

Afstudeerverslag

Dataselectietool E2Ma

Algemene gegevens

Student	Ronald Paul
Opleidingsinstituut	Haagse Hogeschool Sector Informatica Johanna Westerdijkplein 75 2521 EN Den Haag
Opleiding	Informatica en Informatiekunde (I&I), Vormgeving en Ontwerp van Interactie (VIA)
Afstudeerbedrijf	E2Ma Consumer Profiling Paardenhoeve 78 3992 PK Houten
Bedrijfsbegeleider	dhr. B. Roos
Examinatoren	ing W.G. Renden ir T.H.M. Spaan
Periode	7 februari 2005 t/m 10 juni 2005

Referaat

Paul, R.C., *Afstudeerverslag. Dataselectietool E2Ma*. Houten: E2Ma Consumer Profiling, 2005.

Dit afstudeerverslag is geschreven in het kader van de stage bij E2Ma Consumer Profiling te Houten. De opdracht is uitgevoerd voor het afstuderen aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

E2Ma verzamelt en verhuurt consumentenprofielen die worden ingezet voor direct-marketing-doeleinden. De gegevens in deze profielen worden verzameld door het houden van thematische enquêtes die iedereen via internet kan invullen. Nadat deze gegevens in databases terecht komen, moeten ze er aan de andere kant ook weer uitgehaald kunnen worden. De opdracht is dan ook het ontwerpen, beschrijven en ontwikkelen van een systeem waarmee selecties op die databases kunnen worden verricht.

Descriptoren:

- GUIDE
- IAD
- MySQL
- PHP
- SUMI
- UML

Voorwoord

Afstuderen is de manier om voor jezelf na te gaan of de opleiding in de afgelopen jaren voldoende zoden aan de dijk heeft gezet. Maar niet alleen dat, want het is formeel ook de periode waarin het opleidingsinstituut hetzelfde doet. Op mij persoonlijk heeft de afgelopen periode veel extra kennis en ervaring in het laadje gebracht. Het was een leerzame en leuke tijd in een gezonde en productieve werkomgeving.

Dit afstudeerverslag heb ik primair geschreven voor de examencommissie. Hoofdstukken 2 en 3 gaan in op de opdracht en de aanpak. Hoofdstukken 4 t/m 11 zijn synchroon gehouden met de doorlopen fasen en de proces en productevaluaties zijn tot slot te vinden in hoofdstukken 12 en 13.

Ik wil bij deze graag van de gelegenheid gebruik maken om een aantal mensen te bedanken. Allereerst mijn begeleider binnen E2Ma, Bob Roos. Vanaf de sollicitatieprocedure tot en met de laatste loodjes heeft hij mij erg goed begeleid in het proces gezien vanuit het bedrijf. Ook Jacco van Hooren heeft mij goed geholpen met wat zaken op het technische vlak. Daarnaast bedank ik mijn examinatoren Wijnand Renden en Tom Spaan voor de begeleiding vanuit school en voor de goede adviezen omtrent het schrijven van dit verslag.

Ronald Paul

Houten, 9 juni 2003

Inhoudsopgave

Algemene gegevens	3
Referaat	5
Voorwoord	7
Inhoudsopgave	9
1 Inleiding	13
2 Opdracht	15
3 Aanpak	21
3.1 Ontwikkelmethodiek	21
3.2 Projectbesturing	22
3.3 Projectscenario	23
3.4 Iteratiestrategie	24
3.5 Algemeen Plan van Aanpak	24
4 Definitiestudie	25
4.1 Plan van Aanpak	25
4.2 Systeemeisen	25
4.3 Taakdiagrammen	26
4.4 Taakscenario's	27
4.5 Use-case diagram	27
4.6 Klassendiagram	28
4.7 Selectieproces	31
4.8 Stijlgids	31
4.9 Pilotplan	32
4.10 Overig	33
5 Pilot #1, GUI-standaard	35
5.1 Plan van Aanpak	35
5.2 Globaal-functionele structuur	35
5.3 Globaal-technische structuur	35
5.4 Globaal-organisatorische inrichting	38
5.5 GUI-ontwerp	39
5.6 Testplan	39
5.7 Ontwerp & bouw	40
5.8 Beoordelen en testen pilot	46

6	Pilot #2, Variabelebeheer	47
6.1	Plan van Aanpak	47
6.2	SUMI-test	47
6.3	Globaal-functionele structuur	48
6.4	Globaal-technische structuur	48
6.5	GUI-ontwerp	50
6.6	Testplan	51
6.7	Ontwerp & bouw	52
6.8	Beoordelen en testen pilot	55
7	Pilot #3, Selectiebeheer	57
7.1	Plan van Aanpak	57
7.2	SUMI-test	57
7.3	Globaal-functionele structuur	57
7.4	Globaal-technische structuur	58
7.5	GUI-ontwerp	60
7.6	Testplan	62
7.7	Ontwerp en bouw	62
7.8	Beoordelen en testen pilot	64
8	Pilot #4, Selectieactivatie	65
8.1	Plan van Aanpak	65
8.2	Globaal-functionele structuur	65
8.3	Globaal-technische structuur	65
8.4	Testplan	67
8.5	Ontwerp en bouw	67
8.6	Beoordelen en testen pilot	72
9	Pilot #5, Resultaatbeheer	73
9.1	Plan van Aanpak	73
9.2	Globaal-functionele structuur	73
9.3	GUI-ontwerp	73
9.4	Testplan	74
9.5	Ontwerp en bouw	75
9.6	Beoordelen en testen pilot	75
10	Pilot #6, Gebruikersbeheer	77
11	Invoering	79
11.1	Plan van Aanpak	79
11.2	Uitvoeren SUMI-test	79

11.3	Pilots invoeren	80
11.4	Pilotacceptatie uitvoeren	80
11.5	Pilots overdragen	80
11.6	Feedback verzamelen	80
11.7	Pilots ondersteunen	80
12	Procesevaluatie	81
13	Productevaluatie	83
Bijlage 1: Opdrachtschrijving		85
1	Inleiding	85
2	Opdrachtschrijving	85
Bijlage 2: Algemeen Plan van Aanpak		89
1	Inleiding	89
2	Achtergrond	90
3	Probleemstelling	90
4	Doelstelling	90
5	Uitgangssituatie	91
6	Werkzaamheden	92
7	Planning	94
8	Producten	95
Bijlage 3: Externe bijlagen		97
Literatuurlijst		99

1 Inleiding

Dit is het afstudeerverslag. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie). De organisatie waar de opdracht bij is uitgevoerd heet E2Ma en verzamelt en verhuurt consumentenprofielen die worden ingezet voor direct-marketing-doeleinden. De gegevens in deze profielen worden verzameld door het houden van thematische enquêtes die iedereen via internet kan invullen. Nadat deze gegevens in databases terecht komen, moeten ze er aan de andere kant ook weer uitgehaald kunnen worden. De opdracht is dan ook het ontwerpen, beschrijven en ontwikkelen van een systeem waarmee selecties op die databases kunnen worden verricht.

Het verslag geeft informatie over de opdracht, het gevolgde proces, de hierbij gemaakte keuzes en redenen en de ontstane producten. Alles wordt hierbij bekeken vanuit het gevolgde proces, wat het verslag voor het overgrote deel ook chronologisch maakt, maar het product wordt er ook veelvuldig bijgehaald. Een digitale versie van die document is te vinden onder <http://rcpaul.nl/hhs/afst.pdf>.

De opbouw van dit verslag is als volgt. Hoofdstuk 2 heeft een omschrijving van de organisatie, de opdracht, de huidige situatie en de alternatieven. Hoofdstuk 3 gaat over de keuze voor een ontwikkelmethodiek, de projectbesturing, het projectscenario en het algemeen Plan van Aanpak. Hoofdstukken 4 t/m 11 vormt de kern. Hierin wordt achtereenvolgens ingegaan op de werkzaamheden in de fase Definitiestudie, de zes pilots en de fase Invoering.

2 Opdracht

In dit hoofdstuk staan de organisatie, de opdracht, de huidige situatie en de alternatieven besproken. Zie bijlage 1 voor de volledige omschrijving van organisatie en opdracht. Bij de alternatieven staan de redenen voor de organisatie om te kiezen voor de doelstelling zoals die aan deze opdracht ten grondslag ligt.

Organisatie

De organisatie E2Ma (spreek uit als een afkorting) is gevestigd in Houten. Ze verzamelt en verhuurt consumentenprofielen die worden ingezet voor direct-marketing-doeleinden. Deze consumentenprofielen bestaan uit: persoons-, sociodemografische- en gedragsinformatie. De gegevens in deze profielen worden verzameld door het houden van enquêtes waar iedereen via internet aan kan deelnemen. Dit zijn zogenaamde *sweepsteaks* waarbij mensen worden aangetrokken mee te doen onder verloting van prijzen. Nadat de gegevens zijn verzameld, worden deze via verschillende kanalen op de markt gebracht.

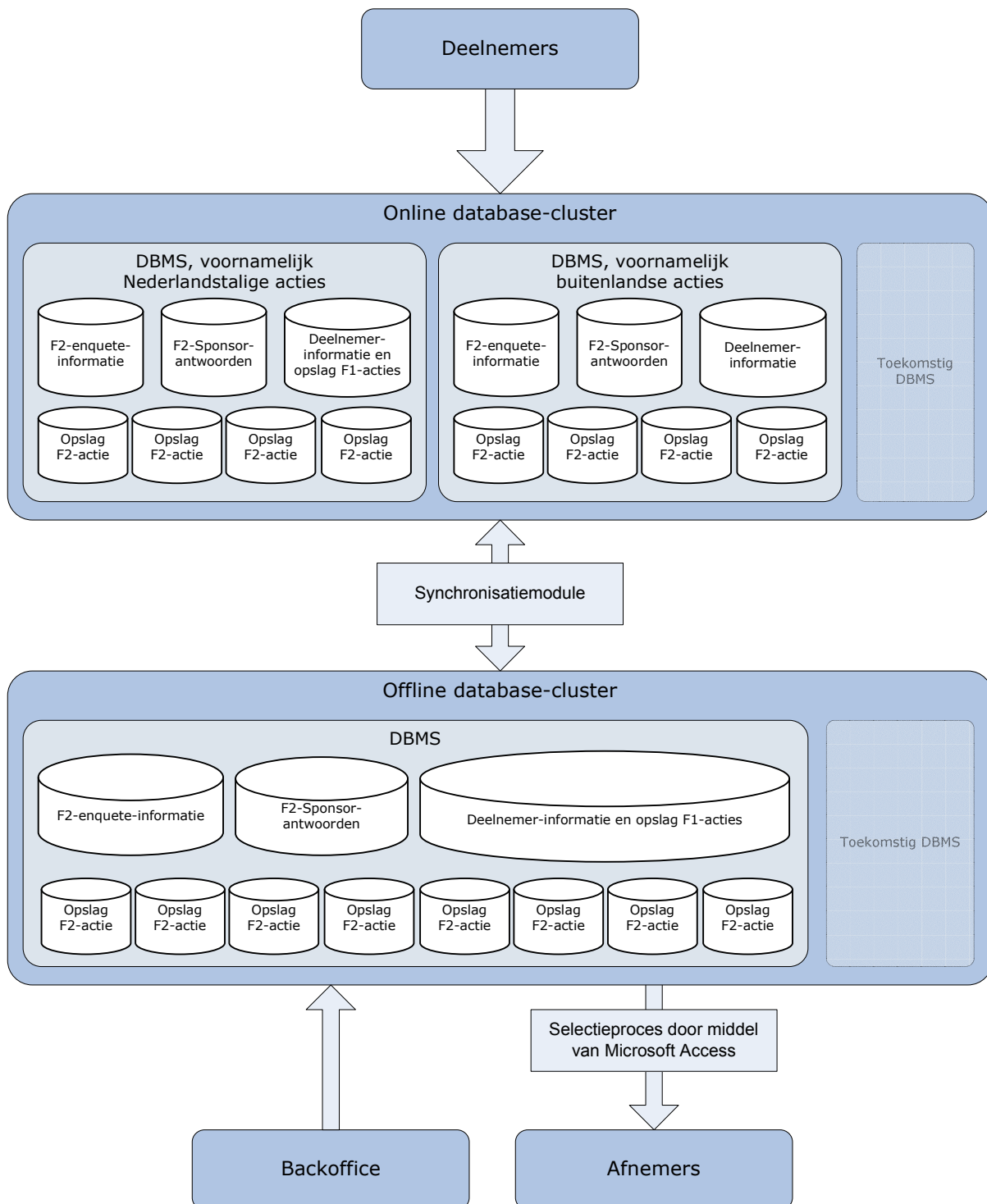
Opdracht

De hoeveelheid data die E2Ma over consumenten verzamelt neemt alsmaar toe. Deze informatie wordt, afhankelijk van het doel, in een aantal verschillende MySQL-databases opgeslagen. Ook neemt de vraag uit de markt naar kwalitatief hoogwaardige data steeds verder toe. Deze twee feiten leiden ertoe dat selecties van data uit deze consumentenprofielen (aggregaties, tellingen, samenvattingen, enzovoorts) steeds complexer worden. Om deze complexiteit in te dammen en te voorkomen dat het ten koste gaat van de efficiëntie, groeit de behoefte aan een gebruikersvriendelijke en complete interface voor het uitvoeren van deze selecties.

Het doel van de afstudeeropdracht is dan ook het ontwerpen, beschrijven en ontwikkelen van een systeem waarmee operationele medewerkers selecties op de databases kunnen verrichten. Dit dient snel, makkelijk en betrouwbaar te kunnen.

Huidige situatie

E2Ma is al een paar jaar operationeel. In die tijd zijn verschillende systemen gebouwd, verbeterd en geïntegreerd tot een nu nog steeds evoluerend en werkend systeem. Als inleiding op de deel- en randsystemen volgt in deze paragraaf een samenvatting van deze



Figuur 1 Samenhang tussen de verschillende DBMS'en, databases, randsystemen en raakvlakken met de deelnemers en afnemers.

structuur. Deze context is nodig voor het makkelijk kunnen begrijpen van de later te geven voorbeelden.

De basis ligt in de enquêtes. Intern worden deze acties genoemd. Alle acties die op dit moment lopen hebben elk een thema. Zo zijn er acties

genaamd Fit en fris en Autovoorkeur. Elk van deze enquêtes hebben vragen in drie verschillende categorieën, namelijk:

- Deelnemersspecifiek: vragen naar geboortedatum, woonplaats, telefoonnummer, opleidingen, gezinssituaties, enzovoorts. Deze vragen overlappen voor het grootste gedeelte tussen alle acties.
- Actiespecifiek: vragen naar onderwerpen waar de actie over gaat. Bij Autovoorkeur is dit bijvoorbeeld de vraag of de deelnemer de komende zes maanden van plan is om een nieuwe auto te kopen.
- Sponsor: vragen die voor externe partijen zijn opgenomen. Aan de hand van een eerder gegeven antwoord, wordt een vraag uit deze categorie wel of niet gesteld.

Zie Figuur 1 voor een schematische weergave van de huidige situatie. De afgelopen jaren is er een belangrijke technologische vooruitgang geboekt, namelijk de ontwikkeling van de F2-acties. De belangrijkste verschillen:

- Waar F1-acties nog bestond uit een plat stuk HTML, bij F2 kunnen de enquêtes via een backoffice worden samengesteld en aangepast. Deze gegevens komen terecht in de `F2-enquete-informatie-tabellen`.
- Data uit de F2-acties worden elk in afzonderlijke databases opgeslagen (met elk antwoord in een aparte tabel) terwijl alle antwoorden bij F1 nog terecht kwam in één tabel (met voor elk antwoord een apart veld).
- Door middel van een voorselectie aan de hand van geslacht, leeftijd, zwangerschap, aantal kinderen en de leeftijden hiervan, wordt een zogenaamde DNA-sleutel samengesteld. Bij elk van de vragen kunnen vervolgens eisen aan deze sleutel worden gegeven zodat voorkomen wordt dat vragen die bijvoorbeeld alleen logisch zijn om aan een vrouw te stellen, niet aan een man worden gesteld.

In deze situatie zijn diverse trends:

- In de loop van tijd worden alle nog operationele F1-acties omgezet in F2-acties. Dit betekent overigens niet dat de selectietool niet hoeft te kunnen selecteren uit data afkomstig van F1-acties, want informatie uit het verleden blijft in de oude F1-structuren bestaan terwijl er nog wel uit geselecteerd moet worden. Tegelijkertijd is ook F2 nog niet perfect om helemaal te kunnen voldoen aan de juiste mix tussen 'wie, wat, waar en wanneer' voor het stellen van vragen, dus voor de toekomst zit er wellicht een "F3" aan te komen.
- Er worden vooral in het buitenland steeds meer acties uitgerold. Hiervoor is het online databasecluster al uitgebreid met een DBMS voor deze buitenlandse acties, maar het is ook voor de toekomst zeker niet uitgesloten dat dit zich herhaalt. Ook aan de offline-kant zou een uitbreiding in de toekomst nodig kunnen blijken.

Alternatieven

In de vorige paragraaf is al een schets gemaakt van de probleemstelling. Om de probleemstelling af te dekken is de keuze gemaakt voor het ontwerpen, beschrijven en ontwikkelen van een nieuw systeem (het doel van de opdracht). Er waren echter ook alternatieve oplossingsmogelijkheden voorhanden die door het bedrijf al voor de aanvang van de afstudeerperiode in overweging zijn genomen. In deze paragraaf worden deze mogelijkheden beschreven, en worden de redenen in beeld gebracht die hebben geleid tot de keuze van deze nieuwbouw.

De volgende alternatieven zijn voorhanden:

- Het uitbreiden van het personeelsbestand, zodat meer mensen zich tegelijkertijd kunnen bezighouden met het (handmatig) uitvoeren van selecties.
- Idem, maar dan met een verandering van de procedure, hierbij kan bijvoorbeeld gedacht worden aan een aparte werknemer die zich bezighoudt met het opstellen en controleren van kwaliteitsregels of planning of een combinatie
- Het aanschaffen en in gebruik nemen van andere, alternatieve of additionele programmatuur voor de selectie van data, niet specifiek aan deze situatie.
- Het aanpassen of opnieuw bouwen van de intern gebruikte technologieën voor het verzamelen en opslaan van de data op een dusdanige manier dat het selecteren van data gemakkelijker wordt.

De eerste twee punten zijn mogelijkheden, maar bieden slechts een tijdelijke dekking van het probleemgebied. Met de door het bedrijf verwachte groei de komende jaren zal het knelpunt snel weer op gaan spelen. Door daarna nog meer mensen aan te nemen, begint het financiële aspect op te spelen. Daarnaast is er niet concreet gezorgd voor verbetering van de efficiëntie.

Er zijn genoeg programma's op de markt die functionaliteiten aan boord hebben op het gebied van dataselectie en nabewerking. In totaal zijn dit misschien wel tientallen producten, maar die hebben allemaal dezelfde beperking als het momenteel veelvuldig gebruikte grafische databasebeheer-pakket, namelijk: deze programma's zijn geen van allen opgezet met functionaliteiten op het niveau zoals E2Ma deze zou kunnen gebruiken. Voor alle pakketten is alleen de basis aanwezig in de vorm van basisbouwstenen, en het combineren hiervan kost nou juist veel tijd.

E2Ma heeft geconcludeerd dat het bouwen van maatwerk de meest logische oplossing is. Er zullen geen pakketten bestaan die alle of veel eisen van de operationele medewerkers al aan boord hebben, simpelweg omdat het systeem met enquêtes en de opslag hierachter ook in eigen beheer is gebouwd. Het aanpassen van dit reeds

bestaande systeem is tenslotte ook geen optie omdat dit grote implicaties zou hebben voor alle randsystemen en dergelijke. En de gedane investeringen voor het in eigen beheer ontwikkelen van deze maatwerksystemen worden dan nog even buiten beschouwing gelaten.

De organisatie heeft de eis gesteld dat het stukje te ontwikkelen maatwerk gerealiseerd moet gaan worden als webapplicatie met de scripttaal PHP en eventueel een DBMS van het merk MySQL ter ondersteuning.

3 Aanpak

Vanuit de opdrachtschrijving zijn er verschillende wegen die naar Rome leiden, maar niet allemaal zijn deze even direct en begaanbaar. Hoe de juiste route is uitgestippeld en wat deze is staat in dit hoofdstuk. Er wordt eerst ingegaan op de keuze voor een ontwikkelmethodiek, vervolgens op de projectbesturing in het algemeen, het projectscenario, de iteratiestrategie en het algemeen Plan van Aanpak

3.1 Ontwikkelmethodiek

Bij projecten als deze is er de keuze te maken over welke ontwikkelmethodiek er gebruikt dient te worden. Er is gekozen voor IAD wat staat voor *Iterative Application Development*. Omdat de gekozen methodiek invloed heeft op de planning en producten zoals die - ter goedkeuring van de opdracht voor aanvang van het project - in de opdrachtschrijving vermeld zijn, is de keuze al in een vroeg stadium gemaakt.

De redenen zijn als volgt. Allereerst is IAD de methodiek die vanuit de studie bijna vier jaar lang is aangeleerd. De basis van IAD is hierdoor niet alleen bekend, de benodigde literatuur en een aantal voorbeeldsituaties zijn ook aanwezig. Een tweede reden is dat IAD bij mij een beetje een ondergeschoven kindje is. Er is gedurende de jaren veel gebruik gemaakt van door de school aangeboden stramien en sjablonen die eigenlijk veel te lang zijn herbruikt. De literatuur is hierdoor weinig gelezen en de flexibiliteit van IAD minder doorgekomen. Dit afstuderen wilde ik aangrijpen om IAD zoveel mogelijk zoals het hoort te doorlopen, onafhankelijk van wat in het verleden door school is aangedragen, en zoveel mogelijk zoals de literatuur dit voorschrijft. Dit heeft als achterliggende gedachte dat ik op deze manier voor mijzelf kan nagaan of IAD voor mij een waardig hulpmiddel is voor het oplossen van dit soort vraagstukken en of ik het ook kan toepassen.

Omdat de opdracht op het eerste gezicht weinig kenmerken heeft die overeenkomen met wat er binnen VIA aan vormgevings- en mens-machine-interactie-vakken is gegeven, is bij de goedkeuring van de opdracht overeengekomen, dat er wat elementen uit GUIDE vermengd worden met de activiteiten uit IAD. GUIDE beschrijft een proces voor het ontwerpen van een grafische gebruikersinterface voor een softwaresysteem en staat voor Graphical User Interface Design and Evaluation. Wat deze elementen precies zijn, is te vinden in bijlage 2, het Algemene Plan van Aanpak. Daarnaast is er ook nog een SUMI-test gehouden om de gebruikersvriendelijkheid te kunnen beoordelen.

Binnen IAD is in een later stadium nog een keuze gemaakt voor een zogenaamde iteratiestrategie tijdens het ontwikkelen van het projectscenario.

3.2 Projectbesturing

Vorbereiding

In het prille begin van de periode is er de gebruikelijke dag waarin kennis wordt gemaakt met de medewerkers - voor zover dit tijdens de sollicitatieprocedure nog niet is gebeurd - en de werkplek wordt ingericht. Ook tijdens dit project was dit het geval.

Het inhoudelijke begin was ook wat aftastend. Met de keuze voor de ontwikkelmethodiek IAD was de implicatie vermoed dat het boek behorende bij deze werkwijze grondig moest worden doorgespit. In de afgelopen jaren is deze methodiek aan de onderwijsinstelling niet alleen veelvuldig voorgeschoteld, maar ook in een vorm die het lastig maakt af te wijken van het aangeleerde stramien. Voor dit afstuderen wilde ik het eens een keer goed aanpakken en ben van het begin af aan gaan lezen.

Planning

Het plannen van een automatiseringsproject gebeurt vaak, zometijds altijd, op verschillende niveaus. Bij de aanvang ervan wordt een globale planning opgezet, en gedurende dat het project vordert, worden delen van de planning die aan de beurt zijn verder en gedetailleerder uitgewerkt.

Ook tijdens dit project is dit zo. Aan het begin is een Algemeen Plan van Aanpak opgesteld met de planning over het gehele project met activiteiten en deelactiviteiten tot één niveau diep. Voor elke fase (Definitiestudie, Pilot #1, Pilot #2, enzovoorts) is vervolgens ook nog een fasespecifiek Plan van Aanpak gemaakt, met hierin alleen de activiteiten die behoren tot deze fase, maar met meer detail.

Beheersing

Een planning is er niet om van af te wijken, maar het gebeurt vrijwel altijd. Of het nou is omdat de planning verkeerd is, omdat de werkzaamheden moeilijker of moeizamer zijn dan gepland of vanwege een verminderde beschikbaarheid van het A-team, tijdens de beheersing van het project worden deze afwijkingen geconstateerd, en worden de juiste acties eraan verbonden.

Gedurende het project is het inderdaad een aantal keren voorgekomen dat er van de planning is afgeweken. Halverwege was er net te weinig tijd over om de planning bij te houden vanwege een aantal vrije- en

vakantiedagen. Na 75% van het project werd deze achterstand nog flink wat dagen groter vanwege een onderschatte moeilijkheidsgraad voor Pilot #3, maar vrijwel alle achterstand is hierna gelijk weer weggewerkt door een óverschatte moeilijkheidsgraad voor Pilot #4. Meer over de planning en beheersing in de hoofdstukken hierna.

Afronding

Tijdens de afronding wordt hetgeen geproduceerd is overgedragen. Er wordt nog eens extra gecontroleerd of het op te leveren product voldoet aan de daaraan gestelde normen, en er wordt voor gezorgd dat de oplevering plaats vind via de afgesproken procedures.

3.3 Projectscenario

De eerste activiteit rolde uit het doornemen van het boek van IAD was het opstellen van een Projectscenario. Tijdens deze werkzaamheid zijn knopen doorgehakt ten aanzien van de iteratiestrategie, het aantal pilots en reikwijdte, fasen, activiteiten en deelactiviteiten, producten, teststrategieën, enzovoorts.

Voor ik deze activiteit ben gaan uitvoeren, heb ik mij echter eerst wat verder verdiept in de procesgangen achter IAD. De methodiek kent veel workshops, maar het overgrote deel hiervan verloopt met mij als enig deelnemer. Er zijn echter wel twee belangrijkere workshops ten aanzien van het projectscenario (de ontwikkelscenario-workshop) en het pilotplan (de pilotplan-workshop). Voor een vlotte afhandeling voor de betrokken personen en de verwachting dat de ontwikkelscenario-workshop niet zoveel tijd in beslag zou nemen, heb ik deze twee workshops samen gehouden.

Omdat er bij de betrokken personen niets tot weinig bekend was over IAD en mijn in de toekomst te volgen traject, heb ik uitleg hierover opgenomen in het document dat gold als voorbereiding van deze workshops (zie bijlage 6). Het was ook voor mijzelf gelijk een goede manier om wat ik gelezen had te reproduceren. Als deel van dit document heb ik ook een aanzet tot een systeemconcept op papier proberen te zetten, en dan vooral een manier waarop een selectie kan worden gespecificeerd: waar een selectie uit kan bestaan. Voor en tijdens de workshop kwam hierover echter de opbouwende kritiek naar voren dat dit te abstract was opgezet. Aan de ene kant was dit ook wel een doel van deze opzet, want het maakt snel duidelijk wat de mogelijkheden zijn. Het was echter gegoten in een wat lastig te lezen notatiewijze en dat maakte het denk ik wat moeilijk.

Een dag voor de workshop heb ik nog een appendix behorende bij het voorbereidend document gemaakt en verspreid waarin ik een agenda en de planning heb opgenomen. Ik had er op dit moment ook voor kunnen kiezen dit appendix nog uit te breiden met een

verconcretisering van de aanzet van het systeemconcept, maar heb hier niet voor gekozen omdat ik dacht dit tijdens de workshop wel op te lossen. Het essentiële wat daaruit duidelijk moest worden was alleen dat een selectie verdeeld is in voorwaarden en dat er recursiviteit kan bestaan.

De workshops zelf gingen redelijk tot goed. Naar mijn idee had ik in de gesprekken wat meer initiatief kunnen en moeten houden omdat het gesprek zich op een gegeven moment zich wat ongecontroleerd over de gebieden heen bewoog, maar desondanks is er veel goed informatie uitgekomen. Het verconcretiseren van de aanzet van het systeemconcept bracht geen problemen met zich mee door voorbeelden te geven was een succes.

3.4 Iteratiestrategie

Een belangrijk facet van de aanpak is de keuze voor de iteratiestrategie. IAD bestaat uit de fasen Definitiestudie, Pilotontwikkeling en Invoering, en binnen het ontwikkelscenario is vast gelegd op welke wijze er door deze fasen heen wordt geïtereerd.

Er zijn vier hoofdvarianten te onderscheiden. Bij evolutionair ontwikkelen worden alle fasen in elke iteratie doorlopen. Bij incrementeel ontwikkelen wordt de fase Definitiestudie slechts éénmaal gevolgd. Bij big-bang-invoeren is het juist andersom: hier vindt alleen de invoeringsfase eenmaal plaats. Tot slot is er ook nog de combinatie tussen de twee laatstgenoemden: incrementeel ontwikkelen. Hierbij wordt alleen de fase Pilotontwikkeling meermalen doorlopen.

Er is gekozen voor incrementeel ontwikkelen. De fase Definitiestudie wordt hierbij eenmalig doorlopen, de fase Pilotontwikkeling – naargelang het aantal tijdens de fase Definitiestudie gedefinieerde pilots – meermalen met ook weer eenmalig de fase Invoering. Er is voor deze iteratiestrategie gekozen omdat het beoogde systeem goed van te voren kan worden gespecificeerd, de omvang van het systeem is te overzien en er weinig behoefte bestaat aan flexibiliteit tijdens het ontwikkelen.

3.5 Algemeen Plan van Aanpak

Het Algemeen Plan van Aanpak is een belangrijk hulpmiddel voor het overzicht, de projectbesturing, en het opstellen van de Plannen van Aanpak voor de volgende fasen. Zie bijlage 2 voor het volledige Algemeen Plan van Aanpak.

Veel van de inhoud bestond al in de opdrachtschrijving, dus het opstellen was meer een kwestie van overnemen, controleren en eventueel bijwerken, uitbreiden of actualiseren.

4 Definitiestudie

De eerste activiteit na het opstellen van het algemene Plan van Aanpak is het uitvoeren van de fase Definitiestudie met de definitiestudie als resultaat. Omdat er bij het maken van de iteratiestrategie voor gekozen is deze fase maar eenmalig uit te voeren, is dit een eenmalige activiteit. Zie bijlage 7 voor het volledige rapport definitiestudie.

4.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld. Zoals eerder al vermeld, verloopt het plannen op meerdere tijdstippen en niveaus. Waar de grote lijnen voor alle fasen uiteen zijn gezet in het algemene Plan van Aanpak, hier komen de activiteiten en producten specifiek voor deze fase aan bod om ingepland te worden.

De beoogde start- en einddatum waren al bekend, en met dat als uitgangspunt zijn de activiteiten ingepland. Bij het opstellen van het algemene Plan van Aanpak is al rekening gehouden met de activiteiten tijdens deze fase, dus het is niet zo dat nu pas duidelijk wordt hoe lang de fase zal duren; puur alleen wanneer welke taak aan zou moeten vangen. Zie bijlage 5 voor het plan.

4.2 Systeemeisen

Met de gehouden workshop als voornaamste basis en mijn inleving ter aanvulling op een enkele plek, is er een begin gemaakt aan het definitiestudierapport. De systeemeisen zijn opgesteld en geprioritiseerd vanuit het oogpunt van de gebruiker. Informatie over de gebruikers zelf is niet alleen uit de workshop naar voren gekomen, maar ook uit ervaringen en (informele) gesprekken. Na het op papier zetten van deze eisen is het werk nog extra gecontroleerd met de verificatie- en validatiepunten uit IAD. Dit heeft er voor gezorgd dat de systeemeisen eenduidig, consistent, veranderbaar, verifieerbaar en herleidbaar zijn. Zie Fragment 1 voor een deel van de overige eisen en Fragment 2 voor een deel van de prioriteiten.

Integriteitseisen zijn de eisen die een bedrage leveren aan de betrouwbaarheid en accuratesse. Het systeem zal de gebruiker om een bevestiging moeten vragen voordat deze op het punt staat een ingrijpende actie uit te voeren (5.1). Daarnaast dienen er controles op gebruikersinvoer aanwezig te zijn om onduidelijkheden terug te dringen (5.2).

Fragment 1 Een deel van de overige eisen.

Nadat er nog een nacht over was geslapen, werd deze opzet toch niet optimaal bevonden. Na wat naslag- en leeswerk in IAD en GUIDE, is hieruit naar voren gekomen dat de eisen beter op een

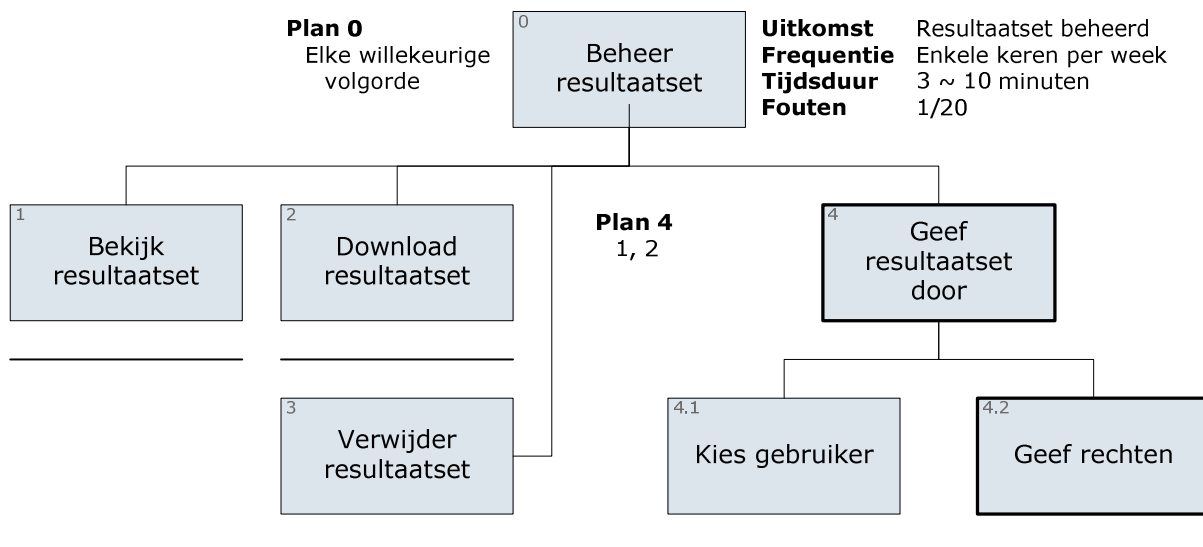
gebruikersneutrale wijze opgeschreven konden worden in plaats van zoals nu vanuit de gebruiker gezien. Het visualiseren van de koppeling tussen systeemeisen en gebruikers is veel meer geschikt om gedragen te worden door taakdiagrammen en use-case diagrammen.

- 4: Gebruikers toevoegen, bewerken en verwijderen.
- 5: Bevestigingen & invoercontroles.
- 2.4: Resultaatsets doorgeven aan iemand anders.

Fragment 2 Een deel van de prioriteiten.

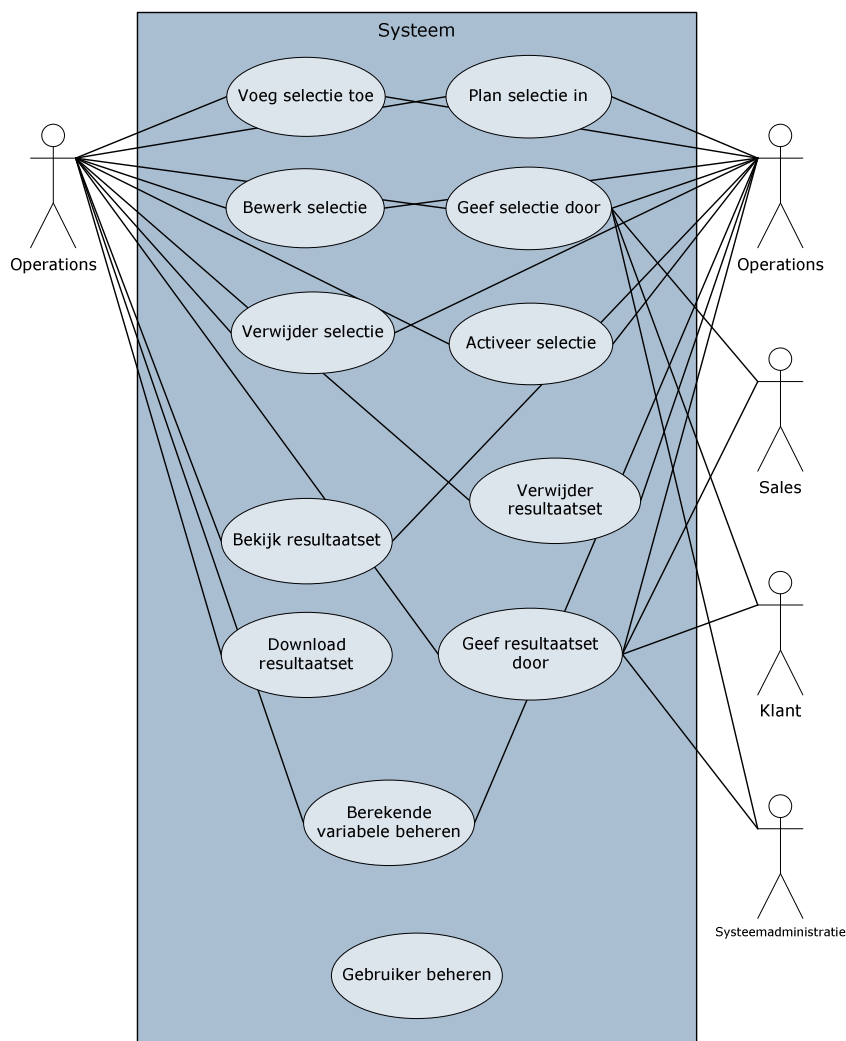
4.3 Taakdiagrammen

Het fabriceren van de taakdiagrammen is gebeurd in twee stappen. Allereerst is er een platte tekstuele indeling van alle taken gemaakt. De taakdiagrammen konden in deze vorm veel makkelijker nog worden aangepast als in de situatie dat de Visietekeningen al zouden bestaan. Er is niet alleen veel aandacht besteed aan de juiste inhoud, maar ook aan de juiste wijze van notatie en aan een consistent uiterlijk. Een hulpmiddel hierbij was het afzonderlijk en van te voren maken van een sjabloon waarbij de kleuren, lijnen, spatieringen en dergelijke vast stonden. Dit geeft het geheel een schone aanblik en is snel te begrijpen. Zie Figuur 2 voor een voorbeeld van een taakdiagram.



Figuur 2 Een van de taakdiagrammen.

Een klein probleem wat naar boven kwam tijdens het maken van de taakdiagrammen was het feit dat de bedrijfsprocessen voor een groot deel alleen in het digitale plaats vinden, en dat er hierdoor weinig tot geen *real-life*-objecten zijn terwijl hier in de afgelopen jaren altijd wel zo tegenaan is gekeken. Ik heb ervoor gekozen extra goed af te wegen hoe dit feit invloed heeft op de afbeeldingswijzen. Een voorbeeld van een concept die alleen in de digitale wereld bestaat is de variabele. Dit object (en de bijbehorende taken in het taakdiagram) is wel belangrijk in de bedrijfsprocessen, maar er wordt nooit echt direct mee gewerkt.



Figuur 3 Één van de verschijningsvormen van het use-case diagram.

4.4 Taakscenario's

Er zijn drie belangrijke gebruiksvoorbeelden van het systeem bedacht, namelijk 'Selectie uitvoeren', 'Resultaatset doorgeven' en 'Selectie bewerken'. Voor elk van deze scenario's is een omschrijving gegeven, wat omgevingssituaties en een beknopt stappenplan voor wat de gebruiker op het scherm voor acties moet uitvoeren om tot uitvoering van het scenario te komen.

4.5 Use-case diagram

Van de systeemeisen komt via een concretere weergave van wat er met het systeem gedaan moet kunnen worden het use-case diagram aan de beurt. Hierin is aangegeven welke personen, gebruikers en/of gebruikersgroepen in de meest vrije situatie toegang moeten hebben tot welke functionaliteiten en wie er vervolgens, aan de andere kant van het diagram, ook baat bij hebben.

Met een handjevol hoofdtaken en acht actoren binnen één conceptueel systeem, ontstond er in het diagram met alle actor-taak-combinaties haast een massief zwart vlak gevormd door alle verbindingslijnen. Er is daarom voor gekozen het diagram uit te splitsen in vier verschijningsvormen, in overeenstemming met met de vier gebruikersgroepen danwel actoren aan de initiërende kant. Zie Figuur 3 voor één van deze verschijningsvormen.

4.6 Klassendiagram

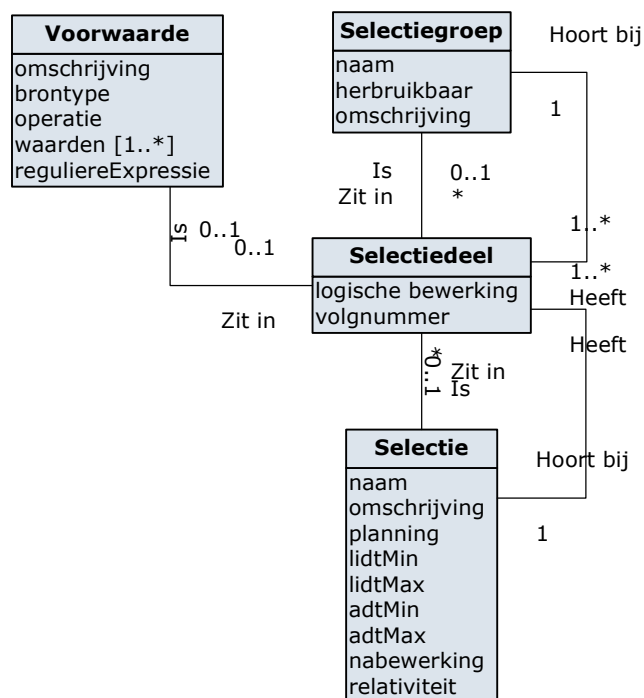
Toen het use-case diagram was afgerond, kon er verder worden gegaan met het opzetten van het klassendiagram. Dit diagram is een belangrijk hulpmiddel bij het vastleggen van de structuur der gegevens en is zeer belangrijke invoer voor de weg naar een fysieke database-implementatie. Als er over gegevens wordt gepraat, worden hiermee de gegevens die te maken hebben met selectieproces bedoeld.

Tijdens het op papier zetten van de eerste ideeën, kwamen er diversie problemen om de hoek kijken:

- Er zijn nu hoofdzakelijk twee verschillende technieken. Zo bestaat momenteel de 'F2'-techniek, maar deze heeft ook nog een voorganger gehad, namelijk de 'F1'-techniek. Beide technieken hebben een andere wijze van opslag van data en het is lastig hier op dit moment mee om te gaan.
- De belangrijkste consumentengegevens staan opgeslagen in 'profiles'. Deze gegevens staan nu al verdeeld over fysiek verschillende locaties, terwijl de selecties over al deze gegevens in één keer plaats moet kunnen vinden. Vragen over hoe deze onafhankelijkheden en schaalbaarheden moeten worden opgeslagen zijn hierbij dus aan de orde van de dag.
- Een van de eisen was dat er bij het aanmaken van een variabele, de in de reeds bestaande systemen gebruikte naam geselecteerd moet kunnen worden door middel van een wizard. Ook hier is weer de vraag hoe dit opgeslagen moet worden, nog even daargelaten hoe de wizard hier later mee om zou moeten gaan, wat ook invloed kan hebben op de klassen in het diagram.

Al met al kan er gezegd worden dat het maken van het diagram in het begin erg stroef verliep. Dit kwam niet alleen door bovenstaande problemen, maar ook door de drang tegelijkertijd de kern van het systeem al uit te denken, namelijk het algoritme die daadwerkelijk de data uit het systeem selecteert. Bepaalde delen uit dit algoritme impliceren klassen, attributen of relaties en vice versa. Daarnaast was algemene kennis over selecties toentertijd nog erg summier wat voor het bedenken van het algoritme, en laat staan het valideren, niet echt als een duw in de rug is gekomen.

Door eerst verder te verdiepen in de materie achter de bestaande architectuur, kon het proces met meer zekerheid worden vastgesteld worden waarna het klassendiagram ook afgerond kon worden. Zie Figuur 4 voor een deel van het klassendiagram.



Figuur 4 Een deel van het klassendiagram.

In dit deel zijn alleen de klassen ter representatie van de structuur van een selectie weergegeven. Een selectie bestaat uit een aantal voorwaarden waarmee bij uitvoering van elke deelnemer wordt gecontroleerd of hij voldoet en opgenomen kan worden in het eindresultaat. Deze voorwaarden hoeven niet allemaal naast elkaar te bestaan, maar kunnen gegroepeerd worden in selectiegroepen of andere selecties. Hiervoor is de klasse Selectiedeel in het leven geroepen om van elk deel van een selectie te kunnen uitmaken of dit een groep, voorwaarde of andere selectie is. Om het rondje compleet te maken: ook groepen hebben weer selectiedelen. Al met al kunnen op deze manier zeer complexe selecties toch op een relatief eenvoudige manier worden opgeslagen. Er zal wel gewaakt moeten worden voor oneindige recursie, waarbij een kringverwijzing ontstaat binnen een combinatie van selectiedelen en selectiegroepen of selecties.

Op enkele plekken in het diagram is afgeweken van de standaardnotatiewijze. Op sommige plekken zijn multipliciteiten weggelaten, van sommige attributen zijn beperkingen met accolades aangegeven en niet alle relaties hebben namen. Soms is dit gedaan om het klassendiagram overzichtelijk te houden, maar ook om informatie op bepaalde plekken overzichtelijk bij elkaar te houden. Het diagram is compact, maar met enig redeneren moet het nog goed mogelijk zijn

alle multipliciteiten en relatienamen terug te brengen tot de goede plek.

Net als bij het opstellen van de taakdiagrammen, kwam hier de lastigheid om de hoek kijken in verband met dat de objecten waarover gesproken wordt, alleen in het virtuele bestaan. Het was hierdoor lastig sommige beslissingen te nemen. Zo kan de associatieklasse Resultaat tussen Variabele en Resultaatset ook net zo goed een normale klasse zijn met twee één-op-veel relaties. Dit ligt ook gelijk meer in de lijn van het implementatiemodel, maar hier mag bij het opstellen van het diagram niet naar gekeken worden.

Parallel aan het maken van het klassendiagram, zijn er tekstuele omschrijvingen gemaakt van de klassen en waar nodig ook van attributen. Dit is allemaal gebeurd met gedeeltelijke inachtneming van het aanbevolen traject volgens GUIDE. Gedeeltelijk omdat het ten eerste een mix blijft met IAD en ten tweede omdat ieder project weer uniek is. Zie Fragment 3 voor een van de omschrijvingen.

De objecten van deze klasse representeren elk een Actie zoals 'Prijnsreis' en 'Win een droomreis'. Er is plaats gemaakt voor de twee laatste technologieën, type 1 (generatie 2) en type 2 (F2, generatie 3).

Standaard heeft een Actie één koppeling met een DBMS, maar deze kan voor verschillende zaken worden overschreven. Dit is het geval voor de DBMS voor de enquête-informatie, de profiles-data, de sponsortabel en de unsubtabel. Indien deze en bijbehorende attributen niet worden aangegeven, wordt uitgegaan van de standaardlocaties. Voor de unsub kan ook worden aangegeven wat de waarde kan zijn om een deelnemer als uitgeschreven te kunnen beschouwen.

Het attribuut enquetetabel is een prefix waar 'questions' en 'answers' achter wordt geplakt om achter een variabelennaam te kunnen komen.

Fragment 3 Een deel van de omschrijvingen.

In een tussentijds gesprek met de bedrijfsbegeleider, is besproken dat integratie met bestanden van derden niet nodig is. Tevens hoeft er geen directe interface te komen met externe validatieservices als Adres Correct¹. Om al deze zaken alsnog te kunnen integreren kan op een later moment buiten het systeem om alsnog rekening gehouden worden door deze gegevens te integreren in de databases, parallel aan bijvoorbeeld de voornaam van een deelnemer. Later in dit gesprek kwam ook naar boven dat er gezien de hoeveelheid werk die tot het gesprek aan toe in de definitiestudie is gaan zitten, het misschien verstandig had geweest om na afloop van de workshop een terugkoppelmoment te hebben naar diegene die bij de workshop

¹ Een commerciële dienst die onder www.adrescorrect.nl wordt geëxploiteerd ter validatie van adressen.

aanwezig zijn geweest. Dit zou dan fungeren als een extra zekerheid voor de uit de workshop gehaalde informatie.

In de tijd erna, ook tijdens de volgende fasen, zijn nog kleine wijzigingen doorgebracht aan het klassendiagram, maar elk van deze wijzigingen zijn niet van noemenswaardige grootte.

4.7 Selectieproces

In combinatie met het opstellen van de taakdiagrammen en later ook met het klassendiagram, was tegelijkertijd al begonnen met het schetsen van het te volgen proces. Dit proces zal worden uitgevoerd op het moment dat er een selectiespecificatie moet worden gebruikt om met de data in de databases te komen tot resultaat.

Eerst is er voorzichtig een schets op papier gemaakt tegelijkertijd met het uitzoeken van hoe data nu is opgeslagen en wat de structuur van een specificatie van een selectie zou kunnen worden. Nadat meer zekerheid was verkregen, is dit proces in platte tekst in een tekstbestand gezet. Het stond nu op een hiërarchische wijze op chronologische volgorde. De hiërarchie was nodig om overzichtelijk te maken waar iteraties of beslissingen voorkomen. Zie Fragment 4 voor een deel van de procesomschrijving.

```
- Alle gekozen acties doorlopen, per actie:
  - Alle deelnemers doorlopen (uit de bij de actie behorende
    profiles-tabel), per deelnemer
    - De gewenste variabelen achterhalen door
      - F1: Selecteer van de opslagtabel de gewenste
        variabelen van alle records.
      - F2: Van elk van de gewenste opslagtabellen:
        - Selecteer het antwoord.
    - Per resultaat-rij:
      - Bekijk of de variabelen door de selectie komen (op
        basis van laatste resultaat)
      - Zo ja: deelnemer in de uitvoer zetten met
        gevraagde variabelen.
```

Fragment 4 Een deel van het proces.

Nadat de eerste versie op papier stond, is deze in gedachte een keer doorlopen zoals het in het echt ook zou gaan. Hierbij zijn nog enkele optimalisaties naar boven gekomen.

4.8 Stijlgids

In IAD wordt er niet direct vermelding van gemaakt, maar vanuit GUIDE bestaat de vraag naar een stijlgids. Hierin is vastgelegd welke uitstraling de interface moet hebben. Er is hier en daar gekeken naar al bestaande maatwerksystemen binnen het bedrijf. Daar kwam geen eenduidige look en feel vanuit waar ik het te bouwen systeem in kon indenken, dus is er vanaf nul begonnen met het beschrijven van de eisen. Zie Fragment 5 voor een deel uit de stijlgids.

De verschillende schermen moeten een vergelijkbare indeling hebben. Het menu moet hierbij altijd aan de linkerkant van het scherm staan.

Fragment 5 Een deel uit de stijlguides.

4.9 Pilotplan

Nu de systeemeisen bekend zijn en de afgeleiden zijn gecreëerd, is het tijd voor het pilotplan. Hierin staat een geprioritiseerde lijst van pilots die sequentieel of parallel ontwikkeld gaan worden. Elke pilot heeft hierbij betrekking op een bepaald deel van het systeem. Van elke pilot wordt een inschatting gemaakt van de tijd die voor de ontwikkeling benodigd is en er wordt een voorlopige verdeling in pilotdelen gemaakt.

#	Naam	Omschrijving	Uren	Prioriteit
1	GUI-standaard	Standaardisering van communicatie met de gebruiker en interne aangelegenheden (code-gewijs).	56	Kern
2	Variabelebeheer	Toevoegen, bewerken & verwijderen van variabelen.	25	Basis
3	Selectiebeheer	Toevoegen, bewerken & verwijderen van selecties.	57	Basis
4	Selectieactivatie	Het daadwerkelijk uitvoeren van een selectie: het aan de hand van een selectie, variabelen en data in de databases produceren van een resultaatset.	84	Basis
5	Resultaatset-beheer	Bekijken, downloaden & verwijderen van resultaatsets.	25	Basis
6	Gebruikersbeheer	Toevoegen, bewerken (m.n. de rechten) & verwijderen van gebruikers.	9,5	Comfort

Tabel 1 Een opsomming van de pilots.

Een belangrijke keuze bij dit soort projecten is of de database van het klassendiagram in één keer moet worden omgezet of dat dit bijvoorbeeld gebeurt per pilot. In voorgaande jaren is het alleen maar voorgekomen dat dit gebeurde in een afzonderlijke eerste kernpilot, maar voor dit project is besloten het anders aan te pakken. Uit deze voorgaande projecten kwam namelijk naar voren dat de druk op het correct bouwen van de pilot groot is omdat de goede werking van veel volgende pilots hierop berust. De gedachtegang dat het opsplitsen van de database naargelang de verdeling in overige pilots ook deze druk zou verminderen en het werk tevens aantrekkelijker zou maken, is als interessant bevonden. Er is dus voor gekozen de database niet als aparte pilot op te voeren. In plaats hiervan zullen de relevante pilots een databasedeel krijgen waarin de voor de desbetreffende pilot benodigde databasestructuren worden gemaakt.

#	Naam	Omschrijving	Uren	Prioriteit
1	Database	De benodigde meta-data in de database t.b.v. de variabelen.	4	Basis
2	Overzicht	Het kunnen oproepen van een overzicht van alle op dat moment	4	Basis

		bij het systeem bekende variabelen.		
3	Toevoegen	Het kunnen toevoegen van een variabele.	12	Basis

Tabel 2 Een voorbeeldoverzicht van pilotdelen.

Na deze keuze zijn de pilots en de delen opgesteld en is er een inschatting gemaakt van de benodigde tijd. Zie Tabel 1 voor een opsomming en Tabel 2 voor een voorbeeldoverzicht van pilotdelen.

4.10 Overig

De technische structuur en de organisatorische inrichting zijn gemaakt zoals IAD voorschrijft. Ook hier zijn een aantal activiteiten weggevallen vanwege de aard van het project. Voor wat betreft de technische structuur is volstaan met een omschrijving van de tiers en een invulling van de systeemeisen bij de ontwikkeling en na invoering. Bij de organisatorische structuur is alleen vermeld welke soorten en typen gebruikersdocumentatie er moeten komen.

Na het afronden van het rapport en het bijbehorende pilotplan, is er een kleine tweede workshop belegd voor de controle van wat op papier is gezet. Voor aanvang van de workshop is er weer een agenda gemaakt. De indeling in pilots en pilotdelen werd hierbij goed bevonden.

5 Pilot #1, GUI-standaard

In dit hoofdstuk staan de activiteiten beschreven ten behoeve van de eerste pilot. Deze pilot gaat in op de ontwikkeling van een zogenaamde GUI-standaard. Dat is een raamwerk waarbinnen alle schermen van de overige pilots makkelijk in gemaakt kunnen worden wat automatisch leidt tot een hoge mate van consistentie. Zowel aan de buitenkant als aan de binnenkant (code-gewijs). Zie bijlage 9 voor het volledige pilotontwikkelplan.

5.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld. Zoals eerder al vermeld, verloopt het plannen op meerdere tijdstippen en niveaus. Waar de grote lijnen voor alle fasen uiteen zijn gezet in het algemene Plan van Aanpak, hier komen de activiteiten en producten specifiek voor deze fase aan bod om ingepland te worden. Zie bijlage 8 voor het plan.

De beoogde start- en einddatum waren al bekend, en met dat als uitgangspunt en de in de pilotplan geschatte werkbelastingen meegerekend, zijn de activiteiten ingepland.

Er was aan het begin nog wel wat twijfel over de relatie tussen de fasen Pilotontwikkeling, de verschillende pilots en het Plan van Aanpak binnen elk van deze fases. In het IAD-boek is het een en ander wat dubbelzinnig vermeld, waardoor de weg ook vrij leek voor één Plan van Aanpak voor alle fasen Pilotontwikkeling. Uiteindelijk is er wel gekozen voor zes afzonderlijke Plannen van Aanpak voor de overzichtelijkheid.

5.2 Globaal-functionele structuur

In de globaal-functionele structuur komen de functionaliteiten naar voren die in deze pilot zijn ontworpen en gebouwd. Deze functionaliteiten zijn geen detaillering van iets wat in het systeemconcept staat, maar is meer een afgeleide omdat de pilot zelf ook meer een raamwerk is. Er is gekeken naar de mogelijke grafische gebruikersinterfaces die voortkomen uit het systeemconcept van de definitiestudie, en deze zijn vervolgens gegeneraliseerd tot een paar basisdelen. De indeling van de pilot in pilotdelen is hierbij ook als leidraad genomen.

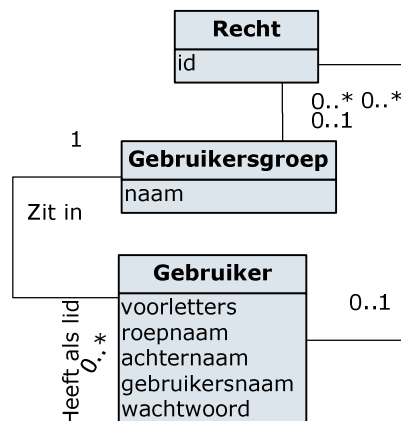
5.3 Globaal-technische structuur

Voor in de globaal-technische structuur zijn de voor deze pilot relevante klassen omgewerkt tot een relationeel representatiemodel en een relationeel implementatiemodel. Daarnaast is er ook een rechtensysteem uitgedacht.

Databasestructuur

Voor wat betreft het omzetten van de relevante klassen is er eerst een keuze gemaakt voor welke klassen uit het klassendiagram in de definitiestudie er omgezet moeten gaan worden. Omdat de code achter het inloggen ook tijdens deze fase geschreven is, is ervoor gekozen de klassen 'Recht', 'Gebruikersgroep' en 'Gebruiker' om te zetten.

Eerst was ik nog van plan bij deze omzettingen gebruik te maken van een (E)ER-diagram als tussenstap, maar omdat de inhoud van dit model dermate dicht bij het klassendiagram staat en daardoor de hoeveelheid toegevoegde informatie niet opweegt tegen de kosten van het maken, heb ik ervoor gekozen deze tussenstap over te slaan. De reden van dat er in eerste instantie wel het plan bestond (E)ER-diagrammen te maken was omdat ik in de veronderstelling was dat het normalisatieproces hier (voor een deel) al plaats vond. Vandenbulcke e.a. (2000) bracht andere informatie.



Figuur 5 Deel van het klassendiagram ten behoeve van deze pilot.

Met het klassendiagram hebben we de objecten, attributen en relaties reeds in kaart gebracht. In het proces tot het implementeren van de database, kan niet alles hieruit 1-op-1 worden overgenomen. De vertaalslag vindt plaats in de tussenliggende activiteiten van het samenstellen van een relationeel representatiemodel. Hierbij worden ook alle normalisaties behandeld. Vanuit het relationeel representatiemodel ligt de weg vrij tot het maken van een relationeel implementatiemodel.

Normaliseren is een stapsgewijs proces wat op relaties van een relationeel representatiemodel wordt toegepast om uiteindelijk te komen tot een gegevensmodel zonder overtolligheden en tegenspraken. Er is getracht te komen tot volledige normalisatie (2NF, 3NF & 4NF). Dit betekent:

- dat er geen meerwaardige attributen meer zijn;
- ieder attribuut dat geen deel uitmaakt van een kandidaatsleutel, volledig functioneel afhankelijk is van elke kandidaatsleutel;

- er geen transitieve afhankelijkheden meer zijn;
- elke determinant een kandidaatsleutel is van de relatie;
- er geen meerwaardige afhankelijkheden meer zijn.

Initieel bestaat de volgende relatie:

```
recht(id)
```

Een recht hoort bij 0 of 1 gebruikersgroepen of bij 0 of 1 gebruikers. Hierdoor krijgen we het volgende:

```
recht(id, gebruikersgroepId, gebruikerId)
```

Omdat recht nu een zwak entiteitstype is (het is bestaansafhankelijk van óf gebruikersgroep óf gebruiker), nemen we deze twee vreemde sleutels op als extra attribuuttypen voor de primaire sleutel:

```
recht(id, gebruikersgroepId, gebruikerId)
```

- gebruikersgroepId: vreemde sleutel, verwijst naar id in gebruikersgroep, null toegelaten.

gebruikerId: vreemde sleutel, verwijst naar id in gebruiker, null toegelaten.

Fragment 6 De resultaten van het omzetten van de klasse Recht in een relatie.

Zie Fragment 6 voor voorbeeldresultaten van het omzetten van een klasse in een relatie. Zie Figuur 5 voor de klassen 'Recht', 'Gebruikersgroep' en 'Gebruiker'. Het attribuut `id` van de klasse 'Recht' is een simpel getal waar door het systeem een betekenis aan wordt gegeven.

Het is vaak voorgekomen dat er, ondanks dat er al één of meer kandidaatsleutels bestaan, toch nog een extra numerieke primaire sleutel is toegevoegd, en wel om de volgende redenen:

- Opslagefficiëntie. Een relationele database is *value*-based en dat betekent dat de primaire sleutel waar de vreemde sleutel naar verwijst letterlijk wordt overgenomen in de verwijzende tabel. Het kost extra schijfruimte voor dit attribuut, maar deze wordt dubbel en dwars teruggewonnen bij de vreemde sleutels.
- Geen verwijzingsproblematiek door vreemde tekens. Vanuit de webapplicatie zullen primaire sleutels (ook in de gedaante van vreemde sleutels) veelvuldig heen en weer worden gezonden tussen de verschillende tiers. Omdat de kanalen waar deze gegevens door reizen gevoelig zijn voor vreemde tekens, kan dit problemen opleveren als er zich daadwerkelijk vreemde tekens bevinden in de normale kandidaatsleutels.
- Veranderingen van kandidaatsleutels die ook bestaan als vreemde sleutels. Omdat ondersteuning voor het controleren en bijhouden van referentiële integriteit op het productieplatform nog niet worden ondersteund, is het veel veiliger te kiezen voor een primaire sleutel die nooit hóeft te veranderen.

Na deze conversie was het relationeel representatie binnen handbereik. Met SQL als DDL ontstonden kant en klare definities van de tabellen met attributen en relaties. Zie Fragment 7 voor de SQL-CREATE-opdracht voor het aanmaken van de tabel `recht`. Hoewel MySQL het afdwingen van relationele integriteit niet kent (tijdens productie zal er een versie in gebruik zijn die deze mogelijkheden nog niet ondersteunt), zijn er toch `FOREIGN KEY`-delen opgenomen voor voorwaartse compatibiliteit. In de applicatie-tier zullen deze regels alsnog worden geïmplementeerd.

```
CREATE TABLE recht
(
    id MEDIUMINT UNSIGNED NOT NULL,
    gebruikersgroepId TINYINT UNSIGNED,
    gebruikerId SMALLINT UNSIGNED,
    PRIMARY KEY (id, gebruikersgroepId, gebruikerId),
    FOREIGN KEY (gebruikersgroepId)
        REFERENCES gebruikersgroep ON DELETE CASCADE,
    FOREIGN KEY (gebruikerId)
        REFERENCES gebruiker ON DELETE CASCADE
);
```

Fragment 7 De SQL-CREATE-opdracht voor het aanmaken van de tabel `recht`.

Rechtensysteem

Het mechanisme achter het beperken van de bewegingsvrijheid van gebruikers wilde ik opzetten met een oog op de toekomst. Het moest niet allen alle relevante acties van alle basis-pilotdelen ondersteunen, er moest ook ruimte zijn voor de acties van comfort- en luxe-delen voor zover van toepassing. Met terugkijkende blik op het verleden en soortgelijke problemen, heb ik gekozen voor bitcoderingen. Ik heb hiervoor gekozen omdat de opslag ervan efficiënt is en het - als het eenmaal draait - eenvoudig, compact en leesbaar is te implementeren.

Net als het attribuut `id` van de klasse 'Recht', bevat de kolom `id` van de tabel `recht`, steeds een simpel getal verwijzend naar een specifiek recht. Zo verwijzen recht 16, 32 en 64 respectievelijk naar de mogelijkheden voor het bewerken, verwijderen en activeren van een selectie. Om de rechten bitgewijs te kunnen vergelijken, zijn alle getallen machten van twee.

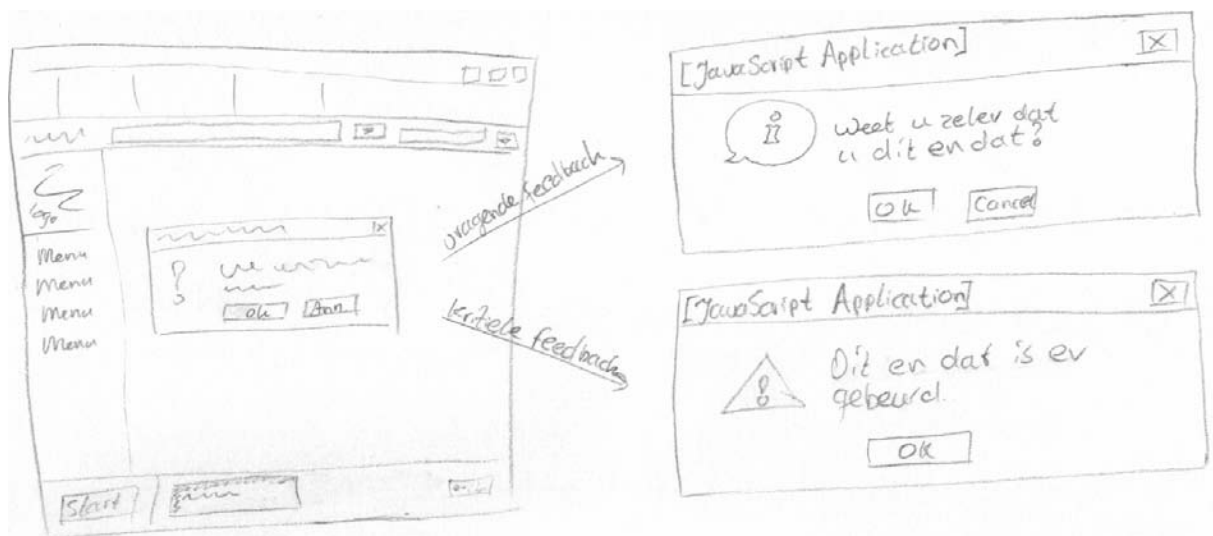
5.4 Globaal-organisatorische inrichting

In dit hoofdstuk is de organisatorische inrichting behandeld. Er is een specificatie gegeven van de organisatorische aspecten rond de pilot in ontwikkeling voor zover deze in het belang zijn voor de gebruikers. Uit het eerste deel van de workshop is al naar voren gekomen dat er geen behoefte is aan een handleiding, behalve dan een korte omschrijving van hoe het systeem moet worden geïnstalleerd, geconfigureerd en onderhouden. Er is dan ook vastgelegd dat er een productiehandleiding moet komen met hierin de genoemde punten.

5.5 GUI-ontwerp

In het hoofdstuk GUI-ontwerp is het ontwerp van de grafische gebruikerinterface aan bod gekomen. Omdat deze pilot een kernpilot is en alle mogelijke schermen moet beschrijven, zijn deze schetsen belangrijk geweest omdat ze voor een deel de look en feel van het hele systeem hebben bepaald.

Voor wat betreft schermschetsen is allereerst de vraag gesteld over welke schermen er geschetst moeten worden. Niet alle mogelijke schermen en situaties hoeven op papier te belanden, en geen van de schermen hoeft tot in de kleine details uitgewerkt te worden. Er is gekozen schetsen te maken voor de verschillende vormen van feedback en voor de twee soorten overige schermen, namelijk de overzichtschermen en de detailschermen. Zie Figuur 6 voor de schets voor feedback die middels JavaScript aan de man wordt gebracht.



Figuur 6 De schets voor feedback die middels JavaScript aan de man wordt gebracht.

Het mooie van schetsen is natuurlijk dat het een goed beeld geeft van hoe het eruit komt te zien, maar dat het aan de andere kant toch nog makkelijk te wijzigen is. De gum is dan ook wel een aantal keren ter hand genomen.

Voor het hoofdstuk GUI-ontwerp is tot slot ook nog een navigatieschema gemaakt voor de schermen 'Beginscherm', 'Overzichtscherm' en 'Detailscherm'.

5.6 Testplan

Voor het testplan zijn een aantal testscenario's samengesteld en uitgelegd die verderop gebruikt kunnen worden voor het testen van de pilot. Ik heb geprobeerd zoveel mogelijk alle mogelijkheden in zo compact mogelijke scenario's te vatten. Zie Fragment 8 voor één van de scenario's.

Zorg ervoor dat het systeem een fout genereert in productiemode. Er dient nu een foutrapport verstuurd te worden naar het van te voren opgegeven e-mailadres.

Fragment 8 Één van de scenario's.

5.7 Ontwerp & bouw

In deze paragraaf worden alle werkzaamheden, aangaande het ontwerp en de bouw van de verschillende delen van deze pilot zelf, behandeld.

Coderaamwerk

Het coderaamwerk zorgt voor een consistent gebruik van commentaarinstructies en vergelijkbare dingen. Ik heb voor mijzelf altijd de richtlijn een apart PHP-bestand aan te maken per 'object'. De functionaliteiten moeten in elk van deze scripts vervolgens ook weer op een vergelijkbare wijze worden geïmplementeerd voor de leesbaarheid en onderhoudbaarheid. Zo begint elk van deze bestanden met de functie `doBegin()`. In deze functie wordt gekeken naar welke actie er uitgevoerd moet worden, en roept de corresponderende functie hiervoor aan. Zie Fragment 9 voor de eerste regels van zo'n script.

```
<?php

/**
 * Copyright © 2005 - E2Ma Consumer Profiling
 * Auteur: Ronald Paul
 *
 * Dit is een testbestand.
 *
 * Datum: 20050329
 */

include('includes/library.php');
include('includes/config.php');

/*****
 * Dit is de functie die als eerste wordt aangeroepen bij elk request. Het
 * koppelt de binnenkomende action-variabele aan een functie. De rechten-
 * controle vindt ook pas plaats in deze functie, omdat we hier geen weet
 * hebben van objecten.
 *
 * Inkomende gegevens:
 * - string $rt['action'] uit $_REQUEST.
 */
function doBegin()
{
```

Fragment 9 Een stukje PHP-broncode ter illustratie van de standaardbecommentariseringen.

Voordat hieraan begonnen kon worden, is er ook nog wat tijd gespendeerd aan het uitzoeken en instellen van het tekstverwerkingsprogramma `vim`. Dit is een erg uitgebreid programma

gebleken, en ik heb het gelijk aangepast aan het coderaamwerk om bijvoorbeeld commentaarsjablonen te kunnen oproepen.



Figuur 7 De vormgeving van het hoofdscherm.

Ook de basis van de grafische interface is hier aan bod gekomen, zie Figuur 7.

Feedback

Het object is met succes toegevoegd

Figuur 8 Positieve feedback.

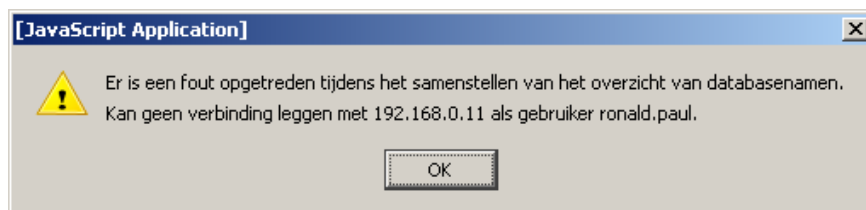
Feedback is informatie die vanuit tier 2 via tier 3 terugloopt naar de gebruiker. Deze informatie is bijvoorbeeld een waarschuwing dat er een gevaarlijke actie op het punt van gebeuren staat, of een stukje informatie dat het wijzigen van het object met succes is voltooid.

Het object kon niet worden toegevoegd

- ◆ de tekst in het invoerveld is te kort (minimaal 3 karakters)
- ◆ de tekst in het invoerveld voldoet niet aan de invoerspecificaties

Figuur 9 Negatieve feedback.

In eerste instantie was er het idee om ook vragende feedback (feedback met directe antwoordmogelijkheden) mogelijk te maken, maar dit heb ik laten vallen omdat hiervoor halsbrekende touren met sessies en dergelijke nodig bleken.



Figuur 10 Kritieke feedback.

In de tussenliggende tijd is er een kleine achterstand ontstaan vanwege niet in de planning meegenomen vrije dagen en vakantiedagen. Aangezien het wel een belangrijke pilot betreft en een kwaliteitsvermindering zou doorklinken in alle volgende pilots, zijn er geen concessies gedaan aangaande het functionele aspect

Foutafhandeling

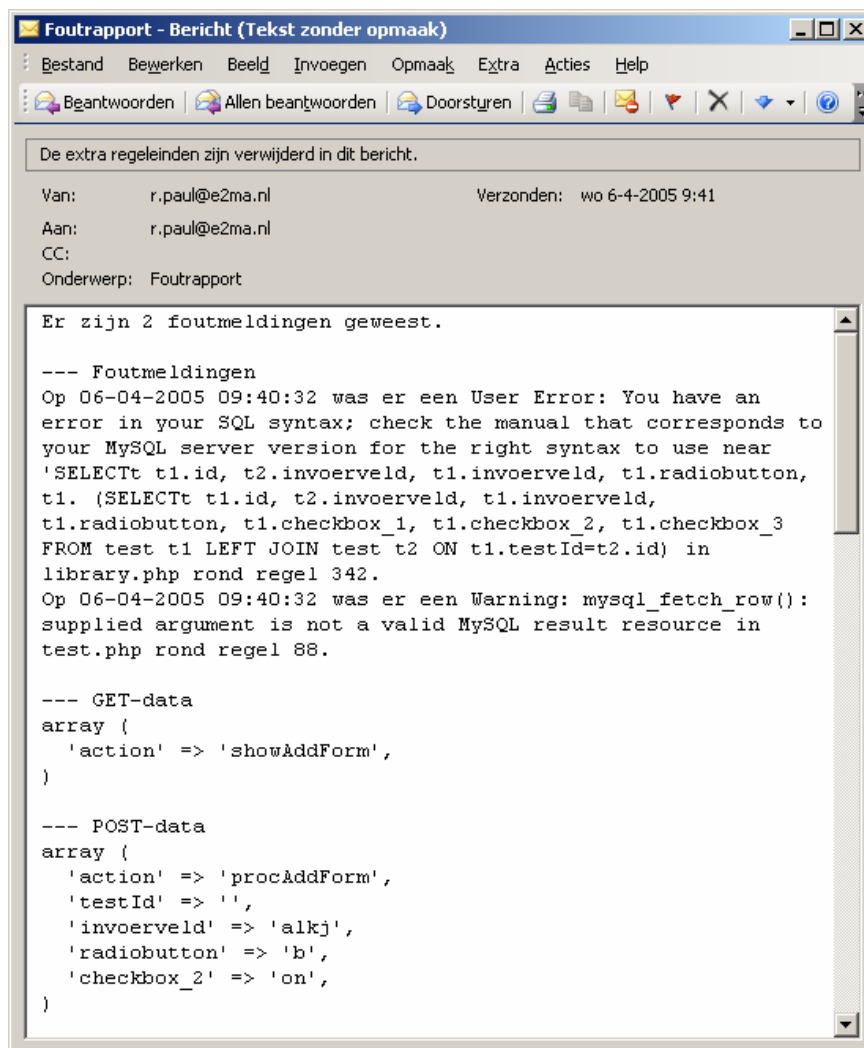
Wederom vanuit ervaringen uit eerdere projecten, was er in gedachten al een basis aanwezig voor de uitwerking voor het afhandelen van fouten. Waar de feedback gebruikersmeldingen en fouten aan de gebruiker brengt, is er ook nog een mechanisme nodig om semantische fouten (wanneer deze naar boven komen) te melden aan de verantwoordelijke. In voorgaande projecten is het subsysteem hiervoor continue geëvolueerd tot wat het nu is. Foutmeldingen worden hierbij - afhankelijk van of het systeem zich in de ontwikkel- of productiemodus bevindt - naar een e-mailadres verstuurd. Zie Figuur 12 voor zo'n e-mail.

Overzichtschermen

Voor een overzicht van objecten zijn er overzichtschermen. In voorgaande projecten waren deze schermen voor de eindgebruiker uitgerust met filter-, sorteer- en bladeropties, maar deze zijn voor dit project niet geïmplementeerd vanwege de verwachting dat deze eigenschappen niet veel toegevoegde waarde zullen hebben.

			Moeder	invoerveld	radiobutton	checkbox_1	checkbox_2	checkbox_3
☐			test 2	test 2	0	0	1	0
☐			test	test 2	0	0	1	0
☐			test 2	test	0	0	1	0
☐			test	test	0	0	1	0
	Selecteer alles							
	Selecteer niets							
	Inverteer selectie							
	Verwijderen							
	Toevoegen							

Figuur 11 Een overzichtscherf.



Figuur 12 Een foutrapport als e-mail gestuurd naar de beheerder.

Detailschermen

In het overzichtsscherm worden van een object niet alle attributen weergegeven en er is aldaar ook geen mogelijkheid tot het bewerken of toevoegen van een object. Voor deze doelen zijn er detailschermen gemaakt.

Er is bij het ontwerp zoveel mogelijk rekening gehouden met de universele inzetbaarheid. De code achter het formulier moet eenvoudig te herbruiken zijn voor ontwikkeling van detailschermen voor andere objecten.

Op de achtergrond is een detailscherm geschikt gemaakt voor hybride gebruik: dezelfde code kan ingezet worden voor het weergeven van een scherm voor het toevoegen van een object, maar ook voor het weergeven van een scherm voor het bewerken van een object.

Moeder-test

Invoerveld

Radiobutton ☒ a
☐ b
☐ c

Checkbox 1 ☐
 Checkbox 2 ☒
 Checkbox 3 ☐

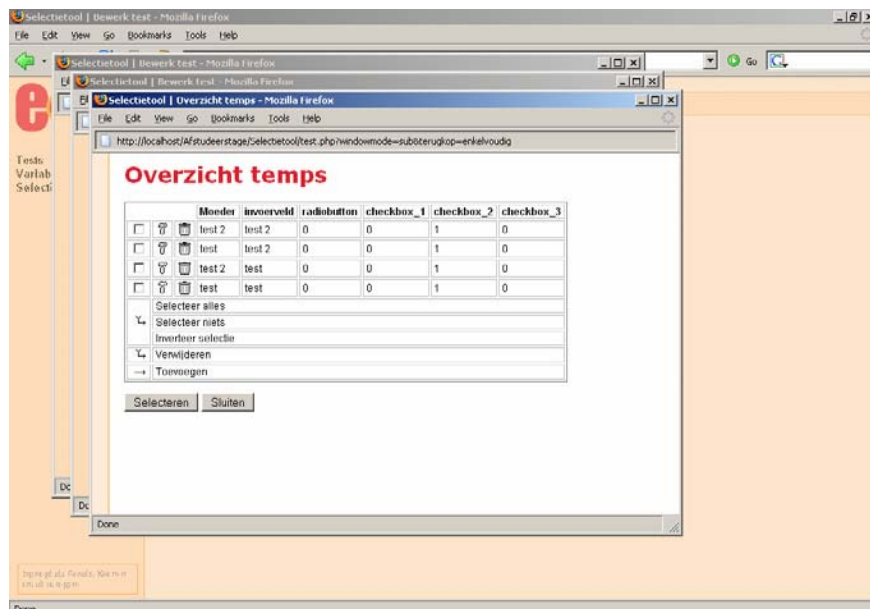
Figuur 13 Een detailscherm.

Subwindows

Objecten moeten aan elkaar gelinkt kunnen worden. Vanuit een detailscherm moet het mogelijk zijn een overzichtscherm (van een andere klasse) op te roepen, en daar een object uit te selecteren.

Ook op dit gebied is al ervaring aanwezig vanuit eerder uitgevoerde projecten. Er kon wat code herbruikt worden, ookal is deze wel grondig doorgenomen voor het consistent maken van functie- en variabelen-namen binnen de PHP- en JavaScript-code. Ook ter conformantie met het coderaamwerk.

Zie Figuur 14 voor een voorbeeld. Hier is vanuit het hoofdscherm een beroep gedaan op een subwindow voor het kiezen van een ander object. Vanuit dit subwindow is vervolgens weer hetzelfde gebeurd voor twee maal.

**Figuur 14** Een subwindow-situatie met meerdere subwindows die elk weer een ander subwindow hebben aangeroepen.

Database-basis t.b.v. inloggen

De databasebasis voor het inloggen is - inclusief de tabellen voor een basis aan het rechtensysteem - tijdens het opstellen van het pilotontwikkelpplan al ontworpen. Tijdens de activiteiten voor dit pilotdeel, was het alleen een kwestie van het als SQL-code uitvoeren van het relationeel implementatiemodel. Zie Fragment 10 voor de structuur van de tabel volgens MySQL.

```
mysql> DESCRIBE recht;
+-----+-----+-----+-----+-----+-----+
| Field          | Type                               | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | mediumint(8) unsigned             |      | PRI | 0        |       |
| gebruikersgroepId | tinyint(3) unsigned               |      | PRI | 0        |       |
| gebruikerId    | smallint(5) unsigned              |      | PRI | 0        |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Fragment 10 De structuur van de tabel recht volgens MySQL.

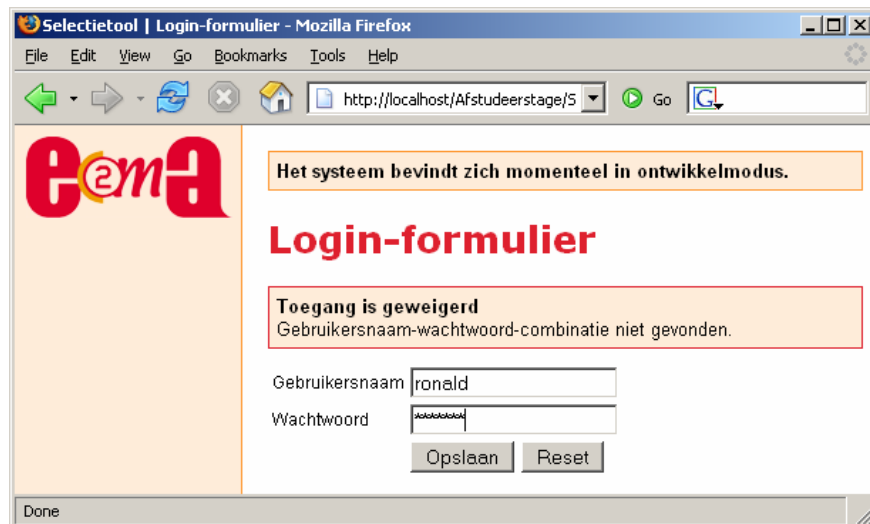
Het was in eerste instantie het idee om `NULL` toe te staan in de kolommen `gebruikersgroepId` en `gebruikerId`. Als er een bepaald recht wordt toegewezen aan een gebruiker, kan de kolom `gebruikersgroepId` hierop ingesteld worden. Het is - bleek later - onmogelijk dit soort kolommen op te nemen in een primaire sleutel. Er waren nu drie mogelijkheden:

- een extra numerieke kolom toevoegen als alternatieve primaire sleutel;
- niet toestaan dat er `NULL` in de kolommen komt te staan, met als gevolg dat de kolom `gebruikersgroepId` in bovenstaand voorbeeld een letterlijke 0 als waarde zou krijgen;
- helemaal geen primaire sleutel gebruiken.

Als we in beschouwing nemen dat een rij - ongeacht van het wel of niet aanwezig zijn van een primaire sleutel - altijd te identificeren is als het `id` en de `gebruikersgroep` of de `gebruiker` worden opgegeven, vervalt de eerste optie met voorkeur voor de derde. Maar de primaire sleutel is er om meer redenen buiten alleen identificatie van een rij, namelijk voor het afdwingen van uniciteit en voor het indexeren voor snelle toegang, dus hierom is uiteindelijk gekozen voor de tweede mogelijkheid.

Inloggen zelf

Voor elke aanvraag aan het systeem moet gecontroleerd worden of er is ingelogd. Indien dit niet het geval is wordt het inlogformulier weergegeven.



Figuur 15 Het inlogformulier.

5.8 Beoordelen en testen pilot

Tijdens het opstellen van het Plan van Aanpak van pilot #2, kwam ik erachter dat deze activiteit voor deze pilot nog niet was gedaan. Er waren nog geen testspecificaties, en de tests zelf konden hierdoor natuurlijk ook niet uitgevoerd worden. Ik heb ervoor gekozen per pilot geen afzonderlijk testplan te maken, maar dit als hoofdstuk eenvoudig op te nemen in het pilotontwikkelplan.

Na het opstellen van de scenario's, zijn ze allemaal zonder noemenswaardige fouten doorlopen, dus de pilot is als goed getest gekenmerkt. Alleen het automatisch e-mailen van eventueel opgetreden fouten verliep in eerste instantie in zijn geheel niet, maar dit was te wijten aan twee configuratie-instellingen van PHP.

6 Pilot #2, Variabelebeheer

In dit hoofdstuk staan de activiteiten beschreven ten behoeve van de tweede pilot. Deze pilot gaat in op de ontwikkeling van de mogelijkheden voor het beheren van variabelen. Variabelen moeten kunnen worden aangemaakt, bewerkt, verwijderd enzovoorts. Een variabele is in feite een antwoord of gegeven van een deelnemer. Voordat deze antwoorden in een selectie gebruikt kunnen worden, moeten ze bij de selectietool kenbaar worden gemaakt. Zie bijlage 12 voor het volledige ontwikkelplan.

Een speciaal soort variabele is de berekende variabele: met één of meer andere variabelen als invoer, kan deze middels een vrij op te geven stuk PHP-code worden omgewerkt tot uitvoer. Een berekende variabele kan op zijn beurt ook weer gebruik maken van andere berekende variabelen.

6.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld, zie bijlage 11 voor het plan.

Er is gekozen voor het laten vervallen van de organisatorische inrichting, omdat er voor deze pilot geen organisatorische aspecten te vermelden zijn: er zijn geen opleidingen en/of handleidingen behorende bij deze pilot.

Direct na deze activiteit ben ik erachter gekomen dat de productiehandleiding tijdens de vorige fase nog niet was gemaakt, dit is toen alsnog gebeurd. Een leermoment voor de volgende keer is om dit soort activiteiten in het vervolg directer op te nemen in de planning. De weg zoals die tot nu aan toe is gevolgd, is echter niet veranderd omdat er voor de volgende pilot geen productiehandleidingen meer zijn, en de wijzigingen te grote watervaleffecten hebben op voorgaande documenten.

6.2 SUMI-test

Voor het systeem is een SUMI-test opgesteld. Voor deze pilot zijn een aantal scenario's opgesteld die tijdens de fase 'Invoering' gebruikt zijn voor de SUMI-test zelf. Zie Fragment 11 voor één van de scenario's.

Er moet een nieuwe normale variabele worden toegevoegd. Voor het kiezen van de systeemnaam moet gebruik gemaakt worden van de wizard (achter de knop 'Kies') waarna er een willekeurige variabele gekozen moet worden.

Fragment 11 Één van de scenario's.

6.3 Globaal-functionele structuur

In de globaal-functionele structuur komen de functionaliteiten naar voren die in de pilot moeten worden ontworpen en gebouwd. Voor deze pilot zijn de functionaliteiten als volgt:

- het kunnen oproepen van een overzicht van alle variabelen;
- het kunnen toevoegen van een variabele;
- het kunnen bewerken van een variabele;
- het kunnen verwijderen van een variabele;
- het kunnen testen van een variabele;
- het kunnen valideren van de code van een berekende variabele;
- het kunnen zien van een kleurcodering voor de code van een berekende variabele.

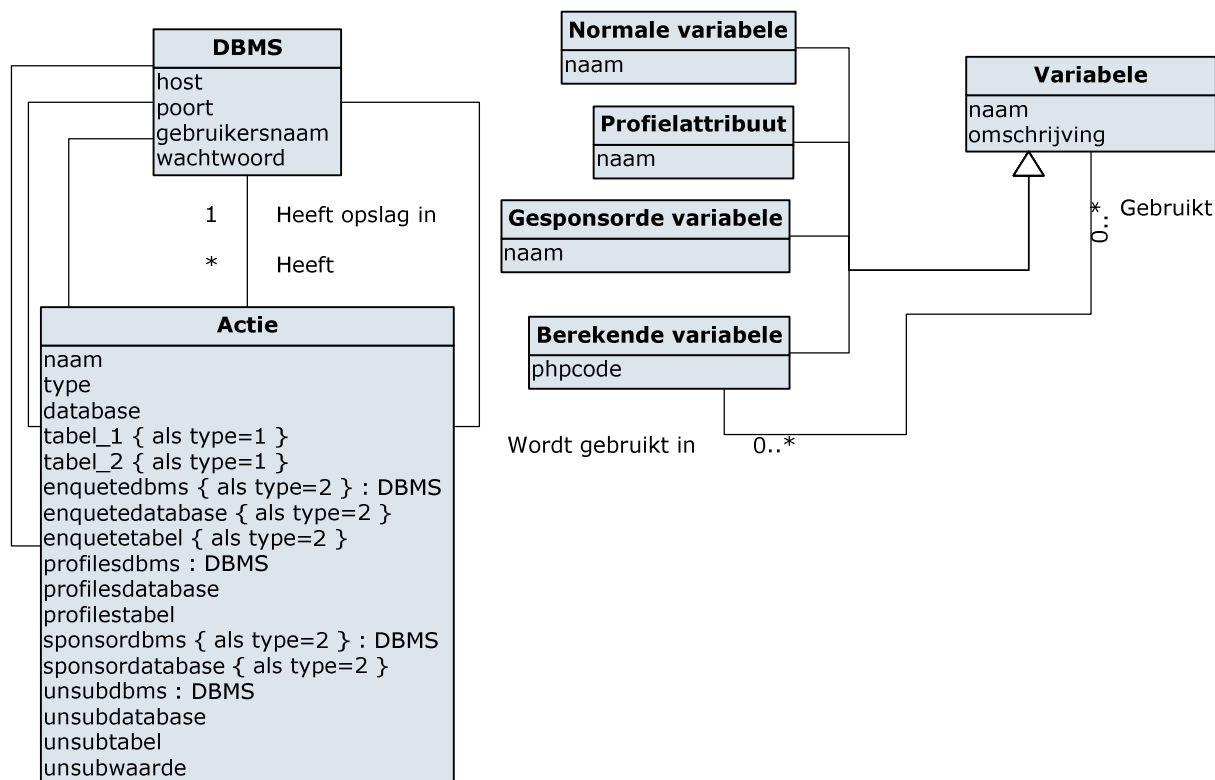
Globaal waren deze eisen al opgezet in de systeemeisen van de definitiestudie, maar op dit punt zijn ze verder tot in detail uitgewerkt. Er is bijvoorbeeld aangegeven welke gegevens van een variabele er in het overzicht naar voren moeten komen. Ik heb op dit punt tevens bedacht dat variabelen uit een F2-actie via een wizard te kiezen moeten zijn. Op basis van de vraaginformatie in de E2Ma-databases, kan na het kiezen van een actie, een lijst worden gegenereerd van alle bij deze actie gebruikte variabelen.

6.4 Globaal-technische structuur

Voor in de globaal-technische structuur zijn de voor deze pilot relevante klassen omgewerkt tot een relationeel representatiemodel en een relationeel implementatiemodel.

Databasestructuur

Voor wat betreft het omzetten van de relevante klassen is er eerst een keuze gemaakt voor welke klassen uit het klassendiagram in de definitiestudie er omgezet moeten gaan worden. Omdat deze pilot gaat over de variabelen, is allereerst gekozen voor de klassen 'Variabele', 'Normale variabele', 'Profielattribuut', 'Gesponsorde variabele', en 'Berekende variabele'. Daarnaast zijn de klassen 'DBMS' en 'Actie' ook omgezet omdat voor de normale variabelen een wizard voor het achterhalen van een databasenaam te starten is, en vanuit deze wizard moet verbinding gemaakt worden met een F2-actie.



Figuur 16 Deel van het klassendiagram ten behoeve van deze pilot.

Voor de omzetting van deze klassen in een relationeel implementatiemodel, zijn dezelfde werkzaamheden uitgevoerd zoals deze in §5.3 zijn beschreven.

Voor het omzetten van de klasse 'Variabele' met de bijbehorende subklasse waren verschillende mogelijkheden voorhanden. Er is voor gekozen om de attributen van de subklassen allemaal op te nemen in dezelfde relatie, omdat dit overzichtelijk is, omdat er een vergelijkbaar attribuut voorkomt bij een overgroot deel van de subtypen. Zie Fragment 12.

Initiël bestaan de volgende relaties:

```

variabele(naam, omschrijving)
var_norm(naam)
var_profiel(naam)
var_sponsor(naam)
var_berekend(naam, phpcode)
var_speciaal(naam, index)

```

Aan alle relaties wordt een extra numerieke primaire sleutel toegevoegd. Er bestaat tevens een veel-op-veel relatie tussen een berekende variabele en variabele wat wordt opgenomen als een afronderlijke relatie:

```

variabele(id, naam, omschrijving)
var_norm(id, variabeleId, naam)
var_profiel(id, variabeleId, naam)
var_sponsor(id, variabeleId, naam)
var_berekend(id, variabeleId, naam, phpcode)
var_speciaal(id, naam, index)
variabele_var_berekend(varBerekendId, variabeleId)

```

Maar: een variabele is in het klassendiagram gemodelleerd als een superklasse met vijf subklassen ter representatie van normale variabelen, profielattributen, gesponsorde variabelen, berekende variabelen en speciale variabelen. Er zijn verschillende manieren om dit om te zetten in een stukje relationeel representatiemodel, maar er is voor gekozen om alle objecten in één relatie te gieten en met een extra attribuut te bepalen van welk soort de relatie is:

```
variabele(id, type, systeemnaam, omschrijving, databasenaam,
         phpcode, speciaal)
variabele_variabele(varBerekendId, variabeleId)
```

Fragment 12 Relationeel representatiemodel voor klasse 'Variabele'.

Nu deze omzetting was voltooid, was het relationeel representatiemodel weer binnen handbereik. Wederom met SQL als DDL zijn kant en klare definities van de tabellen met attributen en relaties ontstaan. Zie Fragment 13 voor de SQL-CREATE-opdrachten voor het aanmaken van de tabellen `variabele` en `variabele_variabele`.

```
CREATE TABLE variabele
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    type ENUM('normaal', 'profiel', 'sponsor',
             'berekend', 'speciaal') NOT NULL,
    systeemnaam TINYTEXT,
    omschrijving TEXT,
    databasenaam TINYTEXT,
    phpcode TEXT,
    PRIMARY KEY (id)
);

CREATE TABLE variabele_variabele
(
    varBerekendId MEDIUMINT UNSIGNED NOT NULL,
    variabeleId MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (varBerekendId, variabeleId),
    FOREIGN KEY (varBerekendId)
        REFERENCES variabele ON DELETE CASCADE,
    FOREIGN KEY (variabeleId)
        REFERENCES variabele ON DELETE RESTRICT
);
```

Fragment 13 Een deel van het implementatiemodel.

6.5 GUI-ontwerp

In tegenstelling tot het ontwerpen van de schermen voor de vorige pilot, zijn de activiteiten in deze fase van minder groot belang. Met de schetsen uit deze eerste pilot is al een groot deel van het ontwerpen gedaan. Voor wat betreft deze pilot is het meer het invullen van de details geweest. De volgende schermen zijn geschetst:

- Overzichtscherf: overzicht van alle variabelen.
- Toevoeg- & bewerkscherf, met subschermen voor de wizard voor het kiezen van een databasenaam.
- Schermen voor het testen.

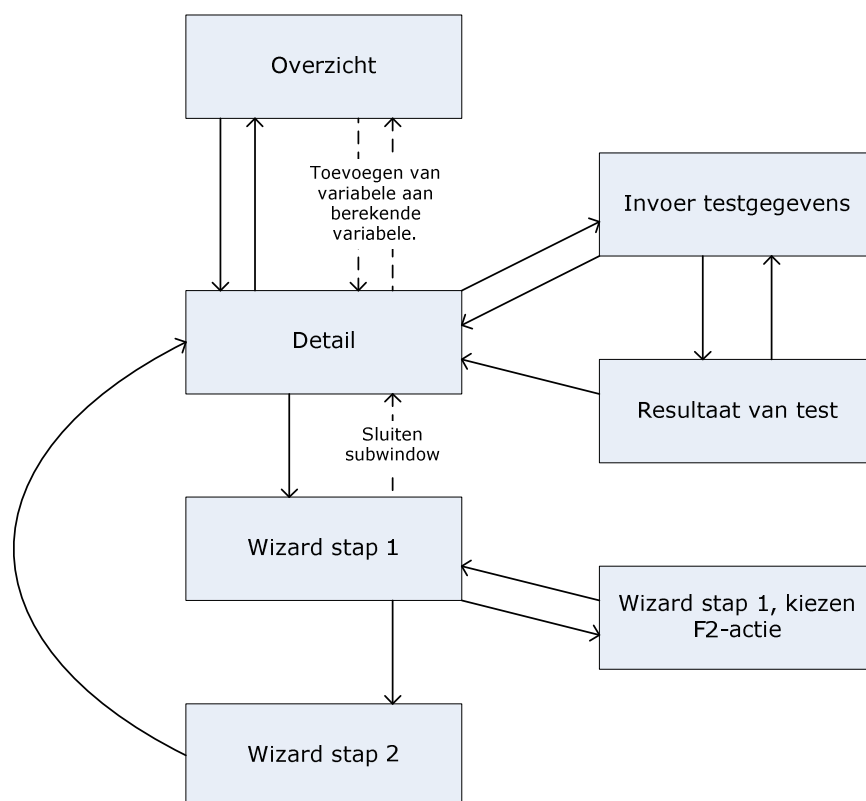
Zie Figuur 17 voor een van de schetsen.

eat test-actie

	Type	Systeemnaam	Databasenaam	Omschrijving
☐ B ✓	Normal	~~~~~	~~~~~	~~~~~
☐ B U	Bevelend	~~~~~	~~~~~	~~~~~
☐ B ✓	~~~~~	~~~~~	~~~~~	~~~~~
Y	Selecteer alles			
	Selecteer niets			
	Inverteer selectie			
Y	Verwijderen			
→	Toevoegen			

Figuur 17 Schets van het overzichtscherf.

Naast het schetsen van de schermen is er ook een navigatieschema gemaakt om de navigatiemogelijkheden tussen de schermen visueel weer te geven. Zie Figuur 18 voor dit navigatieschema.



Figuur 18 Het navigatieschema.

6.6 Testplan

Voor het testplan zijn weer een aantal testscenario's samengesteld en uitgelegd die verderop gebruikt kunnen worden voor het testen van de pilot. Voor de rest is het proces vergelijkbaar verlopen als beschreven in §5.6.

6.7 Ontwerp & bouw

In deze paragraaf worden alle werkzaamheden, aangaande het ontwerp en de bouw van de verschillende delen van deze pilot zelf, behandeld.

Overzicht

In dit deel is gebouwd aan een overzicht voor de variabelen. Als deze pagina wordt aangeroepen, genereert het systeem een tabel met hierin een lijst met alle bij het systeem bekende variabelen inclusief een aantal belangrijke attributen en mogelijkheden de bestaande variabelen te wijzigen, te verwijderen of om nieuwe variabelen toe te voegen. Zie Figuur 19 voor een voorbeeldoverzicht.

			Type	Naam	Databasenaam	Omschrijving
☐			Normaal	Leefomgeving	gz1	
☐			Normaal	verz1_1	verz1_1	
☐			Profiel	Achternaam	lname	
☐			Profiel	Email	email1	
☐			Profiel	Geboortejaar	byear	
☐			Profiel	Geboortemaand	bmonth	
☐			Berekend	Leeftijd		
☐			Speciaal	Actienaam		
☐			Speciaal	Antwoordtijd		
Selecteer alles						
	Selecteer niets					
	Inverteer selectie					
	Verwijderen					
	Toevoegen					

Figuur 19 Overzicht van variabelen.

Dankzij de eerste pilot was er al een voorbeeldoverzicht, dus was het een kwestie van het overnemen van de grote lijnen en het veranderen danwel invullen van de details.

☐			Normaal	verz1_1	verz1_1
☐			Profiel	Achternaam	lname
☐			Profiel	Email	email1
☐					
☐					
☐					
☐			Profiel	Huisnummer_1	hnr1
☐			Profiel	Huisnummer_2	hnr2
☐			Profiel	Postcode	pc1
☐			Profiel	Straat	street

Deze variabele kan niet verwijderd worden om de volgende reden(en):

- Er is/zijn 1 andere variabele(n) van deze variabele afhankelijk.

Er trad wel een klein luxeprobleem op. Er is van te voren al rekening mee gehouden dat het systeem van te voren van een variabele moet uitzoeken of het verwijderd mag worden of niet. Als een variabele in één of meer andere (berekende) variabelen is gebruikt, mag dit namelijk niet gebeuren. Om dat aan de gebruiker duidelijk te maken, heb ik voorheen altijd gebruik gemaakt van een simpele 'tooltip' via het `title` attribuut van HTML's ankerelement. Het bleek echter dat de browser deze op een zeker lengte afkapte, dus heb ik een mechanisme ontwikkeld voor het kunnen weergeven van grotere tooltips. Enkel en alleen voor dit doel zonde van het werk, maar door het te voor andere doeleinden te herbruiken, is het de investering waard gebleken.

Toevoegen

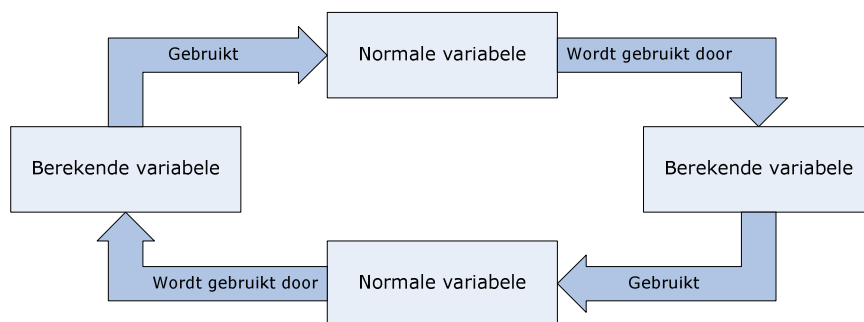
Ook voor wat betreft het toevoegen van een variabele was het voor een deel een kwestie van herbruiken. Het detailscherm en de wizard zijn zonder problemen gebouwd. Er is overigens van JavaScript gebruik gemaakt om attributen die voor bepaalde typen niet van belang zijn, te verbergen, zie Figuur 20 voor een indruk hiervan.

The screenshot shows a web interface for managing variables. It includes the following elements:

- Type:** A dropdown menu currently showing 'Berekend'.
- Systeemnaam:** A text input field.
- Variabelnaam:** A text input field with a date and time placeholder 'Datum en tijd van antwoord'.
- Databasenaam:** A text input field.
- PHP-code:** A large text area for entering code.
- Action Buttons:**
 - For 'Variabelnaam': 'Opslaan', 'Reset', 'Verwijderen', 'Invoegen'.
 - For 'Databasenaam': 'Opslaan', 'Reset', 'Kies'.
 - At the bottom: 'Opslaan', 'Reset'.

Figuur 20 De weergegeven velden van alle typen tegelijk weergegeven.

Wel ontstond er een klein probleem met de omgang met het recursieve karakter van variabelen, en dan met name de detectie van kring- en/of kruisverwijzingen. Zie Figuur 21 voor een illustratie voor een mogelijke kringverwijzing tussen variabelen. Het probleem ontstond in relatie met het gegeven dat deze detectie hybride moest werken voor zowel deze toevoeg- als de bewerkfunctionaliteiten. Na wat uitproberen en goed nadenken, kon het probleem gelukkig goed opgelost worden. Er wordt overigens niet op een ingewikkelde manier bijgehouden welke 'paden' tussen variabelen er worden bewandeld en welke patronen hierin ontstaan, maar er wordt simpelweg gekeken naar het recursieniveau. Indien deze uitkomt op meer dan 25, is het waarschijnlijk dat dit veroorzaakt is door een kringverwijzing, en wordt hiervan een waarschuwing aan de gebruiker gegeven.



Figuur 21 Een kringverwijzing tussen normale en berekende variabelen. Er kan overigens ook een kringverwijzing ontstaan met een berekende variabele die door zichzelf wordt gebruikt.

Bewerken

De werkzaamheden voor het maken van het bewerk-scherm stelden niets voor. Tijdens pilot #1 zijn hier al voorbereidingen voor geweest, en het enige wat hier feitelijk is gebeurd, is het op een andere manier aanroepen van het detailscherm en het schrijven van een paar regels code die zorgdragen voor opslag in de database.

Verwijderen

Tijdens het genereren van het overzicht is al een controle ingebouwd op of een variabele verwijderd mag worden. Toch is deze controle nog een keer extra geïmplementeerd, omdat het overzicht slechts een momentopname is.

Overige

De pilotdelen met betrekking tot het testen (van een variabele), code-validatie en code-kleurcoderingen zijn niet aan bod gekomen. Op het moment dat het pilotdeel voor het verwijderen af was, is er in eerste instantie voor gekozen deze pilotdelen met een comfort- of een luxe-prioriteit even te bewaren voor als er op een later tijdstip weer tijd voor vrijkomt. Achteraf is deze tijd echter nooit vrijgekomen, dus zodoende zijn deze delen blijven liggen. Zie ook de procesevaluatie in hoofdstuk 12.

6.8 Beoordelen en testen pilot

Alle scenario's zijn met goed gevolg doorlopen, dus de pilot is als goed getest gekenmerkt.

7 Pilot #3, Selectiebeheer

In dit hoofdstuk staan de activiteiten beschreven ten behoeve van de tweede pilot. Deze pilot gaat in op de ontwikkeling van mogelijkheden voor het beheren van selecties. Selecties moeten kunnen worden aangemaakt, bewerkt, verwijderd, enzovoorts. Een selectie is de specificatie voor welke data er door het systeem geselecteerd moeten worden. Zie bijlage 14 voor het volledige ontwikkelplan.

7.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld, zie bijlage 13 voor het plan.

Er is weer voor gekozen om de IAD-activiteit 'Globaal organisatorische inrichting specificeren' over te slaan, omdat er voor deze pilot geen organisatorische aspecten te vermelden zijn: er zijn geen opleidingen en/of handleiding behorende bij deze pilot.

Met inmiddels al twee voorgaande plannen van aanpak van de voorgaande twee pilots, was er voldoende herbruikbaar materiaal aanwezig om deze activiteit vlot te kunnen verrichten.

7.2 SUMI-test

Voor het systeem is een SUMI-test opgesteld. Voor deze pilot zijn een aantal scenario's opgesteld die tijdens de fase 'Invoering' gebruikt zijn voor de SUMI-test zelf. Zie Fragment 14 voor één van de scenario's.

Er moet een nieuwe selectie worden toegevoegd met drie voorwaarden die aan elkaar gekoppeld zijn middels een logische 'en'.

Fragment 14 Één van de scenario's.

7.3 Globaal-functionele structuur

In de globaal-functionele structuur komen de functionaliteiten naar voren die in de pilot zijn ontworpen en gebouwd. Voor deze pilot waren de functionaliteiten als volgt:

- het kunnen oproepen van een overzicht van alle selecties;
- het kunnen toevoegen van een selectie;
- het kunnen bewerken van een selectie;
- het kunnen activeren van een selectie;
- het kunnen inplannen van een selectie;
- het kunnen doorgeven van een selectie.

Globaal waren deze eisen al opgezet in de systeemeisen van de definitiestudie, maar op dit punt zijn ze verder tot in detail uitgewerkt.

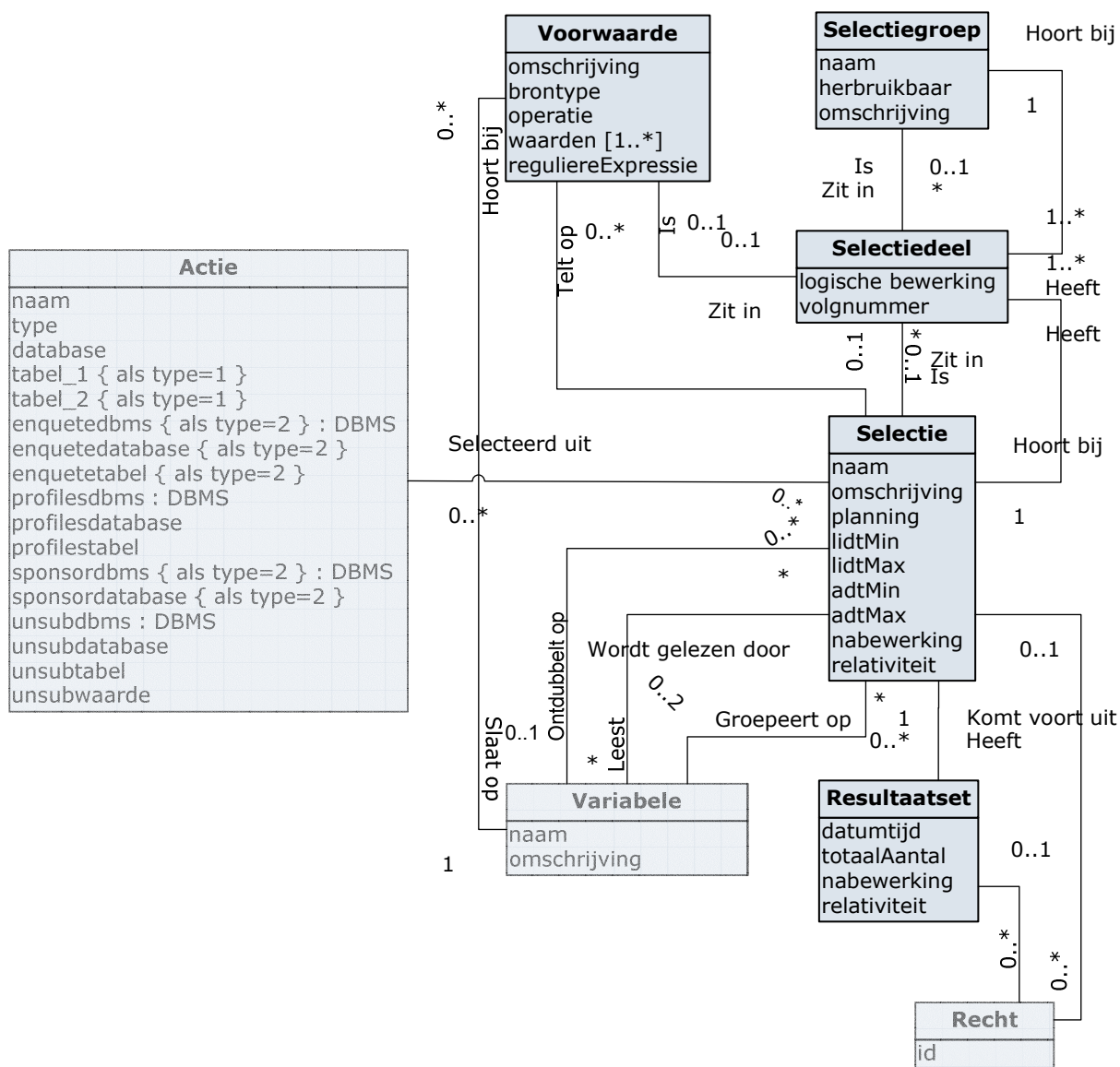
Er is bijvoorbeeld aangegeven welke gegevens van een selectie er in het overzicht naar voren moeten komen.

7.4 Globaal-technische structuur

Voor in de globaal-technische structuur zijn de voor deze pilot relevante klassen omgewerkt tot een relationeel representatiemodel en een relationeel een implementatiemodel.

Databasestructuur

Voor wat betreft het omzetten van de relevante klassen is er eerst een keuze gemaakt voor welke klassen uit het klassendiagram in de definitiestudie er omgezet moeten gaan worden. Omdat deze pilot over selecties gaat, is er gekozen voor de klassen 'Selectie', 'Selectiedeel', 'Selectiegroep', 'Voorwaarde' en 'Resultaatset'. Voor de duidelijkheid



Figuur 22 Deel van het klassendiagram ten behoeve van deze pilot.

zijn de reeds omgezette klassen 'Actie', 'Variabele' en 'Recht' ook opgenomen in het klassendiagram relevant voor dit project om wat extra context te geven of omdat er relaties mee gelegd moeten worden. Zie Figuur 22.

Voor de omzetting van deze klassen in een relationeel implementatiemodel, zijn dezelfde werkzaamheden uitgevoerd zoals deze in §5.3 zijn beschreven, behalve voor de klasse 'Recht'. Deze klasse bestond al en er hoefde alleen een extra relatie gelegd te worden met de klasse 'Resultaatset'. De entiteit is hiervoor overgenomen uit het relationeel representatiemodel van de voorgaande pilot en uitgebreid met een extra vreemde sleutel.

Zie Fragment 15 voor een deel van het representatiemodel van de klasse 'Selectiedeel'. Alle vreemde sleutels beginnend met *bron* slaan op waar het selectiedeel een onderdeel van is: een selectie of een selectiegroep. Alle vreemde sleutels beginnend met *doel* slaan op wat het selectiedeel is: een selectie of een selectiegroep.

Initieel bestaat de volgende relatie:

```
selectiedeel(bronSelectieId, bronSelectiegroepId,
             logischeBewerking, volgnummer, doelVoorwaardeId, doelSelectieId,
             doelSelectiegroepId)
```

bronSelectieId en *bronSelectiegroepId* sluiten elkaar uit, en geven respectievelijk aan bij welke selectie danwel selectiegroep het deel hoort.

doelVoorwaardeId, *doelSelectieId* en *doelSelectiegroepId* sluiten elkaar ook uit, en geven respectievelijk aan of het deel een voorwaarde, een selectie of een selectiegroep is.

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
selectiedeel(id, bronSelectieId, bronSelectiegroepId,
             logischeBewerking, volgnummer, doelVoorwaardeId, doelSelectieId,
             doelSelectiegroepId)
```

- *bronSelectieId*: vreemde sleutel, verwijst naar *id* in *selectie*, null toegelaten.
- *bronSelectiegroepId*: vreemde sleutel, verwijst naar *id* in *selectiegroep*, null toegelaten.
- *doelVoorwaardeId*: vreemde sleutel, verwijst naar *id* in *voorwaarde*, null toegelaten.
- *doelSelectieId*: vreemde sleutel, verwijst naar *id* in *selectie*, null toegelaten.
- *doelSelectiegroepId*: vreemde sleutel, verwijst naar *id* in *selectiegroep*, null toegelaten.

Fragment 15 Een deel van het representatiemodel van de klasse 'Selectiedeel'.

Nadat de klassen en attributen weer waren omgezet in een relationeel representatiemodel, was de weg weer vrij voor het maken van het implementatiemodel. Ook dit is net als de vorige keer weer op routine gegaan en zonder problemen verlopen. Zie Fragment 16 voor een deel van het implementatiemodel.

```
CREATE TABLE selectiedeel
```

```
(
  id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
  bronSelectieId MEDIUMINT UNSIGNED,
  bronSelectiegroepId MEDIUMINT UNSIGNED,
  logischeBewerking ENUM('en', 'of', 'xof', 'nen',
    'nof', 'nxof'),
  volgnummer SMALLINT UNSIGNED NOT NULL,
  doelVoorwaardeId MEDIUMINT UNSIGNED,
  doelSelectieId MEDIUMINT UNSIGNED,
  doelSelectiegroepId MEDIUMINT UNSIGNED,
  PRIMARY KEY (id),
  FOREIGN KEY (bronSelectieId)
    REFERENCES selectie ON DELETE CASCADE,
  FOREIGN KEY (bronSelectiegroepId)
    REFERENCES selectiegroep ON DELETE CASCADE,
  FOREIGN KEY (doelVoorwaardeId)
    REFERENCES voorwaarde ON DELETE RESTRICT,
  FOREIGN KEY (doelSelectieId)
    REFERENCES selectie ON DELETE RESTRICT,
  FOREIGN KEY (doelSelectiegroepId)
    REFERENCES selectiegroep ON DELETE RESTRICT
);
```

Fragment 16 Een deel van het implementatiemodel.

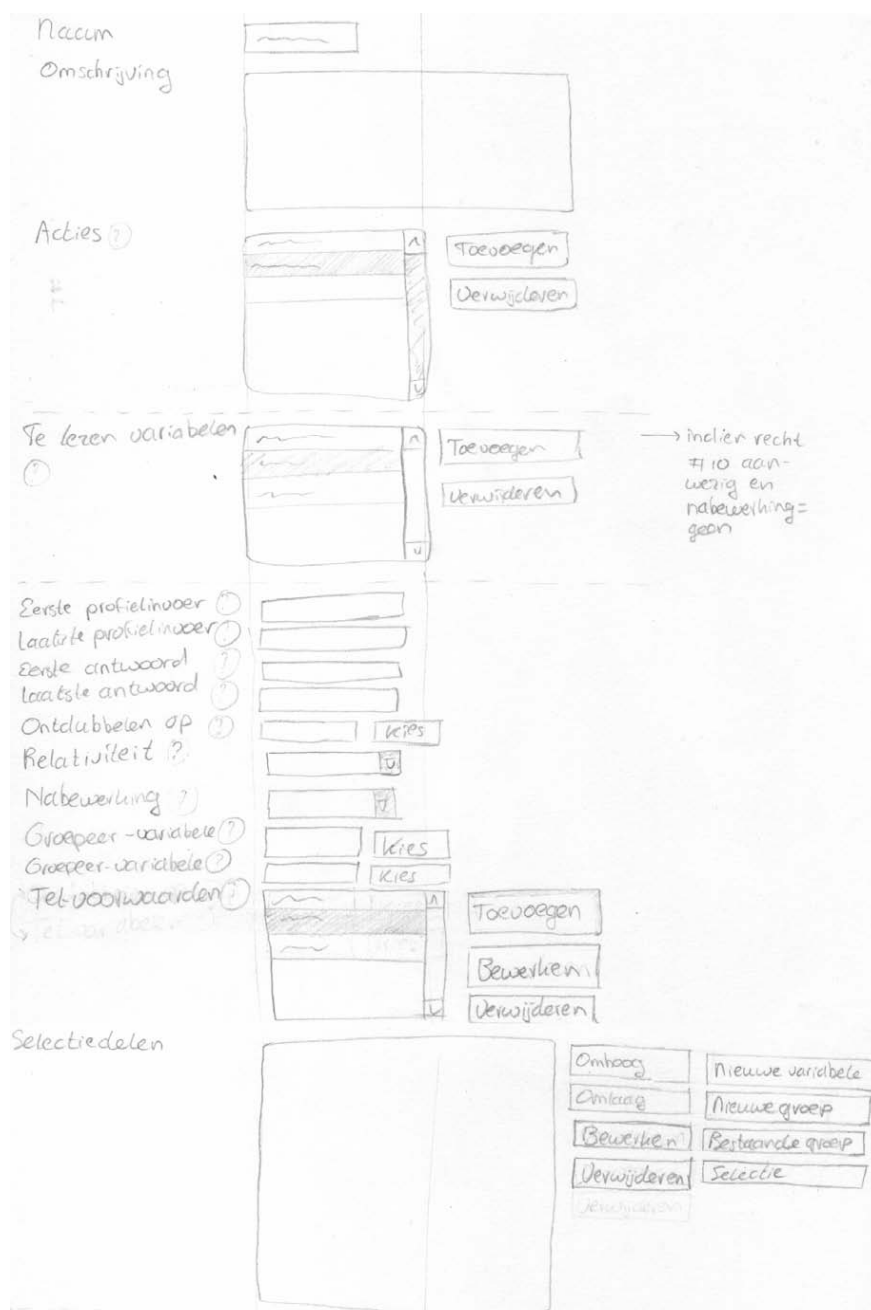
7.5 GUI-ontwerp

De volgende schermen zijn geschetst:

- Overzichtscherm: overzicht van alle selectie.
- Toevoeg- & bewerk scherm.
 - Overzichtscherm acties.
 - Overzichtscherm herbruikbare groepen
 - Subwindow selectiedeel
- Doorgeven
- Overzichtscherm gebruikers
- Inplannen

De grafische interface voor deze pilot is een van de meest uitdagende stukjes geweest. Er moet aan de ene kant de mogelijkheid bestaan tot het specificeren van een complexe selectie, maar aan de andere kant moet de gebruikersvriendelijkheid toch zo hoog mogelijk gehouden worden. Ik ondervond met name problemen rond de recursiviteit van selectiedelen. Het maken van de schetsen of het uitdenken van een interface is in eerste instantie niet zo lastig, maar tegelijkertijd is er de hele tijd ook gedacht aan de technische onmogelijk- of moeilijkheden wat enige beperkingen oplegde.

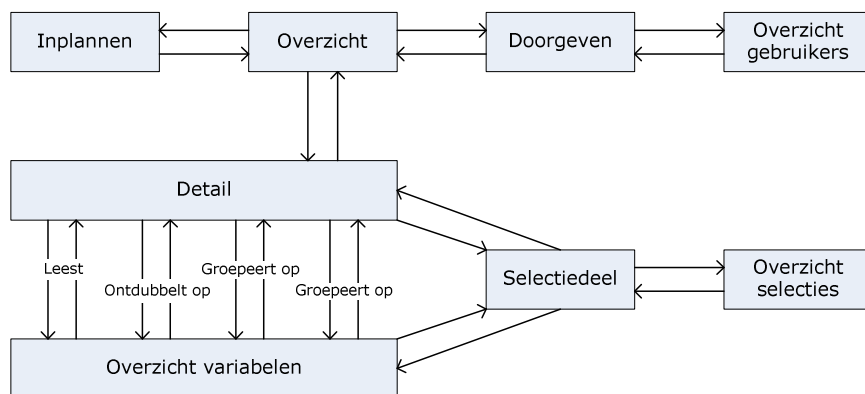
Wat betreft deze delen was ik in eerste instantie van plan gebruik te maken van de kracht van subwindows. Hier ben ik echter vanaf gestapt met de gebruikersvriendelijkheid als argument. Ik heb er toen voor gekozen alle selectiedelen van een selectie onder elkaar te plaatsen in een kieslijst en door middel van inspringen het niveau van een selectiedeel te tonen. Zie Figuur 23 voor de schets van het detailscherm.



Figuur 23 Het detailscherm voor het bewerken van een selectie.

Tijdens het maken van de schetsen is nog een klein foutje ontdekt het klassendiagram voor wat betreft de associatie tussen een selectie en voorwaarden. Dit was eerst een één-op-veel-relatie, maar een voorwaarde kan - als hij bij een selectie als telvoorwaarde hoort en niet bestaat als selectiedeel - maar bij één selectie tegelijkertijd horen. Dit heeft een kleine waternal van wijzigingen in documentatie, representatie- en implementatiemodellen voortgebracht maar gelukkig niet met veel impact.

Naast het schetsen van de schermen, is er ook een navigatieschema gemaakt om de navigatiemogelijkheden tussen de schermen visueel weer te geven.



Figuur 24 Het navigatieschema.

7.6 Testplan

Voor het testplan zijn weer een aantal testscenario's samengesteld en uitgelegd die verderop gebruikt kunnen worden voor het testen van de pilot. Voor de rest is het proces vergelijkbaar verlopen als beschreven in §5.6.

7.7 Ontwerp en bouw

In deze paragraaf worden alle werkzaamheden, aangaande het ontwerp en de bouw van de verschillende delen van deze pilot zelf, behandeld.

Overzicht

In dit deel is gebouwd aan een overzicht voor de selecties. Als deze pagina wordt aangeroepen, genereert het systeem een tabel met hierin een lijst met alle bij het systeem bekende variabelen inclusief een aantal belangrijke attributen en mogelijkheden de bestaande selecties te wijzigen, verwijderen of om nieuwe selecties toe te voegen.

Toevoegen

Het detailscherm voor de selecties bestaat uit verschillende onderdelen die op verschillende vlakken met elkaar samenwerken. Hoofdzakelijk kunnen we het scherm onderverdelen in een scherm voor een selectie (deel I) en een scherm voor een selectiedeel (deel II, opent in een subwindow). Het scherm voor een selectiedeel wordt vervolgens vanuit twee punten gebruikt: voor het toevoegen van selectiedelen, of voor het toevoegen van telvoorwaarden (een selectiedeel van het type voorwaarde).

Het bouwen verliep in eerste instantie vlot, maar gedurende het proces kwamen er problemen bovendrijven. Allereerst bleken de functionaliteiten voor het veranderen van de volgorde van selectiedelen in combinatie met het hiërarchische karakter ervan erg lastig te

maken. Er kunnen zoals het geval is bij een platte lijst, niet zomaar twee elementen verwisseld worden, want als er een groep verplaatst moet worden, moeten alle elementen uit deze groep tegelijkertijd meeverplaatst worden. Tegelijkertijd mag een groep ook niet verder naar boven of naar beneden verplaatst worden dan dat er (in een bovenliggende groep) plaats voor is. Om nog maar te zwijgen wat er moet gebeuren als een groep één positie omhoog moet worden bewogen terwijl het bovenliggende element zelf ook een groep is.

Er is voor gekozen toch door te zetten en er dan maar extra veel tijd aan kwijt te zijn. Er was namelijk ook nog een andere aanpak mogelijk die bij het schetsen in eerste instantie naar boven is gekomen: elke selectiegroep bekijken vanuit een individueel subwindow. Achteraf gezien weet ik niet of dit de juiste keuze is geweest.

Een tweede probleem dat naar boven kwam zat in de communicatie tussen deel I en PHP. In deze communicatie dient plaats te zijn voor hiërarchie, maar deze was maar tot op vier niveaus aanwezig in de methode die tot dan aan toe is gebruikt. Er zat voor dit probleem weinig anders op dan zelf een nieuwe methode te bedenken om variabelen met een hiërarchie veilig over te geven vanuit JavaScript aan PHP en weer terug, al dan niet via een HTML-formulerveld. Hieruit is een stukje functionaliteit gerold voor het serialiseren en deserialiseren van een willekeurige variabele. Bij het serialiseren wordt de variabele ontleed en geconverteerd tot een simpele tekenreeks. In deze tekenreeks zit voor het deserialiseerproces dan weer informatie verweven voor het weer opbouwen van de in eerste instantie geserialiseerde variabele. Nadat deze twee functies met exact dezelfde semantiek waren geïmplementeerd in zowel JavaScript als PHP, was de nieuwe methode compleet.

De communicatie tussen deel I en deel II verliep al goed met gebruik van de oude methode, maar met de nieuwe serialiseermethode is dit ook gelijk aangepast om onduidelijkheid te voorkomen.

Net als bij de vorige pilot, ontstond er een klein probleem met de omgang met het recursieve karakter van de selectiedelen. Het probleem kon hier echter niet als bij de variabelen neergehaald worden, dus is besloten de detectie hiervoor te verplaatsen naar de selectie-activatie.

Bewerken

De werkzaamheden voor het maken voor het bewerk-scherm stelden - relatief gezien aan de problemen bij het maken van het toevoeg-scherm - weinig voor. Wederom hoefde er alleen code geschreven te worden voor het ophalen van een selectie uit en bijwerken van een selectie in de database.

Verwijderen

Door de complexe structuur van een selectie, is er veel aandacht besteed worden aan de afhankelijkheden en andere relaties die bij het verwijderen meeverwijderd moeten worden, of die er juist voor zorgen dat het verwijderen niet mogelijk is.

Overige

De pilotdelen met betrekking tot het doorgeven en inplannen zijn niet aan bod gekomen. Op het moment dat het pilotdeel voor het verwijderen af was, is er voor gekozen deze pilotdelen met een comfort- of luxe-prioriteit te bewaren voor als er op een later tijdstip eventueel weer tijd voor zou vrijkomen. Achteraf is deze tijd nooit vrijgekomen, dus zodoende zijn deze delen blijven liggen. Zie ook de procesevaluatie in hoofdstuk 12.

7.8 Beoordelen en testen pilot

Alle scenario's zijn met goed gevolg doorlopen, dus de pilot is als goed getest gekenmerkt.

8 Pilot #4, Selectieactivatie

In dit hoofdstuk staan de activiteiten beschreven ten behoeve van de tweede pilot. Deze pilot gaat in op de ontwikkeling van code voor het activeren van het proces. Dit is eigenlijk waar het allemaal om draait: een selectiespecificatie wordt gebruikt om samen met de data in de databases te komen tot uitvoer. Zie bijlage 16 voor het volledige ontwikkelplan.

8.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld, zie bijlage 15 voor het plan.

Er is gekozen voor het laten vervallen van de organisatorische inrichting, omdat er voor deze pilot geen organisatorische aspecten te vermelden zijn: er zijn geen opleidingen en/of handleidingen behorende bij deze pilot. Ook zullen de activiteiten 'SUMI-test opstellen' en 'GUI-ontwerp' worden overgeslagen omdat er geen of slechts een zeer beperkte GUI bestaat.

8.2 Globaal-functionele structuur

Tijdens het bepalen van de globaal-functionele structuur komen de functionaliteiten naar boven die in de pilot moeten worden ontworpen en gebouwd. Voor deze pilot zijn de functionaliteiten heel simpel te beschrijven: het daadwerkelijk selecteren van data uit de databases met inachtneming van de opgegeven specificaties en voorwaarden. Daarbinnen moet data kunnen worden nabewerkt en opgeslagen en moet de gebruiker tijdens het proces continue op de hoogte worden gehouden van de voortgang.

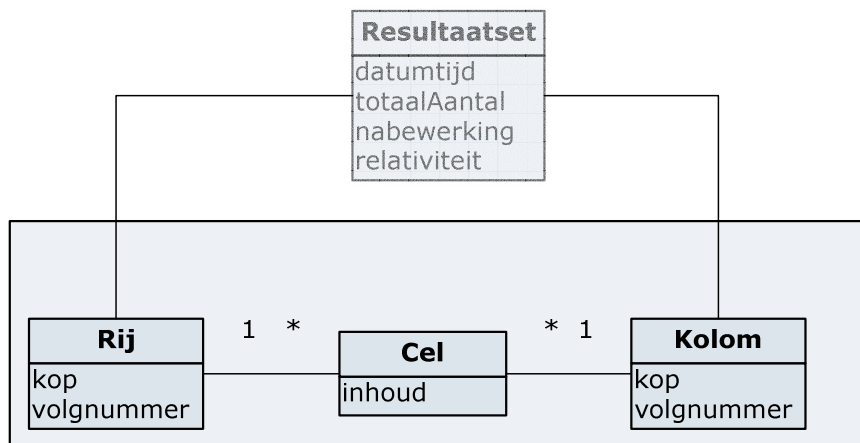
8.3 Globaal-technische structuur

Tijdens het bepalen van de globaal-technische structuur, zijn de voor deze pilot relevante klassen omgewerkt tot een relationeel representatiemodel en een relationeel implementatiemodel.

Databasestructuur

Voor wat betreft het omzetten van de relevante klassen is er eerst een keuze gemaakt welke klassen uit het klassendiagram in de definitiestudie er omgezet moeten gaan worden. De klassen 'Rij', 'Cel', 'Kolom' waren als enige nog over en dat klopte ook. De klasse 'Resultaatset' waarvan je zou denken dat dit een belangrijke is vanwege de opslag van het resultaat, hoefde hier echter niet meer

omgezet te worden, omdat dit al gebeurd is tijdens de vorige pilot. Zie Figuur 25.



Figuur 25 Deel van het klassendiagram ten behoeve van deze pilot.

Voor de omzetting van deze klassen in een relationeel implementatiemodel, zijn dezelfde werkzaamheden uitgevoerd zoals deze in §5.3 zijn beschreven.

Proces

Het uitdenken en het noteren van het uiteindelijke proces is een belangrijke activiteit geweest als voorbereiding voor de bouw van deze pilot. Het proces bestaat uit de daadwerkelijke stappen, iteraties en beslissingen die uitgevoerd moeten worden om van een selectie te komen tot een resultaatset.

Veel van het denkwerk was al gebeurd tijdens het ontwikkelen van het klassendiagram tijdens de fase Definitiestudie en tijdens activiteiten tussendoor. Hierdoor is veel inzicht verkregen in hoe de zaken geïmplementeerd zouden moeten of kunnen worden en heeft de materie in het algemeen ook veel makkelijker beet te pakken gemaakt.

Voor in de technische structuur is het proces wat uitgedacht is tijdens de fase Definitiestudie als basis genomen voor een volledige herziening ervan. Naar aanleiding van het beschrijven, ontwerpen en bouwen van de voorgaande pilots is sowieso op sommige punten op andere inzichten gekomen. Zie Fragment 17 voor een deel van het oude proces.

- Alle gekoppelde acties doorlopen, per actie:
 - Alle deelnemers doorlopen, per deelnemer:
 - Alle gewenste profile-attributen opslaan.
 - Alle gekoppelde acties doorlopen, per actie:
 - Controleren of de deelnemer ook heeft meegedaan aan de actie:
 - F1: Aantal opslagtabellen:
 - 1: Kijk of er een rij in de opslagtabel staat.
 - 2: Kijk of er een rij in beide opslagtabellen staat.

- F2: Alle variabelen langsgaan:
 - Kijken of er in minimaal één van deze tabellen een rij staat.

Fragment 17 Een deel van het oude proces.

Voor het nieuwere proces (zie Fragment 18 voor een deel) zijn niet alleen alle stappen herzien, herafgewogen en opnieuw geformuleerd, er is ook een sectie met een beschrijving van de datastructuren die tijdens het proces gebruikt zullen worden.

- Originele data selecteren: alle gekoppelde acties dooropen, per actie: alle deelnemers doorlopen (F1: de eerste opslagtabel, F2: de tabel partnerid) met inachtneming van adtMin en adtMax. Opslag van data gebeurd in structuur #2. Per deelnemer:
 - Alle gewenste profielattributen opslaan met inachtneming van lidtMin en lidtMax. Als het profiel niet bestaat of als de datum van laatste activiteit buiten bereik ligt breken we de iteratie af.
 - Controleren of het e-mailadres van de deelnemer niet voorkomt in de unsub-tabel. Als hij wel voorkomt breken we de iteratie af.
 - Alle gewenste normale variabelen achterhalen.
 - Alle gewenste sponsorvariabelen achterhalen.
- Tijdsontdubbeling, structuurmutatie 2.1.

Fragment 18 Een deel van het nieuwere proces.

8.4 Testplan

Voor het testplan zijn weer een aantal testscenario's samengesteld en uitgelegd die verderop gebruikt kunnen worden voor het testen van de pilot. Voor de rest is het proces vergelijkbaar verlopen als beschreven in §5.6.

8.5 Ontwerp en bouw

In deze paragraaf worden alle werkzaamheden, aangaande het ontwerp en bouw van de verschillende delen van deze pilot zelf, behandeld.

Stappen

Voor het makkelijker kunnen begrijpen van de uitgevoerde werkzaamheden, volgen hier de hoofdstappen van het proces:

- Selectiespecificaties ophalen.
- Originele data selecteren: alle data die het systeem ook maar denkt nodig te gaan hebben, selecteren.
- Tijdsontdubbeling: zorgen dat er van elke deelnemer maar één antwoord per variabele bekend is. Nieuw gaat hierbij boven oud.
- Berekende variabelen berekenen.
- Voorwaarden toepassen.
- Eventueel extra ontdebellen.
- Nabewerken: groeperen, tellen, groeperen en tellen of alleen het aantal weergeven.

- Opslaan: resultaatset opslaan in de database voor latere oproep.

Deze stappen komen overigens in zijn geheel niet overeen met de pilotdelen die in het pilotplan zijn opgesteld, maar de tijdsbelasting zou ongeveer evenredig moeten blijven. Er is voor gekozen het pilotplan zo te houden zoals hij is, omdat de indeling toch maar voorlopig was en het zijn doel al heeft gediend bij het opstellen van de planning.

In onderstaande paragrafen geef ik een omschrijving van de uitgevoerde bouwwerkzaamheden.

Selectiespecificaties ophalen

Het ophalen van de selectiespecificaties houdt in dat de informatie waar een selectie uit bestaat (welke er via de GUI van de vorige pilot is ingezet) wordt opgehaald. Er was al code aanwezig welke voor pilot #3 in staat was informatie omtrent een selectie op te halen, maar deze was niet helemaal geschikt voor dit doeleinde. Ik had dus de keuze tussen: deze functies als basis gebruiken voor een geheel nieuwe set functies voor het ophalen van de specificaties voor dit proces, óf de bestaande functies enkel aanpassen met het risico dat de integratie later onduidelijkheden laat ontstaan.

Er is voor gekozen de huidige functies aan te passen. Middels een extra argument in de functieaanroep, wist de code erachter dat de informatie bestemd was voor het proces, en niet voor de GUI voor het bewerken van een selectie. Hiervoor is gekozen omdat er al veel werk in deze bestaande functies zat, en 90% prima direct te gebruiken was. Er is voor gezorgd dat uitbreidingen op deze functies goed gedocumenteerd zijn om onduidelijkheden in de toekomst te voorkomen.

Originele data selecteren

Het selecteren van de originele data houdt in dat alle gegevens van alle kandidaat-deelnemers in het systeem worden ingeladen. Dit deel was snel geschreven en spoedig was de code al zover om de eerste gegevens te kunnen inladen.

Er is op dit punt een eerste significante wijziging geweest op het selectieproces. Binnen E2Ma bestaat een grote tabel met hierin een aantal honderdduizenden rijen voor het vasthouden van de algemene deelnemerinformatie als geboortjaar, leeftijd, woonplaats, en dergelijke. Het plan was in eerste instantie zo dat deze hele tabel per gekoppelde actie in zijn geheel doorlopen zou moeten worden. Per rij die erin voorkomt moet dan worden gekeken of de deelnemer ook daadwerkelijk heeft meegedaan aan de desbetreffende actie. Voor het selecteren van data uit alle acties, zou dit betekenen dat het systeem van 6,5 miljoen deelnemers moet nagaan of hij heeft meegedaan aan een bepaalde actie. Met een snelheid van ± 0.003 seconden per

controle zou zo'n selectie per definitie al ruim vijf uur duren. En dat is te lang omdat de huidige methode veel minder tijd in beslag neemt.

De oplossing lag in het omdraaien van de rollen. Na navraag bleek dat er voor beide technieken een tabel bestaat waarin sleutels van deelnemer die aan die actie hebben meegedaan, staan. Door vanuit deze tabel te kijken naar de tabel met deelnemerinformatie in plaats van andersom, wordt al flink wat tijds winst verkregen. De procesbeschrijving op papier is hierop aangepast, inclusief de al geschreven code. De snelheid waarmee gegevens opgehaald worden is met deze optimalisatie niet toegenomen, maar het aantal deelnemers per actie is alleen veel lager. Er wordt van rond de 25-75 deelnemers per seconde informatie opgehaald.

Tijdsontdubbeling

Voor één enkele variabele bestaat de mogelijkheid dat eenzelfde deelnemer hier meerdere keren waarde aan heeft gegeven. Al dan niet via twee verschillende acties op dezelfde dag, of na een aantal maanden via dezelfde actie. Voor dit deel van het proces is code geschreven die deze dubbele antwoorden eruit haalt kijkend naar wat het meest recentste antwoord is.

Berekende variabelen berekenen

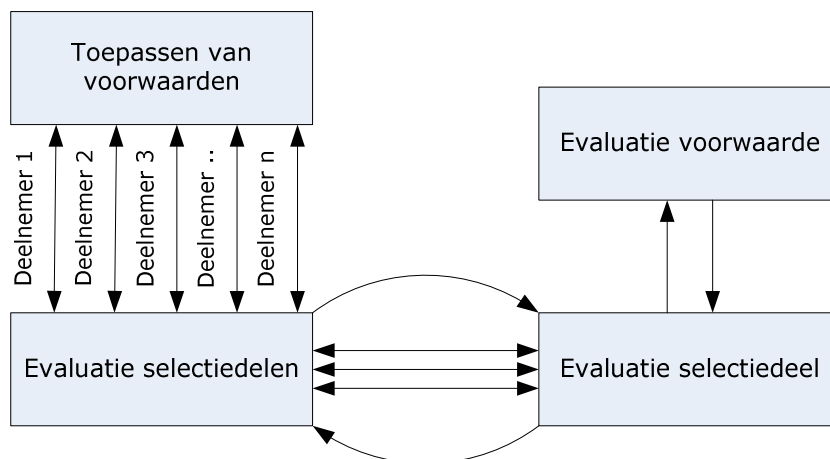
Het schrijven van de code achter het berekenen van de berekende variabelen was een zeer grote meevaller qua tijdsbesteding. Ik had in het achterhoofd al even gedacht aan de manier waarop dit zou moeten verlopen, maar door gebruik te maken van recursie waren de problemen die bij deze gedachtes naar boven kwamen totaal niet aan de orde. Een andere reden voor deze meevaller was dat het in een PHP-proces uitvoeren van een door de gebruiker gespecificeerd stuk andere code en het doorgeven van variabelen hieraan, erg simpel was.

Voorwaarden toepassen

Nu alles voor het proces aanwezig is, kunnen de voorwaarden worden toegepast. Dit was een grote uitdaging omdat ik zo iets nog nooit had gedaan. Ten eerste hebben de voorwaarden een hiërarchie, maar tegelijkertijd moet er ook rekening gehouden worden met de volgorde waarin operaties worden uitgevoerd. Ter illustratie: twee voorwaarden die aan elkaar zijn gekoppeld middels een logische 'of', mogen op het zelfde niveau pas geëvalueerd worden nadat er geen sets voorwaarden meer zijn die gekoppeld zijn middels een logische 'en'.

De reden waarom het toch zo simpel bleek is weer te vinden in recursie. Zie Figuur 26 voor de flow tussen functies. Per deelnemer wordt `evalueerSelectiedelen()` aangeroepen doe op zijn beurt met

inachtneming van logische operaties `evalueerSelectiedeel()` weer aanroept. In het geval dat `evalueerSelectiedeel()` zelf weer een groep tegenkomt, wordt hiervoor `evalueerSelectiedelen()` weer aangeroepen. Als `evalueerSelectiedeel()` een voorwaarde binnenkrijgt, wordt hiervoor `evalueerVoorwaarde()` aangeroepen. Op deze manier is het zo gemaakt dat er per deelnemer op een afgebakende manier kan worden opgevraagd of hij voldoet of niet.



Figuur 26 Relaties tussen functies voor toepassen van voorwaarden.

Extra ontdebellen

De optie bestaat om een extra variabele aan te geven waar vervolgens op ontdebeld wordt. Ook het schrijven van deze code verliep vlot en zonder problemen.

Nabewerken

Tegen het nabewerken werd voor ik ben begonnen met het schrijven van de code ervoor, ook opgekeken. Maar na enkele minuten goed nadenken, bleek het toch niet zo heel lastig tot een eenvoudige en goede oplossing te komen.

Tijdens het maken van de code is al begonnen met het schrijven van code voor de opslag van de resultaten, en het kunnen oproepen ervan. Dit laatste is een pilotdeel van de volgende pilot, dus deze is voor dit doel wat naar voren geschoven.

Ik was in eerste instantie van plan de relativiteit toe te voegen voor het moment van opslaan, maar voor de opslagefficiëntie heb ik toch gekozen om dit dynamisch te berekenen op het moment dat de resultaatset wordt bekeken of gedownload.

Opslaan

De code achter het opslaan draagt zorg voor het aanmaken van een resultaatset en het erachter plaatsen van de inhoud van de resultaatset. Deze programmatuur is zonder problemen geschreven.

Automatisch koppelen waar mogelijk

De tijd was nu rijp voor het informeel testen van het systeem met levensechte voorbeeldselecties. De selecties die ik in eerste instantie zelf heb gemaakt om wat mee te testen, bleken helaas niet vergelijkbaar met wat praktijksituaties.

In één van deze situaties kwam al vrij snel naar voren dat het selecteren van de basisdata teveel tijd in beslag nam, ondanks de reeds doorgevoerde optimalisatie. Met een snelheid van 25-75 deelnemers per seconde, zou een selectie uit een actie met 100.000 deelnemers al minimaal een half uur duren. Er is onderzoek uitgevoerd naar waar de flessennek zich bevond, en gekeken naar hoe dit opgelost kon worden. Het knelpunt bevond zich in het feit dat er massieve hoeveelheden simpele en kleine queries werden uitgevoerd. Deze bevestigingen waren wel erg snel uit te voeren door het DBMS, de opstartkosten per query voor het initialiseren, uitvoeren en retourneren van de resultaten bleken simpelweg te groot.

De oplossing lag in het automatisch koppelen van zoveel mogelijk tabellen waar mogelijk. In eerste instantie is ervoor gekozen dit niet te doen omdat opslagtabellen zich fysiek op verschillende DBMS'en zouden kunnen bevinden. Omdat nu blijkt dat dit toch vaak wel het geval is, en dat het koppelen een flinke tijdsinstaat op kan leveren, is deze optimalisatie doorgevoerd, met twee beperkingen. Tabellen worden alsnog individueel bevestigd indien:

- Er meer dan 31 tabellen gekoppeld moeten worden. MySQL kent namelijk een limiet van 32 tabellen binnen één enkele query.
- Tabellen zich niet binnen hetzelfde DBMS bevinden.

Er hoeven nu veel minder queries uitgevoerd te worden. De belasting op de databaseserve is nu per query wel hoger, maar per saldo zal de belasting zelfs lager uitvallen. Met deze optimalisatie wordt nu standaard een snelheid behaald van ongeveer 3000 deelnemers per seconde. Het ophalen van de data voor een actie met 100.000 deelnemers kost nu minder dan een minuut.

Geheugengebruik

Met het feit dat het nu makkelijker mogelijk is grote selecties te testen, kwam een ander groot probleem om de hoek kijken. Het leek erop alsof PHP voor elke byte aan geheugen die het in een variabele moest opslaan, tien bytes aan geheugenruimte nodig had. Een machine met

1,5 GB te gebruiken geheugen, zat vrij snel vol als er data van meer dan 100.000 deelnemers werd geselecteerd. Een simpele testsituatie bracht aan het licht, dat PHP vooral zoveel geheugen nodig heeft voor de opslag van lijsten. Hierbij kwam ook aan het licht dat het geheugengebruik onder Microsoft Windows vier keer zo inefficiënt was dan onder Linux. In eerste instantie stelde deze uitkomst mij gerust, maar ook onder Linux was de grens bij een grote telling op een gegeven moment bereikt.

Een oplossing zat hem in een klein hoekje, namelijk het gebruik van een tweetal in PHP ingebouwde functies die een lijst omzetten in een tekenreeks en vice versa. Hiermee neemt data die voorheen 1 GB in beslag nam, nu nog maar een paar honderd megabyte in. Nadeel hiervan is wel weer dat er flink meer rekenkracht nodig is om elke keer de tekenreeks om te zetten in een lijst wanneer deze gebruikt moet worden en andersom als deze niet meer gebruikt hoeft te worden.

Conclusie

De bouw van het belangrijkste deel van het systeem is overschat. Er is meer tijd voor ingepland dan dat er voor nodig is gebleken, maar een deel van deze tijd is wel weer besteed aan drie belangrijke optimalisaties. Zowel snelheid als geheugengebruik zijn hierbij fors verbeterd respectievelijk verminderd.

8.6 Beoordelen en testen pilot

Alle scenario's zijn met goed gevolg doorlopen, dus de pilot is als goed getest gekenmerkt. Tijdens de fase Invoering zijn er echter toch nog een paar onregelmatigheden naar boven gekomen. Zie §11.6 voor meer informatie.

9 Pilot #5, Resultaatbeheer

In dit hoofdstuk staan de activiteiten beschreven ten behoeve van de tweede pilot. Deze pilot gaat in op het bekijken, downloaden, verwijderen en doorgeven van resultaatsets. Zie bijlage 18 voor het volledige ontwikkelplan.

9.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld, zie bijlage 17 voor het plan.

Er is gekozen voor het laten vervallen van de organisatorische inrichting, omdat er voor deze pilot geen organisatorische aspecten te vermelden zijn: er zijn geen opleidingen en/of handleidingen behorende bij deze pilot. Ook is de activiteit 'SUMI-test opstellen' overgeslagen omdat er slechts een zeer beperkte GUI bestaat. Tot slot is ook de activiteit 'Globaal technische structuur specificeren' overgeslagen omdat de database reeds volledig is omgezet er ook geen andere technische overwegingen te maken zijn.

9.2 Globaal-functionele structuur

Tijdens het bepalen van de globaal-functionele structuur komen de functionaliteiten naar boven die in de pilot moeten worden ontworpen en gebouwd. Voor deze pilot waren de functionaliteiten als volgt:

- het kunnen oproepen van een overzicht van alle resultaatsets;
- het kunnen bekijken van een resultaatset;
- het kunnen downloaden van een resultaatset;
- het kunnen verwijderen van een resultaatset;
- het kunnen doorgeven van een resultaatset.

Globaal waren deze eisen al opgezet in de systeemeisen van de definitiestudie, maar op dit punt zijn ze verder tot in detail uitgewerkt. Er is bijvoorbeeld aangegeven welke gegevens van een resultaatset er in het overzicht naar voren moeten komen.

9.3 GUI-ontwerp

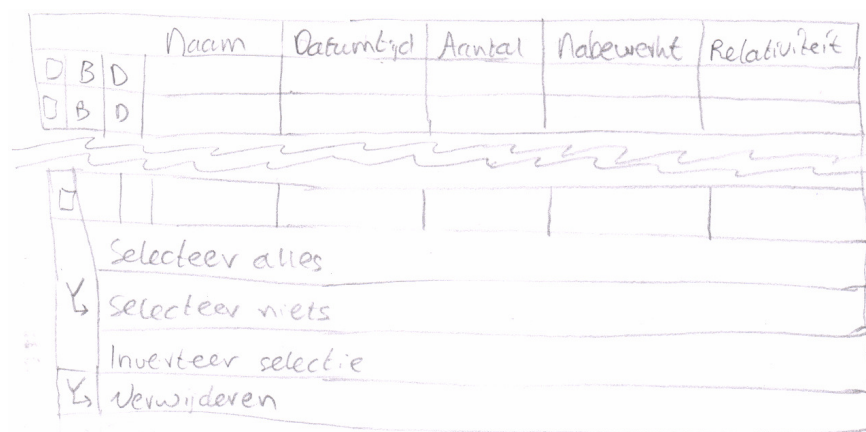
De volgende schermen zijn geschetst:

- Overzichtscherm
- Doorgeven

Zie Figuur 27 voor een van de schetsen.

geslacht			
leeftijd	m	v	Totaal
15	191 (0.4%)	220 (0.5%)	411 (0.9%)
16	245 (0.5%)	382 (0.8%)	627 (1.4%)
17	299 (0.7%)	727 (1.6%)	1026 (2.2%)
18	369 (0.8%)	1013 (2.2%)	1382 (3.0%)
19	408 (0.9%)	1211 (2.6%)	1619 (3.5%)
20	383 (0.8%)	1216 (2.6%)	1599 (3.5%)
21	422 (0.9%)	1185 (2.6%)	1607 (3.5%)
22	407 (0.9%)	1189 (2.6%)	1596 (3.5%)
23	423 (0.9%)	1083 (2.4%)	1506 (3.3%)
24	410 (0.9%)	1121 (2.4%)	1531 (3.3%)
25	445 (1.0%)	967 (2.1%)	1412 (3.1%)
26	372 (0.8%)	840 (1.8%)	1212 (2.6%)
27	386 (0.8%)	898 (2.0%)	1284 (2.8%)
28	378 (0.8%)	796 (1.7%)	1174 (2.6%)
29	399 (0.9%)	839 (1.8%)	1238 (2.7%)
30	366 (0.8%)	785 (1.7%)	1151 (2.5%)

Figuur 28 Een voorbeeldresultaatset.



Figuur 27 Schets van het overzichtscherm.

Naast het schetsen van de schermen is er ook een eenvoudig navigatieschema gemaakt om de navigatiemogelijkheden tussen de schermen visueel weer te geven.

9.4 Testplan

Voor het testplan zijn weer een aantal testscenario's samengesteld en uitgelegd die verderop gebruikt kunnen worden voor het testen van de

pilot. Voor de rest is het proces vergelijkbaar verlopen als beschreven in §5.6.

9.5 Ontwerp en bouw

In deze paragraaf worden alle werkzaamheden, aangaande het ontwerp en de bouw van de verschillende delen van deze pilot zelf, behandeld.

Overzicht

In dit deel is gebouwd aan een overzicht voor de resultaatsets. Als deze pagina wordt aangeroepen, genereert het systeem een tabel met hierin een lijst met alle bij het systeem bekende resultaatsets inclusief een aantal belangrijke attributen en mogelijkheden de resultaatsets te bekijken, downloaden of verwijderen.

Bekijken

Voor het bekijken heb ik een stukje code geschreven die simpelweg itereert door de verschillende cellen van de resultaatset en hierbinnen een HTML-tabel genereert. Waar nodig worden relatieve getallen toegevoegd of worden de absolute hiervoor vervangen. Zie Figuur 28 voor een voorbeeldresultaatset.

Downloaden

Het downloaden is op bijna dezelfde manier gemaakt. Maar in plaats van dat het einddoel de browser is, is het einddoel nu een CSV-bestand welke ter download wordt aangeboden.

Verwijderen

Een resultaatset mag altijd verwijderd worden, dus is op deze plek geen controle ingebouwd zoals bij sommige van voorgaande pilots.

Overige

Het pilotdeel met betrekking tot het doorgeven is niet aan bod gekomen. Op het moment dat het pilotdeel voor het verwijderen af was, is er in eerste instantie voor gekozen het pilotdeel even te bewaren voor als er op een later tijdstip weer tijd voor vrijkomt. Achteraf is deze tijd echter nooit vrijgekomen, dus zodoende zijn deze delen blijven liggen. Zie ook de procesevaluatie in hoofdstuk 12.

9.6 Beoordelen en testen pilot

Alle scenario's zijn met goed gevolg doorlopen, dus de pilot is als goed getest gekenmerkt.

10 Pilot #6, Gebruikersbeheer

Deze pilot is een deel van het systeem dat zorgdraagt voor een grafische gebruikersinterface ten behoeve van het beheer van gebruikers (die de selectietool kunnen gebruiken).

Veel van de pilotdelen zijn bewaard voor op een later tijdstip voor als er weer tijd voor vrijkomt, maar achteraf gezien is deze tijd nooit vrijgekomen wat ervoor heeft gezorgd dat deze pilotdelen nooit zijn gemaakt. De pilotdelen hebben gelukkig ook een zeer geringe waarde voor het systeem, omdat deze mogelijkheden toch zeer zelden gebruikt zouden worden. Indien een gebruiker namelijk is aangemaakt, hoeft hier niet meer naar omgekeken te worden, en kan deze gebruiker net zo vaak van het systeem gebruik maken als hij zelf wil. Daarnaast bestaat er via de database alsnog een manier om gebruikers te beheren.

Het enige pilotdeel met de basisprioriteit (de databasebasis) is al gebouwd voor pilot #1 voordat de code voor het inloggen is geschreven.

11 Invoering

De laatste activiteit van het project is het uitvoeren van de activiteiten in de fase Invoering. In deze fase is de pilot operationeel gemaakt in de gebruikersomgeving. Omdat er bij het maken van de iteratiestrategie voor gekozen is deze fase maar eenmalig uit te voeren, is dit een eenmalige activiteit geweest.

11.1 Plan van Aanpak

Als eerste is het Plan van Aanpak opgesteld. Zoals eerder al vermeld, verloopt het plannen op meerdere tijdstippen en niveaus. Waar de grote lijnen voor alle fasen uiteen zijn gezet in het algemene Plan van Aanpak, hier komen de activiteiten en producten specifiek voor deze fase aan bod om ingepland te worden.

De beoogde start- en einddatum waren al bekend, en met dat als uitgangspunt zijn de activiteiten ingepland. Bij het opstellen van het algemene Plan van Aanpak is al rekening gehouden met de activiteiten tijdens deze fase, dus het is niet zo dat nu pas duidelijk wordt hoe lang de fase zal duren; puur alleen wanneer welke taak aan zou moeten vangen. Zie bijlage 19 voor het plan.

11.2 Uitvoeren SUMI-test

Tijdens het opstellen van het pilotontwikkelplan voor een aantal van de pilots, zijn per pilot een aantal scenario's opgesteld. Voor het houden van de SUMI-test zijn deze scenario's in een afzonderlijk document bij elkaar gezet en aangevuld met wat begeleidende tekst. Zie bijlage 20 voor deze voorbereiding van de SUMI-test.

De scenario's zijn vervolgens voorgelegd aan de operationele medewerkers (diegene die met het systeem moeten gaan werken) en hebben deze uitgevoerd. Na afloop is de SUMI-vragenlijst ingevuld t.b.v. de beoordeling van de gebruikersvriendelijkheid. Deze resultaten zijn vervolgens doorgegeven aan SUMISCO (SUMI scoring package) om de antwoorden te interpreteren en te komen tot een aantal indexgetallen. Zie Fragment 19 voor de resultaten van de test. De schaal ligt van 0 (slecht) tot 100 (goed).

User	Global	Efficiency	Affect	Helpfulness	Control	Learnability
1	61	65	62	53	63	63
2	67	53	71	71	56	70

Fragment 19 Resultaten van de test.

11.3 Pilots invoeren

Tijdens het invoeren van het systeem is de database geïmplementeerd en zijn de bronbestanden geplaatst en geconfigureerd. Dit alles is nu even gebeurd op een tijdelijke productieomgeving, alwaar in ieder geval voor de eerste tijd even gekeken kan worden wat het systeem concreet voor belasting legt op de onderliggende systemen.

Ook is de database zelf gevuld met informatie over alle Nederlandse acties en zijn er een aantal gebruikers aangemaakt.

11.4 Pilotacceptatie uitvoeren

De acceptatie van de pilot is uitgevoerd door de opdrachtgever. Hij heeft formeel vastgesteld dat het systeem voldoet aan de gestelde systeemeisen en de daaruit afgeleide functionele specificaties. Tevens is nagegaan of het systeem effectief en efficiënt werkt.

11.5 Pilots overdragen

Tijdens de overdracht van de pilot zijn de verschillende belangengroepen in staat gesteld het systeem overeenkomstig de doelstellingen en opzet te gebruiken en te beheren. Voor dit systeem betekende dit dat de programmeur binnen E2Ma is ingeleid in de belangrijke ideeën achter de broncode.

11.6 Feedback verzamelen

Tijdens de acceptatie zijn een aantal kleine en middelgrote verbeterpunten naar voren gekomen. Omdat er geen tijd meer is deze feedback binnen de afstudeerperiode te verwerken, is met het bedrijf overeengekomen dat deze werkzaamheden na het inleveren van dit verslag alsnog zullen worden uitgevoerd.

Een van de verbeterpunten is dat er in de toekomst ontdubbeld moet worden op meer dan één variabele. Pas in deze laatste fase is opgemerkt dat dit wel nodig kan zijn. Er is altijd gedacht dat het ook voldoende is om - als er ontdubbeld moet worden op twee afzonderlijke variabelen - deze eerst door middel van een berekende variabele aan elkaar te concateneren, maar in feite ontdubbeld dit alleen op unieke combinaties tussen deze twee variabelen.

11.7 Pilots ondersteunen

Tijdens ondersteuning van het systeem zijn knellende problemen opgelost die in de weg stonden voor een goed gebruik van het systeem. Zo is er op deze manier nog een kinderziekte ontdekt en opgelost.

12 Procesevaluatie

In dit hoofdstuk zal ik ingaan op hoe ik denk over het gevolgde proces. Er zijn zaken die goed zijn gegaan, maar ook een aantal die beter hadden gekund.

Ik ben begonnen met de gedachte om de vanuit school aangeleerde basisprincipes eerst zoveel mogelijk te toetsen met de literatuur en mij vooral door dit laatste te laten leiden. Dit om een zo goed mogelijke afstemming te verkrijgen tussen het te volgen proces volgens de ontwikkelmethodiek en wat er daadwerkelijk bij het project past. Dit is een goede keuze geweest omdat ik het gevoel heb dat er geen overbodige dingen zijn gebeurd en de juiste activiteiten zijn weggelaten of juist vanuit GUIDE zijn opgenomen.

Een punt waar kritisch naar gekeken moet worden is het langzaam maar zeker ontstane verschil met de planning. Gedurende het project is namelijk een achterstand ontstaan van ruim anderhalve week. Op een gegeven moment was de achterstand zelfs nog meer, maar door een overschatting van de benodigde tijd voor de pilot Selectieactivatie kon dit voor een deel weer ingelopen worden. Veel van de achterstand is te wijten aan de onderschatting van de benodigde tijd voor de pilot Selectiebeheer. De uitvoering van deze pilot heeft zoveel meer tijd gekost vanwege twee hete vuren waar de beste tussenweg door gevonden moest worden. Aan de ene kant moest het via de gebruikersinterface mogelijk zijn de meest ingewikkelde selectie-specificaties op te bouwen, aan de andere kant mocht dit niet ten koste gaan aan de gebruiksvriendelijkheid. De gekozen weg is een lange weg geweest, maar achteraf ben ik blij dat ik het zo heb opgelost.

Hieruit heb ik in ieder geval geleerd dat zaken die in het begin als een echte uitdaging worden gezien, dit in werkelijkheid misschien helemaal niet zijn en zonder problemen te realiseren zijn. Er zit voor mij ook een andere kant aan het verhaal, want er zijn ook dingen geweest waar ik in het begin te licht over heb gedacht.

Als gevolg van deze achterstand is slechts een enkel pilotdeel met de comfort- of luxeprioriteit ontworpen of gebouwd. Wel zijn de specificaties gemaakt en is er plaats voor in de database, dus deze niet-basis-delen kunnen relatief gemakkelijk alsnog gemaakt worden. Alle pilotdelen met de basisprioriteit zijn wel gewoon ontworpen en gebouwd.

Tot slot denk ik dat ik meer tussentijds had moeten communiceren met de opdrachtgever. Tijdens de iteraties van de fase Pilotontwikkeling is er weliswaar geen pure radiostilte geweest, toch denk ik dat een aantal

verbeterpunten die nu tijdens de fase Invoering naar boven zijn gekomen, eerder boven water hadden kunnen komen door meer te communiceren.

Concluderend kan gesteld worden dat ik tevreden ben over het gevolgde proces. Er is veel werk verzet, en ik heb het gevoel de juiste keuzes te hebben gemaakt.

13 Productevaluatie

In dit hoofdstuk staan de producten beschreven met een kritische blik. Achtereenvolgens zal worden ingegaan op de documentatie, de database en de afzonderlijke pilots.

De documentatie is uitgebreid en consistent. Hier en daar kunnen wel wat spellings- en grammaticafoutjes voorkomen, maar dit heeft nooit invloed gehad op de producten die voor een deel weer voortkwamen uit deze documentatie.

De database is opgezet met normalisatie tot de Boyce-Codd normaalvorm en derhalve bevat het geen redundantie. Er zijn sleutels toegevoegd op strategische plaatsen, en het geheel zou schaalbaar en gemakkelijk uitbreidbaar moeten zijn.

De kwaliteit van pilot #1, de GUI-standaard, is erg hoog gebleken. Met mijn letterlijke jarenlange ervaring op dit vlak was ook niet anders te verwachten. De robuustheid en consistentie van deze pilot is merkbaar door het gehele systeem heen. Toch zou ik hier voor de volgende keer en specifiek voor een project als deze nog wat meer aandacht besteden aan foutafhandeling. In pilot #4 is namelijk de behoefte ontstaan om foutmeldingen - onafhankelijk van of het systeem zich dan in productiemodus bevindt - toch te tonen aan de gebruiker, waar op dit moment alle foutmeldingen zich onconditioneel begeven naar de beheerder.

Pilots #2 en #5 (Variabelenbeheer en Resultaatsetbeheer) zijn beide relatief eenvoudig te maken pilots geweest, en de kwaliteit hiervan is daarom ook gewoon goed.

Voor wat betreft pilot #3 kan er gezegd worden dat het een erg complexe pilot is geworden. Hoewel goed getest, is dit typisch een pilot waarvoor het onmogelijk is alle mogelijke combinaties te testen. Deze pilot is wat dat betreft meer gevoelig voor kinderziekten dan de andere. Tevens zal het uitbreiden ervan tot een van de lastigere taken behoren. Het zal een onbeantwoorde vraag blijven hoe deze situatie had geweest als er bij de ontwikkeling van deze pilot een andere weg was gekozen, bijvoorbeeld door veel intensiever gebruik te maken van pilotdelen.

Pilot #4, de belangrijkste van allemaal voor het activeren van een selectie en het produceren van een resultaatset, is erg robuust. De snelheid was in het begin te laag, maar er zijn een paar significante wijzigingen doorgevoerd die de snelheid op een acceptabel niveau hebben gebracht. Er bevindt zich hier nog een enkel verbeterpunt op het afhandelen van fouten.

Op het gebied van omgang met het geheugen is PHP uit de bus gekomen als niet uitermate geschikt voor dit soort analysetaken, maar het voldoet voor dit project nog zeker wel. Ik vermoed niet alleen dat een gecompileerde taal als C++ hier niet alleen nog veel sneller mee zal omgaan, maar ik denk ook dat het geheugenverbruik hiermee beter in de hand te houden is.

Uit de SUMI-test is gebleken dat de gebruikersvriendelijkheid als goed wordt ervaren.

De producten zijn allemaal van hoge kwaliteit. Hier en daar zijn mogelijkheden voor verbeteringen, maar vooralsnog verwacht ik dat dit alles op dit moment goed aansluit bij de doelstelling van dit project.

Bijlage 1: Opdrachtomschrijving

1 Inleiding

Dit is de definitieve opdrachtomschrijving van mijn afstudeerstage. Achtereenvolgens zal ik in gaan op de organisatie en de opdracht, de probleemstelling, de doelstelling, de uitgangssituatie, de werkzaamheden en op te leveren producten.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 4.

Datum	Versie	Omschrijving
21-01-2005	1.0	Eerste versie aan de hand van eerste gesprek en informatie van StagePlaza.
27-01-2005	1.1	Geschied gemaakt voor eerste gesprek met keurend docent met informatie uit tweede gesprek met E2Ma.
02-02-2005	1.2	Uitbreidingen en veranderingen naar aanleiding van eerste gesprek met keurend docent. Doelstellingen en werkzaamheden opgeSMART.
03-02-2005	1.3	Probleemstelling verduidelijkt, twee kleine formuleringsveranderingen doorgevoerd en structuur gelijkgetrokken aan de structuur zoals die in de afstudeer-mhgroff te vinden is.
08-02-2005	1.4	Nieuw sjabloon toegepast en wat kleine dingetjes doorgevoerd.
29-03-2005	1.5	Gesynchroniseerd met algemeen Plan van Aanpak en aangepast voor bedrijfsbezoek waarin tot een definitieve versie gekomen zal worden.
12-04-2005	2.0	Gesynchroniseerd met inmiddels samengestelde definitieve opdrachtomschrijving.

Tabel 3 Veranderingengeschiedenis.

2 Opdrachtomschrijving

Ontwikkeling van een dataselectietool bij E2Ma Consumer Profiling

De organisatie E2Ma (spreek uit als een afkorting) is gevestigd in Houten. Ze verzamelt en verhuurt consumentenprofielen die worden ingezet voor direct-marketing-doeleinden. Deze consumentenprofielen bestaan uit: persoons-, sociodemografische- en gedragsinformatie. De gegevens in deze profielen worden verzameld door het houden van enquêtes die iedereen via internet kan invullen. Om mensen te aan te trekken en te motiveren hun juiste gegevens in te vullen worden prijzen verloot. Nadat de gegevens eenmaal zijn verzameld, worden deze via verschillende kanalen op de markt gebracht. Voorbeelden van afnemers zijn Daimler Chrysler, MyTravel en het Wereld Natuur Fonds.

De hoeveelheid data die E2Ma over consumenten verzamelt neemt alsmaar toe. Deze informatie wordt, afhankelijk van het doel, in een aantal verschillende MySQL-databases opgeslagen. Ook neemt de

vraag uit de markt naar kwalitatief hoogwaardige data steeds verder toe. Deze twee feiten leiden ertoe dat selecties van data uit deze consumentenprofielen (aggregaties, tellingen, samenvattingen, enzovoorts) steeds complexer worden. Om deze complexiteit in te dammen en te voorkomen het ten koste gaat van de efficiëntie, groeit de behoefte aan een gebruikersvriendelijke en complete interface voor het uitvoeren van deze selecties.

Het doel van de afstudeeropdracht is het ontwerpen, beschrijven en ontwikkelen van een systeem waarmee operationele medewerkers selecties op de databases kunnen verrichten. Dit dient snel, makkelijk en betrouwbaar te kunnen. Het maakt niet uit welke selectie er uitgevoerd moet worden, maar dit moet altijd binnen vijf à tien minuten kunnen. Dat het makkelijk moet zijn speelt hier een beetje op in: operationele gebruikers moeten eenvoudiger dan voorheen met de selectiesoftware overweg kunnen waardoor ze er minder tijd aan kwijt zijn. De gegevens die eruit komen moeten ook betrouwbaar zijn: ze moeten voor 100% representeren wat de gebruiker heeft gevraagd en zoals in de achterliggende databases is opgeslagen. Communicatie dient te gebeuren via een grafische gebruikersinterface. Deze dient gebruikersvriendelijk te zijn wat zal moeten blijken uit een SUMI-test. De gebruikersinterface zal verantwoord worden op het vlak van de gebruikersvriendelijkheid, maar ook op het vlak van de vormgevingsvakken. Tot slot moet het systeem ook schaalbaar zijn: het moet mee kunnen groeien met ontwikkelingen binnen E2Ma en daarbij behorende databaseclustering.

Er kunnen optioneel voorzieningen aanwezig zijn voor het kunnen aanmaken en uitvoeren van selecties als voorzieningen voor de afnemer van E2Ma: bijvoorbeeld dat een selectie periodiek automatisch wordt uitgevoerd en op een of andere wijze wordt afgeleverd bij de klant in een bepaald formaat.

En hieraan gelieerd:

- Het analyseren en eventueel optimaliseren van de bestaande MySQL-databases op efficiëntie.
- Werkzaamheden uitvoeren op het gebied van netwerkbeheer, internet, databaseselecties en vormgevingen.

Voor het uitvoeren van de opdracht is een geschikte PC benodigd met de volgende software:

- Microsoft Windows
- Microsoft Office (inclusief Visio)
- GNU/Linux
- vim
- Apache webserver
- PHP
- MySQL

Op dit moment zijn er verschillende systemen (zowel aan de kant van de informatieverstrekken, aan de kant van de opslag en aan de kant van de selectie) die samen een al werkend en geïntegreerd geheel vormen. In principe hoeft hier niets aan veranderd te worden.

In het kader van de afstudeeropdracht zullen de volgende activiteiten verricht worden:

- Projectscenario maken
- Algemeen Plan van Aanpak opstellen
- Fase Definitiestudie:
 - Plan van Aanpak opstellen
 - Workshop voorbereiden
 - Ontwikkelscenario-workshop: ontwikkelscenario bepalen
 - Pilotplanworkshop voorbereiden
 - Pilotplanworkshop:
 - Systeemeisen definiëren:
 - Gebruikers en usability-eisen definiëren
 - Systeemconcept bepalen:
 - Selecties uitvoeren
 - Analyseren bestaande databases
 - Taakdiagrammen
 - Taakscenario's
 - Use-case diagram
 - Klassendiagram
 - Selectieproces
 - Stijlgids maken
 - Technische structuur beschouwen
 - Organisatorische inrichting beschouwen
 - Pilotplan opstellen
- Fase Pilotontwikkeling (wordt net zo vaak doorlopen als er pilots zijn gedefinieerd):
 - Pilotplan bijwerken
 - Plan van Aanpak opstellen
 - SUMI-test opstellen
 - Globaal-functionele structuur pilot specificeren
 - Globaal-technische structuur pilot specificeren
 - Globaal-organisatorische inrichting specificeren
 - GUI-ontwerp:
 - Schermschetsen
 - Navigatieschema
 - Pilotontwikkelplan opstellen
 - Per softwarebouweenheid (wordt dus net zo vaak doorlopen als er bouweenheden zijn gedefinieerd):
 - Ontwerpen softwarebouweenheden
 - Bouw softwarebouweenheden
 - Samenstellen handleidingen
 - Opstellen invoeringsprocedures
 - Beoordelen en testen pilot
- Fase invoering:

- Plan van Aanpak opstellen
- SUMI-test uitvoeren
- Pilot invoeren
- Pilotacceptatie uitvoeren
- Pilot overdragen
- Feedback verzamelen
- Pilot ondersteunen

Bij de uitvoering van de opdracht gaat gebruik gemaakt worden van de IAD-methodiek met GUIDE voor de ontwikkeling van de grafische gebruikersinterface.

Te gebruiken technieken zijn:

- Databasemodelleringstechnieken
- UML-tekentechnieken
- Interviewtechnieken.

De volgende producten zullen opgeleverd worden:

- Projectscenario
- Algemeen Plan van Aanpak
- Definitiestudierapport
- Per pilot in de fase Pilotontwikkeling:
 - Plan van Aanpak
 - Pilotontwikkelplan
 - Testplan
 - Pilot zelf (programmacode, databasestructuren, enzovoorts).
 - Handleiding
- Fase Invoering:
 - Plan van Aanpak
 - Resultaten SUMI-test
 - Eventueel: feedback

Bijlage 2: Algemeen Plan van Aanpak

1 Inleiding

Dit is het algemene Plan van Aanpak. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Projectscenario en voor de fase Definitiestudie. Globaal omdat het in de Plannen van Aanpak uit de verschillende fasen en activiteiten verder zal worden uitgewerkt. Dit rapport is geschreven met de voorlopige opdrachtomschrijving als voorname invoer. Er zal echter ook een terugkoppeling plaats vinden vanaf dit Plan van Aanpak terug naar de opdrachtomschrijving om deze een wat meer definitief karakter te geven.

In hoofdstuk 2 staat de achtergrond van het project. Een kleine inleiding op de opdracht en wat informatie over het bedrijf. Hoofdstuk 3 bevat vervolgens de probleemstelling. Welke delen van het probleem er aangepakt worden staat vermeld in de doelstelling in hoofdstuk 4. De uitgangssituatie is te vinden in hoofdstuk 5. De doelstelling in combinatie met de uitgangssituatie leidt tot een aantal werkzaamheden en die staan vermeld in hoofdstuk 6. De planning hiervan staat vervolgens in hoofdstuk 7 om in hoofdstuk 8 vervolgd te worden door de concrete producten die hier uit voortkomen. In hoofdstuk 9 staan de risicofactoren en het rapport wordt tenslotte afgesloten met de te vermelden nadrukken in hoofdstuk 10.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 4.

Datum	Versie	Omschrijving
10-02-2005	1.0	Initiële versie.
14-02-2005	1.1	De ontwikkelscenario-workshop en een eerste deel van het pilotplan-workshop vallen nu gelijk. De planning is hierop aangepast.
01-03-2005	1.2	Het uitwerken van het selectieproces toegevoegd aan de activiteiten en de hieruit voortkomende producten. Urenbegroting herzien op basis van eigen ervaringen.
17-03-2005	1.3	Planning ingevuld met de tijdens de fase Definitiestudie gevonden pilots en hun nu bekende urenbegrotingen.
16-04-2005	1.4	Gesynchroniseerd met definitieve opdrachtomschrijving.

Tabel 4 Veranderingengeschiedenis.

2 **Achtergrond**

In dit hoofdstuk worden de achtergronden van de opdracht besproken. Allereerst wordt ingegaan op het bedrijf, en vervolgens op de opdracht.

De organisatie heet E2Ma (spreek uit als een afkorting) en is gevestigd in Houten. Ze verzamelen en verhuren consumentenprofielen die worden ingezet voor direct-marketing-doeleinden. Deze consumentenprofielen bestaan uit: persoons- sociodemografische- en gedragsinformatie. De gegevens in deze profielen worden verzameld door het houden van enquêtes die iedereen via internet kan invullen. Om mensen te aan te trekken en te motiveren hun gegevens in te vullen worden prijzen verloot. Nadat de gegevens eenmaal zijn verzameld, worden deze via verschillende kanalen op de markt gebracht. Voorbeelden van afnemers zijn Daimler Chrysler, MyTravel en het Wereld Natuur Fonds.

De hoeveelheid data die E2Ma over consumenten verzamelt neemt alsmat toe. Deze informatie wordt, afhankelijk van het doel, in een aantal verschillende MySQL- en Microsoft Access-databases opgeslagen. Ook neemt de vraag uit de markt naar kwalitatief hoogwaardige data steeds verder toe. Deze twee feiten leiden ertoe dat selecties van data uit deze consumentenprofielen (aggregaties, tellingen, samenvattingen, enzovoorts) steeds complexer worden. Om deze complexiteit in te dammen, groeit de behoefte aan een gebruikersvriendelijke en complete interface voor het uitvoeren van deze selecties.

3 **Probleemstelling**

In dit deel wordt de probleemstelling beschreven.

Het selecteren van data wordt een steeds complexer proces vanwege de volgende drie factoren: de hoeveelheid data neemt toe, de complexiteit van de data neemt toe en door de klant gevraagde kwaliteit neemt toe. Dit is een probleem omdat dit spaak zal lopen met kennis en kunde van de operationele medewerkers (gebruikers die op dit moment selecties uitvoeren via de oude manier): zij zullen er op den duur geen selecties meer uit kunnen voeren binnen een aanvaardbare tijdsduur en dit kost E2Ma geld.

4 **Doelstelling**

In dit hoofdstuk wordt de doelstelling beschreven. Nadat het probleem is vastgesteld, wordt hier gespecificeerd wat van dit probleem er opgelost moet gaan worden.

De doelstelling is het ontwerpen, beschrijven en ontwikkelen van een systeem waarmee operationele medewerkers selecties op de databases kunnen verrichten. Dit dient snel, makkelijk en betrouwbaar te kunnen.

Het maakt niet uit welke selectie er uitgevoerd moet worden, maar dit moet altijd binnen vijf à tien minuten kunnen. Dat het makkelijk moet zijn speelt hier een beetje op in: operationele gebruikers moeten eenvoudiger dan voorheen met de selectiesoftware overweg kunnen waardoor ze er minder tijd aan kwijt zijn. De gegevens die eruit komen moeten ook betrouwbaar zijn: ze moeten voor 100% representeren wat de gebruiker heeft gevraagd en zoals in de achterliggende databases is opgeslagen. Tevens dient de gebruikersinterface gebruiksvriendelijk te zijn. Dit zal blijken uit de uitkomsten van een SUMI-test. De gebruikersinterface zal verantwoord worden op het vlak van de gebruikersvriendelijkheid, maar ook op het vlak van de vormgevingsvakken.

Er moeten zowel voorzieningen aanwezig zijn voor het kunnen aanmaken en uitvoeren van selecties als voorzieningen voor de afnemer van E2Ma: bijvoorbeeld dat een selectie periodiek automatisch wordt uitgevoerd en op een of andere wijze wordt afgeleverd bij de klant in een bepaald formaat.

En hieraan gelieerd:

- Het analyseren en eventueel optimaliseren van de bestaande MySQL-databases op efficiëntie en er voor zorgen dat er in de toekomst niet tegen onvoorziene dingen aangelopen kan gaan worden.
- Het opstellen van een nieuw databasemodel volgens de literatuur die door de opleiding is aangeboden. Dit heeft verder geen verdere consequenties voor enige bestaande structuren maar is puur om te kijken wat de verschillen zijn.
- Werkzaamheden uitvoeren op het gebied van netwerkbeheer, internet en databaseselecties.

5 Uitgangssituatie

5.1 Benodigde software

Er dient gebruik gemaakt te worden van de volgende software:

- Microsoft Windows
- Microsoft Office
- GNU/Linux
- vim
- Apache webserver
- PHP
- MySQL

Op het moment zijn er verschillende systemen (zowel aan de kant van de informatieverstrekkers, aan de kant van de opslag en aan de kant van de selectie) die samen een al werkend en geïntegreerd geheel vormen. Op een eventueel verbeteradvies voortkomend uit de analyse van de database hoeft hier overigens niets aan veranderd te worden.

5.2 Benodigde hardware

Voor de ontwikkeling is een PC benodigd met eigenschappen om de uit te voeren activiteiten soepel te kunnen uitvoeren.

5.3 Beschikbare rapporten

Er is documentatie beschikbaar ter beschrijving van deze systemen en de werking ervan.

5.4 Aanwezige ideeën

Er zijn ideeën aanwezig over hoe de selectieprocedure in de ideale situatie zou moeten verlopen.

6 Werkzaamheden

6.1 Uit te voeren activiteiten

Onderstaand – globaal – alle te volgen fasen en activiteiten met de gekozen iteratiestrategie in het achterhoofd. Zie Tabel 5 voor een legenda om te kunnen zien waar elke activiteiten vandaan komt.

Legenda
IAD
GUIDE
Overig

Tabel 5 Legenda

- 1 Projectscenario maken
- 2 Algemeen Plan van Aanpak opstellen
- 3 Fase Definitiestudie:
 - 1 Plan van Aanpak opstellen
 - 2 Workshop voorbereiden
 - 3 Ontwikkelscenario-workshop: ontwikkelscenario bepalen
 - 4 Pilotplanworkshop voorbereiden
 - 5 Pilotplanworkshop:
 - 1 Systeemeisen definiëren
 - 1 Gebruikers en usability-eisen definiëren
 - 2 Systeemconcept bepalen
 - 1 Taakdiagrammen
 - 2 Taakscenario's
 - 3 Use-case diagram
 - 4 Klassendiagram
 - 5 Selectieproces
 - 6 Stijlgids maken
 - 3 Technische structuur beschouwen
 - 4 Organisatorische inrichting beschouwen
 - 5 Pilotplan opstellen
- 4 Fase Pilotontwikkeling (wordt net zo vaak doorlopen als er pilots worden gespecificeerd in 3.5.5)
 - 1 Pilotplan bijwerken
 - 2 Plan van Aanpak opstellen
 - 3 SUMI-test opstellen
 - 4 Globaal-functionele structuur pilot specificeren
 - 5 Globaal-technische structuur pilot specificeren
 - 6 Globaal-organisatorische inrichting specificeren
 - 7 GUI-ontwerp

- 1 Schermschetsen
- 2 Navigatieschema
- 8 Testplan opstellen.
- 9 Pilotontwikkelplan opstellen
- 10 Per softwarebouweenheid (wordt dus net zo vaak doorlopen als er bouweenheden worden gespecificeerd in 5.8):
 - 1 Ontwerpen softwarebouweenheden
 - 2 Bouw softwarebouweenheden
 - 3 Samenstellen handleidingen
 - 4 Opstellen invoeringsprocedures
- 11 Beoordelen en testen pilot
- 5 Fase Invoering
 - 1 Plan van Aanpak opstellen
 - 2 SUMI-test uitvoeren
 - 3 Pilot invoeren
 - 4 Pilotacceptatie uitvoeren
 - 5 Pilot overdragen
 - 6 Feedback verzamelen
 - 7 Pilot ondersteunen

Daarnaast zullen er ook nog afzonderlijk werkzaamheden verricht worden, waaronder:

- 6 Analyseren databases en eventueel doen van aanbevelingen.
- 7 Een databasemodel maken voor hoe de dataopslag opgezet zou worden hoe dit vanuit de opleiding is aangeleerd. Dit heeft verder vooralsnog geen verdere consequenties.
- 8 Stroomlijnen planningsproces.
- 9 Databaseselecties en ondersteunende werkzaamheden.

Aanpassingen aan IAD

In deze paragraaf de uitzonderingen en veranderingen op de standaard IAD-indeling van fasen en activiteiten. Er wordt hierbij indien mogelijk direct verwezen naar de elementen uit bovenstaande opsomming.

- IAD-activiteit 2.02 (voorbereiden pilotontwerpworkshop) wordt hier overgeslagen vanwege het feit dat de pilotontwerpworkshop eigenlijk geen workshop is met de afstudeerder als enige benodigde deelnemer.
- 4.1: Pilotplan bijwerken vindt nu ook plaats in de fase Pilotontwikkeling omdat de fase Definitie slechts eenmaal wordt doorlopen en er na elke pilot toch nog feedback moet zijn naar dit pilotplan.
- IAD-activiteiten 2.09 (aangepassen extra componenten), 2.10 (wijzigen andere informatiesystemen), 2.11 (bouw conversie-tools), 2.12 (ontwerp handmatige procedures), 2.16 (integreren bouweenheden) worden ook overgeslagen omdat ze voor dit project irrelevant zijn.
- 4.11: het beoordelen en testen zal betrekking hebben op de gehele pilot met al zijn delen in plaats van alleen op een pilotdeel. Hier is

voor gekozen omdat het in de lijn der verwachtingen licht dat de pilotdelen klein in omvang zullen worden.

- IAD-activiteiten 2.13 (opleidingsmateriaal samenstellen) en 3.03 (opleidingen verzorgen) worden overgeslagen omdat – behoudens dan misschien een korte introductie tijdens een presentatie – de verwachting bestaat dat er geen opleiding nodig zal zijn.

6.2 Te hanteren methodieken

Er gaat gebruik gemaakt worden van de IAD-methodiek met GUIDE voor de ontwikkeling van de grafische gebruikersinterface.

6.3 Te hanteren technieken

Bij het gebruik van de IAD-methodiek en bij andere werkzaamheden, zullen verschillende technieken benodigd zijn. Dit zijn:

- Databasemodelleringstechnieken;
- UML-tekentechnieken;
- interviewtechnieken.

7 Planning

In dit deel bevindt zich een al wat uitgebreidere planning als dan die in de opdrachtomschrijving te vinden is, zie Tabel 6. Deze planning wordt tijdens de fase Definitiestudie nog wat verder uitgediept.

Activiteit	Weken																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Opdrachtomschrijving																		
Projectscenario																		
Algemeen Plan van Aanpak opstellen																		
Fase Definitiestudie																		
Plan van Aanpak opstellen																		
Workshops voorbereiden																		
Ontwikkelscenario-workshop																		
Pilotplanworkshop voorbereiden																		
Pilotplanworkshop																		
Fase Pilotontwikkeling																		
GUI-standaard																		
Variabelenbeheer																		
Selectiebeheer																		
Selectieactivatie																		
Resultaatsetbeheer																		
Gebruikersbeheer																		
Fase Invoering																		
Plan van Aanpak																		
Pilot invoeren																		
Pilotacceptatie uitvoeren																		
Pilot overdragen																		
Feedback verzamelen																		
Pilot ondersteunen																		
Analyseren databases																		
Maken nieuw databasemodel																		
Stroomlijnen planningsproces																		
Databaseselecties en ondersteunende werkzaamheden.																		

Elk moment

Tabel 6 De planning

8 Producten

In deze paragraaf worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 1: Opdrachtoomschrijving
- 2: Projectscenario
- 3: Algemeen Plan van Aanpak
- 4: Definitiestudie:
 - Plan van Aanpak.
 - Alle producten voortkomend uit 4.2 t/m 4.6.
- Voor naargelang het aantal tijdens 4.5.5 gedefinieerde pilots:
 - 5.2: Plan van Aanpak.
 - 5.9: Pilotontwikkelplan met de uitkomsten van punten 5.3 t/m 5.8.
 - Per softwarebouweenheid zoals deze tijdens 5.9 zijn gedefiniëerd: het resultaat van 5.10.1.
 - De pilot zelf (programmacode, databasestructuren, enzovoorts).
 - Handleiding
- 6: Invoering:
 - Plan van Aanpak
 - Resultaten SUMI-test
 - Eventueel: feedback.

Bijlage 3: Externe bijlagen

Alle overige documentatie is extern opgenomen. Zie onderstaande lijst voor een inhoudsopgave van het document met deze externe bijlagen.

4	Projectscenario	5
5	Plan van Aanpak fase Definitiestudie	9
6	Vorbereiding workshops	11
7	Definitiestudie	19
8	Plan van Aanpak pilot #1	45
9	Pilotontwikkelplan pilot #1	49
10	Productiehandleiding	61
11	Plan van Aanpak pilot #2	65
12	Pilotontwikkelplan pilot #2	69
13	Plan van Aanpak pilot #3	81
14	Pilotontwikkelplan pilot #3	85
15	Plan van Aanpak pilot #4	103
16	Pilotontwikkelplan pilot #4	107
17	Plan van Aanpak pilot #5	117
18	Pilotontwikkelplan pilot #5	121
19	Plan van Aanpak fase Invoering	125
20	Vorbereiding SUMI-test	127

Literatuurlijst

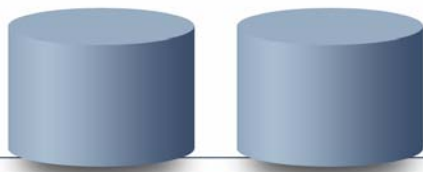
Elling, E., B. Andeweg, J. de Jong, C. Swankhuisen,
Rapportagetechniek. 2^e druk. Groningen: Wolters-Noordhoff, 2000

Pyle, D en A. Moore, *GUIDE. Graphical User Interface Design and Evaluation*. Prentice-Hall, 1999

Schmuller, J., *UML in 24 uur*. Schoonhoven: Academic Service, 2000.

Tolido, R.J.H., *IAD. Het evolutionair ontwikkelen van informatiesystemen*. Schoonhoven: Academic Service, 2000.

Vandenbulcke, J.A. en W.O. Lemahieu, *Databasesystemen voor de praktijk*. Den Haag: Ten Hagen en Stam Uitgevers, 2000.



Afstudeerverslag

Dataselectietool E2Ma

Externe bijlagen

Inleiding

Dit document omvat alle externe bijlagen behorende bij het afstudeerverslag. Bijlagen 1 t/m 3 zijn niet extern en bevinden zich in het verslag zelf. Een digitale versie van dit document is te vinden onder <http://rcpaul.nl/hhs/afst-exb.pdf>.

4	Projectscenario	5
5	Plan van Aanpak fase Definitiestudie.....	9
6	Vorbereiding workshops	11
7	Definitiestudie.....	19
8	Plan van Aanpak pilot #1	45
9	Pilotontwikkelplan pilot #1	49
10	Productiehandleiding	61
11	Plan van Aanpak pilot #2	65
12	Pilotontwikkelplan pilot #2	69
13	Plan van Aanpak pilot #3	81
14	Pilotontwikkelplan pilot #3	85
15	Plan van Aanpak pilot #4	103
16	Pilotontwikkelplan pilot #4	107
17	Plan van Aanpak pilot #5	117
18	Pilotontwikkelplan pilot #5	121
19	Plan van Aanpak fase Invoering	125
20	Vorbereiding SUMI-test	127

4 Projectscenario

1 Inleiding

Dit is het document waarin het projectscenario wordt beschreven. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de Opdrachtschrijving en voor het algemene Plan van Aanpak en is onderdeel van de IAD-methodiek. Het projectscenario bevat een ontwikkelscenario (inhoudelijke activiteiten; hoofdstuk 2) en een projectmanagementscenario (omringende procesbeheersing; hoofdstuk 3).

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
09-02-2005	1.0	Initiële versie.
01-03-2005	1.1	Aangevuld en veranderd naar aanleiding van ontwikkelscenario-workshop.
12-04-2005	1.2	Fasen, activiteiten, deelactiviteiten en producten uit dit document verwijderd. Deze zaken worden al in de opdrachtschrijving en in het algemene Plan van Aanpak vermeldt.

Tabel 1 Veranderingengeschiedenis.

2 Ontwikkelscenario

In dit hoofdstuk wordt invulling gegeven aan een aantal zaken die samen het ontwikkelscenario vormen. Zie de komende paragrafen voor wat dit zijn, wat ze inhouden, en hoe ze voor dit project zijn ingevuld.

Iteratiestrategie

Een belangrijk facet van het ontwikkelscenario is de keuze voor de iteratiestrategie. IAD bestaat uit de fasen Definitiestudie, Pilotontwikkeling en Invoering, en binnen het ontwikkelscenario is vast gelegd op welke wijze er door deze fasen heen wordt geïtereerd.

Er zijn vier hoofdvarianten te onderscheiden. Bij evolutionair ontwikkelen worden alle fasen in elke iteratie doorlopen. Bij incrementeel ontwikkelen wordt de fase Definitiestudie slechts éénmaal gevolgd. Bij big-bang-invoeren is het juist andersom: hier vindt alleen de invoeringsfase eenmaal plaats. Tot slot is er ook nog de combinatie

tussen de twee laatstgenoemden: incrementeel ontwikkelen. Hierbij wordt alleen de fase Pilotontwikkeling meermalen doorlopen.

Er is gekozen voor incrementeel ontwikkelen. De fase Definitiestudie wordt hierbij eenmalig doorlopen, de fase Pilotontwikkeling – nagelang het aantal tijdens de fase Definitiestudie gedefiniëerde pilots – meermalen met ook weer eenmalig de fase Invoering. Er is voor deze iteratiestrategie gekozen vanwege het feit dat het beoogde systeem goed van te voren kan worden gespecificeerd, de omvang van het systeem is te overzien en er weinig behoefte bestaat aan flexibiliteit tijdens het ontwikkelen.

Pilots en reikwijdte

In deze paragraaf wordt ingegaan op de globale pilotstrategie. Hierbij is zeer globaal een premature indeling gemaakt van het te maken systeem in pilots waardoor de manier waarop de specificaties in delen worden verdeeld, duidelijk wordt.

Het idee is nu om het systeem in te delen in twee pilots:

- 1 Een pilot met allen ontwikkeling rond een gestandaardiseerde manier van interactie met gebruikers, broncode-indelingen en communiceren met externe componenten als databases.
- 2 Een pilot voor het aanmaken en uitvoeren van selecties en voor het uitvoeren van data. De eventueel te ontwikkelen database is hier een onderdeel van.

Technieken en afbeeldingswijzen

Er zal tijdens dit gebruikt worden van:

- Databasemodelleringstechnieken;
- UML-tekentechnieken;
- interviewtechnieken.

Iteraties binnen fasen

Er zal binnen fase alleen worden geïtereerd door de softwarebouweenheden heen. Het is vermeldingswaardig dat de beoordeling en het testen op een hoger niveau zal gebeuren vanwege de te verwachte grootte van deze eenheden.

Parallele ontwikkeling

Er zal geen parallele ontwikkeling van pilots of softwarebouweenheden plaats vinden vanwege het eenledige ontwikkelteam en de onnodige hoeveelheid coördineerwerk die het met zich meebrengt.

Workshops

De te houden workshops zullen relatief kort zijn vanwege het relatief kleine project en het feit dat men er snel achter zal komen wat de wensen nou precies zijn.

Time-boxing

Bij de ontwikkeling van de pilotdelen zal gebruik gemaakt worden van time-boxing in de zin van het beperken van de tijd tegenover het inleveren van functionaliteit. Ook zal time-boxing worden gebruikt bij het beoordelen en testen van een pilot. Een deel van het beoordelen en testen kan ook nog terugkomen tijdens activiteit 5.4.

Mijlpalen

Tijdens het project zullen de volgende mijlpalen en mijlpaalproducten gehaald worden. In het Plan van Aanpak worden hier concrete tijdstippen aan gekoppeld.

Prototyping

Er zal op kleine schaal prototyping plaatsvinden bij het ontwerp van de gebruikersinterface.

3 Projectmanagementscenario

Voortgang en rapportage

Er zal wekelijks een gesprek plaats vinden met de begeleider vanuit het bedrijf.

Het afstudeerproject loopt van 7 februari t/m 10 juni voor een lengte van 18 weken. Rond de 10^e week zal er een mini-voortgangsverslag gemaakt worden en verstuurd via e-mail aan de examinatoren.

Terminologie

De terminologie is aan de ene kant gedeeltelijk afwijkend, maar aan de andere kant ook gedeeltelijk afwezig. Er zullen geen extra maatregelen genomen worden om verwarring te voorkomen aangezien de verwachting is dat het voldoende is dit te corrigeren danwel aan te vullen *as we go*.

5 Plan van Aanpak fase Definitiestudie

1 Inleiding

Dit is het op de fase Definitiestudie toegespitste Plan van Aanpak. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt als eerste in de fase Definitiestudie. Deze fase start na het algemene Plan van Aanpak en wordt vervolgd door de ontwikkelscenario-workshop. Voor een deel is het een verdere detaillering van het algemene Plan van Aanpak maar dit extra detail wordt alleen aangebracht voor deze fase.

In hoofdstuk 2 worden de in deze fase uit te voeren activiteiten uiteengezet. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat deze producten zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
10-02-2005	1.0	Initiële versie.
14-02-2005	1.1	De ontwikkelscenario-workshop en een eerste deel van het pilotplan-workshop vallen nu gelijk en op vrijdag in week 2. De planning is hierop aangepast.
01-03-2005	1.2	Het uitwerken van het selectieproces toegevoegd aan de activiteiten en de hieruit voortkomende producten.

Tabel 2 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

Onderstaand – globaal – alle te volgen fasen en activiteiten met de gekozen iteratiestrategie in het achterhoofd. Zie Tabel 3 voor een legenda om te kunnen zien waar elke activiteiten vandaan komt.

Legenda
IAD
GUIDE

Tabel 3 Legenda

- 3 Plan van Aanpak opstellen
- 4 Workshops voorbereiden
- 5 Ontwikkelsscenario-workshop: ontwikkelscenario bepalen
- 6 Pilotplanworkshop voorbereiden
- 7 Pilotplanworkshop:
 - 1 Systeemeisen definiëren
 - 1 Gebruikers en usability-eisen definiëren
 - 2 Systeemconcept bepalen

- 1 Taakdiagrammen
- 2 Taakscenario's
- 3 Use case diagram
- 4 Klassendiagram
- 5 Selectieproces
- 6 Stijlgids maken
- 3 Technische structuur beschouwen
- 4 Organisatorische inrichting beschouwen
- 5 Pilotplan opstellen

3 Planning

In dit hoofdstuk staat de planning van de fase Definitiestudie. Een aanzet hiervoor was al gegeven in het algemene Plan van Aanpak, maar in onderstaande Tabel 4 zijn ook de subactiviteiten opgenomen alsmede een planning die op de dag in plaats van op de week nauwkeurig is.

Activiteit	Weken																													
	2					3					4					5					6					7				
	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V
Plan van Aanpak opstellen																														
Workshops voorbereiden																														
Ontwikkelscenario-workshop																														
Pilotplanworkshop voorbereiden																														
Pilotplanworkshop																														
Systeemeisen																														
Gebruikers en usability-eisen																														
Systeemconcept																														
Taakdiagrammen																														
Taakscenario's																														
Use-case diagram																														
Klassendiagram																														
Selectieproces																														
Stijlgids																														
Technische structuur																														
Organisatorische inrichting																														
Pilotplan																														

Tabel 4 De planning.

4 Producten

In deze paragraaf worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- Plan van Aanpak.

Alle producten voortkomend uit 2 t/m 6, samengevat in het definitiestudierapport.

6 Voorbereiding workshops

1 Inleiding

Dit is de voorbereiding op de ontwikkelscenario-workshop en het eerste deel van de pilotplan-workshop. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Deze workshops zijn samengenomen vanwege de relatief kleine omvang van het project. Dit heeft namelijk als gevolg dat de overhead van twee afzonderlijke workshops groter is dan de extra tijd die gaat zitten in het tegelijkertijd voorbereiden en houden van deze twee workshops.

Dit document is een product van de tweede activiteit in de fase Definitiestudie. Het is voorafgegaan door het Plan van Aanpak specifiek aan de fase Definitiestudie en zal vervolgd worden door de workshops zelf.

Hoofdstuk 2 bevat de voorbereiding op de ontwikkelscenario-workshop: welke onderwerpen hier aan bod zullen komen. Hoofdstuk 3 bevat de voorbereiding op de pilotplan-workshop: welke onderwerpen hier aan bod zullen komen. Als voorzet op de pilotplan-workshop is in hoofdstuk 4 ook nog een aanzet te vinden tot het systeemconcept.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
15-02-2005	1.0	Initiële versie.

Tabel 5 Veranderingengeschiedenis.

2 Ontwikkelscenario-workshop

In dit hoofdstuk staan de ontwerpen die naar voren zullen komen in de ontwikkelscenario-workshop, vrijdag in week 2. Deze workshop heeft tot doel de reikwijdte en de context van het project te bepalen. Tevens worden andere kleinere beslissingen genomen betreffende bijvoorbeeld de iteratiestrategie.

Ontwikkelpoces

In deze paragraaf wordt een voorstel gedaan tot het te volgen ontwikkelproces. Er wordt ingegaan op de volgende belangrijke kenmerken: iteratiestrategie, pilots en time-boxing.

De gekozen ontwikkelmethodiek is IAD, wat staat voor *Iterative Application Development*. Het heeft drie fasen, namelijk: Definitiestudie, Pilotontwikkeling en Invoering. De methodiek is iteratief, en dat wil zeggen dat fasen meermalen telkens met een andere context doorlopen kunnen worden. Ook kunnen sommige activiteiten binnen een fase op hun beurt ook meerdere keren worden uitgevoerd. Er is gekozen voor de iteratiestrategie waarbij enkel de fase Pilotontwikkeling iteratief wordt uitgevoerd. De fase Definitiestudie en de fase Invoering komen daarbij dus slechts eenmalig aan bod (deze strategie ligt dicht bij RAD¹).

Tijdens de fase Definitiestudie worden alle systeemeisen verzameld en verdeeld over een aantal pilots. Een pilot is een coherente set functionaliteiten en elk van de pilots wordt in een afzonderlijke iteratie van de fase Pilotontwikkeling ontwikkeld. Elke pilot wordt hierbij ook nog verdeeld in een aantal pilotdelen.

Bij de ontwikkeling van pilotdelen zal gebruik worden gemaakt van time-boxing. De delen worden hierbij elk toegewezen aan een time-box gekoppeld aan een prioriteit. Aan een time-box is een vaste hoeveelheid tijd (en budget) gekoppeld en als tijd eerder opraakt dan is begroot, zullen de functionaliteiten met de laagste prioriteiten achterwege blijven. De basisprioriteiten zijn basis, comfort en luxe. Pilotdelen met de basisprioriteit dienen ten alle tijden binnen begrenzungen worden gerealiseerd. Wanneer dit niet gebeurt, is er sprake van een ernstige situatie. Van alle comfort-pilotdelen zullen er waarschijnlijk een paar wel en een paar niet worden gerealiseerd. Van pilotdelen die de luxe-prioriteit toegewezen hebben gekregen wordt normaliter niet verwacht dat ze binnen de gestelde tijd kunnen worden gerealiseerd.

Globale pilotstrategie

In deze paragraaf wordt ingegaan op de globale pilotstrategie. Hierbij wordt zeer globaal een premature indeling gemaakt van het te maken systeem in pilots waardoor de manier waarop de specificaties in delen worden verdeeld, duidelijk wordt.

Het idee is nu om het systeem in te delen in twee pilots:

¹ *Rapid Application Development*

- 8 Een pilot met allen ontwikkeling rond een gestandaardiseerde manier van interactie met gebruikers, broncode-indelingen en communiceren met externe componenten als databases.
- 9 Een pilot voor het aanmaken en uitvoeren van selecties en voor het uitvoeren van data. De eventueel te ontwikkelen database is hier een onderdeel van.

Zie hoofdstuk 4 voor een diepere beschrijving van de tweede en belangrijkste pilot en de functionaliteiten hiervan in de vorm van een systeemconcept.

Globale teststrategie

Naast de normale testactiviteiten die plaats zullen vinden tijdens de verschillende iteraties van de fase Pilotontwikkeling, zal er ook een SUMI-test uitgevoerd worden voor het meten van de gebruikersvriendelijkheid. De SUMI (*Software Usability Measurement Inventory*)-test is de defacto standaard voor het bepalen van de kwaliteit van software op het gebruikersvlak.

Onderwerpen voor workshop

In deze paragraaf staan een aantal onderwerpen voor de ontwikkelscenario-workshop. Het zal gaan over of er opmerkingen of onduidelijkheden bestaan over:

- het te volgen ontwikkelproces;
- de pilotstrategie (exclusief concrete indeling, dit komt in hoofdstuk 4 aan de orde);
- de teststrategie.

3 Pilotplan-workshop

In dit hoofdstuk staan de de onderwerpen die naar voren zullen komen in de pilotplan-workshop waarvan het eerste deel valt op vrijdag in week 2. Er wordt ingegaan op waar de workshop voor is, wat zijn producten zijn, en welke onderwerpen er tijdens de bijeenkomst zelf ter sprake dienen te komen. Deze workshop is een essentieel onderdeel van de ontwikkelcyclus want het is de grootste invoer voor de fase Definitiestudie: veel specificaties en functionaliteiten zullen op basis van de pilotplan-workshop worden vastgelegd. Er zal verder ingegaan worden op systeemeisen, het systeemconcept, de technische structuur en de organisatorische inrichting.

Systeemeisen

De systeemeisen zijn een aantal eisen aan het te bouwen systeem genoteerd in bedrijfsterminologie en die tevens geprioriteerd moeten zijn. De eisen moeten een juiste weergave zijn van de behoeften van de organisatie. De lijst van systeemeisen vormt de basis voor alle

ontwerpactiviteiten. Er zijn een aantal verschillende soorten systeemeisen:

- Basissysteemeisen: eisen waaraan het systeem moet voldoen door events, afhandelende processen en gebruikers-interfaces vast te stellen. De eisen moeten worden gecategoriseerd en geprioriteerd.
- Interface-eisen (bijvoorbeeld gebruikersvriendelijkheid of geschiktheid voor slechtzienden).
- Integriteits-eisen (betrouwbaarheid, accuratesse).
- Performance-eisen (bijvoorbeeld minimale verwerkingscapaciteit).
- Operationele eisen (kenmerken die verband houden met organisatieverandering, invoering en met kwaliteit).

Alle eisen moeten vervolgens in één lijst worden geprioritiseerd met vermelding van de impact voor gebruikers, vaardigheden van gebruikers, werkprocedures, interfaces, ontwikkeling, invoering, enzovoorts.

Systeemconcept

De systeemeisen staan voor een specificatie van de probleemstelling. Het systeemconcept is een voornamelijk beschrijvende weergave op globaal niveau van de oplossing. In dit concept staan de belangrijkste functionele delen en hun afhankelijkheden. Het wordt gemodelleerd vanuit het gezichtspunt van de gebruiker en wordt uitgedrukt in de vorm van taken die door een actor (iemand die een bepaalde functie vervult) binnen een bedrijfsproces worden uitgevoerd.

Om tot een systeemconcept te kunnen komen, moeten allereerst de volgende vragen worden beantwoord:

- Actoren bepalen: welke functionarissen in welke rollen voeren taken uit binnen het te beschouwen bedrijfsproces?
- Gebeurtenissen: onder welke voorwaarden begint een actor met het uitvoeren van een taak, welke informatie wordt gebruikt en wat is het resultaat?
- Proces: welke acties worden er (globaal) uitgevoerd voor het uitvoeren van een taak.

In hoofdstuk 4 staat een eerste aanzet van een systeemconcept.

Technische structuur

Hier worden de te verwachten systeemeisen afgezet tegen de technologische structuur. Vooralsnog is dit het uitgangspunt zoals eerder – min of meer op papier – vastgelegd: webgebaseerd, client-server, platformonafhankelijk, PHP & MySQL.

Organisatorische inrichting

De organisatorische inrichting geeft aan wat de impact is van de te definiëren pilots is. Voor dit project is hierbij alleen relevant wat de vereiste gebruikersdocumentatie moet worden.

Onderwerpen voor workshop

In deze paragraaf staan een aantal onderwerpen voor het eerste deel van de pilotplan-workshop:

- de systeemeisen;
- het systeemconcept (zie hoofdstuk 4);
- technische structuur;
- organisatorische inrichting.

4 Aanzet systeemconcept

In dit hoofdstuk staat een eerste aanzet tot het systeemconcept. Er wordt tekstueel beschreven hoe een basis van het systeem er (functioneel) uit zou kunnen komen te zien. Zoals in de globale pilotstrategie in hoofdstuk 4 al is aangegeven, is het idee nu het systeem op te delen in een pilot voor het raamwerk en een pilot voor het aanmaken en uitvoeren van selecties en het uitvoeren van data. Dit hoofdstuk gaat alleen maar in op deze tweede pilot.

Terminologie

In deze paragraaf staan een aantal termen, zodat er geen interpretatieverschillen kunnen ontstaan vanwege verkeerd gebruik van termen. Zie Tabel 6.

Term	Omschrijving
profile	Een deelnemer aan een actie. Wordt geïdentificeerd door middel van <i>profile_id</i> .
profile_id	Identificatie-data van een <i>profile</i> .

Tabel 6 Terminologie

Selectie van profiles

Hieronder staat een schematische invulling van de selectiemogelijkheden zoals deze gerealiseerd zou kunnen worden. Het is slechts een notatiewijze om aan te geven hoe een selectie er inhoudelijk uit kan zien; de grafische gebruikersinterface is hier nog niet gespecificeerd. Uit de selectie komen een aantal profiles.

Op de schematische invulling zijn de volgende notatiewijzen van toepassing, zie Tabel 7.

Element	Houdt het volgende in
<i>cursief:</i>	Definitie
<i>cursief</i>	Variabel in te vullen, eventueel volgens een definitie.

<code>{ element_1 element_2 }</code>	Een van de elementen moet gekozen worden.
<code>[optie]</code>	Optionele optie.
<code>[optie 1 optie 2]</code>	Een van de opties <i>mag</i> gekozen worden.
<code>[...]</code>	Optionele herhaling van het laatste <code>[]</code> -blok voor een onbepaald aantal keer.
<code>(foobar)</code>	Groepering, heeft verder weinig betekenis.

Tabel 7 Notatiewijze schematische weergave selectiemogelijkheden.

```
selectie:
{voorwaarde | (selectie)}
[logische_operator {voorwaarde | (selectie)}]
[...]
```

```
voorwaarde:
{profileattribuut | variabele}
[niet]
{
    gelijk aan waarde [of waarde] [...] |
    ongelijk aan waarde [en waarde] [...] |
    kleiner dan [of gelijk aan] waarde |
    groter dan [of gelijk aan] waarde |
    matcht reguliere_expressie
}
```

```
profileattribuut:
```

Elk willekeurig attribuut van een profile.

```
variabele:
```

Elke willekeurige variabele.

```
logische_operator:
[niet] {en | of | xof}
```

Uitbreidingen

Momenteel moet er ook gedacht worden aan faciliteiten om bijvoorbeeld de leeftijd van iemand te berekenen, dus: berekende profileattributen en berekende variabelen, misschien door de gebruiker zelf aan te maken.

Uitvoer

Aan de hand van de profiles die uit de selectie komen, kunnen een aantal dingen worden gedaan:

- het aantal ervan weergeven;
- van elk profile één of meer profileattributen weergeven;
- van elk profile één of meer variabelen weergeven;

Uitbreidingen

Er moet onder meer nog nagedacht worden over hoe er met sponsorvragen moet worden omgegaan.

7 Definitiestudie

1 Inleiding

Dit is de definitiestudie, het rapport wat voortkomt uit de fase Definitiestudie. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt als derde product naar voren in de fase Definitiestudie. Het komt na de ontwikkelscenario-workshop en na het eerste deel van de pilotplan-workshop en zal worden aangevuld dan wel bijgewerkt aan de hand van de volgende delen van de pilotplan-workshop. De inhoud van deze definitiestudie zal in de volgende fase worden gebruikt als basis voor de verschillende pilotontwikkelplannen en waar nodig daar nog worden aangevuld en uitgebreid.

Hoofdstuk 2 bevat de systeemeisen. Deze systeemeisen zijn een weergave van de behoeften van de organisatie en vormen een basis voor alle ontwerpactiviteiten. Hoofdstuk 3 bevat het systeemconcept; een oplossing op globaal niveau. Hoofdstuk 4 bevat de technische structuur waarin de technische aspecten van het project worden beschouwd en gaat in op de organisatorische inrichting; hier wordt vastgelegd wat de impact is op de organisatie. Tot slot staat in hoofdstuk 5 het pilotplan.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
21-02-2005	1.0	Initiële versie.
15-03-2005	1.1	Aanpassingen naar aanleiding van tweede deel van de pilotplanworkshop. Urenbegroting van de pilots en pilotdelen aangepast naar ervaring.
06-04-2005	1.2	Berekende variabelen zijn overal variabelen in het algemeen geworden.
12-04-2005	1.3	Uitbreiding van de omschrijving der klassen van het klassendiagram. Onder andere een betere uitleg over een aantal attributen en wat er moet gebeuren als ze leeg zijn.
19-05-2005	1.4	Associatieklasse 'Resultaat' is veranderd in een normale klasse.
30-05-2005	1.5	Pilotdelen van de pilot 'Resultaatsetbeheer' aangepast op ontwikkelingen tijdens het bouwen van de voorgaande pilot: er is geen onderscheid meer tussen nabewerkte en rauwe resultaatsets.

Tabel 8 Veranderingengeschiedenis.

2 **Systeemeisen**

In dit hoofdstuk staan de systeemeisen. Deze systeemeisen zijn een weergave van de behoeften van de organisatie en vormen een basis voor alle ontwerpactiviteiten. Er wordt allereerst ingegaan op de basissysteemeisen waarin wordt vastgelegd welke gebeurtenissen door welke afhandelende processen worden afgehandeld. Vervolgens komen alle overige eisen aan bod waarna aan elke eis een prioriteit wordt gegeven. Tot slot worden alle gebruikergroepen, kenmerken hiervan en bijbehorende usability-eisen vastgelegd.

2.1 **Basissysteemeisen**

In deze paragraaf staat een hiërarchisch overzicht van de basissysteemeisen. Deze bestaan uit wat het systeem moet kunnen. Zie §3.4 voor een uitleg en definitie van gebruikersobjecten.

Het systeem moet het volgende kunnen of gebruikers moeten het volgende ermee kunnen uitvoeren:

10 Selecties:

- 1 toevoegen;
- 2 bewerken;
- 3 verwijderen;
- 4 activeren;
- 5 inplannen;
- 6 doorgeven aan iemand anders.

11 Resultaatsets:

- 1 bekijken;
- 2 downloaden;
- 3 verwijderen;
- 4 doorgeven aan iemand anders.

12 Variabelen:

- 1 toevoegen;
- 2 bewerken;
- 3 verwijderen.

13 Gebruikers:

- 1 toevoegen;
- 2 bewerken;
- 3 verwijderen.

2.2 **Overige eisen**

In deze paragraaf worden alle overige eisen vastgelegd. Dit zijn alle eisen in de volgende categoriën: integriteits-eisen, performance-eisen & operationele eisen. De numerieke getallen na elke eis vergemakkelijken de verwijzing vanuit §2.3.

Integriteits-eisen zijn de eisen die een bedrage leveren aan de betrouwbaarheid en accuatesse. Het systeem zal de gebruiker om een bevestiging moeten vragen voordat deze op het punt staat een

ingrijpende actie uit te voeren (5.1). Daarnaast dienen er controles op gebruikersinvoer aanwezig te zijn om onduidelijkheden terug te dringen (5.2).

Performance-eisen zijn de eisen die te maken hebben met de maximale verwerkingsmogelijkheden van het systeem. Het systeem moet voldoende capaciteit hebben de vraag vanuit de verschillende gebruikersgroepen aan te kunnen (6).

Operationele eisen zijn de eisen die te maken hebben met organisatieverandering, invoering en kwaliteit. Het systeem zal gecentraliseerd ingevoerd moeten worden op één geografische locatie (7.1), en het gebruik ervan zal in principe ook beperkt mogen blijven tot één netwerk (7.2).

2.3 Prioriteiten

In deze paragraaf staan alle eisen uit voorgaande paragrafen op volgorde van prioriteit.

- 1.1 t/m 1.4: Selecties toevoegen, bewerken, verwijderen & uitvoeren.
- 3: Variabelen toevoegen, bewerken en verwijderen.
- 6: Voldoende verwerkingscapaciteit.
- 2.1 t/m 2.3: Resultaatsets bekijken, downloaden & verwijderen.
- 4: Gebruikers toevoegen, bewerken en verwijderen.
- 5: Bevestigingen & invoer-controles.
- 2.4: Resultaatsets doorgeven aan iemand anders.
- 1.6: Selecties doorgeven aan iemand anders.
- 1.5: Selecties inplannen.

Eis 7 is eigenlijk geen eis, maar meer een niet beperkende factor.

2.4 Gebruikers & usability-eisen

Er zijn vier gebruikersgroepen: Operations, Sales, Systeem-administratie en Klanten. Elk van de gebruikersgroepen wordt apart en in detail langsgegaan. Aan het einde bevinden zich nog een samenvatting en usability-eisen voortkomend uit deze gebruikersgroep-specificaties.

Operations

Type gebruiker

Hoofdzakelijk direct, maar een enkele keer ook indirect als iemand uit een andere gebruikersgroep wordt gevraagd een bepaalde selectie aan te maken en uit te voeren.

Ervaring

De gebruikers in deze gebruikersgroep zijn gevorderden.

Frequentie van gebruik

Het systeem zal van enkele keren per dag tot enkele keren per uur in piektijden gebruikt worden.

Verplicht gebruik

Het systeem is voor gebruikers uit deze gebruikersgroep in principe verplicht in gebruik, maar er kan natuurlijk altijd teruggegrepen worden naar de huidige methode van het uitvoeren van selecties. Omdat deze oude methode duidelijk zo zijn nadelen kent, zal de keuze voor gebruik van dit systeem snel vallen, mede gezien de voordelen die het met zich meebrengt op het vlak van integratie en communicatie met andere gebruikers.

Aanwezige computerervaring en -kennis

Er is een gemiddelde hoeveelheid computerervaring en -kennis aanwezig. Een deel van deze computerervaring en -kennis is afkomstig van het gebruik van het andere systeem en deelname aan het evaluatieproces van de huidige informatiesystemen bij E2Ma.

Andere systemen

Er is ook een E2Ma backoffice dat parallel gebruikt zal moeten worden door enkele leden van deze gebruikersgroep.

Motivatie voor gebruik

Ten opzichte van de momenteel bestaande werkwijze voor het selecteren van data, brengt dit systeem voordelen met zich mee op de gebieden van snelheid en eenvoud. De vraag naar selecties zelf komen voort uit specifieke vraag van E2Ma zelf, iemand van de sales-afdeling of een klant.

Aantal gebruikers

Momenteel is deze gebruikersgroep gevuld met twee à drie personen en het is niet de verwachting dat dit aantal de komende jaren significant zal gaan stijgen.

Sales

Type gebruiker

Iemand uit deze gebruikersgroep zal het systeem op twee verschillende manieren kunnen gebruiken: direct als indirect. Direct voor de eenvoudigere selecties en indirect als de selecties uitgevoerd moeten worden door iemand uit de gebruikersgroep Operations.

Ervaring

De gebruikers in deze gebruikersgroep variëren van beginners tot gevorderden.

Frequentie van gebruik

Voor zowel het direct als indirecte gebruik geldt dat het systeem enkele keren per week gebruikt zal gaan worden.

Verplicht gebruik

Het systeem is voor gebruikers uit deze gebruikersgroep niet verplicht in gebruik. Voor de wat simpelere selecties kunnen deze gebruikers ook een beroep doen op mensen uit de gebruikersgroep Operations. Wel drukt dit de algehele efficiëntie op de werkvloer, en worden niet alle mogelijkheden van het systeem benut, dus dit zijn goede redenen voor deze gebruikers er toch gebruik van te maken.

Aanwezige computerervaring en -kennis

Er is weinig tot een gemiddelde hoeveelheid computerervaring en -kennis aanwezig.

Andere systemen

Er hoeft geen gebruik gemaakt te worden van andere systemen.

Motivatie voor gebruik

Ten opzichte van de huidige werkwijze waarin deze gebruikers altijd een beroep doen uit gebruikers uit de groep Operations, zullen deze gebruikers kleine selecties uit de vraag van klanten in de nieuwe situatie zelf kunnen vervullen.

Aantal gebruikers

Momenteel zal deze gebruikersgroep gevuld worden met 4 à 6 personen, en het is de verwachting dat dit aantal de komende jaren nog met enkele personen zal toenemen.

Systeemadministratie

Type gebruiker

De gebruikers in deze gebruikersgroep zijn direct; ze zijn direct in contact met de gebruikersinterface.

Ervaring

De gebruikers in deze gebruikersgroep zijn gevorderden tot experts.

Frequentie van gebruik

Het systeem zal enkele keren per week worden gebruikt, maar deze frequentie kan vlak na invoering van het systeem of na introductie van een nieuwe gebruiker tijdelijk wat hoger liggen.

Verplicht gebruik

Het systeem is voor gebruikers in deze gebruikersgroep verplicht in gebruik, want als het systeem niet wordt gebruikt kunnen de gebruikers uit andere gebruikersgroepen er in verloop van tijd ook niet optimaal mee werken.

Aanwezige computerervaring en -kennis

Bij alle leden van deze gebruikersgroep is veel computerervaring en -kennis aanwezig.

Andere systemen

Er is ook een E2Ma backoffice dat parallel loopt en gebruikt kan worden.

Motivatie voor gebruik

Vereist om gebruikers in andere gebruikersgroepen ook de gelegenheid te geven van dit systeem gebruik te maken en dit optimaal te kunnen blijven doen.

Aantal gebruikers

Momenteel is deze gebruikersgroep gevuld met een à twee personen. Het is de verwachting dat dit aantal de komende jaren met enkele personen zal gaan stijgen.

Klanten

Type gebruiker

Gebruikers in deze gebruikersgroep zijn vaak indirect, maar soms ook direct. De gebruiker is indirect als een klant de informatievraag neelegt bij iemand van de sales-afdeling of iemand uit de gebruikersgroep Operations. Het systeem wordt direct gebruikt als een resultaatset door iemand uit de overige gebruikersgroepen aan de klant is doorgegeven.

Ervaring

De gebruikers in deze gebruikersgroep zijn beginner tot gevorderde.

Frequentie van gebruik

Door elke gebruiker in deze gebruikersgroep zal het systeem enkele keren per week direct en enkele keren per maand indirect worden gebruikt.

Verplicht gebruik

Er zijn geen alternatieven op indirect gebruik anders dan direct gebruik. Voor direct gebruik zijn er geen alternatieven.

Aanwezige computerervaring en -kennis

De aanwezige computerervaring en -kennis van de gebruikers binnen deze gebruikersgroep is variabel.

Andere systemen

Er zijn geen andere systemen waar de gebruikers van deze gebruikersgroep gebruik van kunnen maken.

Motivatie voor gebruik

Voor het indirecte gebruik veranderd er niets, voor het directe gebruik levert het een voordeel op t.o.v. indirect gebruik in de gebieden van efficiëntie e.d. (dus in de situatie van dat iemand uit de andere gebruikersgroepen resultaatsets aan de klant doorgeeft).

Aantal gebruikers

Het aantal gebruikers in deze gebruikersgroep zal variëren.

Samenvatting

Zie Tabel 9 voor de samenvatting van de groepsspecificaties.

	Operations	Sales	Systeemadministratie	Klanten
Type gebruiker	Direct	Direct / indirect	Direct	Indirect
Ervaring	Gevorderde	Beginner ~ gevorderde	Gevorderde ~ expert	Beginner ~ gevorderde
Frequentie	enkele/dag	enkele/week	enkele/week	enkele/week
Verplicht	Ja	Nee	Ja	Ja
Aanwezige computer-ervaring en -kennis.	Gemiddeld	Weinig ~ gemiddeld	Veel	Variëert
Andere systemen	Ja	Nee	Ja	Nee
Motivatie	Veel	Gemiddeld	Veel	Gemiddeld
Aantal gebruikers	2 à 3 (+±0)	4 à 6 (+±4)	1 à 2 (+±2)	Variëert

Tabel 9 Samenvatting van de groepsspecificaties.

Usability-eisen

In Tabel 10 staan de usability-eisen die aan de hand van bovenstaande specificaties zijn opgesteld. Alle eisen zijn afkomstig uit de SUMI-vragenlijst. De SUMI (*Software Usability Measurement Inventory*)-test is de defacto standaard voor het bepalen van de kwaliteit van software op het gebruikersvlak.

Attribuut	Minimaal acceptabele waarde	Doelwaarde	Maximaal haalbare
Efficiëntie	40	55	70
Gevoel	30	45	65
Assistentie	30	50	65
Controle	40	50	65
Leerbaarheid	30	45	65

Tabel 10 Usability-eisen.

3 Systeemconcept

In dit hoofdstuk staat het systeemconcept. Het is een voornamelijk beschrijvende weergave op globaal niveau van de oplossing. Het systeemconcept wordt gemodelleerd vanuit het gezichtspunt van de gebruiker. Als eerste komen de taakdiagrammen voorbij welke laten zien welke taken er (op globaal niveau) voor de gebruiker uit te voeren zijn. Hieruit worden vervolgens een aantal taakscenario's gehaald om daarna vervolgd te worden tot een use-case diagram. Alle objecten worden hierna ook nog vastgelegd in een klassendiagram, een omschrijving van het selectieproces volgt hierop en de uiteindelijke stijl van de grafische gebruikersinterface is tot slot te vinden in de stijlgids.

3.1

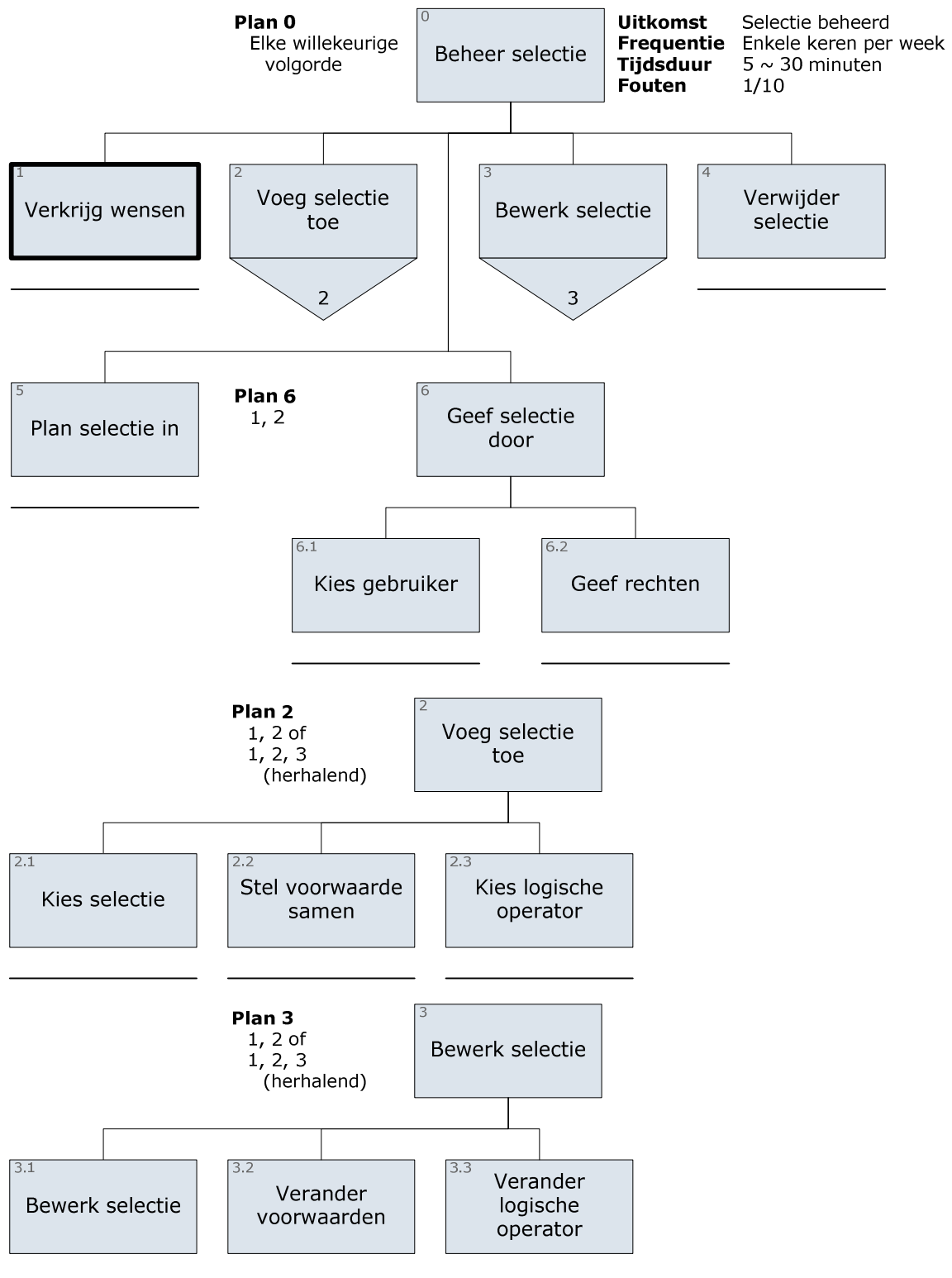
Taakdiagrammen

In deze paragraaf staan taakdiagrammen welke laten zien welke taken er (op globaal niveau) voor de gebruiker uit te voeren zijn.

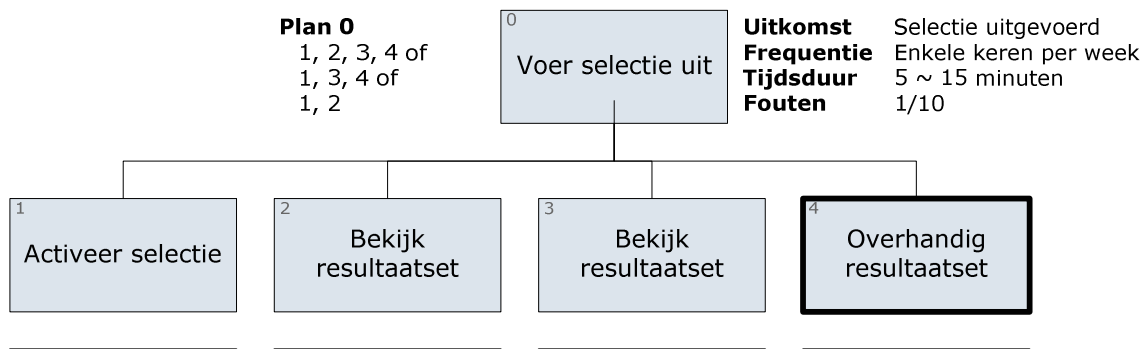
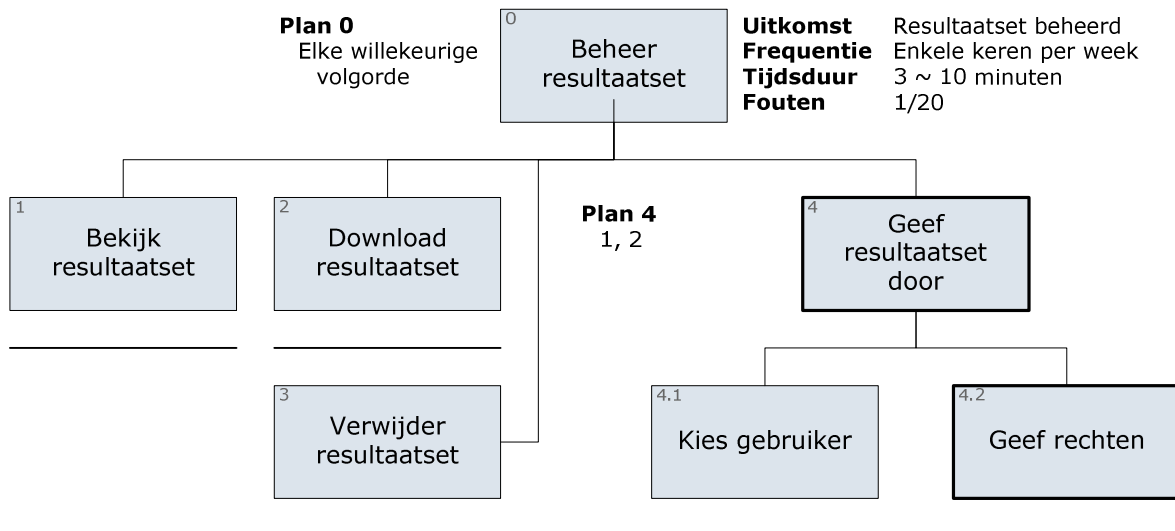
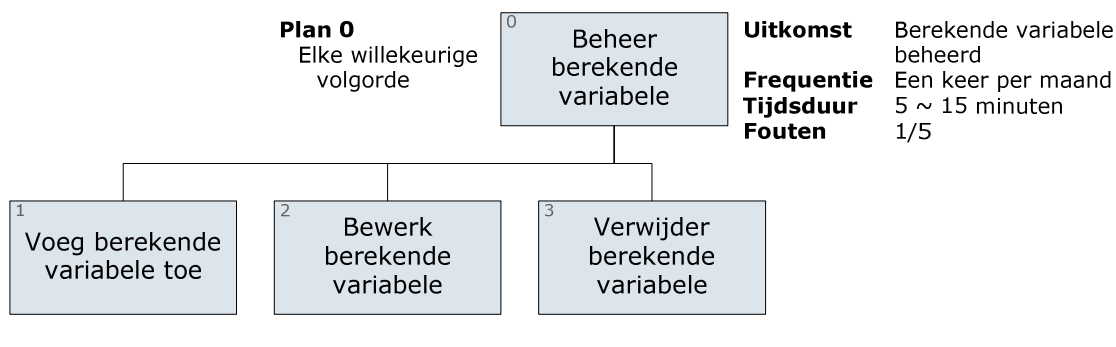
Er zijn vijf hoofdtaken te onderscheiden:

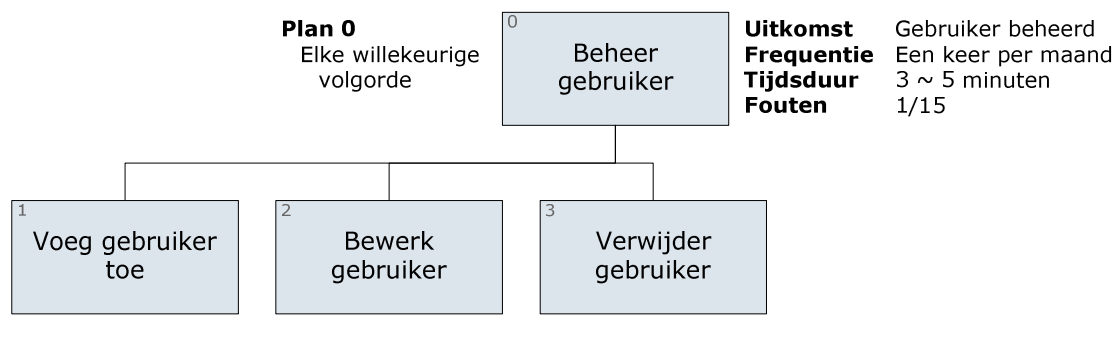
- beheren van selectie;
- uitvoeren van selectie;
- beheren van selectie;
- beheren van variabelen;
- beheren van gebruikers.

Alle taken worden door automatisering ondersteund, met uitzondering van de taken die dikomlijnd zijn.



Figuur 1 Taakdiagram 1: beheer selectie.

**Figuur 2** Taakdiagram 2: voer selectie uit.**Figuur 3** Taakdiagram 3: beheer resultaatset.**Figuur 4** Taakdiagram 4: beheer variabele.



Figuur 5 Taakdiagram 5: beheer gebruiker.

Zie §3.4 voor een uitleg en definitie van gebruikersobjecten.

3.2 Taakscenario's

In deze paragraaf staan een aantal taakscenario's. Het zijn voorbeeldtaken met wat extra achtergrondgegevens die later bij het testen gebruikt kunnen worden.

Selectie uitvoeren

Bij het uitvoeren van een selectie wordt het systeem gebruikt om aan de hand van enerzijds in de databases aanwezige data en anderzijds een al aangemaakte selectie, te komen tot een resultaatset met hierin een aantal gegevens. Het doel van dit scenario is ook het uitlezen van de resultaatset en overhandigen aan een klant, maar niet voor deze is bekeken.

De volgende stappen worden achtereenvolgens door diegene die dit scenario volgt, uitgevoerd:

- Selectie activeren: er moet een reeds aangemaakte selectie worden gekozen en geactiveerd. Na activatie maakt het systeem onderhuids een resultaatset aan.
- Bekijk de resultaatset: de resultaatset afkomstig van het activeren van de selectie kan nu worden bekeken. In geval van veel data zal deze verspreid zijn over meerdere pagina's wat het noodzakelijk maakt nog wat extra te klikken om alle data te kunnen doorlopen.
- Download de resultaatset: na het bekijken kan de resultaatset worden gedownload in een nader te specificeren formaat. De gebruiker kan dit bestand openen met een hiervoor geschikt programma of ergens opslaan.
- Overhandig het resultaat door de klant wie de informatie heeft aangevraagd het gedownloade bestand te doen toekomen.

Resultaatset doorgeven

Bij het doorgeven van een resultaatset wordt deze beschikbaar gemaakt aan een andere gebruiker ('ontvanger'). Bij het doorgeven

moeten tevens rechten worden toegewezen aan de resultaatset, om er alleen al van aan te geven dat de ontvanger deze mag inzien.

De volgende stappen worden achtereenvolgens door diegene die dit scenario volgt, uitgevoerd:

- Resultaatset doorgeven: de door te geven resultaatset wordt geselecteerd, en hiervan wordt aangegeven dat deze doorgegeven moet worden.
- Gebruiker kiezen: de ontvanger wordt gekozen.
- Rechten geven: de ontvanger ontvangt tezamen met de resultaatset een aantal rechten, waaronder dus of deze bekeken mag worden.

Selectie bewerken

Bij het bewerken van een selectie kan de inhoud hiervan worden aangepast. Zoals in §3.4 is aangegeven, bestaat een selectie uit een of meer voorwaarden en/of andere selecties gekoppeld met een logische operator. Bij het bewerken van een selectie kan deze combinatie dus worden aangepast.

Voor dit scenario gaan we uit van de volgende twee bestaande selecties:

- Selectie x : Selectie y en voorwaarde x .
- Selectie y : voorwaarde y of voorwaarde z .

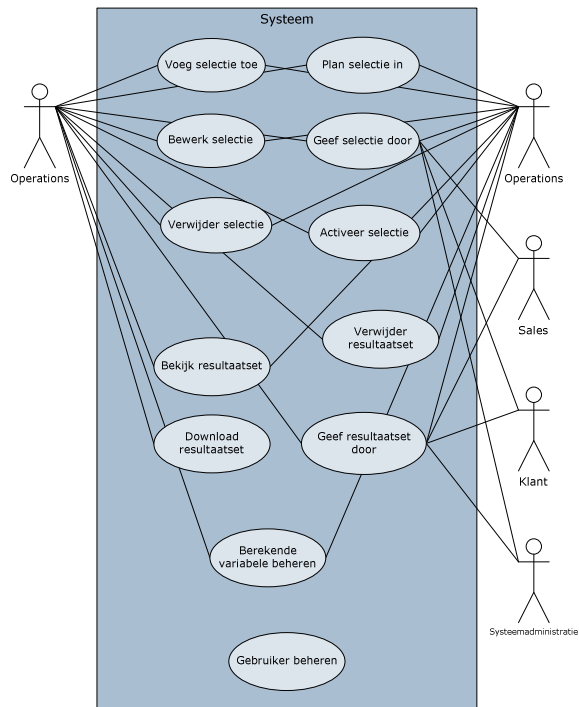
De bedoeling is dat van y (tevens een onderdeel van x) de logische operator wordt veranderd in een 'en', en z in a . Van x zelf moet x veranderen in w .

De volgende stappen worden achtereenvolgens door diegene die dit scenario volgt, uitgevoerd:

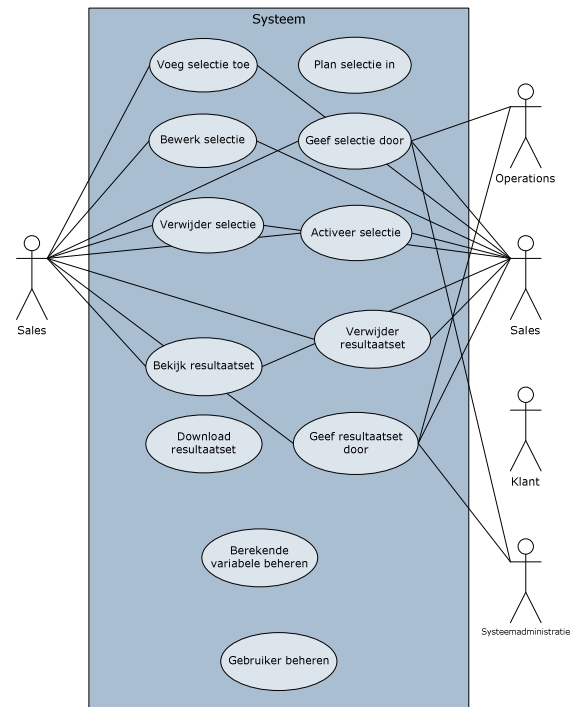
- x kiezen: de aan te passen selectie wordt geselecteerd, en hiervan wordt aangegeven dat deze doorgegeven moet worden.
- Doornavigeren naar y : omdat y een onderdeel is van x kan deze via een snelkoppeling worden bijgewerkt.
- z verwijderen en a toevoegen.
- logische operator tussen y en a instellen op 'en'.
- Terugnavigeren naar x .
- x verwijderen en w toevoegen.

3.3 Use-case diagram

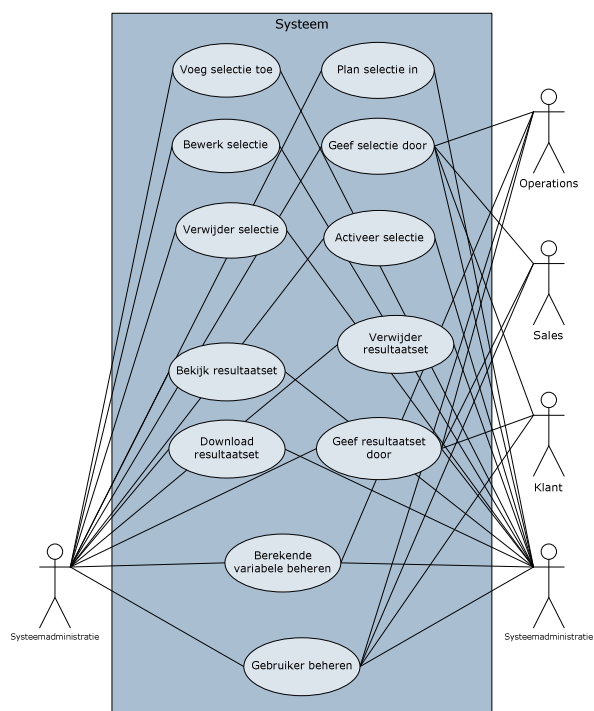
In deze paragraaf staat het use-case diagram. Het geeft aan welke gebruikersgroep welke taken mag uitvoeren. Voor de duidelijkheid is deze opgesplitst in vier delen, namelijk voor elke gebruikersgroep eentje. Zie Figuur 6 tot en met Figuur 9.



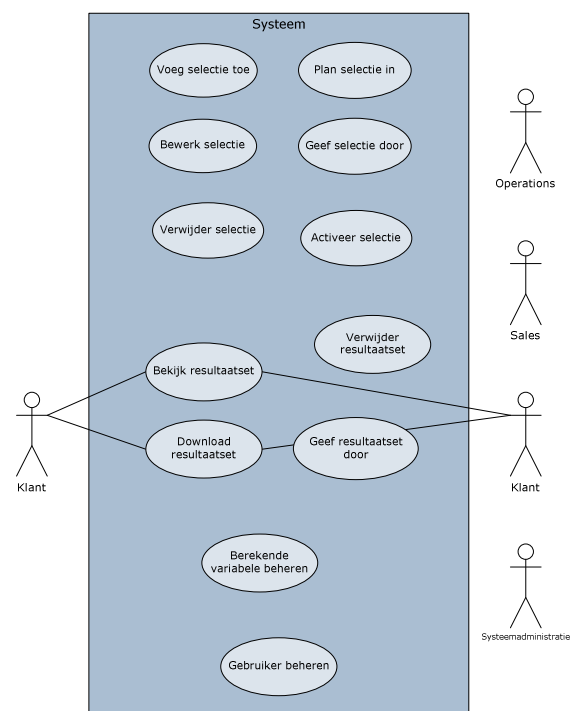
Figuur 6 Use-case diagram van gebruikersgroep Operations.



Figuur 7 Use-case diagram van gebruikersgroep Sales.



Figuur 8 Use-case diagram van gebruikersgroep Systeemadministratie.



Figuur 9 Use-case diagram van gebruikersgroep Klant.

Hier zijn een aantal opmerkingen bij te plaatsen:

- Dit zijn maximale mogelijkheden voor iemand uit een bepaalde gebruikersgroep. Door middel van het kunnen instellen van rechten op een gebruiker (onder andere door iemand van de gebruikersgroep Systeembeheer, maar ook bij het doorgeven van

een selectie of resultaatset) kunnen de mogelijkheden nog worden beperkt. Maar: ook deze maximale mogelijkheden zijn bij te werken door manipulatie van de objecten uit de klasse Gebruikersgroepen.

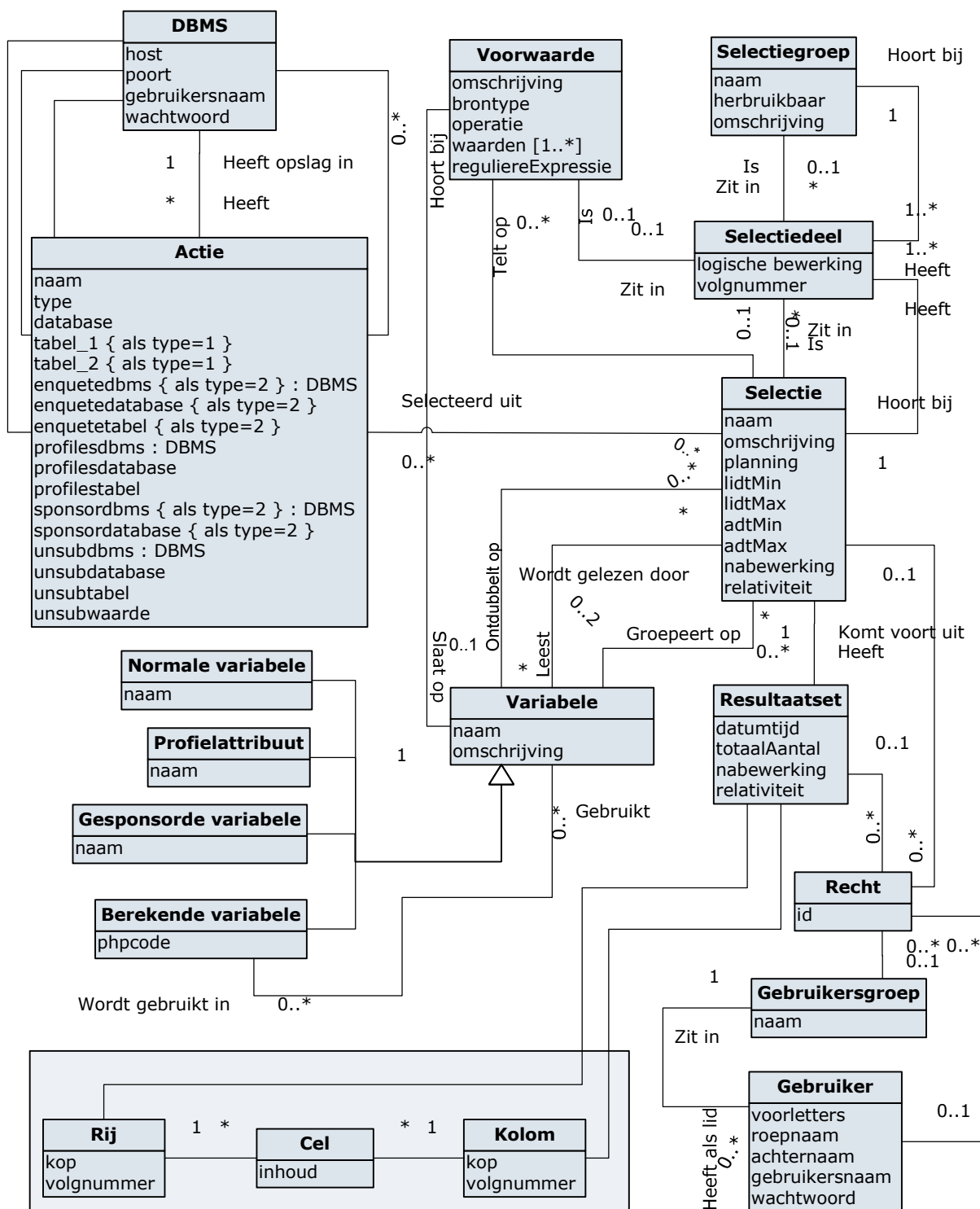
- Zoals blijkt uit Figuur 7, kan iemand uit de gebruikersgroep Sales een resultaatset niet downloaden maar alleen maar inzien. Een klant kan een resultaatset echter wel downloaden. Dit is zo gedaan omdat mensen van Sales deze functionaliteiten niet nodig hebben. Klanten daarentegen kunnen weer geen acties aanmaken of uitvoeren.
- Doorgeef-acties (het doorgeven van selecties of resultaatsets aan derden) hebben uitwerking op alle ontvangende actoren.
- Inplannen heeft uitwerking op een voor de eenvoud weggelaten actor, namelijk een stukje software in de rol van een taakplanner. Deze taakplanner zorgt voor de coördinatie van ingeplande acties, en activeert deze - als initiërende actor - vervolgens op gezette tijden.

3.4

Klassendiagram

In deze paragraaf staat het klassendiagram volgens UML. Het geeft aan welke objecten er te onderscheiden zijn voor het systeem. Niet alle objecten komen voor de eindgebruiker op het scherm.

Dit klassendiagram zal o.a. als basis gaan dienen voor de databasestructuur.



Figuur 10 Het klassendiagram.

Omschrijving der objecten

In dit deel wordt van alle objecten een omschrijving gegeven.

DBMS

De objecten van deze klasse representeren elk een Database Management System. In deze objecten worden de gegevens

bijgehouden die het voor het systeem mogelijk maken fysiek te verbinden met deze database-servers. Vooralsnog zal het voldoende zijn te werken met maar één instantie, maar voor de toekomst kan dit nog wel eens nodig worden.

Alle attributen zijn optioneel. Voor host wordt `localhost` als default genomen. Als de poort niet bekend is, wordt hier `3306` voor gebruikt. Zonder gebruikersnaam wordt er ingelogd als `root` en eventueel zonder wachtwoord.

Actie

De objecten van deze klasse representeren elk een Actie zoals 'Prij reis' en 'Win een droomreis'. Er is plaats gemaakt voor de twee laatste technologieën, type 1 (generatie 2) en type 2 (F2, generatie 3).

Standaard heeft een Actie één koppeling met een DBMS, maar deze kan voor verschillende zaken worden overschreven. Dit is het geval voor de DBMS voor de enquête-informatie, de profiles-data, de sponsortabel en de unsubtabel. Indien deze en bijbehorende attributen niet worden aangegeven, wordt uitgegaan van de standaardlocaties. Voor de unsub kan ook worden aangegeven wat de waarde kan zijn om een deelnemer als uitgeschreven te kunnen beschouwen.

Het attribuut `enquetetabel` is een prefix waar 'questions' en 'answers' achter wordt geplakt om achter een variabele-naam te kunnen komen.

Selectie

De objecten van deze klasse representeren elk een Selectie. Een selectie is een combinatie van andere selecties, voorwaarden en/of selectiegroepen aan elkaar gekoppeld door middel van selectiedelen, die uitkomen op een aantal gegevens van een aantal deelnemers.

Deze gegevens van deelnemers kunnen als nabewerking geteld en/of gegroepeerd worden. Door te groeperen op een variabele wordt het aantal voorkomens van een gelijkblijvende variabele geteld. Door te tellen wordt het aantal keer dat een voorwaarde *matched*, geteld.

Optioneel kan ook een alternatieve variabele aangegeven worden waarop ontdubbeld wordt. Normaal gesproken is dit op profileid, maar via deze weg kan er bijvoorbeeld gekozen te worden te ontdubbelen op postcode. Indien nodig wordt altijd de nieuwste waarde van een variabele verkozen boven de oudere.

Selectiedeel

De objecten van deze klasse representeren een deel van een selectie. Selectiedelen zijn t.o.v. elkaar verbonden met een logische bewerking (en, of, xof en niet) en verwijzen naar een voorwaarde, selectiegroep of andere selectie.

Selectiegroep

De objecten van deze klasse representeren een groep selectiedelen. Net als elke selectie, is een selectiegroep een combinatie van andere selecties, voorwaarden en/of selectiegroepen die aan elkaar gekoppeld zijn door middel van selectiedelen, die uitkomen op een aantal gegevens van een aantal deelnemers.

Voorwaarde

De objecten van deze klasse representeren een koppeling tussen een selectiedeel, één variabele en een of meer waarden. `vartype` is het type van de variabele die vergeleken moet worden, zoals datum of tijd. De operatie is bijvoorbeeld 'gelijk aan' of 'kleiner dan'.

Variabele

De objecten van deze klasse representeren elk een gegeven uit de database. Er zijn drie subtypen van dit object:

- Normaal: een verwijzing naar een variabele zoals een gebruiker deze heeft ingevuld in de verschillende enquêtes.
- Profiel: een verwijzing naar een gegeven van een deelnemer, zoals voornaam of postcode.
- Gesponsord: een variabele afkomstig van een gesponsorde vraag.
- Berekend: een berekening met één of meerdere andere variabelen, uitgevoerd door middel van het op kunnen geven van kant en klare PHP-scripts. Omdat een berekende variabele kan verwijzen naar een andere berekende variabele, moet bij de ontwikkeling gecontroleerd worden of er geen kringverwijzing ontstaat.

Resultaatset

De objecten van deze klasse representeren elk een product van een activatie van een selectie. Indien er geen nabewerking bestaat, vormen ze samen met de waarden in de klasse Resultaat waarin de daadwerkelijke resultaten staan. Indien er wel een nabewerking is geweest, komen de gegevens te staan in objecten van de klassen Rij, Kolom en Cel. Aan de hand van het totaal aantal resultaten kunnen, indien relativiteit wordt verlangd, de percentages worden uitgerekend.

Gebruiker

De objecten van deze klasse representeren elk een gebruiker van het systeem. Een gebruiker is iemand die gebruik maakt van de selectietool. Wat een gebruiker mag (met eventueel een object) wordt door de volgende drie niveaus bepaald:

- Per gebruikersgroep zijn vast een aantal rechten vastgelegd. Dit zijn de standaardrechten van deze groep. Per recht zijn hiervoor een aantal Recht-objecten met elk enkel een associatie met een Gebruikersgroep-object.

- Per gebruiker kunnen bepaalde rechten van de gebruikersgroep in beide richtingen worden overschreven. Een gebruiker kan dus meer rechten krijgen dan standaard het geval is voor alle leden van die gebruikersgroep, maar ook minder. Per recht zijn hiervoor een aantal Recht-objecten met elk enkel een associatie met een Gebruiker-object.
- Sommige rechten kunnen ook op objectniveau worden overschreven wat gebruikt wordt bij het doorgeven van Selectie- of Resultaatset-objecten. Per recht is hiervoor een associatie met het desbetreffende object. Hierbij kan de doorgever aangeven wat diegene aan wie het object wordt doorgegeven, er mee kan. Hierbij geldt altijd dat er nooit een recht vergeven kan worden die de doorgever zelf ook niet heeft.
- Rechten die gelden voor objecten die ook andere objecten 'in' zich hebben, worden door deze subobjecten overgeërft.

Gebruikersgroep

De objecten van deze klasse representeren elk een gebruikersgroep. Elke gebruiker valt binnen één gebruikersgroep. Het is een hulpmiddel voor het bijhouden van de rechten van de gebruikers.

Recht

De objecten van deze klasse representeren elk een recht; iets wat een gebruiker of leden van een gebruikergroep mogen doen (met een object). Een onderscheid tussen de twee typen selecties is hierbij inbegrepen. De betekenis van het attribuut 'id' is een getal welke een bepaald recht representeert. Welk recht overeenkomt met welke waarden, wordt in de fase Pilotontwikkeling omschreven. Maar zie de omschrijving van het object Gebruiker voor een verdere uitleg.

3.5 Selectieproces

In deze paragraaf wordt het belangrijkste proces van de gehele applicatie beschreven, namelijk het proces achter het daadwerkelijk selecteren van data. Wederom op globaal niveau; in het pilotontwikkelplan zal het tot in meer detail worden beschreven.

- 14 Lijst maken van unieke profile-tabellen aan de hand van de aan de selectie gekoppelde acties.
- 15 Alle profile-tabellen doorlopen (kan eventueel nog beperkt worden door een datumbereik op de laatste activiteit) en per deelnemer:
 - 1 Alle aan de selectie gekoppelde acties doorlopen en per actie:
 - 1 Kijken of de deelnemer voorkomt in de unsub-tabel, zo ja: negeren.
 - 2 Alle gewenste normale variabelen ophalen. Voor één variabele kunnen dit per deelnemer voor één actie meerdere waarden zijn, omdat een deelnemer meerdere keren aan een actie kan deelnemen (kan eventueel nog

beperkt worden door een datumbereik op het moment van deelnemen).

- 3 Alle gewenste gesponsorde variabelen ophalen. Ook hier kunnen er voor één variabele per deelnemer per actie meerdere waarden zijn (ook dit wordt beperkt door het datumbereik).
- 2 Alle gewenste profile-attributen opslaan.
- 16 Berekende variabelen berekenen.
- 17 De verzamelde gegevensverzameling moet nu op een dusdanige bewerkt worden dat van alle variabelen de laatst bekende waarde overblijft, rekening houdend met de ontdubbelings-variabele.
- 18 Deelnemers selecteren op basis van de voorwaarden.

Er moet rekening gehouden worden met dat selecties en variabelen recursief zijn.

3.6 Stijlgids

In deze paragraaf staat de stijlgids. Hierin zijn een aantal regels vastgelegd welke uiteindelijk zorgen voor een consistent uiterlijk. Dit heeft op zijn beurt ook weer een aantal positieve uitwerking zoals een kleinere kans op fouten die door de gebruiker gemaakt worden of snellere ontwikkeling.

Dit zijn de richtlijnen:

- Het systeem moet een *look and feel* krijgen die men associëert met E2Ma; er zullen dus delen of fragmenten van de huisstijl in verwerkt moeten worden.
- De verschillende schermen moeten een vergelijkbare indeling hebben. Het menu moet hierbij altijd aan de linkerkant van het scherm staan.
- Feedback moet op een eenduidige manier aan de gebruiker worden getoont.
- Bij het uitvoeren van langdurige taken van het systeem moet de voortgang nauwkeurig kunnen worden bijgehouden.

4 Overig

In dit hoofdstuk staat informatie overeenkomstig met de twee laatste activiteiten van de pilotplan-workshop, te weten: de technische structuur en de organisatorische inrichting. Omdat de hoeveelheid vast te leggen informatie hieruit een lage quantiteit heeft, is dit gegroepeerd in een apart hoofdstuk.

4.1 Technische structuur

In deze paragraaf staat de technische structuur. Concreet houdt dit in dat we beschrijven wat de minimale systeemeisen zijn voor de verschillende systemen die tijdens de ontwikkeling en na de fase Invoering nodig zijn.

De selectietool zal gebruik maken van een 3-tier architectuur. Dit houdt in dat verschillende delen afzonderlijk kunnen draaien en dat de delen met elkaar communiceren via van tevoren vastgestelde communicatieprotocollen.

Tier 1 is de database. Hierin wordt alle data t.b.v. de selectietool opgeslagen, waaronder dus alle gegevens omtrend selecties, resultaatsets, variabelen en dergelijke. Waar het systeem zijn data vandaan haalt wordt niet beschouwd als een onderdeel van de applicatie en valt hier dan ook niet onder.

Tier 2 is de applicatie-laag waarin alle berekeningen plaats vinden. Invoer vanaf tier 3 wordt verwerkt, geïnterpreteerd en omgezet in acties gebruik makend van tier 1 met vaak weer een terugkoppeling naar de gebruiker via tier 3.

En tot slot is zoals gezegd tier 3 de laag waarin gecommuniceerd wordt met de gebruiker. Deze kan hier acties uitvoeren, objecten manipuleren en dat soort dingen.

Ontwikkeling

In deze paragraaf beschrijven we de systeemeisen voor het systeem welke benodigd zijn tijdens de ontwikkeling van het systeem, uitgesplitst naar hardware en software. Alle drie tiers bevinden zich tijdens de ontwikkeling op één afzonderlijke machine, dus wordt hier verder geen onderscheid in gemaakt.

Als hardware is een normale PC voldoende. Wel dient de volgende software erop te kunnen draaien:

- Microsoft Windows;
- Microsoft Office;
- Microsoft Visio;
- PHP;
- MySQL;
- Photoshop;
- Vim;
- Mozilla Firefox.

Productie

In deze paragraaf beschrijven we de systeemeisen voor het systeem welke benodigd zijn in de tijd dat het systeem daadwerkelijk in productie is. Voor alle tiers geldt dat ze elkaar middels TCP/IP kunnen bereiken.

Database-tier

De database-tier (de eerste) heeft hoofdzakelijk de MySQL-databaseserver als software nodig, met hieraan gelieerde

hardwarematige kenmerken. Een hiervan is in ieder geval aanspraak gemaakt kan worden op voldoende geheugen.

Applicatie-tier

De applicatie-tier (de tweede) is het deel wat door middel van PHP de door de gebruiker aangevraagde acties daadwerkelijk uitvoert. Hoewel de minimale systeemeisen hiervoor erg laag liggen, is het voor deze toepassing wel aangeraden niet te bezuinigen op geheugengebruik.

GUI-tier

De GUI-tier (de derde) is het deel wat zorgdraagt voor de gebruikersinterface. Op deze machines moet op een vloeiende manier gebruik gemaakt worden van een webbrowser.

4.2

Organisatorische inrichting

Er is geen behoefte aan een handleiding voor de eindgebruiker, maar er zal wel een handleiding gemaakt moeten worden omtrend het operationele aspect.

5 Pilotplan

In dit hoofdstuk staat een geprioritiseerde lijst van pilots die sequentieel ontwikkeld gaan worden. Elke pilot heeft betrekking op een bepaald deel van het ontworpen systeemconcept. Voor elke pilot is een inschatting gemaakt van de benodigde tijd op basis van de te verwachte pilotdelen. Elke pilot - behalve de kernpilots - kan worden ingezet als zelfstandige volledig operationele subset van het uiteindelijk beoogde systeem.

Er zijn zes pilots onderscheiden (zie Tabel 11 voor een opsomming), en in elk van de komende paragrafen worden de afzonderlijke pilots beschreven. Er wordt aangegeven wat de inhoud van de pilot is (welke systeemeisen ermee gedekt worden), welke mogelijke pilotdelen er te onderscheiden zullen zijn en wat de prioriteit en de geschatte benodigde tijd voor elk pilotdeel is. De prioriteit is de gemiddelde prioriteit van alle pilotdelen. De belasting wordt als volgt uitgerekend: de som van 100% van alle basis-delen en 50% van alle comfort-delen.

#	Naam	Omschrijving	Uren	Prioriteit
1	GUI-standaard	Standaardisering van communicatie met de gebruiker en interne aangelegenheden (code-gewijs).	56	Kern
2	Variabelebeheer	Toevoegen, bewerken & verwijderen van variabelen.	25	Basis
3	Selectiebeheer	Toevoegen, bewerken & verwijderen van selecties.	57	Basis
4	Selectieactivatie	Het daadwerkelijk uitvoeren van een selectie: het aan de hand van een selectie, variabelen en	84	Basis

		data in de databases produceren van een resultaatset.		
5	Resultaatsset-beheer	Bekijken, downloaden & verwijderen van resultaatsets.	25	Basis
6	Gebruikersbeheer	Toevoegen, bewerken (m.n. de rechten) & verwijderen van gebruikers.	9,5	Comfort

Tabel 11 De pilots opgesomd.

5.1

GUI-standaard

Omschrijving

Dit is de enige kernpilot van het systeem. Het herbergt alle standaard-mogelijkheden in communicatie met de gebruiker, en code-conventies en indelingen.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 12.

#	Naam	Omschrijving	Uren	Prioriteit
1	Coderaamwerk	Een raamwerk waar later alle acties ingehangen kunnen worden.	8	Basis
2	Feedback	Standaardwijze voor het verzamelen en terugkoppelen van feedback uit acties.	4	Basis
3	Foutafhandeling	Standaardwijze voor het afhandelen van fouten die uit het systeem afkomstig zijn.	4	Basis
4	Overzicht-schermen	Een standaard overzichtsscherm met pagina-scheidingen, zoekfunctionaliteiten e.d.	16	Basis
5	Detailschermen	Standaard detailschermen t.b.v. toevoegen en bewerken. Inclusief achterliggende standaardcode voor conversie en validatie.	12	Basis
6	Subwindows	Functionaliteiten voor het werken met subwindows: een manier om objecten te koppelen.	6	Basis
7	Database-basis t.b.v. inloggen	De benodigde meta-data in de database t.b.v. het inloggen.	2	Basis
8	Inloggen zelf	Het inlogproces zelf: een login-formulier, afhandelende validatie-code en uitlogmogelijkheden.	4	Basis

Tabel 12 Pilotdelen GUI-standaard.

5.2

Variabelebeheer

Omschrijving

Deze pilot draagt zorg voor het beheer van variabelen. Variabelen kunnen worden: toegevoegd, bewerkt, verwijderd, getest, en mogelijk kan ook de code worden gevalideerd en kan deze in kleur worden weergegeven in het geval van een berekende variabele.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 13.

#	Naam	Omschrijving	Uren	Prioriteit
1	Database	De benodigde meta-data in de database t.b.v. de variabelen.	4	Basis
2	Overzicht	Het kunnen oproepen van een overzicht van alle op dat moment bij het systeem bekende variabelen.	4	Basis
3	Toevoegen	Het kunnen toevoegen van een variabele.	12	Basis
4	Bewerken	Het kunnen bewerken van een variabele.	2	Basis
5	Verwijderen	Het kunnen verwijderen van een variabele.	2	Basis
6	Testen	Het kunnen testen van een berekende variabelen door alle bronvariabelen met de hand in te vullen.	2	Comfort
7	Code-validatie	Automatische validatie van de code tijdens het toevoeg- of bewerk-proces.	4	Luxe
8	Code-kleurcoderingen	Kleurenweergave van de code achter een berekende variabele.	2	Luxe

Tabel 13 Pilotdelen Variabelebeheer.

5.3

Selectiebeheer

Omschrijving

Deze pilot draagt zorg voor het beheer van selecties. Een selectie is het belangrijkste deel van de gehele applicatie omdat het de basis vormt achter het selecteren van data. Selecties kunnen worden toegevoegd, bewerkt en verwijderd. Het activeren van selecties is vanwege de complexiteit gepromoveerd tot een aparte pilot.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 14.

#	Naam	Omschrijving	Uren	Prioriteit
1	Database	De benodigde meta-data in de database t.b.v. de selecties.	8	Basis
2	Overzicht	Het kunnen oproepen van een overzicht van alle op dat moment bij het systeem bekende selecties	8	Basis
3	Toevoegen	Het kunnen toevoegen van een selectie.	28	Basis
4	Bewerken	Het kunnen bewerken van een selectie.	8	Basis
5	Verwijderen	Het kunnen verwijderen van een selectie.	3	Basis
6	Doorgeven	Het kunnen doorgeven van een selectie zodat deze door een andere gebruiker is te gebruiken.	4	Comfort

7	Inplannen	De mogelijkheid om een selectie in te plannen zodat de activatie op een van te voren aan te geven tijdstip automatisch plaats zal vinden. De wenselijkheid van dit deel hangt voor een deel nog af van de tijd dat de activatie van een selectie in beslag zal nemen.	4	Luxe
---	-----------	---	---	------

Tabel 14 Pilotdelen Selectiebeheer.

5.4

Selectieactivatie**Omschrijving**

Deze pilot draagt zorg voor het activeren van selecties. Het is een dermate complex proces, dat het een eigen pilot verdient. De uitvoer van een activatie is een resultaatset, en daarom zal de database voor resultaatsets in deze pilot al worden aangemaakt. Daarnaast is er plek voor een voortgangsindicator, het selectieproces zelf, het achterhalen van berekende variabelen, het groeperen en/of tellen van de resultaten en de opslag in de database.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 15.

#	Naam	Omschrijving	Uren	Prioriteit
1	Database	De benodigde meta-data in de database t.b.v. opslag in resultaatsets.	4	Basis
2	Voordoorloop	Het voor-doorlopen van de selectie voor bepaling van de diepte en de op te halen variabelen.	12	Basis
3	Selectieproces zelf	Het uitvoeren van de selectie zelf.	32	Basis
4	Achterhalen berekende variabelen	In samenwerking met het vorige deel, worden in dit deel de berekende variabelen berekend.	12	Basis
5	Groeperen en/of tellen	Nabewerking van hetgeen uit een selectie komt door te groeperen en/of te tellen.	12	Basis
6	Ontdubbelen	Het optioneel unificeren van de resultaten op een bepaalde variabele.	4	Basis
7	Opslag in de database	Het proces achter het plaatsen van de (nabewerkte) data in de database.	4	Basis
8	Voortgangs-indicator	Een gebruikersvriendelijke indicator van de voortgang.	8	Comfort

Tabel 15 Pilotdelen Selectieactivatie.

5.5 Resultaatsetbeheer

Omschrijving

Deze pilot draagt zorg voor het beheren van resultaatsets. Zoals gezegd volgt er een resultaatset uit elke activatie van een selectie, en met de functionaliteiten die in deze pilot worden gedefiniëerd, kunnen de resultaatsets daadwerkelijk worden bekeken, gedownload of verwijderd.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 16.

#	Naam	Omschrijving	Uren	Prioriteit
1	Overzicht	Het bekijken van alle resultaatsets.	5	Basis
2	Bekijken	Het bekijken van een resultaatset in de GUI van de selectietool zelf.	8	Basis
3	Download	Het downloaden van een resultaatsets als TSV- of CSV-bestand.	6	Basis
4	Verwijderen	Het verwijderen van een resultaatset	2	Basis
5	Doorgeven	Het kunnen doorgeven van een resultaatset zodat deze door een andere gebruiker te bekijken is.	4	Comfort

Tabel 16 Pilotdelen Resultaatsetbeheer.

5.6 Gebruikersbeheer

Omschrijving

Deze pilot draagt zorg voor het beheren van gebruikers. Om te bepalen wat iemand met het systeem mag doen, wordt er eerst gekeken naar de gebruikersgroepen en wat de standaardrechten zijn van alle leden van deze groepen. Deze kunnen vervolgens op meerdere lagere niveau's worden overschreven. De functionaliteiten rond het toevoegen, bewerken en verwijderen van gebruikers zijn in deze pilot samengevat. Gebruikersgroepen kunnen alleen direct in de database beheerd worden.

Het is een Confort-pilot, maar dat betekent niet dat zonder dit er geen rechten te beheren zijn. Het is in grote lijnen namelijk alleen een GUI. Beheer van de gebruikers zelf zou ook enkel via een database-administratieprogramma kunnen gebeuren.

Hoewel er in §2.4 een verdeling is gemaakt in vier vaste gebruikersgroepen, zal het systeem zelf gebouwd worden met een flexibelere gedachte. Dat betekent dat er zonder al te veel moeite gebruikersgroepen kunnen worden toegevoegd of verwijderd.

Pilotdelen

In deze paragraaf een eerste voorzichtige verdeling van de pilot in pilotdelen en een schatting van de arbeidsbelasting. Zie Tabel 17.

#	Naam	Omschrijving	Uren	Prioriteit
1	Database	De benodigde meta-data in de database t.b.v. de gebruikers.	4	Basis
2	Overzicht	Het kunnen oproepen van een overzicht van alle op dat moment bij het systeem bekende gebruikers.	3	Comfort
3	Toevoegen	Het kunnen toevoegen van een gebruiker en het hem doen toekomen van privileges.	6	Comfort
4	Bewerken	Het kunnen bewerken van een gebruiker en/of zijn privileges.	4	Comfort
5	Verwijderen	Het kunnen verwijderen van een gebruiker.	2	Comfort

Tabel 17 Pilotdelen Gebruikersbeheer.

8 Plan van Aanpak pilot #1

1 Inleiding

Dit is het Plan van Aanpak van pilot #1. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de fase Definitiestudie en is deels gebaseerd op het pilotplan. Het hoort bij de eerste iteratie van de fase Pilotontwikkeling voor de pilot GUI-standaard. Deze pilot moet uitkomen op een standaardisering van communicatie met de gebruiker en interne aangelegenheden (code-gewijs) op het vlak van de basis van de selectietool.

In hoofdstuk 2 worden de in deze iteratie van de fase Pilotontwikkeling uit te voeren activiteiten uiteengezet, tevens op basis van de voorlopige verdeling in pilotdelen die gemaakt is in de definitiestudie. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat dit zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
17-03-2005	1.0	Initiële versie.

Tabel 18 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze iteratie van de fase Pilotontwikkeling te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaan komt. Als basis worden de activiteiten die zijn opgesomd in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE

Tabel 19 Legenda

- 19 Plan van Aanpak opstellen
- 20 Globaal-functionele structuur pilot specificeren
- 21 Globaal-technische structuur pilot specificeren
- 22 Globaal-organisatorische inrichting specificeren
- 23 GUI-ontwerp
 - 1 Schermschetsen
 - 2 Navigatieschema
- 24 Testplan opstellen
- 25 Pilotontwikkelplan opstellen

26 Ontwerpen & bouwen van softwarebouweenheden, samenstellen handleidingen en opstellen invoeringsprocedures voor de volgende pilotdelen:

- 1 Coderaamwerk
- 2 Feedback
- 3 Foutafhandeling
- 4 Overzichtschermpjes
- 5 Detailschermpjes
- 6 Subwindows
- 7 Database-basis t.b.v. inloggen
- 8 Inloggen zelf

27 Beoordelen en testen pilot

Aanpassingen aan globaal Plan van Aanpak

De activiteit 'Pilotplan bijwerken' wordt overgeslagen omdat dit de eerste iteratie van deze fase is, en er nog geen feedback naar het pilotplan geformuleerd kan worden.

De activiteit 'SUMI-test opstellen' wordt overgeslagen omdat er gezien de aard van de pilot - veel basiszaken die herbruikt zullen gaan worden - weinig onderdelen bij zitten die het testen op gebruikersvriendelijkheid waard zijn.

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 6 en eindigt halverwege week 8.

Activiteit	Weken														
	6					7					8				
	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V
Plan van Aanpak opstellen															
Functionele structuur															
Technische structuur															
Organisatorische structuur															
GUI-ontwerp															
Schermshots															
Navigatieschema															
Testplan opstellen															
Pilotontwikkelplan opstellen															
Ontwerpen, bouwen, samen- en opstellen															
Coderaamwerk															
Feedback															
Foutafhandeling															
Overzichtschermpjes															
Detailschermpjes															
Subwindows															
Database-basis t.b.v. inloggen															
Inloggen zelf															
Beoordelen en testen															

Tabel 20 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 1: Plan van Aanpak.
- 7: Pilotontwikkelpunten met de uitkomsten van de punten 2 t/m 6.
- Per softwarebouweenheid: de resultaten van het ontwerpen & bouwen, het samenstellen van de handleidingen en het opstellen van de invoeringsprocedures.
- De pilot zelf (programmacode, databasestructuren, enzovoorts).
- Handleiding.

9 Pilotontwikkelplan pilot #1

1 Inleiding

Dit is het Pilotontwikkelplan van pilot #1. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Plan van Aanpak behorende bij deze pilot. De definitiestudie is gebruikt als basis, en is waar nodig in het gebied van deze pilot nog aangevuld en uitgebreid. Deze pilot moet uitkomen op een GUI-standaard. Het is een raamwerk waarbinnen alle schermen van de overige pilots makkelijk in gemaakt kunnen worden wat automatisch zal leiden tot een hoge mate van consistentie, zowel aan de buitenkant als aan de binnenkant (code-gewijs).

Hoofdstuk 2 bevat de functionele structuur. Dit omvat een detaillering van zaken uit het systeemconcept in de definitiestudie. Concreet voor deze pilot betekend dit een beschrijving van de basisfunctionaliteiten van de GUI. Hoofdstuk 3 bevat de technische structuur. De relevante objecten van het klassendiagram worden omgezet naar een relationeel implementatiemodel, en er wordt diep ingegaan op het rechtensysteem. Hoofdstuk 4 bevat de organisatorische inrichting: de eisen die gesteld worden aan de productiehandleiding voor dit deel van het systeem. In hoofdstuk 5 wordt een GUI-ontwerp gegeven in de vorm van schermetschetsen en een navigatieschema, en tot slot komt in hoofdstuk 6 het testplan waarin een aantal testscenario's zijn vastgelegd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
17-03-2005	1.0	Initiële versie.

Tabel 21 Veranderingengeschiedenis.

2 Functionele structuur

In dit hoofdstuk staat de functionele structuur beschreven. Allereerst komen de basisfunctionaliteiten aan bod. Deze functionaliteiten zijn geen detaillering van iets wat in het systeemconcept staat, maar is meer een afgeleide omdat de pilot zelf ook meer een raamwerk is. Er is gekeken naar de mogelijke grafische gebruikersinterfaces die

voortkomen uit het systeemconcept van de definitiestudie, en deze zijn vervolgens gegeneraliseerd tot een paar basisdelen.

2.1 Coderaamwerk

Er is al eerder gekozen voor een op het web gebaseerde client-server-architectuur met PHP als scripttaal aan de serverkant en JavaScript als scripttaal aan de clientkant. Omdat er voor de verschillende in de andere pilots te ontwikkelen schermen veel overeenkomstige zaken te vinden zijn, moeten er van te voren ook een aantal scripts worden geschreven of herbruikt om te komen tot een raamwerk waarin bijvoorbeeld objectspecifieke code gemakkelijk ingezet kan worden.

De functionaliteiten zijn als volgt. Manieren tot:

- becommentariseren van code;
- indienen en afvangen van informatieaanvragen;
- verbinden met hoofddatabase;
- starten van sessies;
- doorgeven van informatie tussen de verschillende delen;
- voorbereiden van informatie voor gebruik in SQL-queries;
- snel nagaan van of iemand het recht heeft tot het doen van een actie op een bepaald object.

Er moet tevens plaats zijn voor het specificeren van een mode waar het systeem zich op dat moment in bevindt. Er zijn twee modi, namelijk productiemodus of ontwikkelmodus. In productiemodus is alles als normaal, maar de ontwikkelmodus verandert een aantal zaken aan het rapporteren van foutmeldingen bijvoorbeeld. Tevens zal het voor iedereen die niet het recht 'gebruik tijdens ontwikkelmodus' hebben, onmogelijk zijn het systeem te gebruiken, ook als deze reeds zijn ingelogd op het moment dat de modus verandert.

2.2 Feedback

Als een gebruiker een actie doet, bijvoorbeeld het aanmaken van een nieuw object, moeten hierbij meestal een aantal gegevens bij ingevoerd worden. Nadat het systeem de actie heeft ingewilligd, wil de gebruiker vervolgens weten hoe het proces is verlopen of wat er eventueel fout is.

Er wordt onderscheid gemaakt tussen vier verschillende aarden feedback:

- vragend (voor bevestigingen);
- informatief (voor meldingen over dat iets succesvol is verlopen);
- waarschuwend (voor meldingen dat bepaalde attributen niet aan de invoereisen voldoen);
- kritieke fout (voor meldingen over dat iets onverwachts onsuccesvol is verlopen).

Voor alle aarden bestaat de mogelijkheid tot het geven van een hiërarchische lijst van redenen of extra informatie.

Vragende feedback of feedback met een kritieke fout worden altijd getoond in een JavaScript `alert()` of `confirm()`. Informatieve of waarschuwende feedback wordt altijd *in-line* getoond waardoor de gebruiker, zonder deze informatie eerst van het scherm te moeten halen, direct kan beginnen met het verwerken van de feedback.

2.3 Foutafhandeling

Indien er zich systeemfouten voordoen in het systeem, moeten deze zich - afhankelijk van de modus van het systeem - via verschillende kanalen kunnen verspreiden. Indien het systeem in ontwikkelmodus, worden alle foutmeldingen 1 op 1 doorgegeven aan de gebruiker. Indien het systeem in productiemode staat, worden alle foutmeldingen geë-maild naar een nader te specificeren e-mailadres om gevoelige informatie niet bloot te stellen aan het publiek.

2.4 Overzichtschermen

Informatie wordt veelal weergegeven in overzichtschermen of detailschermen. Overzichtschermen geven een overzicht van (een selectie van) objecten van een bepaald type. Bijvoorbeeld een overzicht van alle bij het systeem bekende variabelen.

Elk overzicht heeft een aantal acties (niet te verwarren met de enquêtes van E2Ma zelf). Er zijn algemene, enkelvoudige en meervoudige acties. Algemene acties hebben geen betrekking op een object. Enkelvoudige acties hebben altijd betrekking op één object. Meervoudige acties echter hebben altijd betrekking op één óf meer objecten tegelijk. Normaal gesproken zijn er de volgende acties, maar dit is voor een deel objectspecifiek:

- Algemene actie: toevoegen.
- Enkelvoudige actie: bewerken.
- Meervoudige actie: verwijderen.

Daarnaast heeft elk overzicht de mogelijkheden om een selectie van objecten te maken. Een selectie kan ook gemanipuleerd worden middels drie selectieopties: alles selecteren, selectie inverteren of niets selecteren.

2.5 Detailschermen

Om alle details van een object inzichtelijk te maken, wordt er gebruik gemaakt van detailschermen. Deze worden altijd gebruikt voor het toevoegen of bewerken van een object, maar ook voor het alleen weergeven van de details. In zo'n geval wordt de mogelijkheid tot het aanmaken of wijzigen van een object uitgezet door de desbetreffende opslag- of wijzigknoppen onklikbaar te maken.

2.6 Subwindows

Een subwindow is een popup met als functie een deel van een taak die de gebruiker aan het uitvoeren is, uit te besteden aan een afzonderlijk stukje grafische interface. Omdat het afzonderlijk is, blijft de interface waarvandaan het subwindow en hiermee de context van de taak in het subwindow duidelijk voor de gebruiker. Een subwindow kan zelf ook weer een subwindow openen, om zo te komen tot een waterval aan schermen waarin elk scherm toegespitst is op een enkele taak.

Een subscherm kan een terugkoppeling hebben naar zijn 'ouder', bijvoorbeeld bij een selecteren van een object om te koppelen aan een andere.

Er zijn een aantal verschillende soorten subschermen:

- De feedbacksubwindows. Deze zijn vergelijkbaar met de schermen zoals deze door een besturingsysteem worden getoond.
- Zonder terugkoppeling.
- Met terugkoppeling.

Een subwindow met terugkoppeling, kan dit vervolgens ook in twee smaken hebben: enkelvoudig of meervoudig. Bij enkelvoudige feedback weet het subwindow dat na selectie van één element het subwindow weer gesloten mag worden, maar bij meervoudige feedback blijft het subwindow nog open voor verdere selecties.

2.7 Inloggen

Ook het inloggen en hierbij gelijk ook het basisdeel van het rechtensysteem, is inbegrepen bij deze pilot. Inloggen gebeurt door middel van het aan het systeem geven van een gebruikersnaam en wachtwoord. In een opeenvolgende stap worden deze gegevens gecontroleerd op juistheid. Indien er een bijpassende gebruiker is gevonden, moet gekeken worden of deze voldoende rechten heeft in te loggen. Als ook dit goed is, is de inlogprocedure voltooid. Op het moment dat er ergens fouten optreden, wordt dit door middel van waarschuwende feedback doorgegeven.

3 Technische structuur

In dit hoofdstuk staat de technische structuur van deze pilot. De relevante objecten van het klassendiagram worden omgezet in een relationeel representatiemodel en in een relationeel implementatiemodel, en er wordt diep ingegaan op het rechtensysteem.

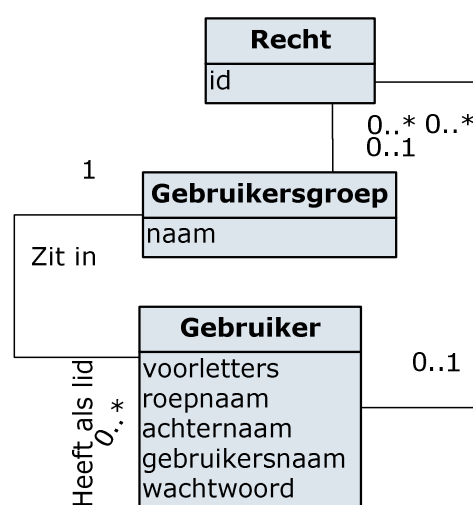
3.1 Databasestructuur

In deze paragraaf worden de voor deze pilot relevante klassen omgewerkt tot een relationeel representatiemodel en een relationeel implementatiemodel.

Met het klassendiagram (of een deel ervan) hebben we de objecten, attributen en relaties in kaart gebracht. In het proces tot het implementeren van de database, kan niet alles hieruit 1-op-1 worden overgenomen. De vertaalslag tussen het klassendiagram en het relationeel implementatiemodel vindt plaats in de tussenliggende activiteit van het samenstellen van een relationeel representatiemodel. Hierbij worden ook alle normalisaties behandeld. Vanuit het relationeel representatiemodel ligt de weg vrij tot het maken van een relationeel implementatiemodel.

Klassendiagram

Zie Figuur 11 voor het klassendiagram specifiek aan de gebruikers.



Figuur 11 Het klassendiagram.

Dit is inclusief rechten voor gebruikers en gebruikersgroepen, maar exclusief rechten specifiek voor selecties en resultaatsets. Zie §3.2 voor een uitgebreide uitleg over de werking van het rechtensysteem.

Relationeel representatiemodel

Zoals in de inleiding al vermeld, is dit relationeel representatiemodel een tussenstap tussen het klassendiagram en het implementatiemodel. In deze paragraaf wordt elk object apart behandeld met vastlegging van het verwerken van de correlaties en het normalisatieproces.

Normaliseren is een stapsgewijs proces wat op relaties wordt toegepast om uiteindelijk te komen tot een gegevensmodel zonder overtolligheden en tegenspraken. Er wordt getracht te komen tot volledige normalisatie (2NF, BCNF & 4NF). Dit betekent:

- dat er geen meerwaardige attributen meer zijn;
- ieder attribuut dat geen deel uitmaakt van een kandidaatsleutel, volledig functioneel afhankelijk is van elke kandidaatsleutel;
- er geen transitieve afhankelijkheden meer zijn;

- elke determinant een kandidaatsleutel is van de relatie;
- er geen meerwaardige afhankelijkheden meer zijn.

Het zal vaak voorkomen dat er, ondanks dat er al één of meer kandidaatsleutels bestaan, toch nog een extra numerieke primaire sleutel wordt toegevoegd, en wel om de volgende redenen:

- Opslageefficiëntie. Een relationele database is *value*-based en dat betekent dat de primaire sleutel waar de vreemde sleutel naar verwijst letterlijk wordt overgenomen in de verwijzende tabel. Het kost extra schijfruimte voor dit attribuut, maar deze wordt dubbel en dwars teruggewonnen bij de vreemde sleutels.
- Geen verwijzingsproblematiek door vreemde tekens. Vanuit de webapplicatie zullen primaire sleutels (ook in de gedaante van vreemde sleutels) veelvuldig heen en weer worden gezonden tussen de verschillende tiers. Omdat de kanalen waar deze gegevens door reizen gevoelig zijn voor vreemde tekens, kan dit problemen opleveren als er zich daadwerkelijk vreemde tekens bevinden in de normale kandidaatsleutels.
- Veranderingen van kandidaatsleutels die ook bestaan als vreemde sleutels. Omdat ondersteuning voor het controleren en bijhouden van referentiële integriteit op het productieplatform nog niet worden ondersteund, is het veel veiliger te kiezen voor een primaire sleutel die nooit hóeft te veranderen.

Recht

Initieel bestaat de volgende relatie:

```
recht(id)
```

Een recht hoort bij 0 of 1 gebruikersgroepen of bij 0 of 1 gebruikers. Hierdoor krijgen we het volgende:

```
recht(id, gebruikersgroepId, gebruikerId)
```

Omdat recht nu een zwak entiteitstype is (het is bestaansafhankelijk van óf gebruikersgroep óf gebruiker), nemen we deze twee vreemde sleutels op als extra attribuuttypen voor de primaire sleutel:

```
recht(id, gebruikersgroepId, gebruikerId)
```

- gebruikersgroepId: vreemde sleutel, verwijst naar id in gebruikersgroep, null toegelaten.
- gebruikerId: vreemde sleutel, verwijst naar id in gebruiker, null toegelaten.

Gebruikersgroep

Initieel bestaat de volgende relatie:

```
gebruikersgroep(naam)
```

Een gebruikersgroep heeft 0 of meer rechten en heeft 0 of meer gebruikers als leden. Er bestaan dus geen vreemde sleutels. Er zal wel een extra numeriek attribuut als primaire sleutel worden toegevoegd:

```
gebruikersgroep(id, naam)
```

Gebruiker

Initieel bestaat de volgende relatie:

```
gebruiker(voorletters, roepnaam, achternaam, gebruikersnaam,  
          wachtwoord)
```

Een gebruiker heeft 0 of meer rechten en zit in één gebruikergroep. Er komt dus één vreemde sleutel en tevens een extra numeriek attribuut als primaire sleutel worden toegevoegd:

```
gebruiker(id, voorletters, roepnaam, achternaam,  
          gebruikersnaam, wachtwoord, gebruikersgroepId)  
▪ gebruikersgroepId: vreemde sleutel, verwijst naar id in  
  gebruikergroep, null niet toegelaten.
```

Relationeel implementatiemodel

Nu het relationeel representatiemodel er is, kan het relationeel implementatiemodel worden gemaakt. Dit gebeurt middels SQL als DDL.

Recht

```
CREATE TABLE recht  
(  
    id MEDIUMINT UNSIGNED NOT NULL,  
    gebruikersgroepId TINYINT UNSIGNED,  
    gebruikerId SMALLINT UNSIGNED,  
    PRIMARY KEY (id, gebruikersgroepId, gebruikerId),  
    FOREIGN KEY (gebruikersgroepId)  
        REFERENCES gebruikersgroep ON DELETE CASCADE,  
    FOREIGN KEY (gebruikerId)  
        REFERENCES gebruiker ON DELETE CASCADE  
);
```

Omdat is gebleken dat attributen die deelnemen aan een samengestelde primaire sleutel geen null-waarden kunnen bevatten, kan in bovenstaande code ook net zo goed twee keer NOT NULL toegevoegd worden.

Gebruikersgroep

```
CREATE TABLE gebruikersgroep  
(  
    id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    naam TINYTEXT NOT NULL,  
    PRIMARY KEY (id)  
);
```

Gebruiker

```
CREATE TABLE gebruiker  
(
```

```

id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
voorletters TINYTEXT NOT NULL,
roepnaam TINYTEXT NOT NULL,
achternaam TINYTEXT NOT NULL,
gebruikersnaam TINYTEXT NOT NULL,
wachtwoord TINYTEXT NOT NULL,
gebruikersgroepId TINYINT UNSIGNED NOT NULL,
PRIMARY KEY (id),
UNIQUE (gebruikersnaam(255)),
FOREIGN KEY (gebruikersgroepId)
REFERENCES gebruikersgroep ON DELETE RESTRICT
);

```

3.2 Rechtensysteem

In deze paragraaf wordt het rechtensysteem uitgelegd. Er wordt ingegaan op de doelstellingen en de wijze waarop het geïmplementeerd zal worden.

We hebben te maken met:

- gebruikers;
- gebruikersgroepen;
- objecten;
- rechten;

Elke combinatie van een van deze vier zaken moet met het rechtensysteem kunnen worden vastgelegd en afgedwongen. Elk recht heeft een numerieke waarde, overeenkomend met het `id`-attribuut van de klasse `Recht`. Door op bit-basis de rechten van de gebruikersgroepen en de gebruikers op te tellen, kunnen we voor een specifiek recht voor een specifiek persoon achterhalen of iets een toegestane actie is of niet.

De rechten zijn vermeld in Tabel 22. Grijs rijen zijn rechten die van toepassing zijn bij pilotdelen die de Comfort of Luxe-prioriteit hebben.

Bit	Numerieke waarde	Recht	Impliceert
1	1	Gebruik van het systeem.	
2	2	Gebruik van het systeem in ontwikkelmode.	1
3	4	Overzicht van alle selecties oproepen	
4	8	Selecties toevoegen	6
5	16	Selectie bewerken	
6	32	Selectie verwijderen	4
7	64	Selectie activeren	
8	128	Selectie inplannen	7
9	256	Selectie doorgeven	
10	512	Ook rauwe selecties zonder nabewerking	
16	32768	Overzicht van alle resultaatsets oproepen	
17	65536	Resultaatset bekijken	
18	131072	Resultaatset downloaden	17
19	262144	Resultaatset verwijderen	
20	524288	Resultaatset doorgeven	

Tabel 22 De rechten.

Er zijn twee gebruikersgroepen, te weten 'Data' en 'Sales', met de elk twee personen met volgende rechten:

- Data: 1, 7, 16 en 17:
 - Piet: 3, 4, 5, 6, 9, 10
 - Klaas: 3, 16, 18
- Sales: 1:
 - Gerard: 3, 7
 - Adam: geen

Piet kan nu een rauwe selectie aanmaken, doorgeven aan Adam, hierbij de rechten 4, 5, 6, 7, 9 en 17 vergeven. Rechten 3 en 10 niet omdat die niet objectspecifiek zijn. Rechten 8 en 18 niet omdat Piet hier zelf niet over beschikt. Merk op dat er dus ook rechten niet specifiek aan het selectie-object worden gebonden. Dit heeft als doel dat deze rechten gaan gelden voor elk resultaatset-object welke voort kunnen komen uit het selectie-object. Als Adam deze selectie activeert, krijgt hij voor de resulterende resultaatset recht 17 toegewezen. De resultaatset is daarnaast ook nog door Klaas te zien, omdat hij recht 16 heeft.

De bepaling van of een recht gegeven is in een bepaalde context, gebeurd door alle aanwezige rechten op te tellen, en deze binair te vergelijken met wat er benodigd is. Stel dat Gerard een selectie wil toevoegen, hebben we 8 als eis. Hij heeft 69 ($1 + 4 + 64$), dus dan krijgen we deze bitgewijze operatie: $8 \ \& \ 69$. Dit levert `false` op waarmee het systeem weet dat Gerard geen selecties mag toevoegen.

4 Organisatorische inrichting

In dit hoofdstuk wordt de organisatorische inrichting behandeld. Het doel van bijbehorende activiteit is het specificeren van de organisatie-aspecten rond de pilot in de ontwikkeling, voor zover van belang voor de gebruikers. Voor dit project en deze pilot betekend dit concreet welke eisen er gesteld worden aan de productiehandleiding.

De productiehandleiding is een handleiding welke gebruikt kan worden voor het beheer van het systeem. Hierbij moet gedacht worden aan invoering, verhuizing, onderhoud en dat soort zaken.

De inhoud moet minimaal ingaan op de volgende punten:

- Invoeringsprocedure: welke handelingen moeten verricht worden om het systeem te installeren of te verhuizen naar een andere machine.
- Onderhoudsprocedure: welke handelingen moeten verricht worden om het systeem te onderhouden. Hierbij moet gedacht worden aan bijvoorbeeld het schoon houden van de database.

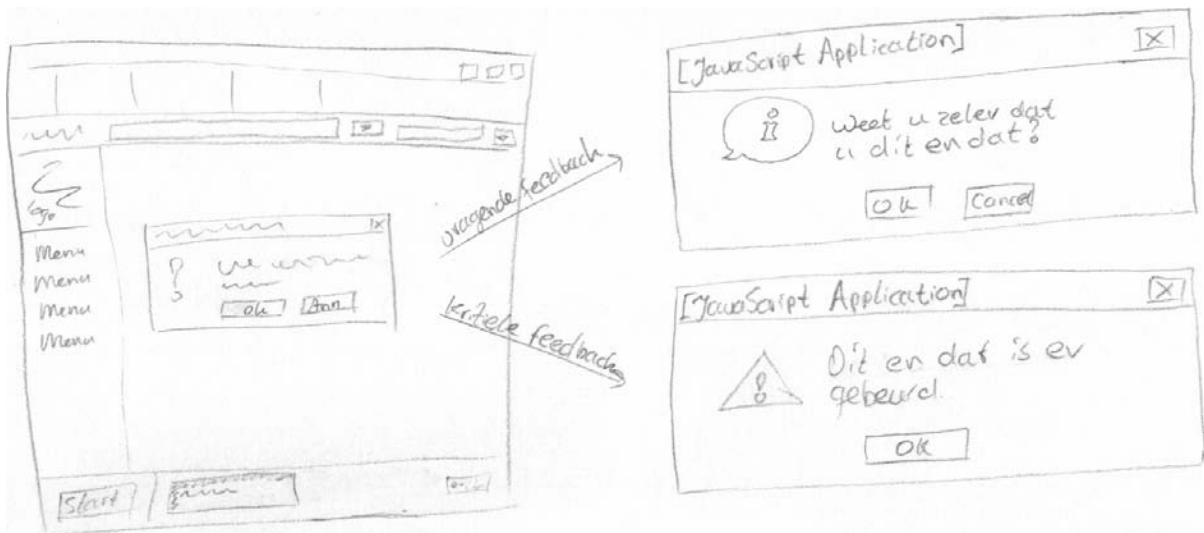
5 GUI-ontwerp

In dit hoofdstuk komt het ontwerp van de GUI aan bod. Omdat dit een kernpilot is en de mogelijke soorten schermen beschrijft, zijn deze schetsen belangrijk omdat ze voor een deel de look en feel van het hele systeem zullen bepalen.

Allereerst komen er een aantal schermschetsen voorbij, vervolgens gevolgd door een veralgemeniseerd navigatieschema.

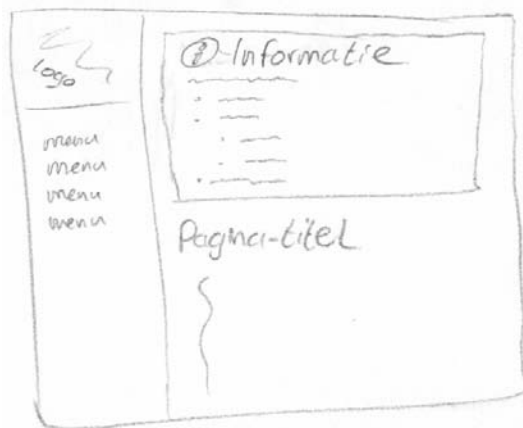
5.1 Schermschetsen

In deze paragraaf staan een aantal schermschetsen. Allereerst drie schetsen van hoe de verschillende manieren van feedback eruit zullen komen te zien. Zie Figuur 12 voor de vragende feedback en voor de feedback met kritieke informatie.

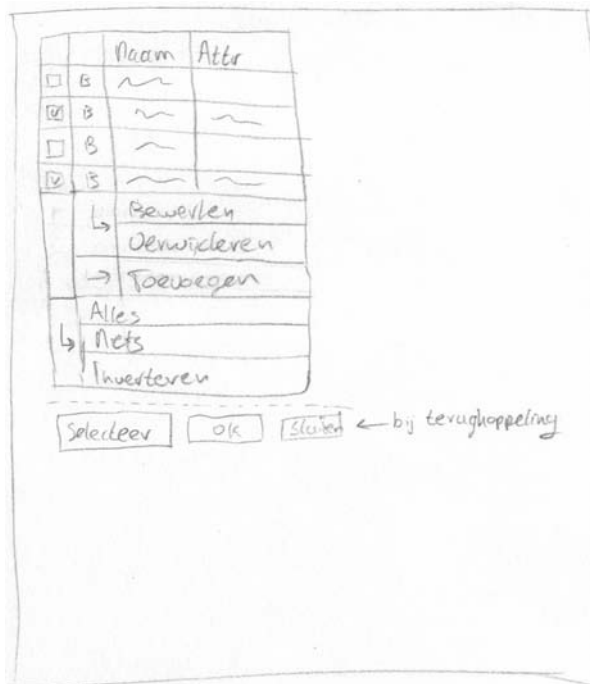
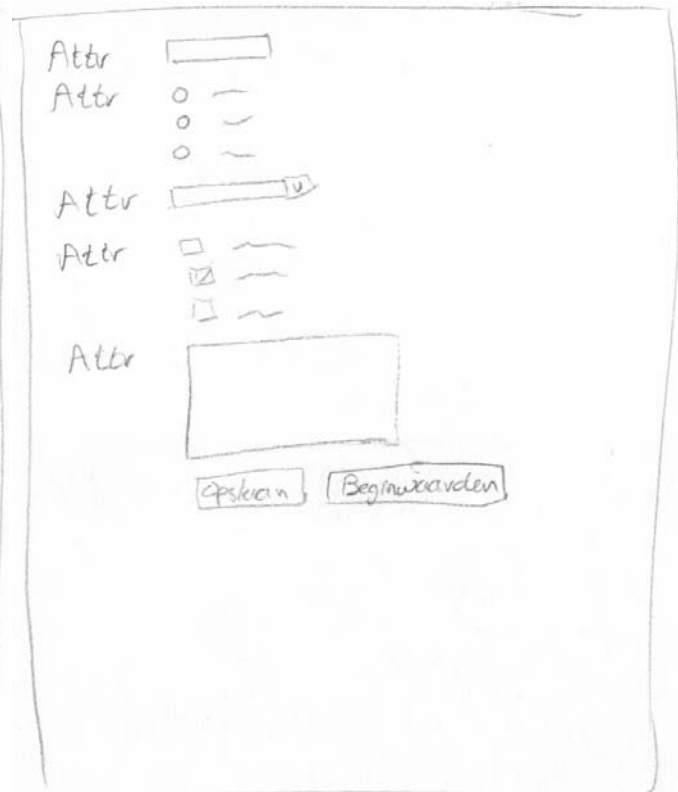


Figuur 12 JavaScript-feedback.

Zoals in §2.2 al vermeld is, komt informatieve en waarschuwende feedback *in-line* op het scherm, dus midden in de informatie van die pagina zelf. Zie Figuur 13 voor hoe de informatieve feedback in het scherm verwerkt gaat worden. Datzelfde, maar dan voor waarschuwende feedback, is geschetst in Figuur 14.

**Figuur 13** Informatieve feedback.**Figuur 14** Waarschuwend feedback.

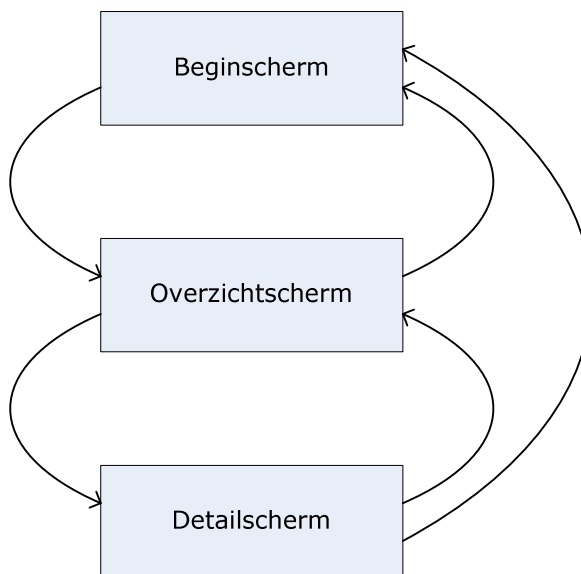
Voor wat betreft de schermen zelf, zijn er twee soorten: de overzichtschermen en detailschermen (zie §2.4 en §2.5). Zie Figuur 16 en Figuur 17. Het menu komt altijd nog aan de linkerkant van deze schermen; tenzij het scherm wordt geopend in een subwindow want dan is het menu niet zichtbaar.

**Figuur 15** Overzichtscherm.**Figuur 16** Detailscherm.

1.1 Navigatieschema

Hieronder staat het navigatieschema voor de schermen die in deze pilot zijn omschreven, zie Figuur 17. In de praktijk zijn er meerdere verschillende overzichtschermen en meerdere verschillende

detailschermen, dus elk scherm behalve het beginscherm in dit schema kan meerdere keren voorkomen.



Figuur 17 Navigatieschema.

6 Testplan

In dit hoofdstuk komt het testen aan bod. Er worden een aantal testscenario's samengesteld en uitgelegd. Tijdens de activiteit Beoordelen en testen moeten deze scenario's met goed gevolg gevolgd worden.

De scenario's zijn als volgt:

- Voeg een nieuw testobject toe, vul geen enkel gegeven in en verzend het formulier. Het systeem dient nu aan te geven dat er invalide invoer is hoe dit op te lossen.
- Zorg ervoor dat het systeem een fout genereerd in productiemode. Er dient nu een foutrapport verstuurd te worden naar het van te voren opgegeven e-mailadres.
- Log in, ga naar het overzicht van testobjecten, log uit en probeer wederom te gaan naar het overzicht van testobjecten. Dit dient niet te kunnen; in plaats hiervan moet de gebruiker de melding krijgen dat hij zijn inloggegevens moet ingeven.

Maak een nieuw testobject aan en koppel hieraan een ander test-object door gebruik te maken van een subwindow. Het subwindow dient te verschijnen in modale vorm (zonder dat het achterliggende hoofdwindow op een een of andere manier nog focus kan krijgen en behouden), de inhoud van het hoofdwindow dient afgeschermd te zijn middels een 'gordijn', en het gekozen test-object moet goed terecht komen in het testobject dat wordt aangemaakt.

10 Productiehandleiding

1 Inleiding

Dit is de productiehandleiding. Het is geschreven tijdens de fase Pilotontwikkeling van pilot #1, maar zal uiteindelijk voor het gehele systeem gaan gelden, omdat er vanuit de overige pilots geen toevoegingen zijn. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

De productiehandleiding is een handleiding welke gebruikt kan worden voor het beheer van het systeem. Hierbij moet gedacht worden aan invoering, verhuizing, onderhoud en dat soort zaken. In hoofdstuk 2 wordt ingegaan op de invoeringsprocedure: de werkzaamheden die uitgevoerd moeten worden om het systeem te installeren of te verhuizen naar een andere machine. In hoofdstuk 3 tot slot wordt ingegaan op de onderhoudsprocedure: welke handelingen moeten verricht worden om het systeem te onderhouden. Hierbij moet gedacht worden aan bijvoorbeeld het schoon houden van de database.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
06-04-2005	1.0	Initiële versie.
12-04-2005	1.1	Uitbreiding met uitleg over optionele attributen van de tabellen dbms en actie.

Tabel 23 Veranderingengeschiedenis.

2 Invoeringsprocedure

In dit hoofdstuk staat beschreven welke eisen er zijn verbonden aan en welke handelingen er moeten worden uitgevoerd bij het installeren of verhuizen van het systeem.

2.1 Systeemeisen

De selectietool maakt gebruik van een 3-tier architectuur en de systeemeisen zijn over deze drie tiers te verdelen. Dat het systeem zo is verdeeld houdt in dat verschillende delen afzonderlijk kunnen draaien en dat de delen met elkaar communiceren via van tevoren vastgestelde communicatieprotocollen.

Tier 1 is de database. Hierin wordt alle data t.b.v. de selectietool opgeslagen, waaronder dus alle gegevens omtrend selecties, resultaatsets, variabelen en dergelijke. Waar het systeem zijn data vandaan haalt wordt niet beschouwd als een onderdeel van de applicatie en valt hier dan ook niet onder.

Tier 2 is de applicatie-laag waarin alle berekeningen plaats vinden. Invoer vanaf tier 3 wordt verwerkt, geïnterpreteerd en omgezet in acties gebruik makend van tier 1 met vaak weer een terugkoppeling naar de gebruiker via tier 3.

En tot slot is zoals gezegd tier 3 de laag waarin gecommuniceerd wordt met de gebruiker. Deze kan hier acties uitvoeren, objecten manipuleren en dat soort dingen.

De database-tier (de eerste) heeft hoofdzakelijk de MySQL-databaseserver als software nodig, met hieraan gelieerde hardwarematige kenmerken. Een hiervan is in ieder geval aanspraak gemaakt kan worden op voldoende geheugen.

De applicatie-tier (de tweede) is het deel wat door middel van PHP de door de gebruiker aangevraagde acties daadwerkelijk uitvoert. Hoewel de minimale systeemeisen hiervoor erg laag liggen, is het voor deze toepassing wel aangeraden niet te bezuinigen op geheugengebruik. Van PHP moet minimaal versie 4.1.0 aanwezig zijn voor gebruik van de superglobals.

De GUI-tier (de derde) is het deel wat zorgdraagt voor de gebruikersinterface. Op deze machines moet op een vloeiende manier gebruik gemaakt worden van een webbrowser.

2.2 Database

Voor tier 1 is een databasestructuur beschikbaar gemaakt. Voor alle duidelijk moet hierbij nog vermeldt wordt dat dit alleen de database voor de selectietool zelf is. Uit welke databases het systeem de enquête-data zelf haalt, wordt opgeslagen als data in deze meta-data: een niveau lager dus.

Als de structuur is geplaatst moeten sommige tabellen met de hand gevuld worden. Zo zijn er de acties zelf, maar ook de gebruikers en gebruikersgroepen t.b.v. het inloggen. Er is voor gekozen hier geen GUI voor te maken omdat deze functionaliteiten nauwelijks gebruikt zouden worden.

Alle attributen van de dbms-tabel zijn optioneel. Voor host wordt `localhost` als default genomen. Als de poort niet bekend is, wordt hier `3306` voor gebruikt. Zonder gebruikersnaam wordt er ingelogd als `root` en eventueel zonder wachtwoord.

Voor wat betreft de actie-tabel heeft deze ook een aantal optionele gegevens. Indien deze attributen niet worden aangegeven, wordt uitgegaan van de standaardlocaties. Voor de unsub kan ook worden aangegeven wat de waarde kan zijn om een deelnemer als uitgeschreven te kunnen beschouwen. Het attribuut enquetetabel is een prefix waar 'questions' en 'answers' achter wordt geplakt om achter een variabele-naam te kunnen komen. Zie §3.4 van de definitiestudie voor meer informatie.

Er is in de meta-data gebruik gemaakt van notatiewijzen voor het definiëren van automatische controle op de referentiële integriteit. Het systeem is gemaakt met de gedachte in het achterhoofd dat MySQL dit negeert en draagt dus zelf zorg voor de handhaving hiervan. Dit heeft gevolgen voor de voor MySQL onderliggende opslagstructuren; er kan het beste worden gekozen voor MyISAM.

2.3 PHP- en overige programmacode

Alle code is hiërarchisch in een archief beschikbaar gemaakt. Deze bestanden moeten uitgepakt worden op een gewenste locatie met behoudt van de hiërarchie. Alle bestanden hebben UNIX-regeleinden, dus enkel een newline. Dit levert complicaties op met het bewerken van deze bestanden in bijvoorbeeld Microsoft's Notepad.

PHP's instelling `magic_quotes_gpc` dient uit te staan.

2.4 Configuratie

In de map `includes` staat een bestand genaamd `config.php`. Het systeem is opgezet met de gedachte dat elk denkbare configureerbare aspect van het systeem in dit bestand te bereiken en te veranderen is. Tevens zijn er profielen ingebouwd om snel te kunnen schakelen tussen sets instellingen. Maak voor de installatie op een nieuwe machine een nieuw profiel aan, stel de juiste parameters in en loop ook de rest van het bestand nog even door voor mogelijk noodzakelijke veranderingen.

Als de configuratie is bijgewerkt, kan het systeem getest worden. Log hiervoor in met een eerder in de tabel `gebruiker` aangemaakte gebruiker, inclusief bijbehorende gegevens in de tabellen `gebruikersgroep` en `recht`. Bekijk hierna enkele objecten via het menu om er zeker van te zijn dat de configuratie juist is.

Het systeem heeft twee modi: ontwikkelmodus of productiemodus. Het is aan te bevelen tijdens het testen in de ontwikkelmodus te opereren, daar eventuele foutmeldingen hiermee direct op het scherm komen. Na configuratie is het zaak het systeem in productiemodus te zetten waarna eventuele systeemfouten geë-mailt worden naar het opgegeven e-mailadres.

2.5 Verhuizing

Bovenstaande procedure is ook te volgen bij verhuizing van het systeem, met de uitzondering dat al bestaande gegevens in de database hierbij gewoon meegenomen kunnen worden en er geen noodzaak meer bestaat tot het eerst bijwerken van de data waar geen GUI voor bestaat.

3 Onderhoudsprocedure

In dit hoofdstuk wordt ingegaan op de onderhoudsprocedure: welke handelingen moeten verricht worden om het systeem te onderhouden. Hierbij moet gedacht worden aan bijvoorbeeld het schoon houden van de database. Het systeem is over het algemeen onderhoudsvrij, maar er zijn toch twee punten die het vermelden waard zijn: vervuiling van de opslagstructuren, en het opraken van primaire sleutels.

De natuur van MySQL staat het toe dat indien gegevens worden verwijderd, deze in de onderliggende opslagstructuren slechts worden gemarkeerd als verwijderd. Dit heeft als gevolg dat ze wel schijfruimte in blijven nemen, maar biedt een hogere performantie bij verwijder-acties. Eens in de zoveel tijd moet voor alle tabellen de volgende query worden uitgevoerd: `OPTIMIZE TABLE tabel_1, tabel_2, ..., tabel_n;` waarbij alle tabellen van het systeem worden opgenomen. Dit onderhoud heeft een zeer lage prioriteit en kan prima ad hoc of jaarlijks worden uitgevoerd.

Voor een aantal objecten wordt gebruik gemaakt van een kunstmatig gegenereerd numeriek getal. De kolom waar dit getal in staat is niet oneindig. Hoewel er tijdens bouw gezorgd is voor voldoende ruimte voor de toekomst, kan hier een inschattingsfout in gemaakt zijn en de ruimte eerder op zijn dan verwacht wat problemen veroorzaakt bij het aanmaken van nieuwe objecten. Er zijn twee oplossingen:

- De efficiënte oplossing: primaire sleutels herplaatsen. Hierbij worden de 'vrije' plekken opnieuw ingenomen. Let hierbij wel op dat de vreemde sleutels ook worden bijgewerkt en dat het systeem op dat moment in onderhoudsmodus staat.
- De snelle oplossing: kolomtypes veranderen. Hierbij wordt de maximaal toegestane waarde simpelweg opgehoogd. Let ook hierbij op dat ook de kolomtypes van de vreemde sleutels worden bijgewerkt.

11 Plan van Aanpak pilot #2

1 Inleiding

Dit is het Plan van Aanpak van pilot #2. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de iteratie van de fase Pilotontwikkeling voor pilot GUI-standaard en is deels gebaseerd op het pilotplan uit de fase Definitiestudie. Deze pilot moet uitkomen op mogelijkheden voor het beheren van variabelen. Hieronder wordt verstaan het toevoegen, bewerken en verwijderen.

In hoofdstuk 2 worden de in deze iteratie van de fase Pilotontwikkeling uit te voeren activiteiten uiteengezet, tevens op basis van de voorlopige verdeling in pilotdelen die gemaakt is in de definitiestudie. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat dit zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
4 april 2005	1.0	Initiële versie.

Tabel 24 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze iteratie van de fase Pilotontwikkeling te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaan komt. Als basis worden de activiteiten die zijn opgesomd in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE
Overig

Tabel 25 Legenda

- 1 Pilotplan bijwerken
- 2 Plan van Aanpak opstellen
- 3 SUMI-test opstellen
- 4 Globaal-functionele structuur pilot specificeren
- 5 Globaal-technische structuur pilot specificeren
- 6 GUI-ontwerp
 - 1 Schermschetsen
 - 2 Navigatieschema
- 7 Testplan opstellen

- 8 Pilotontwikkelplan opstellen
- 9 Ontwerpen & bouwen van softwarebouweenheden, samenstellen handleidingen en opstellen invoeringsprocedures voor de volgende pilotdelen:
 - 1 Database
 - 2 Overzicht
 - 3 Toevoegen
 - 4 Bewerken
 - 5 Verwijderen
 - 6 Testen (van variabele)
 - 7 Code-validatie
 - 8 Code-kleurcoderingen
- 10 Beoordelen en testen pilot

Aanpassingen aan globaal Plan van Aanpak

De activiteit 'Globaal organisatorische inrichting specificeren' wordt overgeslagen omdat er bij deze pilot geen organisatorische aspecten zijn te vermelden.

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 8 en eindigt halverwege week 9.

Activiteit	Weken									
	8					9				
	M	D	W	D	V	M	D	W	D	V
Pilotplan bijwerken										
Plan van Aanpak opstellen										
SUMI-test opstellen										
Functionele structuur										
Technische structuur										
GUI-ontwerp										
Schermsschetsen										
Navigatieschema										
Testplan opstellen										
Pilotontwikkelplan opstellen										
Ontwerpen, bouwen, samen- en opstellen										
Database										
Overzicht										
Toevoegen										
Bewerken										
Verwijderen										
Testen (van variabele)										
Code-validatie										
Code-kleurcoderingen										
Beoordelen en testen										

Tabel 26 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 2: Plan van Aanpak
- 8: Pilotontwikkelpunten met de uitkomsten van de punten 3 t/m 7.
- Per softwarebouweenheid: de resultaten van het ontwerpen & bouwen, het samenstellen van de handleidingen en het opstellen van de invoeringsprocedures.
- De pilot zelf (programmacode, databasestructuren, enzovoorts).

12 Pilotontwikkelplan pilot #2

1 Inleiding

Dit is het Pilotontwikkelplan van pilot #2. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Plan van Aanpak behorende bij deze pilot. De definitiestudie is gebruikt als basis, en is waar nodig in het gebied van deze pilot nog aangevuld en uitgebreid. Deze pilot moet uitkomen op mogelijkheden voor het beheren van berekende variabelen. Hieronder wordt verstaan het toevoegen, bewerken en verwijderen.

Hoofdstuk 2 bevat de scenario's die doorlopen moeten worden tijdens de SUMI-test. Hoofdstuk 3 bevat de functionele structuur. Dit omvat een detaillering van zaken uit het systeemconcept in de definitiestudie. Concreet voor deze pilot betekend dit een beschrijving van de functionaliteiten. Hoofdstuk 4 bevat de technische structuur. De relevante objecten van het klassendiagram worden hier omgezet naar een relationeel implementatiemodel. Hoofdstuk 5 bevat een GUI-ontwerp in de vorm van schermschetsen en een navigatieschema en hoofdstuk 6 bevat tot slot een testplan waarin een aantal testscenario's zijn vastgelegd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
07-04-2005	1.0	Initiële versie.
26-05-2005	1.1	Nieuw type variabelen geïntroduceerd: speciaal. Dit om bijvoorbeeld om de datum/tijd van het antwoord (UNIX-tijd) te kunnen bereiken.

Tabel 27 Veranderingengeschiedenis.

2 SUMI-test

Voor het systeem zal een SUMI-test opgesteld worden. SUMI staat voor *Software Usability Measurement Inventory* en is de de facto standaard voor het objectief meten van de gebruikervriendelijkheid van software. Ondanks dat de SUMI-test pas aan het einde zal worden uitgevoerd als alle pilots ontwikkeld zijn, worden hier al enkele scenario's met betrekking tot deze pilot opgesteld die tijdens de SUMI-test doorlopen dienen te worden.

De scenario's zijn als volgt:

- Er moet een nieuwe normale variabele worden toegevoegd. Voor het kiezen van de systeemnaam moet gebruik gemaakt worden van de wizard (achter de knop 'Kies') waarna er een willekeurige variabele gekozen moet worden.
- Er moet een berekende variabele gewijzigd worden; hierbij moet de PHP-code op een dusdanige manier veranderd worden dat er van een extra variabele gecontroleerd wordt of deze een bepaalde waarde heeft.

3 Functionele structuur

In dit hoofdstuk staat de functionele structuur beschreven. Het bestaat uit een tekstuele omschrijving van de functionele eisen aan het overzicht van de variabelen, het toevoegen, bewerken en verwijderen van een variabele en het testen van berekende variabelen. De als luxe geprioriteerde pilotdelen Code-validatie en Code-kleurcoderingen voor wat betreft berekende variabelen worden ook omschreven. Er is overal rekening gehouden met de mogelijkheden van de GUI-standaard van pilot #1.

3.1 Overzicht

Het overzicht van variabelen geeft een overzicht weer van alle op dat moment bij het systeem aangemaakte variabelen. Van elke variabele wordt weergegeven wat het type, de systeemnaam, de databasenaam en de omschrijving is. De systeemnaam is een aparte naam die aan de variabele meegegeven kan worden, maar indien weggelaten wordt de databasenaam overgenomen. De databasenaam is de naam zoals deze nu in de bestaande opslagstructuren voorkomt.

3.2 Toevoegen

Bij het toevoegen van een variabele moet het volgende proces doorlopen kunnen worden:

- 1 Type kiezen: normaal, profiel, gesponsord, berekend en speciaal.
- 2 Ingeven van een systeemnaam.
- 3 Bij normale, profiel- en gesponsorde variabelen, één van deze mogelijkheden:
 - 1 ingeven van een databasenaam;
 - 2 starten van een wizard welke een databasenaamachterhaald middels het kiezen van een F2-actie, een vraag en een antwoord.
- 4 Bij berekende variabelen:
 - 1 Kiezen van één of meer andere variabelen die gebruikt gaan worden in de PHP-code.
 - 2 Invoeren van de PHP-code zelf.
- 5 Bij speciale variabelen:

- 1 Kiezen van een subtype. Op dit moment is hier alleen nog maar de mogelijkheid om te kiezen voor het moment dat het antwoord is opgeslagen (als UNIX-tijd).
- 6 Ingeven van een omschrijving.

3.3 Bewerken

Bij het bewerken kan hetzelfde proces als bij §3.2 doorlopen worden, alleen dan voor het bewerken van gegevens in plaats van het toevoegen.

3.4 Verwijderen

De volgende relaties kunnen afhankelijk zijn van een variabele:

- rauwe resultaatsets;
- selecties (groepering op);
- selecties (lezen van);
- selecties (ontdubbelen op);
- andere berekende variabelen;
- voorwaarden.

Als dit het geval is is het onmogelijk een variabele te verwijderen. Aan de gebruiker zal bekend worden gemaakt waarom niet; hoeveel rauwe resultaatsets er van afhankelijk zijn, hoeveel selecties, enzovoorts.

3.5 Testen

Als afzonderlijke actie van een berekende variabele kan deze getest worden. Dit proces verloopt dan in twee stappen: allereerst moeten alle variabelen waar de berekende variabele gebruik van maakt ingevuld worden met dummy-data. In de tweede stap wordt de code uitgevoerd met de ingevoerde dummy-data en de uitvoer wordt getoond.

3.6 Code-validatie

Na het aanmaken of bewerken van een berekende variabele wordt automatisch gecontroleerd of de ingegeven code valide PHP vormt. Indien dit niet het geval is, vormt dit extra feedback en is het onmogelijk de onjuiste code op te slaan.

Merk op dat dit alleen werkt op syntactische fouten, semantische fouten zijn van een hoger niveau en hier zal niet op gecontroleerd worden.

3.7 Code-kleurcoderingen

Tijdens het aanmaken of bewerken van een berekende variabele bestaat een mogelijkheid om van de op dat moment ingegeven code een kleurenversie weer te geven.

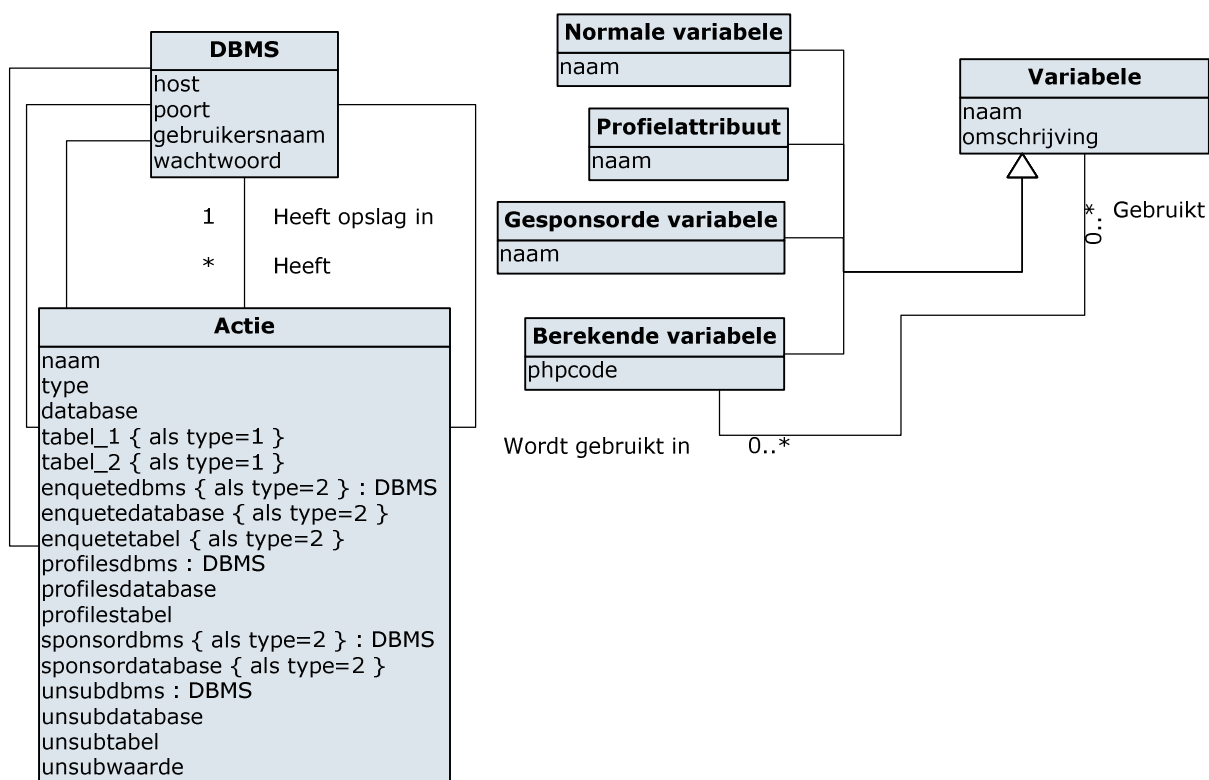
4 Technische structuur

In dit hoofdstuk staat de technische structuur van deze pilot. De relevante objecten van het klassendiagram worden omgezet in een relationeel representatiemodel en in een relationeel implementatiemodel.

Met het klassendiagram (of een deel ervan) zijn de objecten, attributen en relaties in kaart gebracht. In het proces tot het implementeren van de database, kan niet alles hieruit 1-op-1 worden overgenomen. De vertaalslag tussen het klassendiagram en het relationeel implementatiemodel vindt plaats in de tussenliggende activiteit van het samenstellen van een relationeel representatiemodel. Hierbij worden ook alle normalisaties behandeld. Vanuit het relationeel representatiemodel ligt de weg vrij tot het maken van een relationeel implementatiemodel.

4.1 Klassendiagram

Zie Figuur 18 voor het klassendiagram specifiek aan de variabelen.



Figuur 18 Het klassendiagram.

Dit is inclusief het Actie- en het DBMS-object omdat deze benodigd zijn voor de wizard bij het selecteren van een variabele van een actie van de F2-technologie. Voor de rest hebben deze objecten geen directe relatie met variabelen.

Voor de speciale variabele hoort hier nog een subklasse bij. Deze klasse heeft een naam en een index.

4.2

Relationeel representatiemodel

Zoals in de inleiding al vermeld, is dit relationeel representatiemodel een tussenstap tussen het klassendiagram en het implementatiemodel. In deze paragraaf wordt elk object apart behandeld met vastlegging van het verwerken van de correlaties en het normalisatieproces. Zie §3.1.2 van het pilotontwikkelplan van pilot #1 voor wat achtergrondinformatie rond het globaal gevolgde normalisatieproces en het toevoegen van extra numerieke primaire sleutels ondanks dat er één of meer kandidaatsleutels bestaan.

DBMS

Initieel bestaat de volgende relatie:

```
dbms(host, poort, gebruikersnaam, wachtwoord)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
dbms(id, host, poort, gebruikersnaam, wachtwoord)
```

Actie

Initieel bestaat de volgende relatie

```
actie(naam, type, database, tabel_1, tabel_2, enquetedbmsId,
      enquetedatabase, enquetetabel, profilesdbmsId,
      profilesdatabase, profilestabel, sponsordbmsId,
      sponsordatabase, unsubdbmsId, unsubdatabase, unsubtabel,
      unsubwaarde)
```

Een Actie hoort altijd bij 1 DBMS voor de bepaling van waar de gegevens van de enquetes zelf zijn opgeslagen. Via *enquetedbms*, *profiledbms*, *sponsordbms* en *unsubdbms* kan een ander DBMS gekozen worden specifiek voor respectievelijk de enquête, de profile-tabel, de sponsor-tabellen en de unsub-tabel. Met toevoeging van een extra numerieke primaire sleutel komen we uiteindelijk op de volgende relatie:

```
actie(id, dbmsId, naam, type, database, tabel_1, tabel_2,
      enquetedbmsId, enquetedatabase, enquetetabel,
      profilesdbmsId, profilesdatabase, profilestabel,
      sponsordbmsId, sponsordatabase, unsubdbmsId,
      unsubdatabase, unsubtabel, unsubwaarde)
```

- *dbmsId*: vreemde sleutel, verwijst naar *id* in *dbms*, null niet toegelaten.
- *enquetedbmsId*: vreemde sleutel, verwijst naar *id* in *dbms*, null toegelaten.

- `profilesdbmsId`: vreemde sleutel, verwijst naar `id` in `dbms`, null toegelaten.
- `sponsordbmsId`: vreemde sleutel, verwijst naar `id` in `dbms`, null toegelaten.
- `unsubdbmsId`: vreemde sleutel, verwijst naar `id` in `dbms`, null toegelaten.

Variabele

Initiël bestaan de volgende relaties:

```
variabele(naam, omschrijving)
var_norm(naam)
var_profiel(naam)
var_sponsor(naam)
var_berekend(naam, phpcode)
var_speciaal(naam, index)
```

Aan alle relaties wordt een extra numerieke primaire sleutel toegevoegd. Er bestaat tevens een veel-op-veel relatie tussen een berekende variabele en variabele wat wordt opgenomen als een afronderlijke relatie:

```
variabele(id, naam, omschrijving)
var_norm(id, variabeleId, naam)
var_profiel(id, variabeleId, naam)
var_sponsor(id, variabeleId, naam)
var_berekend(id, variabeleId, naam, phpcode)
var_speciaal(id, naam, index)
variabele_var_berekend(varBerekendId, variabeleId)
```

Maar: een variabele is in het klassendiagram gemodelleerd als een superklasse met vijf subklassen ter representatie van normale variabelen, profielattributen, gesponsorde variabelen, berekende variabelen en speciale variabelen. Er zijn verschillende manieren om dit om te zetten in een stukje relationeel representatiemodel, maar er is voor gekozen om alle objecten in één relatie te gieten en met een extra attribuut te bepalen van welk soort de relatie is:

```
variabele(id, type, systeemnaam, omschrijving, databasenaam,
        phpcode, speciaal)
variabele_variabele(varBerekendId, variabeleId)
```

- `varBerekendId`: vreemde sleutel, verwijst naar `id` in `variabele` waarbij de variabele een berekende variabele is, null niet toegelaten.
- `variabeleId`: vreemde sleutel, verwijst naar `id` in `variabele`, null niet toegelaten.

Het gegeven `speciaal` bevat de data die normaal onder de naam `index` bekend stond. Dit omdat `index` voor in de toekomst misschien wat te algemeen opgevat wordt en zo voor verwarring kan zorgen.

4.3

Relationeel implementatiemodel

Nu het relationeel representatiemodel er is, kan het relationeel implementatiemodel worden gemaakt. Dit gebeurt middels SQL als DDL.

DBMS

```
CREATE TABLE dbms
(
    id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    host TINYTEXT,
    poort SMALLINT UNSIGNED,
    gebruikersnaam TINYTEXT,
    wachtwoord TINYTEXT,
    PRIMARY KEY (id)
);
```

Actie

Omdat 'database' voor MySQL een gereserveerde term is, is het attribuut hernoemd naar 'databasenaam'.

```
CREATE TABLE actie
(
    id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    dbmsId SMALLINT UNSIGNED NOT NULL,
    naam TINYTEXT NOT NULL,
    type ENUM('1', '2') NOT NULL,
    databasenaam TINYTEXT NOT NULL,
    tabel_1 TINYTEXT NOT NULL,
    tabel_2 TINYTEXT,
    enquetedbmsId SMALLINT UNSIGNED,
    enquetedatabase TINYTEXT,
    enquetetabel TINYTEXT,
    profilesdbmsId SMALLINT UNSIGNED,
    profilesdatabase TINYTEXT,
    profilestabel TINYTEXT,
    sponsordbmsId SMALLINT UNSIGNED,
    sponsordatabase TINYTEXT,
    unsubdbmsId SMALLINT UNSIGNED,
    unsubdatabase TINYTEXT,
    unsubtabel TINYTEXT,
    unsubwaarde TINYTEXT,
    PRIMARY KEY (id),
    FOREIGN KEY (dbmsId)
        REFERENCES dbms ON DELETE CASCADE,
    FOREIGN KEY (enquetedbmsId)
        REFERENCES dbms ON DELETE RESTRICT,
    FOREIGN KEY (profilesdbmsId)
        REFERENCES dbms ON DELETE RESTRICT,
    FOREIGN KEY (sponsordbmsId)
        REFERENCES dbms ON DELETE RESTRICT,
    FOREIGN KEY (unsubdbmsId)
```

```
REFERENCES dbms ON DELETE RESTRICT
);
```

Variabele

Bij de relatie variabele behoren twee tabellen, variabele en variabele_variabele. Allereerst de tabel variabele.

```
CREATE TABLE variabele
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    type ENUM('normaal', 'profiel', 'sponsor',
        'berekend', 'speciaal') NOT NULL,
    systeemnaam TINYTEXT,
    omschrijving TEXT,
    databasenaam TINYTEXT,
    phpcode TEXT,
    PRIMARY KEY (id)
);
```

En vervolgens de tabel variabele_variabele voor de vastlegging van een veel-op-veel relatie tussen berekende variabelen en variabelen.

```
CREATE TABLE variabele_variabele
(
    varBerekendId MEDIUMINT UNSIGNED NOT NULL,
    variabeleId MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (varBerekendId, variabeleId),
    FOREIGN KEY (varBerekendId)
        REFERENCES variabele ON DELETE CASCADE,
    FOREIGN KEY (variabeleId)
        REFERENCES variabele ON DELETE RESTRICT
);
```

5 GUI-ontwerp

In dit hoofdstuk komt het ontwerp van de GUI aan bod. Aan de hand van de al gemaakte schetsen in het pilotontwikkelplan van pilot #1, wordt hier een invulling gemaakt specifiek voor het beheer van variabelen.

Allereerst komen er een aantal schermschetsen voorbij, vervolgens gevolgd voor een navigatieschema.

5.1 Schermschetsen

In deze paragraaf staan de schermschetsen voor de volgende schermen²:

- Overzichtscherm
- Toevoeg- & bewerk scherm

² De cursief weergegeven elementen hebben geen schermschets.

- Wizard kiezen databasenaam
 - Overzichtscherms acties
 - Overzicht v.d. vragen
- Testen
 - Invoer
 - Resultaat

Het overzichtscherms, zie Figuur 19, is het centrale punt rond het beheren van variabelen. Het geeft een overzicht van alle aanwezige variabelen, en van hieruit kunnen variabelen worden toegevoegd, bewerkt, verwijderd en getest.

een test-actie

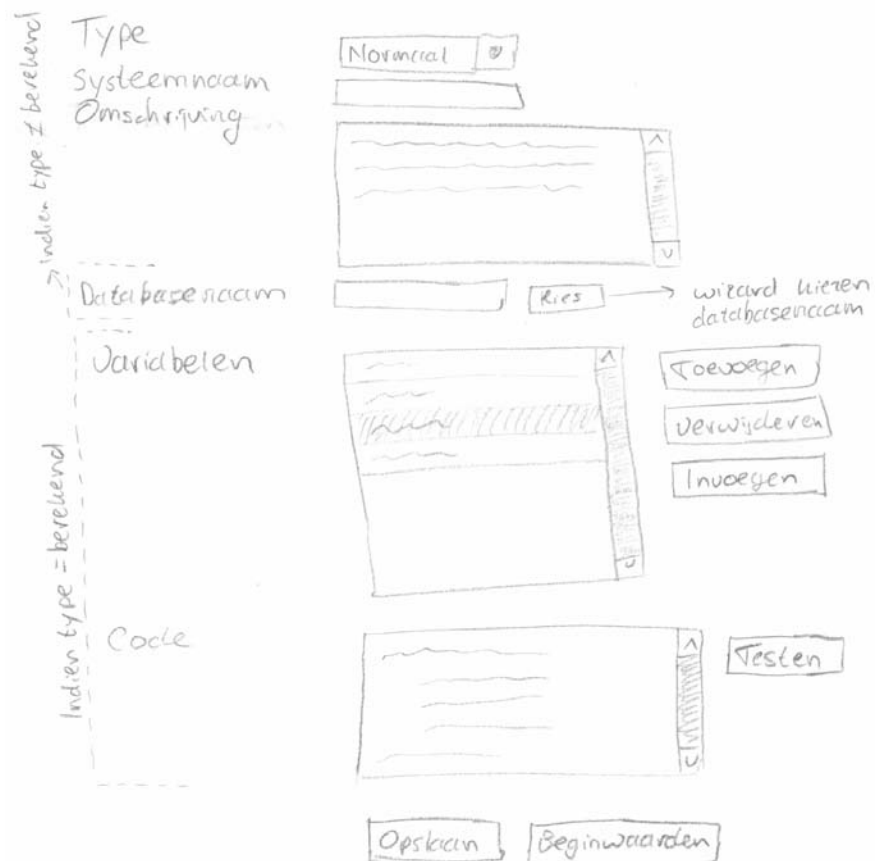
		Type	Systeemnaam	Databasenaam	Omschrijving
☐	B	✓	Normal	~	~
☐	B	U	Bevriend	~	~
☐	B	✓	~	~	~
Selecteer alles					
Selecteer niets					
Inverteer selectie					
Verwijderen					
→ Toevoegen					

Figuur 19 Het overzicht van variabelen.

Figuur 20 is een schets van het scherm waarmee een variabele kan worden toegevoegd of bewerkt. Een aantal elementen verschijnt wel of niet, nagelang of het een berekende variabele is.

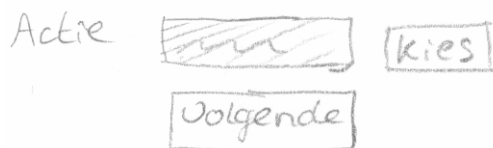
Indien er op de 'Toevoegen' knop wordt geklikt, opent zich in een subwindow een nieuw overzicht van variabelen waarvandaan een variabele geselecteerd kan worden.

Voor de luxe-pilotdelen voor het valideren of gekleurd weergeven van de code, is onder de 'Testen'-knop nog plaats voor twee andere knoppen 'Valideren' en 'Kleurcoderen'.



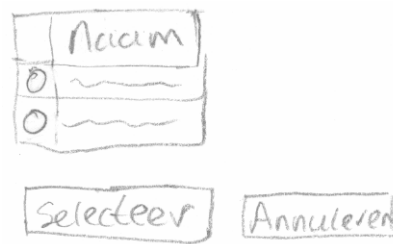
Figuur 20 Het toevoeg- & bewerksscherm.

De databasenaam is de naam zoals de variabele in de achterliggende al bestaande systemen heet. Door middel van een wizard kan deze naam eenvoudig worden achterhaald. De wizard opent zich in een subwindow na het klikken op 'Kies' waar Figuur 21 illustreert wat er dan op het scherm verschijnt. Deze knop verschijnt overigens alleen als er op dat moment een normale variabele wordt aangemaakt of bewerkt.



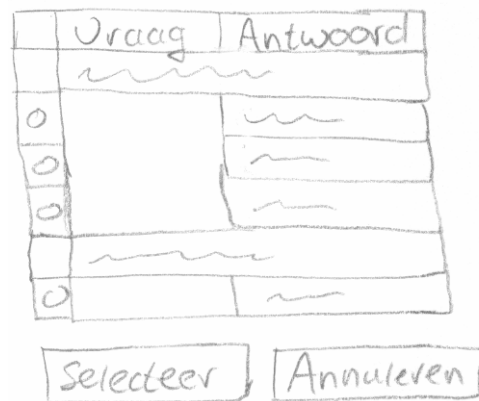
Figuur 21 Stap 1 van de wizard.

Door de 'Kies'-knop te activeren, wordt er weer een subwindow geopend waarin alle F2-acties zijn opgesomd. Zie Figuur 22.



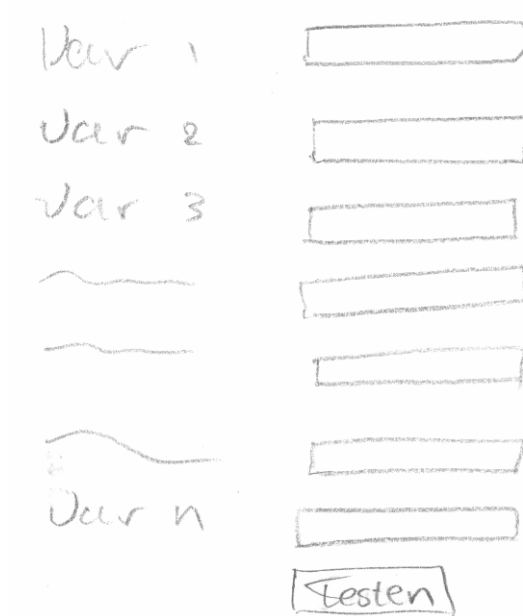
Figuur 22 Selectie van een F2-actie.

Als er na het selecteren op 'Volgende' wordt gedrukt, volgt er een overzicht op het scherm met alle bij de vraag behorende vragen en databasenames, zie Figuur 23.



Figuur 23 Selectie van een databasenaam.

Het testen van een berekende variabele vindt ook plaats in twee stappen. Allereerst wordt de invoer gevraagd van alle variabelen waar de te testen variabele afhankelijk van is, en vervolgens kan hiervan de uitvoer worden opgeroepen. Zie Figuur 24.

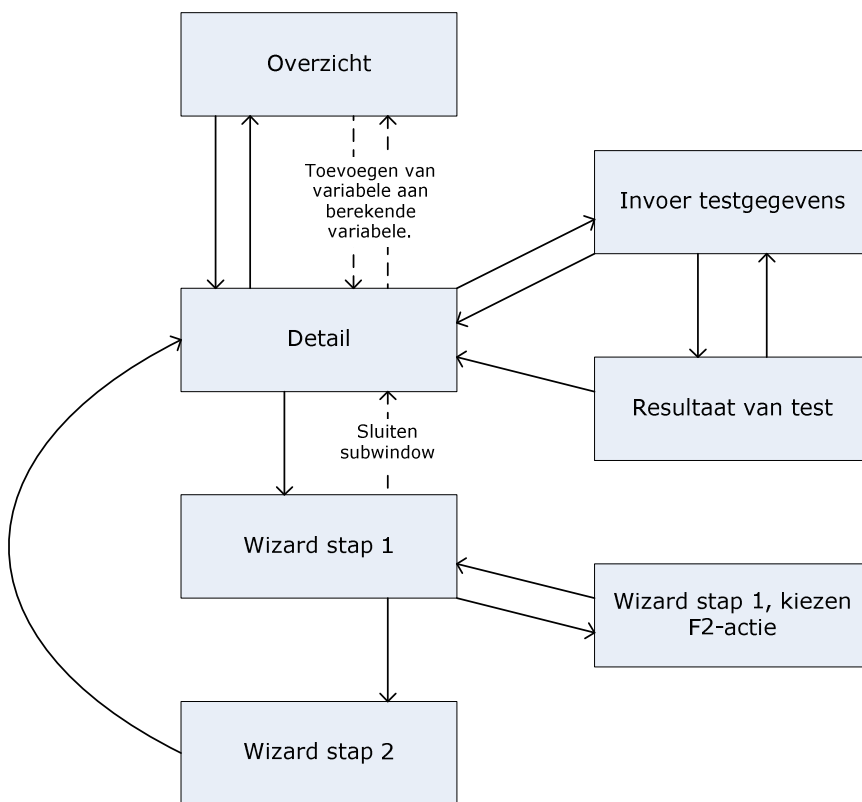


Figuur 24 Invoer van testgegevens.

5.2

Navigatieschema

Hieronder staat het navigatieschema voor de schermen die in deze pilot zijn omschreven, zie Figuur 25.



Figuur 25 Navigatieschema.

6 Testplan

In dit hoofdstuk komt het testen aan bod. Er worden een aantal testscenario's samengesteld en uitgelegd. Tijdens de activiteit Beoordelen en testen moeten deze scenario's met goed gevolg gevolgd worden.

De scenario's zijn als volgt:

- Voeg een nieuwe normale variabele toe en kies een systeemnaam door gebruik te maken van de wizard.
- Voeg een berekende variabele toe met drie andere variabelen als basis.
- Bewerk een berekende variabele zodat een kringverwijzing ontstaat in de hiërarchie van afhankelijkheden. De variabele mag niet opgeslagen worden, en er moet ook een melding over worden weergegeven.

13 Plan van Aanpak pilot #3

1 Inleiding

Dit is het Plan van Aanpak van pilot #3. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de iteratie van de fase Pilotontwikkeling voor de pilot Variabelebeheer en is deels gebaseerd op het pilotplan uit de fase Definitiestudie. Deze pilot moet uitkomen op mogelijkheden voor het beheren van selecties. Hieronder wordt verstaan het toevoegen, bewerken en verwijderen.

In hoofdstuk 2 worden de in deze iteratie van de fase Pilotontwikkeling uit te voeren activiteiten uiteengezet, tevens op basis van de voorlopige verdeling in pilotdelen die gemaakt is in de definitiestudie. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat dit zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
4 april 2005	1.0	Initiële versie.

Tabel 28 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze iteratie van de fase Pilotontwikkeling te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaankomt. Als basis worden de activiteiten die zijn opgesomd in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE
Overig

Tabel 29 Legenda

- 1 Pilotplan bijwerken
- 2 Plan van Aanpak opstellen
- 3 SUMI-test opstellen
- 4 Globaal-functionele structuur pilot specificeren
- 5 Globaal-technische structuur pilot specificeren
- 6 GUI-ontwerp
 - 1 Schermschetsen
 - 2 Navigatieschema
- 7 Testplan opstellen

- 8 Pilotontwikkelpplan opstellen
- 9 Ontwerpen & bouwen van softwarebouweenheden, samenstellen handleidingen en opstellen invoeringsprocedures voor de volgende pilotdelen:
 - 1 Database
 - 2 Overzicht
 - 3 Toevoegen
 - 4 Bewerken
 - 5 Verwijderen
 - 6 Doorgeven
 - 7 Inplannen
- 10 Beoordelen en testen pilot

Aanpassingen aan globaal Plan van Aanpak

De activiteit 'Globaal organisatorische inrichting specificeren' wordt overgeslagen omdat er bij deze pilot geen organisatorische aspecten zijn te vermelden.

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 9 en eindigt halverwege week 11.

Activiteit	Weken														
	9					10					11				
	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V
Pilotplan bijwerken															
Plan van Aanpak opstellen															
SUMI-test opstellen															
Functionele structuur															
Technische structuur															
GUI-ontwerp															
Schermsschetsen															
Navigatieschema															
Testplan opstellen															
Pilotontwikkelpplan opstellen															
Ontwerpen, bouwen, samen- en opstellen															
Database															
Overzicht															
Toevoegen															
Bewerken															
Verwijderen															
Doorgeven															
Inplannen															
Beoordelen en testen															

Tabel 30 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 2: Plan van Aanpak

- 8: Pilotontwikkelpunten met de uitkomsten van de punten van 3 t/m 7.
- Per softwarebouweenheid: de resultaten van het ontwerpen & bouwen, het samenstellen van de handleidingen het opstellen van de invoeringsprocedures.

De pilot zelf (programmacode, databasestructuren, enzovoorts).

14 Pilotontwikkelplan pilot #3

1 Inleiding

Dit is het Pilotontwikkelplan van pilot #3. Het is voor het afstuderen in de periode 2005-1 welke geschied bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Plan van Aanpak behorende bij deze pilot. De definitiestudie is gebruikt als basis, en waar nodig in het gebied van deze pilot nog aangevuld en uitgebreid. Deze pilot moet uitkomen op mogelijkheden voor het beheren van selecties. Hieronder wordt verstaan het toevoegen, bewerken en verwijderen.

Hoofdstuk 2 bevat de scenario's die doorlopen moeten worden tijdens de SUMI-test. Hoofdstuk 3 bevat de functionele structuur. Dit omvat een detaillering van zaken uit het systeemconcept in de definitiestudie. Concreet voor deze pilot betekend dit een beschrijving van de functionaliteiten. Hoofdstuk 4 bevat de technische structuur. De relevante objecten van het klassendiagram worden hier omgezet naar een relationeel implementatiemodel. Hoofdstuk 5 bevat een GUI-ontwerp in de vorm van schermschetsen en een navigatieschema en hoofdstuk 6 bevat tot slot een testplan waarin een aantal testscenario's zijn vastgelegd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
12 april 2005	1.0	Initiële versie.

Tabel 31 Veranderingengeschiedenis.

2 SUMI-test

Voor het systeem zal een SUMI-test opgesteld worden. SUMI staat voor *Software Usability Measurement Inventory* en is de de facto standaard voor het objectief meten van de gebruikersvriendelijkheid van software. Ondanks dat de SUMI-test pas aan het einde zal worden uitgevoerd als alle pilots ontwikkeld zijn, worden hier al enkele scenario's met betrekking tot deze pilot opgesteld die tijdens de SUMI-test doorlopen dienen te worden.

De scenario's zijn als volgt:

- Er moet een nieuwe selectie worden toegevoegd met drie voorwaarden die aan elkaar gekoppeld zijn middels een logische 'en'.
- De zojuist aangemaakte selectie moet worden gewijzigd: er moet, wederom met een logische 'en' ook gezocht worden op twee andere voorwaarden in een groep die aan elkaar zijn gekoppeld middels een logische 'of'.
- Er moet een selectie worden aangemaakt met de laatst gewijzigde selectie als selectiedeel voor een nieuwe selectie.

3 Functionele structuur

In dit hoofdstuk staat de functionele structuur beschreven. Het bestaat uit een tekstuele omschrijving van de functionele eisen aan het overzicht van selecties, het toevoegen, bewerken, verwijderen en doorgeven van een selectie. Het als luxe geprioriteerde pilotdeel Inplannen wordt ook omschreven. Er is overal rekening gehouden met de mogelijkheden van de GUI-standaard van pilot #1.

3.1 Overzicht

Het overzicht van selecties geeft een overzicht weer van alle op dat moment bij het systeem aangemaakte selecties waar . Van elke selectie wordt weergegeven wat de naam en de omschrijving is.

3.2 Toevoegen

Het toevoegen van een selectie is een complexe taak, en vergt veel functionaliteiten om te zorgen dat deze complexe taak goed uitgevoerd kan worden.

Een selectie bestaat uit één of meer van de volgende delen:

- een selectie;
- een groep;
- een voorwaarde.

De volgende selecties zijn dus bijvoorbeeld mogelijk:

- drie voorwaarden;
- een selectie en een groep;
- twee selecties en twee groepen;
- een groep en een voorwaarde;
- een voorwaarde en een selectie.

De complexiteit zit hem in het feit dat selecties en groepen op hun beurt ook weer bestaan uit meerdere delen waarbij het verhaal weer van voren af aan begint.

Een selectie ligt vast indien deze als deel in een andere selectie wordt opgenomen, wel kunnen de onderdelen ervan in een subwindow worden bewerkt. Door meerdere selectiedelen te selecteren en deze vervolgens te laten inspringen middels een druk op een knop, wordt er

een groep gecreëerd. Van elke groep kan vervolgens worden aangegeven wat de naam en de omschrijving is en of deze in andere selecties herbruikbaar is.

Het volgende proces moet doorgelopen kunnen worden bij het toevoegen van een selectie:

- 1 Ingeven van een naam.
- 2 Ingeven van een omschrijving.
- 3 Kiezen uit welke acties de data gehaald moet worden.
- 4 Kiezen welke variabelen er gelezen worden.
- 5 Ingeven van selectiedelen, een van de volgende acties:
 - 1 Toevoegen van voorwaarde:
 - 1 kiezen van een variabele;
 - 2 ingeven van omschrijving;
 - 3 ingeven van brontype (tekenreeks of getal);
 - 4 ingeven van operatie:
 - 1 indien brontype is tekenreeks: gelijk, ongelijk, komt voor of komt niet voor;
 - 2 indien brontype is getal: groter dan, kleiner dan, groter dan of gelijk aan, kleiner dan of gelijk aan of ongelijk aan;
 - 5 ingeven van één of meer waarde(n);
 - 6 ingeven van een reguliere expressie.
 - 2 Toevoegen van een selectie.
 - 3 Groeperen van delen.
 - 4 Degroeperen van delen.
 - 5 Naam, omschrijving en of een groep herbruikbaar is ingeven.
 - 6 Een al eerder ingegeven herbruikbare groep selecteren.
- 6 Ingeven van een beperking op datum/tijd van laatste invoer van een profiel.
- 7 Ingeven van een beperking op datum/tijd van antwoord.
- 8 Type nabewerking kiezen.
- 9 Maximaal één of twee groepeer-variabelen kiezen.
- 10 Maximaal één telvoorwaarde kiezen.
- 11 Eventueel een ontdubbelingsvariabele kiezen.
- 12 Aangeven of de resultaatset relatief moet zijn.

3.3 Bewerken

Bij het bewerken kan hetzelfde proces als bij §3.2 doorlopen worden, alleen dan voor het bewerken van gegevens in plaats van het toevoegen.

3.4 Verwijderen

Er zijn geen relaties met objecten die verwijdering in de weg zouden staan. Wel moet voor een verwijder actie een bevestiging gevraagd worden aan de gebruiker met eventueel de melding van hoeveel resultaatsets er met de verwijdering meeverwijderd worden.

3.5 Doorgeven

Als een selectie wordt doorgegeven, wordt deze beschikbaar gemaakt voor een andere gebruiker. Op deze manier kunnen gebruikers die hier normaliter geen mogelijkheden voor hebben, toch de beschikking hebben over complexere of uitgebreidere selecties. Bij het doorgeven moeten een aantal rechten gegeven worden die gaan gelden voor dat object gekoppeld aan een specifieke gebruiker.

3.6 Inplannen

Door een selectie in te plannen, bestaat de mogelijkheid om een selectie op een later tijdstip automatisch uit te laten voeren. Bij het inplannen zelf wordt aangegeven wanneer het plaats dient te vinden. Aan de kant van de uitvoering moet software ontwikkeld worden welke periodiek kijkt of er ingeplande selecties bestaan die in aanmerking komen voor activatie.

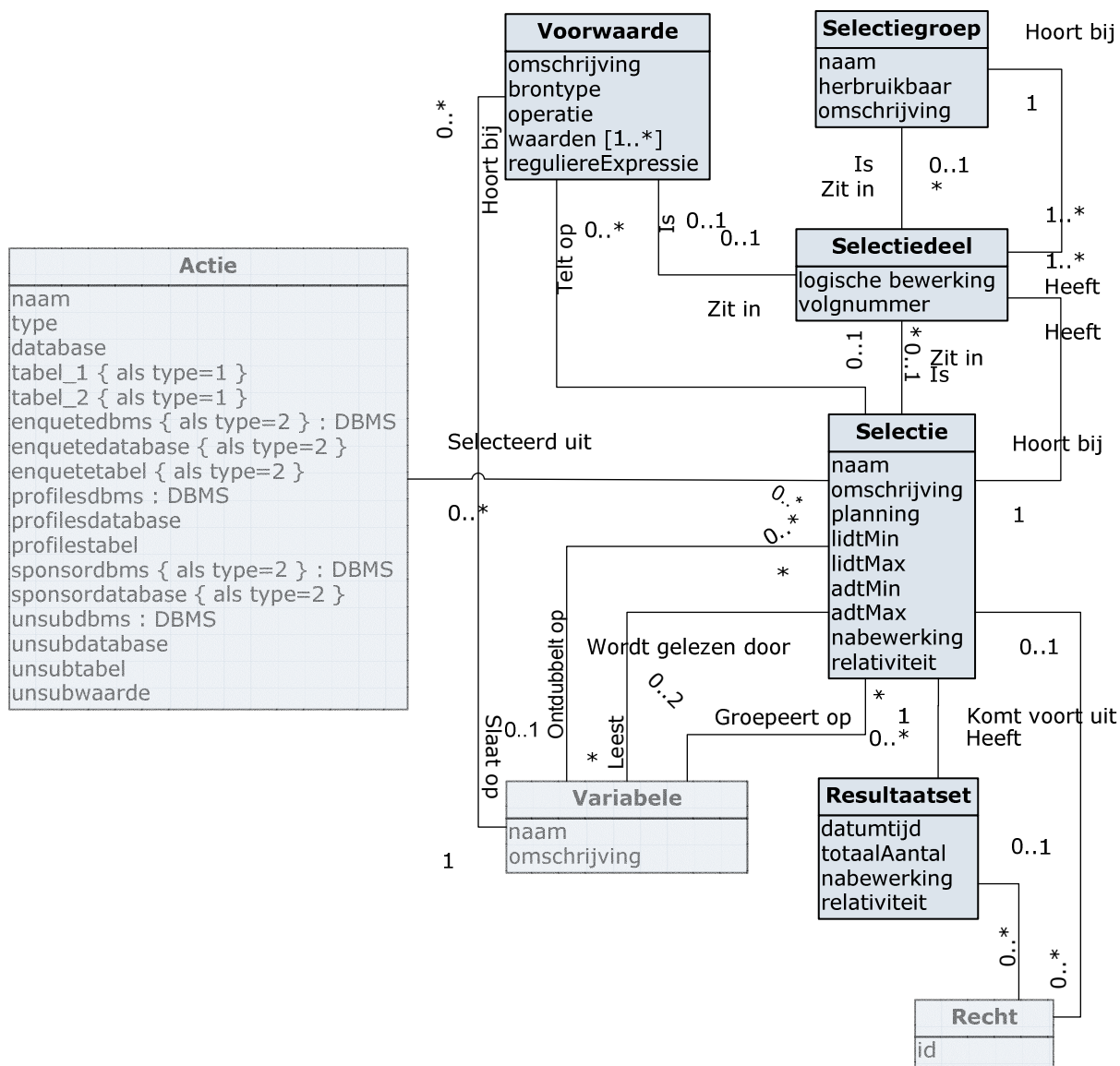
4 Technische structuur

In dit hoofdstuk staat de technische structuur van deze pilot. De relevante objecten van het klassendiagram worden omgezet in een relationeel representatiemodel en in een relationeel implementatiemodel.

Met het klassendiagram (of een deel ervan) zijn de objecten, attributen en relaties in kaart gebracht. In het proces tot het implementeren van de database, kan niet alles hieruit 1-op-1 worden overgenomen. De vertaalslag tussen het klassendiagram en het relationeel implementatiemodel vindt plaats in de tussenliggende activiteit van het samenstellen van een relationeel representatiemodel. Hierbij worden ook alle normalisaties behandeld. Vanuit het relationeel representatiemodel ligt de weg vrij tot het maken van een relationeel implementatiemodel.

4.1 Klassendiagram

Zie Figuur 26 voor het klassendiagram specifiek aan de selecties.



Figuur 26 Het klassendiagram.

De gedimde klassen zijn reeds aanwezig, maar alsnog opgenomen vanwege nog toe te voegen relaties. Van reeds bestaande entiteiten waaraan nieuwe relaties moeten worden toegevoegd, worden alleen de wijzigingen op het relationeel representatiemodel gegeven. Datzelfde geldt voor het relationeel implementatiemodel waarvoor DDL wordt gebruikt om al bestaande structuren aan te passen (SQL's `ALTER TABLE`).

4.2 Relationeel representatiemodel

Zoals in de inleiding al vermeld, is dit relationeel representatiemodel een tussenstap tussen het klassendiagram en het implementatiemodel. In deze paragraaf wordt elk object apart behandeld met vastlegging van het verwerken van de correlaties en het normalisatieproces. Zie §3.1.2 van het pilotontwikkelplan van pilot #1 voor wat achtergrondinformatie rond het globaal gevolgde normalisatieproces en

het toevoegen van extra numerieke primaire sleutels ondanks dat er één of meer kandidaatsleutels bestaan.

Voorwaarde

Initieel bestaat de volgende relatie:

```
voorwaarde(omschrijving, brontype, operatie, waarden,  
           reguliereExpressie, variabeleId, selectieId)
```

Het attribuut `waarden` is hier meerwaardig, en door het normalisatieproces wordt dit een afzonderlijke relatie. Tezamen met het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
voorwaarde(id, omschrijving, brontype, operatie,  
           reguliereExpressie, variabeleId, selectieId)
```

- `variabeleId`: vreemde sleutel, verwijst naar `id` in `variabele`, null niet toegelaten.
- `selectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null toegelaten.

```
voorwaarde_waarde(voorwaardeId, waarde)
```

- `voorwaardeId`: vreemde sleutel, verwijst naar `id` in `voorwaarde`, null niet toegelaten.

Deze laatste relatie `voorwaarde_waarde` heeft geen primaire sleutel omdat de individuele waarden nooit individueel geïdentificeerd hoeven worden.

Selectiegroep

Initieel bestaat de volgende relatie:

```
selectiegroep(naam, herbruikbaar, omschrijving)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
selectiegroep(id, naam, herbruikbaar, omschrijving)
```

Selectie

Initieel bestaat de volgende relatie:

```
selectie(naam, omschrijving, planning, lidtMin, lidtMax,  
         adtMin, adtMax, nabewerking, relativiteit,  
         ontdubbelVariabeleId)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
selectie(id, naam, omschrijving, planning, lidtMin, lidtMax,  
         adtMin, adtMax, nabewerking, relativiteit,  
         ontdubbelVariabeleId)
```

- `ontdubbelVariabeleId`: vreemde sleutel, verwijst naar `id` in `variabele`, null toegelaten.

Er zijn ook nog een aantal veel-op-veel relaties met andere objecten, te weten:

- Een relatie met `variabele` om vast te kunnen leggen welke variabelen er worden geselecteerd voor uitvoer.
- Een relatie met `variabele` om vast te kunnen leggen waarop gegroepeerd wordt.
- Een relatie met `actie` om vast te kunnen leggen uit welke acties een selectie selecteert.

Als we deze relaties verwerken, ontstaan deze volgende relaties:

`selectie_uitvoer(selectieId, variabeleId)`

- `selectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null niet toegelaten.
- `variabeleId`: vreemde sleutel, verwijst naar `id` in `variabele`, null niet toegelaten.

`selectie_groepering(selectieId, variabeleId)`

- `selectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null niet toegelaten.
- `variabeleId`: vreemde sleutel, verwijst naar `id` in `variabele`, null niet toegelaten.

`selectie_actie(selectieId, actieId)`

- `selectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null niet toegelaten.
- `actieId`: vreemde sleutel, verwijst naar `id` in `actie`, null niet toegelaten.

Selectiedeel

Initieel bestaat de volgende relatie:

`selectiedeel(bronSelectieId, bronSelectiegroepId,
logischeBewerking, volgnummer, doelVoorwaardeId,
doelSelectieId, doelSelectiegroepId)`

`bronSelectieId` en `bronSelectiegroepId` sluiten elkaar uit, en geven respectievelijk aan bij welke selectie danwel selectiegroep het deel hoort.

`doelVoorwaardeId`, `doelSelectieId` en `doelSelectiegroepId` sluiten elkaar ook uit, en geven respectievelijk aan of het deel een voorwaarde, een selectie of een selectiegroep is.

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

`selectiedeel(id, bronSelectieId, bronSelectiegroepId,
logischeBewerking, volgnummer, doelVoorwaardeId,
doelSelectieId, doelSelectiegroepId)`

- `bronSelectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null toegelaten.
- `bronSelectiegroepId`: vreemde sleutel, verwijst naar `id` in `selectiegroep`, null toegelaten.
- `doelVoorwaardeId`: vreemde sleutel, verwijst naar `id` in `voorwaarde`, null toegelaten.
- `doelSelectieId`, vreemde sleutel, verwijst naar `id` in `selectie`, null toegelaten.
- `doelSelectiegroepId`, vreemde sleutel, verwijst naar `id` in `selectiegroep`, null toegelaten.

Resultaatset

Initieel bestaat de volgende relatie:

```
resultaatset(selectieId, datumtijd, totaalAantal,  
             nabewerking, relativiteit)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
resultaatset(id, selectieId, datumtijd, totaalAantal,  
             nabewerking, relativiteit)
```

- `selectieId`, vreemde sleutel, verwijst naar `id` in `selectie`, null niet toegelaten.

Recht

Tijdens pilot #1 is al tot de volgende relatie gekomen:

```
recht(id, gebruikersgroepId, gebruikerId)
```

- `gebruikersgroepId`: vreemde sleutel, verwijst naar `id` in `gebruikersgroep`, null toegelaten.
- `gebruikerId`: vreemde sleutel, verwijst naar `id` in `gebruiker`, null toegelaten.

Zie het pilotontwikkelplan van die iteratie van de fase Pilotontwikkeling voor meer informatie. Het moet nu ook mogelijk worden een recht voor een gebruiker of een gebruikersgroep specifiek te koppelen aan een resultaatset of selectie. Hiervoor worden twee vreemde sleutels toegevoegd waardoor het volgende ontstaat:

```
recht(id, gebruikersgroepId, gebruikerId, selectieId,  
      resultaatsetId)
```

- `gebruikersgroepId`: vreemde sleutel, verwijst naar `id` in `gebruikersgroep`, null toegelaten.
- `gebruikerId`: vreemde sleutel, verwijst naar `id` in `gebruiker`, null toegelaten.
- `selectieId`: vreemde sleutel, verwijst naar `id` in `selectie`, null toegelaten.
- `resultaatsetId`: vreemde sleutel, verwijst naar `id` in `resultaatset`, null toegelaten.

4.3

Relationeel implementatiemodel

Nu het relationeel representatiemodel er is, kan het relationeel implementatiemodel worden gemaakt. Dit gebeurt middels SQL als DDL.

Voorwaarde

Bij de relatie voorwaarde horen twee tabellen, `voorwaarde` en `voorwaarde_waarde`. Allereerst de tabel `voorwaarde`.

```
CREATE TABLE voorwaarde
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    omschrijving TEXT,
    brontype ENUM('tekenreeks', 'getal') NOT NULL,
    operatie ENUM('gelijk', 'ongelijk', 'komt voor', 'komt
        niet voor', 'groter dan', 'kleiner dan', 'groter dan
        of gelijk aan', 'kleiner dan of gelijk aan',
        'ongelijk aan', 'RE') NOT NULL,
    reguliereExpressie TINYTEXT,
    variabeleId MEDIUMINT UNSIGNED NOT NULL,
    selectieId MEDIUMINT UNSIGNED,
    PRIMARY KEY (id),
    FOREIGN KEY (variabeleId)
        REFERENCES variabele ON DELETE RESTRICT,
    FOREIGN KEY (selectieId)
        REFERENCES selectie ON DELETE CASCADE
);
```

En vervolgens de tabel `voorwaarde_waarde` voor de vastlegging van één of meerdere waarden behorende bij één voorwaarde.

```
CREATE TABLE voorwaarde_waarde
(
    voorwaardeId MEDIUMINT UNSIGNED NOT NULL,
    waarde TEXT,
    FOREIGN KEY (voorwaardeId)
        REFERENCES voorwaarde ON DELETE CASCADE
);
```

Selectiegroep

```
CREATE TABLE selectiegroep
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    naam TINYTEXT,
    herbruikbaar BOOL NOT NULL,
    omschrijving TEXT,
    PRIMARY KEY (id)
);
```

Selectie

Het selectie-object is bij het maken van het relationeel representatiemodel opgesplits in totaal vier tabellen. Allereerst de hoofdtabel selectie.

```
CREATE TABLE selectie
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    naam TINYTEXT NOT NULL,
    omschrijving TEXT,
    planning DATETIME,
    lidtMin DATE,
    lidtMax DATE,
    adtMin DATE,
    adtMax DATE,
    nabewerking ENUM('enkel groeperen',
        'dubbel groeperen', 'tellen',
        'groeperen en tellen', 'aantal'),
    relativiteit ENUM('ook', 'uitsluitend'),
    ontdubbelVariabeleId MEDIUMINT UNSIGNED,
    PRIMARY KEY (id),
    FOREIGN KEY (ontdubbelVariabeleId)
        REFERENCES variabele ON DELETE RESTRICT
);
```

Vervolgens de tabel voor de koppeling van een selectie met variabelen die niets te maken hebben met het selectieproces, maar die wel in de uitvoer moeten komen te staan.

```
CREATE TABLE selectie_uitvoer
(
    selectieId MEDIUMINT UNSIGNED NOT NULL,
    variabeleId MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (selectieId, variabeleId),
    FOREIGN KEY (selectieId)
        REFERENCES selectie ON DELETE CASCADE,
    FOREIGN KEY (variabeleId)
        REFERENCES variabele ON DELETE RESTRICT
);
```

Nu de tabel voor het aangeven van waarop een actie groepeert.

```
CREATE TABLE selectie_groepering
(
    selectieId MEDIUMINT UNSIGNED NOT NULL,
    variabeleId MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (selectieId, variabeleId),
    FOREIGN KEY (selectieId)
        REFERENCES selectie ON DELETE CASCADE,
    FOREIGN KEY (variabeleId)
        REFERENCES variabele ON DELETE RESTRICT
);
```

Om aan te geven welke acties meegenomen moeten worden bij het selectieproces, is de volgende tabel.

```
CREATE TABLE selectie_actie
(
    selectieId MEDIUMINT UNSIGNED NOT NULL,
    actieId SMALLINT UNSIGNED NOT NULL,
    PRIMARY KEY (selectieId, actieId),
    FOREIGN KEY (selectieId)
        REFERENCES selectie ON DELETE CASCADE,
    FOREIGN KEY (actieId)
        REFERENCES actie ON DELETE RESTRICT
);
```

Selectiedeel

```
CREATE TABLE selectiedeel
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    bronSelectieId MEDIUMINT UNSIGNED,
    bronSelectiegroepId MEDIUMINT UNSIGNED,
    logischeBewerking ENUM('en', 'of', 'xof', 'nen',
        'nof', 'nxof'),
    volgnummer SMALLINT UNSIGNED NOT NULL,
    doelVoorwaardeId MEDIUMINT UNSIGNED,
    doelSelectieId MEDIUMINT UNSIGNED,
    doelSelectiegroepId MEDIUMINT UNSIGNED,
    PRIMARY KEY (id),
    FOREIGN KEY (bronSelectieId)
        REFERENCES selectie ON DELETE CASCADE,
    FOREIGN KEY (bronSelectiegroepId)
        REFERENCES selectiegroep ON DELETE CASCADE,
    FOREIGN KEY (doelVoorwaardeId)
        REFERENCES voorwaarde ON DELETE RESTRICT,
    FOREIGN KEY (doelSelectieId)
        REFERENCES selectie ON DELETE RESTRICT,
    FOREIGN KEY (doelSelectiegroepId)
        REFERENCES selectiegroep ON DELETE RESTRICT
);
```

Resultaatset

```
CREATE TABLE resultaatset
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    selectieId MEDIUMINT UNSIGNED NOT NULL,
    datumtijd DATETIME NOT NULL,
    totaalAantal INT UNSIGNED NOT NULL,
    nabewerking BOOL NOT NULL,
    relativiteit ENUM('ook', 'uitsluitend'),
    PRIMARY KEY (id),
    FOREIGN KEY (selectieId)
        REFERENCES selectie ON DELETE CASCADE
);
```

Recht

Omdat de tabel `recht` al bestaat, en er slechts een aantal wijzigingen op doorgevoerd moeten worden, maken we gebruik van SQL's `ALTER TABLE` in plaats van `CREATE TABLE`.

```
ALTER TABLE recht
  ADD selectieId MEDIUMINT UNSIGNED,
  ADD resultaatsetId MEDIUMINT UNSIGNED,
  DROP PRIMARY KEY,
  ADD PRIMARY KEY (id, gebruikersgroepId, gebruikerId,
    selectieId, resultaatsetId);
```

5 GUI-ontwerp

In dit hoofdstuk komt het ontwerp van de GUI aan bod. Aan de hand van de al gemaakte schetsen in het pilotontwikkelplan van pilot #1, wordt hier een invulling gemaakt specifiek voor het beheer van selecties.

Allereerst komen er een aantal schermschetsen voorbij, vervolgens gevolgd door een navigatieschema.

5.1 Schermschetsen

In deze paragraaf staan de schermschetsen voor de volgende schermen:

- Overzichtscherm
- Toevoeg- & bewerk scherm
 - Overzichtscherm acties
 - Overzichtscherm herbruikbare groepen
 - Subwindow selectiedeel
- Doorgeven
- Overzichtscherm gebruikers
- Inplannen

Het overzichtscherm, zie Figuur 27, is het centrale punt rond het beheren van selecties. Het geeft een overzicht van alle aanwezige selecties, en vanuit hier kunnen selecties worden toegevoegd, bewerkt, verwijderd, doorgegeven en getest.

					Naam	Omschrijving
☐	B	U	D	I	~	~
☐	B	U	D	I	~	~
☐	B	U	D	I	~	~
☐						
☐	Selecteer alles					
☐	Selecteer niets					
☐	Inverteer selectie					
☐	Verwijderen					
☐	Toevoegen					

Figuur 27 Het overzicht van selecties.

De overzichtschermen voor acties, herbruikbare groepen en gebruikers worden geopend vanuit het toevoeg- & bewerkscherm, zie het navigatieschema in §5.2.

Figuur 28 vormt een schets van het scherm waarmee een selectie kan worden toegevoegd of bewerkt. Een aantal elementen verschijnt wel of niet, nagelang de keuze van bijvoorbeeld nabewerking.

Een element wat hier voor het eerst in het systeem naar voren komt zijn de vraagteken-icoontjes na de namen van een aantal formulervelden. Door hier met de muis boven te gaan hangen verschijnt er direct toegankelijke help over dat debetreffende veld.

Nacem
Omschrijving

Acties ?

Te lezen variabelen ?

Eerste profiel invoer ?
Laatste profiel invoer ?
Eerste antwoord ?
Laatste antwoord ?
Ontclubbelen op ?
Relativiteit ?
Nabewerking ?
Groepeer-variabele ?
Groepeer-variabele ?
Telvoorwaarden ?
→ 11 van 1022

Selectiedelen

Toevoegen
Verwijderen

Toevoegen
Verwijderen

→ indien recht
#10 aan-
wezig en
nabewerking=
geen

Kies
Kies

Toevoegen
Bewerken
Verwijderen

Omhoog
Omlaag
Bewerken
Verwijderen
Verschuiven

Nieuwe variabele
Nieuwe groep
Bestaande groep
Selectie

Figuur 28 Het toevoeg- & bewerkscherm.

Bij het toevoegen van acties opent zich een subwindow met een overzicht van acties, zie Figuur 29.

Bij de 'Kies'-knoppen achter het ontdebellen en het groeperen, openen een subwindow met hierin het overzicht van variabelen, zie §5.2 van het pilotontwikkelplan van pilot #2.

Bij het toevoegen van een bestaande groep als selectiedeel (een groep die in een andere selectie als herbruikbaar is gemarkeerd), opent zich ook een subwindow, maar nu met het overzicht van herbruikbare groepen, zie Figuur 30.

	Naam
☐	~~~~~
☐	~~~~~
☒	Selecteer alles
☐	Selecteer niets
☐	Inverteer selectie

Figuur 29 Het overzicht van acties.

	Naam	Omschrijving
☐	~~~~~	~~~~~
☐	~~~~~	~~~~~
☐	~~~~~	~~~~~
☒	Selecteer alles	
☐	Selecteer niets	
☐	Inverteer selectie	

Figuur 30 Het overzicht van herbruikbare groepen.

Bij het toevoegen of bewerken van een nieuw selectiedeel (een variabele, groep of selectie) opent zich een subwindow met de inhoud zoals deze in Figuur 31 is geschetst.

Hand-drawn sketch of a software interface for selecting variables and groups. The interface is divided into three horizontal sections by dashed lines.

Top Section:

- Label: *Selectiedeel*
- Field: *Variabele* (with a small box next to it)
- Field: *Omschrijving* (with a large text area below it)
- Field: *Bron type* (with a dropdown menu)
- Field: *Operatie* (with a dropdown menu)
- Field: *Waarde (n)* (with a large text area below it)
- Button: *Kies* (with an arrow pointing to *Variabele*)

Middle Section:

- Label: *Selectiedeel*
- Field: *Reguliere expressie* (with a small box next to it)
- Field: *Naam* (with a small box next to it)
- Field: *Omschrijving* (with a large text area below it)
- Field: *Herbruikbaar* (with a checked checkbox and an arrow pointing to *alleen indien niet herbruikt*)
- Button: *Details* (with an arrow pointing to *Naam*)

Bottom Section:

- Label: *Selectiedeel*
- Field: *Selectie* (with a small box next to it)
- Field: *Opstroom* (with a small box next to it)
- Field: *Beginwaarden* (with a small box next to it and an arrow pointing to *reload*)

Figuur 31 Selectiedeel.

Voor het doorgeven van een selectie wordt gebruik gemaakt van het scherm zoals deze in Figuur 32 is weergegeven. Bij het kiezen van een gebruiker wordt gebruik gemaakt van het scherm zoals deze in Figuur 33 is geschetst.

Figuur 32 Het doorgeef-scherm.

Figuur 33 Het overzicht van gebruikers.

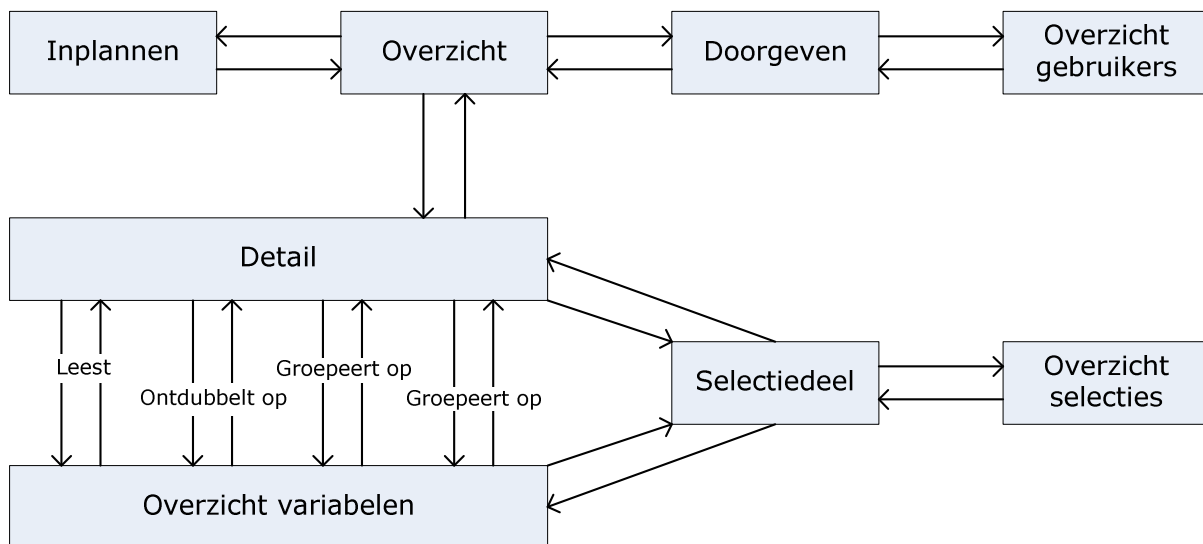
Bij het inplannen van een selectie, wordt gebruik gemaakt van het scherm zoals deze in Figuur 34 is weergegeven.

Figuur 34 Het inplan-scherm.

5.2

Navigatieschema

Hieronder staat het navigatieschema voor de schermen die in deze pilot zijn omschreven, zie Figuur 35.



Figuur 35 Navigatieschema.

6 Testplan

In dit hoofdstuk komt het testen aan bod. Er worden een aantal testscenario's samengesteld en uitgelegd. Tijdens de activiteit Beoordelen en testen moeten deze scenario's met goed gevolg gevolgd worden.

De scenario's zijn als volgt:

- Maak een nieuwe selectie aan met de volgende selectiedelen:
 - voorwaarde, brontype 'string', logische operatie 'en';
 - voorwaarde, brontype 'numeriek', logische operatie 'xof';
 - groep (herbruikbaar) met de volgende selectiedelen:
 - voorwaarde, logische operatie 'of'
 - voorwaarde.
- Maak een nieuwe selectie aan met hierin een groep en een andere selectie.
- Maak een nieuwe selectie aan en bewerk de eerder aangemaakte selectie om een kringverwijzing te krijgen.

15 Plan van Aanpak pilot #4

1 Inleiding

Dit is het Plan van Aanpak van pilot #4. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de iteratie van de fase Pilotontwikkeling voor de pilot Selectiebeheer en is deels gebaseerd op het pilotplan uit de fase Definitiestudie. Deze pilot moet uitkomen op het activeren selectie en het produceren van een resultaatset. De selecties zijn gemaakt met het inmiddels gemaakte selectiebeheer en de resultaatsets kunnen na activatie vervolgens worden uitgelezen, opgevraagd en gedownload middels functionaliteiten welke in de volgende en voorlaatste pilot gebouwd zullen gaan worden.

In hoofdstuk 2 worden de in deze iteratie van de fase Pilotontwikkeling uit te voeren activiteiten uiteengezet, tevens op basis van de voorlopige verdeling in pilotdelen die gemaakt is in de definitiestudie. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat dit zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
10 mei 2005	1.0	Initiële versie.

Tabel 32 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze iteratie van de fase Pilotontwikkeling te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaan komt. Als basis worden de activiteiten die zijn opgesomd in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE
Overig

Tabel 33 Legenda

- 1 Pilotplan bijwerken
- 2 Plan van Aanpak opstellen
- 3 Globaal-functionele structuur pilot specificeren
- 4 Globaal-technische structuur pilot specificeren
- 5 Testplan opstellen
- 6 Pilotontwikkelplan opstellen

- 7 Ontwerpen & bouwen van softwarebouweenheden, samenstellen handleidingen en opstellen invoeringsprocedures voor de volgende pilotdelen:
 - 1 Database
 - 2 Overzicht
 - 3 Selectieproces zelf
 - 4 Achterhalen berekende variabelen
 - 5 Groeperen en/of tellen
 - 6 Ontdubbelen
 - 7 Opslag in de database
 - 8 Voortgangsindicator
- 8 Beoordelen en testen pilot

Aanpassingen aan globaal Plan van Aanpak

De activiteiten 'SUMI-test opstellen' en 'GUI-ontwerp' worden overgeslagen omdat er geen of slechts een zeer beperkte GUI bestaat.

De activiteit 'Globaal organisatorische inrichting specificeren' wordt overgeslagen omdat er bij deze pilot geen organisatorische aspecten zijn te vermelden.

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 11 en eindigt halverwege week 14.

	Weken																			
	11					12					13					14				
Activiteit	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V
Pilotplan bijwerken																				
Plan van Aanpak opstellen																				
Functionele structuur																				
Technische structuur																				
Testplan opstellen																				
Pilotontwikkelplan opstellen																				
Ontwerpen, bouwen, samen- en opstellen																				
Database																				
Overzicht																				
Selectieproces zelf																				
Achterhalen berekende variabelen																				
Groeperen en/of tellen																				
Ontdubbelen																				
Opslag in de database																				
Voortgangsindicator																				
Beoordelen en testen																				

Tabel 34 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 2: Plan van Aanpak
- 6: Pilotontwikkelpunten met de uitkomsten van de punten van 3 t/m 5.
- Per softwarebouweenheid: de resultaten van het ontwerpen & bouwen, het samenstellen van de handleidingen het opstellen van de invoeringsprocedures.
- De pilot zelf (programmacode, databasestructuren, enzovoorts).

16 Pilotontwikkelplan pilot #4

1 Inleiding

Dit is het Pilotontwikkelplan van pilot #4. Het is voor het afstuderen in de periode 2005-1 welke geschied bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Plan van Aanpak behorende bij deze pilot. De definitiestudie is gebruikt als basis, en waar nodig in het gebied van deze pilot nog aangevuld en uitgebreid. Deze pilot moet uitkomen op het activeren selectie en het produceren van een resultaatset. De selecties zijn gemaakt met het inmiddels gemaakte selectiebeheer en de resultaatsets kunnen na activatie vervolgens worden uitgelezen, opgevraagd en gedownload middels functionaliteiten welke in de volgende en voorlaatste pilot gebouwd zullen gaan worden.

Hoofdstuk 2 bevat de functionele structuur. Dit omvat een detaillering van zaken uit het systeemconcept in de definitiestudie. Concreet voor deze pilot betekend dit een beschrijving van de functionaliteiten. Hoofdstuk 3 bevat de technische structuur. De relevante objecten van het klassendiagram worden hier omgezet naar een relationeel implementatiemodel en tevens wordt het selectieproces tot in detail op papier uitgewerkt. Hoofdstuk 4 bevat tot slot een testplan waarin een aantal testscenario's zijn vastgelegd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
10 mei 2005	1.0	Initiële versie.
19 mei 2005	1.1	Aanpassing aan klassendiagram: associatieklasse tussen Variabele en Resultaatset vervangen door normale klasse gelinkt aan Resultaatset.

Tabel 35 Veranderingengeschiedenis.

2 Functionele structuur

In dit hoofdstuk staat de functionele structuur beschreven. Het bestaat uit een tekstuele omschrijving van de functionele eisen aan het selectieproces.

2.1 Invoer

Een selectie bestaat uit een aantal gegevens die in zijn geheel invoer vormen voor het selectieproces. Een selectie kan zeer complex in

elkaar zitten. Zie voor de mogelijkheden het pilotontwikkelplan van pilot #3: Selectiebeheer.

2.2 Activatie

Het daadwerkelijk selecteren van de data uit de databases. Hierbij moet de invoer worden omgewerkt in een rauwe resultaatset met alle geselecteerde of later anderszinds benodigde gegevens. De variabelen zijn na de activatie al berekend, beperkt door de gestelde voorwaarden en ontdubbeld.

2.3 Nabewerking

Een selectie heeft een aantal opties aangaande nabewerking, namelijk:

- Enkel groeperen: hierbij wordt één variabele gebruikt om per unieke waarde van deze variabele het aantal resultaten te tellen. De uitvoer is eendimensionaal.
- Dubbel groeperen: hierbij worden twee variabelen gebruikt om per unieke combinatie van waarden van deze variabelen het aantal resultaten te tellen. De uitvoer is tweedimensionaal.
- Tellen: achterhalen hoe vaak een voorwaarde waar oplevert. Er kunnen meerdere telvoorwaarden per selectie worden gespecificeerd. De uitvoer is eendimensionaal.
- Tellen en groeperen: een combinatie tussen enkel groeperen en tellen. De uitvoer is tweedimensionaal.
- Aantal weergeven: geeft alleen het aantal resultaten weer.

Bij nabewerking is het ook mogelijk relatieve getallen te laten berekenen. Deze relatieve getallen kunnen dan óf samengaan met de absolute óf er voor in de plaats komen.

2.4 Uitvoer

De uitvoer van de nabewerkte data komt als resultaatset in de database te staan. In deze opslagplek zijn faciliteiten aanwezig voor het opslaan van nabewerkte selecties.

2.5 Voortgangsindicator

Omdat het selectieproces een langdurige aangelegenheid kan zijn, is er behoefte aan een voortgangsindicator welke nauwkeurig aan de gebruiker laat zien waar het selectieproces zich op dat moment bevindt.

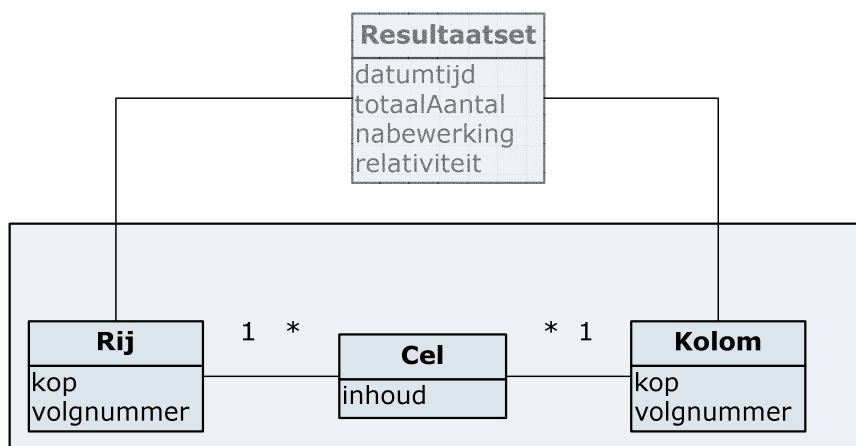
3 Technische structuur

In dit hoofdstuk staat de technische structuur van deze pilot. De relevante objecten van het klassendiagram worden omgezet in een relationeel representatiemodel en in een relationeel implementatiemodel.

Met het klassendiagram (of een deel ervan) zijn de objecten, attributen en relaties in kaart gebracht. In het proces tot het implementeren van de database, kan niet alles hieruit 1-op-1 worden overgenomen. De vertaalslag tussen het klassendiagram en het relationeel implementatiemodel vindt plaats in de tussenliggende activiteit van het samenstellen van een relationeel representatiemodel. Hierbij worden ook alle normalisaties behandeld. Vanuit het relationeel representatiemodel ligt de weg vrij tot het maken van een relationeel implementatiemodel.

3.1 Klassendiagram

Zie Figuur 26 voor het klassendiagram specifiek aan de resultaatsets.



Figuur 36 Het klassendiagram.

De gedimde klasse is reeds aanwezig, maar alsnog opgenomen vanwege nog toe te voegen relaties. Van reeds bestaande entiteiten waaraan nieuwe relaties moeten worden toegevoegd, worden alleen de wijzigingen op het relationeel representatiemodel gegeven. Datzelfde geldt voor het relationeel implementatiemodel waarvoor DDL wordt gebruikt om al bestaande structuren aan te passen (SQL's `ALTER TABLE`).

3.2 Relationeel representatiemodel

Zoals in de inleiding al vermeld, is dit relationeel representatiemodel een tussenstap tussen het klassendiagram en het implementatiemodel. In deze paragraaf wordt elk object apart behandeld met vastlegging van het verwerken van de correlaties en het normalisatieproces. Zie §3.1.2 van het pilotontwikkelplan van pilot #1 voor wat achtergrondinformatie rond het globaal gevolgde normalisatieproces en het toevoegen van extra numerieke primaire sleutels ondanks dat er één of meer kandidaatsleutels bestaan.

Rij

Initiëel bestaat de volgende relatie:

```
rij(resultaatsetId, kop, volgnummer)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
rij(id, resultaatsetId, kop, volgnummer)
```

- resultaatsetId, vreemde sleutel, verwijst naar id in resultaatset, null niet toegelaten.

Kolom

Initiëel bestaat de volgende relatie:

```
kolom(resultaatsetId, kop, volgnummer)
```

Na het toevoegen van een extra numeriek attribuut als primaire sleutel, krijgen we dit:

```
kolom(id, resultaatsetId, kop, volgnummer)
```

- resultaatsetId, vreemde sleutel, verwijst naar id in resultaatset, null niet toegelaten.

Cel

Initiëel bestaat de volgende relatie:

```
cel(rijId, kolomId, inhoud)
```

- rijId, vreemde sleutel, verwijst naar id in rij, null niet toegelaten.
- kolomId, vreemde sleutel, verwijst naar id in kolom, null niet toegelaten.

3.3 Relatieve implementatiemodel

Nu het relationeel representatiemodel er is, kan het relationeel implementatiemodel worden gemaakt. Dit gebeurt middels SQL als DDL.

Rij

Omdat de tabelnaam `rij` wat algemeen is en in de toekomst tot onduidelijkheden zou kunnen leiden, is ervoor gekozen de tabelnaam te beginnen met `rs_`, verwijzend naar de tabel `resultaatset`.

```
CREATE TABLE rs_rij
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    resultaatsetId MEDIUMINT UNSIGNED NOT NULL,
    kop TINYTEXT,
    volgnummer MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
```

```

        FOREIGN KEY (resultaatsetId)
        REFERENCES resultaatset ON DELETE CASCADE
    );

```

Kolom

Omdat de tabelnaam `kolom` ook wat algemeen is en in de toekomst tot onduidelijkheden zou kunnen leiden, is ervoor gekozen de tabelnaam te beginnen met `rs_`, verwijzend naar de tabel `resultaatset`.

```

CREATE TABLE rs_kolom
(
    id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    resultaatsetId MEDIUMINT UNSIGNED NOT NULL,
    kop TINYTEXT,
    volgnummer MEDIUMINT UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (resultaatsetId)
        REFERENCES resultaatset ON DELETE CASCADE
);

```

Cel

Omdat de tabelnaam `cel` ook wat algemeen is en in de toekomst tot onduidelijkheden zou kunnen leiden, is ervoor gekozen de tabelnaam te beginnen met `rs_`, verwijzend naar de tabel `resultaatset`.

```

CREATE TABLE rs_cel
(
    rijId MEDIUMINT UNSIGNED NOT NULL,
    kolomId MEDIUMINT UNSIGNED NOT NULL,
    inhoud TINYTEXT,
    PRIMARY KEY (rijId, kolomId),
    FOREIGN KEY (rijId)
        REFERENCES rs_rij ON DELETE CASCADE,
    FOREIGN KEY (kolomId)
        REFERENCES rs_kolom ON DELETE CASCADE
);

```

3.4

Selectieproces

Het selectieproces is opgedeeld in een aantal delen. Allereerst een beschrijving van de variabelestructuren die het proces gebruikt. In het hoofdproces staat vervolgens het selectieproces zelf voor een groot deel beschreven. In dit hoofdproces worden uitstapjes gemaakt naar het ontdebellen.

Structuren

In deze paragraaf worden een aantal variabelestructuren beschreven voor ge- en verbruik door het hoofdproces.

#1: Variabelen

```
$variabelen_normaal = array
(
    systeemnaam => databasenaam, [...]
);
$variabelen_profiel = idem
$variabelen_sponsor = idem
$variabelen_berekend = array
(
    systeemnaam => array
    (
        array(systeemnaam, [...]),
        phpcode
    ),
    [...]
)
```

#2: Data

De variabele voor tijdelijke opslag van de gegevens. Wordt gebruikt om de data direct na de selectie in op te slaan alvorens de voorwaarden er overheen worden gelegd.

```
$data = array
(
    profielId => array
    (
        'profielattributen' => array
        (
            systeemnaam => array
            (
                array(datum, waarde),
                [...]
            ),
            [...]
        ),
        'normale_variabelen' => array
        (
            systeemnaam => array
            (
                array(datum, waarde),
                [...]
            ),
            [...]
        ),
        'sponsorvariabelen' => array
        (
            systeemnaam => array
            (
                array(datum, waarde),
                [...]
            ),
            [...]
        ),
    ),
)
```

```
[..]
)
```

Met de punten tussen blokhaken wordt aangegeven dat het element wat erboven staat voor een onbepaald aantal keer herhaald kan worden.

Er vinden in de loop der tijd van het selectieproces twee mutaties plaats op deze structuur:

- 9 De array's met tijden en waarden worden vervangen door de waarde met de laatste datum, of anders gezegd: met het laatst gegeven antwoord. Dit alles natuurlijk nog wel met de scheiding tussen de profielen.
- 10 Onder de `profielId`-ingang komt een vierde element (naast `profielattributen`, `normale_variabelen` en `sponsorvariabelen`) voor opslag van de dan berekende berekende variabelen.

#3: Ontdubbelhulp

```
$dubbelen = array
(
  waarde => array
  (
    profile => datum,
    [..]
  ),
  [..]
)
```

Hoofdproces

In deze paragraaf wordt het hoofdproces beschreven.

- 1 Selectiespecificaties ophalen.
 - 1 Alle selectie-attributen.
 - 2 Een hiërarchische lijst met selectiedelen, inclusief die van de selecties die gelinkt zijn. Op dit punt dient een controle voor kringverwijzingen plaats te vinden.
 - 3 Aan de hand van 1 en 2: een invulling aan structuur #1 met alle te achterhalen variabelen.
- 2 Originele data selecteren: alle gekoppelde acties dooropen, per actie: alle deelnemers doorlopen (F1: de eerste opslagtabel, F2: de tabel `partnerid`) met inachtneming van `adtMin` en `adtMax`. Opslag van data gebeurd in structuur #2. Per deelnemer:
 - 1 Alle gewenste profielattributen opslaan met inachtneming van `lidtMin` en `lidtMax`. Als het profiel niet bestaat of als de datum van laatste activiteit buiten bereik ligt breken we de iteratie af.
 - 2 Controleren of het e-mailadres van de deelnemer niet voorkomt in de `unsub`-tabel. Als hij wel voorkomt breken we de iteratie af.

- 3 Alle gewenste normale variabelen achterhalen.
- 4 Alle gewenste sponsorvariabelen achterhalen.
- 3 Tijdsontdubbeling, structuurmutatie 2.1.
- 4 Berekende variabelen berekenen.
- 5 Voorwaarden toepassen. Structuur #2 wordt hierbij doorlopen waarbij per deelnemer de boom van voorwaarden tot een boolean wordt gereduceerd. Deelnemers voor wie deze boolean onwaar is, worden uit structuur #2 gehaald.
- 6 Eventueel extra ontdebelen: van alle profielen met voor de ontdebeldvariabele dezelfde variabele, blijft alleen het profiel over met het laatst gegeven antwoord. Hierbij wordt gebruik gemaakt van structuur #3.
- 7 Nabewerken: optioneel één van de volgende stappen:
 - 1 Groeperen op één variabele: structuur #2 doorlopen en de groepeer-variabele hier per deelnemer uithalen. Kijken of de waarde hiervan al bestaat in een associatieve array. Zo niet: aanmaken en de waarde op één zetten, zo wel: waarde met één verhogen. Aan het einde sorteren op de waarde van de variabele.
 - 2 Groeperen op twee variabelen: structuur #2 doorlopen en de groepeer-variabelen hier per deelnemer uithalen. Kijken of de eerste groepeer-variabele al bestaat in een associatieve array. Zo niet: aanmaken en de tweede groepeer-variabele aanmaken in een associatieve array van een niveau lager, zo wel: kijken of de waarde hiervan al bestaat in de associatieve array van een niveau lager. Zo niet: aanmaken, de waarde op één zetten en tegelijkertijd een tweede array vullen voor een overzicht van de waarden van deze tweede dimensie, zo wel: de waarde met één verhogen. Aan het einde sorteren op de waarden van de variabele.
 - 3 Tellen: structuur #2 doorlopen en elke telvoorwaarde evalueren. Indien de voorwaarde true oplevert: kijken of de naam ervan al voorkomt in een associatieve array. Zo niet: aanmaken en de waarde op één zetten, zo wel: waarde met één verhogen. Aan het einde sorteren op namen van de variabelen.
 - 4 Groeperen op één variabele en tellen: structuur #2 doorlopen en de groepeer-variabelen hier per deelnemer uithalen. Kijken of de eerste groepeer-variabele al bestaat in een associatieve array. Zo niet: aanmaken en de telvoorwaarde aanmaken in een associatieve array van een niveau lager, zo wel: kijken of de voorwaardenaam hiervan al bestaat in de associatieve array van een niveau lager. Zo niet: aanmaken en de waarde op één zetten, zo wel: de waarde met één verhogen. Aan het einde sorteren op de waarden van de variabele.
 - 5 Aantal weergeven: eigenlijk hoeft hier niets te gebeuren: de hoeveelheid elementen in structuur #2 is het getal wat we willen.

- 8 Opslaan. De gegevens in die na stap 5, 6 of 7 zijn overgebleven, worden hierbij opgeslag in de database onder een nieuwe resultaatset.

4 Testplan

In dit hoofdstuk komt het testen aan bod. Er worden een aantal testscenario's samengesteld en uitgelegd. Tijdens de activiteit Beoordelen en testen moeten deze scenario's met goed gevolg gevolgd worden.

De scenario's zijn als volgt:

- De selecties die zijn aangemaakt tijdens het testen van de vorige pilot, moeten allemaal met succes kunnen worden uitgevoerd.

17 Plan van Aanpak pilot #5

1 Inleiding

Dit is het Plan van Aanpak van pilot #5. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de iteratie van de fase Pilotontwikkeling voor de pilot Selectieactivatie en is deels gebaseerd op het pilotplan uit de fase Definitiestudie. Deze pilot moet uitkomen op het beheren van resultaatsets. De resultaatsets die zijn voortgekomen uit activatie van de selecties, kunnen worden bekeken, gedownload en verwijderd.

In hoofdstuk 2 worden de in deze iteratie van de fase Pilotontwikkeling uit te voeren activiteiten uiteengezet, tevens op basis van de voorlopige verdeling in pilotdelen die gemaakt is in de definitiestudie. Wanneer deze activiteiten uitgevoerd gaan worden is te vinden in hoofdstuk 3. Veel activiteiten leiden tot producten, en wat dit zijn staat in hoofdstuk 4 opgesomd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
10 mei 2005	1.0	Initiële versie.

Tabel 36 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze iteratie van de fase Pilotontwikkeling te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaan komt. Als basis worden de activiteiten die zijn opgesomd in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE

Tabel 37 Legenda

- 1 Pilotplan bijwerken
- 2 Plan van Aanpak opstellen
- 3 Globaal-functionele structuur pilot specificeren
- 4 GUI-ontwerp
 - 1 Schermschetsen
 - 2 Navigatieschema
- 5 Testplan opstellen
- 6 Pilotontwikkelplan opstellen

- 7 Ontwerpen & bouwen van softwarebouweenheden, samenstellen handleidingen en opstellen invoeringsprocedures voor de volgende pilotdelen:
 - 1 Bekijken
 - 2 Downloaden
 - 3 Verwijderen
 - 4 Doorgeven
- 8 Beoordelen en testen pilot

Aanpassingen aan globaal Plan van Aanpak

De activiteit 'SUMI-test opstellen' wordt overgeslagen omdat er slechts een zeer beperkte GUI bestaat. De activiteit 'Globaal organisatorische inrichting specificeren' wordt overgeslagen omdat er bij deze pilot geen organisatorische aspecten zijn te vermelden. De activiteit 'Globaal technische structuur specificeren' wordt overgeslagen omdat de database reeds volledig is omgezet er ook geen andere technische overwegingen te maken zijn.

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 14 en eindigt halverwege week 15.

Activiteit	Weken									
	14					15				
	M	D	W	D	V	M	D	W	D	V
Pilotplan bijwerken										
Plan van Aanpak opstellen										
Functionele structuur										
GUI-ontwerp										
Schermetschetsen										
Navigatieschema										
Testplan opstellen										
Pilotontwikkelplan opstellen										
Ontwerpen, bouwen, samen- en opstellen										
Overzicht										
Bekijken										
Downloaden										
Verwijderen										
Doorgeven										
Beoordelen en testen										

Tabel 38 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 2: Plan van Aanpak
- 7: Pilotontwikkelplan met de uitkomsten van de punten 3 t/m 6.

- Per softwarebouweenheid: de resultaten van het ontwerpen & bouwen, het samenstellen van de handleidingen en het opstellen van de invoeringsprocedures.
- De pilot zelf (programmacode, databasestructuren, enzovoorts).

18 Pilotontwikkelplan pilot #5

1 Inleiding

Dit is het Pilotontwikkelplan van pilot #5. Het is voor in het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na het Plan van Aanpak behorende bij deze pilot. De definitiestudie is gebruikt als basis, en is waar nodig in het gebied van deze pilot nog aangevuld en uitgebreid. Deze pilot moet uitkomen op mogelijkheden voor het beheren van resultaatsets. Hieronder wordt verstaan het bekijken, downloaden en verwijderen.

Hoofdstuk 2 bevat de functionele structuur. Dit omvat een detaillering van zaken uit het systeemconcept in de definitiestudie. Concreet voor deze pilot betekent dit een beschrijving van de functionaliteiten. Hoofdstuk 3 bevat een GUI-ontwerp in de vorm van schermschetsen en een navigatieschema en hoofdstuk 4 bevat tot slot een testplan waarin een aantal testscenario's zijn vastgelegd.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
07-04-2005	1.0	Initiële versie.

Tabel 39 Veranderingengeschiedenis.

2 Functionele structuur

In dit hoofdstuk staat de functionele structuur beschreven. Het bestaat uit een tekstuele omschrijving van de functionele eisen aan het bekijken, downloaden en verwijderen van variabelen. Er is overal rekening gehouden met de mogelijkheden van de GUI-standaard van pilot #1.

2.1 Overzicht

Het kunnen oproepen van een overzicht van alle op dat moment bij het systeem bekende resultaatsets. Van elke resultaatset wordt weergegeven wat de naam van de selectie is, wat de datum van generatie is, hoeveel resultaten erin zitten, of de resultaatset bewerkt is en welk soort van relativiteit is toegevoegd.

2.2 **Bekijken**

Resultaatsets worden gegenereerd door het activeren van een selectie. Van één selectie kunnen meerdere resultaatsets bestaan. Bij het bekijken van een resultaatset worden de daadwerkelijke resultaten op het scherm van de gebruiker getoond. Voor resultaatsets met een significante hoeveelheid cellen zal voor het bekijken een bevestiging worden gevraagd.

2.3 **Downloaden**

Naast dat een resultaatset kan worden bekeken, is het ook mogelijk een resultaatset als CSV-bestand te downloaden om te openen of te importeren in een extern programma. De rijen en kolommen zullen exact gelijk zijn aan de rijen en kolommen zoals die naar voren komen bij het regulier bekijken.

2.4 **Verwijderen**

Een resultaatset kan ten alle tijden verwijderd worden, zover de rechten dat toelaten.

2.5 **Doorgeven**

Bij het doorgeven wordt de resultaat beschikbaar gemaakt voor een andere gebruiker. Op deze manier kunnen gebruikers die hier normaliter geen mogelijkheden voor hebben, toch de beschikking hebben over resultaatset uitkomende uit complexere of uitgebreidere selecties. Bij het doorgeven moeten een aantal rechten gegeven worden die gaan gelden voor dat object gekoppeld aan een specifieke gebruiker.

3 **GUI-ontwerp**

In dit hoofdstuk komt het ontwerp van de GUI aan bod. Aan de hand van de al gemaakte schetsen in het pilotontwikkelplan van pilot #1, wordt hier een invulling gemaakt specifiek voor het beheer van de selecties.

Allereerst komen er een aantal schermschetsen voorbij, vervolgens gevolgd door een navigatieschema.

3.1 **Schermschetsen**

In deze paragraaf staan de schermschetsen voor de volgende schermen:

- Overzichtscherf
- Doorgeven

Het overzichtscherf van gebruikers wordt herbruikt vanuit pilot #3, Selectiebeheer.

Het overzichtsscherm, zie Figuur 37, is het centrale punt rond het beheren van resultaatsets. Het geeft een overzicht van alle aanwezige sets, en vanuit hier kunnen selecties worden bekeken, gedownload, doorgegeven en verwijderd.

			Naam	Datum/tijd	Aantal	Nabewerkt	Relativiteit
☐	B	D					
☐	B	D					
☐							
			Selecteer alles				
☒			Selecteer niets				
			Inverteer selectie				
☒			Verwijderen				

Figuur 37 Het overzicht van resultaatsets.

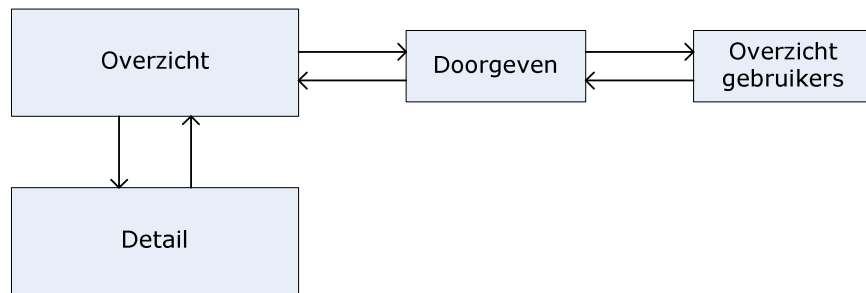
Voor het doorgeven van een resultaatset wordt gebruik gemaakt van het scherm zoals deze in Figuur 38 is weergegeven. Bij het kiezen van een gebruiker wordt gebruik gemaakt van Figuur 8 uit het pilotontwikkelplan van pilot #3.

Gebruiker	☐ <u>niets</u>
Rechten	☐ Behouden ☐ Downloaden ☐ Verwijderen ☐ Doorgeven
	Opslaan Bevestigwaarden

Figuur 38 Het doorgeef-scherm.

3.2 Navigatieschema

Hieronder staat het navigatieschema voor de schermen die in deze pilot zijn omschreven, zie Figuur 39.



Figuur 39 Navigatieschema.

4 Testplan

In dit hoofdstuk komt het testen aan bod. Er worden een aantal testscenario's samengesteld en uitgelegd. Tijdens de activiteit Beoordelen en testen moeten deze scenario's met goed gevolg gevolgd worden.

De scenario's zijn als volgt:

- De resultaatsets die zijn ontstaan na activatie van de selecties die zijn aangemaakt tijdens het testen van pilot #3, moeten allemaal een correcte representatie zijn van de werkelijkheid. Dit moet zowel blijken uit het bekijken van een resultaatset, maar ook uit de CSV-bestanden die kunnen worden gedownload.

19 Plan van Aanpak fase Invoering

1 Inleiding

Dit is het Plan van Aanpak voor de fase invoering. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Dit document komt na de laatste iteratie van de fase Pilotontwikkeling en aan het einde van het project en het beschrijft de werkzaamheden en de planning hiervan.

In hoofdstuk 2 worden de activiteiten uiteengezet, in hoofdstuk 3 zijn deze activiteiten gepland, en in hoofdstuk 4 staan de producten vermeld.

Veranderingengeschiedenis

In de loop der tijd heeft dit document een aantal veranderingen ondergaan. De versienummers, gekoppeld aan een datum staan met een omschrijving van de veranderingen in Tabel 1.

Datum	Versie	Omschrijving
3 juni 2005	1.0	Initiële versie.

Tabel 40 Veranderingengeschiedenis.

2 Uit te voeren activiteiten

In dit hoofdstuk staan de activiteiten die voor deze fase Invoering te volgen zijn. Zie Tabel 3 voor een legenda om te kunnen zien waar welke activiteit vandaan komt. Als basis worden de activiteiten die zijn opgesoms in het algemene Plan van Aanpak genomen, maar uitzonderingen zijn goed mogelijk.

Legenda
IAD
GUIDE
Overig

Tabel 41 Legenda

- 1 Plan van Aanpak opstellen
- 2 SUMI-test uitvoeren
- 3 Pilots invoeren
- 4 Pilotacceptatie uitvoeren
- 5 Pilots overdragen
- 6 Feedback verzamelen
- 7 Pilots ondersteunen

3 Planning

In dit hoofdstuk staat de planning van de hierboven opgesomde activiteiten. Deze iteratie van de fase Pilotontwikkeling begint halverwege week 9 en eindigt halverwege week 11.

Activiteit	Weken														
	15					16					17				
	M	D	W	D	V	M	D	W	D	V	M	D	W	D	V
Plan van Aanpak opstellen															
SUMI-test uitvoeren															
Pilots invoeren															
Pilotacceptatie uitvoeren															
Pilots overdragen															
Feedback verzamelen															
Pilots ondersteunen															

Tabel 42 De planning.

4 Producten

In dit hoofdstuk worden alle concrete producten beschreven die voortkomen uit bovenstaande activiteiten. Er wordt waar nodig verwezen naar de activiteiten in bovenstaand overzicht.

- 1: Plan van Aanpak
- SUMI-testscenario's
- SUMI-resultaten

20 Voorbereiding SUMI-test

1 Inleiding

Dit is de SUMI-test voor de fase invoering. Het is voor het afstuderen in de periode 2005-1 welke geschiedt bij het bedrijf E2Ma aan de opleiding Informatica & Informatiekunde van de Haagse Hogeschool en in de richting VIA (Vormgeving en Ontwerp van Interactie).

Tijdens de iteraties van de fase Pilotontwikkeling, zijn voor een aantal pilots al een aantal scenario's opgesteld. Voor de SUMI-test worden al deze scenario's hier overgenomen zodat ze doorlopen kunnen worden. Aan het einde van de SUMI-test dient een vragenlijst ingevuld te worden voor de beoordeling van de gebruikersvriendelijkheids-aspecten.

2 SUMI-test

Vorbereidende procedure

Het systeem bevindt zich tijdelijk onder het volgende adres:
<http://192.168.0.3/backoffice/selectietool/login.php>. Inloggen kan aldaar met sumi en sumi als gebruikersnaam en wachtwoord.

Variabele toevoegen

Er moet een nieuwe normale variabele worden toegevoegd. Voor het kiezen van de databasenaam moet gebruik gemaakt worden van de wizard (achter de knop 'Kies') waarna er een willekeurige variabele van een willekeurige actie gekozen moet worden.

Variabele wijzigen

Er moet een berekende variabele gewijzigd worden; hierbij moet de PHP-code op een dusdanige manier veranderd worden dat er van een extra variabele gecontroleerd wordt of deze een bepaalde waarde heeft.

Klik hiervoor op het hamertje voor de variabele met de naam 'SUMI-var'. Kies de knop 'Toevoegen', schakel het selectievakje voor de variabele met de naam 'Geboortejaar' in, druk op 'Selecteren' en voeg aan de PHP-code voor het laatste)-haakje het volgende toe:

```
&& $Geboortejaar == 1950
```

Selectie toevoegen

Er moet een nieuwe selectie worden toegevoegd met drie voorwaarden die aan elkaar gekoppeld zijn middels een logische 'en'.

Selectie wijzigen

De zojuist aangemaakte selectie moet worden gewijzigd: er moet, wederom met een logische 'en' ook gezocht worden op twee andere voorwaarden in een groep die aan elkaar zijn gekoppeld middels een logische 'of'. Selecteer de selectiegroep na het aanmaken ervan om er voorwaarden aan toe te voegen.

Selectie aanmaken

Er moet een selectie worden aangemaakt met de laatst gewijzigde selectie als selectiedeel voor een nieuwe selectie.