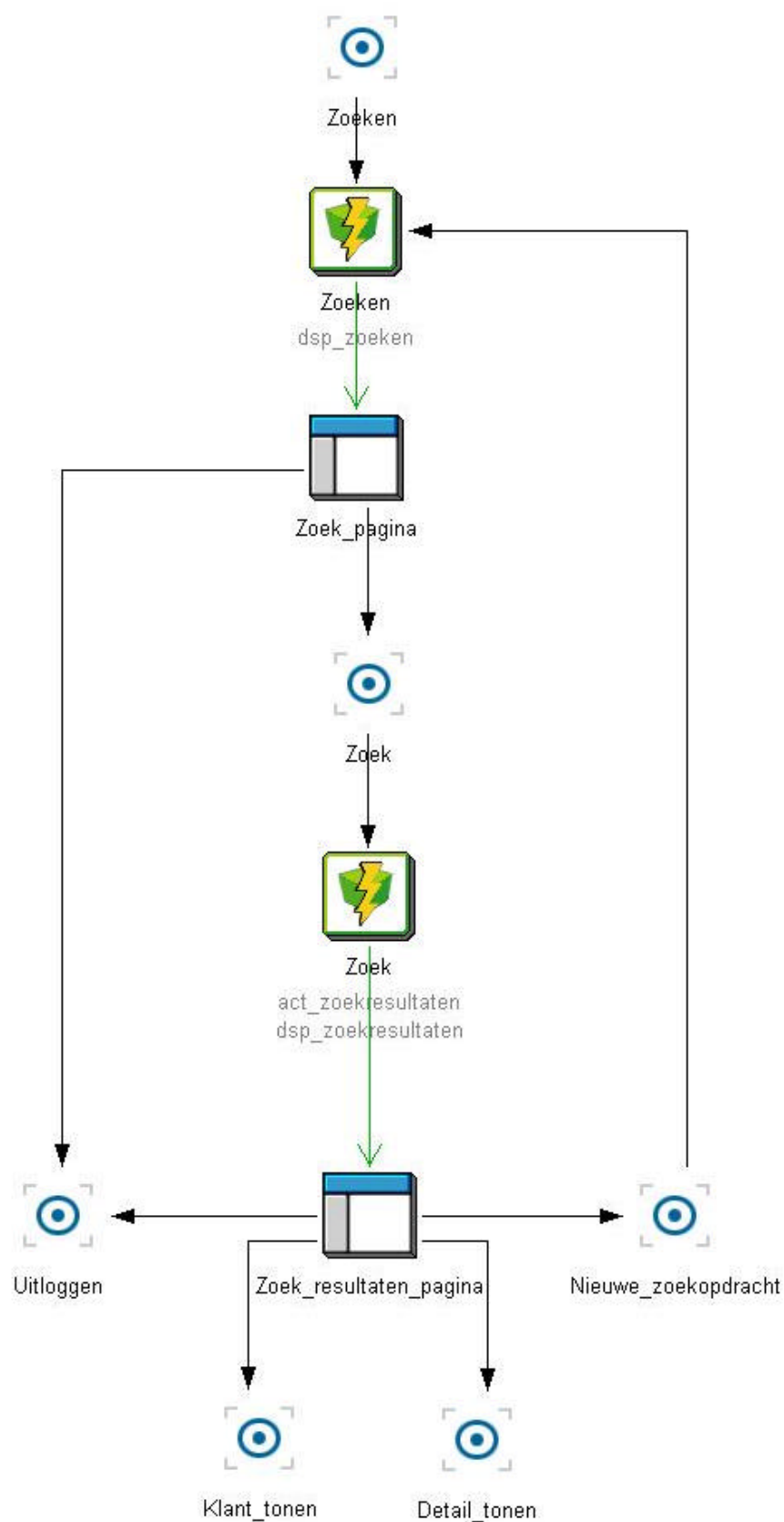
Figuur 22: Detail pagina



Figuur 23: Todo pagina



Figuur 24: Urendeclaratie

**Figuur 25: Zoek pagina**

6. Vragenlijst KCC Klantenmap

1. Hoe wordt het netwerk structuur bijgehouden.
2. Welke softwarematige instellingen moeten in de klantenmap bewaard worden.
3. Welke gegevens hebben jullie nodig voor het gebruik van Remote Login.
4. Welke gegevens zijn belangrijk van het geheugen van de server? Bv. In welke geheugen bank hij zit.
5. Welke gegevens zijn belangrijk van de harde schijf van de server.
6. Is de partitie indeling op een server of workstation van belang. Met andere woorden, moet deze opgeslagen worden in de klantenmap.
7. Welke randapparatuur onderscheiden jullie zoal. En welke gegevens zijn van belang.
8. Is het handig om een mogelijkheid tot het uploaden van bestanden aanwezig te hebben. (zo ja, wat voor bestanden zullen dat zijn, extensies, grootte, lengte bestandsnaam)
9. Zijn er nog wensen met betrekking tot de mogelijkheden?

7. Handleiding opzetten meta bestand

Een invoer XML bestand omschrijft de invoer. De volgende onderdelen moeten beschrijven worden. Allen onderdelen moeten in kleine letters geschreven worden anders werkt de applicatie niet.

id

Een unieke ID van het veld. Is verplicht

text

Een omschrijving van het veld, is verplicht. Bv Klantnaam, servermerk etc

type

Hier wordt het type omschreven, is verplicht. Mogelijke types zijn: text, textarea, pulldown.

size

De maximale lengte van het veld. Alleen bij het type text

values

Dit is een blok met alle mogelijke keuzen. Alleen bij pulldown

val

Hierin staat exact één mogelijk waarde van de pulldown. Er kunnen meerdere val tags voorkomen in een values blok. Alleen bij pulldown en bevind zich altijd in een values blok.

rows

Dit geeft aan hoeveel regels het textarea groot is op het scherm. Alleen bij textarea

cols

Dit geeft aan hoeveel kolommen het texarea groot is op het scherm. Alleen bij textarea

hint

Een mogelijke hint, deze mag leeg zijn.

displayonly

Dit geeft aan of een veld alleen getoond mag worden. Tijdens het toevoegen kan hij ingevuld worden, hij kan daarna echter niet meer gewijzigd worden. Geldige waarden zijn 'y' en 'n' (kleine letters!)

defaultvalue

Een standaard waarde voor een veld. Deze mag leeg zijn.

required

Geeft aan of het veld verplicht is of niet. Mogelijke waarden zijn "y" en "n" (kleine letters!). Dit veld mag niet leeg zijn.

validation

Dit veld geeft aan welke controles er over de invoer moeten voordat deze correct is. De mogelijke invoercontroles worden later beschreven. Het veld mag leeg zijn, dan zijn er geen controles.

Validation mogelijkheden

De volgende invoer controles zijn mogelijk. Deze is niet hoofdletter gevoelig. Er zijn meerdere invoer controles mogelijk, om dit te gebruiken dienen ze gescheiden worden door een komma (,).

cannotBeBlank

Het veld mag niet leeg zijn. Deze is nodig indien required op 'y' staat.

cannotContainAlpha

Het veld mag geen letters bevatten. Het gaat om alle hoofd en/of kleine letters.

cannotContainNumber

Het veld mag geen nummers bevatten. Het gaat om de getallen 0-9 of een combinatie daarvan (dus ook 766435).

cannotContainSpecial

Het veld mag geen speciale tekens bevatten. De volgende tekens mogen dan niet ~\$^*{}V voorkomen in de invoer.

mustBeDate

Het veld moet voldoen aan het datum formaat. Dit gaat om het DD-MM-YYYY formaat.

mustBeEmail

Het veld moet een e-mail formaat hebben. Hij controleert niet of een e-mail adres bestaat maar of de structuur overeen komt met een e-mail adres.

marcels@korton.nl is correct terwijl marcels@korton niet correct is

mustBeTime

Het veld moet een tijd formaat hebben. Dit gaat om het HH:MM formaat. Seconden worden niet meegenomen in de controle.

Voorbeelden

Pulldown

```
<input>
  <id>klant_wintax</id>
  <text>wintax:</text>
  <type>pulldown</type>
  <hint></hint>
  <values>
    <val>ja</val>
    <val>nee</val>
  </values>
  <defaultvalue>nee</defaultvalue>
  <displayonly>n</displayonly>
  <required>y</required>
  <validation>cannotbeblank</validation>
</input>
```

text

```
<input>
  <id>klant_naam</id>
  <text>naam:</text>
  <type>text</type>
  <hint>De naam</hint>
  <displayonly>y</displayonly>
  <size>50</size>
  <defaultvalue></defaultvalue>
  <required>y</required>
  <validation>cannotbeblank, cannotcontainnumber</validation>
</input>
```

Textarea

```
<input>
  <id>klant_opmerkingen</id>
  <text>Opmerkingen:</text>
  <type>textarea</type>
  <hint></hint>
  <rows>10</rows>
  <cols>40</cols>
  <displayonly>n</displayonly>
  <defaultvalue>Geen bijzonderheden te melden.</defaultvalue>
  <required>n</required>
  <validation></validation>
</input>
```

8. Testrapport

Onderdeel:	Inloggen
Invoer:	Foutieve inlognaam correct wachtwoord
Verwachte uitvoer:	Gebruiker blijft op de inlogpagina
Gekregen uitvoer:	Gebruiker blijft op de inlogpagina
Opmerking:	Een opmerking dat de inlog niet correct was zou net wezen

Onderdeel:	Inloggen
Invoer:	Correcte inlognaam, foutieve wachtwoord
Verwachte uitvoer:	Gebruiker blijft op de inlogpagina
Gekregen uitvoer:	Gebruiker blijft op de inlogpagina
Opmerking:	

Onderdeel:	Inloggen
Invoer:	Foutieve inlognaam en foutieve wachtwoord
Verwachte uitvoer:	Gebruiker blijft op de inlogpagina
Gekregen uitvoer:	Gebruiker blijft op de inlogpagina
Opmerking:	

Onderdeel:	Inloggen
Invoer:	Inloggen met correcte login
Verwachte uitvoer:	Klantenpagina
Gekregen uitvoer:	Klantenpagina
Opmerking:	

Onderdeel:	Inloggen
Invoer:	Inloggen met correcte login, back drukken en met foutieve login inloggen
Verwachte uitvoer:	Gebruiker blijft op de inlogpagina
Gekregen uitvoer:	Gebruiker blijft op de inlogpagina
Opmerking:	

Onderdeel:	Inloggen
Invoer:	Inloggen met correcte login, back drukken, handmatig de url terugzetten zoals deze was na inloggen
Verwachte uitvoer:	Klantenpagina
Gekregen uitvoer:	Klantenpagina
Opmerking:	

Onderdeel:	Uitloggen
Invoer:	Op link uitloggen klikken
Verwachte uitvoer:	Inlogpagina
Gekregen uitvoer:	Inlogpagina
Opmerking:	

Onderdeel:	Uitloggen
Invoer:	Op link uitloggen klikken, back drukken

Verwachte uitvoer:	Inlogpagina
Gekregen uitvoer:	Error: Info: Undefined index: gebruikerID Line: 126 File: D:\WWW\KCC Klantenmap\prg_lib\libLocks.php PHP :5.0.0RC1
Opmerking:	Het verwijderen van de locks bij uitloggen gaat fout. Bij een niet ingelogde gebruiker geeft dit fouten.
Oplossing	Controleren of sessie bestaat en indien dit is de locks verwijderen, anders niets doen.
Nieuwe testresultaten	Inlogpagina. Probleem opgelost

Onderdeel:	Klanten tonen
Invoer:	Correcte inlog
Verwachte uitvoer:	Klanten pagina
Gekregen uitvoer:	Klanten pagina
Opmerking:	

Onderdeel:	Klant tonen
Invoer:	Klik een klant aan
Verwachte uitvoer:	Algemene klant gegevens
Gekregen uitvoer:	Algemene klant gegevens
Opmerking:	

Onderdeel:	Klant wijzigen
Invoer:	Klik op klant wijzigen, voer een nieuwe naam in, klik op opslaan.
Verwachte uitvoer:	Klantenpagina
Gekregen uitvoer:	Klantenpagina
Opmerking:	

Onderdeel:	Detail tonen
Invoer:	Klik op een link van een detail (algemeen)
Verwachte uitvoer:	Detail pagina met de gekozen details
Gekregen uitvoer:	Detail pagina met de gekozen details
Opmerking:	

Onderdeel:	Klant toevoegen
Invoer:	Klant toevoegen link in klanten tonen scherm. Als naam 'marcel' invoeren.
Verwachte uitvoer:	Detail toevoegen voor algemene info
Gekregen uitvoer:	Detail toevoegen voor algemene info
Opmerking:	Omschrijving van identifier kan stuk netter en beter. (opgelost. 18 mei 2004)

Onderdeel:	Detail toevoegen (bij toevoegen klant)
Invoer:	Detail toevoegen na klant toevoegen. Correcte invoer meegeven.
Verwachte uitvoer:	Detail opgeslagen en getoond op het scherm
Gekregen uitvoer:	Detail op scherm.
Opmerking:	

Onderdeel:	Detail toevoegen (bij bestaande klant, eerste detail)
Invoer:	Detail toevoegen klikken, detail kiezen(invoer controle), gegevens invoeren, opslaan
Verwachte uitvoer:	Detail pagina.
Gekregen uitvoer:	Detail pagina.
Opmerking:	

Onderdeel:	Detail toevoegen (bij bestaande klant, detail bijvoegen)
Invoer:	Detail toevoegen klikken, detail kiezen(invoer controle), gegevens invoeren, opslaan. Gegevens bevat fouten
Verwachte uitvoer:	Detail pagina.
Gekregen uitvoer:	Foutmeldingen over invoer, na correctie wel detail pagina
Opmerking:	Indien invoer niet verplicht is en de invoer is leeg krijg ik alsnog foutmeldingen. Kan komen door het vullen van de invoer met een spatie voor de invoer controle.

Onderdeel:	Details wijzigen
Invoer:	Algemene gegevens wijzigen van de zojuist ingevoerde klant. Correcte invoer.
Verwachte uitvoer:	Detail pagina met gewijzigde gegevens
Gekregen uitvoer:	Detail pagina met de gewijzigde gegevens
Opmerking:	

Onderdeel:	Details wijzigen
Invoer:	Algemene gegevens wijzigen van de zojuist ingevoerde klant. Foutieve invoer.
Verwachte uitvoer:	Error bericht op het scherm
Gekregen uitvoer:	Error bericht op het scherm
Opmerking:	

Onderdeel:	Klant toevoegen.
Invoer:	Klant toevoegen, controleren of het annuleren knopje er staat.
Verwachte uitvoer:	Knop niet aanwezig
Gekregen uitvoer:	Knop niet aanwezig
Opmerking:	

Onderdeel:	Detail wijzigen
Invoer:	Detail wijzigen, controleren of de annuleren knop er staat
Verwachte uitvoer:	Detail pagina met knop.
Gekregen uitvoer:	Annuleren knop aanwezig
Opmerking:	

Onderdeel:	Detail wijzigen,
Invoer:	Detail wijzigen terwijl een andere gebruiker dat al aan het doen is.
Verwachte uitvoer:	Melding, bestand is in gebruik door gebruiker ...
Gekregen uitvoer:	ERROR [8] Info: Undefined variable: arrVraagAntwoord Line: 48 File: D:\WWW\KCC Klantenmap\prg\intern\dsp_detail.php PHP :5.0.0RC1
Opmerking:	ArrVraagAntwoord niet bekend binnen functie ivm aanwezig zijn van een lock.
Mogelijke oplossing	ArrVraagAntwoord aan het begin definiëren als array zodat hij altijd bestaat (indien lock is hij leeg).
Testresultaten	Zie volgende tabel.

Onderdeel:	Detail wijzigen,
Invoer:	Detail wijzigen terwijl een andere gebruiker dat al aan het doen is.
Verwachte uitvoer:	Melding, bestand is in gebruik door gebruiker ...
Gekregen uitvoer:	ERROR [8] Info: Undefined variable: fuseOpslaan Line: 52 File: D:\WWW\KCC Klantenmap\prg\intern\dsp_detail.php PHP :5.0.0RC1
Opmerking:	fuseOpslaan niet bekend binnen het systeem
Mogelijke oplossing	fuseOpslaan definiëren. Tevens fuseAnnuleren en annuleren definiëren.
Testresultaten	Test geslaagd. Melding op scherm. Gebruiker klopt niet. Is de huidige gebruiker en niet de lock gebruiker. Zie volgende tabel

Onderdeel:	Detail wijzigen,
Invoer:	Detail wijzigen terwijl een andere gebruiker dat al aan het doen is.
Verwachte uitvoer:	Bestand 'algemeen.XML' is in gebruik door 'Test gebruiker'
Gekregen uitvoer:	Bestand 'algemeen.XML' is in gebruik door 'Marcel Stokman'
Opmerking:	Foutieve gebruiker tonen. Tevens aangeven dat Detail algemeen in gebruik is ivp bestand.
Mogelijke oplossing	Juiste gebruiker ophalen uit db. Foutmelding wijzigen.
Testresultaten	Melding gewijzigd. Juiste gebruiker wordt getoond dmv functie welke de juiste lock eigenaar ophaalt.