

Ontwikkeling van BackupTools

Procesbeschrijving van het afstudeertraject

Auteur: P.T. Debets
Datum: 12 april 2005
Plaats: Den Haag



Afstudeerperiode: 15 november 2004 t/m 24 maart 2005, 2004-2.2
Opleidingsvariant: OSTI
Stagiair: dhr P.T. Debets, 99007446
Contactdocenten: dhr F. Sikkema
dhr J.L.A. Schramp

Stagebedrijf: Invers B.V.
Het Kleine Loo 414 G
2592 CK Den Haag
Afdeling: Systeemontwikkeling
Bedrijfsmentor: dhr. M. van Romswinkel

Referaat

Dit rapport beschrijft de werkzaamheden die zijn uitgevoerd voor de ontwikkeling van een Back-up applicatie. Na het uitvoeren van een concurrentie onderzoek is een remote back-up tool ontwikkeld. Met deze tool kunnen gebruikers hun data via internet back-uppen naar een server.

De werkzaamheden zijn uitgevoerd door P.T. Debets tijdens zijn stageperiode van 15 november 2004 tot en met 24 maart 2005 bij Invers B.V. in Den Haag

Descriptoren:

- DSDM
- Delphi 5.0
- Advantage
- Back-up
- Encryptie
- Haagse Hogeschool
- Invers B.V.
- Afstuderen

Voorwoord

Voor u ligt het stageverslag over de afstudeerperiode die ik bij Invers B.V. te Den Haag heb doorlopen. Dit verslag geeft een overzicht van de werkzaamheden die tijdens het stagetraject zijn uitgevoerd.

Voor informatie over Invers B.V wordt u verwezen naar Hoofdstuk 2. Hoofdstuk 3 beschrijft mijn voorbereidingen voor de aanvang van de afstudeerperiode. De totstandkoming van de diverse rapporten wordt besproken in hoofdstuk 4 tot en met 7. Het verwerken tot de rapporten tot programmatuur wordt beschreven in hoofdstuk 8. Ik sluit af met een evaluatie van de afstudeerperiode. De gemaakte rapporten zijn als bijlagen bij dit document terug te vinden.

Voor het uitvoeren van de afstudeerperiode wil ik iedereen binnen Invers B.V. bedanken voor hun medewerking. In het bijzonder wil ik de volgende mensen bedanken. Bart, Urban en Marlon voor de interviews en de tijd die ze hebben genomen voor het leveren van feedback. Marco, Alexander, Colin en André voor de begeleiding bij en uitleg van Delphi. De Haagse Hogeschool voor het bijbrengen van kennis over informatietechnologie en informatievoorziening. Als laatste wil ik Willem bedanken voor de controle van de inhoud van dit rapport.

Thijs Debets
Den Haag, 24 maart 2005

Inhoudsopgave

1	INLEIDING	1
2	ORGANISATIE.....	2
2.1	HISTORIE EN OMVANG	2
2.2	ORGANISATIE STRUCTUUR.....	2
2.3	WERKZAAMHEDEN	3
2.3.1	<i>Nieuwe producten.....</i>	<i>3</i>
2.3.2	<i>Ontwerp en ontwikkeling.....</i>	<i>4</i>
2.3.3	<i>Distributie</i>	<i>5</i>
2.3.4	<i>Helpdesk.....</i>	<i>5</i>
2.3.5	<i>Toekomst</i>	<i>5</i>
3	VOORBEREIDING	6
3.1	AANLEIDING	6
3.2	KENNISMAKING	6
3.3	ONTWIKKELMETHODE	6
3.4	OPDRACHTOMSCHRIJVING	7
3.4.1	<i>Doelstelling</i>	<i>7</i>
3.4.2	<i>Nadrukken</i>	<i>7</i>
3.4.3	<i>Uitgangssituatie</i>	<i>7</i>
3.4.4	<i>Concrete werkzaamheden</i>	<i>7</i>
3.4.5	<i>Op te leveren producten</i>	<i>8</i>
4	PLAN VAN AANPAK	9
4.1	EISEN EN BEPERKINGEN	9
4.2	KWALITEITSBORGING	10
4.2.1	<i>Productkwaliteit.....</i>	<i>10</i>
4.2.2	<i>Proceskwaliteit.....</i>	<i>10</i>
4.2.3	<i>Voorgestelde maatregelen.....</i>	<i>10</i>
4.3	PLANNING.....	12
4.4	AANPASSINGEN	12
5	BEDRIJFSANALYSE.....	14
5.1	INFORMATIE VERGARING	14
5.2	PAKKETBESCHRIJVING.....	15
5.2.1	<i>Back-up methoden</i>	<i>15</i>
5.2.2	<i>Beveiliging</i>	<i>17</i>
5.2.3	<i>Compressie.....</i>	<i>18</i>
5.2.4	<i>Opslagmoment</i>	<i>18</i>
5.2.5	<i>Restore</i>	<i>18</i>
5.2.6	<i>Overige informatie</i>	<i>18</i>
5.3	GEGEVENSSTROMEN.....	19
5.4	KLASSENDIAGRAM	20
5.5	SYSTEEMARCHITECTUUR	22
5.6	PROTOTYPING.....	23
6	FUNCTIONEEL MODEL.....	24
6.1	INTERACTIE	24
6.2	FUNCTIESTRUCTUUR.....	26
6.3	TOESTANDSDIAGRAM	28
6.4	SEQUENCE DIAGRAM	28
6.5	ACTIVITEITSDIAGRAM EN PROTOTYPING	29
7	TECHNISCH ONTWERP	30
7.1	PLANNING.....	30
7.2	MUST HAVES - SHOULD HAVES - COULD HAVES	31
7.3	TIMEBOXEN	33
7.4	TIMEBOX 1.....	35

7.5	TIMEBOX 2.....	37
7.6	TIMEBOX 3.....	38
8	PROGRAMMA ONTWIKKELING	39
8.1	VOORBEREIDING.....	39
8.2	TIMEBOX 1.....	39
8.3	TIMEBOX 2.....	42
8.3.1	<i>Opslag</i>	42
8.3.2	<i>Selectie</i>	42
8.4	TIMEBOX 3.....	47
9	EVALUATIE	51
9.1	PROCESEVALUATIE	51
9.2	PRODUCTEVALUATIE	52
9.3	SLOTWOORD	53
	LITERATUURLIJST	54
	BIJLAGE I: OPDRACHTOMSCHRIJVING.....	I
	BIJLAGE II: PLAN VAN AANPAK	II
	BIJLAGE III: BEDRIJFSANALYSE.....	III
	BIJLAGE IV: FUNCTIONEEL MODEL	IV
	BIJLAGE V: TECHNISCH ONTWERP	V

1 Inleiding

Als afronding van de studie Informatievoorziening en Informatietechnologie aan de Haagse Hogeschool moet de student een afstudeeropdracht uitvoeren. De afstudeeropdracht moet aansluiten bij de afstudeerrichting van de student. Bij Invers B.V. wordt het assortiment uitgebreid met een back-up applicatie.

Dit rapport geeft een beschrijving van de door de student uitgevoerde werkzaamheden in een periode van 18 weken bij Invers B.V. Hiermee wordt het ontwikkelproces beschreven en worden gemaakte keuzes toegelicht.

Hoofdstuk 2 beschrijft de organisatie waar de student de opdracht heeft uitgevoerd. In hoofdstuk 3 worden getroffen voorbereidingen uiteengezet. De totstandkoming van het Plan van Aanpak wordt besproken in hoofdstuk 4. De werkzaamheden die hebben geleid tot de Bedrijfsanalyse worden beschreven in hoofdstuk 5. Vervolgens komt in hoofdstuk 6 de samenstelling van het Functioneel model aan bod. Hoofdstuk 7 beschrijft de opbouw van het Technisch ontwerp. De omzetting van documentatie naar programmacode wordt besproken in hoofdstuk 8. Er wordt afgesloten met een evaluatie van het afstudeertraject in hoofdstuk 9.

2 Organisatie

Invers B.V is een software ontwikkelbedrijf. Het bedrijf is ontstaan uit het idee om software te maken voor thuisgebruikers. De belangrijkste eis aan de software was dat de softwarepakketten gebruiksvriendelijk moesten zijn. Hier werd uiting aan gegeven door de software Nederlandstalig, makkelijk te installeren en makkelijk in gebruik te maken. Daarnaast moest de software betaalbaar zijn. Momenteel worden er onder diverse labels producten op de markt gezet.

2.1 Historie en omvang

Het idee voor het ontwikkelen van gebruiksvriendelijke software is in 1993 ontstaan. Invers B.V werd in 1994 opgericht, waarna het businessplan, met daarin onder andere de technische specificaties voor de software, de kosten, verkoopplaatsen en de prijs hiervan, werd opgesteld. In 1995 ontving Invers B.V. steun van een bank om het bedrijf op te starten. In ongeveer drie jaar tijd groeide de tweemanszaak uit tot een bedrijf met zo'n 10 medewerkers.

In 1999 werd er een apart bedrijf opgericht voor activiteiten op het gebied van Internet en multimedia. That's-Id verzorgde de grafische vormgeving van de eerste pakketten van Invers B.V., maar kon zich ook aanbieden voor externe klanten. Naast That's-Id is ook Quintens Media uit Invers B.V. ontstaan. Naast de verkoop van Invers producten werden ook producten van derden door dit nieuwe bedrijf verkocht. Door de opsplitsing in drie aparte bedrijven, konden de verschillende takken efficiënter met de beschikbare capaciteit omgaan.

De drie bedrijven zijn na een verhuizing in hetzelfde pand gevestigd en tellen in totaal zo'n 30 medewerkers, waarvan Invers B.V. er zestien telt.

2.2 Organisatie structuur

Door de beperkte omvang van Invers B.V. kan er een open organisatiestructuur bestaan. Informele contacten spelen hierin een grote rol.

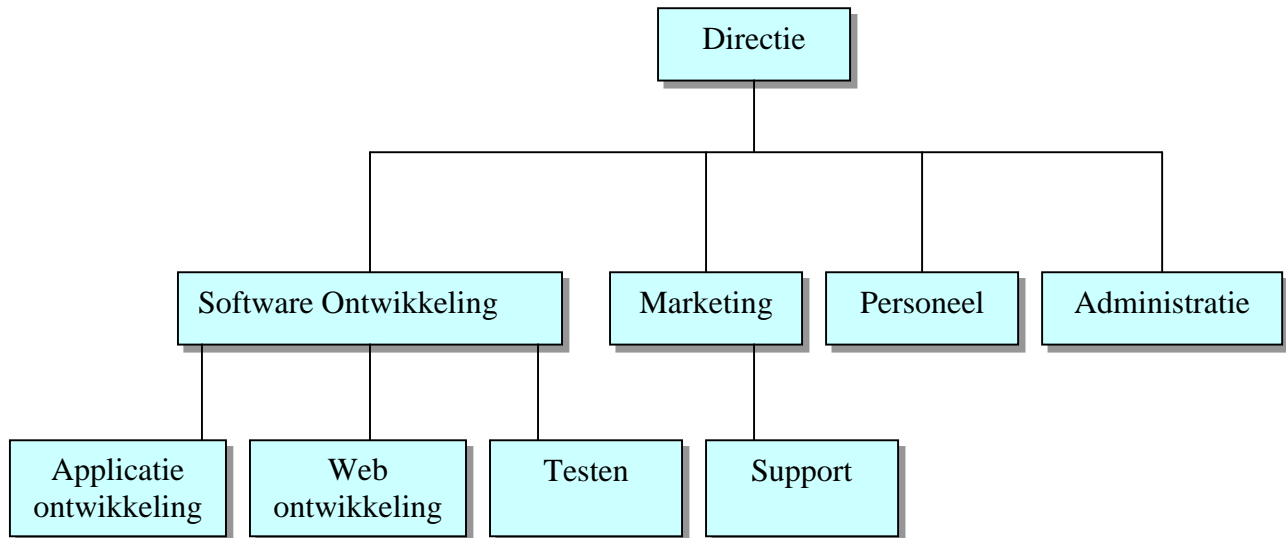
De directie wordt gevormd door de oprichters van het bedrijf. Onder het management zijn diverse afdelingen te vinden.

Het primaire proces van het bedrijf is software ontwikkeling, waardoor deze afdeling de grootste is. De afdeling is op te delen in applicatieontwikkeling, webontwikkeling en testen. Deze afdeling wordt aangestuurd door de SO Manager.

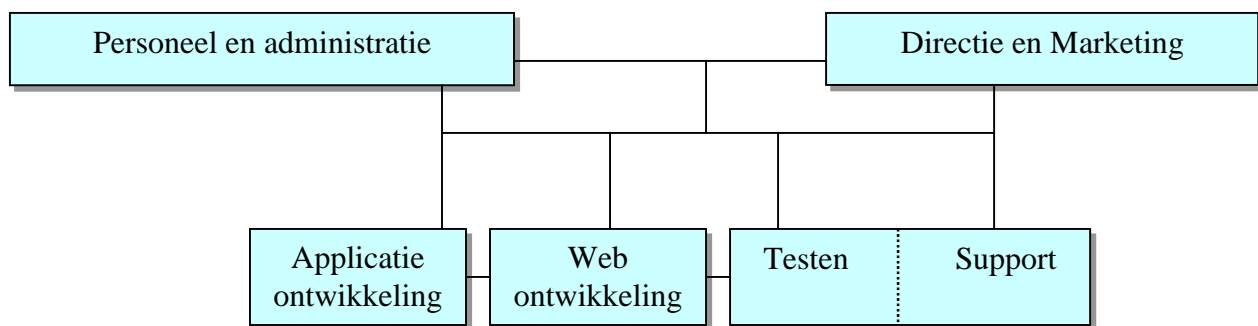
Naast deze afdeling zijn er de afdeling marketing, personeelszaken en administratie. Omdat de afdeling marketing momenteel geen personeelsbezetting kent, worden de taken uitgevoerd door één van de directeuren. Het bieden van support aan gebruikers van de eindproducten valt onder de taken van de afdeling marketing.

De afdelingen personeelszaken en administratie worden momenteel ingevuld door één persoon. Verder is er één verantwoordelijke voor het testen van de diverse software pakketten. Deze voert tevens diverse helpdesk taken uit en wordt geholpen door diverse helpdeskmedewerkers.

Organogram I geeft een beeld van de officiële organisatiestructuur. Organogram II geeft inzicht in de werkelijke communicatielijnen:



Organogram I: Organisatiestructuur



Organogram II: Communicatielijnen

2.3 Werkzaamheden

Invers ontwikkelt software pakketten voor de thuisgebruiker. Deze software pakketten zijn eenvoudig in het gebruik en daardoor geschikt voor een breed publiek. Om tot een succesvol eindproduct te komen worden diverse stappen doorlopen. Kort na de start van het bedrijf werd de ontwikkeling van de producten nog uitbesteed, maar tegenwoordig worden de pakketten intern ontwikkeld.

2.3.1 Nieuwe producten

De eerste stap bij het uitbrengen van een nieuw product is het leggen van een basis voor verdere ontwikkeling. Ideeën voor nieuwe producten komen veelal bij de directie vandaan. Daarnaast zijn er nog diverse andere manieren om aan ideeën voor een nieuw product te komen:

- Onderzoek

Door onderzoek te doen naar ontwikkelingen in de Nederlandse en buitenlandse markt, wordt gezocht naar producten die voor Invers interessant zijn om te ontwikkelen. Hierbij wordt gekeken naar producten van concurrenten, maar ook naar diverse nieuwsgroepen en algemene trends.

- Samenwerking

Vanwege de aanwezige kennis bij Invers kunnen producten redelijk snel ontwikkeld worden. Invers biedt zich daarom aan om applicaties te ontwikkelen welke door andere bedrijven uitgebracht worden. Ook wordt er gezocht naar nieuwe partners voor toekomstige projecten.

- Consumenten

Door consumenten de mogelijkheid te geven suggesties te doen voor nieuwe of al ontwikkelde programma's kan het pakket beter aan de wensen van de consument worden aangepast. Hierdoor wordt het draagvlak bij de consument verhoogd.

Een combinatie van bovenstaande methodes wordt tevens toegepast.

Na het opstellen van de eerste ideeën over een pakket, wordt onderzocht of er voldoende vraag vanuit de markt is. Als de vraag groot genoeg blijkt, worden de ideeën verder uitgewerkt.

2.3.2 Ontwerp en ontwikkeling

Bij het uitwerken van de applicatie wordt op een aantal punten gelet:

- Wijze van aanbieden aan de consument

Momenteel worden er door Invers twee manieren gebruikt om een applicatie aan de consument aan te bieden.

Producten van het Easy-disc label worden momenteel via retailers verkocht. Via deze wijze van verkoop, koopt de consument een doos met daarin een Cd-rom en handleiding in de winkel.

De laatste serie producten, waaronder BankingTools en HenzoYourPhotos, worden via Internet aangeboden. Deze on-line producten kunnen gratis geprobeerd worden. Afhankelijk van de functies van het pakket, moet het na de proefperiode geregistreerd worden om verder te kunnen gebruiken.

- De functionele werking

Door middel van sessies wordt de gewenste functionaliteit vastgelegd. Indien nodig worden externe instellingen of personen benaderd om bij te dragen aan de functionele eisen van het pakket.

- Grafical User Interface (GUI)

De GUI ontwikkeling wordt deels uitbesteed aan That's-Id. Vanuit de functionele eisen wordt een prototype ontwikkeld. Dit prototype wordt aan That's-Id geleverd om een passende GUI te ontwerpen. Door de nauwe samenwerking kan That's-Id een passende GUI ontwikkelen. Door een evaluatie van het prototype en de nieuwe GUI wordt de uiteindelijke stijl opgesteld.

- Productie software

Door de interne ontwikkeling van de applicaties heeft Invers volledige controle over de wijze van ontwikkelen. Hierdoor kan Invers snel op wijzigingen inspringen en veranderingen in de software aanbrengen. Door te kiezen voor flexibele ontwikkel- en productiesoftware kan een snelle reactietijd op veranderingen gerealiseerd worden. Er wordt niet alleen een keuze gemaakt voor de software voor de ontwikkeling van het pakket, maar ook voor de software voor ondersteunende diensten, zoals een website en een registratiesysteem.

- Fysieke productie

Voor fysieke producten moeten producenten van de verpakking en cd-roms worden aangetrokken. Omdat de verkoop van fysieke producten door Quintens Media wordt uitgevoerd, worden keuzes hierover in samenwerking met Quintens Media gemaakt.

2.3.3 Distributie

De distributie van fysieke producten wordt afgehandeld door Quintens Media. Door de afsplitsing van deze taak kan dit nieuw opgerichte bedrijf efficiënter werken.

De distributie van on-line producten wordt door Invers uitgevoerd. De pakketten kunnen vanaf de Internet server van Invers worden gedownload. Updates voor de fysieke en on-line producten worden ook via deze server aangeboden.

2.3.4 Helpdesk

Afhankelijk van het ontwikkelde pakket biedt of Invers, of de afnemer van het pakket ondersteuning aan de eindgebruikers. Invers heeft voor de support van de Easy-Disc serie één vast personeelslid. Dezelfde persoon is verantwoordelijk voor het testen van de diverse pakketten.

Voor de ondersteuning van BankingTools heeft Invers diverse studenten in dienst. Vanwege grote belangstelling voor deze serie worden deze studenten momenteel bijgestaan door enkele medewerkers. De studenten kunnen op kantoor of vanuit huis hun werkzaamheden uitvoeren. Via mail en instant messaging programma's wordt gecommuniceerd over eventuele problemen. Om op de hoogte te blijven van de laatste ontwikkelingen wordt de studenten aangeraden minimaal één dag in de week op kantoor door te brengen.

De ondersteuning voor het fotopakket van HenzoYourPhotos wordt door Henzo aan de gebruikers aangeboden. Indien er zich onoplosbare problemen voordoen, kunnen medewerkers van Henzo contact opnemen met Invers.

2.3.5 Toekomst

Door de ontwikkelingen in de markt te volgen kan Invers goed inspelen op veranderingen. Een verschuiving van verkoop via winkels naar verkoop via Internet is duidelijk waarneembaar. Om klaar te zijn voor deze trend, worden nieuwe producten alleen nog maar on-line aangeboden. De verwachting is dat het Easy-Disc label ook aangepast zal worden, om on-line verkoop mogelijk te maken.

Met de Easy-Disc serie werden alleen inkomsten gerealiseerd met de verkoop van het fysieke product. Met BankingTools is ook het servicecontract gerealiseerd. Omdat de BankingTools applicatie afhankelijk is van regelmatig veranderende importbestanden van banken, moet het pakket hier constant aan aangepast worden. Om de kosten voor deze aanpassingen terug te verdienen, komen alleen mensen met een servicecontract in aanmerking voor deze aanpassingen. Het servicecontract voorziet Invers daarnaast van een stabiele bron van inkomsten.

Met HenzoYourPhotos is Invers nog een stap verder gegaan. De ontwikkeling van het product is volledig op kosten van Invers gebeurd. De verkoop van fotoalbums, welke met HenzoYourPhotos worden gemaakt, zal de investering moeten terugverdienen.

Invers wil de inkomsten in de toekomst steeds meer verplaatsen van eenmalige inkomsten uit de verkoop van een pakket naar inkomsten uit een abonnement. Dit wordt gerealiseerd door een servicecontract aan te bieden, of een pakket te gaan verhuren. Zolang de gebruiker het pakket huurt, kan het gebruikt worden. Zodra de huurovereenkomst wordt verbroken, is het niet meer mogelijk om de applicatie te gebruiken.

3 Voorbereiding

Voor aanvang van de afstudeerperiode heb ik kennis gemaakt met het bedrijf en een opdrachtschrijving opgesteld.

3.1 Aanleiding

Invers is constant op zoek naar nieuwe markten. De directeuren hebben diverse ideeën. Helaas is er niet altijd genoeg tijd en capaciteit om volledig nieuwe projecten op te zetten. Een belangrijk onderdeel van de nieuwe ideeën is het genereren van een constante bron van inkomsten.

Tijdens mijn helpdeskwerk voor BankingTools kwam het voorstel om mij, voor mijn afstuderen, één van deze ideeën uit te laten werken tot een applicatie. Vanwege de beginnende samenwerking met Henzo voor HenzoYourPhotos was er geen tijd om nieuwe projecten op te starten. Er waren diverse ideeën, waarvan Invers het meest geïnteresseerd was in een back-up applicatie en een herziening van het huidige registratiesysteem van BankingTools.

De back-up applicatie sprak mij het meest aan. Bij het ontwikkeltraject dat bij de back-up applicatie hoorde, zou ik de stof van het laatste leerjaar van mijn studie goed kunnen toepassen, en zouden veel aspecten van ICT aan bod komen.

3.2 Kennismaking

Vanwege mijn werk als support medewerker bij Invers konden de diverse ideeën gemakkelijk besproken worden. De open bedrijfsstructuur droeg ook bij aan het opstellen van de opdrachtschrijving. Tijdens het opstellen van de opdrachtschrijving heb ik overlegd met diverse mensen binnen Invers. Voor de ideeën over het pakket heb ik gesproken met de directeuren, bij wie het idee oorspronkelijk vandaan kwam. Deze zouden user requirements opleveren bij aanvang van de afstudeerperiode. Daarnaast heb ik gesproken met de SO-Manager over de werkwijze binnen Invers tijdens het ontwikkelen van een nieuw pakket.

3.3 Ontwikkelmethode

Invers gebruikt een eigen ontwikkelmethode, welke is gebaseerd op SDM (System Development Methodology). De bruikbare onderdelen van deze ontwikkelmethode zijn geselecteerd en aangevuld om een snelle en flexibele ontwikkeling van applicaties mogelijk te maken. De ideeën voor de nieuwe applicatie worden uiteengezet en geanalyseerd. Het resultaat hiervan wordt verwerkt in functionele specificaties. Vanuit deze functionele specificaties worden de diverse programmeurs ingedeeld om verschillende onderdelen van het pakket te ontwikkelen.

Vanwege de gebruikte technieken binnen Invers was mijn eerste idee om SDM te gebruiken, met behulp van de technieken van Yourdon. Wanneer deze methode strikt gevolgd wordt, geeft dit weinig mogelijkheid tot interactie met de gebruikers vanwege het waterval model waarmee het systeem ontwikkeld wordt. Daarnaast sluiten de technieken van Yourdon niet goed aan bij de ontwikkeltaal Delphi. Tijdens mijn stage heb ik enkele technieken van Yourdon als hinderlijk ervaren, vanwege de object georiënteerde opbouw van Delphi.

De flexibiliteit van de Invers Methode komt gedeeltelijk overeen met DSDM (Dynamic Systems Development Method), welke dynamisch van opzet is.

Er vindt eerst een onderzoek plaats naar de haalbaarheid en de afzetmarkt van de nieuwe applicatie. Vervolgens worden de functionele eisen vastgelegd. Met deze

functionele eisen wordt de applicatie ontwikkeld en technisch gedocumenteerd. Als laatste wordt het pakket verkocht en ondersteund.

Omdat ik geen ervaring had met DSDM ben ik op zoek gegaan naar boeken hierover. DSDM is ontwikkeld door het DSDM Consortium. Tevens worden cursussen over DSDM door het Consortium georganiseerd. Waarschijnlijk zorgt dit ervoor dat er weinig officiële DSDM leerboeken te krijgen zijn. Het boek 'A Framework for Business-Centered Development' was bij diverse bedrijven niet op voorraad en kon ook niet worden besteld. Als reden werd gegeven dat de drukker bezig was met een herdruk en geen oude boeken uitleverde. Het boek 'DSDM: de methode in de praktijk', van dezelfde schrijfster beschrijft de werking van DSDM, maar is geen gebruiksaanwijzing voor de methode. Ik kon hier wel voldoende informatie uit halen om een goede opzet voor mijn afstuderen op te stellen, en een goed beeld van de methode te krijgen.

'DSDM: de methode in de praktijk' is een Nederlandse vertaling van het Engelstalige 'DSDM: The method in Practice'. Door de vertaalslag komen sommige termen verwarrend over. De inhoud van het boek gaf wel voldoende duidelijkheid over de werking van DSDM.

3.4 Opdrachtomschrijving

De opdrachtomschrijving is tot stand gekomen in overleg met de directeurs en de SO-Manager. Invers wil haar productlijn uitbreiden met een remote back-up tool.

3.4.1 Doelstelling

Ontwikkel en lever een back-up tool op, inclusief documentatie, welke geschikt is voor verkoop aan consumenten.

De nieuwe tool moet de consument de mogelijkheid bieden back-ups te maken van de lokale pc, waarop de tool geïnstalleerd is, naar diverse locaties, waaronder een remote locatie, welke door Invers of een partner wordt beheerd. Daarnaast moet de consument zijn gegevens terug kunnen halen om de lokale gegevens te herstellen. Het systeem moet gebruiksvriendelijk zijn en de gebruiker begeleiden in het gebruik.

3.4.2 Nadrukken

De opgeleverde tool moet gebruiksvriendelijk zijn.

De hoeveelheid gegenereerde datatraffic moet geminimaliseerd worden.

Er moet een optie aanwezig zijn voor het aanbrengen van historische informatie en recovery.

Eisen aan en uitvoering van de beveiliging van het dataverkeer, de opgeslagen data en de betrouwbaarheid van de opgeslagen gegevens moeten duidelijk beschreven worden.

3.4.3 Uitgangssituatie

Het pakket moet volledig nieuw ontwikkeld worden.

3.4.4 Concrete werkzaamheden

Plan van aanpak

Bedrijfsanalyse

- Gebruikersgroepen
- Globaal gegevensstroom diagram
- Klassediagram

- System Architecture Definition
- Global Prototyping Plan

Functioneel model

- Beschrijving van interactie met het systeem
- Use case diagrammen
- Toestandsdiagrammen
- Sequence diagrammen
- Activiteitsdiagrammen
- Prototypen tbv de user interface

Technisch ontwerp en realisatie

- Beeldscherm en lijstindeling
- Opslagstructuur
- Code beschrijving
- Testset
- Prototype tbv van Server en Client
- Handleiding

Intern en Extern testen

- Test server
- Test clients

3.4.5 Op te leveren producten

De volgende producten dienen te worden opgeleverd:

- Plan van Aanpak

Het plan van aanpak legt de basis voor de aanpak van het ontwikkeltraject. In dit plan worden diverse voorwaarden beschreven, zoals randvoorwaarden en planning.

- Bedrijfsanalyse

In de bedrijfsanalyse wordt de ontwikkel- en gebruiksomgeving beschreven. Aan de hand van diverse schema's worden de gebruikersgroepen, informatiestromen en globale ontwikkelparameters beschreven

- Functioneel model

In het functioneel model wordt de functie- en gegevensstructuur van de tool beschreven. De diverse schema's worden begeleid door een prototype voor het testen van de User Interface.

- Technisch ontwerp

Tijdens de diverse fasen van het technisch ontwerp wordt de interne werking van de tool beschreven. Aan deze interne werkingen wordt een testset toegekend.

- Prototype

Elke timebox uit het functioneel model en het technisch ontwerp levert een prototype op. Deze prototypen worden gekeurd aan de hand van de opgestelde testsets.

Daarnaast worden de prototypen gebruikt om de wisselwerking van de tool met de gebruikers te beoordelen.

- Eindproduct

Het op te leveren eindproduct, inclusief beknopte gebruiksaanwijzing, wordt gebruikt om de eindkeuring te doorlopen. Aan het eindproduct kunnen nog technische wijzigingen worden uitgevoerd, vanwege eventuele fouten die tijdens de definitieve testronden ontdekt worden. Na aanpassing zal het eindproduct gereed zijn voor verkoop.

4 Plan van aanpak

In de eerste week van mijn afstuderen heb ik het Plan van Aanpak (PvA) opgesteld. In het PvA worden de doelstelling, randvoorwaarden en planning besproken. Het PvA zorgt voor overeenstemming over de aangegeven punten, en geven een richtlijn voor de uit te voeren werkzaamheden. Het Plan van Aanpak heb ik opgesteld volgens het voorbeeld van ZBC Consultants. Door dit voorbeeld te volgen, worden alle benodigde onderdelen van een PvA behandeld. Dit voorbeeld heb ik al vaker gebruikt. Conflicten tijdens een project konden vaak opgelost worden door de volledigheid van het opgestelde PvA.

Voor de inhoud van het PvA ben ik uitgegaan van de opdrachtomschrijving. Diverse onderdelen, zoals de doelstelling en de op te leveren producten zijn daar al in aanwezig. Daarnaast zijn er diverse onderdelen toegevoegd, zoals deze in het voorbeeld van ZBC worden voorgesteld.

4.1 Eisen en beperkingen

Na overleg met medewerkers van Invers en onderzoek over het maken van Back-ups heb ik de eisen en beperkingen van de opdracht opgesteld. De onderdelen systeemeisen, taal en portabiliteit zijn directe keuzes van de directie van Invers. De overige onderdelen zijn in overleg en door onderzoek opgesteld.

De tool moet op de pc van de eindgebruiker geïnstalleerd worden. Aan de hand van diverse instellingen kunnen er back-ups gemaakt worden van bestanden en/of mappen. De keuze van deze bestanden en mappen wordt door de gebruiker ingesteld. De interface van het pakket moet gebruiksvriendelijkheid uitstralen. Door gebruik van wizards moet de gebruiker begeleid worden in het gebruik van het pakket.

Systeemeisen

De tool moet geschikt zijn voor Windows 2000 en XP. Lagere versies van Windows behoren tot de mogelijkheid, mogelijk met beperkte functionaliteit. Er worden geen andere besturingsystemen ondersteund. De aangesloten pc moet over een breedband verbinding met het Internet beschikken.

Betrouwbaarheid van de gegevens

Het pakket moet betrouwbaar zijn.

Een gemaakte back-up moet teruggelezen kunnen worden.

De teruggelezen gegevens moeten gelijk zijn aan de geback-upte gegevens, niet meer en niet minder.

Stabiliteit

Bij normaal gebruik van het pakket moet het verwachte gedrag worden uitgevoerd.

De tool moet geen onverwacht gedrag vertonen.

Omvang

Het pakket wordt eenmalig gedownload. Omdat een breedband verbinding een minimale systeemeis is, is de omvang van het pakket niet relevant.

De omvang van de data, welke verstuurd wordt tussen de client en de server, moet beperkt worden.

Taal

Het pakket wordt ontwikkeld in het Nederlands, met mogelijkheid tot omzetting naar andere talen.

Beveiliging

De gebruiker moet de opgeslagen gegevens kunnen beveiligen, zodat ongeoorloofd gebruik door derden niet mogelijk is. Deze functie is vooral van belang bij opslag op een remote locatie.

Portabiliteit

Diverse onderdelen van het pakket moeten los gebruikt kunnen worden in andere Invers pakketten. Het pakket moet voorzien kunnen worden van een ander uiterlijk, zodat het pakket onder het label van een partner kan worden uitgebracht.

Cruciale succesfactoren

De gegevens op de client pc moeten remote kunnen worden opgeslagen en terug te plaatsen zijn.

De software moet duidelijk zijn in het gebruik door de klant.

4.2 Kwaliteitsborging

De opgeleverde producten moeten aan diverse eisen voldoen. Hierbij wordt rekening gehouden met de huidige werkwijze binnen Invers. Wanneer er geen richtlijnen aanwezig zijn, wordt een nieuwe richtlijn opgezet.

4.2.1 Productkwaliteit

Rapporten binnen Invers veranderen vaak. Zodra de basis voor een pakket is vastgelegd in functionele specificaties, wordt er beperkt gedocumenteerd in rapporten. Voor het opstellen van programmacode zijn richtlijnen aanwezig, waarin gesteld wordt dat de code moet lezen als een goed boek. Eventueel kan de code worden toegelicht. Deze werkwijze zorgt ervoor dat het bijhouden van een aparte technische documentatie niet nodig is.

De rapporten voldoen aan de op de Haagse Hogeschool gestelde rapportage eisen. De rapporten moeten naast technische schema's tekstuele uitleg bevatten. De rapporten worden geschreven in Word.

De remote back-up tool moet met een beperkte uitleg te begrijpen zijn voor de gebruikers. Daarnaast wordt de tool voorzien van wizards en een gebruikershandleiding.

De code van de tool wordt in de programmatuur in het Engels toegelicht.

4.2.2 Proceskwaliteit

Net als DSDM schrijft Invers het gebruik van mijlpalen voor. Bij Invers hebben de mijlpalen meer betrekking op het bouwen van de applicatie. Om het DSDM traject voor de betrokken personen inzichtelijk te maken, wordt de oplevering van documenten apart opgenomen in het Plan van Aanpak.

De rapporten worden door de uitvoerder, aan het einde van de geplande opleveringstijd, via e-mail als bijlage aan de betrokken personen aangeleverd. De e-mail bevat een korte, begeleidende tekst.

Tijdens het gehele project is inzicht mogelijk in de tussenproducten.

4.2.3 Voorgestelde maatregelen

Om tijdens de ontwikkeling niet te lang met één onderdeel bezig te zijn, wordt er gebruik gemaakt van timeboxing. Deze werkwijze geeft de ontwikkelaar een periode waarin enkele cruciale onderdelen opgeleverd moeten worden. Daarnaast worden ook enkele minder belangrijke onderdelen opgeleverd. Mocht de oplevering van de cruciale onderdelen meer tijd kosten dan gepland, dan kunnen de minder belangrijke onderdelen worden geschrapt of verplaatst naar een andere timebox.

De timeboxen van het technisch ontwerp bevatten maximaal 2 must haves, 3 should haves en 3 could haves.

4.3 Planning

Door het opgestelde PvA is ook nu een duidelijke richtlijn aanwezig. Door gebruik te maken van een globale planning, kon de voortgang van het project door alle betrokkenen gevolgd worden:

Week	47	48	49	50	51	52		1	2	3	4	5	6	7	8	9	10	11	12	13
Mijlpaal																				
afstudeerweek	1	2	3	4	5	6		8	9	10	11	12	13	14	15	16	17	18	19	20
Plan van aanpak																				
Bedrijfs analyse																				
Functioneel model																				
Technisch ontwerp																				
Prototypen																				
Testset																				
Gebruikers handleiding																				
Backuptool																				
Bedrijfsbezoek																				
Definitieve opdrachtomschrijving																				
Voortgangverslag																				
Conceptverslag																				
Definitief verslag																				

4.4 Aanpassingen

Achteraf zijn er nog enkele aanpassingen gedaan aan het Plan van Aanpak. Vanwege het beperkte vooronderzoek was het daarnaast verstandig om een concurrentie onderzoek te doen. Om een goed beeld te krijgen van aanwezige pakketten in de markt, ben ik diverse soorten pakketten gaan onderzoeken. Omdat dit onderzoek niet gepland was, maar wel noodzakelijk bleek te zijn, moest ik de planning aanpassen.

Omdat er nog geen duidelijke user requirements aanwezig zijn, moet er een concurrentie onderzoek gedaan worden. Door diverse pakketten te vergelijken, en de functionaliteit van deze pakketten te onderzoeken, wordt een beeld gecreëerd van de concurrentie. Na dit onderzoek zullen de wensen en eisen van het te ontwikkelen pakket worden opgesteld. Vervolgens worden deze wensen en eisen beoordeeld en geprioriteerd.

Het concurrentie onderzoek zal zo'n tien pakketten bevatten. Voor dit onderzoek wordt 40 uur gepland. Hiermee wordt de beschrijving van de interactie met het pakket uit het functionele model verplaatst naar de bedrijfsanalyse.

Door de aanpassing in de planning kwam deze er als volgt uit te zien:

Mijlpaal \ Week	47	48	49	50	51	52	53	1	2	3	4	5	6	7	8	9	10	11	12	13
afstudeerweek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Concurrentie onderzoek																				
Bedrijfs analyse																				

Tijdens het concurrentie onderzoek heb ik diverse programma's op de pc geïnstalleerd. Ook was de pc al voor diverse andere doeleinden gebruikt.

Om de pc waar het pakket op wordt ontwikkeld vrij te maken van sporen van de vorige gebruikers, wordt deze volledig opnieuw geïnstalleerd. Alle benodigde pakketten moeten opnieuw worden ingesteld.

Het volledig opnieuw installeren, vooral de installatie van Delphi en alle door Invers gebruikte componenten, leverde diverse problemen op. De verschillende componenten van externe leveranciers moeten in een vaste volgorde geïnstalleerd worden, en zelfs deze vaste volgorde geeft geen garantie voor een correcte werking van de componenten. Een correcte installatie is echter noodzakelijk om ontwikkelde programmatuur met collega's te kunnen delen. Deze herinstallatie kostte veel tijd, waardoor de planning nogmaals aangepast moest worden. De definitieve planning kwam er als volgt uit te zien:

Mijlpaal \ Week	47	48	49	50	51	52	vakantie	1	2	3	4	5	6	7	8	9	10	11	12	13
afstudeerweek	1	2	3	4	5	6		8	9	10	11	12	13	14	15	16	17	18	19	20
Plan van aanpak																				
Concurrentie onderzoek																				
Bedrijfs analyse																				
PC installatie																				
Functioneel model																				
Technisch ontwerp																				
Prototypen																				
Testset																				
Gebruikers handleiding																				
Backuptool																				
Bedrijfsbezoek																				
Definitieve opdrachtomschrijving																				
Voortgangsverslag																				
Conceptverslag																				
Definitief verslag								Donderdag 24 maart vóór 13:00												

5 Bedrijfsanalyse

In dit hoofdstuk wordt de totstandkoming van de bedrijfsanalyse beschreven. Hoewel de naam een volledige analyse van het bedrijf suggereert, gaat het hier om de Business Study. De Bedrijfsanalyse dient om inzicht te krijgen in de te automatiseren bedrijfsprocessen en de bijbehorende informatiebehoeften. Voor mijn afstudeeropdracht wordt niet zozeer een bedrijfsproces geautomatiseerd, maar wordt de eindgebruiker een nieuw proces aangeboden om niet aanwezige en/of ingewikkelde handelingen uit te voeren.

5.1 Informatie vergaring

Tijdens het opstellen van de opdrachtschrijving heeft Invers aangegeven dat de user requirements aanwezig zouden zijn bij aanvang van de afstudeeropdracht. Tijdens het overleg over de werking van het pakket, bleken er veel ideeën te zijn, maar het was nog niet duidelijk wat de eisen waren. Door over het pakket met de aanwezige personen te brainstormen, kwamen er veel ideeën op tafel. De belangrijkste punten waren:

- Naast remote ook locale backups
- Backup functionaliteit los te gebruiken
- Private label mogelijkheid
- Brede gebruikersgroep
- Release plan gewenst
- System specs: W2K en WinXP

Er zijn redelijk wat back-up programma's beschikbaar, van simpele, geautomatiseerde kopieerprogramma's tot volledige bedrijfsback-up pakketten naar remote servers. De diverse pakketten heb ik op verschillende manieren gevonden. Binnen Invers had al een oriëntatie plaatsgevonden, maar echt onderzoek was er nog niet gedaan. Hierdoor had ik een basis van vijf pakketten om te onderzoeken. Om een goed beeld te krijgen van verschillen en overeenkomsten tussen back-up pakketten, heb ik hier nog vijf pakketten bijgezocht. Deze heb ik via zoekmachines, advertenties en forums gevonden. De meeste pakketten waren gratis, of konden een periode gratis worden getest. Eén van de internetpagina's gaf alleen informatie over het back-up proces en de opslag van de gegevens. Omdat het aanbieden van de applicatie werd uitbesteed, kon ik geen pakket testen. Ook bij de aangesloten leveranciers waren geen testapplicaties aanwezig.

Van elk pakket heb ik geanalyseerd wat de mogelijkheden zijn. Door het pakket te installeren en in te stellen, konden alle mogelijkheden van de verschillende pakketten worden opgenomen in een vergelijkend schema. Dit overzicht geeft een duidelijk inzicht in de werking van de diverse pakketten, en kan als basis worden gebruikt voor de ontwikkeling van een nieuw pakket. In de bedrijfsanalyse ben ik uitgegaan van dit schema gecombineerd met de hoofdpunten uit de vergadering, om een zo breed mogelijke basis te leggen voor het te ontwikkelen pakket.

Naast het aanbieden van de pakketten, gaven de internetsites ook veel informatie over back-ups. Omdat technische achtergrond niet altijd goed uitgelegd werd, heb ik deze oriëntatie ook gebruikt voor het opdoen van kennis over back-ups, encryptie en compressie.

Deze informatie kwam goed van pas tijdens het beschrijven van mogelijke back-up methoden, opslag en beveiliging.

5.2 Pakketbeschrijving

Na het concurrentie onderzoek en het inlezen in de gerelateerde onderwerpen, heb ik een pakketbeschrijving opgesteld. Bij deze pakketbeschrijving ben ik van een ideaal pakket uitgegaan. Door uit te gaan van een ideaal programma worden alle mogelijkheden die het pakket kan bieden onderzocht. Zodra de uitgebreide beschrijving bekend is, kan er worden onderzocht welke functionaliteit bij de eerste release van de software gewenst is. Na de introductie van het pakket op de markt kan deze zelfde beschrijving gebruikt worden om een uitbreiding van het pakket te realiseren.

Vanwege de brede gebruikersgroep heb ik de pakketbeschrijving opgesplitst in twee delen. Voor de beginnende gebruiker moet het pakket makkelijk in gebruik zijn, voor de gevorderde gebruiker moet er meer invloed uit te oefenen zijn op de werking van het pakket.

In de beschrijving voor de eenvoudige back-up worden de drie basisstappen beschreven die nodig zijn voor het maken van een back-up. De drie stappen zijn:

- Selectie van bestanden en mappen
- Opslaglocatie
- Moment van uitvoeren

De lastige termen voor de beginnende gebruiker, zoals encryptie en compressie, worden verborgen en automatisch ingesteld. Ook het soort back-up, volledig, incrementeel of differentiëel, wordt voor de gebruiker verborgen. Door hier standaard opties te gebruiken zal de gebruiker minder afgeschrikt worden door het pakket.

Door deze opties wel aan de gevorderde gebruiker aan te bieden, kan het pakket op veel verschillende manieren worden toegepast.

5.2.1 Back-up methoden

Door de mogelijke back-up manieren uiteen te zetten, wordt deze stof ook duidelijk voor de medebetrokkenen

De gekozen data kan op diverse wijzen geback-upped worden:

Full back-up:

Bij een full back-up wordt alle aanwezige data gekopieerd naar de doellocatie. Eventuele verwijderde bestanden worden niet gekopieerd, maar blijven wel behouden op de doellocatie.

Incrementele back-up:

Bij een incrementele back-up worden alle bestanden waarin iets is gewijzigd sinds de laatste back-up, gekopieerd naar de doellocatie. Eventuele verwijderde bestanden worden niet gekopieerd, maar blijven wel behouden op de doellocatie.

Differentiële back-up:

Bij een differentiële back-up worden alle bestanden, waarin iets is gewijzigd sinds de laatste full of incrementele back-up, gekopieerd naar de doellocatie.

Mirror:

Bij een mirror back-up wordt de inhoud van twee locaties gelijk gemaakt. Gewijzigde bestanden worden op de doellocatie vernieuwd. Verwijderde bestanden worden van de doellocatie verwijderd.

Onderstaand schema geeft aan welke bestanden geback-upped worden.

Soort Nummer	Aanwezige bestanden	Full- back-up	Incrementeel	Differentieel
1	123	123	123	123
2	123456	123456	456	123456
3	12345678	12345678	78	12345678
Back-up resultaat		12345678	12345678	12345678
Verbruik		123 + 123456 + 12345678 = 17	123 + 456 + 789 = 9	123 + 123456 + 12345678 = 17

Bij een eerste back-up sessie is het aan te raden een full back-up te maken. Hierdoor kan alle informatie hersteld worden. Een volgende back-up sessie kan gemaakt worden door een incrementele back-up uit te voeren. Hierbij worden alle wijzigingen ten opzichte van de laatste back-up gekopieerd. Om deze back-up te herstellen moet eerst de full back-up hersteld worden, gevolgd door de incrementele back-up. De vorm levert een goede ruimtebesparing op, maar kan bij vele back-up sessies lang duren om terug te zetten. In plaats van een incrementele back-up, kan een differentiële back-up gemaakt worden. Om een differentiële back-up te herstellen, moet de laatste full back-up teruggezet worden, gevolgd door één differentiële back-up:

Full	Dif	Dif	Dif	Dif	Incr	Dif	Dif	Dif	Dif	Incr	Dif	Dif	Dif	Dif
←					←					←				
←	←				←	←				←	←			
←	←	←			←	←	←			←	←	←		
←	←	←	←		←	←	←	←		←	←	←	←	
←	←	←	←	←	←	←	←	←	←	←	←	←	←	←

Het herstel van een differentiële back-up is aanzienlijk sneller dan een incrementele back-up, doordat er geen wijziging na wijziging na wijziging hersteld hoeft te worden, maar alleen een wijziging op het oorspronkelijk geback-upte. Doordat er echter vanaf verder in het verleden wordt geback-upped, neemt deze vorm meer ruimte in op de doellocatie.

Wanneer deze back-up methode in de praktijk wordt toegepast, levert dit de volgende stappen op:

Soort Dag	Aanwezige bestanden	Full- back-up	Incrementeel	Differentieel
1	1	1		
2	12			2
3	123			23
4	1234			234
5	12345			2345
6	123456		123456	
7	1234567			7
8	12345678			78
9	123456789			789

Na het maken van de incrementele back-up, kunnen de voorliggende, differentiele back-ups overschreven worden. Wanneer een back-up van dag 5 teruggezet moet worden, moeten de Full Back-up van dag 1 en de differentieële back-up van dag 5 teruggezet worden. Voor een Restore van dag 9 worden achtereenvolgend de Full Back-up, de incrementele back-up en van dag negen de differentiele back-up teruggezet.

5.2.2 Beveiliging

Omdat de gebruikers persoonlijke gegevens opslaan, is het belangrijk dat er een goede encryptie gebruikt wordt. Er is uitgebreide informatie over dit onderwerp te vinden. Naast besprekingen over de werking van de encryptie methodes, wordt ook de mogelijkheid tot het kraken van de encryptie besproken. Onder ander Blowfish kwam hier goed uit. Er werd gesproken over het breken van de encryptie, maar dit bleek door een implementatiefout van de methode te gaan. Daarnaast is Blowfish gratis in gebruik en sneller dan gangbare encryptie methoden.

Speed Comparisons of Block Ciphers on a Pentium				
Algorithm	Clock cycles per round	# of rounds	# of clock cycles per byte encrypted	Notes
Blowfish	9	16	18	Free, unpatented
Khufu/Khafre	5	32	20	Patented by Xerox
RC5	12	16	23	Patented by RSA
DES	18	16	45	Data Security
IDEA	50	8	50	56-bit key
Triple-DES	18	48	108	patented by Ascom-Systec

Bron: <http://www.schneier.com/blowfish-speed.html>

De versleuteling van de gegevens is noodzakelijk indien derden toegang hebben tot de opgeslagen gegevens. Voor extra veiligheid wordt een door derden beheerde opslaglocatie extra beveiligd met een toegangscode.

5.2.3 Compressie

Een juiste keuze van de back-up methoden zorgt voor een optimale combinatie tussen ruimteverbruik en herstelsnelheid. Om het ruimteverbruik extra te minimaliseren, worden de bestanden gecomprimeerd opgeslagen:

Naast het gebruik van diverse vormen van back-ups kan er ruimte bespaard worden door de back-up te comprimeren. Dit zorgt voor een kleinere back-up, waarbij alle informatie behouden blijft. Het meest gangbare formaat om bestanden te comprimeren is het zip formaat. Deze wijze van compressie zorgt voor een toegang van de back-up bestanden zonder dat het back-up pakket nodig is.

5.2.4 Opslagmoment

Een back-up maken heeft tot doel om gedaan werk niet verloren te laten gaan. Om er voor te zorgen dat dit niet gebeurt, moet het werk met regelmaat worden veiliggesteld. Deze regelmaat kan het beste geautomatiseerd worden, zodat de gebruiker hier niet aan hoeft te denken. Om dit mogelijk te maken, moet er een tijdstip van back-uppen worden ingesteld. Voor pc's die het grootste gedeelte van de tijd aan staan, is dagelijks een vast tijdstip aan te raden. Voor pc's die regelmatig worden uitgezet, is het moment van afsluiten een zeer geschikt moment. Indien er gebruik gemaakt wordt van diverse back-up methoden, moeten er op verschillende dagen verschillende methoden gebruikt worden. Een planning per week of per maand is dan nodig.

5.2.5 Restore

Na een systeemcrash moeten de bestanden snel teruggezet kunnen worden. Door er van uit te gaan dat er geen informatie op de lokale pc meer aanwezig is over de geback-upte bestanden, worden situaties waarin de schade minder erg is ook afgeschermd.

Wanneer er geen informatie meer op de pc aanwezig is, wil de gebruiker toch de bestanden in de back-up terug kunnen plaatsen. Dit vereist een opslag van de back-up informatie in de back-up, en niet alleen in de gebruikte applicatie.

Ervan uitgaande dat de pc opnieuw is voorzien van een besturingssysteem, moet de gebruiker eerst het pakket opnieuw downloaden. De registratiegegevens zorgen ervoor dat de gebruiker herkend wordt als bestaande gebruiker. Vervolgens wordt de informatie over de back-up van de back-up locatie ingelezen. De gebruiker kan kiezen voor een snelle restore, zodat de meest actuele situatie van de gebruiker direct wordt hersteld.

Daarnaast moet ook de mogelijkheid bestaan om, net als bij het instellen van de back-up, meer invloed uit te kunnen oefenen op de wijze van restoren. Zo kunnen bestanden op alternatieve locaties worden teruggezet, of worden oudere versies van bestanden geselecteerd, omdat de huidige versie op de lokale pc niet juist is opgeslagen, of doordat hier onjuiste bewerkingen op hebben plaatsgevonden.

5.2.6 Overige informatie

Naast de beschrijving van het back-uppen en restoren heb ik nog aanvullende informatie toegevoegd aan de beschrijving van het pakket. Deze beschrijvingen hebben geen direct verband met de werking van het pakket, maar zorgen dat een gebruiker het pakket vertrouwt. Zo kan er een voortgangsfunctie worden toegevoegd om de gebruiker te laten controleren of het pakket actief is, of dat er mogelijk iets is misgegaan. Ook kan de gebruiker testen of de bestanden teruggezet kunnen worden vanaf de opslaglocatie. Dit geeft de gebruiker zekerheid over de correcte werking van het pakket. Daarnaast heb ik tevens de wijze van kostenberekening beschreven.

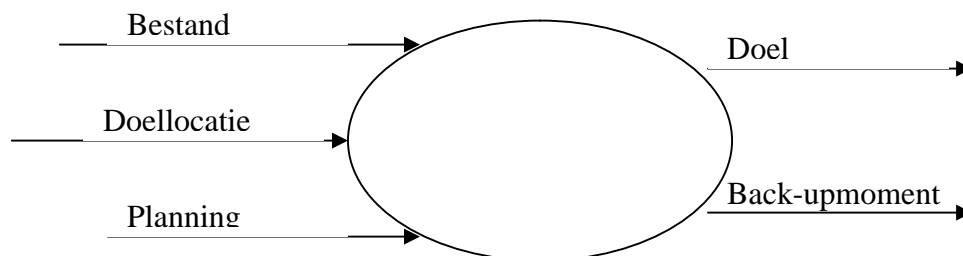
In de vorm van een stand-alone pakket, met locale back-up mogelijkheden wordt voor de aanschaf eenmalig betaald. Daarnaast kan de gebruiker, door middel van een onderhoudscontract, in aanmerking komen voor diverse updates en upgrades. Een onderhoudscontract, met een jaarlijkse of maandelijkse betaling, voorziet Invers van een vaste bron van inkomsten.

In de vorm van een client-server pakket betaalt de consument voor het gebruik van opslagruimte. Met deze maandelijkse betaling kan Invers de opslagruimte voor de consument bekostigen, en wordt ook een vaste bron van inkomsten gerealiseerd.

5.3 Gegevensstromen

De beschrijving van het pakket heb ik omgezet in informatiestromen. Omdat er diverse hoofdkeuzes in het pakket zijn (back-up/ restore/ enz), heb ik er voor gekozen om per keuze een Data Flow Diagram (DFD) te maken. De DFD's geven een schematische weergave van de informatiestromen. De DFD's hebben weinig gezamenlijke interactie, waardoor ik het niet nodig vond om de losse DFD's te combineren in één DFD 0. Het maken van deze DFD's zorgt voor inzicht in de informatie die aan het pakket gegeven wordt en welke informatie het pakket als resultaat geeft. Omdat de technieken van Yourdon bekend zijn binnen Invers, was het niet noodzakelijk om de DFD's verder toe te lichten. De betrokken personen kunnen de informatie in de DFD's goed interpreteren.

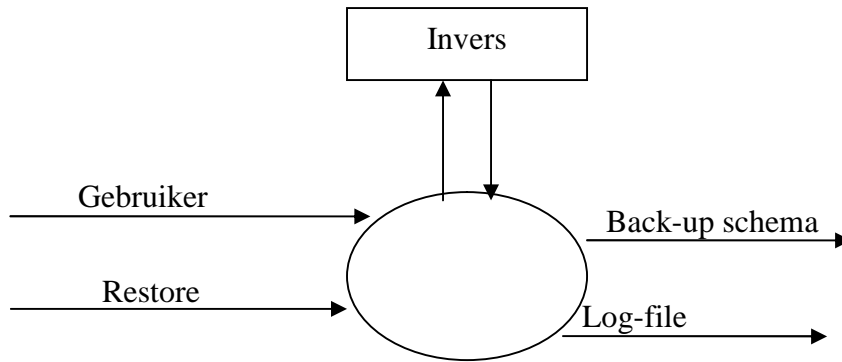
Een voorbeeld van een DFD is de DFD van de Snelle back-up.



Bestand	=	{pad} + (bestandnaam + extensie)
Back-upmoment	=	{dag}
Doel	=	{schijf + pad}
Doellocatie	=	schijf + pad + compressie + encryptie
Planning	=	[nu {dag} + herhaling]

Er wordt een pad van de lokale pc aangegeven ter back-up. Eventueel kan hier een bestand aan worden toegevoegd. De applicatie geeft aan welke locaties als doellocatie beschikbaar zijn. De gebruiker kiest één of meerder doellocaties. Ook voor de back-up momenten geeft de applicatie een keuzemogelijkheid. Deze mogelijkheden gebruikt de gebruiker voor het ingeven van de planning.

Na het uitvoeren van een back-up kan er een Snelle restore plaatsvinden.



Gebruiker	=	naam + [adres + huisnummer + postcode + woonplaats registratienummer]
Back-up schema	=	{TAAK}
Restore	=	TAAK

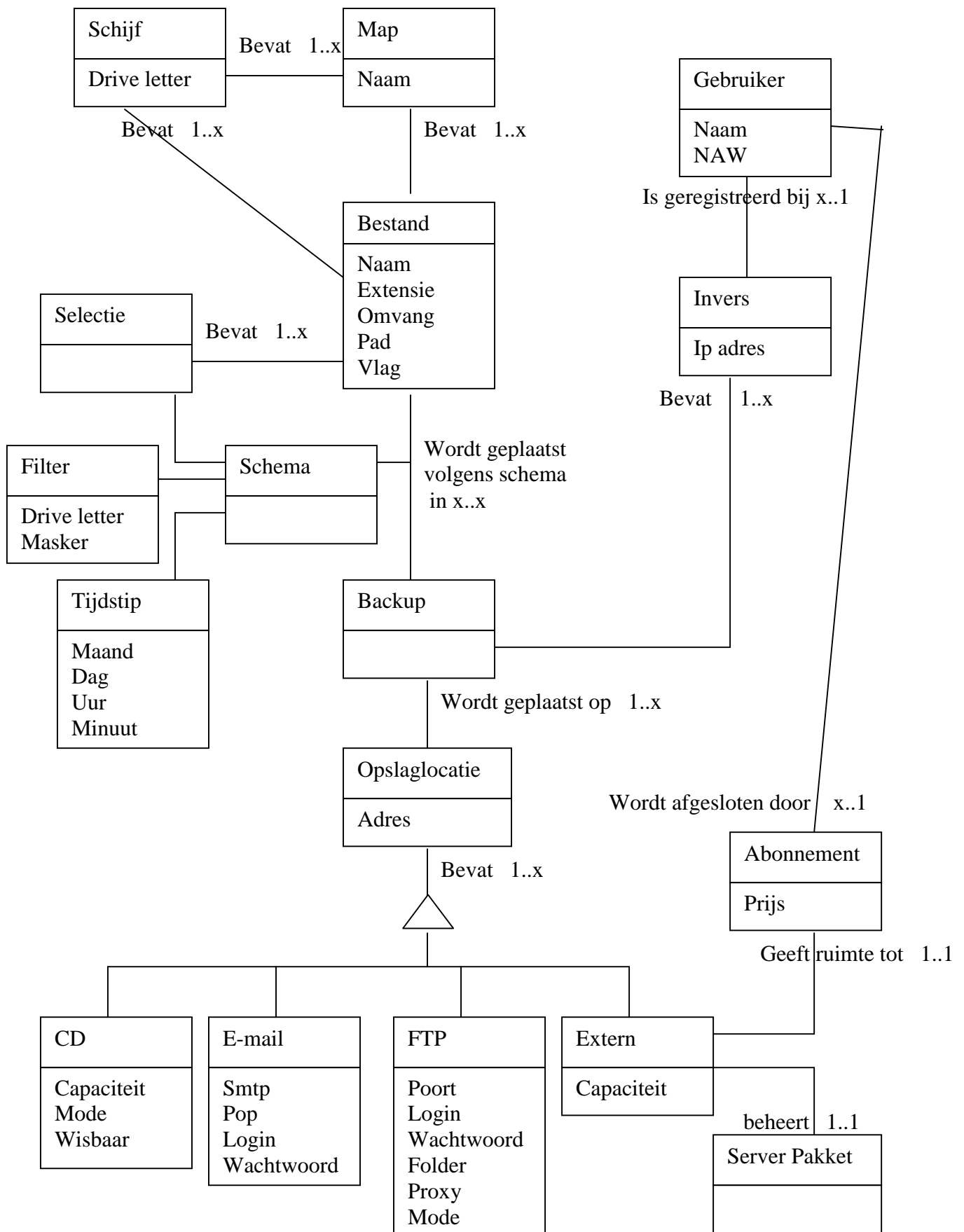
Ervan uitgaande dat de gebruiker geen gegevens over de back-ups heeft, geeft de gebruiker zijn gegevens op. Aan de hand van deze gegevens wordt er contact gezocht met Invers. De beschikbare back-up schema's worden aan de gebruiker getoond. De gebruiker geeft aan welke taak gerestored moet worden. Na het verwerken van de restore geeft de applicatie het resultaat weer in een logfile.

5.4 Klassendiagram

Na het in kaart brengen van de gewenste functionaliteit van de applicatie en communicatie van de gebruiker met de applicatie kunnen de klassen in kaart worden gebracht door middel van een Klassendiagram. Ik heb de tekst uit de pakketbeschrijving onderzocht op zelfstandige naamwoorden. Deze heb ik vervolgens gesorteerd in klassen en eigenschappen. Dit leverde de volgende lijst op:

<u>Klasse</u>		<u>Eigenschap</u>	
Pakket	Logfile	Locatie	Bestandextensie
Back-up	Selectie	Back-up functie	Vlaggen
Schijf	Tijdstip	Restore functie	Wildcards
Gebruiker	Browser	Instellingen van het programma	Administator
Bestand	Abonnement	Gebruikersgegevens	Andere gebruiker
Schema/ taak	Extern programma	Opslag locatie	Rechten
Opslagmethode	Test	Standaard	Disc-Spanning
- Lokaal	Verwijderd bestand	instellingen	CD-code
- e-mail	Pc	Mappen structuur	Timestamp
- ftp server	Filter	Prioriteit	Historie
- cd	Invers server	Benodigde ruimte	Bewaartijd
- externe locatie	Map	Procesvoortgang	Versies
		Overzicht	Beveiliging
		Taakoverzicht	Wachtwoord
			Compressie
			Encryptie

De eigenschappen heb ik vervolgens aan de klassen toegekend. Sommige klassen en eigenschappen hebben niet direct betrekking op het verdere ontwikkeltraject, en heb ik daarom weggelaten.



Dit klassendiagram geeft een goede basis voor de verdere indeling van het pakket. De intern te gebruiken klassen zijn af te leiden, maar ook wordt er aangegeven wat de externe klassen zijn waar een database voor opgezet moet worden wanneer het product commercieel uitgebracht wordt. In het klassendiagram heb ik de operaties van de klassen weggelaten. Omdat de uiteindelijke werking van het pakket nog niet bekend is, kan hier beter mee worden gewacht totdat de indeling en de gewenste interactie tussen de klassen bekend is.

5.5 Systeemarchitectuur

Voor de ontwikkeling van de applicatie heb ik gekozen voor het platform Windows XP en de programmeeromgeving Delphi 5. Tijdens het ontwikkelen is het belangrijk dat de pc stabiel is, en dat eventuele fouten niet direct ernstige gevolgen hebben. Daarnaast moet de ontwikkelde software door de doelgroep gebruikt kunnen worden. De doelgroep maakt over het algemeen gebruik van Windows. Binnen de Windows productgroep is Windows XP momenteel het meest stabiele platform. Ter voorkoming van ernstige fouten tijdens de ontwikkeling, start Delphi de applicatie op in een beschermde omgeving. Indien er een fout in de code aanwezig is, of er onverwachts een fout optreedt, vangt Delphi deze fout af en zorgt ervoor dat de oorzaak achterhaald kan worden, zonder de stabiliteit van het systeem aan te tasten. Door de ontwikkeling en het testen tijdens het programmeren, wordt het pakket geschikt gemaakt voor Windows XP. Daarnaast zal het pakket ook getest worden op Windows 2000 en pas worden goedgekeurd wanneer de werking op dit platform ook correct is. Na goedkeuring zal verder onderzocht worden of het pakket ook op lagere versies van Windows werkt.

Voor de ontwikkeling wordt uitgegaan van twee applicaties. De client applicatie wordt aan de klant geleverd en biedt de functionaliteit zoals deze in het functioneel ontwerp wordt beschreven. De client applicatie moet zowel in combinatie met de server als zonder server kunnen functioneren. Door de meeste functionaliteit naar de client te verplaatsen, wordt voorkomen dat er functionaliteit in beide pakketten wordt ingebouwd. De server applicatie zorgt voor registratie en de bewaking van de bestanden.

Door de back-up methoden als losse componenten te ontwikkelen, wordt voldaan aan de wens om de portabiliteit hoog te houden. De verwerking van de lokale gegevens is een algemene functie van het pakket, en is voor elke back-up methode nodig. Vervolgens wordt het wegschrijven van de aangeboden bestanden overgelaten aan de losse componenten. De losse componenten kunnen op deze manier ook in andere pakketten worden toegepast. Daarnaast levert de splitsing van het pakket een groot voordeel op bij het updaten van het pakket. Bij een update hoeven alleen de losse onderdelen opnieuw aangeboden te worden. Dit zorgt voor minder data transfer van de servers, dan wanneer het volledige pakket gedownload en vervangen moet worden.

De standaard interface van Windows XP maakt gebruik van ronde vormen, felle kleuren. Daarnaast kent Windows de mogelijkheid deze instellingen aan te passen. Door dezelfde opzet te gebruiken voor de Back-up applicatie, zal de gebruiker snel een gevoel van herkenning krijgen.

Voor simpel gebruik wordt de standaard interface van Windows XP overgenomen. Voor de uitgebreidere instellingen wordt gekozen voor een weergave op de klassieke manier.

5.6 Prototyping

Binnen DSDM is de interactie met de gebruiker erg belangrijk. Door een regelmatige terugkoppeling te krijgen, kan de programmatuur tijdens de ontwikkeling worden aangepast, waardoor er achteraf geen grote wijzigingen meer doorgevoerd hoeven te worden. Doordat er, zoals aangegeven in het begin van dit hoofdstuk, nog geen bestaande gebruikersgroep is, heb ik ervoor gekozen de terugkoppeling te verkrijgen van medewerkers van Invers. Deze terugkoppeling moet samen met het concurrentie onderzoek voldoende input geven om de applicatie gebruiksvriendelijk te maken. Voor het verkrijgen van de terugkoppeling heb ik een standaard formulier opgesteld.

Pakketnaam:	BackupTools	Versie:	0.0.1.____
Datum:		Build:	_____

Onderdeel	Oordeel
<div> <div>Functie</div> <div> <div></div> <div></div> </div> </div>	<div>--/-/0/+ /++/nvt</div> <div></div>
<div> <div>User Interface</div> <div> <div></div> <div></div> </div> </div>	<div>--/-/0/+ /++/nvt</div> <div></div>
<div> <div>Gebruiksvriendelijk</div> <div> <div></div> <div></div> </div> </div>	<div>--/-/0/+ /++/nvt</div> <div></div>
<div> <div>Technische werking</div> <div> <div></div> <div></div> </div> </div>	<div>--/-/0/+ /++/nvt</div> <div></div>
<div> <div>Overige</div> <div> <div></div> <div></div> </div> </div>	<div>--/-/0/+ /++/nvt</div> <div></div>

In dit formulier is ruimte om de wijzigingen sinds het laatste prototype aan te geven. Zo weet de tester welke onderdelen er getest moeten worden. Daarnaast ontvang ik de feedback schriftelijk, waardoor het tijdens de ontwikkeling altijd nageslagen kan worden.

De volledige bedrijfsanalyse heb ik naar diverse personen binnen Invers verstuurd en de inhoud besproken.

6 Functioneel model

De besprekingen over de bedrijfsanalyse waren positief. De inhoud was goed te begrijpen en de beschrijving van het product sloot goed aan bij de wensen. Bij de eerste release van de software zou een volledige integratie van de beschreven functionaliteit echter te overweldigend zijn voor de eindgebruikers. Een introductie van het pakket in stappen, waarbij elke stap meer functionaliteit biedt, sluit meer aan bij de werkwijze van Invers. Daarnaast wil Invers een stabiele inkomstenbron realiseren door abonnementen met gebruikers af te sluiten. Dit wordt het best verwezenlijkt door de functionaliteit in de eerste release van de applicatie beperkt te houden. De eerste release moet één back-up methode bevatten, namelijk de opslag op een remote server. Door deze eis kan een groot gedeelte uit de bedrijfsanalyse worden overgeslagen. Door de inhoud niet aan te passen, kan de bedrijfsanalyse wel worden gebruikt om de ontwikkeling van het pakket in de toekomst richting te geven.

6.1 Interactie

Voor de interactie met de gebruiker blijf ik uitgaan van dezelfde invalshoek. De gebruiker kan een eenvoudige of uitgebreide interface selecteren. Van de belangrijkste schermen heb ik een schermvoorbeeld gemaakt. Omdat de clientapplicatie het belangrijkste verkoopmiddel is van het back-up pakket, wordt aan de vormgeving hiervan de meeste aandacht besteld. De twee belangrijkste schermen van de applicatie zijn het overzichtsscherm, waarin de diverse back-up taken zien zijn, en het scherm waarin een nieuw back-up schema wordt aangemaakt. Het voorbeeld van het overzichtsscherm geeft een indruk van de interactie die voor de gebruiker mogelijk is. De tekst erbij geeft aan wat er getoond wordt en wat hiermee gedaan kan worden. Door de standaard lay-out van Invers pakketten aan te houden, is de bedoeling van de schermen direct duidelijk.

In de uitgebreide versie krijgt de gebruiker een overzicht van de aanwezige schema's te zien, met daarbij de aanwezige mogelijkheden. Omdat het back-uppen in dit overzicht de belangrijkste functionaliteit van het pakket is, wordt hier de aandacht op gelegd.

In dit overzicht kan een nieuw back-up schema worden toegevoegd, of een reeds aanwezig schema worden gewijzigd of verwijderd.

Ook kan vanuit dit hoofdscherm een back-up gerestored worden. Afhankelijk van de instellingen worden de beschikbare back-up bestanden of schema's getoond.

Ook kan de gebruiker vanuit het hoofdscherm direct naar de website of naar de helpdesk van het pakket.

The diagram illustrates a web page layout with the following components:

- Header:** A horizontal bar at the top containing four circular icons.
- Sidebar:** A vertical column on the left side containing three rectangular buttons and several horizontal lines.
- Main Content Area:** A large rectangular area on the right side, featuring five horizontal lines at the top and a large empty space below.

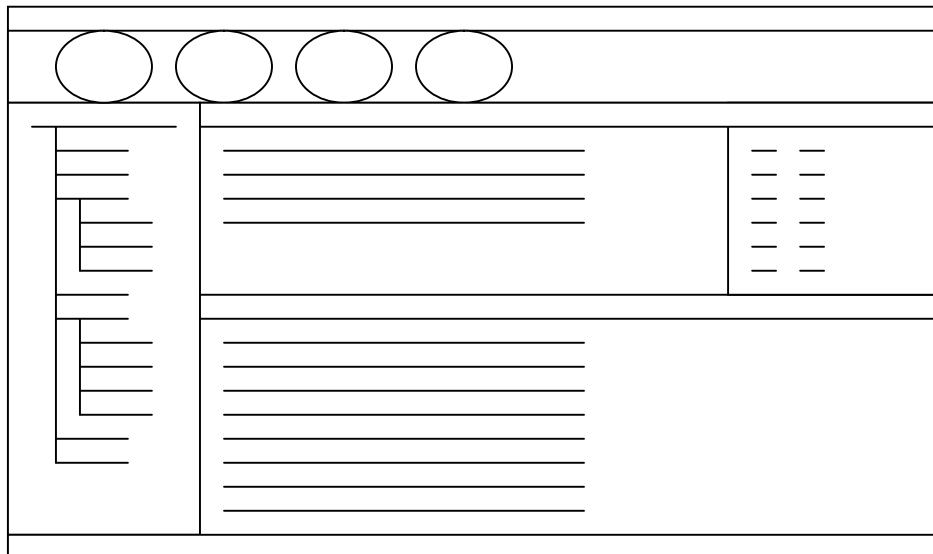
Bij het aanmaken van een nieuw schema wordt er een scherm over het schema overzicht heen geopend. Hierin wordt de gebruiker door middel van een wizard door de te volgen stappen heen geleid. Bij het wijzigen van een schema wordt ditzelfde scherm gebruikt, maar kunnen alle opties direct benaderd worden.

Het voorbeeld voor het scherm waar een nieuwe back-up wordt aangemaakt is gelijk van opzet.

In een overzicht worden de gekozen / te kiezen bestanden en mappen getoond. Dit overzicht wordt getoond in een mappenstructuur. De inhoud van de mappen wordt ook getoond. Een derde overzicht toont de geselecteerde mappen en bestanden. Door middel van een filter kan sneller gekozen worden.

Naast de mappenstructuur kunnen diverse extra's opgeslagen worden, zoals de favorieten, Outlook (Express) bestanden het Windows adresboek. Deze standaarden kunnen later toegevoegd worden.

In de standaard instelling voor de easy versie wordt de map 'Mijn documenten' bewaard, exclusief *.exe en *.bak bestanden.



Locatie

De optie om een locatie te kiezen is niet aanwezig in de eerste versie. Er wordt altijd geback-upped naar een remote server. Er moet wel rekening mee gehouden worden dat een andere locatie ingesteld kan worden.

Interval

Het tijdsinterval van het schema wordt in het volgende scherm ingesteld. Er kan gekozen worden voor een vast of variërend tijdstip op een dag van de week of maand:

Dagelijks – weekdays te selecteren

Maandelijks – dagnummers te selecteren

Daarbij kan worden aangegeven of de pc bij het afsluiten moet back-uppen of dat de pc na het back-uppen moet afsluiten.

Voor het starten van de back-up kan een programma uitgevoerd worden of gesloten worden. Ook kan na het uitvoeren van de back-up een programma gekozen worden.

Easy versie: De voorgestelde interval is of dagelijks op een vast tijdstip, of bij het afsluiten van de pc.

Backupwijze

Voor verschillende back-up schema's kunnen verschillende soorten back-ups gemaakt worden:

Full: Alle bestanden (archive bit wordt omgezet)

Incrementel: Gewijzigde bestanden (archive bit wordt omgezet)

Differentieel: Gewijzigde bestanden (archive bit wordt niet omgezet)

Mirror: Twee locatie worden gelijk gemaakt, met/zonder overschrijving in client/server.

Easy versie: Het gebruik van de incrementele back-up wordt aangeraden. Bij de eerste back-up wordt hierdoor een full back-up gemaakt, maar in het vervolg worden alleen de wijziging doorgevoerd.

Alleen in de uitgebreide versie kan worden gekozen voor een historie optie. Gekozen kan worden tussen het behouden van bestanden op versienummer (laatste x versies) of tijdsduur (bestanden ouder dan x dagen verwijderen).

Encryptie en compressie.

Omdat er op een server van een derde partij wordt geback-upped, moeten de gegevens altijd gecomprimeerd en beveiligd worden. Om een efficiënte transport en opslag te verwezenlijken, moeten de bestanden lokaal gecomprimeerd worden. Vervolgens kunnen de bestanden verzonden worden. Op de server kunnen de bestanden opnieuw gerangschikt worden indien dit nodig is.

De bestanden moeten vòòr compressie beveiligd worden, zodat de informatie op de server niet toegankelijk gemaakt kan worden zonder de juiste beveiligingssleutel.

Door deze schermvoorbeelden kan er direct commentaar geleverd worden op de opzet van de schermen. De gedetailleerde invulling van deze schermen wordt uitgevoerd tijdens het testen van de prototypen.

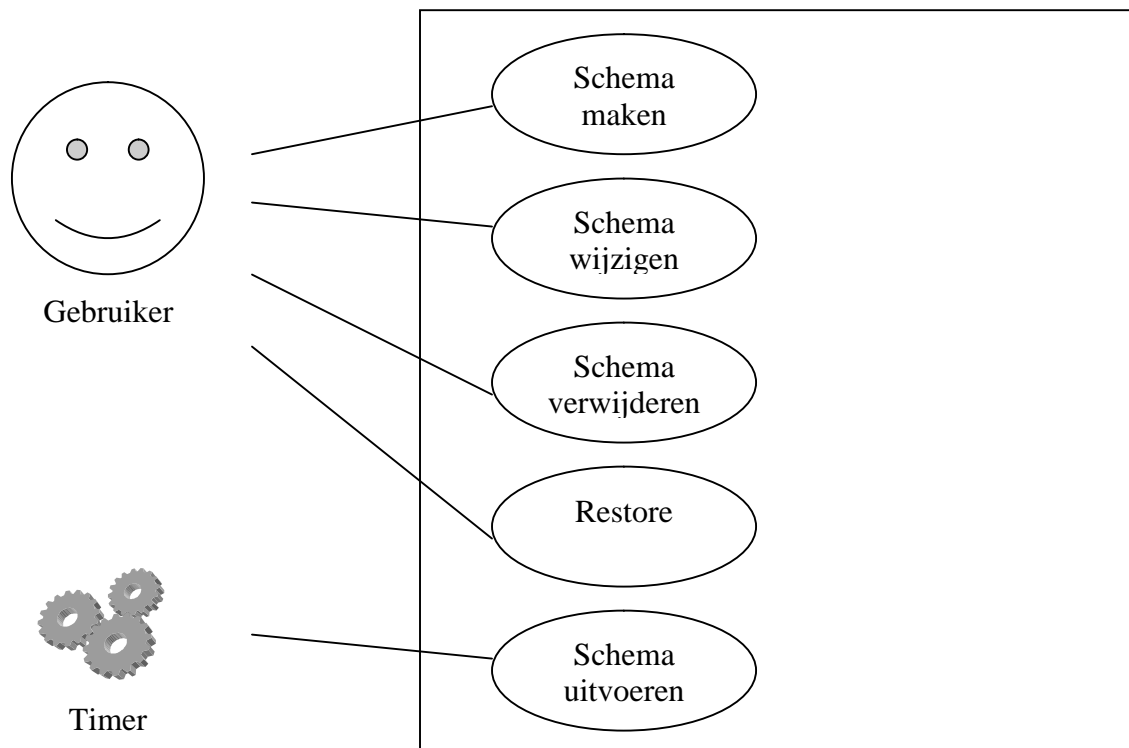
6.2 Functiestructuur

De functies van het pakket heb ik aangegeven door middel van use cases en toestanddiagrammen. In de use cases worden de gebeurtenissen in het pakket weergegeven. Hierbij wordt tevens aangegeven waardoor de gebeurtenis wordt veroorzaakt en wat er vervolgens gebeurt. Indien er uitzonderingen kunnen optreden, worden deze benoemd. Als laatste wordt het gewenste resultaat ingevuld.

Naam	1. Schema maken
Samenvatting	Gebruiker maakt een nieuw back-up schema aan
Actoren	Gebruiker
Aannamen	Pakket is actief
Beschrijving	(1) De gebruiker kiest de optie om een nieuw schema aan te maken. Door verschillende instellingen te doen worden de eigenschappen van het schema bepaald. (2) Het systeem controleert of alle

	instellingen aanwezig zijn. Het systeem slaat het schema vervolgens op.
Uitzonderingen	[instelling afwezig] Als er een instelling niet is ingevuld wordt deze instelling aan de gebruiker getoond, om de instelling te voltooien. Hierna wordt de use-case verder uitgevoerd
Resultaat	Er is een nieuw schema aangemaakt.

Wanneer een nieuw schema wordt aangemaakt, geeft de gebruiker dit aan. Er wordt vervolgens een aantal stappen doorlopen. Indien de controle geen goed resultaat oplevert, wordt de uitzondering uitgevoerd. Uiteindelijk resulteert dit in een nieuw schema. De actor wordt opgenomen in het overzicht van actoren met de applicatie.



Naast de gebruiker als actor kent het pakket ook een timer als actor. Deze timer controleert of er een back-up uitgevoerd moet worden.

Naam	5. Schema uitvoeren
Samenvatting	Een back-up schema wordt uitgevoerd
Actoren	Timer
Aannamen	Het pakket draait op de achtergrond
Beschrijving	Zodra een ingesteld tijdstip is aangebroken wordt het bijbehorende schema uitgevoerd. Het proceslog wordt bijgewerkt.
Uitzonderingen	[verbroken verbinding] Als er geen verbinding met de server kan worden gemaakt, wordt de het schema na 5 minuten opnieuw uitgevoerd. Na 10 mislukte pogingen wordt de gebruiker geïnformeerd en de uitvoer afgebroken
Resultaat	De bestanden zijn op de server geplaatst.

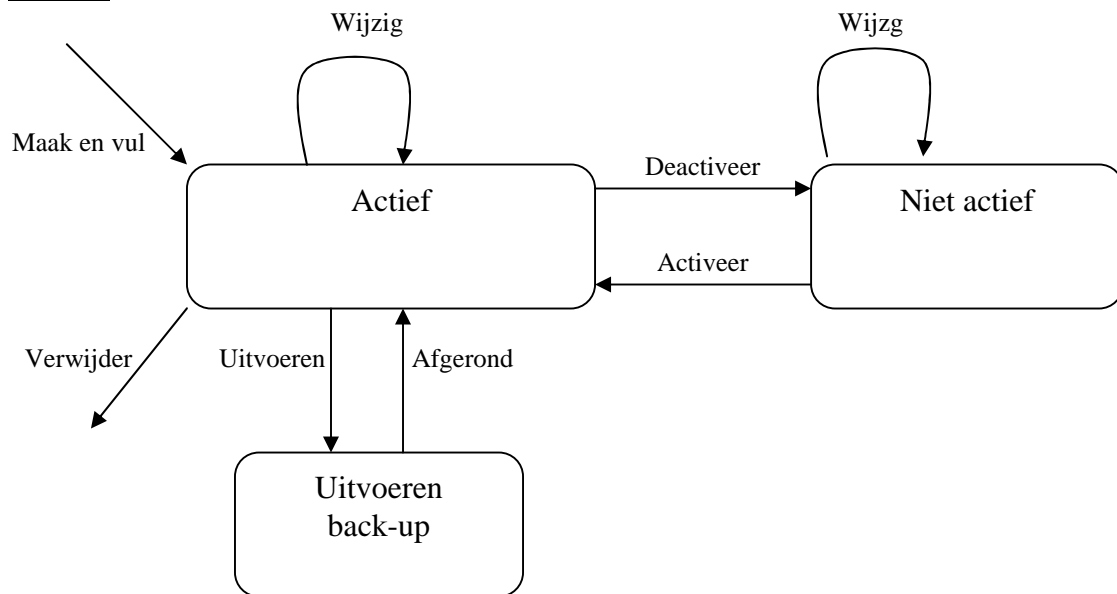
De vijf gemaakte use cases gaan over de belangrijkste en regelmatig terugkerende onderdelen van het pakket. Daarnaast moet het voor de gebruiker mogelijk zijn om

algemene instellingen aan het pakket te wijzigen. Omdat dit geen betrekking heeft op de belangrijkste functionaliteit, heb ik hier geen use cases van gemaakt. Deze instellingen zullen aan het pakket worden toegevoegd naar aanleiding van de testresultaten.

6.3 Toestandsdiagram

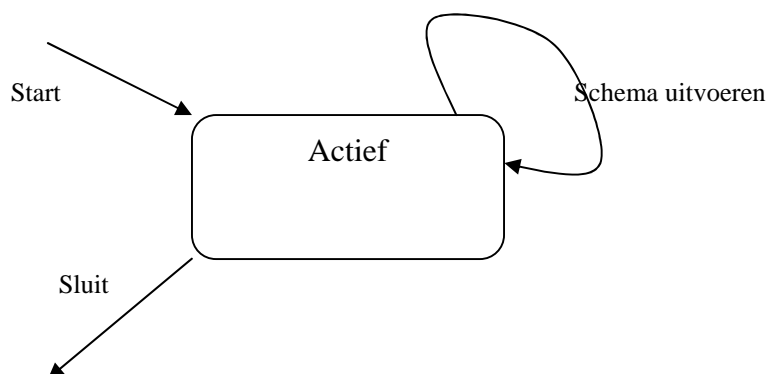
In de toestandsdiagrammen heb ik de belangrijkste objecten opgenomen. Hierin worden de verschillende toestanden van de objecten weergegeven.

Schema



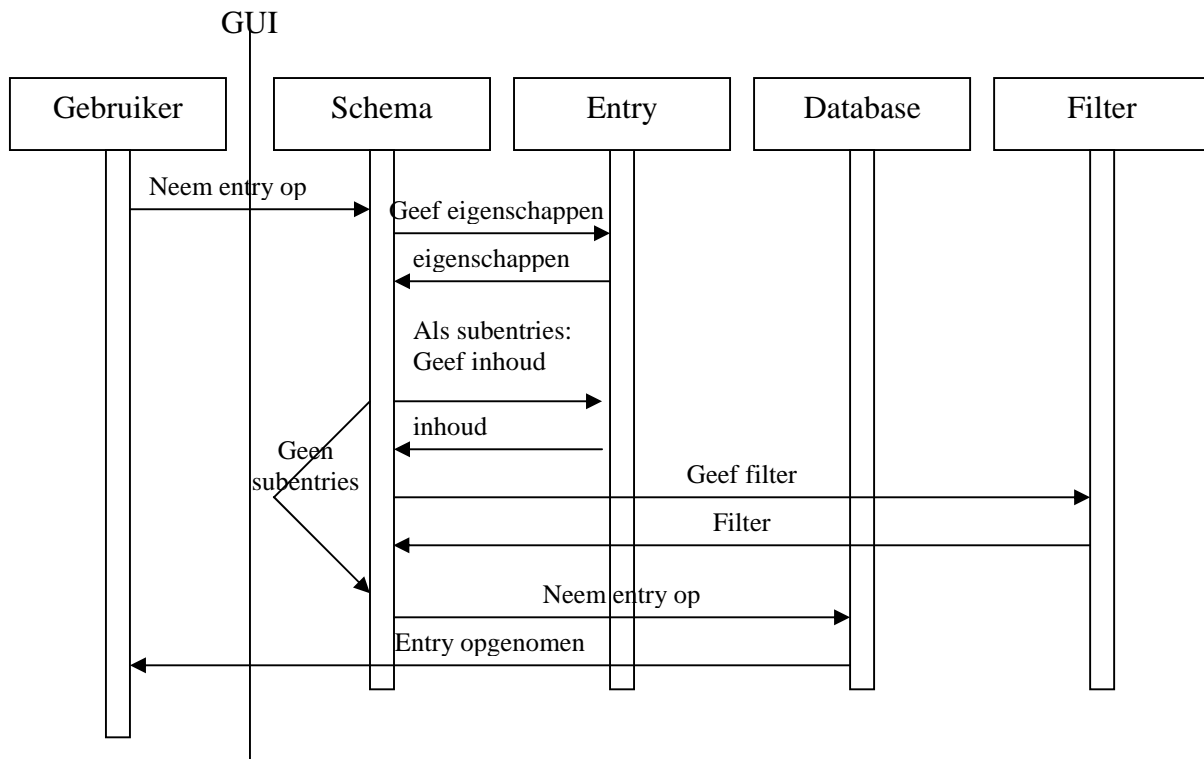
Alleen wanneer het schema actief is, kan het een back-up uitvoeren. Dit uitvoeren gebeurt wanneer de timer het aangeeft.

Timer



6.4 Sequence diagram

In de sequence diagrammen wordt de interactie tussen de klassen in de applicatie getoond. Ook hier heb ik gekozen om enkele sequence diagrammen te maken. Een voorbeeld van zo'n sequence diagram is het toevoegen van een entry aan een schema.



De gebruiker geeft een entry (bijvoorbeeld een map of een bestand) op aan het schema om toe te voegen. Het schema vraagt de eigenschappen hiervan op en neemt deze op in het schema. Eventueel worden subentries (bijvoorbeeld bestanden in een map) ook opgenomen in het schema.

6.5 Activiteitsdiagram en Prototyping

Volgens mijn planning zou ik in het functioneel model ook activiteitsdiagrammen en de werkwijze van prototyping opnemen. Prototyping heb ik verplaatst naar de bedrijfsanalyse. Deze verplaatsing naar de bedrijfsanalyse zorgt voor inzicht in het verloop van het ontwikkeltraject en de gewenste respons van de betrokken partijen in een vroeg stadium van het project.

Het activiteitsdiagram heb ik volledig overgeslagen. Het activiteitsdiagram geeft de flow of control weer. Omdat de volgorde van activiteiten door zowel de bedrijfsanalyse en het functioneel ontwerp duidelijk genoeg is uitgelegd, voegt het activiteitsdiagram weinig extra informatie. De toevoeging van een activiteitsdiagram zou ervoor zorgen dat de stromen overzichtelijker in beeld worden gebracht. Wanneer er met een team aan een applicatie gewerkt wordt, kan dit overzicht noodzakelijk zijn. Omdat ik als enige programmeur aan het pakket zal werken, is deze afstemming met andere programmeurs niet noodzakelijk.

7 Technisch ontwerp

Het technisch ontwerp en de ontwikkeling van de applicatie heb ik synchroon uitgevoerd. Het functionele model bevatte al veel gegevens die normaal niet door Invers op papier worden gezet tijdens de ontwikkeling van een applicatie. Het opzetten van schermvoorbeelden met een beschrijving van de functionaliteit is vaak het enige dat op papier wordt gezet. Omdat de code als een goed boek moet lezen, zit het technisch ontwerp verwerkt in de code van de applicatie, en hebben de programmeurs de vrijheid om de techniek achter een functie volledig zelf te bepalen. Door regelmatige wijzigingen in de functionaliteit tijdens de ontwikkelingsfase zou een technisch ontwerp te snel verouderen. Het omzetten van definitieve code naar een technisch ontwerp doet het doel van het technisch ontwerp teniet.

7.1 Planning

In mijn planning voor het technisch ontwerp en had ik de volgende onderdelen gepland:

- Beeldscherm en lijstindeling
- Opslagstructuur
- Code beschrijving
- Testset
- Prototype tbv van Server en Client
- Handleiding

De indeling van de beeldschermen is reeds in de bedrijfsanalyse weergegeven. De opzet uit de bedrijfsanalyse wordt gehandhaafd en een herhaling in het technisch ontwerp zou overbodig zijn. Als naslagwerk heeft het opzetten van schermindelingen geen nut, omdat de actuele schermindeling te allen tijde uit de broncode van de applicatie is op te vragen.

Over opslagstructuur van de applicatie moet goed nagedacht worden. Diverse klassen uit de applicatie moeten worden opgeslagen. Hierbij moet goed worden nagedacht waar de gegevens worden opgeslagen. Op de lokale pc kunnen de gegevens op drie locaties worden opgeslagen. Allereerst in de map waar de applicatie is geïnstalleerd wordt, een submap van Program Files. Daarnaast heeft Windows XP een map Mijn Documenten, waar documenten van de actieve gebruiker in worden opgeslagen. De derde locatie is de registry van Windows. Binnen Windows worden instellingen van de applicatie zoveel mogelijk opgeslagen in de Registry. Alle overige gegevens worden als bestanden op schijf opgeslagen. Binnen Invers wordt gebruik gemaakt van de Advantage database van Extended Systems.

Om gegevens van de applicatie ordelijk op te slaan op de harde schijf, is het aan te raden de data in een submap van de installatie directory op te slaan. Op de helpdesk van Invers zijn hier echter enkele problemen aan ontdekt in andere programma's. Wanneer een gebruiker beperkte rechten heeft, mag er niet in de Program Files directory geschreven worden. Hierdoor kunnen gegevens niet opgeslagen worden. Om het programma voor alle gebruikers op een pc toegankelijk te maken, moeten de gegevens worden opgeslagen in een submap van Mijn Documenten. Een bijkomend voordeel is dat elke gebruiker automatisch zijn eigen back-up schema's heeft, en dat schema's voor andere gebruikers worden afgeschermd.

De meest actuele wijzigingen aan het pakket worden aangebracht in de code. Het opnemen van een apart document waarin deze code wordt beschreven, zal in de

toekomst hoogstwaarschijnlijk voor inconsistentie zorgen. Om dit te voorkomen wordt de Delphi code voorzien van commentaar. Dit toevoegen van commentaar zorgt dat de code die wordt gelezen direct begrijpbaar is, zonder een naslagwerk erbij te moeten pakken. Wijzigingen in de code kunnen direct worden verwerkt in het bijgevoegde commentaar. Door het aantal handelingen dat nodig is om documentatie consistent te houden, kan worden voorkomen dat diverse documenten verschillen gaan vertonen.

Het opzetten van testsets is belangrijk voor de beoordeling van de applicatie. Binnen Invers wordt hier, mede door de diverse wijzigingen die tijdens het ontwikkeltraject worden doorgevoerd, zelden gebruik van gemaakt. De applicatie wordt door de ontwikkelaars getest op werking, waarna er een aparte test van de applicatie wordt gedaan. Na het doorstaan van deze test, zonder vooraf gedefinieerde testonderdelen, wordt de applicatie beschikbaar gemaakt voor het publiek.

Tijdens het programmeren zal er met regelmaat een prototype worden opgeleverd. Deze wordt voorgelegd aan diverse personen, zodat er feedback kan worden gegeven. De prototypen zullen aan de hand van het schema in het functioneel model worden beoordeeld.

Tijdens de testfase van de applicatie wordt ook de handleiding samengesteld. In deze handleiding worden de standaard mogelijkheden van de applicatie uitgelegd. Geavanceerde mogelijkheden worden niet opgenomen in de handleiding, om gebruikers niet teveel af te schrikken.

Om het technisch ontwerp als voorbereiding op het programmeren en als goed naslagwerk te laten fungeren, heb ik de inhoud beperkt gehouden. De opslaglocatie en opslagstructuur worden het beste opgezet door deze los van het programmeerwerk uit te denken. Het resultaat wordt opgenomen in het technisch ontwerp. Vanuit het technisch ontwerp wordt de database opgezet. Tijdens het programmeren kunnen hier nog wijzigingen in aangebracht worden. Deze zijn vervolgens bijgewerkt in het technisch ontwerp.

7.2 Must have's - should have's - could have's

Na aanleiding van het commentaar op het functioneel model heb ik de lijst van must have's, should have's en could have's opgesteld. Door deze naar aanleiding van het commentaar op het functioneel model op te stellen, kunnen deze wensen nog goed worden afgesteld op de indruk van de betrokken personen van het gewenste pakket. Deze lijst heb ik in het technisch ontwerp opgenomen en tijdens het programmeren aangepast indien dit nodig was. De eerste opzet hiervan was als volgt:

Onderdeel	Prioriteit
Functie	
Easy backup maken	+
- selectie maken	+
- back-up moment	+
Advanced back-up maken	0
- meerdere schema's	+
- selectie maken	+
- back-up moment	+
- back-up type	0
- back-up locatie	-
- historie aanleggen	-
Easy restore	+
- bestanden overzicht	+
- beveiliging	+
Advanced restore	0
- bestanden overzicht	+
- beveiliging	+
- herstellocatie	+
- historie	0
Server	+
- bestanden ontvangen en verzenden	+
- statistieken	0
- automatisch incasso	-

User Interface	
Easy interface	+
- map structuur	+
- map inhoud	0
- filter	-
- selectie	+
- overzicht	+
Advanced interface	0
- meerdere schema's	+
- map structuur	+
- map inhoud	0
- filter	-
- selectie	+
- overzicht	+

+ = must have

0 = should have

- = could have

Technische werking	
Verzenden en ontvangen bestanden	+
Draaien op achtergrond	+
Opstarten bij Windows start	+
Overige	
Interne toegang tot internetsite	0
Interne toegang tot support pagina	0

+ = must have

0 = should have

- = could have

7.3 Timeboxen

Het bovenstaande schema heb ik opgedeeld in perioden, zodat er tijdeenheden (timeboxen) ontstaan waarbinnen enkele must haves, should haves en could haves aanwezig zijn.

Voor het opleveren van mijn definitieve prototypen zijn negen weken gepland. De totale software bestaat uit drie onderdelen, namelijk de server applicatie en de easy interface en de advanced interface in de client applicatie.

Door deze driedeling aan te houden, ben ik op timeboxen van drie weken uitgekomen met de volgende inhoud:

Timebox 1: Server

Must have	Bestanden ontvangen en verzenden
Should have	Statistieken
Could have	Automatische incasso

Timebox 2: Easy interface (rekening houden met uitbreiding naar Advanced)

Must have	Selectie maken in mapstructuur Overzicht geselecteerde mappen Beveiliging Back-up moment Bestanden overzicht op de server Verzenden en ontvangen bestanden Draaien op achtergrond Opstarten bij Windows start
Should have	Bestanden selectie Interne toegang tot internetsite Interne toegang tot support pagina
Could have	Filter

Timebox 3: Advanced interface

Must have	Meerdere schema's Selectie maken Back-up moment Overzicht bestanden op server Beveiliging Herstellocatie Verzenden en ontvangen bestanden Draaien op achtergrond Opstarten bij Windows start
Should have	Back-up type Historie per bestand mogelijk Interne toegang tot internetsite Interne toegang tot support pagina
Could have	Back-up locatie Historie aanleggen

Timebox 2 en 3 bevatten teveel must have's. De oorzaak hiervan komt doordat diverse onderdelen voor zowel de easy als advanced interface gebruikt worden.

Dit kan op drie manieren opgelost worden. De prioriteit van diverse onderdelen moet verlaagd worden of de must have kan verplaatst worden naar een andere timebox, waardoor de timeboxen in evenwicht met elkaar komen. Als laatste kan een complete herziening van de timeboxen worden uitgevoerd.

Door de overlap in timebox 2 en 3, en de samenwerking tussen de server en de client applicatie, heb ik voor deze laatste mogelijkheid gekozen. Mijn hoogste prioriteit heb ik gelegd op de communicatie en bestandsoverdracht tussen de client en de server. Daarnaast heb ik enkele prioriteiten aangepast, bijvoorbeeld de mogelijkheid tot het maken van meerdere schema's in de advanced versie. De advanced interface heeft minder prioriteit dan de easy interface, waardoor ook automatisch de prioriteit voor meerdere schema's lager wordt. Door deze mogelijkheid echter direct op te nemen, en dit af te schermen in de easy versie, is de programmacode beter voorbereid op deze uitbreiding.

Timebox 1:

Must have	Server en Client Client	Bestanden ontvangen en verzenden Beveiliging
Should have	Server	Statistieken
Could have	Server	Automatische incasso

Timebox 2:

Must have	Client Client	Selectie in mappenstructuur Meerdere schema's
Should have	Client Client	Overzicht geselecteerde mappen Overzicht bestanden op server
Could have	Client Client	Bestanden selectie Filter

Timebox 3:

Must have	Client Client	Back-up moment Draaien op achtergrond
Should have	Client Client Client	Opstarten bij Windows start Back-up type Back-up locatie
Could have	Client Client Client	Interne toegang tot internetsite/ helpdesk Herstellocatie Historie

Deze vernieuwde timeboxen zorgen voor een beter overzicht en een betere opbouw van de applicatie.

7.4 Timebox 1

Om de server en de client goed samen te laten werken, moeten er afspraken worden gemaakt over de uit te wisselen informatie. Deze afspraken worden ondergebracht in het client-server protocol. Tekst in de commando's wordt kommagescheiden verstuurd. Het onderstaande schema geeft de belangrijkste commando's.

Verzender	Commando	Betekenis	Actie
Client	LOGIN + UserName + Password	Client meldt zich aan voor communicatie	S: Controleer login in gebruikersdatabase
Server	LGNOK	LOGIN OK: Login goedgekeurd	S: Verbinding blijft behouden
Server	LGNFL	LOGIN FULL: gebruiker is reeds ingelogd	S: Verbinding wordt verbroken
Server	LGNUK	LOGIN UNKNOWN: Gebruiker niet bekend	S: Verbinding wordt verbroken C: Opstarten registratieprocedure
Server	LGNWP	LOGIN WRONG PASSWORD	S: Verbinding wordt verbroken C: Wachtwoord

			opnieuw vragen
Client	LOGOF + UserName	LOG OFF: gebruiker logged uit	S: verwijder verbinding uit verbindingen
Client	LOGNW + UserName + Password + FirstName + MiddleName + Lastname + Street + Housenumber + HousenumberAddition + Address2 + Zip + City + Country + Accountnumber + Paymentmethod	LOGIN NEW: Nieuwe gebruiker meldt zich aan met de vereiste gegevens	S: Voeg gebruiker toe aan de database
Server	LOGAD	LOGIN ADDED: Gebruiker is toegevoegd	

Wanneer er een verbinding tot stand is gekomen, kan de client een bestand naar de server versturen, een overzicht opvragen van de bestanden op de server of geselecteerd bestand terug laten sturen. Voor het versturen wordt een aparte socket opgezet.

Verzender	Commando	Betekenis	Actie
Client	FLSND + UserName + Naam + path + omvang + checksum	SEND FILE: bestandsgegevens	S: Voeg gegevens toe aan de bestandsdatabase. Vraag bestand op
Client	FLGDL + UserName	GET DIRECTORY LIST: vraag bestandslijst van de server op	S: Verstuur de aanwezige bestanden van de gebruiker
Client	FLGET + UserName + number	GET FILE: vraag bestand	S: verstuur bestand
Server	FLSDR	Send Requested File	C: sla bestand op op gewenste locatie
Client/Server	SIZE!	Size of File Omvang van het bestand	S/C reserveer ruimte

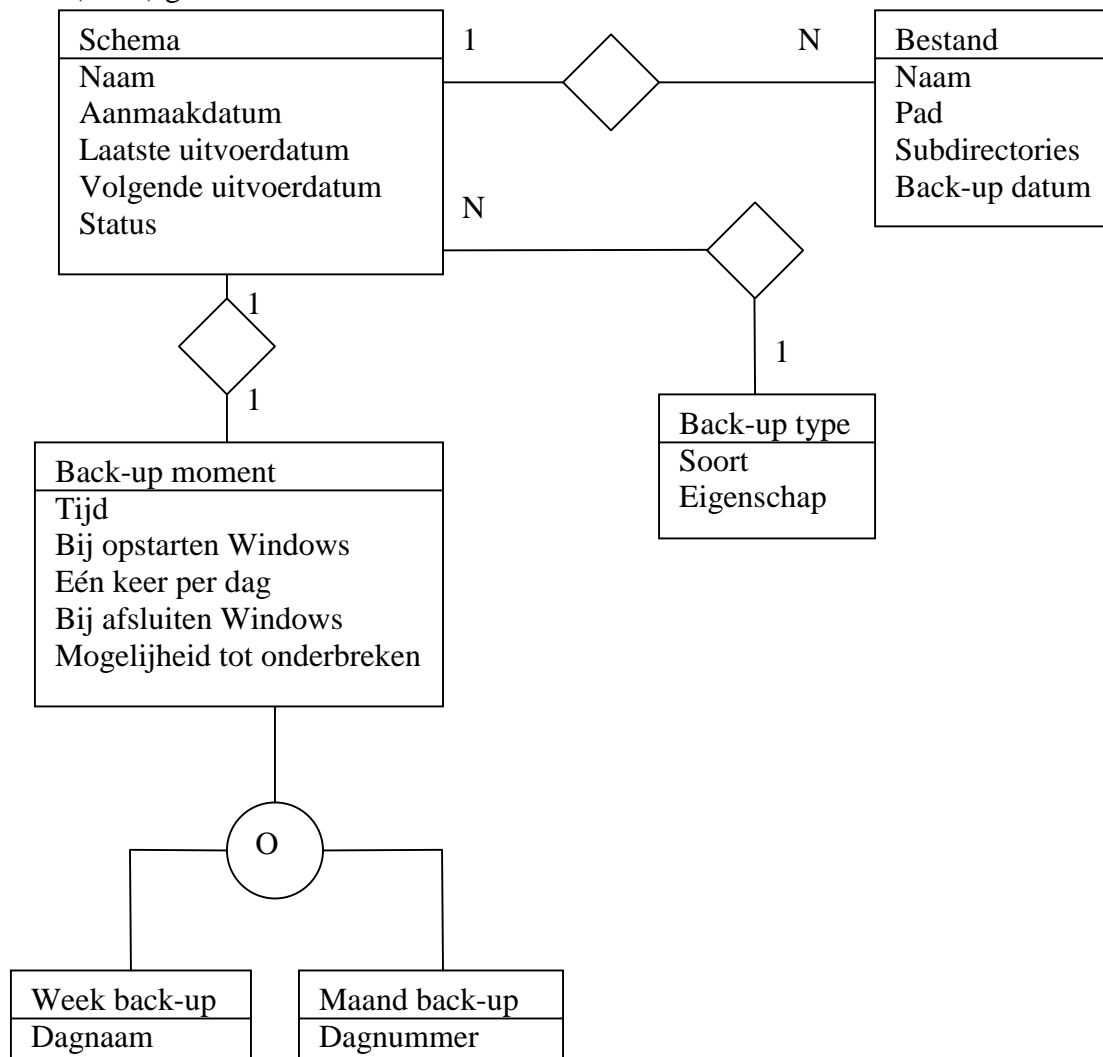
Daarnaast moeten gebruikersgegevens worden opgeslagen op de server om het verbruik aan de juiste gebruiker in rekening te kunnen brengen.

Users

Username	20
Password	40 (HASH value)
GUID	12
FirstName	20
MiddleName	10
Lastname	20
Street	20
Housenumber	8
HousenumberAddition	8
Address2	20
Zip	8
City	20
Country	20
Accountnumber	20
Paymentmethod	4 (integer)

7.5 Timebox 2

In timebox 2 moet de meeste aandacht besteed worden aan de opslag van de schema's en het toevoegen van mappen en bestanden aan dit schema. Om het verband tussen de schema's en de mappen in kaart te brengen heb ik een Entity Relationship Diagram (ERD) gemaakt.



Het schema bevat meerdere bestanden. Voor de bestanden neem ik een aparte tabel op. Omdat de back-up typen beperkt zijn, heb ik ervoor gekozen deze vast te definiëren in de tabel, waardoor queries op de tabel sneller worden uitgevoerd. Omdat elk schema maar één back-up moment heeft, waarin optioneel diverse dagen per week of per maand kunnen worden opgenomen, kon ik kiezen tussen het opslaan in één tabel of het aanmaken en koppelen voor aparte tabellen voor de dagnamen of dagnummer. Om de database structuur snel en eenvoudig te houden, heb ik voor elke mogelijke dagnaam en dagnummer een veld in de tabel opgenomen. Door gebruik te maken van velden van het type boolean, blijft de omvang sterk beperkt. De tabel voor de schema's komt er in de implementatie als volgt uit te zien:

Veldtype	Veldnaam
Integer	id
String	naam
Datum	Aanmaakdatum
Datum	Laatste datum
Datum	Volgende datum
Integer	Status
Tijd	Tijdstip
Boolean	Bij opstarten
Boolean	Één per dag
Boolean	Bij afsluiten
Boolean	Afbreken mogelijk
Boolean	Maandag
Boolean	..
Boolean	Zondag
Boolean	Dag 1
Boolean	..
Boolean	Dag 31

Dit levert records op die maximaal 76 bytes (7 dagen per week + 31 dagen per maand * 2) groter zijn dan de minimale omvang van records waar dagelijks een back-up wordt gemaakt. De vermindering van het aantal tabellen en koppelingen tussen tabellen zorgt echter voor een beter overzicht tijdens het programmeren, vooral tijdens het updaten van de records.

7.6 Timebox 3

Het uitvoeren van een back-up op de achtergrond op een ingestelde tijd heeft de hoogste prioriteit in timebox 3.

De instellingen in het programma worden opgeslagen in de registry. Het bijhouden van de opslaglocatie en de opslagwijze is erg belangrijk voor latere naslag. In een schema in het technisch ontwerp worden de diverse instellingen bijgehouden:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run	application.Title	application.exename
HKEY_CURRENT_USER\Software\BackupTools\		
other	ShowSplash	Boolean
other	RunOnStartup	Boolean
other	MinimizeToTray	Boolean
view	ShowNavBar	Boolean
view	EasyInterface	Boolean
user	FirstName	String
user	LastName	String

8 Programma ontwikkeling

Dit hoofdstuk beschrijft het omzetten van de rapporten naar Delphi programmatuur. Er wordt begonnen met de getroffen voorbereiding. Vervolgens worden de diverse timeboxen besproken.

8.1 Voorbereiding

Twee jaar geleden heb ik voor mijn stage in Delphi gewerkt. Mijn kennis van object georiënteerd programmeren was toen nog zeer beperkt en deze heb ik vooral uit het gebruikte handboek voor Delphi gehaald. Tijdens het laatste jaar aan de Haagse Hogeschool is de wijze van programmeren in een object georiënteerde omgeving uitgelegd. Deze kennis kwam me nu goed van pas. Omdat ik al redelijk lang niet meer in Delphi had gewerkt, ben ik eerst begonnen met het opnieuw wegwijs worden in Delphi. Hiervoor heb ik het boek Mastering Delphi van M. Cantù gebruikt. Hierin wordt Delphi zeer uitgebreid besproken. Door de diverse oefeningen in het boek te doen, of de uitgewerkte opgaven door te nemen, werd de werking van Delphi me weer duidelijk. Het doorlezen van dit boek zorgde later ook voor een goed naslagwerk. Naast dit boek, heb ik ook het boek Basiscursus Delphi van M. Stefanski doorgelezen. Dit zorgde dat ook mijn basiskennis van Delphi weer op het benodigde niveau kwam.

De Delphi interface toont een overzicht van de beschikbare componenten. Elke component, bijvoorbeeld een knop of een tekstvak, kan op een formulier (Form) geplaatst worden. Van de componenten worden in de Object Inspector de eigenschappen (properties) en de gebeurtenissen (events) getoond. De eigenschappen, bijvoorbeeld grootte en tekst, kunnen tijdens het ontwerpen van het programma worden aangepast. In de code kunnen deze eigenschappen ook worden aangeroepen en gewijzigd. De events kunnen tijdens het uitvoeren van het programma worden aangeroepen. Wanneer er bijvoorbeeld op een knop wordt geklikt, kan een stuk code worden uitgevoerd.

De oefeningen die ik heb uitgevoerd, hadden voor het grootste gedeelte betrekking op de applicatie die ik ontwikkel. Naast de oefeningen uit het boek heb ik diverse kleine programmaatjes geschreven om de functionaliteit van diverse niet-standaard Delphi componenten te leren kennen. Ook heb ik in deze programmaatjes onderdelen uitgetoetst die ik later nodig zou hebben, zoals het verbergen van een applicatie, of het in de system tray actief houden en toegang verschaffen tot de applicatie.

8.2 Timebox 1

Timebox 1 heeft als prioriteit het beveiligen, versturen en ontvangen van bestanden. Voor het versturen van de bestanden moet er een verbinding tussen de client en de server worden opgesteld. Om de opslaglocatie op de server aan te geven, moet de gebruiker met zijn persoonlijke gegevens worden aangemeld en opgeslagen. Indien de gebruiker bekend is op de server, kunnen bestanden direct worden verstuurd of opgehaald. Binnen Delphi kan voor het uitwisselen van informatie tussen de client en de server gebruik gemaakt worden van een socket. Met een socket kan tekst of binaire informatie worden verstuurd en ontvangen. Voor de aanmeldprocedure van een nieuwe gebruiker maak ik gebruik van een client- en serversocket. De communicatie verloopt volgens het protocol in het technisch ontwerp.

De server applicatie moet altijd actief zijn. Wanneer de applicatie opstart, wordt ook de serversocket opgestart. Deze wacht vervolgens totdat er een verbinding door een clientsocket wordt aangevraagd. De client zal vervolgens zichzelf aanmelden. Om dit

tijdens de ontwikkeling te kunnen testen, heb ik het communicatie onderdeel van zowel de server als client gelijktijdig geprogrammeerd.
De client zal na aanmelden zich proberen in te loggen:

```
procedure TFormConnection.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  if ClientSocket1.Active then
  ..
  begin
    ClientSocket1.Socket.SendText(StringEncrypt(('LOGIN'+
edtUser.Text + ';' + Hash(edtPassword.text)), Cypher));
    EdtSendText.Text := 'LOGIN'+ edtUser.Text + ';' +
Hash(edtPassword.text) + ';';
  end
  else
    MessageDlg('Error: You are not connected', mtError, [mbOK], 0)
end;
```

De serversocket controleert de verstuurde gegevens en verstuurt de bijbehorende respons:

```
procedure TfrMain.ServerSocket1ClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var
  strSocketText, strCommand, strComText: string;
begin
  strSocketText := StringDecrypt(Socket.Receivetext, Cypher);
  strCommand := Copy(strSocketText, 0, 5);
  strComText := Copy(strSocketText, 6, Length (strSocketText) - 5);
  Communicate (Socket.RemoteHost + ': ' + strCommand + ': ' +
    strComText);
  if strCommand = 'FLSND' then //SEND FILE
  ..
  else if strCommand = 'FLGET' then //GET FILE
  ..
  else if strCommand = 'LOGIN' then //LOGIN
    //strComText = User;PassWord
    begin
      if (CheckLogin(GetStringPart(strComText, ';', 1),
        GetStringPart(strComText, ';', 2)) = 'unknown') then
      begin
        Socket.SendText(StringEncrypt('LGNUK', Cypher)); //LOGIN
        UnKnown
        Communicate ('>Login Unknown ' + GetStringPart(strComText, ';',
          1));
        // Socket.Close; //Keep Connection open for adding user
      end
      else if (CheckLogin(GetStringPart(strComText, ';', 1),
        GetStringPart(strComText, ';', 2)) = 'wrong') then
      begin
        Socket.SendText(StringEncrypt('LGNWP', Cypher)); //LOGIN Wrong
        Password
        Communicate ('>Login Wrong ' + GetStringPart(strComText, ';',
          1));
        Socket.Close;
      end
      else if (CheckLogin(GetStringPart(strComText, ';', 1),
        GetStringPart(strComText, ';', 2)) = 'ok') then
      ..
```

```

end //end if LOGIN

else if strCommand = 'LOGOF' then //LOGOFF //strComText = User
..
end // end LOGOF

else if strCommand = 'LOGNW' then //LOGIN NEW new user is created
//strCommand = UserName;Password;FirstName;MiddleName;Lastname;
//          Street;Housenumber;HousenumberAddition;Address2;Zip;
//          City;Country;Accountnumber;Paymentmethod
begin
    tblUsers.Insert;
..
    tblUsers.Post;
    if not DirectoryExists(ApplicationPath + 'userfiles\') then
        CreateDir(ApplicationPath + 'userfiles');
        CreateDir(ApplicationPath + 'userfiles\' +
GetStringPart(strComText, ';',1));
..
    end;
end;

```

De nieuwe gebruiker wordt door de server toegevoegd, waarna een normale login wordt gestart. Om te zorgen dat de verzonden gegevens niet afgeluisterd kunnen worden, worden ze versleuteld. Hiervoor heb ik een aparte unit binnen Delphi gemaakt. Deze unit wordt door zowel de client- als de serverapplicatie aangeroepen. Door de encryptie unit apart te ontwikkelen, hoeft een wijziging maar één keer te worden aangebracht. Dit bespaart veel werk bij het onderhoud van de code. Omdat deze encryptie goed bij de socket werkte, kwam ik op het idee om ook de bestanden via de socket uit te wisselen. Dit zou me veel tijd besparen bij het uitzoeken van de werking van het Delphi FTP component. Het versturen van bestanden via de socket leek goed te werken. Bij het versturen van bestanden groter dan 2 MB kon de server applicatie de bestanden echter niet meer goed opslaan. Dit probleem trad al op als er één bestand werd verzonden. Wanneer de server in de toekomst vele bestanden tegelijk moet ontvangen, zou dit tot grote problemen kunnen leiden. Het gebruik van de Delphi socket viel daarom af voor gebruik in de applicatie.

Na overleg over deze gebeurtenis wilde de Manager Software ontwikkeling de server applicatie volledig laten vervallen. Het gebruik van een FTP-server voor de ontvangst van bestanden zou diverse voordelen bieden:

- De server is platform onafhankelijk
- De server is grondig getest en berekend op vele simultane connecties door clients
- De server hoeft niet apart te worden ontwikkeld.

Het grootste nadeel is echter:

- Er is geen direct inzicht in het verbruik per gebruiker

Omdat de voordelen sterk afwegen tegen de nadelen, hebben we deze aanpassing zo snel mogelijk doorgevoerd. Als gevolg van deze wijziging kon ik me tijdens de resterende tijd van de eerste timebox volledig richten op de overdracht van bestanden vanuit de client. Voor deze overdracht kon de encryptie unit opnieuw worden gebruikt in combinatie met de Delphi FTP component. Later zal naar een oplossing gezocht worden om het verbruik per gebruiker te meten.

Timebox 1 is hierdoor niet volledig behaald. Ik heb tijdens het schrijven van de communicatie veel kennis opgedaan. Het opzetten van de communicatie tussen de client en een server zal daardoor niet veel tijd meer kosten. Voor de beveiliging zijn al diverse functies geschreven. Deze kunnen in de vernieuwde communicatiewijze direct gebruikt worden. De should en could haves zijn nog niet verwerkt.

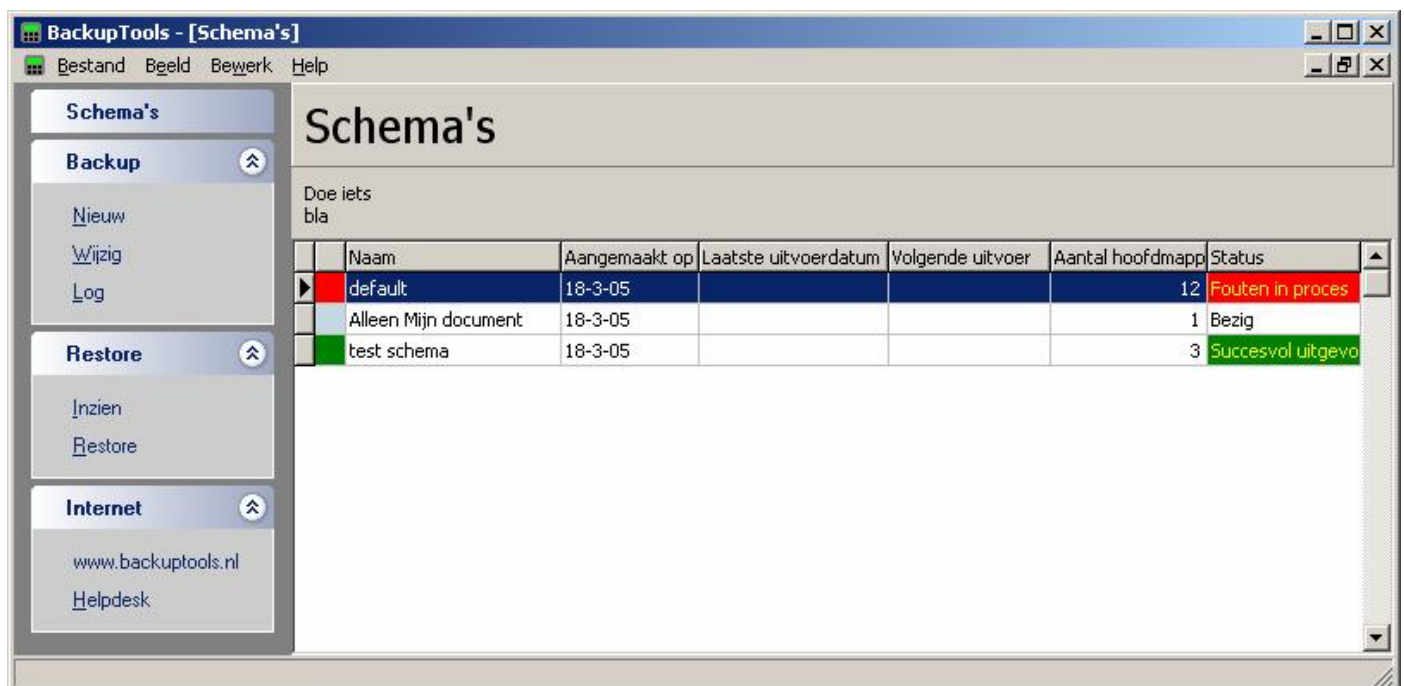
Must have	Server en Client Client	Bestanden ontvangen en verzenden Beveiliging	Gedeeltelijk Aanwezig
Should have	Server	Statistieken	Afwezig
Could have	Server	Automatische incasso	Afwezig

8.3 Timebox 2

De selectie van bestanden en het opslaan van deze selectie in meerdere schema's zijn de belangrijkste punten in timebox 2.

8.3.1 Opslag

Voor het opslaan van de schema's wordt gebruik gemaakt van het ERD uit het technisch ontwerp. De back-up wijze zal niet als tekst in de tabel aan de gebruiker getoond worden, maar wordt vastgelegd in de programmatuur. De status zou nog kunnen wijzigen en wordt daarom in een aparte tabel opgenomen. De tabellen heb ik gemaakt in Advantage Data Architect. Advantage heeft een grafische interface, waardoor het aanmaken van tabellen eenvoudig is. De Advantage database wordt in Delphi gekoppeld. Het openingsscherm ziet er als volgt uit:



8.3.2 Selectie

Voor het selecteren van de gewenste mappen heb ik in Delphi gezocht naar een component waarin dit mogelijk is. Enkele componenten kwamen of qua uiterlijk of

qua functionaliteit dicht in de buurt van de gewenste mappenstructuur, maar boden niet de mogelijkheid om de mappen afzonderlijk te selecteren. Het zoeken op Internet gaf weer diverse componenten, maar slechts één bevatte het uiterlijk en de functionaliteit zoals deze gewenst was. Deze component moest gekocht worden en werd geleverd zonder broncode. Het aankopen van componenten gebeurt wel bij Invers, maar alleen als er geen mogelijkheid is om de component zelf te ontwikkelen, of wanneer deze ontwikkelen teveel tijd kost.

Om deze reden ben ik gaan onderzoeken welke component globaal de gewenste functionaliteit bevatte, om dit verder zelf uit te breiden. Om niet de overige code van BackupTools te vervuilen, heb ik een aparte applicatie gemaakt met hierin een form voor de mapselectie.

De standaard treeview van Delphi heeft de juiste basis om verder op te bouwen. Hierin kan de mappenstructuur van Windows worden weergegeven. Om de interface van de mappenstructuur in Windows Verkenner na te bootsen, heb ik iconen uit de Windows Verkenner gekopieerd naar Delphi. Het opslaan en aanroepen van de iconen binnen Delphi heeft ten opzichte van het dynamisch aanroepen van de werkelijke iconen in Verkenner als voordeel dat de mappenstructuur snel is.

Windows maakt gebruik van de Desktop als hoofdnod (stam) in de mappenstructuur. Daaronder worden de map Mijn Documenten en de aanwezige opslagmedia getoond in childnodes (taken). Door uit te gaan van deze structuur, kon ik de basis leggen voor de nieuwe component. Ik heb deze component in een aparte applicatie geschreven om de code van de Back-up applicatie niet te vervuilen met onjuiste code. Het resultaat van de basis zag er als volgt uit:



Voor het vullen van de nu aanwezige nodes moet de inhoud van de schijven en mappen worden gelezen. Om de snelheid van de component te garanderen worden subnodes pas toegevoegd op het moment dat een node wordt uitgeklaapt.

Vervolgens moet de gebruiker een map kunnen selecteren. Een map kent drie selectiestatussen. Allereerst kan de map niet geselecteerd zijn. De map zal dan niet verwerkt worden. Daarnaast kan de map volledig geselecteerd zijn; de inhoud van de map en alle submappen worden geback-upped. Bij de laatste wordt alleen de inhoud van de map, exclusief submappen geselecteerd. Deze statussen brachten diverse problemen met zich mee. Wanneer een map volledig geselecteerd wordt, moeten ook alle getoonde onderliggende mappen volledig geselecteerd worden. Wanneer een map half geselecteerd wordt, moeten de onderliggende mappen niet geselecteerd worden. Wanneer alle onderliggende mappen geselecteerd worden, en de moedernode van deze mappen is half geselecteerd, dan moet deze volledig geselecteerd worden. Bij het deselecteren van een onderliggende map, moet ook de moedernode weer half geselecteerd worden. Om dit mogelijk te maken heb ik twee recursieve procedures geschreven. De eerste recursieve procedure zorgt dat kinderen op de juiste manier geselecteerd worden.

```
procedure TfrmDirectoryListing.ToggleCheckBoxChildren(aNode:
TTreeNode);
```



```

var
  tmpActiveNode : TTreeNode;
  i : integer;
begin
  if aNode.HasChildren then
    begin
      aNode.getFirstChild.StateIndex := aNode.StateIndex;
      ToggleCheckBoxChildren(aNode.getFirstChild);
      if aNode.Count > 1 then
        begin
          tmpActiveNode := aNode.getFirstChild;
          for i := 2 to (aNode.Count) do
            begin
              tmpActiveNode := tmpActiveNode.getNextChild(tmpActiveNode);
              tmpActiveNode.StateIndex := aNode.GetFirstChild.StateIndex;
              ToggleCheckBoxChildren(tmpActiveNode);
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

Het omzetten van de selectiestatus (StateIndex) van de kinderen wordt aangeroepen met de actieve node. Vervolgens wordt de status van alle kinderen aangepast. Vervolgens wordt de functie binnen zichzelf aangeroepen, om ook de status van de kinderen van de kinderen (enz.) aan te passen.

Ook voor het controleren van de ouders (en grootouders enz.) van de actieve node wordt een recursieve functie gebruikt. Van de ouder wordt de status van alle kinderen gecontroleerd voor het aanpassen van de status van de ouder:

```

procedure TfrmDirectoryListing.ToggleCheckboxParents(aNode:
TTreeNode);
var
  tmpActiveNode : TTreeNode;
  i, sum: integer;
begin
  if (aNode.level > 1) or (aNode.text = Translate('Mijn Documenten'))
  then
    begin
      sum := 0;
      if (aNode.HasChildren) and (aNode.StateIndex <> cUncheck) then
        begin
          if aNode.getFirstChild.StateIndex = cChecked then sum := sum+2;
          if (aNode.getFirstChild.StateIndex = cHalfCheckUp) or
            (aNode.getFirstChild.StateIndex = cHalfCheckUp) then
            inc(sum);
          if aNode.Count > 1 then
            begin
              tmpActiveNode := aNode.getFirstChild;
              for i := 2 to (aNode.Count) do
                begin
                  tmpActiveNode :=
                    tmpActiveNode.getNextChild(tmpActiveNode);
                  if tmpActiveNode.StateIndex = cChecked then sum := sum+2;
                  if (tmpActiveNode.StateIndex = cHalfCheckUp) or
                    (tmpActiveNode.StateIndex = cHalfCheckDown) then
                    inc(sum);
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

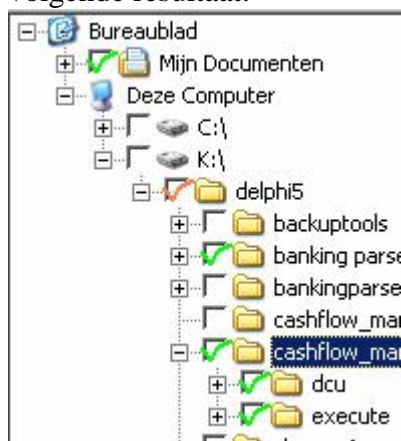
```

```

    if (Sum = 0) and (aNode.StateIndex = cChecked) then
        aNode.StateIndex := cHalfCheckDown
    else if sum >= (2 * aNode.Count) then
        begin
            if (aNode.Count = 1) then
                aNode.StateIndex := aNode.GetFirstChild.StateIndex
            else
                aNode.StateIndex := cChecked;
            end
        else aNode.StateIndex := cHalfCheckDown;
        ToggleCheckBoxParents(aNode.Parent);
    end;
end;
end;
end;

```

Het gedrag van de boomstructuur is zoals deze gewenst is. Een selectie geeft het volgende resultaat:



De gegevens hierin moet ook nog aan andere componenten bekend gemaakt worden. De meest eenvoudige wijze is het gebruik van een lijst (TList). Vanwege de beperkte informatie, namelijk de naam van het pad en de selectiestatus, heb ik gekozen voor een lijst met strings waarin de status en de padnaam van een node worden opgeslagen per string. Het gebruik van een string voor eenvoudige communicatie tussen verschillende programmaonderdelen is mij tijdens het opstellen van het client-server protocol goed bevallen. Om tekstdelen uit een string te extraheren voor het client-server protocol kon ik na een kleine aanpassing ook goed gebruiken voor het extraheren van een padnaam uit een actieve node. Om de padnaam te krijgen, moet de node worden gelezen. Vervolgens worden de namen van de ouders hier voor geplaatst, waardoor de volledige padnaam ontstaat.

```

function TfrmDirectoryListing.SetNodePath(aNode: TTreeNode): String;
begin
    if aNode.Level > 1 then
        begin
            If aNode.text[2] = ':' then
                result := aNode.Text
            else
                result := SetNodePath(aNode.Parent) + aNode.Text + '\';
            end
        else if aNode.Text = Translate('Mijn Documenten') then
            result := GetSpecialFolder(CSIDL_PERSONAL)+ '\';
        end;
end;

```

Wanneer een node wordt bereikt met op de tweede plaats een dubbele punt, betekent dit dat een schijfletter is geselecteerd en dat de volledige padnaam bekend is. Indien de node 'Mijn Documenten', dan moet de verwijzing naar de opslaglocatie hiervan uit Windows worden uitgelezen.

Deze informatie wordt samen met de status in de lijst toegevoegd. Vervolgens wordt gecontroleerd of er geen hoger pad aanwezig met een volledige selectie. Indien deze aanwezig is, moet de nieuwe toevoeging worden verwijderd.

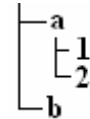
```
function TfrmDirectoryListing.SelectedList (aNode: TTreeNode; var
aList: TStringList) : String;
var
  nodestate, i, index : integer;
  nodetext : string;
begin
  aList.Sort;
  NodeText := aNode.text;
  NodeState := aNode.StateIndex;
  if aNode.level > 0 then
  begin
    if aNode.Parent.StateIndex = cChecked then
    begin
      if (aList.Find('A' + SelectedPath(aNode), index)) then
        aList.Delete(index);
      if (aList.Find('H' + SelectedPath(aNode), index)) then
        aList.Delete(index);
      aList.Sort;
    end

    else if (aNode.StateIndex = cChecked) and (aNode.Text <> '.') then
    begin
      if not aList.Find('A'+ SelectedPath(aNode), index) then
      begin
        if (aList.Find('H' + SelectedPath(aNode), index)) then
        begin
          aList.Delete(index);
        end;
        if aNode.Parent.StateIndex <> cChecked then
          aList.Add('A'+ SelectedPath(aNode));
        end;
        aList.Sort;
      end
      else if ((aNode.StateIndex = cHalfCheckUp) or (aNode.StateIndex =
        cHalfCheckDown) ) and (aNode.text <> '.') then
      begin
        ..
      end
      else if ((aNode.StateIndex = cUnCheck) {or (aNode.StateIndex =
        -1)}) and (aNode.text <> '.') then
      begin
        ..
      end;
    end;
    NodeText := aNode.text;

    for i := 0 to (aNode.Count -1) do
      SelectedList(aNode.Item[i], aList);
    end;
  end;
```

Voor het doorlopen van de lijst heb ik weer gebruik gemaakt van recursiviteit. Dit geeft een lijst waarin alleen de benodigde paden aanwezig zijn.

Het opbouwen van de boomstructuur en de selectielijst heeft me veel tijd gekost. Het gedrag van de nodes was soms volledig anders dan verwacht. Een oorzaak hiervan was dat de naamgeving van de bij de nodes behorende procedures soms verwarrend was. De functie `SelectFirstChild` geeft het eerste kind(1) van een node(a). Mijn verwachting was dat `SelectNextChild` het volgende kind(2) zou selecteren. Hier werd echter de volgende node van de ouder geselecteerd (b). Door verkeerd gebruik te maken van de functie `SelectNextChild` was het gedrag van de component niet als verwacht. Het vinden van deze fout was erg lastig, omdat de code er goed uit leek te zien.



De code is vervolgens overgezet naar de BankingTools applicatie. Wanneer een schema wordt opgeslagen, worden de geselecteerde mappen opgeslagen in de database.

De resultaten van timebox 2 zijn beter dan van timebox 1.

Must have	Client Client	Selectie in mappenstructuur Meerdere schema's	Aanwezig Aanwezig
Should have	Client Client	Overzicht geselecteerde mappen Overzicht bestanden op server	Aanwezig Afwezig
Could have	Client Client	Bestanden selectie Filter	Afwezig Afwezig

Het overzicht van bestanden op de server kon niet verwezenlijkt worden door de afwezigheid van een juiste communicatie. Daarnaast kostte het opzetten van het mappenselectiescherm veel tijd. Dit onderdeel is echter de basis van de mogelijkheid tot verdere ontwikkeling. Door het gedrag van deze component goed te controleren kan voorkomen worden dat de applicatie bij verdere aanpassing onverwacht gedrag vertoont.

8.4 Timebox 3

Het uitvoeren van een back-up op de achtergrond op een ingestelde tijd heeft de hoogste prioriteit in timebox 3. Om dit in te kunnen stellen, wordt de gebruiker enkele keuzes geboden. Er moet altijd een tijdstip worden gekozen. Als tijdstip kan een werkelijke tijd worden gekozen, maar ook het opstarten en afsluiten van Windows kan als tijdstip worden ingesteld.

Selecteer het tijdstip waarop u deze back-up wil uitvoeren.

☐ Tijdstip 18:00:00

☐ Bij opstarten Windows

☐ Eén keer per dag

☒ Bij afsluiten Windows

☒ Mogelijkheid tot onderbreken

☐ Maandag

☐ Dinsdag

☐ Woensdag

☐ Donderdag

☐ Vrijdag

☐ Zaterdag

☐ Zondag

Daarnaast kan de back-up op vaste dagen per week of per maand worden uitgevoerd. De laatste optie biedt de gebruiker de optie om de back-up eenmalig uit te voeren en het schema, indien gewenst, op te slaan voor toekomstig gebruik. Deze gegevens worden in de schematabel in de database opgeslagen.

Bij het aanmaken van het tijdstip keuzeschermb heb ik gebruik direct extra tabbladen toegevoegd voor het kiezen van het back-up type en de back-up doellocatie. Voorlopig worden deze schermen nog verborgen voor de gebruiker. Wanneer deze mogelijkheden worden toegevoegd, is de basis al gelegd.

Vervolgens moet de applicatie op de achtergrond kunnen draaien. Naast het minimaliseren van de applicatie, waardoor deze niet meer op het scherm getoond wordt, moet de vermelding in de taakbalk worden verwijderd. Hierover wordt in de Delphi handleiding geen vermelding gemaakt. Na enig zoeken op Internet kwam ik een pagina tegen met uitleg over het verbergen van een applicatie voor de gebruiker. Op de "The Big Brother" Delphi pagina (<http://delphi.about.com/od/windowsshellapi/1/aa022701a.htm>) vond ik erg nuttige uitleg hierover. Door het verbergen uit te voeren zodra de applicatie minimaliseert, wordt elke handeling in Windows of de applicatie die hiertoe leidt afgevangen:

```
procedure TfrmMain.ApplicationEvents1Minimize(Sender: TObject);
begin
    ShowWindow(Application.Handle, SW_HIDE);
    frmMain.Hide;
end;
```

Na het minimaliseren wordt alleen nog een icoon van een kluis in de system tray getoond.



Wanneer de applicatie wordt opgestart, wordt een timer gestart. Deze controleert elke minuut of er een schema is dat moet worden uitgevoerd. Daarnaast wordt ervoor gezorgd dat het afsluiten van Windows wordt afgevangen door de applicatie. Zodra Windows afsluit wordt er gecontroleerd of er schema's zijn die uitgevoerd moeten worden. Ook bij het opstarten van de applicatie wordt deze controle uitgevoerd.

Het uitvoeren van een back-up kan alleen gebeuren wanneer de applicatie actief is. Daarom is het verstandig op de applicatie samen met Windows op te laten starten. Dit kan op twee manieren worden verwezenlijkt. Het start menu van Windows kent een

submenu Opstarten. In dit submenu kan een snelkoppeling worden opgenomen naar de applicatie. Dit is echter geen nette manier. Het is netter om het opstarten via de Windows registry te laten verlopen. In HKEY_LOCAL_MACHINE - Software - Microsoft - Windows - CurrentVersion - Run kan een verwijzing naar de applicatie worden opgenomen. Hierdoor start de applicatie samen met Windows op, maar is dit niet voor de gebruiker zichtbaar. Het toevoegen en verwijderen van deze verwijzing heb ik in het instellingen scherm opgenomen:



Vervolgens heb ik de het kiezen van het back-up type mogelijk gemaakt. Bij het opslaan van een schema wordt nu alle benodigde informatie correct opgeslagen:



Na het verwerken van de must haves en enkele should haves ben ik verder gegaan met de belangrijkste must have uit de eerste timebox.

Must have	Client Client	Back-up moment Draaien op achtergrond	Aanwezig Aanwezig
Should have	Client Client Client	Opstarten bij Windows start Back-up type Back-up locatie	Aanwezig Aanwezig Voorbereid
Could have	Client Client Client	Interne toegang tot internetsite/ helpdesk Herstellocatie Historie	Afwezig Afwezig Afwezig

Momenteel ligt de nadruk op de verplaatsing van bestanden van de client pc naar de server.

9 Evaluatie

Het afstuderen bij Invers B.V. is voor mij erg leerzaam geweest. Ik heb kennis gemaakt met een nieuwe ontwikkelmethode en bekende technieken in een echte situatie kunnen toepassen. Ik zal beschrijven hoe ik het procesverloop heb ervaren en hoe ik de opgeleverde producten beoordeel. Daarnaast geef ik aan wat ik in de toekomst zal doen om gemaakte fouten te voorkomen.

9.1 Procesevaluatie

Om de ontwikkeling in goede banen te leiden heb ik gekozen voor DSDM als ontwikkelmethode. De methode van Invers is gebaseerd op SDM. Het grootste gedeelte van mijn studie heb ik producten ontwikkeld volgens SDM. Hierdoor beheers ik deze methode goed. Het kiezen van SDM lag daarom voor de hand. Tijdens mijn stage heb ik ook SDM gebruikt in een object geörienteerde omgeving en heb toen ervaren dat SDM niet voldoende mogelijkheden biedt om een object georienteerde implementatie te begeleiden. De ontwikkelmethode DSDM toont gelijkenis met SDM, maar is flexibler van opzet, waardoor het beter aansluit bij ontwikkeling in een object geörienteerde omgeving. Om kennis op te doen heb ik het handboek voor DSDM besteld. Als spoedig na inleveren van mijn bleek dit boek niet meer geleverd te worden. Uiteindelijk kwam ik op het boek 'DSDM: de methode in de praktijk' uit. Hierin wordt beschreven hoe de richtlijnen van DSDM moeten worden toegepast. De richtlijnen worden echter maar beperkt genoemd. Dit maakte het opstellen van het Plan van Aanpak lastig, maar dit heb ik naar mijn mening goed gedaan. Door gebruik te maken van het voor mij bewezen voorbeeld van ZBC kon ik in korte tijd een duidelijk plan van aanpak opstellen waar het ontwikkeltraject goed in wordt vastgelegd.

Tijdens de eerste vergadering, waarin we de user requirements zouden bespreken, werd ik behoorlijk overvallen door alle ideeën die er bij Invers waren. Ik zou een rapport ontvangen met de user requirements, maar ontving een hoop ideeën over het toekomstige pakket, waarvan geen enkel idee op papier was vastgelegd. Door een concurrentie onderzoek te doen kon ik ontdekken wat er verder op de markt beschikbaar was, om tot een goed voorstel te komen. Dit voorstel heb ik uitgewerkt in de bedrijfsanalyse. In dit voorstel ben ik teveel uitgegaan van de ideeën die tijdens de vergadering naar voren kwamen, en ben ik sterk van de oorspronkelijk beschrijving in de opdrachtomschrijving afgeweken. Bij Invers werd gelukkig ingezien dat ik misschien te hoge eisen heb gesteld aan het eindpakket en me gewezen op de inhoud van de opdrachtomschrijving. In de toekomst zal ik tijdens besprekingen van de systeemeisen de opdrachtgever wijzen op de randvoorwaarden in de opdrachtomschrijving.

Doordat ik werkte op een pc die al voor andere doeleinden was gebruikt en door de installatie van verschillende back-up pakketten was mijn pc zo vervuild geraakt dat de stabiliteit van het systeem sterk afnam. Om de stabiliteit terug te krijgen was een herinstallatie van het besturingssysteem en de benodigde software noodzakelijk. Door de vele externe componenten voor Delphi kostte dit zoveel tijd, dat achteraf in de planning opgenomen moest worden. Dit is een risico factor waar ik nooit eerder mee te maken heb gehad, maar waar ik in de toekomst rekening mee zal houden door gebruik te maken van een apart testsysteem of de standaard installatie van een systeem op te slaan.

Het samenstellen van het functioneel model was tweestrijdig. Door het concurrentie onderzoek en het onderzoeken van andere Invers software had ik een duidelijk beeld

gekregen van het gewenste uiterlijk. De schermvoorbeelden kon ik goed baseren op reeds bestaande Invers software.

Tijdens het uitwerken van de overige onderdelen in het functioneel model merkte ik echter dat ik nog niet veel object geïntegreerd ontwikkeld en geprogrammeerd heb. Voor ontwikkeling in een object geïntegreerde omgeving moet een volledig andere denkwijze worden aangehouden dan bij gestructureerd programmeren. De beperkte kennis van object geïntegreerd programmeren zorgde dat ik de UML techniek lastig vond toe te passen.

Door gelijktijdig te programmeren en het technisch ontwerp samen te stellen, werd het bestaansrecht van het technisch ontwerp mij onduidelijk. Ik ben gewend dat het technisch ontwerp een goede blauwdruk geeft voor de code. Door direct ideeën te coderen bespaarde ik veel tijd. Het technisch ontwerp kreeg voor mij hierdoor de functie van overzichtelijk naslagwerk. Nadat ik een idee had uitgedacht verwerkte ik dit in code. Indien ik de informatie op meerdere locatie in het pakket nodig dacht te hebben naam ik dit over in het technisch ontwerp. Doordat alleen ik aan dit project werkte, leverde deze werkwijze geen problemen op. Wanneer er in teamverband wordt gewerkt, verwacht ik dat deze informatie juist van tevoren moet worden gedocumenteerd om eenduidige programmacode op te leveren.

Het gebruik van timeboxen tijdens het programmeren heb ik als erg prettig ervaren. Ik merkte dat sommige onderdelen meer tijd kostten dan ik van tevoren had verwacht. Door de timeboxen aan te houden, werd voorkomen dat ik me volledig op één onderdeel van de applicatie aan het richten was. Hierdoor kon ik diverse onderdelen ontwikkelen. Dit zorgde echter ook dat de programmaonderdelen niet allemaal afgerond zijn. Het doorgaan met een ander onderdeel zorgde er wel voor dat ik nieuwe technieken ontdekte, die ik in een voorgaande timebox kon toepassen. Wanneer ik gekozen had om een probleem altijd op te lossen, had ik deze nieuwe techniek mogelijk niet ontdekt en had ik het probleem niet of niet efficiënt op kunnen lossen.

Het gebruik van DSDM was een nieuwe ervaring voor mij. Doordat ik direct samenwerkte met teamleden met DSDM kennis, vond ik het volgen van de methode soms erg lastig. De algemene lijn van DSDM heb ik denk ik goed kunnen aanhouden. Helaas heeft dit niet kunnen voorkomen dat ik enige vertraging heb opgelopen. De beperkte kennis van object geïntegreerd ontwikkelen en programmeren en de vele nieuwe Delphi componenten zijn hier de belangrijkste oorzaak van. Aan het einde van de tweede timebox begon ik echt te snappen hoe object geïntegreerd programmeren werkt en kon ik ook op deze manier denken. Voor toekomstige projecten kan ik deze opgedane kennis direct toepassen tijdens het volledige ontwikkeltraject.

9.2 Productevaluatie

De rapporten voor het ontwikkeltraject voldoen goed aan de eisen zoals deze in de opdrachtomschrijving zijn opgesteld. De inhoud van de bedrijfsanalyse is zeer uitgebreid. Hierdoor werd de noodzaak van uitgebreide schema's in het functioneel model en het technisch ontwerp beperkt. De rapporten geven voldoende algemene beschrijving om een goede applicatie te ontwikkelen. De details in de applicatie kunnen zo tijdens het programmeren worden ingevuld. Dit voorkomt inconsistentie in de documentatie. Tijdens de interactie van gebruikers met de prototypen komen steeds kleine aanpassingen aan het licht, die vooraf zeer moeilijk in kaart zijn te brengen. Het gebruik van de rapporten als richtlijn en het gebruik van prototypen als communicatiemiddel naar de gebruiker zorgt voor een applicatie die goed op de

gebruiker is afgestemd. De applicatie is nog niet volledig afgerond. De huidige versie is in een vergevorderd stadium waardoor de afronding snel plaats kan vinden.

9.3 Slotwoord

Het afstudeertraject is erg leerzaam geweest. Het opstellen van rapporten heb ik na vier jaar studie goed onder de knie. Door de vele praktijkopdrachten op de Haagse Hogeschool heb ik een goed inzicht gekregen in de benodigde informatie tijdens het implementatie traject. Helaas heb ik beperkt les gehad in object geïntereerd programmeren, waardoor ik de bijbehorende denkwijze pas laat in het afstudeertraject onder de knie kreeg. Dit heeft ervoor gezorgd dat opgestelde schema's en de ontwikkelde code niet volledig volgens een object geïntereerde manier zijn opgesteld. Het werken in Delphi heeft er voor gezorgd dat ik deze werkwijze ben gaan begrijpen. Hierdoor ben ik voor toekomstige opdrachten voorbereid en kan ik deze denkwijze direct toepassen. Ik verwacht de applicatie in korte tijd af te kunnen ronden en geschikt te maken voor de markt.

Literatuurlijst

Boeken

Stapleton, J., DSDM: De methode in de praktijk
2002 (1^e druk 1999), Academic Service, Schoonhoven

Yourdon, E., Gestructureerde Analyse
1999 (1^e druk 1991), Academic Service, Schoonhoven

SDM System Development Methodology
Nederlandstalige samenvatting
1992 (1^e druk 1988), Cap Gemini Publishing BV,
Rijswijk

Warmer, J., Praktische UML
2001, Addison Wesley Longman Nederlands BV,
Amsterdam

Stefanski, M., Basiscursus Delphi 3
1998, Academic Service, Schoonhoven

Cantù, M., Mastering Delphi 5
1999, SYBEX Inc., Alameda (VS)

Internet

ZBC Consultants
www.zbc.nu/main.asp?ChapterID=461

Back-up algemeen
<http://www.backupenrecover.com/>

Back-up pakket
<http://www.backup.nl/>
<http://www.backupfactory.nl>
<http://www.backuphelp.com>
<http://easybackup.nl>
<http://remote-backup.com>
<http://www.streamload.com>
<http://www.backupsmart.nl>
<http://www.genie-soft.com>

Encryptie en compressie
<http://www.schneier.com/>
<http://computer.howstuffworks.com/>
<http://www.nist.gov/>
<http://www.firstbackup.com>

Delphi
<http://delphi.about.com>
<http://www.google.com>

Divers
<http://forum.computeridee.nl/>
<http://computerbeveiliging.pagina.nl/>
<http://gathering.tweakers.net/forum/>
Bankingtools helpdesk

Invers interne documentatie

Code Conventies
Query conventies

Bijlage I: Opdrachtomschrijving

De bijlagen zijn in de elektronische versie niet aanwezig.

Bijlage II: Plan van Aanpak

De bijlagen zijn in de elektronische versie niet aanwezig.

Bijlage III: Bedrijfsanalyse

De bijlagen zijn in de elektronische versie niet aanwezig.

Bijlage IV: Functioneel model

De bijlagen zijn in de elektronische versie niet aanwezig.

Bijlage V: Technisch ontwerp

De bijlagen zijn in de elektronische versie niet aanwezig.